Electronic Transactions on Numerical Analysis.
Volume 33, pp. 53-62, 2009.
Copyright © 2009, Kent State University.
ISSN 1068-9613.

AN EXTENDED BLOCK ARNOLDI ALGORITHM FOR LARGE-SCALE SOLUTIONS OF THE CONTINUOUS-TIME ALGEBRAIC RICCATI EQUATION[*]

M. HEYOUNI[†] AND K. JBILOU[†]

Abstract. We present a new iterative method for the computation of approximate solutions to large-scale continuous-time algebraic Riccati equations. The proposed method is a projection method onto an extended block Krylov subspace, which can be seen as a sum of two block Krylov subspaces in A and A^{-1}. We give some theoretical results and present numerical experiments for large and sparse problems. These numerical tests show the efficiency of the proposed scheme as compared to the block Arnoldi and Newton-ADI methods.

Key words. Block Arnoldi; Extended block Krylov; Low rank; Riccati equations.

AMS subject classifications. 65F10, 65F30

1. Introduction.

This paper presents a new iterative method for the numerical solution of the continuous-time algebraic Riccati equation (CARE) of the form

$$A^TX + XA - XBB^TX + C^TC = 0, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsingular, $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{s\times n}$. The matrices B and C are assumed to be of full rank with $p \ll n$, $s \ll n$.

Riccati equations play a fundamental role in many areas, such as control, filter design, model reduction, differential equations, and robust control [2, 3, 9, 14, 22, 25, 33]. For historical developments, applications and importance of algebraic Riccati equations, we refer to [1, 9, 13] and the references therein. Usually, in these applications, the so-called stabilizing solution of (1.1) is desired. Such a solution X is symmetric positive semidefinite and has the property that the eigenvalues of the resulting closed-loop matrix $A - BB^TX$ are in the open complex left-half plane (i.e., each eigenvalue of the matrix $A - BB^TX$ has a negative real part). The stabilizing solution exists and is unique under certain assumptions on the problem [12, 22].

Let $x(t) \in \mathbb{R}^n$ be the state vector, $u(t) \in \mathbb{R}^p$ be the control vector, and $y(t) \in \mathbb{R}^s$ be the output vector. We consider the following problem: minimize

$$J(x_0, u) = \frac{1}{2}\int_0^{+\infty} \left(y(t)^T\, y(t) + u(t)^T\, u(t)\right)\, dt \tag{1.2}$$

under the dynamic constrains

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \text{ with } x(0) = x_0, \\ y(t) = Cx(t). \end{cases} \tag{1.3}$$

Under the hypotheses that the pair (A, B) is stabilizable (i.e., there is a matrix S such that $A - BS$ is stable) and the pair (C, A) is detectable (i.e., (A^T, C^T) stabilizable), a unique optimal solution $\bar{u}$ that minimizes the functional $J(x_0, u)$ exists [32], and it can be determined through a feedback operator K, such that $\bar{u}(t) = Kx(t)$, where $K = -B^TX$ and $X \in \mathbb{R}^{n\times n}$ is the unique symmetric positive semidefinite and stabilizing solution of the matrix equation (1.1).

[*] Received January 14, 2008. Accepted July 23, 2008. Published online March 31, 2009. Recommended by Yousef Saad.

[†] L.M.P.A, Université du Littoral Côte d'Opale, 50 rue F. Buisson BP. 699, F-62228 Calais Cedex, France. E-mail: heyouni@lmpa.univ-littoral.fr; jbilou@lmpa.univ-littoral.fr.

The unique stabilizing solution of the CARE equation (1.1) can be obtained by considering $\mathbb{S}_{\mathcal{H}}$, the n-dimensional stable invariant subspace (i.e., the subspace corresponding to the eigenvalues of $\mathcal{H}$ in the open left half plane) of the corresponding Hamiltonian matrix (see [23, 25])

$$\mathcal{H} = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix} \in \mathbb{R}^{2n\times 2n}.$$

If $\mathbb{S}_{\mathcal{H}}$ is spanned by the columns of $\left(X_1^T, X_2^T\right)^T \in \mathbb{R}^{2n\times n}$ and X_1 is nonsingular, then $X = -X_2\, X_1^{-1}$ is the stabilizing solution of (1.1).

During the last decades, many numerical methods for solving the CARE (1.1) with small and dense matrices have been developed. The standard computational methods are based on the Schur and structure-preserving Schur methods [23, 31, 10, 24, 25], matrix sign function methods [5, 11, 20, 29], Newton-type methods [21, 4, 22, 6, 16] and the symplectic Lanczos method [7].

Generally, the matrices A, B and C are obtained from the discretization of operators defined on infinite-dimensional subspaces. Moreover, the matrix A is in general sparse, banded and very large. For such problems, only a few attempts have been made to solve (1.1). The well-known Low Rank Cholesky Factorized Newton (LRCF-Newton) method [28] is one of the widely used methods for solving (1.1). At each step of the outer Newton iteration, one needs to solve a large Lyapunov matrix equation. In the LRCF-Newton method, these Lyapunov matrix equations are solved by the Low Rank Cholesky Factorized method [28], which is based on the solution of linear systems with shifted matrices $A - \mu_i I$ where the μ_i are the ADI parameters. The determination of the “optimal” ADI parameters and the computation of the approximate solution of the Lyapunov equations increase the memory requirements and the CPU time of the LRCF-Newton method.

Projection methods on block Krylov subspaces, using the block Arnoldi process, and on matrix Krylov subspaces, using the global Arnoldi process, also have been applied to compute low rank approximate solutions to large and sparse CAREs [17, 18, 19]. However, these methods usually need many iterations (large projection subspaces) to produce an accurate approximate solution, and this increases considerably the CPU time and the memory requirements.

To remedy the drawbacks of the LRCF-Newton and the block or global Arnoldi algorithms, we present a new projection method that allows us to compute low rank approximations to the stabilizing solution of (1.1). We project the initial problem onto an extended block Krylov subspace generated by the matrices A and A^{-1}, and we obtain a low-dimensional CARE that is solved by a standard algorithm such as the Schur method [23]. The extended block Krylov subspace is generated by means of the extended block Arnoldi process first introduced in [15] and used for solving Lyapunov equations in [30]. We also give new theoretical results, such as an upper bound for the norm of the error and a perturbation result.

The remainder of the paper is organized as follows. In Section 2, we present the extended block Arnoldi algorithm and give some properties. In Section 3, we show how to extract low rank approximate solutions to CAREs and give some theoretical results, such as an expression for the norm of the residual and an upper bound for the norm of the error. Section 4 is devoted to some numerical examples and comparisons with other methods.

Throughout this paper, we use the following notation. The 2-norm and the Frobenius norm of matrices will be denoted by $\|\cdot\|$ and $\|\cdot\|_F$, respectively. The separation between two matrices A and B of dimension $n\times n$ and $p\times p$, respectively, is given by $\mathrm{sep}(A,B) = \min_{\|X\|=1} \|AX - XB\|$. Finally, I_r and $O_{r\times l}$ will denote the identity of size $r\times r$ and the zero matrix of size $r\times l$, respectively.

2. The extended block Arnoldi algorithm.

We first recall the extended block Arnoldi process applied to the pair (F, G), where $F \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{n\times s}$. The projection subspace $\mathcal{K}_k(F, G)$ of $\mathbb{R}^n$ that we will consider was introduced in [15, 30]:

$$\mathcal{K}_k(F, G) = \text{Range}([G, F^{-1}G, FG, F^{-2}G, F^2G, \ldots, F^{-(k-1)}G, F^{k-1}G]).$$

Note that the subspace $\mathcal{K}_k(F, G)$ is a sum of two block Krylov subspaces,

$$\mathcal{K}_k(F, G) = \mathbb{K}_k(F, G) + \mathbb{K}_k(F^{-1}, F^{-1}G),$$

where $\mathbb{K}_k(F, G) = \text{Range}([G, FG, \ldots, F^{k-1}G])$. The following algorithm allows us to compute an orthonormal basis of the extended Krylov subspace $\mathcal{K}_k(F, G)$. This basis contains information on both F and F^{-1}. Let m be some fixed integer which limits the dimension of the constructed basis. The extended block Arnoldi process is described as follows:

ALGORITHM 2.1. **The extended block Arnoldi algorithm (EBA).**

```
• Inputs: $F$ an $n \times n$ matrix, $G$ an $n \times s$ matrix and $m$ an integer.
• Step 0. Compute the QR decomposition of $[G, F^{-1}G]$, i.e., $[G, F^{-1}G] = V_1\Lambda$;
          Set $\mathcal{V}_0 = [\,]$;
• Step 1. For $j = 1, \ldots, m$
• Step 1.1   Set $V_j^{(1)}$: first $s$ columns of $V_j$; $V_j^{(2)}$: second $s$ columns of $V_j$
• Step 1.2   $\mathcal{V}_j = [\mathcal{V}_{j-1}, V_j]$; $\hat{V}_{j+1} = \left[FV_j^{(1)}, F^{-1}V_j^{(2)}\right]$.
• Step 1.3   Orthogonalize $\hat{V}_{j+1}$ with respect to $\mathcal{V}_j$ to get $V_{j+1}$, i.e.,
               For $i = 1, 2, \ldots, j$
                   $H_{i,j} = V_i^T\hat{V}_{j+1}$;
                   $\hat{V}_{j+1} = \hat{V}_{j+1} - V_iH_{i,j}$;
               end for
• Step 1.4   Compute the QR decomposition of $\hat{V}_{j+1}$, i.e., $\hat{V}_{j+1} = V_{j+1}H_{j+1,j}$.
          end for.
```

Since the above algorithm involves implicitly a Gram-Schmidt process, the block vectors $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$ ($V_i \in \mathbb{R}^{n\times 2s}$) have mutually orthogonal columns, provided none of the upper triangular matrices $H_{j+1,j}$ are rank deficient. Hence, after m steps, Algorithm 2.1 builds an orthonormal basis $\mathcal{V}_m$ of the Krylov subspace $\mathcal{K}_{m+1}(F, G)$ and a block upper Hessenberg matrix H_m whose nonzero blocks are the $H_{i,j}$. Note that each submatrix $H_{i,j}$ ($1 \leq i \leq j \leq m$) is of order $2s$.

Let $\mathcal{T}_m \in \mathbb{R}^{2ms\times 2ms}$ be the restriction of the matrix F to the extended Krylov subspace $\mathcal{K}_m(F, G)$, i.e., $\mathcal{T}_m = \mathcal{V}_m^T F\mathcal{V}_m$. It is shown in [30] that $\mathcal{T}_m$ is also block upper Hessenberg with $2s \times 2s$ blocks. Moreover, a recursion is derived to compute $\mathcal{T}_m$ from H_m without requiring matrix-vector products with F. For more details on how to compute $\mathcal{T}_m$ from H_m, we refer to [30]. We note that for large problems the inverse of the matrix F is not computed explicitly, and in this case we can use iterative solvers with preconditioners to solve linear systems with F. However, when these linear systems are not solved accurately, the theoretical properties of the extended block Arnoldi process are no longer valid. Next, we give some properties that will be useful later.

PROPOSITION 2.2. *Let $\bar{\mathcal{T}}_m = \mathcal{V}_{m+1}^T F\mathcal{V}_m$, and suppose that m steps of Algorithm 2.1 have been carried out. Then we have*

$$\begin{aligned} F\mathcal{V}_m &= \mathcal{V}_{m+1}\bar{\mathcal{T}}_m && (2.1)\\ &= \mathcal{V}_m\mathcal{T}_m + V_{m+1}T_{m+1,m}E_m^T. && (2.2)\end{aligned}$$

where $T_{i,j}$ is the $2s \times 2s$ (i, j) block of $\mathcal{T}_m$ and $E_m = [O_{2s\times 2(m-1)s}, I_{2s}]^T$ is the matrix of the last $2s$ columns of the $2ms \times 2ms$ identity matrix I_{2ms}.

Proof. Since $\mathcal{V}_{m+1} = [\mathcal{V}_m, V_{m+1}]$, we have

$$\begin{aligned} \mathcal{T}_{m+1} &= \mathcal{V}_{m+1}^T F \mathcal{V}_{m+1} \\ &= \begin{bmatrix} \mathcal{V}_m^T F \mathcal{V}_m & \mathcal{V}_m^T F V_{m+1} \\ V_{m+1}^T F \mathcal{V}_m & V_{m+1}^T F V_{m+1} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{T}_m & \mathcal{V}_m^T F V_{m+1} \\ V_{m+1}^T F \mathcal{V}_m & V_{m+1}^T F V_{m+1} \end{bmatrix}. \end{aligned}$$

Now, as $\mathcal{T}_{m+1}$ is block upper Hessenberg, we have $V_{m+1}^T F \mathcal{V}_m = T_{m+1,m} E_m^T$, and

$$\bar{\mathcal{T}}_m = \mathcal{V}_{m+1}^T F \mathcal{V}_m = \begin{bmatrix} \mathcal{T}_m \\ T_{m+1,m} E_m^T \end{bmatrix}.$$

Using the fact that $F\mathcal{K}_m \subseteq \mathcal{K}_{m+1}$ and that $\mathcal{V}_{m+1}$ is orthogonal, it follows that there exists a matrix L such that $F\mathcal{V}_m = \mathcal{V}_{m+1} L$. Hence $\mathcal{V}_{m+1}^T F \mathcal{V}_m = L$, which shows that $L = \bar{\mathcal{T}}_m$. Therefore, $F\mathcal{V}_m = \mathcal{V}_{m+1} \bar{\mathcal{T}}_m$. □

3. Low rank approximate solutions to large CAREs. In this section, we will see how to extract low rank approximate solutions to the continuous-time algebraic Riccati equation (1.1). We project the initial problem onto the extended block Krylov subspace $\mathcal{K}_m(A^T, C^T)$. Applying the extended block Arnoldi process (Algorithm 2.1) to the pair (A^T, C^T) gives us an orthonormal basis $\{V_1, \ldots, V_m\}$ of the extended block Krylov subspace $\mathcal{K}_m(A^T, C^T)$. We consider low-rank approximate solutions that have the form

$$X_m = \mathcal{V}_m Y_m \mathcal{V}_m^T, \tag{3.1}$$

where $\mathcal{V}_m = [V_1, \ldots, V_m]$ and $Y_m \in \mathbb{R}^{2ms \times 2ms}$. Note that $\mathrm{rank}(X_m) \leq 2ms$.

We note that the block Arnoldi algorithm also produces low rank approximate solutions of the form (3.1). However, the last method usually needs many iterations to give an accurate approximation, which increases the CPU time. The rational part of the extended block Arnoldi method is considered here as an acceleration procedure.

From now on, the matrix $\mathcal{T}_m$ is defined by $\mathcal{T}_m = \mathcal{V}_m^T A^T \mathcal{V}_m$. Using the expression (3.1) in the matrix equation (1.1), multiplying on the left by $\mathcal{V}_m^T$ and on the right by $\mathcal{V}_m$, we get the low-dimensional continuous-time algebraic Riccati equation

$$\mathcal{T}_m Y_m + Y_m \mathcal{T}_m^T - Y_m \tilde{B}_m \tilde{B}_m^T Y_m + \tilde{C}_m^T \tilde{C}_m = 0, \tag{3.2}$$

with $\tilde{B}_m = \mathcal{V}_m^T B$, $\tilde{C}_m^T = \mathcal{V}_m^T C^T = \mathcal{E}_1 \Lambda_{1,1}$, where $\mathcal{E}_1 = [I_s, O_{s\times(2m-1)s}]^T$ is the matrix of the first s columns of the $2ms \times 2ms$ identity matrix I_{2ms} and $\Lambda_{1,1}$ is the $s \times s$ matrix obtained from the QR decomposition

$$[C^T, A^{-T} C^T] = V_1 \Lambda \;\text{ with }\; \Lambda = \begin{bmatrix} \Lambda_{1,1} & \Lambda_{1,2} \\ 0 & \Lambda_{2,2} \end{bmatrix}. \tag{3.3}$$

We assume that the projected algebraic Riccati equation (3.2) has a unique symmetric positive semidefinite and stabilizing solution Y_m. This solution can be obtained by a standard direct method such as the Schur method [23].

If m steps of Algorithm 2.1 are applied to the pair (A^T, C^T), then using the results of Proposition 2.2 we have

$$A^T \mathcal{V}_m = \mathcal{V}_m \mathcal{T}_m + V_{m+1} T_{m+1,m} E_m^T. \tag{3.4}$$

Now, multiplying the reduced-order continuous-time algebraic Riccati equation (3.2) on the left by $\mathcal{V}_m$ and on the right by $\mathcal{V}_m^T$ and using (3.4), we get

$$\begin{aligned}\left[A^T\mathcal{V}_m - V_{m+1}T_{m+1,m}E_m^T\right]Y_m\mathcal{V}_m^T + \mathcal{V}_mY_m\left[A^T\mathcal{V}_m - V_{m+1}T_{m+1,m}E_m^T\right]^T\\ -\mathcal{V}_mY_m\mathcal{V}_m^TBB^T\mathcal{V}_mY_m\mathcal{V}_m^T + C^TC = 0.\end{aligned}$$

Setting $F_m = V_mT_{m+1,m}^TV_{m+1}^T$ and using the relation

$$V_{m+1}T_{m+1,m}E_m^TY_m\mathcal{V}_m^T = V_{m+1}T_{m+1,m}E_m^T\mathcal{V}_m^TX_m,$$

it follows that

$$(A - F_m)^TX_m + X_m(A - F_m) - X_mBB^TX_m + C^TC = 0. \tag{3.5}$$

The matrix equation (3.5) shows that the approximation X_m is an exact solution of a perturbed continuous-time algebraic Riccati equation.

Let $R_m = A^TX_m + X_mA - X_mBB^TX_m + C^TC$ be the residual associated with the mth approximate solution and let ϵ be some fixed tolerance. The computation of X_m (and of R_m) becomes expensive as m increases. Therefore, in order to stop the iterations, one has to test if $\|R_m\| < \epsilon$ without computing extra products involving the matrix A. The next result shows how to compute the residual norm of R_m without computing the approximation X_m, which is computed in a factored form only when convergence is achieved.

THEOREM 3.1. *Let $X_m = \mathcal{V}_mY_m\mathcal{V}_m^T$ be the approximation obtained at step m by the extended block Arnoldi-CARE method and let Y_m be the symmetric positive semi-definite stabilizing solution of the low-dimensional CARE (3.2). Then the residual R_m satisfies*

$$\|R_m\| = \|T_{m+1,m}\tilde{Y}_m\|, \tag{3.6}$$

where $\tilde{Y}_m$ is the $2s \times 2ms$ matrix corresponding to the last $2s$ rows of Y_m.

Proof. From the relations (3.1) and (3.2), we have

$$R_m = A^T\mathcal{V}_mY_m\mathcal{V}_m^T + \mathcal{V}_mY_m\mathcal{V}_m^TA - \mathcal{V}_mY_m\mathcal{V}_m^TBB^T\mathcal{V}_mY_m\mathcal{V}_m^T + C^TC.$$

Using (3.4) and the fact that $C^T = V_1^{(1)}\Lambda_{1,1}$, where $V_1^{(1)}$ is the matrix of the first s columns of V_1 and $\Lambda_{1,1}$ is defined in (3.3), we get

$$\begin{aligned}R_m &= (\mathcal{V}_m\mathcal{T}_m + V_{m+1}T_{m+1,m}E_m^T)Y_m\mathcal{V}_m^T + \mathcal{V}_mY_m(\mathcal{T}_m^T\mathcal{V}_m^T + E_mT_{m+1,m}^TV_{m+1}^T)\\ &\quad -\mathcal{V}_mY_m\tilde{B}_m\tilde{B}_m^TY_m\mathcal{V}_m^T + V_1^{(1)}\Lambda_{1,1}\Lambda_{1,1}^T{V_1^{1)}}^T\\ &= \mathcal{V}_{m+1}\begin{bmatrix}\mathcal{T}_mY_m - Y_m\tilde{B}_m\tilde{B}_m^TY_m\\ T_{m+1,m}E_m^TY_m\end{bmatrix}\mathcal{V}_m^T +\\ &\qquad \mathcal{V}_m\begin{bmatrix}Y_m\mathcal{T}_m^T & Y_mE_mT_{m+1,m}^T\end{bmatrix}\mathcal{V}_{m+1}^T + V_1^{(1)}\Lambda_{1,1}\Lambda_{1,1}^T{V_1^{(1)}}^T\\ &= \mathcal{V}_{m+1}\begin{bmatrix}\mathcal{T}_mY_m + Y_m\mathcal{T}_m^T - Y_m\tilde{B}_m\tilde{B}_m^TY_m + \mathcal{E}_1\Lambda_{1,1}\Lambda_{1,1}^T\mathcal{E}_1^T & Y_mE_mT_{m+1,m}^T\\ T_{m+1,m}E_m^TY_m & 0\end{bmatrix}\mathcal{V}_{m+1}^T.\end{aligned}$$

Since $\tilde{C}_m = \mathcal{E}_1\Lambda_{1,1}$ and Y_m is the symmetric solution of the reduced CARE (3.2), we have

$$R_m = \mathcal{V}_{m+1}\begin{bmatrix}0 & Y_mE_mT_{m+1,m}^T\\ T_{m+1,m}E_m^TY_m & 0\end{bmatrix}\mathcal{V}_{m+1}^T$$

and

$$\|R_m\| = \|T_{m+1,m}E_m^T Y_m\| = \|T_{m+1,m}\tilde{Y}_m\|,$$

where $\tilde{Y}_m = E_m^T Y_m$ represents the $2s$ last rows of Y_m. □

Theorem 3.1 is important in practice, as it allows us to stop the iteration when convergence is achieved without computing the approximate solution X_m at each iteration. The solution X_m could be given as a product of two matrices of low rank. In fact, since X_m is positive semidefinite, it is possible to decompose it as $X_m = ZZ^T$, where the matrix Z is of rank smaller than or equal to $2m$. Consider the singular value decomposition of the $2m \times 2m$ matrix $Y_m = U\Sigma U^T$ where Σ is the diagonal matrix of the singular values of Y_m sorted in decreasing order. Let U_l be the $2m \times l$ matrix of the first l columns of U corresponding to the l singular values of magnitude greater than some tolerance *dtol*. We obtain the truncated singular value decomposition $Y_m \approx U_l \Sigma_l U_l^T$ where $\Sigma_l = \mathrm{diag}[\sigma_1, \ldots, \sigma_l]$. Setting $Z_m = \mathcal{V}_m U_l \Sigma_l^{1/2}$, it follows that

$$X_m \approx Z_m Z_m^T.$$

We notice that this remark was used in [30] for Lyapunov matrix equations.

Next, we give an upper bound for the norm of the error $X - X_m$, where X is the exact solution of the CARE (1.1).

THEOREM 3.2. *Let X_m, Y_m be the mth approximate solution obtained with the extended block Arnoldi-CARE algorithm and the solution of the projected problem (3.2), respectively. Let $\tilde{Y}_m$ be the $2s \times 2ms$ matrix corresponding to the last $2s$ rows of Y_m. We set $\gamma_m = \|T_{m+1,m}\tilde{Y}_m\|$, $\eta = \|BB^T\|$, and $A_m = A - BB^T X_m$, and assume that $\delta_m = \mathrm{sep}(A_m, -A_m^T) > 0$. Then if $4\gamma_m\eta/\delta_m^2 < 1$, we have*

$$\|X - X_m\| \leq \frac{2\gamma_m}{\delta_m + \sqrt{\delta_m^2 - 4\gamma_m\eta}}. \tag{3.7}$$

Proof. The proof is similar to the one given in [18]. □

The extended block Arnoldi (EBA-CARE) algorithm for the continuous-time algebraic Riccati equation is summarized as follows

ALGORITHM 3.3. **The extended block Arnoldi Riccati algorithm (EBA-CARE)**

```
• Inputs:   $A$ an $n \times n$ matrix, $B$ an $n \times p$ matrix, $C$ an $s \times n$ matrix.
• Step 0.   Choose a tolerance $\epsilon > 0$, a maximum number of iterations $m_{max}$,
              and a tolerance dtol;
• Step 1.   For $m = 1,\ 2, \ldots,\ m_{max}$
• Step 1.1      Compute $V_m$ to update the orthonormal basis $\{V_1, \ldots, V_m\}$
                  by Algorithm 2.1.
                  Compute the block Hessenberg matrix $\mathcal{T}_m$ ;
• Step 1.2      Solve the low-dimensional Riccati equation:
                        $\mathcal{T}_m Y_m + Y_m \mathcal{T}_m^T - Y_m \tilde{B}_m \tilde{B}_m^T Y_m + \tilde{C}_m^T \tilde{C}_m = 0;$
                  where $\tilde{B}_m = \mathcal{V}_m^T B$; $\tilde{C}_m^T = \mathcal{V}_m^T C^T$;
• Step 1.3      Compute the residual norm:
                                    $r_m := \|T_{m+1,m}\tilde{Y}_m\|$;
                  where $\tilde{Y}_m$ is the $2s \times 2ms$ matrix corresponding to
                  the last $2s$ rows of $Y_m$.
• Step 1.4      If $r_m < \epsilon$,
                      go to Step 2,
                  end if.
              end for
```

- *Step 2.* Compute the singular value decomposition of Y_m, i.e., $Y_m = U\Sigma U^T$
  where $\Sigma =$ diag$[\sigma_1, \ldots, \sigma_{2m}]$ and $\sigma_1 \geq \ldots \geq \sigma_{2m}$;
  Determine l such that $\sigma_{l+1} < \mathit{dtol} \leq \sigma_l$, set $\Sigma_l =$ diag$[\sigma_1, \ldots, \sigma_l]$;
  compute $Z_m = \mathcal{V}_m U_l \Sigma_l^{1/2}$;
  The approximation X_m is given by $X_m \approx Z_m Z_m^T$.

In Step 1.2, the low order algebraic Riccati problem is solved by the Schur method [23].

4. Numerical examples. In this section, we provide experimental results to show the effectiveness of the extended block Arnoldi-CARE algorithm. The obtained results are compared with those obtained by the block Arnoldi Riccati method [17, 18] and by the Newton-ADI method, also called the Low Rank Cholesky Factorized Newton (LRCF-Newton) method [8, 28]. We used the function `lp_lrnm` in the LYAPACK package [27]. We notice that in each Newton outer iteration we have to solve a large Lyapunov matrix equation. In the LRCF-Newton method, these Lyapunov matrix equations are solved by the Low Rank Cholesky Factorized ADI method [28]. This last method requires the computation of optimal ADI parameters. In our numerical tests, we used a heuristic procedure: the MATLAB function `lp_para` from the LYAPACK library [27]. This last procedure is based on the classical Arnoldi process and is denoted by LRCF(l, k_p, k_m), where l is the number of ADI parameters, k_p is the number of Arnoldi iterations, and k_m is the number of inverted Arnoldi iterations. A maximum number *maxN* $= 20$ of outer iterations was allowed for LRCF-Newton, while the inner iterations were stopped when the norm of inner residuals was less than 10^{-6} or when the inner iteration number was larger than *maxA* $= 50$.

All the experiments were performed on a computer with an Intel Pentium 4 processor at 3.4 GHz and 2048 MBytes of RAM. The algorithms were coded in MATLAB 7.2. The same stopping criterion is used for both algorithms, and the computations were stopped when $\|R(X_m)\|/\|CC^T\| < \epsilon = 10^{-7}$. In the singular value decomposition of Y_m (Step 2 in Algorithm 3.3), we used *dtol* $= 10^{-12}$ to discard the columns of U corresponding to diagonal elements of Σ less than *dtol*.

EXAMPLE 4.1. This first example describes a model of heat flow with convection in the given domain. We consider the following linear time-invariant system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases}$$

where the matrix A is obtained from the centered finite difference discretization of the operator

$$L(u) = \Delta u - 10y\frac{\partial u}{\partial x} - 2x\frac{\partial u}{\partial y} - (y^2 - x^2)u$$

on the unit square $[0,1] \times [0,1]$ with homogeneous Dirichlet boundary conditions. The dimension of the matrix A is $n = n_0^2$, where n_0 is the number of inner grid points in each direction. Different values of n_0, p, and s are used. Since the matrix A is structured (band matrix), the LU factorization required in Algorithm 2.1 is easily done, even though the problems are relatively large. The entries of the $n \times p$ matrix B and the $s \times n$ matrix C are random values uniformly distributed on $[0,1]$.

In Table 4.1, we report the results obtained with the extended block Arnoldi (EBA-CARE), the block Arnoldi (BA-CARE), and the Low Rank Cholesky Factorized Newton (LRCF-Newton) methods. For each method, we listed the number of iterations needed for convergence, the CPU time in seconds, and the rank of the obtained approximate solution. The results listed in Table 4.1 show the performance of EBA-CARE as compared to the other

TABLE 4.1
Results for Example 4.1.

Test problem	Method	Iter.	CPU time	Rank(X_m)
$n = 6400$	EBA-CARE	14	4.87	93
$s = 5, p = 5$	BA-CARE	99	595.65	93
$l = 10, k_p = 20, k_m = 20$	LRCF-Newton	12	678.85	118
$n = 8100$	EBA-CARE	17	3.96	61
$s = 3, p = 2$	BA-CARE	114	210.96	60
$l = 15, k_p = 40, k_m = 20$	LRCF-Newton	11	365.71	65
$n = 12100$	EBA-CARE	17	12.28	101
$s = 5, p = 2$	BA-CARE	127	1503.21	100
$l = 10, k_p = 20, k_m = 20$	LRCF-Newton	12	927.25	115

two algorithms. For LRCF-Newton, we used different values of the parameters l, k_p, and k_m, and we report those parameters that give the best results.

EXAMPLE 4.2. In this example, we consider a benchmark problem coming from a discretization of a convective thermal flow problem [26]. The associated linear time-invariant system is given by

$$\begin{cases} E\dot{x}(t) = A_0 x(t) + B_0 u(t), \\ y(t) = Cx(t). \end{cases} \tag{4.1}$$

The matrices A_0 (`flow_meter_model_v0.5.A`), B_0 (`flow_meter_model_v0.5.B`), E (`flow_meter_model_v0.5.E`) and C (`flow_meter_model_v0.5.C`) have been extracted from the IMTEK collection[1]. For this example $n = 9669$, $\mathrm{nnz}(A_0) = 67391$, $s = 5$ and $p = 1$. As the matrix E is diagonal and nonsingular, the linear time-invariant system (4.1) can be reformulated as in (1.3) with $A = E^{-1}A_0$ and $B = E^{-1}B_0$. We note that for this example $\|A\|_F \simeq 10^6$ and $\|B\|_F \simeq 10^5$.

TABLE 4.2
Results for Example 4.2.

Method	Iter.	Res. norms	CPU time	Rank(X_m)
EBA-CARE	39	$8.67\,10^{-8}$	$1.2\,10^2$	111
BA-CARE	$>$ 300	$2.34\,10^{-2}$	$>$ $2.2\,10^3$	-
LRCF-Newton	$>$ 20	$3.07\,10^{-4}$	$>$ $4.1\,10^3$	-
$l = 10, k_p = 15, k_m = 15$				

For this experiment, we compared the performance of the EBA-CARE algorithm with the LU factorization of the matrix A, the BA-CARE algorithm, and the LRCF-Newton method. As shown in Table 4.2, the BA-CARE and the LRCF-Newton methods fail to converge within a total number of *maxB* $= 300$ iterations for BA-CARE and *maxN* $= 20$ outer iterations for LRCF-Newton.

5. Conclusion.

We presented in this paper a new iterative method for computing low rank approximate solutions to large scale algebraic Riccati equations. The method is based on the extended block Arnoldi algorithm, which is a Krylov subspace generated by the matrices A and A^{-1}. The advantage of this new method as compared to others is the fact that, in

[1] http://www.imtek.de/simulation/

general, it requires a small number of iterations to give an accurate approximation of the desired solution. As compared to the block Arnoldi and to the LRCF-Newton methods for large and sparse problems, our algorithm requires less CPU time and memory. We notice that when the LU factorization of the matrix A is not possible (or is very expensive), then, in the EBA-CARE algorithm, we can use a preconditioned Krylov method for solving linear systems with A, but in this case some theoretical properties are lost.

Acknowledgments. We would like to thank the referees for their recommendations and helpful suggestions.

REFERENCES

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory*, Systems & Control: Foundations & Applications, Birkh auser, Boston, 2003.

[2] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*, Prentice-Hall, Englewood Cliffs, 1971.

[3] ———, *Optimal Control – Linear Quadratic Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[4] W. F. Arnold, III and A. J. Laub, *Generalized eigenproblem algorithms and software for algebraic Riccati equations*, Proc. IEEE, 72 (1984), pp. 1746–1754.

[5] Z. Bai and J. Demmel, *Using the matrix sign function to compute invariant subspaces*, SIAM J. Matrix Anal. Appl, 19 (1998), pp. 205–225.

[6] P. Benner and R. Byers, *An exact line search method for solving generalized continous algebraic Riccati equations*, IEEE Trans. Automat. Control, 43 (1998), pp. 101–107.

[7] P. Benner and H. Fassbender, *An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem*, Linear Algebra Appl., 263 (1997), pp. 75–111.

[8] P. Benner, H. Mena, and J. Saak, *On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations*, Electron. Trans. Numer. Anal., 29 (2008), pp. 136–149. http://etna.math.kent.edu/vol.29.2007-2008/pp136-149.dir/

[9] S. Bittanti, A. Laub, and J. C. Willems, *The Riccati equation*, Springer-Verlag, Berlin, 1991.

[10] R. Byers, *Hamiltonian and Symplectic Algorithms for the Algebraic Riccati Equation*, Ph.D. thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1983.

[11] ———, *Solving the algebraic Riccati equation with the matrix sign function*, Linear Algebra Appl., 85 (1987), pp. 267–279.

[12] J. L. Casti, *Linear Dynamical Systems*, vol. 135 of Mathematics in Science and Engineering, Academic Press, 1987.

[13] B. N. Datta, *Numerical Methods for Linear Control Systems Design and Analysis*, Elsevier Academic Press, 2003.

[14] E. J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, *State space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems*, IEEE Trans. Automat. Control, 34 (1989), pp. 831–846.

[15] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl, 19 (1998), pp. 755–771.

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

[17] I. M. Jaimoukha and E. M. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Matrix Anal. Appl, 31 (1994), pp. 227–251.

[18] K. Jbilou, *Block Krylov subspace methods for large continuous-time algebraic Riccati equations*, Numer. Algorithms, 34 (2003), pp. 339–353.

[19] ———, *An Arnoldi based algorithm for large algebraic Riccati equations*, Appl. Math. Lett., 19 (2006), pp. 437–444.

[20] C. S. Kenny, A. J. Laub, and P. Papadopoulos, *Matrix sign function algorithms for Riccati equations*, IMA J. Math. Control Inform., 9 (1992), pp. 331–344.

[21] D. L. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Trans. Automat. Control, 13 (1968), pp. 114–115.

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

[23] A. J. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, 24 (1979), pp. 913–921.

[24] V. Mehrmann, *A symplectic orthogonal method for single-input single-output discrete-time optimal linear quadratic control problems*, SIAM J. Matrix Anal. Appl, 9 (1988), pp. 221–248.

[25] ———, *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*, vol. 163 of Lecture Notes in Control and Information Sciences, Springer-Verlag, Heidelberg, 1991.

[26] C. Moosmann, E. B. Rudnyi, A. Greiner, and J. G. Korvink, *Model order reduction for linear*

convective thermal flow, in Proceedings of 10th International Workshops on THERMal INvestigations of ICs and Systems, THERMINIC2004, Sophia Antipolis, France, Sept. 2004, pp. 317–321.

[27] T. PENZL, *LYAPACK: A MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems*. http://www.tu-chemnitz.de/sfb393/lyapack.

[28] ———, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (2000), pp. 1401–1418.

[29] D. J. ROBERTS, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, Internat. J. Control, 32 (1980), pp. 677–687.

[30] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.

[31] P. VAN DOOREN, *A generalized eigenvalue approach for solving Riccati equations*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 121–135.

[32] W. M. WONHAM, *On a matrix Riccati equation of stochastic control*, SIAM J. Control Optim., 6 (1968), pp. 681–697.

[33] K. ZHOU, J. C. DOYLE, AND K. GLOVER, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, 1995.