



The constant solution method for solving large-scale differential Sylvester matrix equations with time invariant coefficients

Abderrahman Bouhamidi¹ · Lakhdar Elbouyahyaoui² · Mohammed Heyouni¹

Received: 11 September 2022 / Accepted: 23 August 2023 / Published online: 15 September 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

This paper is mainly focused on the solution of Sylvester matrix differential equations with time-independent coefficients. We propose a new approach based on the construction of a particular constant solution which allows to construct an approximate solution of the differential equation from that of the corresponding algebraic equation. Moreover, when the matrix coefficients of the differential equation are large, we combine the constant solution approach with Krylov subspace methods for obtaining an approximate solution of the Sylvester algebraic equation, and thus form an approximate solution of the large-scale Sylvester matrix differential equation. We establish some theoretical results including error estimates and convergence as well as relations between the residuals of the differential and its corresponding algebraic Sylvester matrix equation. We also give explicit benchmark formulas for the solution of the differential equation. To illustrate the efficiency of the proposed approach, we perform numerous numerical tests and make various comparisons with other methods for solving Sylvester matrix differential equations.

Keywords Krylov subspace methods · Block Arnoldi · Matrix differential Sylvester equation · Dynamical systems · Control · Ordinary differential equations

Mathematics Subject Classification (2010) MSC 65F

✉ Mohammed Heyouni
mohammed.heyouni@univ-littoral.fr

Abderrahman Bouhamidi
abderrahman.bouhamidi@univ-littoral.fr

Lakhdar Elbouyahyaoui
lakhdarr2000@yahoo.fr

¹ L.M.P.A, Université du Littoral Côte d'Opale, 50 rue F. Buisson BP. 699, F-62228 Calais Cedex, France

² Centre Régional des Métiers de l'Éducation et de la Formation de l'Oriental, Oujda, Maroc

1 Introduction

Differential Lyapunov and Sylvester equations are involved in many areas of applied mathematics and arise in numerous scientific applications. For instance, they play a crucial role in control theory, model order reduction, image processing and the list is not exhaustive. In particular, the differential Lyapunov matrix equation is a useful tool for stability analysis and control design for linear time-dependent systems [2, 3]. In this paper, we are concerned with numerically solving the differential Sylvester matrix equation of the form

$$\begin{cases} \dot{X}(t) = A X(t) + X(t) B - C, & t \in [t_0, T] \\ X(t_0) = X_0, \end{cases} \quad (1)$$

where $[t_0, T] \subset \mathbb{R}$ is a closed and bounded time interval with t_0, T are the initial and final times respectively. We set $\Delta T = T - t_0$, the length of the interval $[t_0, T]$. The coefficient matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{s \times s}$, and $C \in \mathbb{R}^{n \times s}$ are constant real matrices, where the set $\mathbb{R}^{q \times k}$ is the space of real matrices of size $q \times k$. The differential Lyapunov matrix equation corresponds to the symmetric case where $B = A^T$. Before describing the new proposed method, we refer to the algebraic equation canonically associated to (1)

$$A X + X B = C, \quad (2)$$

as the corresponding (or associated) algebraic Sylvester equation. To the best of our knowledge, despite the importance of differential matrix equations, few works have been devoted to their numerical resolution when the matrix coefficients are large. Adaptation of BDF and/or Rosenbrock methods has been described in [7, 8] (see also the references in [5]). However these adaptations usually suffer from a problem of numerical data storage. To remedy this problem, combining Krylov subspaces techniques with BDF methods or with Taylor series expansions have recently been proposed [5, 20]. Other existing methods described in the recent literature for solving large-scale differential Sylvester matrix equation rely on using the integral formula or some numerical ODE solver [21, 38]. The strategy we pursue in this manuscript is different in the sense that our approach for solving differential Sylvester (or Lyapunov) matrix equations is based on the use of the constant solution to the differential equation. The first result we use indicates that the solution of the differential equation is written in terms of the solution of the corresponding algebraic equation. Additionally, in the case where the coefficient matrices A and/or B are large, we combine the new expression of the solution with some projection techniques on Krylov subspaces, such as the block Arnoldi algorithm for solving the corresponding algebraic equations or for approximating the exponential of a matrix.

The outline of this paper is as follows: In Sect. 2, some preliminaries and basic results are recalled and the notations needed in this paper are introduced. In Sect. 3, we introduce our proposed method which is called the constant solution method (CSM in short), describe some of its properties and give some theoretical results. A summarized

and brief description of the corresponding algorithm will also be given. In Sect. 4, we combine CSM with the block Arnoldi algorithm for solving algebraic Sylvester matrix equations in order to tackle large-scale differential Sylvester equations. The two cases: full and low-rank are discussed. Moreover, we establish theoretical results expressing the residual of the differential equation in terms of that of the algebraic equation. We also establish some theoretical results on the convergence and on the error estimates provided by the constant solution method. To have at hand an exact solution to which the approximate solution delivered by CSM, we show in Sect. 5 how to generate two benchmark differential Sylvester matrix equations with a known exact solution. In the last Section 6, which is devoted to numerical experiments, we describe several set of tests whose results indicate that CSM is an efficient and robust method. As usual, the last section is devoted to a brief conclusion.

2 Preliminaries and notations

In this section, we recall some known results and introduce the notations used in the rest of this paper. The identity matrix of $\mathbb{R}^{q \times q}$ is denoted by I_q . The Frobenius inner product is defined by

$$\langle Y, Z \rangle = \text{tr}(Y^T Z), \quad Y, Z \in \mathbb{R}^{l \times q},$$

where $\text{tr}(M)$ denotes the trace of a square matrix M . The associated Frobenius norm is denoted by $\|Y\| = \sqrt{\langle Y, Y \rangle}$. A basis $[W_1, W_2, \dots, W_m]$ of matrices is F-orthogonal with respect the Frobenius inner product if $\langle W_i, W_j \rangle = 0$, for $i \neq j$. For a bounded matrix-valued function G defined on the interval $[t_0, T]$, we consider the following uniform norm given by

$$\|G\|_\infty = \sup_{t \in [t_0, T]} \|G(t)\|.$$

We recall that a bounded function on the compact $[t_0, T]$ is continuous on a such interval. The Kronecker product of two matrices $J \in \mathbb{R}^{n_j \times m_j}$ and $K \in \mathbb{R}^{n_k \times m_k}$ is the matrix $J \otimes K = [J_{i,j} K]$ of size $n_j n_k \times m_j m_k$. The following well known properties are used throughout this paper:

1. $(A \otimes B)(C \otimes D) = (A C) \otimes (B D)$,
2. $(A \otimes B)^T = (A^T \otimes B^T)$,
3. $\text{vec}(A B C) = (C^T \otimes A) \text{vec}(B)$.

The *vec* operator consists in transforming a matrix into a vector by stacking its columns one by one to form a single column vector. We also recall that the Hadamard product of two matrices $J, K \in \mathbb{R}^{p \times q}$ is the matrix $J \odot K = [J_{i,j} K_{i,j}]$ of the same size $p \times q$.

The following proposition is given in [1] without any specific details of its proof. Although this proof does not present any major difficulty, it seemed interesting to us to give the details of this proof.

Proposition 1 *The unique solution of the differential Sylvester matrix equation (1) is given by the following integral formula*

$$X(t) = e^{(t-t_0)A} X_0 e^{(t-t_0)B} - \int_{t_0}^t e^{(t-u)A} C e^{(t-u)B} du. \quad (3)$$

Proof Let x_0, x, c be the vectors such that $x_0 = \text{vec}(X_0)$, $x = \text{vec}(X)$, $c = \text{vec}(C)$ where $X_0, X(t)$, and C are appearing in the system (1) which may be transformed to the following classical differential linear system

$$\begin{cases} \dot{x}(t) = \mathbb{A} x(t) - c, \\ x(t_0) = x_0, \end{cases} \quad (4)$$

where the matrix \mathbb{A} of size $ns \times ns$ is given by

$$\mathbb{A} = I_s \otimes A + B^T \otimes I_n.$$

It is well known that the previous system (4) has a unique solution $x(t)$ which is differentiable with a continuous derivative on $[t_0, T]$ (it is even of infinite class on $[t_0, T]$), and the unique solution of (4) is given by the following formula

$$x(t) = e^{(t-t_0)\mathbb{A}} x_0 - \int_{t_0}^t e^{(t-u)\mathbb{A}} c du. \quad (5)$$

Now, since, the matrices $I_s \otimes A$ and $B^T \otimes I_n$ commute, then using the properties of the Kronecker product and the additive commutativity of the matrix exponential ($e^{M+N} = e^M e^N \iff MN = NM$), it follows that

$$\begin{aligned} e^{(t-t_0)\mathbb{A}} &= e^{(t-t_0)(I_s \otimes A + B^T \otimes I_n)} = e^{(t-t_0)(I_s \otimes A)} e^{(t-t_0)(B^T \otimes I_n)} \\ &= (I_s \otimes e^{(t-t_0)A}) (e^{(t-t_0)B^T} \otimes I_n) = e^{(t-t_0)B^T} \otimes e^{(t-t_0)A}. \end{aligned}$$

Thus,

$$e^{(t-t_0)\mathbb{A}} x_0 = \left[e^{(t-t_0)B^T} \otimes e^{(t-t_0)A} \right] \text{vec}(X_0) = \text{vec}(e^{(t-t_0)A} X_0 e^{(t-t_0)B}).$$

Finally, this implies that the formula (5) giving the solution to the system (4) leads to the formula (3) giving the unique solution to the differential Sylvester system (1). \square

Remark 1 *Although, the systems (1) and (4) are mathematically equivalent, the difficulties one may encounter when solving these systems are different. For moderate size problems, it is possible to apply a numerical integration scheme directly to the system (4) or to use (5). However, in many practical situations, exploiting expression (4) for computing the solution $x(t)$ may be very expensive. Indeed, the matrix \mathbb{A} can be very large and difficult to handle on a computer. Another obstacle that can be encountered*

is in the evaluation of the exponential of matrices. With the form (1), some numerical techniques are available to approximate the matrix exponential, see [25, 33, 36].

3 The constant solution method for the differential Sylvester matrix equation

In the integral Formula (3), quadrature methods are needed to compute numerically the approximate solution. Thus, when one (or both) of the matrix coefficients A or B is (or are) large and has (or have) no particular exploitable structure, the computation of the integral may be expensive or even unfeasible. In this section, we use another expression for the solution of the system (1) which is given in terms of the solution of the corresponding algebraic Sylvester matrix equation (2). This expression avoids the use of quadrature methods since it does not contain an integral. To the best of our knowledge, the approach we describe in this section has never been exploited in the context of solving large-scale differential Sylvester matrix equations. However, it is based on the classical and simple technique of adding a particular constant solution to the general solution of the homogeneous differential equation to form the general solution of a linear differential equation of order one with constant coefficients. Next, we give the following theorem, which gives a useful and interesting expression of the unique solution of the system (1). The result of this theorem is known in the literature [6, 16], but, in practice, it has not been exploited numerically to give approximate solutions. This theorem is not difficult to establish. However, in order to facilitate the reading of the present work, it seems interesting to us to give the proof of this theorem.

Theorem 1 *Suppose that the matrices A and B in the system (1) are such that $\sigma(A) \cap \sigma(-B) = \emptyset$, where $\sigma(M)$ denotes the spectrum of the matrix M , then the unique and exact solution $X^*(t)$ of the system (1) is given by*

$$X^*(t) = e^{(t-t_0)A} (X_0 - \tilde{X}^*) e^{(t-t_0)B} + \tilde{X}^*, \tag{6}$$

where \tilde{X}^* is the unique and exact solution of the algebraic Sylvester equation (2).

Proof The general solution of the homogeneous differential equation associated to (1) is given by

$$Z(t) = e^{tA} Y e^{tB},$$

where $Y \in \mathbb{R}^{n \times s}$ is some constant matrix. Since $\sigma(A) \cap \sigma(-B) = \emptyset$, the algebraic Sylvester equation (2) has a unique solution \tilde{X}^* (see, e.g., [26, Thm. 2.4.4.1]). Thus, the unique solution \tilde{X}^* seen as a constant matrix function (i.e., $\tilde{X}^*(t) = \tilde{X}^*$), may be considered as a particular solution of the differential equation

$$\dot{X}(t) = A X(t) + X(t) B - C.$$

It follows that the general solution of the previous differential equation is given by

$$X(t) = e^{tA} Y e^{tB} + \tilde{X}^*.$$

Finally, since the unique solution of the differential system (1) must satisfy the initial condition $X^*(t_0) = X_0$, it follows that $X^*(t_0) = e^{t_0 A} Y e^{t_0 B} + \tilde{X}^* = X_0$. The last equality implies $Y = e^{-t_0 A} (X_0 - \tilde{X}^*) e^{-t_0 B}$, and expression (6) follows immediately. \square

Remark 2 *As the constant solution method is based on the existence of a solution to its corresponding algebraic matrix equation, our proposed method may not be feasible if the condition $\sigma(A) \cap \sigma(-B) = \emptyset$ is not fulfilled.*

In the remainder of this paper, we assume that the matrices A and B in (1) satisfy the condition

$$\sigma(A) \cap \sigma(-B) = \emptyset.$$

The following property shows the behavior of the matrix solution $X^*(t)$ as the interval $[t_0, T]$ becomes very more and more large, namely, as the final time T goes to $+\infty$.

Proposition 2 [30, Chapter 8] *Suppose that the coefficients A and B in the system (1) are stable matrices, then the unique solution $X^*(t)$ of the differential system (1) satisfies*

$$\lim_{T \rightarrow +\infty} \|X^*(T) - \tilde{X}^*\| = 0,$$

where \tilde{X}^* is the unique solution of the corresponding algebraic Sylvester equation (2).

In the remainder of this section, we suppose that the matrix coefficients A and B are of moderate size. In this case, an approximate solution to the algebraic Sylvester equation (2) may be obtained by a direct solver such as the Bartels-Stewart algorithm, the Schur decomposition, or the Hammarling method [4, 19, 22, 31, 41]. A common point to all these methods is first the computation of the real Schur forms of the coefficient matrices using the QR algorithm. Then, the original equation is transformed into an equivalent form that is easier to solve by a forward substitution. Now, suppose that \tilde{X}_a is an approximate solution to the exact solution \tilde{X}^* of the Sylvester algebraic equation (2), it follows that an approximate solution $X_a(t)$ to the exact solution $X^*(t)$ of the Sylvester differential equation (1) can be expressed in the following form.

$$X_a(t) = e^{(t-t_0)A} (X_0 - \tilde{X}_a) e^{(t-t_0)B} + \tilde{X}_a. \quad (7)$$

Here, as A and B are assumed to be of moderate size, we also assume that both exponential $e^{(t-t_0)A}$ and $e^{(t-t_0)B}$ are computed exactly. To establish an upper bound for the error norm, let us introduce the algebraic error \tilde{E} and the differential error $E(t)$ given by

$$\tilde{E} = \tilde{X}^* - \tilde{X}_a, \quad \text{and} \quad E(t) = X^*(t) - X_a(t), \quad \forall t \in [t_0, T],$$

respectively. Finally, recalling that $\Delta_T = T - t_0$ and $\|E\|_\infty = \sup_{t \in [t_0, T]} \|E(t)\|$, we have the following result

Proposition 3 *In the case where the matrix exponential is computed exactly, we have*

$$\|E(t)\| \leq \left(1 + e^{(t-t_0)(\|A\|+\|B\|)}\right) \|\tilde{E}\|, \quad \forall t \in [t_0, T].$$

It follows that

$$\|E\|_\infty \leq \left(1 + e^{\Delta_T(\|A\|+\|B\|)}\right) \|\tilde{E}\|,$$

where E and \tilde{E} are the errors associated to the approximate solutions $X_a(t)$ and \tilde{X}_a respectively.

Proof Subtracting (7) from (6), we get

$$E(t) = X^*(t) - X_a(t) = -e^{(t-t_0)A} \tilde{E} e^{(t-t_0)B} + \tilde{E}, \quad \forall t \in [t_0, T],$$

and from the triangular inequality, we obtain

$$\|E(t)\| \leq \|e^{(t-t_0)A} \tilde{E} e^{(t-t_0)B}\| + \|\tilde{E}\|, \quad \forall t \in [t_0, T].$$

The Frobenius norm being multiplicative (that is $\|A B\| \leq \|A\| \|B\|$), this implies that $\|e^{sM}\| \leq e^{s\|M\|}$ for all $s \geq 0$ and for any square matrix M . Thus,

$$\|E(t)\| \leq \left(1 + e^{(t-t_0)(\|A\|+\|B\|)}\right) \|\tilde{E}\|, \quad \forall t \in [t_0, T].$$

As E is a continuous matrix function on the interval $[t_0, T]$, $(t - t_0) \leq \Delta_T$ and $\|E\|_\infty = \sup_{t \in [t_0, T]} \|E(t)\|$, then the desired result follows obviously. \square

Let us now introduce $R(t)$ and \tilde{R} the residuals associated to the differential and algebraic Sylvester matrix equations, respectively. These residuals are defined by

$$\begin{cases} R(t) = \dot{X}_a(t) - (A X_a(t) + X_a(t) B - C), & t \in [t_0, T], \\ \tilde{R} = C - (A \tilde{X}_a + \tilde{X}_a B), \end{cases} \tag{8}$$

and satisfy the following proposition.

Proposition 4 *In the case where the matrix exponential is computed exactly, the residual for the differential equation, is time-independent and we have*

$$R(t) = \tilde{R}, \quad \forall t \in [t_0, T].$$

Proof From (7), we have

$$\dot{X}_a(t) = e^{(t-t_0)A} (A(X_0 - \tilde{X}_a) + (X_0 - \tilde{X}_a)B) e^{(t-t_0)B}.$$

On the other hand, we have

$$A X_a(t) + X_a(t) B = e^{(t-t_0)A} \left(A (X_0 - \tilde{X}_a) + (X_0 - \tilde{X}_a) B \right) e^{(t-t_0)B} + A \tilde{X}_a + \tilde{X}_a B.$$

Then subtracting one of the two previous relations from the other, we get

$$R(t) = \dot{X}_a(t) - (A X_a(t) + X_a(t) B - C) = C - (A \tilde{X}_a + \tilde{X}_a B) = \tilde{R}.$$

□

Before ending this section, we sketch in Algorithm 1 below the main steps that must be followed to obtain approximations $X_k = X_a(t_k)$ to the solution of the differential Sylvester equation (1) at different nodes t_k (for $k = 1, \dots, N$) of a suitable discretization of the time interval $[t_0, T]$.

Algorithm 1 Constant solution method in the case of moderate size (CSM).

- 1: **Input:** The matrices A, B, C , the initial and final times t_0, T , the number N of nodes and the step time δ_T .
 - 2: **Output:** X_1, \dots, X_N , (where $X_k = X_a(t_k)$), ($1 \leq k \leq N$)
 - 3: Solve the algebraic Sylvester equation: $A X + X B = C$, to get an approximate solution \tilde{X}_a to the exact solution \tilde{X}^* .
 - 4: **for** $k = 1, \dots, N$ **do**
 - 5: Compute: $t_k = t_{k-1} + \delta_T$;
 - 6: Compute: $X_k = e^{(t_k-t_0)A} (X_0 - \tilde{X}_a) e^{(t_k-t_0)B} + \tilde{X}_a$;
 - 7: **end for**
-

4 Block Arnoldi for solving large-scale differential Sylvester matrix equations

It is well known that computing the matrix exponential may be expensive when the matrix is very large. Thus, expression (6) may not be directly exploitable in the case of large-scale matrix coefficients. In the following, we will see how to circumvent this difficulty using projection methods onto some Krylov subspace. Indeed, in addition to allowing us to obtain a good approximation of the exact solution of the algebraic Sylvester equation (2), Krylov subspace methods are also a useful tool to compute the action of matrix exponential on a block vector with a satisfactory accuracy. During the last three decades, various projection methods on block, global or extended Krylov subspaces have been proposed to solve Sylvester matrix equations (or other similar equations) whose coefficients are large and sparse matrices [10–12, 17, 23, 24, 27–29]. The common idea behind these methods is to first reduce the size of the original equation by constructing a suitable Krylov basis, then solve the obtained low-dimensional equation by means of a direct method such as the Hessenberg-Schur method or the Bartels-Stewart algorithm [4, 19], and finally recover the solution of the original large equation from the smaller one. For a complete overview of the main methods for

solving algebraic Sylvester or Lyapunov equations, we refer to [3, 14, 40] and the references therein. In order to be as general as possible and not to impose restrictive assumptions, we opt for a resolution of the Sylvester (or Lyapunov) equation using the block Arnoldi process rather than the extended block Arnoldi process since the latter requires that the coefficient matrices A and B are non singular. This last condition may not be fulfilled in many practical cases.

We recall that projection techniques on block Krylov subspaces for solving matrix differential equations were first proposed in [20, 21] by exploiting the integral formula (3) and approximating the exponential of a matrix times a block of vectors or by solving a projected low-dimensional differential Sylvester matrix equation by means of numerical integration methods such as the backward differentiation formula (BDF) [13].

As said before, the approach we follow in this work is different from the one proposed in [20, 21]. It consists of exploiting formula (6), instead of the integral formula (3), which is less expensive. To have at hand an adequate basis of the considered Krylov subspace, we will use the block Arnoldi process described in the next subsection.

4.1 The block Arnoldi process

Let M be an $l \times l$ matrix and V an $l \times s$ block vector. We consider the classical block Krylov subspace

$$\mathbb{K}_m(M, V) = \text{Range}([V, M V, \dots, M^{m-1} V]) = \left\{ \sum_{k=0}^{m-1} M^k V \Omega_k, \Omega_k \in \mathbb{R}^{s \times s}, 0 \leq k \leq m-1 \right\}.$$

The block Arnoldi process, described in Algorithm 2, generates an orthonormal basis \mathbb{V}_m^M of the block Krylov subspace $\mathbb{K}_m(M, V)$.

Algorithm 2 The block Arnoldi process (BA) applied to the pair (M, V) .

- 1: **Input:** M a matrix of size $l \times l$, V a matrix of size $l \times s$ and m an integer.
 - 2: **Output:** \mathbb{V}_{m+1}^M and \mathbb{H}_m^M satisfying (9)–(11).
 - 3: Get V_1^M by computing the QR decomposition of V , i.e., $V = V_1^M A_1$;
 - 4: **for** $j = 1, \dots, m$ **do**
 - 5: Compute $U = M V_j^M$;
 - 6: **for** $i = 1, 2, \dots, j$ **do**
 - 7: $H_{i,j} = (V_i^M)^T U$;
 - 8: $U = U - V_i^M H_{i,j}$;
 - 9: **end for**
 - 10: Get V_{j+1}^M and $H_{j+1,j}$ by computing the QR decomposition of U ,
 - 11: i.e., $U = V_{j+1}^M H_{j+1,j}$.
 - 12: Set $H_{i,j} = 0$ for $i > j + 1$
 - 13: Define $\mathbb{V}_{j+1}^M = [V_1^M, \dots, V_j^M, V_{j+1}^M]$ and $\mathbb{H}_j^M = (H_{k,\ell})_{1 \leq k \leq j+1, 1 \leq \ell \leq j}$
 - 14: **end for**
-

Suppose that the upper triangular matrices $H_{j+1,j}$ are full rank then, since the above algorithm involves a Gram-Schmidt procedure, the obtained block vectors $V_1^M, V_2^M, \dots, V_m^M$ ($V_i^M \in \mathbb{R}^{l \times s}$) have their columns mutually orthogonal. Hence, after m steps, Algorithm 2 generates an orthonormal basis $\mathbb{V}_m^M = [V_1^M, V_2^M, \dots, V_m^M]$ of the block Krylov subspace $\mathbb{K}_m(M, V)$ and a block upper Hessenberg matrix \mathbb{H}_m^M whose non zeros blocks are the $H_{i,j} \in \mathbb{R}^{s \times s}$. We have the following and useful algebraic relations [18, 37].

$$M \mathbb{V}_m^M = \mathbb{V}_{m+1}^M \overline{\mathbb{H}}_m^M = \mathbb{V}_m^M \mathbb{H}_m^M + V_{m+1}^M H_{m+1,m}^M (\mathbb{E}_m^{(s)})^T, \tag{9}$$

$$\left(\mathbb{V}_m^M\right)^T M \mathbb{V}_m^M = \mathbb{H}_m^M, \tag{10}$$

$$\left(\mathbb{V}_m^M\right)^T \mathbb{V}_m^M = I_{ms}, \tag{11}$$

where $\overline{\mathbb{H}}_m^M = (\mathbb{V}_{m+1}^M)^T M \mathbb{V}_m^M \in \mathbb{R}^{(m+1)s \times ms}$, $H_{i,j} \in \mathbb{R}^{s \times s}$ is the (i, j) block of \mathbb{H}_m^M and $\mathbb{E}_m^{(s)}$ is the matrix of the last s columns of the $ms \times ms$ identity matrix I_{ms} , i.e., $\mathbb{E}_m^{(s)} = [0_{s \times (m-1)s}, I_s]^T$. In the following, we will use the notation

$$V_{m,s}^M = V_{m+1}^M H_{m+1,m}^M (\mathbb{E}_m^{(s)})^T. \tag{12}$$

4.2 Full-rank case

Here, we suppose that A is a large matrix while B is relatively smaller, i.e., $s \ll n$. We also assume that the derivative $\dot{X}(t_0)$ of X at t_0 is full rank, i.e., $\text{rank}(\dot{X}(t_0)) = s$, where $X(t)$ denotes the exact solution of (1).

Now, as $\dot{X}(t_0) = -C_0 = AX_0 + X_0B - C$, then $C_0 = C - (AX_0 + X_0B)$. It follows that the matrix function given by $Y(t) = X(t) - X_0$ is the unique solution of the following system

$$\begin{cases} \dot{Y}(t) = AY(t) + Y(t)B - C_0, & t \in [t_0, T] \\ Y(t_0) = 0. \end{cases}$$

Consequently, we may first solve the previous differential equation to get $Y(t)$ and then deduce the solution $X(t) = Y(t) + X_0$ of the differential equation (1). Thus, in the rest of this section, we will consider that $X_0 = 0$ as an initial condition in (1)

To obtain approximate solutions to the algebraic Sylvester equation (2), one can use the block Arnoldi method in which we consider approximate solutions that have the following form

$$\tilde{X}_m = \mathbb{V}_m^A \tilde{Y}_m, \tag{13}$$

where \mathbb{V}_m^A is the orthonormal Krylov basis generated by applying m iterations of Algorithm 2 to the pair (A, C) . Let \tilde{R}_m be the algebraic residual given by

$$\tilde{R}_m = C - (A \tilde{X}_m + \tilde{X}_m B). \tag{14}$$

The correction \tilde{Y}_m , is obtained by imposing the Petrov-Galerkin condition

$$(\mathbb{V}_m^A)^T \tilde{R}_m = 0_{m \times s \times s}.$$

Thus, taking into account the relations (9)–(11) and (13), it follows that \tilde{Y}_m is the solution of the reduced Sylvester equation

$$\mathbb{H}_m^A Y + Y B = C_m,$$

where $\mathbb{H}_m^A = (\mathbb{V}_m^A)^T A \mathbb{V}_m^A$ and $C_m = (\mathbb{V}_m^A)^T C$. Note that from Algorithm 2, we also get that $C = \mathbb{V}_m^A C_m$. Now, if $\sigma(\mathbb{H}_m^A) \cap \sigma(-B) = \emptyset$, then the previous Sylvester equation admits a unique solution which can be obtained by a direct method [4, 19]. In addition, from the relations (9)–(11), the residual \tilde{R}_m satisfies the following relation

$$\tilde{R}_m = -V_{m,s}^A \tilde{Y}_m. \tag{15}$$

According to [33, 35], the following approximation to $e^{(t-t_0)A} \tilde{X}^*$ holds

$$e^{(t-t_0)A} \tilde{X}^* \simeq \mathbb{V}_m^A e^{(t-t_0)\mathbb{H}_m^A} (\mathbb{V}_m^A)^T \tilde{X}_m.$$

It follows, that an approximate solution $X_m(t)$, for $t \in [t_0, T]$, to the exact solution $X^*(t)$ of the differential Sylvester matrix equation (1) may be obtained by

$$X_m(t) = -\mathbb{V}_m^A e^{(t-t_0)\mathbb{H}_m^A} (\mathbb{V}_m^A)^T \tilde{X}_m e^{(t-t_0)B} + \tilde{X}_m.$$

Taking into account (13), it follows that

$$X_m(t) = \mathbb{V}_m^A Y_m(t), \quad t \in [t_0, T], \tag{16}$$

where

$$Y_m(t) = -e^{(t-t_0)\mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0)B} + \tilde{Y}_m, \quad t \in [t_0, T]. \tag{17}$$

This matrix function satisfies the following result.

Proposition 5 *The matrix function $Y_m(t)$ given by (17) is the unique solution of the reduced differential Sylvester matrix equation*

$$\begin{cases} \dot{Y}(t) = \mathbb{H}_m^A Y(t) + Y(t) B - C_m, & t \in [t_0, T] \\ Y(t_0) = 0. \end{cases} \tag{18}$$

Proof The derivative of the matrix function $Y_m(t)$ as given by (17) is

$$\dot{Y}_m(t) = -e^{(t-t_0)\mathbb{H}_m^A} \left(\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m B \right) e^{(t-t_0)B}, \quad t \in [t_0, T].$$

On the other hand, we have

$$\begin{aligned} \mathbb{H}_m^A Y_m(t) + Y_m(t) B - C_m &= -e^{(t-t_0)\mathbb{H}_m^A} \left(\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m B \right) e^{(t-t_0)B} \\ &\quad + \mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m B - C_m, \quad t \in [t_0, T]. \end{aligned}$$

Thus, it follows that

$$\dot{Y}_m(t) - \left(\mathbb{H}_m^A Y_m(t) + Y_m(t) B - C_m \right) = 0,$$

and additionally, $Y_m(t)$ satisfies the initial condition $Y_m(t_0) = 0$. □

Remark 3 Proposition 5 shows that another way to obtain an approximation of $Y_m(t)$ can be the resolution of the projected and reduced differential equation (17) by using an adequate numerical ODE solver such as Runge–Kutta or BDF solvers. We recall that such technique was used in [20, 21]. In our proposed method, we do not use such approach, but instead, we solve the reduced corresponding algebraic equation by taking into account the approximations (16) together with the relation (17).

Now, let $R_m(t)$ be the residual associated to the approximate solution $X_m(t)$, i.e.,

$$R_m(t) = \dot{X}_m(t) - (A X_m(t) + X_m(t) B - C), \quad t \in [t_0, T]. \tag{19}$$

The following proposition gives an expression for this residual.

Proposition 6 The residual for the differential equation is given by

$$R_m(t) = -V_{m,s}^A Y_m(t) \tag{20}$$

$$= V_{m,s}^A e^{(t-t_0)\mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0)B} + \tilde{R}_m, \tag{21}$$

where \tilde{R}_m is the algebraic residual given in (14). Moreover

$$(\mathbb{V}_m^A)^T R_m(t) = 0_{m \times s}, \quad \forall t \in [t_0, sT]. \tag{22}$$

Proof Replacing, in (19), $X_m(t)$ by its expression given by (16), we get

$$R_m(t) = \mathbb{V}_m^A \dot{Y}_m(t) - A \mathbb{V}_m^A Y_m(t) - \mathbb{V}_m^A Y_m(t) B + C.$$

Then, using (9), we obtain

$$\begin{aligned} R_m(t) &= \mathbb{V}_m^A \dot{Y}_m(t) - \left(\mathbb{V}_m^A \mathbb{H}_m^A + V_{m,s}^A \right) Y_m(t) - \mathbb{V}_m^A Y_m(t) B + \mathbb{V}_m^A C_m. \\ &= \mathbb{V}_m^A \left(\dot{Y}_m(t) - \left[\mathbb{H}_m^A Y_m(t) + Y_m(t) B - C_m \right] \right) - V_{m,s}^A Y_m(t). \end{aligned}$$

As $Y_m(t)$ is the solution of (18), we then get (20). Now, according to (17), we obtain

$$R_m(t) = V_{m,s}^A e^{(t-t_0) \mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0) B} - V_{m,s}^A \tilde{Y}_m.$$

Finally, from (15), we get (21) and then the relation (22) follows immediately. \square

Note that if \mathbb{H}_m^A and B are stable, i.e., all the eigenvalues of \mathbb{H}_m^A and B belong to the half part of \mathbb{C} whose real part is negative. It follows that,

$$\lim_{T \rightarrow +\infty} e^{(T-t_0) \mathbb{H}_m^A} = 0_{m \times m} \text{ and } \lim_{T \rightarrow +\infty} e^{(T-t_0) B} = 0_{s \times s}.$$

Then, using (21), we get

$$\lim_{T \rightarrow +\infty} R_m(T) = \tilde{R}_m.$$

Now, let In addition and as done in the previous section, we consider the differential error E_m given by

$$E_m(t) = X^*(t) - X_m(t), \quad \forall t \in [t_0, T].$$

We recall that $X_m(t)$ and $X^*(t)$ are the approximate and exact solutions to the differential equation, respectively. The following result gives an error estimate for the error norm E_m .

Theorem 2 *Suppose that m steps of the block Arnoldi process were run and let $Z_m(t) := e^{(t-t_0) \mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0) B}$, then the following error estimate holds*

$$\|E_m(t)\| \leq \left(\frac{e^{(t-t_0)(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) (r_m^A + \|z_m^A(t)\|), \quad \forall t \in [t_0, T].$$

It follows that

$$\|E_m\|_\infty \leq \left(\frac{e^{\Delta T(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) (r_m^A + \|z_m^A\|_\infty),$$

where $r_m^A = \|\tilde{R}_m\| = \|H_{m+1,m}^A \bar{Y}_m\|$ and $z_m^A(t) = H_{m+1,m}^A \bar{Z}_m(t)$, with $\bar{Y}_m, \bar{Z}_m(t)$ are the matrices of size $s \times s$ formed by the s last rows of \tilde{Y}_m and $Z_m(t)$, respectively.

Proof From (19) and the differential Sylvester matrix equation (1), we have

$$\dot{E}_m(t) = \dot{X}^*(t) - \dot{X}_m(t) = A (X^*(t) - X_m(t)) + (X^*(t) - X_m(t)) B - R_m(t),$$

with $E_m(t_0) = 0$. Thus, the function $E_m(t)$ satisfies the following differential Sylvester matrix equation

$$\begin{cases} \dot{E}_m(t) = A E_m(t) + E_m(t) B - R_m(t), & t \in [t_0, T] \\ E_m(t_0) = 0. \end{cases}$$

So, $E_m(t)$ may be written by the following integral formula

$$E_m(t) = - \int_{t_0}^t e^{(t-s)A} R_m(t) e^{(t-s)B} ds.$$

Passing to the norm, for all $t \in [t_0, T]$, we get

$$\|E_m(t)\| \leq \int_{t_0}^t \|e^{(t-s)A} R_m(t) e^{(t-s)B}\| ds \leq \|R_m(t)\| \int_{t_0}^t \|e^{(t-s)A}\| \|e^{(t-s)B}\| ds.$$

As, $\|e^{\alpha M}\| \leq e^{\alpha\|M\|}$ for $\alpha \geq 0$ and $M = A$ or $M = B$, we obtain that, for all $t \in [t_0, T]$,

$$\|E_m(t)\| \leq \|R_m(t)\| \int_{t_0}^t e^{(t-s)(\|A\|+\|B\|)} ds.$$

This gives after integration that, for all $t \in [t_0, T]$ we have

$$\|E_m(t)\| \leq \left(\frac{e^{(t-t_0)(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) \|R_m(t)\|.$$

Then using (21) and the triangular inequality, we get

$$\|E_m(t)\| \leq \left(\frac{e^{(t-t_0)(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) (r_m^A + \|z_m^A(t)\|), \text{ for all } t \in [t_0, T].$$

Finally, the desired result is obtained by passing to the uniform norm. \square

Now, we point out that (20) provides a cheap formula for computing at each node t_k the norm $r_{m,k} := \|R_m(t_k)\|$ of the residual associated to the approximate solution $X_{m,k} := X_m(t_k)$. This formula avoids computing matrix vector products with the large coefficient matrix A since we have

$$r_{m,k} := \|R_m(t_k)\| = \|H_{m+1,m}^A (\mathbb{E}_m^{(s)})^T Y_{m,k}\| = \|H_{m+1,m}^A \bar{Y}_{m,k}\|, \quad (23)$$

where $\bar{Y}_{m,k} = (\mathbb{E}_m^{(s)})^T Y_{m,k}$ is the matrix of size $s \times s$ formed by the last s rows of the matrix $Y_{m,k} := Y_m(t_k)$.

Finally, we end this section by summarizing in Algorithm 3 our proposed method that is the block Arnoldi combined with the constant solution method (BA-CSM) applied for full-rank differential Sylvester equations

4.3 Low-rank case

Now, we consider the case where both A and B are large matrices. Here, we assume that the derivative $\dot{X}(t_0)$ of X at t_0 is low-rank and given under the factored form

Algorithm 3 Block Arnoldi Constant Solution Method (BA-CSM) (Full-rank case).

1: **Input:** The matrices A, B, C , the initial and final times t_0, T , a tolerance $\text{tol} > 0$, a maximum number of iterations M_{max} , a step-size parameter p and N the number of nodes in the time discretization.
 2: **Output** $X_{m,1}, \dots, X_{m,N}$, where $X_{m,k} = X_m(t_k)$, ($1 \leq k \leq N$)
 3: Compute $\delta T = (T - t_0)/N$.
 4: **for** $m = 1, \dots, M_{max}$ **do**
 5: Compute V_m^A to update the orthonormal basis $\mathbb{V}_m^A = [V_1^A, \dots, V_m^A]$ and get the m -th block of \mathbb{H}_m^A by applying Algorithm 2 to (A, C) ;
 6: **if** m is a multiple of p **then**
 7: Compute: $C_m = (\mathbb{V}_m^A)^T C$.
 8: Solve the reduced Sylvester equation: $\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m B = C_m$.
 9: **for** $k = 1, \dots, N$ **do**
 10: Compute $t_k = t_{k-1} + \delta T$.
 11: Compute $Y_{m,k} := Y_m(t_k) = -e^{(t_k-t_0)} \mathbb{H}_m^A \tilde{Y}_m e^{(t_k-t_0) B} + \tilde{Y}_m$.
 12: Compute $r_{m,k} = \|H_{m+1,m}^A \bar{Y}_{m,k}\|$.
 13: **end for**
 14: Compute $r_{max} = \max\{r_{m,1}, \dots, r_{m,N}\}$
 15: **if** $r_{max} < \text{tol}$ **then**
 16: go to line 20;
 17: **end if**
 18: **end if**
 19: **end for**
 20: **for** $k = 1, \dots, N$ **do**
 21: The approximate solution $X_{m,k}$ at time t_k is $X_{m,k} = \mathbb{V}_m^A Y_{m,k}$.
 22: **end for**

$\dot{X}(t_0) = -E F^T$ where $E \in \mathbb{R}^{n \times r}$ and $F \in \mathbb{R}^{s \times r}$ and $X(t)$ is the exact solution of (1). Then, $\dot{X}(t_0) = -E F^T = A X_0 + X_0 B - C$, thus $E F^T = C - (A X_0 + X_0 B)$. As in the full-rank case, it follows that the matrix function given by $Y(t) = X(t) - X_0$ is the unique solution of the following system

$$\begin{cases} \dot{Y}(t) = A Y(t) + Y(t) B - E F^T, & t \in [t_0, T] \\ Y(t_0) = 0. \end{cases}$$

Accordingly, we can first solve the previous differential equation to get $Y(t)$ and subsequently deduce the solution $X(t) = Y(t) + X_0$ of the differential equation (1). Thus, in the rest of this section, we took $X_0 = 0$ as an initial condition and assume that C is factored in the form $C = E F^T$.

To obtain approximate solutions to the low-rank algebraic Sylvester equation (2), we can use the block Arnoldi method in which we consider approximate solutions that have the form

$$\tilde{X}_m = \mathbb{V}_m^A \tilde{Y}_m (\mathbb{V}_m^B)^T, \tag{24}$$

where $\mathbb{V}_m^A, \mathbb{V}_m^B$ are the orthonormal matrices obtained by running m iterations of Algorithm 2 applied to the pairs (A, E) and (B^T, F) respectively.

Enforcing the following Petrov-Galerkin condition

$$(\mathbb{V}_m^A)^T \tilde{R}_m \mathbb{V}_m^B = 0_{m \times m},$$

to the algebraic residual \tilde{R}_m given by

$$\tilde{R}_m = E F^T - (A \tilde{X}_m + \tilde{X}_m B). \tag{25}$$

Multiplying (25) on the left by $(\mathbb{V}_m^A)^T$ and on the right by \mathbb{V}_m^B and taking into account relations (9)–(11) and (24), it follows immediately, that \tilde{Y}_m is the solution of the reduced projected Sylvester equation

$$\mathbb{H}_m^A Y + Y (\mathbb{H}_m^B)^T = E_m F_m^T, \tag{26}$$

where $\mathbb{H}_m^A = (\mathbb{V}_m^A)^T A \mathbb{V}_m^A$, $\mathbb{H}_m^B = (\mathbb{V}_m^B)^T B^T \mathbb{V}_m^B$ are the $mr \times mr$ upper block Hessenberg matrices generated by the block Arnoldi process and $E_m = (\mathbb{V}_m^A)^T E$, $F_m = (\mathbb{V}_m^B)^T F$. Note that from Algorithm 2, we also get that $E = \mathbb{V}_m^A E_m$ and $F = \mathbb{V}_m^B F_m$. Here also, if $\sigma(\mathbb{H}_m^A) \cap \sigma(-\mathbb{H}_m^B) = \emptyset$, then equation (26) admits a unique solution which can be computed using a standard direct method such as those described in [4, 19]. Using the relation (9)–(10) and from the relations (24)–(26), we get

$$\tilde{R}_m = -V_{m,r}^A \tilde{Y}_m (\mathbb{V}_m^B)^T - \mathbb{V}_m^A \tilde{Y}_m (V_{m,r}^B)^T, \tag{27}$$

where $V_{m,r}^A = V_{m+1}^A H_{m+1,m}^A (\mathbb{E}_m^{(r)})^T$ and $V_{m,r}^B = V_{m+1}^B H_{m+1,m}^B (\mathbb{E}_m^{(r)})^T$. We also notice that, according to [33, 35], an approximation to $e^{(t-t_0)A} \tilde{X}^* e^{(t-t_0)B}$ may be obtained as

$$e^{(t-t_0)A} \tilde{X}^* e^{(t-t_0)B} \simeq \mathbb{V}_m^A e^{(t-t_0)\mathbb{H}_m^A} (\mathbb{V}_m^A)^T \tilde{X}_m \mathbb{V}_m^B e^{(t-t_0)(\mathbb{H}_m^B)^T} (\mathbb{V}_m^B)^T.$$

Then, it follows, that an approximate solution $X_m(t)$ to the exact solution $X^*(t)$ of the differential Sylvester matrix equation (1) may be given by

$$X_m(t) = -\mathbb{V}_m^A e^{(t-t_0)\mathbb{H}_m^A} (\mathbb{V}_m^A)^T \tilde{X}_m \mathbb{V}_m^B e^{(t-t_0)(\mathbb{H}_m^B)^T} (\mathbb{V}_m^B)^T + \tilde{X}_m.$$

Taking into account (24) gives that

$$X_m(t) = \mathbb{V}_m^A Y_m(t) (\mathbb{V}_m^B)^T, \quad t \in [t_0, T], \tag{28}$$

where

$$Y_m(t) = -e^{(t-t_0)\mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0)(\mathbb{H}_m^B)^T} + \tilde{Y}_m, \quad t \in [t_0, T]. \tag{29}$$

As in the full-rank case, we have the following proposition.

Proposition 7 *The matrix function $Y_m(t)$ given by (29) is the unique solution of the reduced differential Sylvester matrix equation*

$$\begin{cases} \dot{Y}(t) = \mathbb{H}_m^A Y(t) + Y(t) (\mathbb{H}_m^B)^T - E_m F_m^T, & t \in [t_0, T] \\ Y(t_0) = 0. \end{cases} \tag{30}$$

Proof The derivative of the matrix function $Y_m(t)$ as given by (29) is

$$\dot{Y}_m(t) = -e^{(t-t_0) \mathbb{H}_m^A} \left(\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m (\mathbb{H}_m^B)^T \right) e^{(t-t_0) (\mathbb{H}_m^B)^T}, \quad t \in [t_0, T].$$

On the other hand, we have

$$\begin{aligned} & \mathbb{H}_m^A Y_m(t) + Y_m(t) (\mathbb{H}_m^B)^T - E_m F_m^T = \\ & -e^{(t-t_0) \mathbb{H}_m^A} \left(\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m (\mathbb{H}_m^B)^T \right) e^{(t-t_0) (\mathbb{H}_m^B)^T} \\ & + \mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m (\mathbb{H}_m^B)^T - E_m F_m^T. \end{aligned}$$

Thus, it follows that

$$\dot{Y}_m(t) - \mathbb{H}_m^A Y_m(t) - Y_m(t) (\mathbb{H}_m^B)^T + E_m F_m^T = 0.$$

Moreover, $Y_m(t)$ satisfies the initial condition $Y_m(t_0) = 0$. □

Next, the following proposition gives a useful expression of the residual which is defined, in the low-rank case, by

$$R_m(t) = \dot{X}_m(t) - (A X_m(t) + X_m(t) B - E F^T), \quad t \in [t_0, T]. \tag{31}$$

Proposition 8 *The residual for the differential equation is given by*

$$R_m(t) = -V_{m,r}^A Y_m(t) (V_m^B)^T - V_m^A Y_m(t) (V_{m,r}^B)^T, \tag{32}$$

$$= V_{m,r}^A Z_m(t) (V_m^B)^T + V_m^A Z_m(t) (V_{m,r}^B)^T + \tilde{R}_m, \tag{33}$$

where $Z_m(t) = e^{(t-t_0) \mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0) (\mathbb{H}_m^B)^T}$ and \tilde{R}_m is the algebraic residual given by (25). In addition,

$$\left(V_m^A \right)^T R_m(t) V_m^B = 0_{m \times m}, \quad \forall t \in [t_0, T]. \tag{34}$$

Proof Using the definition (31) of the residual $R_m(t)$ and replacing $X_m(t)$ by its expression given in (28), we get

$$R_m(t) = V_m^A \dot{Y}_m(t) (V_m^B)^T - A V_m^A Y_m(t) (V_m^B)^T - V_m^A Y_m(t) (V_m^B)^T B + E F^T.$$

Now, using the algebraic relation (9) in which M is replaced either by A or by B , we obtain

$$\begin{aligned} R_m(t) = & V_m^A \dot{Y}_m(t) (V_m^B)^T - \left(V_m^A \mathbb{H}_m^A + V_{m,r}^A \right) Y_m(t) (V_m^B)^T \\ & - V_m^A Y_m(t) \left((\mathbb{H}_m^B)^T (V_m^B)^T + (V_{m,r}^B)^T \right) + V_m^A E_m F_m^T (V_m^B)^T. \end{aligned}$$

This may be arranged as following

$$R_m(t) = \mathbb{V}_m^A \left(\dot{Y}_m(t) - \mathbb{H}_m^A Y_m(t) - Y_m(t)(\mathbb{H}_m^B)^T + E_m F_m^T \right) (\mathbb{V}_m^B)^T - \mathbb{V}_{m,r}^A Y_m(t) (\mathbb{V}_m^B)^T - \mathbb{V}_m^A Y_m(t) (\mathbb{V}_{m,r}^B)^T.$$

Taking into account (30), we get (32). The relation (33) follows by replacing $Y_m(t)$ by its expression (29) and taking into account (27). Finally, (34) is straightforward since \mathbb{V}_m^A and \mathbb{V}_m^B are orthogonal matrices. \square

Similarly to the full-rank case, let us remark that if \mathbb{H}_m^A and \mathbb{H}_m^B are stable, then

$$\lim_{T \rightarrow +\infty} R_m(T) = \tilde{R}_m.$$

Let $E_m(t) = X^*(t) - X_m(t)$ be the error at the step m . As in the previous subsection, we have the following error estimates.

Theorem 3 *Suppose that m steps of the block Arnoldi process were run and let $Z_m(t) := e^{(t-t_0)\mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0)(\mathbb{H}_m^B)^T}$. Then, we have the following error estimate:*

$$\|E_m(t)\| \leq \left(\frac{e^{(t-t_0)(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) \sqrt{(r_m^A + \|z_m^A(t)\|)^2 + (r_m^B + \|z_m^B(t)\|)^2}, \quad \forall t \in [t_0, T].$$

It follows that

$$\|E_m\|_\infty \leq \left(\frac{e^{\Delta t(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) \sqrt{(r_m^A + \|z_m^A\|_\infty)^2 + (r_m^B + \|z_m^B\|_\infty)^2},$$

where $r_m^A = \|H_{m+1,m}^A \bar{Y}_m\|$, $r_m^B = \|\hat{Y}_m (H_{m+1,m}^B)^T\|$, $z_m^A(t) = H_{m+1,m}^A \bar{Z}_m(t)$ and $z_m^B(t) = \hat{Z}_m(t) (H_{m+1,m}^B)^T$.

The matrices \bar{Y}_m , $\bar{Z}_m(t)$ are of size $r \times r$ and are formed by the last r rows of \tilde{Y}_m and $Z_m(t)$ respectively while the matrices \hat{Y}_m , \hat{Z}_m are of size $r \times r$ and are formed by the last r columns of \tilde{Y}_m and $Z_m(t)$, respectively.

Proof As previously done in the proof of Theorem 2, we obtain by similar arguments that, for all $t \in [t_0, T]$ we have

$$\|E_m(t)\| \leq \left(\frac{e^{(t-t_0)(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) \|R_m(t)\|.$$

From (33) and (27), we get

$$R_m(t) = \mathbb{V}_{m,r}^A (Z_m(t) - \tilde{Y}_m) (\mathbb{V}_m^B)^T + \mathbb{V}_m^A (Z_m(t) - \tilde{Y}_m) (\mathbb{V}_{m,r}^B)^T.$$

As the $n \times n$ matrices $\mathbb{V}_{m,r}^A (Z_m(t) - \tilde{Y}_m) (\mathbb{V}_m^B)^T$ and $\mathbb{V}_m^A (Z_m(t) - \tilde{Y}_m) (\mathbb{V}_{m,r}^B)^T$ are F-orthogonal, namely

$$\langle \mathbb{V}_{m,r}^A (Z_m(t) - \tilde{Y}_m) (\mathbb{V}_m^B)^T | \mathbb{V}_m^A (Z_m(t) - \tilde{Y}_m) (\mathbb{V}_{m,r}^B)^T \rangle = 0.$$

Therefore,

$$\|R_m(t)\|^2 = \left\| V_{m,r}^A (Z_m(t) - \tilde{Y}_m) (V_m^B)^T \right\|^2 + \left\| \mathbb{V}_m^A (Z_m(t) - \tilde{Y}_m) (V_{m,r}^B)^T \right\|^2.$$

Now, using the triangular inequality, we get that for all $t \in [t_0, T]$, we have

$$\left\| V_{m,r}^A (Z_m(t) - \tilde{Y}_m) (V_m^B)^T \right\| \leq r_m^A + \|z_m^A(t)\|,$$

and similarly, we also have

$$\left\| \mathbb{V}_m^A (Z_m(t) - \tilde{Y}_m) (V_{m,r}^B)^T \right\| \leq r_m^B + \|z_m^B(t)\|.$$

which completes the proof. □

To continue the description of the present method, we notice that (32) enables us to check if $\|R_m(t)\| < \text{tol}$ -where tol is some fixed tolerance-, without having to compute extra products involving the large matrices A and B . More precisely, we have

$$\|R_m(t)\| = \sqrt{\|H_{m+1,m}^A (\mathbb{E}_m^{(r)})^T Y_m(t)\|^2 + \|Y_m(t) \mathbb{E}_m^{(r)} (H_{m+1,m}^B)^T\|^2}. \tag{35}$$

We end this subsection by recalling that in the case of large-scale problems, and as suggested in [24, 39], it is important to get the approximate solution $X_k := X_m(t_k)$ at each time t_k as a product of two low-rank matrices. If $Y_k = V \Sigma W^T$ is the singular value decomposition of Y_k , where $\Sigma = \text{diag}[\sigma_1, \sigma_2, \dots, \sigma_{mr}]$ is the diagonal matrix of the singular values of Y_k sorted in decreasing order, then by considering V_l and W_l the $mr \times l$ matrices of the first l columns of V and W corresponding respectively to the l singular values of magnitude greater than some tolerance τ , we get for each $k = 1, \dots, N$

$$X_k \approx Z_k^A (Z_k^B)^T,$$

where $Z_k^A = \mathbb{V}_m^A V_l \Sigma_l^{1/2}$ and $Z_k^B = \mathbb{V}_m^B W_l \Sigma_l^{1/2}$.

The block Arnoldi combined with the constant solution method (BA-CSM) for solving the differential Sylvester matrix equation, in the case where C is low-rank, i.e., $C = EF^T$, is summarized in Algorithm 4.

Before investigating the performance and efficiency of the different algorithms described previously, we will show in the next section, how to construct a differential Sylvester equation which have a known exact solution.

Algorithm 4 Block Arnoldi Constant Solution Method (BA-CSM) (Low-rank case).

- 1: **Input:** The matrices A, B, E, F , the initial and the final times t_0, T , a tolerance $\epsilon_{\text{tol}} > 0$, a maximum number of iterations M_{max} , a step-size parameter p , the number N of nodes in the time discretization and the tolerance τ for the truncated SVD.
- 2: **Output** $X_{m,1}, \dots, X_{m,N}$, where $X_{m,k} := X_m(t_k)$, ($1 \leq k \leq N$)
- 3: Compute $\delta T = (T - t_0)/N$.
- 4: **for** $m = 1, \dots, M_{\text{max}}$ **do**
- 5: Compute V_m^A and V_m^B to update the orthonormal bases $\mathbb{V}_m^A = [V_1^A, \dots, V_m^A]$, $\mathbb{V}_m^B = [V_1^B, \dots, V_m^B]$ and get the m -th blocks of \mathbb{H}_m^A and \mathbb{H}_m^B by applying Algorithm 2 to (A, E) and (B^T, F) respectively;
- 6: **if** m is a multiple of p **then**
- 7: Compute: $E_m = (\mathbb{V}_m^A)^T E$ and $F_m = (\mathbb{V}_m^B)^T F$.
- 8: Solve the reduced Sylvester equation: $\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m (\mathbb{H}_m^B)^T = E_m F_m^T$.
- 9: **for** $k = 1, \dots, N$ **do**
- 10: Compute $t_k = t_{k-1} + \delta T$.
- 11: Compute $Y_{m,k} := Y_m(t_k) = -e^{(t_k-t_0)\mathbb{H}_m^A} \tilde{Y}_m e^{(t_k-t_0)(\mathbb{H}_m^B)^T} + \tilde{Y}_m$.
- 12: Compute $r_{m,k} = \sqrt{\|H_{m+1,m}^A (\mathbb{E}_m^{(r)})^T Y_k\|^2 + \|Y_k \mathbb{E}_m^{(r)} (H_{m+1,m}^B)^T\|^2}$.
- 13: **end for**
- 14: Compute $r_{\text{max}} = \max\{r_{m,1}, \dots, r_{m,N}\}$
- 15: **if** $r_{\text{max}} < \epsilon_{\text{tol}}$ **then**
- 16: go to line 20;
- 17: **end if**
- 18: **end if**
- 19: **end for**
- 20: **for** $k = 1, \dots, N$ **do**
- 21: Compute the SVD of Y_k , i.e., $Y_k = U \Sigma W^T$ where $\Sigma = \text{diag}[\sigma_1, \dots, \sigma_{mr}]$ and $\sigma_1 \geq \dots \geq \sigma_{mr}$;
- 22: Find l such that $\sigma_{l+1} \leq \tau < \sigma_l$ and let $\Sigma_l = \text{diag}[\sigma_1, \dots, \sigma_l]$;
- 23: Form $Z_k^A = \mathbb{V}_m^A U_l \Sigma_l^{1/2}$ and $Z_k^B = \mathbb{V}_m^B W_l \Sigma_l^{1/2}$;
- 24: The approximate solution $X_{m,k}$ at time t_k is $X_{m,k} \approx Z_k^A (Z_k^B)^T$.
- 25: **end for**

5 Two constructed benchmark examples

In this section, we construct two examples of Sylvester (or Lyapunov) differential systems for which we give an explicit formula for computing the unique solution. The main idea behind such a construction is to have at hand a differential equation for which we have a reference solution to which we can compare the approximate solutions delivered by the different proposed methods. It should be noted that both of the two constructed benchmark examples can be used in large-scale cases, as can be seen in the following subsections.

5.1 A benchmark example based on nilpotent matrices

In this first example, we construct a differential Sylvester matrix equations for which the unique exact solution is explicitly known. To the best of our knowledge, this construction is new and has never been proposed before, in the literature. Moreover, we derive two formulas for the exact solution. The first one is based on the integral formula while the second one is based on the constant solution approach.

Let $p_0 \geq 3$ be a small integer, n_0, s_0 two other integers and $K, R \in \mathbb{R}^{p_0 \times p_0}$ be two nilpotent matrices of index p_0 ($K^{p_0} = R^{p_0} = 0_{p_0}$). Let also $n = p_0 n_0, s = p_0 s_0$ and $A_0 \in \mathbb{R}^{n_0 \times n_0}, B_0 \in \mathbb{R}^{s_0 \times s_0}, X_0, C \in \mathbb{R}^{n \times s}$. For two real numbers α and β , we consider the matrices

$$A = \alpha I_n + A_0 \otimes K, \quad B = \beta I_s + B_0 \otimes R. \tag{36}$$

Then, for any real t and for any matrix X of size $n \times s$, we can check that:

$$e^{tA} X e^{tB} = \left(\sum_{i=0}^{p_0-1} \sum_{j=0}^{p_0-1} t^{i+j} L_{i,j}(X) \right) e^{(\alpha+\beta)t},$$

where $L_{i,j}(X) = \frac{1}{i!j!} (A_0^i \otimes K^i) X (B_0^j \otimes R^j)$.

Assuming that $\alpha + \beta < 0$, then the unique solution \tilde{X}^* of the algebraic Sylvester matrix equation $A X + X B = C$ is given by the formula (see [32]),

$$\tilde{X}^* = - \int_0^{+\infty} e^{tA} C e^{tB} dt.$$

Straightforward calculations give

$$\tilde{X}^* = \sum_{i=0}^{p_0-1} \sum_{j=0}^{p_0-1} \frac{(-1)^{i+j}}{(\alpha + \beta)^{i+j+1}} C_{i,j},$$

where $C_{i,j} = L_{i,j}(C) = \frac{1}{i!j!} (A_0^i \otimes K^i) C (B_0^j \otimes R^j)$. Then, using formula (6), the unique solution of the differential Sylvester matrix equation (1) is the matrix function $X^*(t)$ given by

$$X^*(t) = e^{(t-t_0)A} (X_0 - \tilde{X}^*) e^{(t-t_0)B} + \tilde{X}^*.$$

It follows that

$$X^*(t) = \sum_{i=0}^{p_0-1} \sum_{j=0}^{p_0-1} \left[(t - t_0)^{i+j} e^{(\alpha+\beta)(t-t_0)} L_{i,j}(X_0 - \tilde{X}^*) + \frac{(-1)^{i+j}}{(\alpha + \beta)^{i+j+1}} L_{i,j}(C) \right] \tag{37}$$

Conversely, we may check that the matrix function given by (37) is a solution of the differential Sylvester equation satisfying the given initial condition. This leads to note that the condition $\alpha + \beta < 0$ is in fact superfluous.

We may also obtain the solution $X^*(t)$ of the differential Sylvester matrix equation by using the integral formula (3), since we have

$$X^*(t) = e^{(t-t_0)A} X_0 e^{(t-t_0)B} - \int_{t_0}^t e^{(t-u)A} C e^{(t-u)B} du.$$

It follows that,

$$X^*(t) = \sum_{i=0}^{p_0-1} \sum_{j=0}^{p_0-1} \left[(t - t_0)^{i+j} e^{(\alpha+\beta)(t-t_0)} L_{i,j}(X_0) - I_{i+j}(t) L_{i,j}(C) \right], \quad (38)$$

where the scalar functions $I_k(t)$ are given by $I_k(t) = \int_{t_0}^t (t-u)^k e^{(\alpha+\beta)(t-u)} du$. The expression of the functions $I_k(t)$ are obtained by recursion. Indeed, we have

$$I_0(t) = \frac{1}{\alpha + \beta} \left(e^{(\alpha+\beta)(t-t_0)} - 1 \right),$$

and for $k \geq 1$ by parts integration, we have

$$I_k(t) = \frac{1}{\alpha + \beta} \left((t - t_0)^k e^{(\alpha+\beta)(t-t_0)} - k I_{k-1}(t) \right).$$

Then, we may show by induction that, for all $k \geq 0$, we have

$$I_k(t) = \frac{k!}{(\alpha + \beta)^{k+1}} \left(\sum_{\ell=0}^k (-1)^\ell \frac{(\alpha + \beta)^\ell (t - t_0)^\ell}{\ell!} e^{(\alpha+\beta)^\ell (t-t_0)} - (-1)^k \right).$$

Before ending this subsection, let us remark that in this benchmark example, the matrix C is arbitrary and then can also be taken in the low-rank form $C = E F^T$, where $E \in \mathbb{R}^{n \times r}$ and $F \in \mathbb{R}^{s \times r}$.

5.2 A benchmark example based on the spectral decomposition

The second benchmark example is inspired by the paper by Behr et al, see [5]. However, unlike the techniques described in [5], we do not assume that the matrix coefficients are diagonalizable, and instead we propose a technique based on a truncated (partial) spectral decomposition of the matrix coefficients.

In the following, given a square matrix M of size n and a rank $q_M \ll n$, the partial spectral decomposition of rank q_M associated with M that we consider is the one given by $M U_M = U_M D_M$ where $D_M = \text{diag}(\mu_1, \dots, \mu_{q_M})$ is the $q_M \times q_M$ diagonal matrix formed by the q_M largest magnitude eigenvalues of M and U_M is the rectangular matrix of size $n \times q_M$ whose columns are the corresponding eigenvectors. We recall that, in Matlab, the coefficients D_M and U_M are obtained via the instruction $[U_M, D_M] = \text{eigs}(M, q_M)$.

5.2.1 Sylvester differential case

Here, we assume the existence of two partial spectral decomposition of rank $q_A \ll n$ and $q_B \ll s$ associated to A and B^T respectively. Thus, there exists two rectangular matrices U_A and U_B of size $n \times q_A$ and $s \times q_B$ and two diagonal matrices

$D_A = \text{diag}[\alpha_1, \dots, \alpha_{q_A}]$ and $D_B = \text{diag}[\beta_1, \dots, \beta_{q_B}]$ of size $q_A \times q_A$ and $q_B \times q_B$ respectively, such that $A U_A = U_A D_A$ and $B^T U_B = U_B D_B$. Moreover, we assume that the condition $\alpha_i + \beta_j \neq 0$ holds for all $1 \leq i \leq q_A$ and for all $1 \leq j \leq q_B$. Then, letting E_1 of size $q_A \times r$, F_1 of size $q_B \times r$ be two given block vectors and $Q = \left(\frac{1}{\alpha_i + \beta_j} \right)_{\substack{1 \leq i \leq q_A \\ 1 \leq j \leq q_B}}$, it is obvious to see, that

$$\tilde{Y}^* = Q \odot (E_1 F_1^T),$$

is the unique solution of the following algebraic Sylvester equation $D_A Z + Z D_B = E_1 F_1^T$.

Using the constant solution method, we get that the unique solution $Y^*(t)$ of the linear differential system

$$\begin{cases} \dot{Y}(t) = D_A Y(t) + Y(t) D_B - E_1 F_1^T \\ Y(t_0) = 0, \end{cases} \tag{39}$$

is given by

$$Y^*(t) = -e^{(t-t_0) D_A} \tilde{Y}^* e^{(t-t_0) D_B} + \tilde{Y}^*.$$

As D_A and D_B are diagonal matrices, then

$$\begin{aligned} e^{(t-t_0) D_A} &= \text{diag}[e^{(t-t_0)\alpha_1}, \dots, e^{(t-t_0)\alpha_{q_A}}] \quad \text{and} \quad e^{(t-t_0) D_B} \\ &= \text{diag}[e^{(t-t_0)\beta_1}, \dots, e^{(t-t_0)\beta_{q_B}}]. \end{aligned}$$

It follows that, the unique solution of the differential system (39) is given by

$$Y^*(t) = (-G(t) + Q) \odot (E_1 F_1^T) = H(t) \odot (E_1 F_1^T), \tag{40}$$

where the matrix-valued functions $G(t)$ and $H(t)$ are given by

$$G(t) = \left(\frac{e^{(t-t_0)(\alpha_i + \beta_j)}}{\alpha_i + \beta_j} \right)_{\substack{1 \leq i \leq q_A \\ 1 \leq j \leq q_B}} \quad \text{and} \quad H(t) = -G(t) + Q = \left(\frac{1 - e^{(t-t_0)(\alpha_i + \beta_j)}}{\alpha_i + \beta_j} \right)_{\substack{1 \leq i \leq q_A \\ 1 \leq j \leq q_B}}.$$

Now, multiplying the differential equation in (39), on the left by U_A and on the right by U_B^T and using the fact that $A U_A = U_A D_A$ and $B^T U_B = U_B D_B$, we get

$$U_A \dot{Y}(t) U_B^T = A U_A Y(t) U_B^T + U_A Y(t) U_B^T B - (U_A E_1)(U_B F_1)^T.$$

Consequently, if $E = U_A E_1$ and $F = U_B F_1$, then the following Sylvester differential system

$$\begin{cases} \dot{X}(t) = A X(t) + X(t) B - E F^T \\ X(t_0) = 0, \end{cases} \tag{41}$$

has $X^*(t) = U_A Y^*(t) U_B^T$ as the unique solution. From (40), we have

$$X^*(t) = U_A \left(H(t) \odot (E_1 F_1^T) \right) U_B^T. \tag{42}$$

Furthermore, let us notice that the algebraic Sylvester equation $A X + X B = E F^T$ has as unique solution the matrix \tilde{X}^* given by

$$\tilde{X}^* = U_A \left(Q \odot (E_1 F_1^T) \right) U_B^T.$$

5.2.2 Lyapunov differential case

Let A be a large sparse matrix of size $n \times n$, M be a real symmetric definite positive matrix of size $n \times n$ and $q \ll n$ a small integer. Next, we consider the truncated spectral decomposition of order q of the pair (A, M) given by $A U = M U D$ where $D = \text{diag}[\alpha_1, \dots, \alpha_q]$ of size $q \times q$ is a diagonal matrix containing the q first largest magnitude eigenvalues of $M^{-1}A$ and U of size $n \times q$ whose columns are the corresponding eigenvectors. Let also E_1 be a full-rank block vector of size $q \times r$ with $r \ll n$.

Let us also consider the following $q \times q$ differential system:

$$\begin{cases} \dot{Y}(t) = D Y(t) + Y(t) D - E_1 E_1^T \\ Y(t_0) = 0. \end{cases} \tag{43}$$

Then, its associated algebraic equation $D Z + Z D = E_1 E_1^T$ has as unique solution $\tilde{Y}^* = Q \odot (E_1 E_1^T)$, where the matrix Q is given by $Q = \left(\frac{1}{\alpha_i + \alpha_j} \right)_{1 \leq i, j \leq q}$.

Using the constant solution method, we verify that $Y^*(t) = -e^{(t-t_0)D} \tilde{Y}^* e^{(t-t_0)D} + \tilde{Y}^*$, is the unique solution of the linear system (43). In addition, as D is diagonal, then $e^{(t-t_0)D} = \text{diag}[e^{(t-t_0)\alpha_1}, \dots, e^{(t-t_0)\alpha_q}]$ and the unique solution is given by

$$Y^*(t) = (-G(t) + Q) \odot (E_1 E_1^T) = F(t) \odot (E_1 E_1^T), \tag{44}$$

where the matrix-valued functions $G(t)$ and $F(t)$ are

$$G(t) = \left(\frac{e^{(t-t_0)(\alpha_i + \alpha_j)}}{\alpha_i + \alpha_j} \right)_{1 \leq i, j \leq q} \quad \text{and} \quad H(t) = -G(t) + Q = \left(\frac{1 - e^{(t-t_0)(\alpha_i + \alpha_j)}}{\alpha_i + \alpha_j} \right)_{1 \leq i, j \leq q}.$$

Now, left and right multiplying the differential equation in (43) by $M U$ and $U^T M^T$ respectively, we get

$$M U \dot{Y}(t) U^T M^T = M U D Y(t) U^T M^T + M U Y(t) D U^T M^T - F F^T,$$

where the block vector $F = M U E_1$ is of size $n \times r$. Now, as $A U = M U D$, then we get that $X^*(t) = U Y^*(t) U^T$ is the unique solution the following Lyapunov matrix differential equation

$$\begin{cases} M \dot{X}(t) M^T = A X(t) M^T + M X(t) A^T - F F^T \\ X(t_0) = 0. \end{cases} \tag{45}$$

Finally, we note that from (44), we get

$$X^*(t) = U \left[H(t) \odot (E_1 E_1^T) \right] U^T. \tag{46}$$

We end this subsection by recalling that generalized differential Lyapunov matrix equations of kind (45) are used to define optimal controls for a finite element discretization of a heat equation, see [5, 9]. We also note that the last differential system (45) is equivalent to the following one:

$$\begin{cases} \dot{X}(t) = \tilde{A} X(t) + X(t) \tilde{A}^T - E E^T \\ X(t_0) = 0, \end{cases} \tag{47}$$

where $\tilde{A} = M^{-1} A$ and $E = U E_1$.

6 Numerical experiments

In this section, a serie of numerical tests is presented to examine the performance and potential of Algorithms 1, 3 and 4. We have compared our proposed method which is based on relation (6) with those based on the integral formula (3) and described in [21, 38]. We recall that the algorithms described in the two cited previous papers only provide an approximate solution at the final time T and moreover they only deal with the case of low-rank differential equations. Thus, we modified Algorithm 1 and Algorithm 4 proposed in [21] and [38] respectively, so that they provide an approximate solution $X_{m,k} = X_m(t_k)$ at each node t_k of the discretization of the time interval $[0, T]$ as it is the case in Algorithm 4. Moreover, we have drafted other codes based on the integral formula (3) and equivalent to Algorithms 1 and 3.

It should be noted that in all the examples given here, we suppose that the matrix X_0 appearing in the initial condition of (1) is equal to zero, i.e., $X_0 = 0_{n \times s}$. Furthermore, we consider different time intervals $[t_0, T]$ where $t_0 = 0$ is fixed once and for all, while T is indicated in each example. The time interval $[0, T]$ is divided into sub-intervals of constant length $\delta_T = \frac{T}{N}$ where N is the number of nodes. All the numerical experiments were performed using MATLAB and have been carried out on an Intel(R) Core(TM) i7 with 2.60 GHz processing speed and 16 GB memory. In order to implement the different algorithms described in this work, we used the following MATLAB functions:

- `expm`: it allows to calculate the exponential of a square matrix. This function is based on a scaling and squaring algorithm with a Padé approximation [25].

- `lyap`: it allows to solve Sylvester or Lyapunov matrix equations. For our purposes, the instruction `lyap(A,B,-C)` delivers the matrix X solution of the algebraic Sylvester equation $A X + X B = C$.
- `integral`: it allows to calculate numerically an integral, using the arguments “ArrayValued” and “true.”
- `eigs`: it allows to calculate numerically the partial spectral decomposition of a couple of sparse matrices.

Furthermore, we precise that when the constant solution or integral formula methods are combined with the block Arnoldi process to obtain an approximate solution to the differential equation, the iterations were stopped as soon as the dimension of the Krylov subspace generated by the block Arnoldi process reaches a maximum value $m = M_{max} = 110$ or as soon as the maximal norm r_{max} computed by the algorithm is lower than $10^{-10} \mu$ where $\mu = \|A\| + \|B\| + \|C\|$ in the full-rank case and $\mu = \|A\| + \|B\| + \|E\| \|F\|$ in the low-rank case. We also mention that in the numerical examples, the right-hand side C or its factors E and F were generated randomly.

To compare the performances of the constant solution method (in short CS or CS-BA when combined with the block Arnoldi process) with those of the Integral Formula method (in short IF or IF-BA when combined with the block Arnoldi process or IF-GA when combined with the global Arnoldi process), we used the following comparison criteria:

- TR: the time ratio between the cpu-time of a Constant Solution (CS) based method and an Integral Formula (IF) based method is defined by

$$\text{TR}_{\text{MI}}^{\text{MC}} = \frac{\text{cpu-time}(\text{MC})}{\text{cpu-time}(\text{MI})}, \quad (48)$$

where $\text{MC} \in \{\text{CS}, \text{CS-BA}\}$ stands for one of the methods based on the constant solution approach and $\text{MI} \in \{\text{IF}, \text{IF-BA}, \text{IF-GA}\}$ denotes one of the methods based on the integral formula.

- RDN: the relative difference norm between $X^{\text{CS-BA}}$ and $X^{\text{IF-BA}}$ which are the approximate solutions delivered by the constant solution and the integral formula methods respectively when they are combined the block Arnoldi process.

$$\text{RDN}_{\text{IF-BA}}^{\text{CS-BA}} = \max_{k=0,1,\dots,N} \frac{\|X_k^{\text{CS-BA}} - X_k^{\text{IF-BA}}\|}{\|X_k^{\text{IF-BA}}\|}.$$

We point out that this criteria is used when the exact solution of the differential Sylvester equation is not available.

- REN: the relative error norm between the exact solution and an approximate solution obtained either by a constant solution based algorithm or by an integral formula

based algorithm. More precisely, letting X^{Ref} be the **reference** solution computed by (37), (38), (42) or (46), we define the following quantities:

$$\left\{ \begin{array}{l} \text{REN}^{CS-BA} = \max_{k=0,1,\dots,N} \frac{\|X_k^{CS-BA} - X_k^{Ref}\|}{\|X_k^{Ref}\|}, \\ \text{REN}^{IF-BA} = \max_{k=0,1,\dots,N} \frac{\|X_k^{IF-BA} - X_k^{Ref}\|}{\|X_k^{Ref}\|} \\ \text{REN}^{IF-GA} = \max_{k=0,1,\dots,N} \frac{\|X_k^{IF-GA} - X_k^{Ref}\|}{\|X_k^{Ref}\|} \end{array} \right. \quad (49)$$

Before describing the different numerical tests we performed, we mention that, to generate the exact solution given by (37) in the first benchmark example, we chose $p_0 = 3$ and we considered different values for the parameters α, β as well as different matrices A_0, B_0 . On the other hand the matrices K and R are fixed and are once and for all, as follows

$$K = \begin{bmatrix} 3 & 8 & -19 \\ -1 & -5 & 11 \\ 0 & -1 & 2 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & -1 \end{bmatrix}.$$

6.1 Experiment 1

In this first example, the numerical tests are done with moderate size matrices A and B . We compare the solution provided by our proposed constant solution method implemented via Algorithm 1 with the one obtained using the integral formula (3) as well as with the solutions given by some classical ODE’s solvers from Matlab. The solvers **ode15s**, **ode23s**, **ode23t**, and **ode23tb** are usually used for stiff ODE’s, while the other solvers **ode45**, **ode23**, and **ode113** are used for non stiff ODE’s. Note that since some ODE solvers behave similarly and in order not to overload the plots, we only give the results obtained with the four methods **ode15s**, **ode23s**, **ode23tb**, and **ode45**.

Experiment 1.1 In the following experiment, we consider the time intervals $[0, T]$ with T is either $T = 1$ with the number of nodes is $N = 10$ or $T = 10$ with the number of nodes is $N = 50$ which means that the step time is $\delta_T = 0.1$ when $T = 1$ while $\delta_T = 0.2$ when $T = 10$. Here, we consider the matrices $A_0 = \mathbf{gallery}(\text{'leslie,'}, n_0)$, $B_0 = \mathbf{gallery}(\text{'minij,'}, s_0)$ with $n_0 = 50$ and $s_0 = 10$ and the coefficient matrices A, B of the differential Sylvester equations are generated by (36), as explained in the benchmark example. The parameters α, β are equal to -2 and -1 respectively. As the matrices K, R are those given at the beginning of Section 6, the size of the matrices A, B are now $n = 150$ and $s = 30$, respectively. Here, we point out that the solution computed by Algorithm 1 and those computed by the Algorithm based on integral formula or issued by the Matlab ODE solvers are compared to the exact one

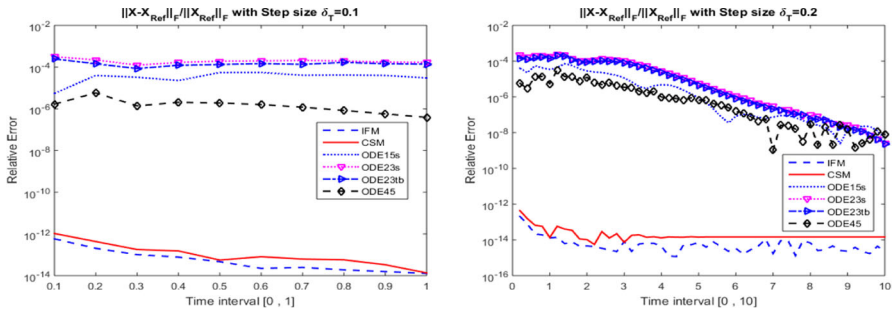


Fig. 1 Experiment 1.1: comparison of the relative error norm. The reference solution is given by (37)

given in (37) which is considered as the reference solution X^{ref} . Thus, in the plots, we represent the behaviour of the norm of the relative error

$$t_k \rightarrow \frac{\|X_k - X_k^{Ref}\|}{\|X_k^{Ref}\|}$$

as a function of t_k where $t_k = k \delta_T$. The obtained plots and results are reported in Fig. 1 and Table 1 respectively.

Experiment 1.2 Here, we give the results when solving a differential Lyapunov matrix equation. We consider the time interval $[0, T]$ with $T = 10$ and the number of nodes $N = 50$ which means that the step time is $\delta = 0.2$. Two test matrices are considered which are $A = A_1 = \text{-gallery('lehmer', } n)$ and $A = A_2 = \text{-gallery('minij', } n)$ with $n = 70$. Note that in the present test, the reference solution is the one obtained via a partial spectral decomposition of rank $q_A = 15$. More precisely, the reference solution X^{Ref} is the one given by (46) for the particular case $M = I_n$. The obtained plots and results are reported in Fig. 2 and Table 2 respectively.

The analysis of results obtained in Experiments 1.1 and Experiments 11.2 shows on the one hand that the CS and IF methods return the best results in terms of the error norm. The **ode45** solver is the best among the other Matlab solvers, but its performance does not match that of the CS and IF which are

$$- \text{TR}_{CS}^{IF} = \frac{12.578}{0.203} \simeq 61 \text{ for } (T, N) = (1, 10) \text{ and } \text{TR}_{CS}^{IF} = \frac{75.703}{0.718} \simeq 105 \text{ for } (T, N) = (10, 50), \text{ in Experiment 1.1.}$$

Table 1 The obtained CPU times (in seconds) in Experiment 1.1

(T, N) Method	CSM	IFM	ode15s	ode23s	ode23tb	ode45
(1, 10)	0.203	12.578	43.546	750.756	108.484	0.343
(10, 50)	0.718	75.703	78.984	1574.980	226.016	0.390

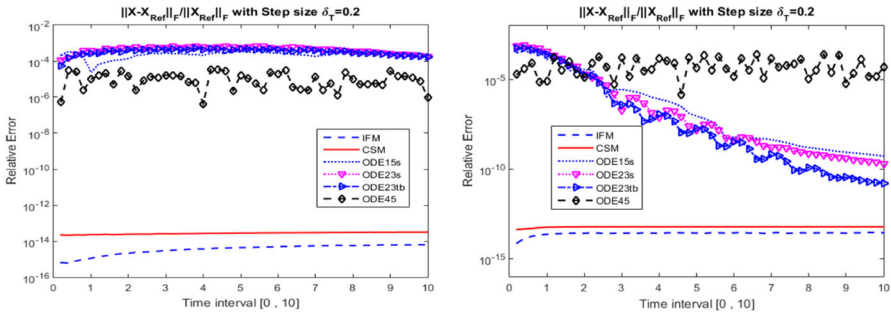


Fig. 2 Experiment 1.2: comparison of the relative error norm. The reference solution is given by (46)

$$- \text{TR}_{\text{CS}}^{\text{IF}} = \frac{49.650}{0.453} \simeq 109 \text{ for } A = A_1 \text{ and } \text{TR}_{\text{CS}}^{\text{IF}} = \frac{59.546}{0.515} \simeq 115 \text{ for } A = A_2, \text{ in Experiment 1.2,}$$

we clearly see that CS is faster than IF because the former avoids using a quadrature formula as it is the case for the later.

6.2 Experiment 2

In this set of numerical tests, the experiments are done with a relatively large matrix A and a moderate size matrix B . We compare the performances of Algorithm 3 -which implements the CS-BA method- and the equivalent algorithm based on the integral formula combined with the block Arnoldi (IF-BA).

Experiment 2.1. The matrices A and B are obtained from the centered finite difference discretization of the operators

$$\begin{cases} L_A(u) = \Delta u - f_A \frac{\partial u}{\partial x} - g_A \frac{\partial u}{\partial y} - h_A u \\ L_B(u) = \Delta u - f_B \frac{\partial u}{\partial x} - g_B \frac{\partial u}{\partial y} - h_B u, \end{cases}$$

on the unit square $[0, 1] \times [0, 1]$ with homogeneous Dirichlet boundary conditions where

$$f_A(x, y) = (x + 10y^2), \quad g_A(x, y) = \sqrt{2x^2 + y^2}, \quad h_A(x, y) = x^2 - y^2,$$

Table 2 The obtained CPU times (in seconds) in Experiment 1.2

(A) Method	CSM	IFM	ode15s	ode23s	ode23tb	ode45
A_1	0.453	49.656	40.484	544.0314	127.438	1.25
A_2	0.515	59.5469	45.593	688.984	126.594	40.2344

Table 3 The obtained times ratio TR and relative difference norms RDN in Experiment 2.1

Test Problem		$T = 1, N = 10$		$T = 2, N = 20$	
n	s	$\overline{\text{TR}}_{\text{CS-BA}}^{\text{IF-BA}}$	$\text{RDN}_{\text{IF-BA}}^{\text{CS-BA}}$	$\overline{\text{TR}}_{\text{CS-BA}}^{\text{IF-BA}}$	$\text{RDN}_{\text{IF-BA}}^{\text{CS-BA}}$
900	9	141	8.111e-15	186	1.887e-14
2500	9	163	1.868e-14	276	3.682e-14
900	25	164	6.329e-15	187	1.248e-14
2500	25	177	1.723e-14	186	2.499e-14

and

$$f_B(x, y) = 10xy + 1, \quad g_B(x, y) = e^{-x^2-y^2}, \quad h_A(x, y) = \frac{1}{1 + x^2 + y^2}.$$

To generate the coefficient matrices A and B , we used the **fdm_2d_matrix** function from the LYAPACK toolbox [34] as following $A=\text{fdm_2d_matrix}(n_0, f_A, g_A, h_A)$ and $B=\text{fdm_2d_matrix}(s_0, f_B, g_B, h_B)$ where n_0 and s_0 are the number of inner grid points in each direction when discretizing the operators L_A and L_B respectively. This gives $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{s \times s}$ with $n = n_0^2$ and $s = s_0^2$.

We examine the performances of CS-BA and IF-BA for four choices of n_0 and s_0 which are $(n_0, s_0) = (30, 3), (n_0, s_0) = (50, 3), (n_0, s_0) = (30, 5)$ and $(n_0, s_0) = (50, 5)$. The considered time intervals are $[0, T]$ where $T = 1$ and $N = 10$ or $T = 2$ and $N = 20$. This means that the step time is always $\delta_T = 0.1$. In Table 3, we reported the time ratio (TR) and the relative difference norm (RDN) between the CPU-time of CS-BA and IF-BA.

Experiment 2.2. In this test, we took $A_0 = \text{gallery}(\text{'hanowa,' } 1500, -5)$ and $B_0 = \text{gallery}(\text{'leslie,' } 6)$ from the Matlab gallery and transform them into A et B of sizes $n = 4500$ and $s = 18$ respectively by using (36) in which we took $\alpha = -7$ and $\beta = -5$. The obtained results for different time intervals which are summarized in Table 4 include the time ratio TR and the relative error norms $\text{REN}^{\text{CS-BA}}, \text{REN}^{\text{IF-BA}}$ between the approximate solutions $X^{\text{CS-BA}}, X^{\text{IF-BA}}$ given by CS-BA and IF-BA respectively and the X^{Exact} the exact solution computed by (37).

6.3 Experiment 3

We describe and report here the results of numerical experiments carried out when solving large-scale low-rank differential Sylvester or Lyapunov equations. The performance of CS-BA is compared with that of IF-BA. The test matrices come either from the centred finite difference discretization of the operators L_A and L_B defined in the previous experiment, or from the Florida suite sparse matrix collection [15]. The invoked matrices for our tests from this collection are: **pde900, pde2961, cdde1, Chem97ZtZ, thermal, rdb5000, sstmodel, add32** and **rw5151**.

Experiment 3.1 (a). In this example, the numerical results are those obtained from solving differential Sylvester equations. The time interval is fixed to $[0, 1], (T = 1)$.

Table 4 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative error norms REN^{CS-BA} , REN^{IF-BA} in Experiment 2.2. with $N = 10$

T	TR	REN^{CS-BA}	REN^{IF-BA}
1	11	4.825e-11	4.143e-12
5	20	1.849e-11	9.097e-12
10	30	1.244e-11	3.387e-12
50	1329	7.852e-13	1.621e-11
100	1230	7.802e-13	1.432e-11

The number of nodes is $N = 10$ which gives a step time $\delta_T = 0.1$. The matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{s \times s}$ come from the discretization of the operators L_A and L_B . As indicated previously, the coefficients of the right-hand side $E, F \in \mathbb{R}^{n \times r}$ are randomly generated. The obtained results for different sizes n, s and ranks r are summarized in Table 5.

Experiment 3.1 (b). Here, we consider two different time intervals $[0, T]$ for $T = 1$ and $T = 10$ in which the number of sub-intervals is always $N = 10$. The matrix A is from the Florida sparse matrix collection. We consider the particular case $B = A^T$ and $F = E$ and report the results obtained when solving low-rank differential Lyapunov equations. The obtained results for $r = 2, r = 5$ or $r = 10$ are displayed in Table 6.

Experiment 3.2 (a). Here, we consider $A_0 = \text{pde2961}$ and $B_0 = \text{pde900}$ and transform them into A et B of sizes $n = 8883$ and $s = 2700$ respectively by using (36). In order to confirm the influence of the rank r and/or length T of the time interval, on the performances of the CS and IF methods, we report in Table 7 the results obtained for two cases: **case 1:** $(\alpha, \beta) = (-3, -1)$ and **case 2:** $(\alpha, \beta) = (-0.7, -0.4)$. For each case, we choose T from the set $\{2, 5, 10\}$ and took $N = 10$ for $T = 2, N = 20$ for $T = 5$ and $N = 40$ for $T = 10$. The rank r of the factors E and F is equal to $r = 5, r = 10$ or $r = 20$.

We notice that in most of tests, both methods manage to provide a good approximate solution and that the CPU time is in favor of the BA-CS method. However, we observed that for small values of α and β and when the values of r and T are large, the BA-IF method failed to converge within a reasonable time. The non-convergence is indicated by “---.”

Table 5 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative difference norms RDN_{IF-BA}^{CS-BA} in Experiment 3.1 (a)

r	Test Problems					
	$n_0 = 40, s_0 = 20$		$n_0 = 30, s_0 = 30$		$n_0 = 50, s_0 = 50$	
	$n = 1600, s = 400$		$n = 900, s = 900$		$n = 2500, s = 2500$	
	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}
2	82	2.528e-14	78	5.819e-14	78	1.354e-13
5	145	1.906e-14	136	1.201e-13	159	1.931e-13
10	170	7.256e-14	219	2.067e-14	156	1.383e-13
20	193	2.647e-14	138	6.153e-14	252	2.205e-13

Table 6 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative difference norms RDN_{IF-BA}^{CS-BA} in Experiment 3.1 (b)

		Test Problems									
		A = -cddel <i>n</i> = 961		A = -Chem97ZlZ <i>n</i> = 2541		A = -pde2961 <i>n</i> = 2961		A = thermal <i>n</i> = 3456		A = rdb5000 <i>n</i> = 5000	
<i>T</i>	<i>r</i>	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}	TR_{IF-BA}^{CS-BA}	RDN_{IF-BA}^{CS-BA}
1	5	17	1.032e-14	18	8.376e-15	6.5	8.639e-15	2.3	7.386e-15	42	2.892e-13
	10	44	1.430e-14	50	1.268e-14	13	1.386e-14	6.1	9.442e-15	83	4.434e-13
	2	47	2.815e-14	15	8.190e-15	12	1.002e-14	2.5	3.592e-15	57	3.482e-13
10	5	68	1.483e-14	30	1.002e-14	38	2.911e-14	10	4.631e-15	135	1.374e-12
	10	73	2.910e-14	110	1.253e-14	57	2.713e-14	25	7.497e-15	255	1.617e-12

Table 7 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative error norms REN^{CS-BA} and REN^{IF-BA} in Experiment 3.2 (a)

T	r	$\alpha = -3, \beta = -1$			$\alpha = -0.7, \beta = -0.4$		
		TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{IF-BA}	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{IF-BA}
2	5	1.277	4.777e-14	4.720e-14	1.234	2.641e-11	2.641e-11
	10	1.118	5.147e-14	5.049e-14	1.139	3.022e-11	3.022e-11
	20	3.858	5.473e-14	5.311e-14	2.331	3.401e-11	3.400e-11
5	5	1.292	4.358e-14	4.251e-14	1.160	2.343e-11	2.343e-11
	10	1.371	4.639e-14	4.507e-14	1.124	2.728e-11	2.728e-11
	20	4.230	4.984e-14	4.855e-14	---	3.148e-11	---
10	5	1.237	4.358e-14	4.251e-14	1.093	2.343e-11	2.343e-11
	10	1.399	4.639e-14	4.507e-14	83.424	2.728e-11	2.728e-11
	20	3.974	4.984e-14	4.855e-14	---	3.148e-11	---

Experiment 3.2 (b). In this last set of experiments, we compare the performances of the CS and IF methods when they are applied to the solution of low-rank differential Lyapunov equations. Unlike the previous series of tests, we did not generate a discretization for the interval $[0, T]$ and only calculated the approximation $X(T)$ at the final time, where $T = 10$. Similarly, the rank of $C = E E^T$ does not vary and is $r = 20$. For each experiment with a matrix A_0 -which is taken from the Florida sparse matrix collection [15]-, we considered four values for the scalar α that was used in the generation of the benchmark example. The size n_0 of each matrix A_0 , the size n of the benchmark matrix A as well as the obtained results are reported in Table 8.

6.4 Experiment 4

Here, we compare three methods which are the constant solution method based on the block Arnoldi process (CS-BA), the Integral Formula method based on the block Arnoldi (IF-BA) [20, 21] and the Integral Formula method based on the global Arnoldi process (IF-GA) [38]. We illustrate the performance of the compared methods when they are applied to solve the generalized differential Lyapunov system (47) according to the construction given in Sect.5.2. We point out that this example is derived from a finite-element discretization of a heat equation and is taken from [5, 9]. The matrices A and M are the **rail** matrices from the Suite Sparse Matrix Collection [15] and we consider the choices: $(A,M)=(\mathbf{rail_1357_A}, \mathbf{rail_1357_E})$ $(A,M)=(\mathbf{rail_5177_A}, \mathbf{rail_5177_E})$ and $(A,M)=(\mathbf{rail_20209_A}, \mathbf{rail_20209_E})$. Each of the preceding matrices is of size $n \times n$ with $n = 1357, 5177, 20209$ respectively. The number of nodes for the time interval $[0, T]$ is fixed and is $N = 10$. The approximate solution delivered by each method is compared to the reference solution given by (46) which corresponds to the partial spectral decomposition of rank $q = 25$. The obtained results for different values of the rank r of the right-hand side $E E^T$ ($r \in \{5, 10, 20\}$) and different values of T ($T \in \{1, 5, 50, 500, 5000\}$) are summarized in Tables 9, 10 and 11.

Table 8 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative error norms REN^{CS-BA} and REN^{F-BA} in Experiment 3.2 (b)

Test Problems		$A_0 = \text{pde2961}$ $n_0 = 961, n = 2883$				$A_0 = \text{sstmodel}$ $n_0 = 3345, n = 10035$			
α	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}
-5	5.12	4.33e-14	3.76e-14	2.27	2.14e-14	1.18e-14	1.82	3.43e-12	9.82e-13
-1	---	5.81e-12	---	1087.5	5.37e-12	1.26e-12	200.43	6.45e-14	6.23e-14
-0.5	---	4.58e-10	---	544.31	4.60e-10	1.01e-11	637.17	5.23e-12	2.69e-13
-0.1	---	8.23e-07	---	---	1.19e-07	---	---	3.38e-08	---
Test Problems									
$A_0 = \text{thermal}$									
$n_0 = 3456, n = 10368$									
α	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}
-5	1.53	2.03e-12	1.03e-12	1.48	5.17e-15	3.90e-15	1.37	5.44e-15	4.85e-15
-1	1.69	2.78e-14	2.56e-14	1.45	8.00e-15	1.29e-15	1.22	2.91e-14	2.50e-14
-0.5	1.58	1.20e-13	1.05e-13	1.55	5.77e-15	1.74e-15	391.43	1.44e-13	1.29e-13
-0.1	277.88	1.12e-10	6.94e-11	1.71	1.26e-14	2.46e-15	---	7.47e-11	---
$A_0 = \text{rw5151}$									
$n_0 = 5151, n = 15453$									
α	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}	TR_{IF-BA}^{CS-BA}	REN^{CS-BA}	REN^{F-BA}
-5	1.53	2.03e-12	1.03e-12	1.48	5.17e-15	3.90e-15	1.37	5.44e-15	4.85e-15
-1	1.69	2.78e-14	2.56e-14	1.45	8.00e-15	1.29e-15	1.22	2.91e-14	2.50e-14
-0.5	1.58	1.20e-13	1.05e-13	1.55	5.77e-15	1.74e-15	391.43	1.44e-13	1.29e-13
-0.1	277.88	1.12e-10	6.94e-11	1.71	1.26e-14	2.46e-15	---	7.47e-11	---

Table 9 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative error norms REN^{CS-BA} and REN^{IF-BA} in Experiment 4 with A and M from the set **rail_1357**

T	r	rail_1357				
		REN^{CS-BA}	REN^{IF-BA}	TR_{IF-BA}^{CS-BA}	REN^{IF-GA}	TR_{IF-GA}^{CS-BA}
1	5	6.18e-15	5.44e-15	1.97	3.50e-12	4.97
	10	3.41e-15	4.81e-15	1.94	4.37e-12	3.27
	20	8.66e-15	8.75e-15	7.14	1.75e-12	4.79
5	5	6.17e-15	7.12e-15	1.39	2.75e-13	2.78
	10	3.51e-15	6.61e-15	1.38	4.37e-12	3.15
	20	5.23e-15	7.10e-15	8.58	6.41e-13	4.08
10	5	6.05e-15	2.16e-14	1.46	2.78e-13	3.03
	10	3.41e-15	2.11e-14	1.32	4.05e-14	3.92
	20	5.23e-15	2.07e-14	9.61	1.91e-14	4.01
20	5	6.05e-15	1.11e-13	1.74	3.13e-13	3.35
	10	3.41e-15	1.43e-13	1.69	1.43e-13	4.78
	20	5.23e-15	1.34e-13	15.65	1.35e-13	5.78
50	5	6.05e-15	6.19e-13	1.86	6.77e-13	4.09
	10	3.41e-15	5.91e-13	2.27	5.99e-13	5.14
	20	5.23e-15	5.59e-13	15.51	5.61e-13	5.22

Table 10 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative error norms REN^{CS-BA} and REN^{IF-BA} in Experiment 4 with A and M from the set **rail_5177**

T	r	rail_5177				
		REN^{CS-BA}	REN^{IF-BA}	TR_{IF-BA}^{CS-BA}	REN^{IF-GA}	TR_{IF-GA}^{CS-BA}
1	5	1.11e-10	1.11e-10	1.13	5.04e-08	1.41
	10	8.22e-15	8.79e-15	1.05	3.07e-11	1.47
	20	2.81e-15	3.50e-15	1.46	1.11e-13	1.92
5	5	1.11e-10	1.11e-10	0.98	5.04e-08	1.17
	10	8.22e-15	2.48e-14	1.01	5.30e-14	1.50
	20	2.80e-15	2.22e-14	1.55	1.41e-13	1.88
10	5	1.11e-10	1.11e-10	1.04	5.04e-08	1.19
	10	8.22e-15	1.05e-13	1.04	1.14e-13	1.50
	20	2.80e-15	1.02e-13	1.74	1.27e-13	1.95
20	5	1.11e-10	1.11e-10	1.03	5.04e-08	1.21
	10	8.22e-15	1.00e-12	1.04	1.00e-12	1.48
	20	2.80e-15	9.76e-13	2.04	9.82e-13	1.98
50	5	1.11e-10	1.11e-10	1.03	5.94e-08	1.23
	10	8.22e-15	5.15e-12	1.06	5.15e-12	1.52
	20	2.75e-15	4.96e-12	2.29	4.97e-12	2.09

Table 11 The obtained times ratio TR_{IF-BA}^{CS-BA} and relative error nnorms REN^{CS-BA} and REN^{IF-BA} in Experiment 4 with A and M from the set **rail_20229**

T	r	rail_20229				
		REN^{CS-BA}	REN^{IF-BA}	TR_{CS-BA}^{IF-BA}	REN^{IF-GA}	TR_{CS-GA}^{IF-GA}
1	5	4.64e−15	8.33e−15	1.57	3.31e−07	1.68
	10	7.74e−15	9.25e−15	1.48	4.73e−09	1.78
	20	2.40e−15	5.02e−15	1.45	8.71e−09	2.06
	5	4.50e−15	5.12e−14	1.55	3.31e−07	1.63
	10	7.74e−15	5.01e−14	1.56	4.73e−09	1.86
	20	2.49e−15	4.68e−14	1.54	8.72e−09	2.14
	5	4.50e−15	2.59e−13	1.50	3.31e−07	1.54
	10	7.74e−15	2.55e−13	1.52	4.73e−09	1.73
	20	2.40e−15	2.51e−13	1.48	8.72e−09	1.88
	5	4.50e−15	3.13e−12	1.52	3.31e−07	1.52
	10	7.74e−15	3.10e−12	1.40	4.73e−09	1.58
	20	2.40e−15	3.05e−12	1.55	8.72e−09	2.01
	5	4.50e−15	3.72e−11	1.49	3.31e−07	1.53
	10	7.74e−15	3.69e−11	1.20	4.73e−09	1.51
	20	2.40e−15	3.65e−11	1.56	8.72e−09	2.03

The analysis of the results reported in the previous tables and those of other experiments that are not reported here shows that the CS-BA method takes less time than IF-BA or IF-GA methods. In some cases, the time ratio can reach or exceed two. The relative errors for the CS-BA method remains satisfactory even when the matrix A and the observation interval $[0, T]$ are very large. Some loss of precision is observed in the two other methods, especially, when th parameters n, r and T become more and more large, see for instance Table 11 with $n = 20229, r = 20$ and $T = 5000$.

6.5 Experiment 5

In this last experiment, we basically check Proposition 3 and Theorem 3. We show how numerically the approximation error for the solution of the algebraic equation will affect the overall approximation error for the solution of the differential equation. For this end, we consider the final time interval $T = 2000$ and the Sylvester differential equation with matrix coefficients $A = \mathbf{rail_5177}$ and $B = \mathbf{rail_1357}$, the right-hand side matrix is $C = U_A E_1 F_1^T (U_B)^T$ where U_A and U_B are the matrices obtained in the partial-rank spectral decomposition of A and B^T as explained in subsection 5.2.1. Thus, the algebraic error is $\tilde{E}_m = \tilde{X}^* - \tilde{X}_m = U_A (Q \odot (E_1 F_1^T)) (U_B)^T - \mathbb{V}_m^A Y_m$, where Y_m is the desired approximate solution of the projected Sylvester equation. In addition, using (42), the differential error at the time $t = t_k$ is $E_m(t_k) = X^*(t_k) - X_m(t_k) = U_A (H(t_k) \odot (E_1 F_1^T)) (U_B)^T - \mathbb{V}_m^A Y_{m,k}$, where $Y_{m,k}$ is the approximation at $t = t_k$ of the projected differential equation. According to Proposition 3, we should

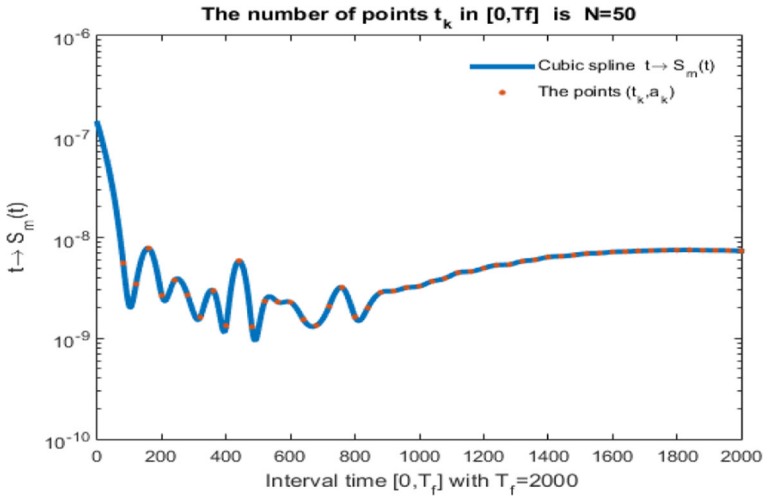


Fig. 3 Experiment 5: Curve illustrating the inequality given in Proposition 3

have

$$\|E_m(t)\| \leq (1 + e^{t(\|A\|+\|B\|)})\|\tilde{E}_m\|, \quad \forall t \in [0, T].$$

In Fig. 3, we plot the points $M_k(t_k, a_k)$ for $k = 0, \dots, N = 50$, where

$$a_k = \frac{\|E_m(t_k)\|}{(1 + e^{t_k(\|A\|+\|B\|)})\|\tilde{E}_m\|}.$$

We also plot the curve of the cubic spline interpolating the points M_k . We observe that, in fact the inequality given in the Proposition 3 is too large and is not optimal. In this test example, the plot in Fig. 3 indicates also that there is a positive constant $0 < C_1 < 10^{-6}$ such that the following inequality holds

$$\|E_m\|_\infty \leq C_1 (1 + e^{T(\|A\|+\|B\|)})\|\tilde{E}_m\|.$$

In this experiment, we may also illustrate Theorem 3, which is more precise than Proposition 3 and which states that we should have

$$\|E_m(t)\| \leq \left(\frac{e^{t(\|A\|+\|B\|)} - 1}{\|A\| + \|B\|} \right) \sqrt{(r_m^A + \|z_m^A(t)\|)^2 + (r_m^B + \|z_m^B(t)\|)^2}, \quad \forall t \in [0, T],$$

In Fig. 4, we plot the points $M_k(t_k, b_k)$ and the curve of the cubic spline interpolating the points M_k for $k = 1, \dots, N = 50$, where

$$b_k = \frac{\|E_m(t_k)\| (\|A\| + \|B\|)}{(e^{t_k(\|A\|+\|B\|)} - 1) \sqrt{(r_m^A + \|z_m^A(t_k)\|)^2 + (r_m^B + \|z_m^B(t_k)\|)^2}}.$$

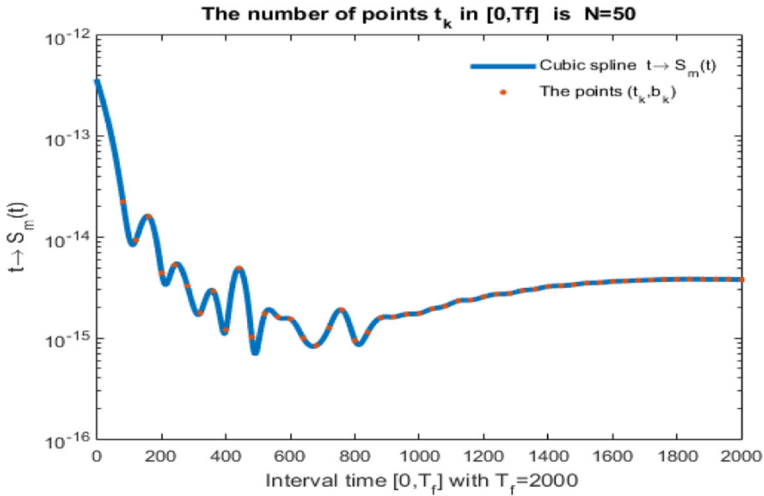


Fig. 4 Experiment 5: Curve illustrating the inequality given in Theorem 3

In this test example, the plot in Fig. 4 indicates also that there is a positive constant $0 < C_2 < 10^{-12}$ such that the following inequality holds

$$\|E_m\|_\infty \leq C_2 \left(\frac{e^{\Delta T (\|A\| + \|B\|)} - 1}{\|A\| + \|B\|} \right) \sqrt{(r_m^A + \|z_m^A\|_\infty)^2 + (r_m^B + \|z_m^B\|_\infty)^2}.$$

7 Conclusion

In this work, we proposed a new method for solving differential Sylvester and Lyapunov matrix equations. Unlike the recent methods proposed in [21, 38], our method avoids the integral formula, which is very benefit since its allows to reduce the computational cost. The constant solution method for solving differential Sylvester equations is related to the solution of the corresponding algebraic equation. As Krylov subspace methods are a good tool for the approximation of the exponential of matrices as well as for the solution of algebraic Sylvester (or Lyapunov) matrix equations, it can be seen that the constant solution method combined with Krylov projection methods is well suited for the solution of Sylvester (or Lyapunov) differential equations. The drawback of our method lies in the necessity to satisfy the condition $\sigma(A) \cap \sigma(-B) = \emptyset$. The robustness and efficiency of the proposed method have been observed on many numerical examples including reference examples that we have built. The convergence of such method is proved and constructive benchmark examples are given. The proposed method is very efficient for large-scale problems by exploiting projection techniques on Krylov subspaces. Numerous numerical tests are used to show the effectiveness of such proposed method, we have reported some of them in a specific section.

Acknowledgements We would like to thank the anonymous reviewer for his comments, criticisms and suggestions that helped us improve this manuscript.

Author contribution The three authors contributed equally to this work.

Data availability No data sets were generated or analyzed during the current work

Declarations

Competing interests The authors declare no competing interests.

References

1. Abou-Kandil, H., Freiling, G., Ionescu, V., Jank, G.: Matrix Riccati equations in control and systems theory, vol. 3. Birkhauser, Basel, Switzerland (2003)
2. Amato, F., Ambrosino, R., Ariola, M., Cosentino, C., De Tommasi, G.: Finite-time stability and control. Springer, (2014)
3. Antoulas, A.C.: Approximation of large-scale dynamical systems, vol. 6. Adv. Des. Control. SIAM Publications, Philadelphia, PA (2005)
4. Bartels, R.H., Stewart, G.W.: Solution of the matrix equation $AX + XB = C$. Commun. ACM **15**(9), 820–826 (1972)
5. Behr, M., Benner, P., Heiland, J.: Solution formulas for differential Sylvester and Lyapunov equations. Calcolo, p 56–51, (2019)
6. Bellman, R.: Introduction to matrix analysis, volume 6. SIAM publications, Philadelphia, PA, 2nd edition, 1997, First edition published by McGraw-Hill in (1970)
7. Benner, P., Mena, H.: BDF methods for large-scale differential Riccati equations. **01** (2009)
8. Benner, P., Mena, H.: Rosenbrock methods for solving Riccati differential equations. IEEE Transactions on Automatic Control **58**(11), 2950–2956 (2013)
9. Benner, P., Saak, J.: A semi-discretized heat transfer model for optimal cooling of steel profiles., volume vol. 45. Lecture Notes of Computer Science and Engineering, Springer Berlin, (2005)
10. Bouhamidi, A., Hached, M., Heyouni, M., Jbilou, K.: A preconditioned block Arnoldi method for large Sylvester matrix equations. Numerical Linear Algebra with Applications **20**(2), 208–219 (2013)
11. Bouhamidi, A., Jbilou, K.: Sylvester Tikhonov-regularization methods in image restoration. J. Comput. Appl. Math. **206**(1), 86–98 (2007)
12. Bouhamidi, A., Jbilou, K.: A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications. Appl. Math. Comput. **206**(2), 687–694 (2008)
13. Curtiss, C.F., Hirschfelder, J.O.: Integration of stiff equations. volume 38, pages 235–243 (1952). National Academy of Sciences of the United States of America
14. Datta, B.N.: Numerical methods for linear control systems. Academic Press, USA (2004)
15. Davis, T.A., Hu, Y.: The University of Florida Sparse Matrix Collection. ACM Trans. Math. Softw., **38**(1), (2011)
16. Davison, E.: The numerical solution of $X = A_1X + XA_2 + D$, $X(0) = C$. IEEE Trans. Autom. Control **20**(4), 566–567 (1975)
17. El Guennouni, A., Jbilou, K., Riquet, A.: Block Krylov subspace methods for solving large Sylvester equations. Numerical Algorithms **29**, 75–96 (2002)
18. Elbouyahyaoui, L., Heyouni, M., Jbilou, K., Messaoudi, A.: A block Arnoldi method for the solution of the Sylvester-Observer equation. Electronic Transactions on Numerical Analysis **47**, 18–36 (2017)
19. Golub, G., Nash, S., Loan, C.V.: A Hessenberg-Schur method for the problem $AX + XB = C$. IEEE Transactions on Automatic Control **24**(6), 909–913 (1979)
20. Hached, M., Jbilou, K.: Numerical solutions to large-scale differential Lyapunov matrix equations. Numerical Algorithms **79**, 741–757 (2017)
21. Hached, M., Jbilou, K.: Computational Krylov-based methods for large-scale differential Sylvester matrix problems. Numerical Linear Algebra with Applications **25**(5), e2187 (2018)

22. Hammarling, S.J.: Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA Journal of Numerical Analysis* **2**(3), 303–323 (1982)
23. Heyouni, M.: Extended Arnoldi methods for large low-rank Sylvester matrix equations. *Applied Numerical Mathematics* **60**(11), 1171–1182 (2010)
24. Heyouni, M., Jbilou, K.: An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation. *Electronic Transactions on Numerical Analysis* **33**, 53–62 (2009)
25. Higham, N.J.: The scaling and squaring method for the matrix exponential revised. *SIAM J. Matrix Anal Appl* **26**(4), 1179–1193 (2005)
26. Horn, R.A., Johnson, C.R.: *Topics in matrix analysis*, volume, 2nd edn. Cambridge University Press, UK (1994)
27. Hu, D., Reichel, L.: Krylov-subspace methods for the Sylvester equation. *Linear Algebra and its Applications* **172**, 283–313 (1992)
28. Jaimoukha, I.M., Kasenally, E.M.: Krylov subspace methods for solving large Lyapunov equations. *SIAM Journal on Numerical Analysis* **31**(1), 227–251 (1994)
29. Jbilou, K.: ADI preconditioned Krylov methods for large Lyapunov matrix equations. *Linear Algebra and its Applications* **432**(10), 2473–2485 (2010)
30. Konstantinov, M., Gu, D.-W., Mehrmann, V., Petkov, P.: *Perturbation Theory for Matrix Equations* **49**, 11 (2004)
31. Kressner, D.: Block variants of Hammarling’s method for solving Lyapunov equations. *ACM Trans. Math. Softw.*, **34**(1), (2008)
32. Lancaster, P.: Explicit solutions of linear matrix equations. *SIAM Review* **12**, 544–566 (1970)
33. Moler, C., Loan, C.V.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* **45**(1), 3–000 (2003)
34. Penzl, T.: *LYAPACK. A MATLAB Toolbox for large Lyapunov and Riccati equations, Model Reduction Problems, and Linear-Quadratic Optimal Control Problems*, (2000)
35. Saad, Y.: Numerical solution of large Lyapunov equations. In: *Signal Processing, Scattering and Operator Theory, and Numerical Methods*, Proc. MTNS-89, p 503–511, (1990), Birkhauser
36. Saad, Y.: Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.* **29**(1), 209–228 (1992)
37. Saad, Y.: *xIterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, second edition, (2003)
38. Sadek, E.M., Bentbib, A., Sadek, L., Alaoui, H.: Global extended Krylov subspace methods for large-scale differential Sylvester matrix equations. *J Appl Math Comput* **62**, 165–177 (2019)
39. Simoncini, V.: A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM Journal on Scientific Computing* **29**(3), 1268–1288 (2007)
40. Simoncini, V.: Computational methods for linear matrix equations. *SIAM Review* **58**(3), 377–441 (2016)
41. Sorensen, D.C., Zhou, Y.: Direct methods for matrix Sylvester and Lyapunov equations. *Journal of Applied Mathematics* **2003**(6), 277–303 (2003)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.