Journal of Mathematical Modeling
Vol. 12, No. 1, 2024, pp. 85–98. Research Article

JMM

An improved extended block Arnoldi method for solving low-rank Lyapunov equation

Ilias Abdaoui*

ENSA Oujda, Equipe MSN, Lab. LM2N, Universit Mohammed Premier, Oujda, Morocco
Email(s): ilias.abdaoui@yahoo.com

Abstract. We are interested in the numerical solution of the continuous-time Lyapunov equation. Generally, classical Krylov subspace methods for solving matrix equations use the Petrov-Galerkin condition to obtain projected equations from the original ones. The projected problems involves the restrictions of the coefficient matrices to a Krylov subspace. Alternatively, we propose a scheme based on the extended block Krylov subspace that leads to a smaller-scale equation, which also incorporates the restriction of the inverse of the Lyapunov equation's square coefficient. The effectiveness of this approach is experimentally confirmed, particularly in terms of the required CPU time.

Keywords: Lyapunov equation, Krylov methods, extended block Arnoldi Process.
AMS Subject Classification 2010: 65F45, 65F50.

1 Introduction

The low-rank continuous-time Lyapunov equation of the form

$$AX + XA^T = BB^T, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a sparse non singular matrix and $B \in \mathbb{R}^{n\times r}$ is full rank with $r \ll n$, admits an unique solution if and only if $\nu_i \neq \nu_j$ for all $i, j = 1,\dots,n$ where $\nu_1, \nu_2, \dots, \nu_n$ are the eigenvalues of A. Eq. (1) appears in many fields, and plays a key role in quadratic integrals in optimal control [5], evaluating covariance matrices in filtering and estimation of continuous systems, and pole assignment [17]. Furthermore, it is involved in reduced-order models [14] and in some iterative schemes for solving the algebraic Riccati equation [13, 18].

There are numerous methods for solving the Lyapunov equation and their relevance depends on the structure of the coefficient matrices and their sizes. For small-scale problems, it is preferable to use direct

*Corresponding author
Received: 28 September 2023 / Revised: 18 November 2023 / Accepted: 28 November 2023
DOI: 10.22124/JMM.2023.25670.2281

methods such as Hessenberg-Schur and Bartel-Stewart methods [3, 9]. However, these methods are not computationally competitive for large-scale equations. In order to overcome the drawbacks of direct approaches, several iterative schemes have been proposed, including the Alternating Direction Implicit (ADI) iteration method [4, 8, 15], the Low-Rank Smith method [16], the Alternating Direction Implicit method with Cholesky factorization [15], and the Matrix Sign Function method [2].

Over the past years, the so-called Krylov subspace methods have been widely used for solving matrix equations, specifically the Lyapunov equation with sparse and large coefficient matrices. The basic idea behind this class of methods is to reduce the order of the original problem by projecting it onto a Krylov subspace that has a generally smaller dimension. The projected problem is then solved directly with standard methods. To build a basis for this subspace, various projection techniques such as the Arnoldi process or its extended variant, which is the key tool adopted in this paper for solving (1), are employed. We recall that an m-dimensional extended block Krylov subspace associated with a square matrix G and a block vector V of compatible sizes is given as

$$\mathbb{K}_m^e(G,V) = \text{blockspan}\{G^{-m}V, \dots, G^{-1}V, V, GV \dots, G^{m-1}V\}. \tag{2}$$

Krylov subspaces of such form were introduced in [7] for approximating matrix functions and efficiently used in [14] for solving the equation (1), the low-rank Sylvester equation [11, 19] and Riccati equation [10]. Considering the fact that it is generated using both G and G^{-1}, this subspace, when used to solve matrix equations, is supposed to contain more information about the coefficient matrices, and most particularly about their inverses. Therefore, it has been noticed that these extra information improve the performance of projection techniques on such subspaces when compared to those based on the regular block Krylov subspaces. To build an orthonormal basis of $\mathbb{K}_m^e(G,V)$, the extended block Arnoldi process is often applied. In addition to the algebraic relations satisfied by this basis, a Petrov-Galerkin condition is imposed and then exploited to reduce the order of the original equations. It should be noted that the projected equations involve the restrictions of the original coefficient matrices to the projection subspace. In this work, however, we seek an approximate solution for the continuous Lyapunov equation by transforming it first to an equivalent discrete Sylvester equation. Then, we use the extended block Arnoldi process as a projection technique to derive a projected Sylvester equation that has, among its coefficient matrices, the restriction of A^{-1} to the Extended Krylov subspace $\mathbb{K}_m^e(G,V)$.

The organization of the remaining sections of this paper is as follows. In Section 2, we will provide a review of the extended block Arnoldi process, along with its relevant algebraic properties. Then, we show how to apply this projection method for solving equation (1). The alternative scheme is described in Section 3. In order to evaluate the effectiveness of the proposed method, we will compare it with the classic extended block Arnoldi process applied to the continuous Lyapunov equation. The focus will be on the CPU time and the number of iterations required by each method. the results of the numerical experiments will be presented in the final section.

Throughout this paper, we will indicate by $E_k^{(i,j)}$ and $e_k^{(i,j)}$ respectively the k^{th} $2ij \times r$ and $ij \times r$ blocks of the identity matrix I_{2ij}. $\|.\|_2$ and $\|.\|_F$ will point out to the 2-norm and the Frobenius norm, respectively and the transpose of a Matrix X will be expressed as X^T.

2 The Extended block Arnoldi process (EBA)

2.1 Algorithm and algebric relations

The EBA process serves as a projection tool onto an extended Krylov subspace $\mathbb{K}_m^e(G,V)$, which is formed by means of a non singular square matrix G and its inverse G^{-1}. This process, outlined in Algorithm 1, enables the construction of an orthonormal basis $\mathbb{V}_m$ for $\mathbb{K}_m^e(G,V)$. As mentioned earlier and drawing upon prior research, we firmly believe that incorporating G and G^{-1} will enhance the projection efficiency. We initiate the process by deriving a unitary block vector V_1 through the QR decomposition of the matrix $[V, G^{-1}V]$. Subsequently, at each iteration j, the first n by r block of V_j is multiplied by G, while the second is multiplied with G^{-1}. Using Gram-Schmit process, the n by $2r$ block $[GV_j^{(1)}, G^{-1}V_j^{(2)}]$ is orthogonalized with respect to the set of block vectors $[V_1,\ldots,V_j]$ giving arise to V_{j+1}. This iterative procedure continues, and after completing m iterations, Algorithm 1 builds an orthonormal basis $\mathbb{V}_m = [V_1,\ldots,V_m]$ of $\mathbb{K}_m^e(G,V)$. Moreover, the algorithm generates also a block upper Hessenberg matrix referred to as $\bar{\mathscr{H}}_m = [H_{i,j}] \in \mathbb{R}^{2(m+1)r\times 2mr}$, where $H_{i,j} \in \mathbb{R}^{2r\times 2r}$ with $i = 1,2,\ldots,m+1$ and $j = 1,2,\ldots,m$. We assume that for the rest of this paper, the occurrence of an undesirable breakdown due to any block $H_{j+1,j}$ being rank deficient is not expected. Let $\mathscr{T}_m = \mathbb{V}_m^T G \mathbb{V}_m$ be the restriction of G to the extended Krylov subspace $\mathbb{K}_m^e(G,V)$. According to [11], the matrix $\mathscr{T}_m$ satisfies the algebraic relations that are similar to those fulfilled by the block upper Hessenberg matrix produced through regular block Arnoldi process. Indeed, let $\mathscr{T}_{i,j}$ be the $(i,j)^{th}$ $2r\times 2r$ block of $\mathscr{T}_m$ and

$$\bar{\mathscr{T}}_m = \mathbb{V}_{m+1}^T G \mathbb{V}_m,$$

then we have

$$G\mathbb{V}_m = \mathbb{V}_{m+1}\,\bar{\mathscr{T}}_m \tag{3}$$

$$= \mathbb{V}_m\,\mathscr{T}_m + V_{m+1}\,T_{m+1,m}\,(E_m^{(m,r)})^T. \tag{4}$$

In terms of computational demands, obtaining $\bar{\mathscr{T}}_m$ can be expensive as it involves matrix-vector products with G. To address this concern, an alternative approach is to recursively compute the columns of $\bar{\mathscr{T}}_m$ based on the columns of $\bar{\mathscr{H}}_m$. Let $H_{i+1,i}$ and Λ be partitioned as follows

$$H_{i+1,i} = \begin{bmatrix} H_{i+1,i}^{(1,1)} & H_{i+1,i}^{(1,2)} \\ 0_{r\times r} & H_{i+1,i}^{(2,2)} \end{bmatrix},\ \Lambda = \begin{bmatrix} \Lambda_{1,1} & \Lambda_{1,2} \\ 0_{r\times r} & \Lambda_{2,2} \end{bmatrix}.$$

Theorem 1. *[11] Let $\bar{\mathscr{T}}_m = [t_{:,1}, t_{:,2},\ldots,t_{:,2m}]$ and $\bar{\mathscr{H}}_m = [h_{:,1}, h_{:,2},\ldots,h_{:,2m}]$ where $t_{:,j}, h_{:,j} \in \mathbb{R}^{2(m+1)r\times r}$ be the block upper Hessenberg matrices defined earlier. For the blocks with odd indices we have*

$$t_{:,2i-1} = h_{:,2i-1}, \tag{5}$$

while for the even blocks we have

$$t_{:,2} = (e_1^{(m+1,r)}\Lambda_{1,1} - t_{:,1}\Lambda_{1,2})\Lambda_{2,2}^{-1}, \tag{6}$$

$$t_{:,2i+2} = \left(e_{2i}^{(m+1,r)} - \begin{bmatrix} \bar{\mathscr{T}}_i \\ 0_{2(m-i)r\times 2ir} \end{bmatrix} h_{:,2i} - t_{:,2i+1}\,H_{i+1,i}^{(1,2)}\right) {H_{i+1,i}^{(2,2)}}^{-1}. \tag{7}$$

Note that equations (3) and (4) hold true for

$$\mathscr{Q}_m = \mathbb{V}_m^T G^{-1} \mathbb{V}_m \in \mathbb{R}^{2mr\times 2mr},$$

which represents the restriction of G^{-1} to the enriched Krylov subspace $\mathbb{K}_m^e(G,V)$. Furthermore, the columns of $\mathscr{Q}_m$ can be computed from those of $\bar{\mathscr{H}}_m$ as given in the following result.

Theorem 2. *Let* $\bar{\mathscr{Q}}_m = [l_{:,1}, l_{:,2}, \ldots, l_{:,2m}]$ *and* $\bar{\mathscr{H}}_m = [h_{:,1}, h_{:,2}, \ldots, h_{:,2m}]$ *where* $l_{:,j}, h_{:,j} \in \mathbb{R}^{2(m+1)r\times r}$ *for* $j = 1,2,\ldots m$*. The odd-numbered blocks satisfy*

$$t_{:,1} = (e_1^{(m+1,r)}\Lambda_{1,2} - h_{:,1}\Lambda_{2,2})\Lambda_{1,1}^{-1}, \tag{8}$$

$$t_{:,2j+1} = (e_{2j-1}^{(m+1,r)} - \begin{bmatrix} \bar{\mathscr{Q}}_j \\ 0_{2(m-j)r\times 2ir} \end{bmatrix} h_{1:2jr,2j-1}) H_{j+1,j}^{(1,1)\;-1}, \tag{9}$$

while for the even-numbered we have

$$t_{:,2j} = h_{:,2j}, \tag{10}$$

Proof. Based on the QR decomposition of $[V, G^{-1}V]$, we have

$$V = V_1^{(1)}\Lambda_{1,1}, \tag{11}$$

$$G^{-1}V = V_1^{(1)}\Lambda_{1,2} + V_1^{(2)}\Lambda_{2,2}. \tag{12}$$

Therefore

$$G^{-1}V_1^{(1)} = (V_1^{(1)}\Lambda_{1,2} + V_1^{(2)}\Lambda_{2,2})\Lambda_{1,1}^{-1}.$$

and the relation (8) is obtained by right-multiplying the previous equation with $\mathbb{V}_{m+1}^T$. For the other odd indices and according to Algorithm 1, we may write

$$\begin{aligned} G^{-1}V_j^{(1)} &= V_{j+1}H_{j+1,j}e_1^{(1,r)} + \mathbb{V}_j\bar{\mathscr{H}}_j e_{2j-1}^{(j,r)} \\ &= V_{j+1}^{(1)}H_{j+1,j}^{(1,1)} + \mathbb{V}_j\bar{\mathscr{H}}_j e_{2j-1}^{(j,r)}. \end{aligned}$$

Multiplying the above equality bt $\mathbb{V}_{m+1}^T G^{-1}$ gives that

$$\begin{aligned} \mathbb{V}_{m+1}^T G^{-1} V_{j+1}^{(1)} H_{j+1,j}^{(1,1)} &= l_{:,2i+1} H_{j+1,j}^{(1,1)} \\ &= e_{2j-1}^{(m+1,r)} - \begin{bmatrix} \bar{\mathscr{Q}}_i \\ 0_{2(m-j)r\times 2jr} \end{bmatrix} h_{1:2jr,2j-1}, \end{aligned}$$

hence (9) is obtained. About even indices, we can deduce from (12) that

$$\begin{aligned} G^{-1}V_j^{(2)} &= V_{j+1,j}H_{j+1,j}e_1^{(1,r)} + \mathbb{V}_j\mathscr{H}_j e_{2j}^{(j,r)} \\ &= \mathbb{V}_{j+1}\bar{\mathscr{H}}_j e_{2j}^{(j+1,r)}, \end{aligned}$$

consequently

$$\mathbb{V}_{m+1}^T G^{-1} V_{j+1}^{(2)} = \begin{bmatrix} I_{2(j+1)r} \\ 0_{2(m-j)r\times 2jr} \end{bmatrix} \bar{\mathscr{H}}_j e_{2j}^{(j+1,r)} = \bar{\mathscr{H}}_m e_{2j}^{(m+1,r)} = h_{:,2j},$$

which completes the proof. □

At the end of this section, it is important to note that in the context of Lyapunov equations of considerable size, explicit computation of the inverse of G is not typically carried out. Instead, it is generally more advantageous to approximate this inverse by solving a linear system using iterative methods in conjunction with specific preconditioning techniques.

Algorithm 1 EBA Process

Require: Matrices : $G \in \mathbb{R}^{n\times n}$; $V \in \mathbb{R}^{n\times r}$; integer : m ;
Ensure: Matrices : $\mathbb{V}_{m+1} = [V_1, V_2, \ldots, V_{m+1}] \in \mathbb{R}^{n\times 2mr}$ and $\overline{\mathscr{H}}_m = [H_{i,j}] \in \mathbb{R}^{2(m+1)r\times 2mr}$;

```
Compute the QR decomposition of [V, G^{-1} V]; i.e., [V, G^{-1} V] = [V_1^{(1)}, V_1^{(2)}] Λ ;
for j = 1, ..., m do
   Set V_j^{(1)} = V_j(:, 1 : r) and V_j^{(2)} = V_j(:, r+1 : 2r) ;
   Compute W = [G V_j^{(1)}, G^{-1} V_j^{(2)}]
   for i = 1, ..., j do
      H_{i,j} = V_i^T W ;
      W = W − V_i H_{i,j} ;
   end for
   Compute the QR decomposition of W, i.e., W = V_{j+1} H_{j+1,j} ;
end for
```

2.2 Application to the Lyapunov equation

Now, we show how the extended block Arnoldi process can be used for solving the equation (1). To that end, let $\mathbb{V}_m$ denote the orthogonal matrix obtained by performing m iterations of Algorithm 1 applied to the pair (A,B). Moreover, We seek approximate solutions of the form

$$X_m = \mathbb{V}_m Y_m (\mathbb{V}_m)^T, \tag{13}$$

where $Y_m \in \mathbb{R}^{2mr\times 2mr}$. We suppose that R_m the residue associated to X_m satisfies the following orthogonality condition:

$$(\mathbb{V}_m)^T R_m \mathbb{V}_m = 0, \tag{14}$$

more precisely

$$(\mathbb{V}_m)^T (A \mathbb{V}_m Y_m (\mathbb{V}_m)^T + \mathbb{V}_m Y_m (\mathbb{V}_m)^T A^T - B B^T) \mathbb{V}_m = 0.$$

Supposing that the eigenvalues of $\mathscr{T}_m = \mathbb{V}_m^T A \mathbb{V}_m$ are all distinct, Y_m is the solution of the following smaller problem which can be effectively solved using direct methods [3]

$$\mathscr{T}_m Y_m + Y_m (\mathscr{T}_m)^T = \tilde{B} \tilde{B}^T, \tag{15}$$

where $\tilde{B} = \mathbb{V}_m^T B = e_1^{(m,r)} \Lambda_{1,1}$ and

$$\Lambda = \begin{bmatrix} \Lambda_{1,1} & \Lambda_{1,2} \\ 0_{r\times r} & \Lambda_{2,2} \end{bmatrix},$$

is obtained from the QR decomposition of $[B, A^{-1}B]$. Regarding the arithmetic cost, as long as the projection process requires more iterations to solve the problem, computing X_m and R_m becomes too demanding. In fact, any matrix-vector product involving the matrix A slows down the convergence rate, that why R_m can be obtained using a more economical formula that avoids such costly operations. Using (4) we show that

$$R_m = -\mathbb{V}_{m+1} \left[\begin{array}{cc} 0_{mr\times mr} & Y_m E_m^{(m,r)} (\mathscr{T}_{m+1,m})^T \\ \mathscr{T}_{m+1,m} (E_m^{(m,r)})^T Y_m & 0_{r\times r} \end{array} \right] \mathbb{V}_{m+1}.$$

Therefore

$$\|R_m\|_F = \sqrt{2} \|\mathscr{T}_{m+1,m} Y_{(m-1)r:mr,:}\|_F. \tag{16}$$

Algorithm 2 outlines the EBA process for solving the Lyapunov equation. We note that the projection can be done every p iteration in order to reduce the operational cost. Moreover, according to lines 12-15 of Algorithm 2, the solution is approximated by a product of two reduced-rank matrices using SVD decomposition.

Algorithm 2 EBA process for solving low-rank Lyapunov equation

---

Require: $A \in \mathbb{R}^{n\times n}$; $B \in \mathbb{R}^{n\times r}$; m_{max} the maximum dimension of the block Krylov subspace; ε the desired precision; τ the threshold used for the truncated SVD; $p \in \mathbb{N}$ the projection step size;

Ensure: X_m an approximate solution of the Lyapunov equation;

1: **for** $m = 1, 2, \ldots, m_{max}$ **do**
2: Generate the m^{th} block of the orthonormal basis $\mathbb{V}_m$ as well as that of the Hessenberg matrix $\mathscr{H}_m$ by applying simultaneously Algorithm 1 to the pair (A, B)
3: Compute the m^{th} block of $\mathscr{T}_m$ using 1
4: **if** m is a multiple of p **then**
5: Solve the projected equation:
$$\mathscr{T}_m Y_m + Y_m (\mathscr{T}_m)^T = \tilde{B}\tilde{B}^T,$$
6: Compute the norm of de R_m using (16)
7: **if** $\|R_m\|_F \le \varepsilon$ **then**
8: Goto line 12
9: **end if**
10: **end if**
11: **end for**
12: Compute the SVD of Y_m, i.e., $Y_m = U \Sigma U^T$ where $\Sigma = \mathrm{diag}[\sigma_1, \ldots, \sigma_{mr}]$ and $\sigma_1 \ge \ldots \ge \sigma_{mr}$;
13: Find l such that $\sigma_{l+1} \le \tau < \sigma_l$ and take $\Sigma_l = \mathrm{diag}[\sigma_1, \ldots, \sigma_l]$;
14: Compute $Z_m = \mathbb{V}_m U_l \Sigma_l^{1/2}$;
15: Compute the approximate solution $X_m \approx Z_m (Z_m)^T$.

---

3 An alternative approach

In this section, we show that an alternative approach can be used for solving equation (1). First, the idea is to note that the continuous Lyapunov equation, if it admits a unique solution, is equivalent by right-multiplying it with A^{-1} to the following discrete Sylvester equation

$$A^{-1}XA^T + X = A^{-1}BB^T, \tag{17}$$

In other words, instead of applying the EBA process to solve (1), we will use it with a few modications to get a low-rank solution to equation (17). Consequently, the projected equation implies restrictions of A and A^{-1} to the extended Krylov subspace $\mathbb{K}_m^e(G,V)$ which will hopefully improve convergence speed compared to the regular EBA method. We still consider that $\tilde{X}_m$ the approximate solution of (17) is of the form (13) meaning that $\tilde{X}_m = \mathbb{V}_m\tilde{Y}_m(\mathbb{V}_m)^T$ where $\mathbb{V}_m$ is defined earlier while $\tilde{Y}_m$, as we will see, is the solution of a projected equation derived from (17). Now, consider $\tilde{R}_m = A^{-1}BB^T - A^{-1}\tilde{X}_mA^T - \tilde{X}$ the residue related to $\tilde{X}_m$. By applying the orthogonality condition (14) to $\tilde{R}_m$, we show that $\tilde{Y}_m$ solves the follwing projected Sylvester equation

$$\mathscr{Q}_m\, Y_m\,(\mathscr{T}_m)^T + \tilde{Y}_m = \hat{B}\tilde{B}^T, \tag{18}$$

where $\mathscr{Q}_m = {\mathbb{V}_m^A}^T A^{-1}\mathbb{V}_m^A$ and $\mathscr{T}_m = {\mathbb{V}_m}^T A\,\mathbb{V}_m$ are respectively the restrictions of A^{-1} and A to the enriched Krylov subspaces $\mathbb{K}_m^e(A,B)$. Using the QR decomposition of $[B, A^{-1}B]$, it is shown that $\tilde{B} = {\mathbb{V}_m^A}^T B = e_1^{(m,r)}\Lambda_{1,1}$ while $\hat{B}$ satisfies

$$\hat{B} = \mathbb{V}_m^T A^{-1}B = e_1^{(m,r)}\Gamma_{2,1}^A + e_2^{(m,r)}\Gamma_{2,2}^A.$$

To optimize computational costs, the residue is computed as mentioned in the preceding section without performing matrix-vector multiplication with the equation coefficients matrices. To that end, we consider the following result.

Theorem 3. *Let $\tilde{R}_m$ be the residue associated to $\tilde{X}_m$ the approximate solution of* (17)*. We have*

$$\|\tilde{R}_m\|_F = \sqrt{\|\lambda_m\|_F^2 + \|\mu_m\|_F^2 + \|\nu_m\|_F^2}, \tag{19}$$

where $\lambda_m = \mathscr{Q}_m\,\tilde{Y}_m\,\mathscr{T}_{m+1,m}$, $\mu_m = \mathscr{Q}_{m+1,m}\,(E_m^{(m,r)})^T\,\tilde{Y}_m\,(\mathscr{T}_m)^T$ *and*

$$\nu_m = \mathscr{Q}_{m+1,m}\,(E_m^{(m,r)})^T\,\tilde{Y}_m E_m^{(m,r)}\,(\mathscr{T}_{m+1,m})^T.$$

Proof. Using the Arnoldi properties satisfied by $\mathscr{Q}_m$ and $\mathscr{T}_m$ and the factorized form of $\tilde{X}_m$, we can show by simple manipulations that

$$\begin{aligned}
\tilde{R}_m &= A^{-1}\tilde{X}A^T + \tilde{X} - A^{-1}BB^T \\
&= A^{-1}\mathbb{V}_m\tilde{Y}_m\mathbb{V}_m^TA^T + \mathbb{V}_m\tilde{Y}_m\mathbb{V}_m^T - \mathbb{V}_m\hat{B}\tilde{B}^T\mathbb{V}_m^T \\
&= -\mathbb{V}_{m+1}\begin{bmatrix} \mathscr{Q}_m\,\tilde{Y}_m\,(\mathscr{T}_m)^T + \tilde{Y}_m - \hat{B}\tilde{B}^T & \mathscr{Q}_m\tilde{Y}_m\,\mathscr{T}_{m+1,m} \\ \mathscr{Q}_{m+1,m}\,(E_m^{(m,r)})^T\,\tilde{Y}_m^c\,(\mathscr{T}_m)^T & \mathscr{Q}_{m+1,m}\,(E_m^{(m,r)})^T\,\tilde{Y}_m^c E_m^{(m,r)}\,(\mathscr{T}_{m+1,m})^T \end{bmatrix}\mathbb{V}_{m+1}.
\end{aligned}$$

Hence the result, (19) is obtained by taking the Frobenius norm and considering that $\tilde{Y}_m^c$ solves (18). □

Remark 3.1. We note that the accuracy of $\tilde{X}_m$ as an approximate solution of the Lyapunov equation can be checked out without computing its corresponding residue $R_m = AX + XA^T - BB^T$. In fact, it is obvious to see that

$$R_m = A\tilde{R}_m = A\mathbb{V}_{m+1}\begin{bmatrix} 0 & -\lambda_m \\ -\mu_m & -\nu_m \end{bmatrix}\mathbb{V}_{m+1}^T.$$

According to (3), we have $A\mathbb{V}_{m+1} = \mathbb{V}_{m+2}\bar{\mathscr{T}}_{m+1}$. Therefore

$$\begin{aligned} \|R_m\|_F &= \|\mathbb{V}_{m+2}\bar{\mathscr{T}}_{m+1}\begin{bmatrix} 0 & -\lambda_m \\ -\mu_m & -\nu_m \end{bmatrix}\mathbb{V}_{m+1}^T\|_F \\ &\leq \|\bar{\mathscr{T}}_{m+1}\|_F\sqrt{\|\lambda_m\|_F^2 + \|\mu_m\|_F^2 + \|\nu_m\|_F^2} \\ &\leq \|\bar{\mathscr{T}}_{m+1}\|_F\|\tilde{R}_m\|_F. \end{aligned}$$

To use the previous relation as stopping criteria, we will call for an additional iteration since $\bar{\mathscr{T}}_{m+1}$ is computed at the $(m+1)^{th}$ iteration. That is why we will halt the process when $\|A\|_F\|\tilde{R}_m\|_F \leq \varepsilon$ where ε is the tolerance used for solving (1).

Now, let X be the exact solution of (17) and $\tilde{X}_m$ the approximate solution provided by the alternative method. Assuming that the matrix A is stable, meaning that all its eigenvalues are inside the unit circle. The following result shows that the error $X - \tilde{X}_m$ may be upper bounded.

Theorem 4. *Let X be the exact solution of* (17) *and $\tilde{X}_m$ the approximate solution provided by the alternative method. We have*

$$\|X-\tilde{X}_m\|_F \leq \frac{\|\mathscr{T}_{m+1,m}\|_F\|\mathscr{Q}_m\|_F + \|\mathscr{Q}_{m+1,m}\|_F\|\mathscr{T}_m\|_F + \|\mathscr{T}_{m+1,m}\|_F\|\mathscr{Q}_{m+1,m}\|_F}{1-\|\mathscr{Q}_m\|_F\|\mathscr{T}_m\|_F}\|\tilde{Y}_m\|_F. \tag{20}$$

Proof. Since $\tilde{Y}_m$ solves (18), we may write

$$\mathbb{V}_m(\mathscr{Q}_m Y_m(\mathscr{T}_m)^T + \tilde{Y}_m - \mathbb{V}_m^T A^{-1}BB^T\mathbb{V}_m)\mathbb{V}_m^T = 0.$$

Using (4), we have

$$(A^{-1}\mathbb{V}_m - V_{m+1}\mathscr{Q}_{m+1,m}(E_m^{(m,r)})^T)\tilde{Y}_m(\mathbb{V}_m^T A^T - E_m^{(m,r)}\mathscr{T}_{m+1,m}V_{m+1}^T) + \mathbb{V}_m\tilde{Y}_m\mathbb{V}_m^T - A^{-1}BB^T = 0.$$

Therefore, $\tilde{X}_m$ is the exact solution of the discrete Sylvester equation

$$(A^{-1}-\psi_m)\tilde{X}_m(A^T-\omega_m) + \tilde{X}_m = A^{-1}BB^T, \tag{21}$$

where $\psi_m = V_{m+1}\mathscr{Q}_{m+1,m}(E_m^{(m,r)})^T\mathbb{V}_m^T$ and $\omega_m = \mathbb{V}_m E_m^{(m,r)}\mathscr{T}_{m+1,m}V_{m+1}^T$. Subtracting (21) from (17) gives that

$$A^{-1}(X-\tilde{X}_m)A^T + X_m - \tilde{X}_m = -A^{-1}\tilde{X}_m\omega_m - \psi_m\tilde{X}_m A^T + \psi_m\tilde{X}_m\omega_m.$$

It follows that

$$X - \tilde{X}_m = \sum_{i=1}^{\infty} A^{-i}\Theta_m(A^T)^i,$$

where $\Theta_m = -A^{-1}\tilde{X}_m\,\omega_m - \psi_m\,\tilde{X}_m A^T + \psi_m\,\tilde{X}_m\,\omega_m$. Considering that $\|A^{-1}\|_F = \|\mathscr{Q}_m\|_F$, $\|A^T\|_F = \|\mathscr{T}_m\|_F$ and assuming that $\|\mathscr{Q}_m\|_F\|\mathscr{T}_m\|_F \leq 1$, we have

$$\begin{aligned}\|X-\tilde{X}_m\|_F &\leq (\sum_{i=1}^{\infty}\|A^{-i}\|_F\|(A^T)^i\|_F)\|\Theta_m\| \\ &\leq (\sum_{i=1}^{\infty}(\|\mathscr{Q}_m\|_F\|\mathscr{T}_m\|_F)^i)\|\Theta_m\| \\ &\leq \frac{\|\Theta_m\|_F}{1-\|\mathscr{Q}_m\|_F\|\mathscr{T}_m\|_F}.\end{aligned}$$

Since $\|\tilde{X}_m\|_F = \|\tilde{Y}_m\|_F$, $\|\psi_m\|_F = \|\mathscr{Q}_{m+1,m}\|_F$ and $\|\omega_m\|_F = \|\mathscr{T}_{m+1,m}\|_F$, we get that

$$\begin{aligned}\|\Theta_m\|_F &= \|-A^{-1}\tilde{X}_m\,\omega_m - \psi_m\,\tilde{X}_m A^T + \psi_m\,\tilde{X}_m\,\omega_m\|_F \\ &\leq (\|\mathscr{T}_{m+1,m}\|_F\,\|\mathscr{Q}_m\|_F + \|\mathscr{Q}_{m+1,m}\|_F\,\|\mathscr{T}_m\|_F + \|\mathscr{T}_{m+1,m}\|_F\,\|\mathscr{Q}_{m+1,m}\|_F)\|\tilde{Y}_m\|_F,\end{aligned}$$

which proofs (20). Algorithm 3 summarizes the proposed method. □

Algorithm 3 Alternative EBA Method for solving low-rank Lyapunov equation (AEBA)

Require: $A \in \mathbb{R}^{n\times n}$; $B \in \mathbb{R}^{n\times r}$; m_{max} the maximum dimension of the Krylov subspace; ε the desired precision; τ the threshold used for the truncated SVD; $p \in \mathbb{N}$ the projection step size;

Ensure: $\tilde{X}_m$ an approximate solution of the Lyapunov equation;

1: **for** $m = 1, 2, \ldots, m_{max}$ **do**
2:   Generate the m^{th} block of the orthonormal basis $\mathbb{V}_m$ as well as that of the Hessenberg matrix $\mathscr{H}_m$ by applying simultaneously Algorithm 1 to the pair (A, B)
3:   Compute the m^{th} blocks of $\mathscr{T}_m$ and $\mathscr{Q}_m$ using (1) and (2)
4:   **if** m is a multiple of p **then**
5:     Solve the projected equation: $\mathscr{Q}_m\,Y_m\,(\mathscr{T}_m)^T + \tilde{Y}_m = \hat{B}\tilde{B}^T$,
6:     Compute the norm of de $\tilde{R}_m$ using (19)
7:     **if** $\|\tilde{R}_m\|_F \leq (\|A\|_F)^{-1}\,\varepsilon$ **then**
8:       Goto line 12
9:     **end if**
10:   **end if**
11: **end for**
12: Compute the SVD of $\tilde{Y}_m$, i.e., $\tilde{Y}_m = U\,\Sigma\,U^T$ where $\Sigma = \mathrm{diag}[\sigma_1, \ldots, \sigma_{mr}]$ and $\sigma_1 \geq \ldots \geq \sigma_{mr}$;
13: Find l such that $\sigma_{l+1} \leq \tau < \sigma_l$ and take $\Sigma_l = \mathrm{diag}[\sigma_1, \ldots, \sigma_l]$;
14: Compute $Z_m = \mathbb{V}_m\,U_l\,\Sigma_l^{1/2}$;
15: Compute the approximate solution $\tilde{X}_m \approx Z_m\,(Z_m)^T$.

We note that a similar approach has been applied to the Sylvester equation with a factored right-side [1], which represents the general case of equation (1). Nevertheless, we tried to find a significantly more accurate solution with a noticeable reduction in execution time. In addition, theoretical results relating to the upper bound of the approximation error and the residual norm have been adjusted to deal with the specific case discussed in this paper.

4 Applications and comparaison

In order to evaluate the alternative method for solving the Lyapunov equation (referred to as AEBA), we will compare its numerical behavior with the classical method denoted as EBA. The CPU time and the number of iterations performed by each algorithm will be taken into account. The tests are conducted using Matlab R2018 on a professional Windows 10 system equipped with an Intel(R) Core(TM) i5 processor @2.40GHz and 8 gigabytes of memory. To solve the projected equations (15) and (17), we used the Matlab functions lyap and dlyap2, respectively. The action of A^{-1} is computed using the LU decomposition while the right hand side of the Lyapunov equations is chosen randomly. The tolerance employed to truncate SVD of Y_m and $\tilde{Y}_m$ is set to $\tau = 10^{-12}$.

Example 1. The first set of experiments (Table 1) involves coefficient matrices from the Suitesparse Matrix collection [6]. In the second set of tests (Table 2), the matrix A is chosen from the MATLAB gallery. For the fist set, we plot the variations of the residual norm against the number of iterations as illustrated in Figure 1. The compared methods are stopped once we have $R_m < 10^{-8}$.

Table 1: Obtained results by the EBA and AEBA methods for the solving Lyapunov equation (Example 1, Set 1).

(A,n,r)	Method	Iterations	Residual norm	CPU time
(swang1,3169,3)	AEBA	11	1.13e-10	0.12
	CEBA	10	1.63e-09	0.23
(cage9,3534,6)	AEBA	10	1.92e-11	1.44
	CEBA	9	2.23e-10	1.86
(poli,4008,10)	AEBA	10	1.68e-09	0.75
	CEBA	16	1.60e-10	1.56
(thermal,3456,5)	AEBA	12	8.12e-11	5.59
	CEBA	11	8.34e-10	5.80
(pde2961,2961,5)	EBA	*	*	*
	AEBA	27	3.28e-09	3.00
(poli_large,15575,2)	AEBA	12	4.84e-09	0.18
	CEBA	13	2.88e-09	0.32

Despite having performed more iterations in some cases, the alternative method converges faster than the classical method. This does not mean that we have not encountered cases where the latter performs better in terms of execution time, as in the case of the matrix “poisson” in the second test set.

Example 2. In the first set of tests, we consider the matrix of the form

$$A = \begin{pmatrix} 4 & 1-p & 0 & \dots & 0 & 1 \\ 1+p & 4 & 1-p & 0 & \dots & 0 \\ 0 & \ddots & \ddots & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & \ddots & 1-p \\ 1 & 0 & \dots & 0 & 1+p & 4 \end{pmatrix},$$

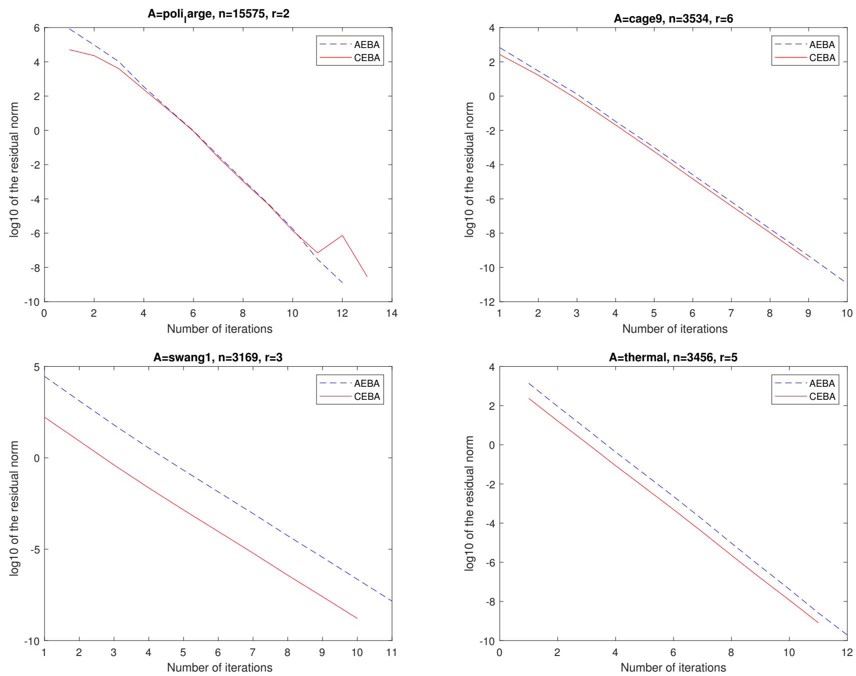

Figure 1: Convergence curves of the CEBA and AEBA methods (Set 1).

Table 2: Obtained results by the EBA and AEBA methods for solving the Lyapunov equation (Example 1, Set 2)

Test		Method	Iterations	Residual norm	CPU time
A =lesp, $n = 10000$	$r = 2$	AEBA	47	1.80e-07	75.7
		CEBA	*	*	*
	$r = 5$	AEBA	39	1.57e-07	78.7
		CEBA	*	*	*
A = triw, $n = 3000$	$r = 2$	AEBA	10	6.51e-10	1.86
		CEBA	10	6.90e-10	1.89
	$r = 5$	AEBA	10	9.94e-10	2.17
		CEBA	10	1.06e-09	3.14
A = poisson, $n = 8100$	$r = 2$	AEBA	49	2.81e-09	3.33
		CEBA	26	7.04e-09	0.81
	$r = 5$	AEBA	10	9.94e-10	2.17
		CEBA	10	1.06e-09	3.14

used in [12] for solving large-scale algebraic Riccati equation with $p \in]0,1[$. In order to check the accuracy of the approximate solutions, the exact solution is computed before running the comparative methods, while the stopping criteria is $R_m < 10^{-10}$. The error norm and the rank of the approximate solution are also provided. In the second set, the matrix A is given such that

$$A = 2^{-t} I_n + diag(1:n) + tridiag(1,0,-1),$$

where $t = 0.4$. Comparing the AEBA and CEBA methods show that the former is generally faster than the classic method. Additionally, the solution provided by the AEBA method consistently achieves a lower rank than the CEBA solution in the first set of this example. Regarding the number of iterations, it is noteworthy that the alternative approach converges with fewer iterations in many cases. However, we emphasize that the CEBA method yields a more accurate solution in the second set concerning the error norm.

Table 3: Obtained results by the EBA and AEBA methods for solving the Lyapunov equation (Example 2, Set 1)

Test		Method	Iterations	Residual norm	CPU time	$\|X - X_m\|_F$	Rank(X_m)
$n = 3000$	$r = 2$	AEBA	10	5.63e-11	0.14	2.70e-11	22
		CEBA	11	1.19e-11	0.31	2.95e-12	23
	$r = 5$	AEBA	11	2.26e-11	0.42	1.18e-11	60
		CEBA	11	9.05e-11	0.98	1.59e-11	65
$n = 5000$	$r = 2$	AEBA	10	9.09e-11	0.093	4.36e-11	24
		CEBA	11	3.09e-11	0.12	9.76e-12	27
	$r = 5$	AEBA	10	5.71e-11	0.37	2.80e-11	55
		CEBA	11	9.87e-11	0.51	5.46e-11	72
$n = 10000$	$r = 2$	AEBA	11	1.11e-11	0.39	1.16e-11	24
		CEBA	13	3.94e-11	1.03	2.46e-11	33
	$r = 5$	AEBA	9	3.43e-12	0.56	2.77e-11	45
		CEBA	9	3.94e-11	0.70	5.38e-11	58

5 Conclusions

In this paper, we have proposed an alternative method based on the extended block Arnoldi process for solving the continuous Lyapunov equation. We have shown that the original equation can be transformed into an equivalent Sylvester equation. Moreover, the projected equation obtained with this approach involves the restrictions of A and A^{-1} to the projection subspace. The main impression is that the machine time, required to obtain an approximate solution for the equation under study, has been improved compared to the classical method .

References

[1] I. Abdaoui, L. Elbouyahyaoui, M. Heyouni, *An alternative extended block Arnoldi method for solving low-rank Sylvester equations*, Comput. Math. Appl. **78** (2019) 2817–2830.

Table 4: Obtained results by the EBA and AEBA methods for solving the Lyapunov equation (Example 2, Set 2).

Test		Method	Iterations	Residual norm	CPU time	$\|X-X_m\|_F$	Rank(X_m)
$n=2500$	$r=2$	AEBA	23	3.46e-11	2.10	1.26e-11	43
		CEBA	27	2.63e-11	2.65	1.15e-12	43
	$r=5$	AEBA	23	3.80e-11	1.87	1.35e-11	44
		CEBA	27	2.78e-11	2.18	1.59e-11	44
$n=5000$	$r=2$	AEBA	26	8.34e-11	9.89	2.83e-11	49
		CEBA	31	7.21e-11	11.45	1.13e-12	49
	$r=5$	AEBA	26	8.70e-11	9.70	7.35e-13	49
		CEBA	31	6.90e-11	11.04	2.79e-11	49
$n=7500$	$r=2$	AEBA	29	4.61e-11	27.8	1.53e-11	51
		CEBA	34	7.85e-11	31.5	1.10e-12	51
	$r=5$	AEBA	29	3.86e-11	29.3	1.23e-11	52
		CEBA	34	6.31e-11	33.5	6.41e-13	52

[2] L.A. Balzer, *Accelerated convergence of the matrix sign function method of solving Lyapunov, Riccati and other matrix equations*, Internat. J. Control **32** (1980) 1057–1078.

[3] R.H. Bartels, G.W. Stewart, *Solution of the matrix equation $AX+XB=C$*, Commun. ACM **15** (1922) 820–826.

[4] P. Benner, P. Krschner, *Computing real low-rank solutions of Sylvester equations by the factored ADI method*, Comput. Math.Appl. **67**, 1656-1672 (2014).

[5] R. Bellman, *Introduction to Matrix Analysis*, SIAM, 1997.

[6] T.A. Davis, Y. Hu, *The university of Florida sparse matrix collection*, ACM Trans. Math. Softw. **38** (2011) 1–25.

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

[8] N.S. Ellner, E.L. Wachspress, *Alternating direction implicit iteration for systems with complex spectra*, SIAM J. Numer. Anal. **28** (1991) 859–870.

[9] G. Golub, S. Nash, C. Van Loan, *A Hessenberg-Schur method for the problem $AX+XB=C$*, IEEE Trans. Autom. Control **24** (1979) 909–913.

[10] M. Heyouni, K. Jbilou, *An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation*, Electron. Trans. Numer. Anal. **33** (2009) 53–62.

[11] M. Heyouni, *Extended Arnoldi methods for large low-rank Sylvester matrix equations*, Appl. Numer. Math. **60** (2010)1171–1182.

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

[13] V. Kuera, *A review of the matrix Riccati equation*, Kybernetika 9 (1973) 42–61.

[14] J.-R. Li, F. Wang, J.K. White, *An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect*, Proceedings of the 36th annual ACM/IEEE Design Automation Conference, pages 1–6, 1999.

[15] J.-R. Li, J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl. 24 (2002) 260–280.

[16] T. Penzl, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput. 21 (1999)1401–1418.

[17] L.B. Rall, *Error in Digital Computation*, United States. Mathematics Research Center (Madison), 1965.

[18] N. Sandell, *On Newton's method for Riccati equation solution*, IEEE Trans. Autom. Control 19 (1974) 254–255.

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