Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 577–590
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2020.21859.2300

On some numerical methods for solving large-scale differential T-Lyapunov matrix equations

Lakhlifa Sadek*, Hamad Talibi Alaoui

Department of Mathematics, Faculty of Science, Chouaib Doukkali University, El Jadida, Morocco

(Communicated by Madjid Eshaghi Gordji)

Abstract

In this paper, we present two new approaches to solve large-scale differential T-Lyapunov matrix equations. The first one is based on the extended block Krylov subspaces, and the second is based on the extended global Krylov subspaces. The initial problem is projected onto an extended block (or global) Krylov subspaces to get a small-scale differential T-Lyapunov matrix equation. The latter problem is solved by iterative methods (Rosenbrock or BDF method), then the obtained solution is used to create a low-rank approximate solution of the original problem. This process is being replicated, which increases the dimension of the projection space until some planned accuracy is achieved. We give some new theoretical results and numerical experiments then we compare the new approaches.

Keywords: Extended block Krylov, Extended global Krylov, Low-rank, Krylov method, Differential T-Lyapunov matrix equation, T-Lyapunov matrix equation, T-Sylvester matrix equation, Rosenbrock method and BDF method
2020 MSC: Primary 65F50, Secondary 15A24

1 Introduction

We consider the differential T-Lyapunov matrix equation (DTLE in short) on the time interval $[t_0, T_f]$ of the form:

$$X'(t) = AX(t) + X(t)^T A^T + BB^T, \tag{1.1}$$

with the initial condition $X(t_0) = X_0$ where $A \in \mathbb{R}^{n\times n}$ is assumed to be large, sparse, and nonsingular and $B \in \mathbb{R}^{n\times s}$ is full rank matrix with $s \ll n$.

The differential T-Lyapunov matrix equation play a fundamental role in numerous problems in control, filter design theory, model reduction problems, differential equations and robust control problems; see, [1, 5] and the references therein. Small or medium-sized differential T-Lyapunov equations can be solved, for example, by Backward Differentiation Formula (BDF) method and Rosenbrock method [4, 14]. The DTLE (1.1) is equivalent to the following linear ordinary differential equation using the Kronecker formulation:

$$x'(t) = \mathfrak{A}x(t) + b, \quad x(t_0) = vec(X(t_0)), \tag{1.2}$$

*Corresponding author
Email addresses: lakhlifasadek@gmail.com; sadek.l@ucd.ac.ma (Lakhlifa Sadek), talibi_1@hotmail.fr (Hamad Talibi Alaoui)

Received: September 2020 *Accepted:* November 2020

where $\mathfrak{A} = I_n \otimes A + A \otimes I_n$, $b = vec(BB^T)$ and $vec(X)$ is the vector of $\mathbb{R}^{ns}$ defined by

$$vec(X) = [X_{11}, X_{21}, ..., X_{n1}, ..., X_{1s}, X_{2s}, ..., X_{ns}]^T \in \mathbb{R}^{ns}.$$

Reasonable size problems, which are given by (1.2), are solved by using an integration method. The Kronecker product $A \otimes B = [a_{ij}B]$, where $A = [a_{ij}]$, satisfies the properties: $(A \otimes B)(C \otimes D) = (AC \otimes BD)$, $(A \otimes B)^T = A^T \otimes B^T$ and $vec(AXB) = (B^T \otimes A)vec(X)$. For large differential T-Lyapunov matrix equations, we propose a new method based on projection onto extended block (or global) Krylov subspaces with an orthogonality Petrov-Galerkin condition.

The rest of the paper is organized as follows: In section 2, we recall the extended block Arnoldi process and some of its properties. In section 3, we recall the extended global Arnoldi process and some of its properties. In section 4, we present a low-rank method for solving large-scale differential T-Lyapunov matrix equations, using projections onto an extended block Krylov subspaces $\mathcal{K}_m^e(A, B)$ (or extended global Krylov subspaces $\mathcal{GK}_m^e(A, B)$), and Galerkin orthogonality condition. Some iterative methods for solving the obtained low dimensional problem are presented in section 5. In section 6, we show theoretical results related to the norm and the error. Finally, numerical examples are presented in section 7 to evaluate the performance of our approaches.

Throughout the paper, we use the following notations. The Frobenius inner product of the matrices X and Y is defined by $\langle X, Y\rangle_F = tr(X^TY)$, where $tr(Z)$ denotes the trace of a square matrix Z. The associated norm is the Frobenius norm denoted by $\|\cdot\|_F$,

$$\|A\|_F^2 = tr(A^TA) = \sum_i^n \sum_j^m a_{ij}^2 \quad \text{with} \quad A = [a_{ij}] \in \mathbb{R}^{n\times m}.$$

And the 2-norm denoted by $\|\cdot\|_2$, $\|A\|_2^2 = \lambda_{\max}(A^TA)$, and the 2-logarithmic norm of the matrix A is defined by $\mu_2(A) = \frac{\lambda_{max}(A+A^T)}{2}$. We also use the matrix product $\diamond$ defined in [3]. The following proposition gives some properties satisfied by this product.

Proposition 1.1. Let $A,\ B,\ C \in \mathbb{R}^{n\times ps}$, $D \in \mathbb{R}^{n\times n}$, $L \in \mathbb{R}^{p\times p}$ and $\alpha \in \mathbb{R}$. Then we have,

1. $(A + B)^T \diamond C = A^T \diamond C + B^T \diamond C$.
2. $A^T \diamond (B + C) = A^T \diamond B + A^T \diamond C$.
3. $(\alpha A)^T \diamond C = \alpha(A^T \diamond C)$.
4. $(A^T \diamond B)^T = B^T \diamond A$.
5. $A^T \diamond (B(L \otimes I_s)) = (A^T \diamond B)L$.
6. $(DA)^T \diamond B = A^T \diamond (D^TB)$.

A block matrix $\mathbb{V}_m = [V_1, V_2, ..., V_m]$ is F-orthonormal if $\mathbb{V}_m^T \diamond \mathbb{V}_m = I$. We have the following result.

Lemma 1.2. [11] Let $\mathbb{V}_m = [V_1, V_2, ..., V_m]$ be an $n \times ms$ F-orthonormal block matrix, $Z \in \mathbb{R}^{m\times s}$ and $Y \in \mathbb{R}^{ms\times q}$. Then we have

$$\|\mathbb{V}_m (Z \otimes I_s)\|_F = \|Z\|_F \quad \text{and} \quad \|\mathbb{V}_mY\|_F \leq \|Y\|_F.$$

2 The extended block Arnoldi process

We will consider extended block Krylov subspaces associated to the pair (A, B) and defined as follows

$$\mathcal{K}_m^e(A, B) = range\left\{B, A^{-1}B, AB, A^{-2}B, A^2B, ..., A^{m-1}B, A^{-m}B\right\}. \tag{2.1}$$

We mention the extended block Arnoldi (EBA) [8, 16] algorithm when applied to the pair (A, B). EBA is described in algorithm 1 as follows

Algorithm 1 The extended block Arnoldi algorithm (EBA)

Inputs: A an $n \times n$ matrix, B an $n \times s$ matrix and m an integer.

1. Compute the QR decomposition of $[B, A^{-1}B]$, i.e, $[B, A^{-1}B] = V_1\Lambda$;
2. Set $\mathcal{V}_0 = [\ \ ]$;
3. **For** $j = 1, 2, 3, ..., m$
4. Set $V_j^{(1)} = V_j(:, 1 : s)$ et $V_j^{(2)} = V_j(:, s+1 : 2s)$
5. $\mathcal{V}_j = [\mathcal{V}_{j-1}, V_j]; \widehat{V}_{j+1} = [AV_j^{(1)}, A^{-1}V_j^{(2)}]$;
6. **For** $i = 1, ..., j$,
7. $T_{i,j} = V_i^T \widehat{V}_{j+1}$;
8. $\widehat{V}_{j+1} = \widehat{V}_{j+1} - V_i T_{i,j}$;
9. **End For** i
10. Compute the QR decomposition of U i.e., $\widehat{V}_{j+1} = V_{j+1}T_{j+1,j}$;
11. **End For** j.

Output: $\mathcal{V}_m = [V_1, ..., V_m]$, $\mathcal{T}_m = [T_{i,j}]$.

After m steps, Algorithm 1 built an orthonormal basis $\mathcal{V}_m$ ($\mathcal{V}_m^T\mathcal{V}_m = I_{2ms}$) of the extended block Krylov subspace $\mathcal{K}_m^e(A, B)$. Let $\mathcal{T}_m = \mathcal{V}_m^T A \mathcal{V}_m$ be an $2s \times 2s$ block upper Hessenberg matrix. Then we have the following relations

$$A\mathcal{V}_m = \mathcal{V}_m\mathcal{T}_m + V_{m+1}T_{m+1,m}E_m^T = \mathcal{V}_{m+1}\begin{bmatrix} \mathcal{T}_m \\ T_{m+1,m}E_m^T \end{bmatrix}, \quad \mathcal{V}_m^T B = \nabla_1\Lambda_{11} \tag{2.2}$$

where $E_m^T = [0_{2s\times 2s(m-1)}, I_{2s}]$ is the matrix formed with the last $2s$ columns of the $2ms \times 2ms$ identity matrix I_{2ms}, $\nabla_1 = [I_s, 0_{s\times(2m-1)s}]^T$ and Λ_{11} is the $s \times s$ matrix obtained from the QR decomposition

$$[B, A^{-1}B] = V_1\Lambda \quad \text{with} \quad \Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ 0_{s\times s} & \Lambda_{22} \end{bmatrix}.$$

3 The extended global Arnoldi process

In this section, we recall the extended global Krylov subspace and the extended global Arnoldi process. Let B be a matrix of dimension $n \times s$ and A be a matrix of dimension $n \times n$, then the extended global Krylov subspace associated to (A, B) is given by

$$\mathcal{GK}_m^g(A, B) = span\left\{B, A^{-1}B, AB, A^{-2}B, A^2B, ..., A^{m-1}B, A^{-m}B\right\}. \tag{3.1}$$

The extended global Arnoldi process constructs an F-orthonormal basis $\{V_1, V_2, ..., V_m\}$ of the extended global Krylov subspace $\mathcal{GK}_m^g(A, B)$ [15]. The algorithm is summarized as follows

Algorithm 2 The extended global Arnoldi process (EGA)

Inputs: $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times s}$ and m an integer.

1. Compute the global QR $[B, A^{-1}B]$, i.e , $[B, A^{-1}B] = V_1(\Omega \otimes I_s)$;
2. Set $\mathbb{V}_0 = [\ ]$;
3. **For** $j = 1, ..., m$
4. Set $V_j^{(1)} = V_j(:, 1 : s)$; $V_j^{(2)} = V_j(:, s+1 : 2s)$;
5. $\mathbb{V}_j = [\mathbb{V}_{j-1}, V_j]; U = [AV_j^{(1)}, A^{-1}V_j^{(2)}]$;
6. **For** $i = 1, ..., j$
7. $H_{i,j} = V_i^T \diamond U$;
8. $U = U - V_i(H_{i,j} \otimes I_s)$;
9. **End for(i)**.
10. Compute the global QR decomposition of U, i.e., $U = V_{j+1}(H_{j+1,j} \otimes I_s)$;
11. **End for(j)**.

Outputs: $\mathbb{V}_m = [V_1, ..., V_m] \in \mathbb{R}^{n\times 2ms}$ and $\mathbb{H}_m = [H_{i,j}] \in \mathbb{R}^{2m\times 2m}$

The obtained $n \times 2ms$ matrix $\mathbb{V}_m = [V_1, V_2, \ldots, V_m]$ is F-orthonormal $\mathbb{V}_m^T \diamond \mathbb{V}_m = I_{2m}$, of the extended global Krylov subspace $\mathcal{GK}_m^g(A, B)$ with $V_i \in \mathbb{R}^{n\times 2s}$ and $2m \times 2m$ upper block Hessenberg matrix

$$\mathbb{H}_m = \mathbb{V}_m^T \diamond (A\mathbb{V}_m).$$

We have the following relation

$$\begin{aligned} A\mathbb{V}_m &= \mathbb{V}_m(\mathbb{H}_m \otimes I_s) + V_{m+1}(H_{m+1,m}(E_m^e)^T \otimes I_s) && (3.2)\\ &= \mathbb{V}_{m+1}(\begin{bmatrix} \mathbb{H}_m \\ H_{m+1,m}(E_m^e)^T \end{bmatrix} \otimes I_s), \quad \mathbb{V}_m = \mathbb{V}_{m+1}\begin{bmatrix} I_{2sm} \\ 0_{2s\times 2sm} \end{bmatrix}, && (3.3)\end{aligned}$$

where $(E_m^e)^T = [0_{2\times 2(m-1)}, I_2]$ is the matrix formed with the last 2 columns of the $2m \times 2m$ identity matrix I_{2m}.

4 Krylov method

4.1 Low-rank method by extended block Arnoldi

In this section, we show how to obtain low-rank approximate solutions to the differential T-Lyapunov matrix equation (1.1) by first projecting directly the initial problem onto extended block Krylov subspaces and then solving the obtained low-dimensional differential matrix equation. We, firstly, apply the extended block Arnoldi algorithm to (A, B) to get the matrices

$$\mathcal{V}_m = [V_1, ..., V_m] \quad \text{and} \quad \mathcal{T}_m = \mathcal{V}_m^T A \mathcal{V}_m$$

with $\mathcal{V}_m$ whose columns form orthonormal bases of the extended block Krylov subspaces $\mathcal{K}_m^e(A, B)$ and $\mathcal{T}_m$ the upper block Hessenberg matrix. After m iterations, we consider the low-rank approximate solutions $X_m(t)$ of exact solution $X(t)$ to equation (1.1) of the form

$$X_m(t) = \mathcal{V}_m Y_m(t) \mathcal{V}_m^T. \tag{4.1}$$

The matrix function $Y_m(t) \in \mathbb{R}^{2ms\times 2ms}$ can be obtained by the Petrov-Galerkin orthogonality condition

$$\mathcal{V}_m^T R_m(t) \mathcal{V}_m = 0_{2ms\times 2ms}, \quad t \in [t_0, T_f], \tag{4.2}$$

where $R_m(t) = X_m'(t) - AX_m(t) - X_m(t)^T A^T - BB^T$ is the residual.

Theorem 4.1. Let $Y_m(t)$ be the matrix function defined by (4.1), then it satisfies the following low-order differential T-Lyapunov matrix equation

$$\begin{cases} Y_m'(t) = \mathcal{T}_m Y_m(t) + Y_m(t)^T \mathcal{T}_m^T + B_m B_m^T, \\ Y_m(t_0) = \mathcal{V}_m^T X_0 \mathcal{V}_m, \end{cases} \tag{4.3}$$

where $B_m = \nabla_1 \Lambda_{11}$.

Proof . Using the condition (4.1) and the relation (4.2) we obtain the reduced differential T-Lyapunov matrix equation $Y_m'(t) = \mathcal{T}_m Y_m(t) + Y_m(t)^T \mathcal{T}_m^T + B_m B_m^T$. $\square$

We have now to solve the last differential matrix equation (4.3) by Resenbrock method or Backward Differentiation Formula (BDF) method.

Next, we give a result that allows us to compute the norm of the residual without forming the approximate solution $X_m(t)$ at each step m of the extended block Arnoldi process (EBA). The approximate solution $X_m(t)$ is computed in a factored form only when convergence is achieved.

Theorem 4.2. Let the matrix function $X_m(t) = \mathcal{V}_m Y_m(t) \mathcal{V}_m^T$ be the approximate solution obtained at step m by extended block Arnoldi process. Then, the Frobenius norm of the residual $R_m(t)$ associated to the approximate solution $X_m(t)$ satisfies the following relation

$$\|R_m(t)\|_F = \sqrt{2}\|T_{m+1,m}\underline{Y}_m(t)\|_F, \tag{4.4}$$

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

Proof . We have $R_m(t) = X'_m(t) - AX_m(t) - X_m(t)^T A^T - BB^T$ and using the relations (4.2) and (2.2), so

$$\begin{aligned} R_m(t) &= \mathcal{V}_{m+1} \begin{bmatrix} Y'_m(t) & 0_{2ms\times 2s} \\ 0_{2s\times 2ms} & 0_{2s\times 2s} \end{bmatrix} \mathcal{V}^T_{m+1} - \mathcal{V}_{m+1} \begin{bmatrix} \mathcal{T}_m Y_m(t) & 0_{2ms\times 2s} \\ T_{m+1,m}E_m^T Y_m(t) & 0_{2s\times 2s} \end{bmatrix} \mathcal{V}^T_{m+1} \\ &\quad -\mathcal{V}_{m+1} \begin{bmatrix} Y_m(t)^T \mathcal{T}_m^T & Y_m(t)^T E_m T^T_{m+1,m} \\ 0_{2s\times 2ms} & 0_{2s\times 2s} \end{bmatrix} \mathcal{V}^T_{m+1} - \mathcal{V}_{m+1} \begin{bmatrix} \mathcal{V}_m^T BB^T \mathcal{V}_m & 0_{2ms\times 2s} \\ 0_{2s\times 2ms} & 0_{2s\times 2s} \end{bmatrix} \mathcal{V}^T_{m+1} \\ &= \mathcal{V}_{m+1} \begin{bmatrix} Y'_m(t) - \mathcal{T}_m Y_m(t) + Y_m^T(t)\mathcal{T}_m^T - B_m B_m^T & Y_m(t)^T E_m T^T_{m+1,m} \\ -T_{m+1,m}E_m^T Y_m(t) & 0_{2s\times 2s} \end{bmatrix} \mathcal{V}^T_{m+1}. \end{aligned}$$

and $Y_m(t)$ is the solution of the reduced differential T-Lyapunov matrix equation (4.3). We get

$$R_m(t) = \mathcal{V}_{m+1} \begin{bmatrix} 0_{2ms\times 2ms} & -Y_m(t)^T E_m T^T_{m+1,m} \\ -T_{m+1,m}E_m^T Y_m(t) & 0_{2s\times 2s} \end{bmatrix} \mathcal{V}^T_{m+1}.$$

Since $\mathcal{V}_{m+1}$ are the orthonormal matrices, so we have $\|R_m(t)\|_F = \sqrt{2}\|T_{m+1,m}E_m^T Y_m(t)\|_F$. □

To save memory, the approximate solution $X_m(t) = \mathcal{V}_m Y_m(t)\mathcal{V}_m^T$ can be given as a product of two low-rank matrices. Thus, we consider the singular value decomposition of the $2ms \times 2ms$ matrix $Y_m = UDV^T$ where D is the diagonal matrix of the singular values of $Y_m(t)$ sorted in decreasing order. Let U_l and V_l be the $2ms \times l$ matrices of the first l columns of U and V respectively, corresponding to the l singular values of magnitude greater than some tolerance d_{tol}. We obtain the truncated singular value decomposition $Y_m \approx U_l D_l V_l^T$, where $D_l = diag[\lambda_1, ..., \lambda_l]$. Setting $Z_{m,1} = \mathcal{V}_m U_l D_l^{\frac{1}{2}}$, $Z_{m,2} = \mathcal{V}_m V_l D_l^{\frac{1}{2}}$, it follows that

$$X_m = Z_{m,1} Z^T_{m,2}. \tag{4.5}$$

This result is important for large-scale problems to decrease central processing unit (CPU) time and memory requirements; the approximate solution could be given as a product of low-rank matrices.

The following result shows that the approximate solution $X_m(t)$ is an exact solution of perturbed differential T-Lyapunov matrix equation.

Theorem 4.3. Let the matrix function $X_m(t)$ be the approximate solution given by (4.1). Then we have

$$X'_m(t) = (A - F_m)X_m(t) + X_m(t)^T (A - F_m)^T + BB^T, \tag{4.6}$$

where $F_m = V_{m+1}T_{m+1,m}\mathcal{V}_m^T$.

Proof . Multiplying (4.3) from left by $\mathcal{V}_m$ and from right by $\mathcal{V}_m^T$, we obtain

$$\mathcal{V}_m Y'_m(t)\mathcal{V}_m^T = \mathcal{V}_m \mathcal{T}_m Y_m(t)\mathcal{V}_m^T + \mathcal{V}_m Y_m(t)^T \mathcal{T}_m^T \mathcal{V}_m^T + \mathcal{V}_m B_m B_m^T \mathcal{V}_m^T,$$

then,

$$\begin{aligned} X'_m(t) &= [A\mathcal{V}_m - V_{m+1}T_{m+1,m}E_m^T]Y_m(t)\mathcal{V}_m^T + \mathcal{V}_m Y_m(t)^T [A\mathcal{V}_m - V_{m+1}T_{m+1,m}E_m^T]^T + BB^T \\ &= [A - V_{m+1}T_{m+1,m}E_m^T\mathcal{V}_m^T]X_m(t) + X_m(t)^T[A - V_{m+1}T_{m+1,m}E_m^T\mathcal{V}_m^T]^T + BB^T, \end{aligned}$$

so $X'_m(t) = (A - F_m)X_m(t) + X_m(t)^T (A - F_m)^T + BB^T$. □

4.2 Low-rank method by extended global Arnoldi

In this section, we show how to obtain low-rank approximate solutions to the differential T-Lyapunov matrix equation (1.1) by first projecting directly the initial problem onto extended global Krylov subspaces and then solving the obtained low dimensional problem. We apply the extended global Arnoldi algorithm to the pair (A, B) to get the matrices

$$\mathbb{V}_m = [V_1, ..., V_m] \text{ and } \mathbb{H}_m = \mathbb{V}_m^T \diamond (A\mathbb{V}_m)$$

with $\mathbb{V}_m$ whose columns form orthonormal bases of the extended global Krylov subspaces $\mathcal{GK}_m^e(A, B)$ and $\mathbb{H}_m$ the upper block Hessenberg matrix. Let $X_m(t)$ be the desired low-rank approximate solution given as

$$X_m(t) = \mathbb{V}_m(\mathbb{X}_m(t) \otimes I_s)\mathbb{V}_m^T. \tag{4.7}$$

where the matrix function $\mathbb{X}_m(t) \in \mathbb{R}^{2m\times 2m}$ is the solution of the low-order differential T-Lyapunov matrix equation

$$\mathbb{X}'_m(t) = \mathbb{H}_m\mathbb{X}_m(t) + \mathbb{X}_m(t)^T\mathbb{H}_m^T + \|B\|_F^2 e_1^{(2m)}(e_1^{(2m)})^T, \tag{4.8}$$

where $(e_1^{(2m)})^T = [1, 0, ..., 0] \in \mathbb{R}^{2m}$.

The next result shows how to compute the norm of $R_m(t)$ without forming the approximate solution $X_m(t)$ which is computed in a factored form only when convergence is achieved.

Theorem 4.4. The Frobenius norm of the residual matrix function $R_m(t)$ associated to the approximation matrix function $X_m(t)$ obtained at step m by the extended global Arnoldi method satisfies the relation

$$\|R_m(t)\|_F \leq \sqrt{2}\|H_{m+1,m}\underline{\mathbb{X}}_m(t)\|_F, \tag{4.9}$$

where $\underline{\mathbb{X}}_m(t)$ is the $2 \times 2m$ matrix corresponding to the last 2 rows of $\mathbb{X}_m(t)$.

Proof . Using the relation (4.7) and the fact that $R_m(t) = X'_m(t) - AX_m(t) - X_m(t)^TA^T - BB^T$, therefore

$$R_m(t) = \mathbb{V}_m(\mathbb{X}'_m(t) \otimes I_s)\mathbb{V}_m^T - A\mathbb{V}_m(\mathbb{X}_m(t) \otimes I_s)\mathbb{V}_m^T - \mathbb{V}_m(\mathbb{X}_m(t)^T \otimes I_s)\mathbb{V}_m^TA^T - BB^T,$$

using the relation (3.2), we obtain

$$\begin{aligned}
R_m(t) \;&=\; \mathbb{V}_{m+1}\begin{bmatrix} \mathbb{X}'_m(t)\otimes I_s & 0_{2ms\times 2s} \\ 0_{2s\times 2ms} & 0_{2s\times 2s}\end{bmatrix}\mathbb{V}_{m+1}^T - \mathbb{V}_{m+1}\begin{bmatrix} \mathbb{H}_m\mathbb{X}_m(t)\otimes I_s & 0_{2ms\times 2s} \\ H_{m+1,m}(E_m^e)^T\mathbb{X}_m(t)\otimes I_s & 0_{2s\times 2s}\end{bmatrix}\mathbb{V}_{m+1}^T \\
&\quad -\mathbb{V}_{m+1}\begin{bmatrix} \mathbb{X}_m(t)^T\mathbb{H}_m^T\otimes I_s & \mathbb{X}_m(t)^TE_m^eH_{m+1,m}^T\otimes I_s \\ 0_{2s\times 2ms} & 0_{2s\times 2s}\end{bmatrix}\mathbb{V}_{m+1}^T \\
&\quad -\mathbb{V}_{m+1}\begin{bmatrix} \mathbb{V}_m^T\diamond BB^T\diamond\mathbb{V}_m & 0_{2ms\times 2s} \\ 0_{2s\times 2ms} & 0_{2s\times 2s}\end{bmatrix}\mathbb{V}_{m+1}^T \\
&=\; \mathbb{V}_{m+1}\left(\begin{bmatrix} \mathbb{X}'_m(t) - \mathbb{H}_m\mathbb{X}_m(t) - \mathbb{X}_m(t)^T\mathbb{H}_m^T - B_mB_m^T & -\mathbb{X}_m(t)^TE_m^eH_{m+1,m}^T \\ -H_{m+1,m}(E_m^e)^T\mathbb{X}_m(t) & 0_{2\times 2}\end{bmatrix}\otimes I_s\right)\mathbb{V}_{m+1}^T
\end{aligned}$$

As $\mathbb{X}_m(t)$ exact solution of the low dimensional differential T-Lyapunov matrix equation (4.8), so

$$R_m(t) = \mathbb{V}_{m+1}\left(\begin{bmatrix} 0_{2m\times 2m} & -\mathbb{X}_m(t)^TE_m^eH_{m+1,m}^T \\ -H_{m+1,m}(E_m^e)^T\mathbb{X}_m(t) & 0_{2\times 2}\end{bmatrix}\otimes I_s\right)\mathbb{V}_{m+1}^T,$$

so

$$\begin{aligned}
\|R_m(t)\|_F^2 \;&=\; \left\|\mathbb{V}_{m+1}\left(\begin{bmatrix} 0_{2m\times 2m} & -\mathbb{X}_m(t)^TE_m^eH_{m+1,m}^T \\ -H_{m+1,m}(E_m^e)^T\mathbb{X}_m(t) & 0_{2\times 2}\end{bmatrix}\otimes I_s\right)\mathbb{V}_{m+1}^T\right\|_F^2 \\
&\leq\; \left\|\left(\begin{bmatrix} 0_{2m\times 2m} & -\mathbb{X}_m(t)^TE_m^eH_{m+1,m}^T \\ -H_{m+1,m}(E_m^e)^T\mathbb{X}_m(t) & 0_{2\times 2}\end{bmatrix}\otimes I_s\right)\mathbb{V}_{m+1}^T\right\|_F^2 \\
&\leq\; \left\|\begin{bmatrix} 0_{2m\times 2m} & -\mathbb{X}_m(t)^TE_m^eH_{m+1,m}^T \\ -H_{m+1,m}(E_m^e)^T\mathbb{X}_m(t) & 0_{2\times 2}\end{bmatrix}\right\|_F^2.
\end{aligned}$$

So $\|R_m(t)\|_F^2 \leq 2\|H_{m+1,m}(E_m^e)^T\mathbb{X}_m(t)\|_F^2$. $\square$

The following result shows that the approximate solution $X_m(t)$ is an exact solution of a perturbed differential T-Lyapunov matrix equation.

Theorem 4.5. Let $X_m(t)$ be the approximate solution given by (1.1). Then we have

$$X'_m(t) = (A - F_m)X_m(t) + X_m(t)^T(A - F_m)^T + BB^T, \tag{4.10}$$

where $F_m = V_{m+1}(H_{m+1,m}(E_m^e)^T \otimes I_s)(\mathbb{V}_m^T\mathbb{V}_m)^{-1}\mathbb{V}_m^T$.

Proof . Multiplying (4.8) from left by $\mathbb{V}_m$ and from right by $\mathbb{V}_m^T$, we obtain

$$\begin{aligned}\mathbb{V}_m(\mathbb{X}'_m(t)\otimes I_s)\mathbb{V}_m^T &= \mathbb{V}_m(\mathbb{H}_m\mathbb{X}_m(t)\otimes I_s)\mathbb{V}_m^T + \mathbb{V}_m(\mathbb{X}_m(t)^T\mathbb{H}_m^T\otimes I_s)\mathbb{V}_m^T + \mathbb{V}_m(B_mB_m^T\otimes I_s)\mathbb{V}_m^T\\ &= \mathbb{V}_m(\mathbb{H}_m\otimes I_s)(\mathbb{X}_m(t)\otimes I_s)\mathbb{V}_m^T + \mathbb{V}_m(\mathbb{X}_m(t)^T\otimes I_s)(\mathbb{H}_m^T\otimes I_s)\mathbb{V}_m^T\\ &\quad +\mathbb{V}_m(B_m\otimes I_s)(B_m^T\otimes I_s)\mathbb{V}_m^T,\end{aligned}$$

since $A\mathbb{V}_m = \mathbb{V}_m(\mathbb{H}_m\otimes I_s) + V_{m+1}(H_{m+1,m}(E_m^e)^T\otimes I_s)$, so

$$\begin{aligned}X'_m(t) &= [A\mathbb{V}_m - V_{m+1}(H_{m+1,m}(E_m^e)^T\otimes I_s)](\mathbb{X}_m(t)\otimes I_s)\\ &\quad +(\mathbb{X}_m(t)\otimes I_s)^T[\mathbb{V}_m^TA^T - (E_mH_{m+1,m}^T\otimes I_s)V_{m+1}^T] + BB^T\\ &= [A - V_{m+1}(H_{m+1,m}(E_m^e)^T\otimes I_s)(\mathbb{V}_m^T\mathbb{V}_m)^{-1}\mathbb{V}_m^T]X_m(t)\\ &\quad +X_m(t)^T[A^T - \mathbb{V}_m(\mathbb{V}_m^T\mathbb{V}_m)^{-1}(E_m^e(H_{m+1,m})^T\otimes I_s)V_{m+1}^T] + BB^T,\end{aligned}$$

so $X'_m(t) = (A-F_m)X_m(t) + X_m(t)^T(A-F_m)^T + BB^T$. □

The approximate solution $X_m(t)$ can be given as a product of two low-rank matrices, i.e., $X_m = Z_1\, Z_2^T$ where the matrix Z_1 and Z_2 are of low rank (lower than $2m$). Consider the singular value decomposition of the $2m\times 2m$ matrix

$$\mathbb{X}_m(t) = \widetilde{G}_1\Sigma\,\widetilde{G}_2^T,$$

where Σ is the diagonal matrix of the singular values of $\mathbb{X}_m$ sorted in decreasing order. Let $\mathbb{X}_{1,l}$ and $\mathbb{X}_{2,l}$ be the $2m\times l$ matrices of the first l columns of $\widetilde{G}_1$ and $\widetilde{G}_2$ respectively, corresponding to the l singular values of magnitude greater than some tolerance d_{tol}. We obtain the truncated SVD

$$\mathbb{X}_m(t) \approx \mathbb{X}_{1,l}\,\Sigma_l\,\mathbb{X}_{2,l}{}^T,$$

where $\Sigma_l = \mathrm{diag}[\sigma_1,\ldots,\sigma_l]$. Setting $Z_{1,m} = \mathbb{V}_m\left(\mathbb{X}_{1,l}\,\Sigma_l^{1/2}\otimes I_s\right)$ and $Z_{2,m} = \mathbb{V}_m\left(\mathbb{X}_{2,l}\,\Sigma_l^{1/2}\otimes I_s\right)$, leads to

$$X_m \approx Z_{1,m}\,Z_{2,m}^T. \tag{4.11}$$

This is very important for large problems when one doesn't need to compute and store the approximate solution X_m at each iteration, see [16, 15].

In the next section, we give some iterative methods to solve the reduced order differential T-Lyapunov matrix equations (4.3) and (4.8).

5 Iterative methods for solving the reduced differential T-Lyapunov matrix equation

5.1 Rosenbrock method

In this section, we will apply Rosenbrock method [4, 14] to the low dimensional differential T-Lyapunov matrix equation (4.3). The new approximation $Y_{m,j+1}$ of $Y_m(t_{j+1})$ obtained at step $j+1$ is defined, by the relations

$$Y_{m,j+1} = Y_{m,j} + \frac{3}{2}P_1 + \frac{1}{2}P_2, \tag{5.1}$$

where P_1 and P_2 solve the following T-Sylvester matrix equations in these two articles [12, 7].

$$\mathbb{T}_{m,1}P_1 + P_1^T\mathbb{T}_{m,2} = \mathcal{T}_mY_{m,j} + Y_{m,j}^T\mathcal{T}_m^T + B_mB_m^T, \tag{5.2}$$

and

$$\mathbb{T}_{m,1}P_2 + P_2^T\mathbb{T}_{m,2} = \mathcal{T}_m(Y_{m,j}+P_1) + (Y_{m,j}+P_1)^T\mathcal{T}_m^T + B_mB_m^T - \frac{2}{h}P_1, \tag{5.3}$$

where $\mathbb{T}_{m,1} = \frac{1}{h}I_{2ms} - \gamma\mathcal{T}_m$, and $\mathbb{T}_{m,2} = -\gamma\mathcal{T}_m^T$.

We summarize the steps of the Rosenbrock method in the following algorithm

Algorithm 3 The Rosenbrock method for (4.3)

Inputs: $\mathcal{T}_m, B_m, t_0, T_f$.

1. Choose h.
2. Compute: $N = \frac{T_f - t_0}{h}$
3. Compute: $\mathbb{T}_{m,1}, \mathbb{T}_{m,2}$
4. For $j = 1 : N$
5. Solve (5.2) and (5.3).
6. Calculate $Y_{m,j+1}$ by (5.1)
7. End For j.

Output: $Y_{m,j+1}$.

In the same way, we solve the reduced-order differential T-Lyapunov matrix equation (4.8) by Rosenbrock method, it is enough to replace $\mathcal{T}_m$ by $\mathbb{H}_m$.

We summarize the steps of the approach EBA-Rosenbrock method for the extended block Arnoldi and EGA-Rosenbrock methods for the extended global Arnoldi in the following algorithm.

Algorithm 4 The EBA-Rosenbrock method or EGA-Rosenbrock method for solving DTLE.

Inputs: A and B an matrix.

1. Choose a tolerance $d_{tol}, tol > 0$ and an integer m_{max}.
2. For $m = 1 : m_{max}$
3. Apply EBA Algorithm 1 for the EBA-Rosenbrock method to (A, B) to get the matrices $\mathcal{V}_m$, $\mathcal{T}_m$ or apply EGA Algorithm 2 for the EGA-Rosenbrock method to (A, B) to get the matrices $\mathbb{V}_m$, $\mathbb{H}_m$.
4. Apply Algorithm 3 to solve the low dimensional differential matrix equation.
5. If $\|R_m\|_F < tol$.
6. End For m
7. Compute the approximate solution X_m in the factored form given by the relation (4.5) for the EBA-Rosenbrock method or the relation (4.11) for the EGA-Rosenbrock method.

Output: X_m.

5.2 BDF method

We use the Backward Differentiation Formula method for solving the reduced differential T-Lyapunov matrix equation (4.3). At each time t_j, let $Y_{m,j}$ be the approximation of $Y_m(t_j)$, where Y_m is a solution of (4.3). Then, the new approximation $Y_{m,j+1}$ of $Y_m(t_{j+1})$ obtained at step $j+1$ by BDF2 is defined by the implicit relation

$$Y_{m,j+1} = \frac{4}{3}Y_{m,j} - \frac{1}{3}Y_{m,j-1} + \frac{2h}{3}(\mathcal{T}_m Y_{m,j+1} + Y_{m,j+1}^T \mathcal{T}_m^T + B_m B_m^T), \tag{5.4}$$

where $h = t_{j+1} - t_j$ is the step size. The approximation $Y_{m,j+1}$ solves the following matrix equation

$$-Y_{m,j+1} + \frac{2h}{3}(\mathcal{T}_m Y_{m,j+1} + Y_{m,j+1}^T \mathcal{T}_m^T + B_m B_m^T) + \frac{4}{3}Y_{m,j} - \frac{1}{3}Y_{m,j-1} = 0. \tag{5.5}$$

Let $\mathbb{T}_m = \frac{2h}{3}\mathcal{T}_m - \frac{1}{2}I_{2ms}$ and $\mathbb{Q}_{m,j+1} = \frac{2h}{3}B_m B_m^T + \frac{4}{3}Y_{m,j} - \frac{1}{3}Y_{m,j-1}$, therefore, we can write (5.5) as the following T-Sylvester matrix equation:

$$\mathbb{T}_m Y_{m,j+1} + Y_{m,j+1}^T \mathbb{T}_m^T = -\mathbb{Q}_{m,j+1}. \tag{5.6}$$

We summarize the steps of the BDF method in the following algorithm

Algorithm 5 The BDF method for (4.3)

Inputs: $\mathcal{T}_m, B_m, t_0, T_f$.

1. Choose h.
2. Compute: $N = \frac{T_f - t_0}{h}$
3. Compute: $\mathbb{T}_m$
4. For $j = 1 : N$
5. Calculate $\mathbb{Q}_{m,j+1}$
6. Solve (5.6)
7. End For j.

Output: $Y_{m,j+1}$.

In the same way, we solve the reduced order differential T-Lyapunov matrix equation (4.8) by BDF method, it is enough to replace $\mathcal{T}_m$ by $\mathbb{H}_m$.

We summarize the steps of the approach EBA-BDF method for the extended block Arnoldi and EGA-BDF method for the extended global Arnoldi in the following algorithm.

Algorithm 6 The EBA-BDF method or EGA-BDF method for solving DTLE.

Inputs: A and B an matrix.

1. Choose a tolerance $d_{tol}, tol > 0$ and an integer m_{max}.
2. For $m = 1 : m_{max}$
3. Apply EBA Algorithm 1 for the EBA-BDF method to (A, B) to get the matrices $\mathcal{V}_m$, $\mathcal{T}_m$ or apply EGA Algorithm 2 for the EGA-BDF method to (A, B) to get the matrices $\mathbb{V}_m$, $\mathbb{H}_m$.
4. Apply Algorithm 5 to solve the low dimensional differential matrix equation.
5. If $\|R_m\|_F < tol$.
6. End For m
7. Compute the approximate solution X_m in the factored form given by the relation (4.5) for the EBA-BDF method or the relation (4.11) for the EGA-BDF method.

Output: X_m.

6 The norm of the error

In this section, we give results related to the norm of the error. The next result states that the error $\mathcal{E}_m(t) = X(t) - X_m(t)$ also satisfies a differential T-Lyapunov matrix equation.

Theorem 6.1. The error $\mathcal{E}_m(t)$ satisfies the following differential T-Lyapunov matrix equation

$$\mathcal{E}'_m(t) = A\mathcal{E}_m(t) + \mathcal{E}_m(t)^T A^T - R_m(t). \tag{6.1}$$

Proof . We have

$$\begin{aligned}
\mathcal{E}'_m(t) &= X'(t) - X'_m(t) \\
&= AX(t) + X(t)^T A^T + BB^T - AX_m(t) - X_m(t)^T A^T - BB^T - R_m(t) \\
&= A(X(t) - X_m(t)) + (X(t) - X_m(t))^T A^T - R_m(t) \\
&= A\mathcal{E}_m(t) + \mathcal{E}_m(t)^T A^T - R_m(t),
\end{aligned}$$

so $\mathcal{E}'_m(t) = A\mathcal{E}_m(t) + \mathcal{E}_m(t)^T A^T - R_m(t)$. □

Theorem 6.2. Let $\beta = \max\limits_{\tau \in [t_0, t]} \|R_m(\tau)\|_F$. The norm of the error $\mathcal{E}_m(t)$ satisfies the following upper bound

$$\|\mathcal{E}_m(t)\|_F \leq \|\mathcal{E}_m(t_0)\|_F e^{2\|A\|_2 (t - t_0)} + \frac{\beta}{2\|A\|_2}(e^{2\|A\|_2 (t - t_0)} - 1).$$

Proof . Applying vec and $\|.\|_2$ the differential T-Lyapunov matrix equation (6.1), we obtain

$$\begin{array}{rcl} \|vec(\mathcal{E}_m(t))\|_2 & \leq & \|vec(A\mathcal{E}_m(t))\|_2 + \|vec(\mathcal{E}_m(t)^T A^T)\|_2 + \|vec(R_m(t))\|_2 \\ & \leq & \|I_n \otimes Avec(\mathcal{E}_m(t))\|_2 + \|A \otimes I_n vec(\mathcal{E}_m(t)^T)\|_2 + \|vec(R_m(t))\|_2 \\ & \leq & \|I_n \otimes A\|_2 \|vec(\mathcal{E}_m(t))\|_2 + \|A \otimes I_n\|_2 \|vec(\mathcal{E}_m(t)^T)\|_2 + \|vec(R_m(t))\|_2, \end{array}$$

As $\|vec(\mathcal{E}_m(t))\|_2 = \|\mathcal{E}_m(t)\|_F$, so

$$\begin{array}{rcl} \|\mathcal{E}_m(t)\|_F & \leq & \|A\|_2\|\mathcal{E}_m(t)\|_F + \|A\|_2\|\mathcal{E}_m(t)\|_F + \|R_m(t)\|_F \\ & \leq & 2\|A\|_2\|\mathcal{E}_m(t)\|_F + \|R_m(t)\|_F, \end{array}$$

through the Grönwall's lemma, so

$$\|\mathcal{E}_m(t)\|_F \quad \leq \quad \|\mathcal{E}_m(t_0)\|_F e^{2\|A\|_2(t-t_0)} + \frac{\max\limits_{\tau \in [t_0,t]} \|R_m(\tau)\|_F}{2\|A\|_2}(e^{2\|A\|_2(t-t_0)} - 1).$$

□

In Figure 1, diagram of the proportion of the Frobenius error norm $\log(||X_m^{EGA} - X_{exact}^{ode23s}||_F)$ in functions of the numbers iterations, and the time interval was $[0, 1]$, $n = 49$, we used $s = 2$. The tolerance was set to 10^{-9} for the stop test on the residual. For the EGA-BDF method, we used a constant timestep $h = 0.01$.

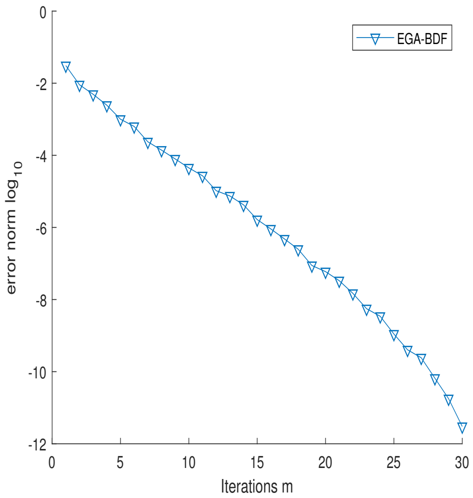

Figure 1: The $\log(||X_m^{EGA} - X_{exact}^{ode23s}||_F)$ in functions of the numbers iterations

7 Numerical experiments

In this section, we compare the approach presented in this article with the solution that is given by the command in MATLAB: ode23s, and trace the residual according to the iterations. All the experiments were performed on a laptop with an Intel Core i3 processor and 4GB of RAM. The algorithms were coded in MATLAB R2018b, the matrix B is generated randomly and their coefficients were uniformly distributed in $[0, 1]$. We compare the two approaches proposed in algorithm 6 (EBA-BDF method and EGA-BDF method), and algorithm 4 (EGA-Rosenbrock method and EBA-Rosenbrock method).

7.1 Example 1

In this example, the matrix A is obtained from the centered finite difference discretization of the operators:

$$\mathcal{L}(u) = \Delta u + f_1(x, y)\frac{\partial u}{\partial x} + f_2(x, y)\frac{\partial u}{\partial y} + f_3(x, y)u$$

on the unit square $[0,1]\times[0,1]$ with homogeneous dirichlet boundary conditions. The number of inner grid points in each direction was n_0 for the operators $\mathcal{L}$. The matrix A was obtained from the discretization of the operator $\mathcal{L}$ with the dimensions $n=n_0^2$. The discretization of the operator $\mathcal{L}(u)$ yields matrix extracted from the `Lyapack` package [13] using the command `fdm_2d_matrix` and denoted as $\mathrm{fdm}(n0,'f\_1(x,y)','f\_2(x,y)','f\_3(x,y)')$. To the author's knowledge, there is no exact solution available in the literature. We compared the results obtained from the new four ways and Matlab's ode23s solver to test if our approach gives reliable results. For this experiment, we consider $A=\mathrm{fdm}(n0,f_1(x,y),f_2(x,y),f_3(x,y))$ with $f_1(x,y)=e^{xy}$, $f_2(x,y)=\sin(xy)$ and $f_3(x,y)=y^2$, we used $s=2$. The time interval considered was $[0,1]$. The tolerance was set to 10^{-9} for the stop test on the residual and $d_{tol}=10^{-10}$ for the EBA-BDF, EGA-BDF, EGA-Rosenbrock and EBA-Rosenbrock methods, we used a constant timestep $h=0.005$. We chose a size of 64×64 for the matrix A. In Figure 2, we plotted the solutions using the new four ways and ode23s method in functions of the times. Computational time taken from an EBA-BDF method is 1.02 seconds, EGA-BDF method is 0.64 seconds, EGA-Rosenbrock method 1.21 seconds, EBA-Rosenbrock method is 3.54 seconds versus ode23s method is 496.24 seconds.

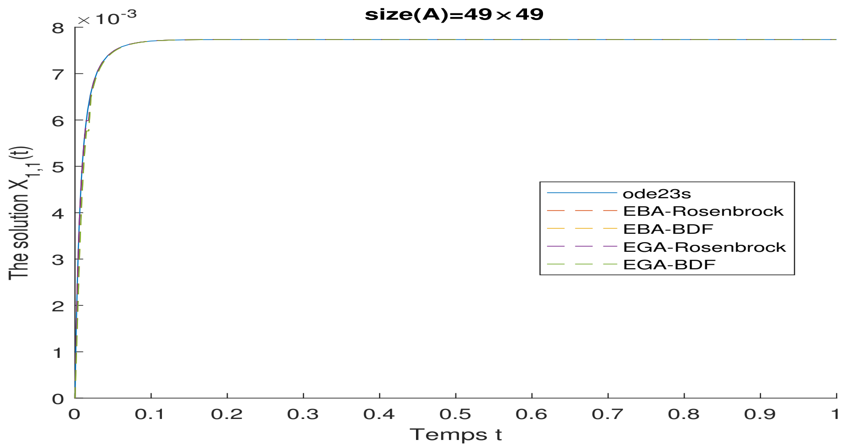

Figure 2: Values of $X_{1,1}(t)$ for $t\in[0,1]$ computed by using the new four ways and ode23s method

In Fig 3, we chose a size of 4096×4096 for the matrix A, we plotted the Frobenius norms of the residuals $\|R_m(T_f)\|_F$ at final time T_f versus the number of EBA and EGA iterations for the EBA-Rosenbrock, EGA-Rosenbrock, EBA-BDF and EGA-BDF methods, we used a constant timestep $h=0.1$.

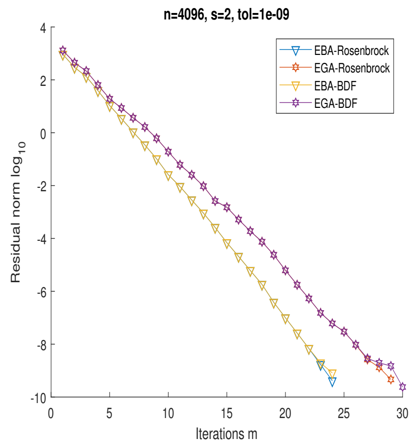

Figure 3: Residual norm versus number m of EBA and EGA iterations

In Table 1, we list the Frobenius residual norms at final time $T_f=1$ and the corresponding CPU time for each method. For this experiment, the algorithms are stopped when the residual norms are smaller than 10^{-9}.

Test Problem	Method	CPU time (seconds)	Iterations (m)	$\|R_m(T_f)\|_F$
4096×4096	EBA-Rosenbrock	4.91	24	4.11×10^{-10}
	EGA-Rosenbrock	7.90	29	4.64×10^{-10}
	EBA-BDF	4.60	24	8.19×10^{-10}
	EGA-BDF	**2.23**	30	2.39×10^{-10}
5776×5776	EBA-Rosenbrock	4.25	26	8.29×10^{-10}
	EGA-Rosenbrock	2.61	31	7.74×10^{-10}
	EBA-BDF	4.94	35	8.83×10^{-10}
	EGA-BDF	**2.03**	32	8.64×10^{-10}

Table 1: Runtimes in seconds and the residual norms for the new four ways

7.2 Example 2

For the second set of experiments, we used the matrices $add32$ and $thermal$ from the University of Florida Sparse Matrix Collection [6] and from the Harwell Boeing Collection

(http://math.nist.gov/MatrixMarket). The tolerance was set to 10^{-11} for the stop test on the residual. For the EBA-BDF and EBA-Rosenbrock methods, we used a constant timestep $h = 0.1$.

For this example, the matrix $A = add32, \quad n = 4960$. The time interval considered was $[0, 1]$. In Figure 4, we plotted the Frobenius residual norm $\|R_m(T_f)\|_F$ in functions of the numbers iterations.

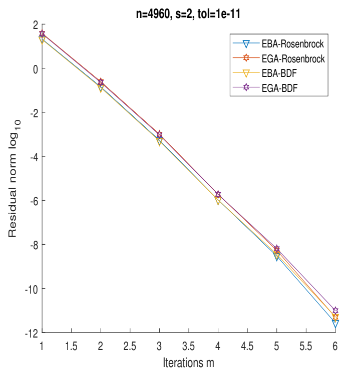

Figure 4: Frobenius residual norm in functions of the numbers iterations for example 2

For this example, the matrice $A = thermal, \quad n = 3456$. The time interval considered was $[0, 1]$. In Figure 5, we plotted the times, and the Frobenius residual norm $\|R_m(T_f)\|_F$ in functions of the numbers iterations.

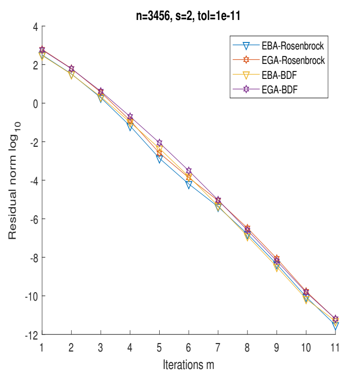

Figure 5: Frobenius residual norm in functions of the numbers iterations for example 2

In Table 2, we list the Frobenius residual norms at final time $T_f = 1$ and the corresponding CPU time for each method. For this experiment, the algorithms are stopped when the residual norms are smaller than 10^{-11}.

Test Problem	Method	CPU time (seconds)	Iterations (m)	$\|R_m(T_f)\|_F$
$A = thermal$	EBA-Rosenbrock	9.09	11	2.88×10^{-12}
3456×3456	EGA-Rosenbrock	**8.74**	11	6.16×10^{-12}
	EBA-BDF	8.87	11	1.95×10^{-12}
	EGA-BDF	9.37	11	2.23×10^{-12}
$A = add32$	EBA-Rosenbrock	25.11	6	2.69×10^{-12}
4960×4960	EGA-Rosenbrock	24.06	6	4.96×10^{-12}
	EBA-BDF	**23.16**	6	5.13×10^{-12}
	EGA-BDF	23.54	6	9.84×10^{-12}

Table 2: Runtimes in seconds and the residual norms for example 2

8 Conclusion

We presented in this paper a way to solve large-scale differential T-Lyapunov matrix equations. Our approach consists in projecting the initial problem on extended Krylov subspaces, through low-rank approximate solutions and an expanded Arnoldi algorithm to obtain a small-scale differential matrix equation that is solved using iterative methods (Rosenbrook method or BDF method). The process is stopped as soon as selected accuracy is achieved. We have given some theoretical results. In particular, we have demonstrated that the residual norm can be calculated without explicitly calculating the approximate solution. This result is very important as calculating an approximate solution can be a major memory challenge for large-scale cases. Moreover, this allowed us to design a stop test for Arnoldi iterations. Numerical experiments have shown that our approach is interesting for large-scale problems, providing accurate results, and that computing time is much lower. As there is no other way to deal with such large dimensions cases available in the literature, we have not been able to cope with our way to other approaches (except for small-scale cases that have led to a very satisfactory way), but the fact that we were able to calculate the remaining standard ensures that the method is accurate.

Acknowledgements

The authors are grateful to professor Madjid Eshaghi, editor-in-Chief, and an anonymous referee for their valuable comments which helped to improve the quality of this paper.

References

[1] H. Abou-Kandil, G. Frelling, V. Ionescu and G. Jank, *Matrix Riccati equations in control and systems theory*, Systems and Control Foundations and Applications, Birkhäuser Basel, 2003.

[2] A.H. Bentbib, K. Jbilou and El. M. Sadek, *On some extended block Krylov based methods for large scale nonsymmetric Stein matrix equations*, Math. **5** (2017), 21.

[3] R. Bouyouli, K. Jbilou, R. Sadaka and H. Sadok, *Convergence properties of some block Krylov subspace methods for multiple linear systems*, J. Comput. Appl. Math. **196** (2006), 498–511.

[4] J.C. Butcher, *Numerical methods for ordinary differential equations*, John Wiley and Sons, 2008.

[5] B.N. Datta,*Numerical methods for linear control systems design and analysis*, Elsevier Academic Press, 2003.

[6] T. Davis, *The University of Florida Sparse Matrix Collection*, NA Digest, **97** (1997), 23, Available online: http://www.cise.ufl.edu/research/sparse/matrices, (Accessed on June, 10, 2016).

[7] f. Dopico, J. Gonzàlez, D. Kressner and V. Simoncini, *Projection methods for large-scale T-Sylvester equations*, Math. Comput. 85 (2016), no. 301, 2427–2455.

[8] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl. **19** (1998), no. 3, 7551–7771.

[9] Hached M. Jbilou. K. *Numerical solutions to large-scale differential lyapunov matrix equations*, Numerical Algorithms, DOI: 10.1007/s11075-017-0458-y, (2018).

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

[11] K. Jbilou, *Low-rank approximate solution to large Sylvester matrix equations*, App. Math. Comput. **177** (2006), 365–376.

[12] M. Oozawa, T. Sogabe, Y. Miyatake and S.L. Zhang, *On a relationship between the T-congruence Sylvester equation and the Lyapunov equation*, J.Comput. Appl. Math. 329 (2018), 51–56.

[13] Penzl, T. *LYAPACK A MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems*, Available online: http://www.tuchemintz.de/sfb393/lyapack.

[14] H.H. Rosenbrock, *Some general implicit processes for the numerical solution of differential equations*, J. Comput. **5** (1963), 329–330.

[15] E. Sadek, A.H. Bentbib, L. Sadek and H.T. Alaoui, *Global extended Krylov subspace methods for large-scale differential Sylvester matrix equations*, J. Appl. Math. Comput. **62** (2020), 1571–7177.

[16] L. Sadek and H.T. Alaoui, . *The extended block Arnoldi method for solving generalized differential Sylvester equations*, J. Math. Model. **8** (2020), no. 2, 189–206.

[17] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput. **29** (2007), 12681–71288.