



HAL
open science

THE CONSTANT SOLUTION METHOD FOR SOLVING LARGE SCALE DIFFERENTIAL SYLVESTER MATRIX EQUATION WITH TIME INVARIANT COEFFICIENTS

Abderrahman Bouhamidi, Mohammed Heyouni, Lakhdar Elbouyahyaoui

► **To cite this version:**

Abderrahman Bouhamidi, Mohammed Heyouni, Lakhdar Elbouyahyaoui. THE CONSTANT SOLUTION METHOD FOR SOLVING LARGE SCALE DIFFERENTIAL SYLVESTER MATRIX EQUATION WITH TIME INVARIANT COEFFICIENTS. 2023. hal-04221763

HAL Id: hal-04221763

<https://hal.science/hal-04221763v1>

Preprint submitted on 28 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

39 Adaptation of BDF and/or Rosenbrock methods has been described in [7, 8] (see
 40 also the references in [5]). However these adaptations usually suffer from a prob-
 41 lem of numerical data storage. To remedy this problem, combining Krylov subspaces
 42 techniques with BDF methods or with Taylor series expansions have recently been
 43 proposed [5, 19]. Other existing methods described in the recent literature for solving
 44 large-scale differential Sylvester matrix equation rely on using the integral formula or
 45 some numerical ODE solver [20, 37]. The strategy we pursue in this manuscript is dif-
 46 ferent in the sense that our approach for solving differential Sylvester (or Lyapunov)
 47 matrix equations is based on the use of the constant solution to the differential equa-
 48 tion. The first result we use indicates that the solution of the differential equation is
 49 written in terms of the solution of the corresponding algebraic equation. Additionally,
 50 in the case where the coefficient matrices A and/or B are large, we combine the new
 51 expression of the solution with some projection techniques on Krylov subspace, such
 52 as the block Arnoldi algorithm for solving the corresponding algebraic equations or
 53 for approximating the exponential of a matrix.

54 we momentarily assume that the matrix C in (1.1), represents a continuous ma-
 55 trix function $C(t)$ defined on $[t_0, T]$ and let us consider next, the following classical
 56 differential linear system

$$57 \quad (1.3) \quad \begin{cases} \dot{x}(t) = \mathbb{A} x(t) - c(t), \\ x(t_0) = x_0, \end{cases}$$

58 where $\mathbb{A} \in \mathbb{R}^{p \times p}$ is time independent and $x(t), c(t) \in \mathbb{R}^p$ for all time $t \in [t_0, T]$. We
 59 assume that c is a continuous function on the interval $[t_0, T]$. In this case, the system
 60 (1.3) has a unique solution $x(t)$ which is differentiable with a continuous derivative
 61 on $[t_0, T]$. It is well known that the unique solution of (1.3) may be written under
 62 the form

$$63 \quad (1.4) \quad x(t) = e^{(t-t_0)\mathbb{A}} x_0 - \int_{t_0}^t e^{(t-u)\mathbb{A}} c(u) du.$$

64 In many practical situations, the matrix \mathbb{A} may be very large. In this case, exploiting
 65 expression (1.4) for computing the solution $x(t)$ is very expensive. However, when
 66 the matrix \mathbb{A} is large and sparse with a specific structure, some numerical techniques
 67 may be done to approximate the matrix exponential time a vector [24, 32, 35]. As, our
 68 interest is to obtain a good approximate solution to the differential Sylvester equation
 69 (1.1), we need to consider the matrix \mathbb{A} in system (1.3) of size $p = ns$ structured as
 70 following:

$$71 \quad \mathbb{A} = I_s \otimes A + B^T \otimes I_n,$$

72 where the matrices A, B are those given in (1.1). The identity matrices I_n, I_s are of
 73 size n and s , respectively. The notations B^T and \otimes stand for the transpose of B
 74 and the Kronecker product of two matrices, respectively. We recall that the Kronecker
 75 product of two matrices J and K of size $n_j \times m_j$ and $n_k \times m_k$, respectively, is the
 76 matrix $J \otimes K = [J_{i,j} K]$ of size $n_j n_k \times m_j m_k$. The following well known properties
 77 will be used throughout this paper:

- 78 1. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$,
- 79 2. $(A \otimes B)^T = (A^T \otimes B^T)$,
- 80 3. $vec(ABC) = (C^T \otimes A) vec(B)$.

81 The vec operator consists in transforming a matrix into a vector by stacking its
 82 columns one by one to form a single column vector.

83 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
 84 $X_0, X(t)$ and C are appearing in (1.1). In [1], it was written, without any specific
 85 details of its proof, that the solution of the differential Sylvester matrix equation (1.1)
 86 is given by the following integral formula

$$87 \quad (1.5) \quad X(t) = e^{(t-t_0)A} X_0 e^{(t-t_0)B} - \int_{t_0}^t e^{(t-u)A} C(u) e^{(t-u)B} du.$$

Although the proof of this result does not present any major difficulty, it seemed interesting to us to give the details of such a proof. Indeed, using the additive commutativity of the matrix exponential which states

$$e^A e^B = e^{A+B} \iff AB = BA,$$

88 and since, the matrices $I_s \otimes A$ and $B^T \otimes I_n$ commute, then using the properties of
 89 the Kronecker product, it follows that

$$\begin{aligned} 90 \quad e^{(t-t_0) \mathbb{A}} &= e^{(t-t_0) (I_s \otimes A + B^T \otimes I_n)} \\ 91 &= e^{(t-t_0) (I_s \otimes A)} e^{(t-t_0) (B^T \otimes I_n)} \\ 92 &= (I_s \otimes e^{(t-t_0)A}) (e^{(t-t_0)B^T} \otimes I_n) \\ 93 &= e^{(t-t_0)B^T} \otimes e^{(t-t_0)A}. \end{aligned}$$

Thus,

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

94 Finally, this implies that the formula (1.4) giving the solution to the system (1.3)
 95 leads to the formula (1.5) giving the solution to the differential Sylvester equation
 96 (1.1).

97 From now on, return back to the case where the vector coefficient c in (1.3) and
 98 also the matrix coefficient C in (1.1) are assumed to be constant functions on the
 99 interval $[t_0, T]$. Thus, as the systems (1.1) and (1.3) are mathematically equivalent,
 100 then for moderate size problems, it is possible to apply directly a numerical integration
 101 to the system (1.3) or to use (1.4) or the above integral formula (1.5). But, for a large
 102 system, we will see that it is more interesting to solve the system in the matrix form
 103 (1.1).

104 Note that it is not restrictive to choose $X_0 = 0$ in the initial condition $X(t_0) = X_0$
 105 of (1.1). Indeed, let $X(t)$ denote the exact solution of the system (1.1), then the matrix
 106 function given by $Y(t) = X(t) - X_0$ is the unique solution of the following system

$$107 \quad \begin{cases} \dot{Y}(t) = AY(t) + Y(t)B - C_0, & t \in [t_0, T] \\ Y(t_0) = 0, \end{cases}$$

where

$$C_0 = C - (AX_0 + X_0B).$$

108 We may first solve the previous differential equation to get $Y(t)$ and then deduce the
 109 solution $X(t) = Y(t) + X_0$ of the differential equation (1.1).

To end this section, we give some useful notations. 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 matrix M . The associated Frobenius norm is denoted by $\|Y\| = \sqrt{\langle Y, Y \rangle}$. For a bounded matrix valued function G defined on the interval $[t_0, T]$, we consider the following uniform convergence norm

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

The outline of this paper is as follows: In Section 2, we introduce our proposed method which will be 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 Section 3, 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. In section 4 which is devoted to numerical experiments, we first show how to generate a benchmark differential Sylvester matrix equation with a known exact solution. To the best of our knowledge, this construction is new and has never been proposed before. The numerical results section continues with several set of experiments whose results indicate that CSM is an efficient and robust method. As usual, the last section is devoted to a brief conclusion.

2. The constant solution method for the differential Sylvester matrix equation. In the integral Formula (1.5), 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 (1.1) which is given in terms of the solution of the corresponding algebraic Sylvester matrix equation (1.2). 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 (1.1). The result of this theorem is known in the literature [6, 15], 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 2.1. *Suppose that the matrices A and B in the system (1.1) 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 (1.1) is given by*

$$(2.1) \quad X^*(t) = e^{(t-t_0)A} \left(X_0 - \tilde{X}^* \right) e^{(t-t_0)B} + \tilde{X}^*,$$

where \tilde{X}^* is the unique and exact solution of the algebraic Sylvester equation (1.2).

Proof. The general solution of the homogeneous differential equation associated to (1.1) is given by

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

149 where $Y \in \mathbb{R}^{n \times s}$ is some constant matrix. Since $\sigma(A) \cap \sigma(-B) = \emptyset$, the algebraic
150 Sylvester equation (1.2) has a unique solution \tilde{X}^* (see, e.g [25, Thm. 2.4.4.1]). Now,
151 as \tilde{X}^* is a constant matrix function (i.e., $\tilde{X}^*(t) = \tilde{X}^*$), then it can be seen as a
152 particular solution of the differential equation

$$153 \quad \dot{X}(t) = A X(t) + X(t) B - C.$$

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

$$155 \quad X(t) = e^{tA} Y e^{tB} + \tilde{X}^*.$$

156 Finally, since the unique solution of the differential equation (1.1) must satisfy the
157 initial condition $X^*(t_0) = X_0$, it follows that $X^*(t_0) = e^{t_0 A} Y e^{t_0 B} + \tilde{X}^* = X_0$.
158 The last equality implies $Y = e^{-t_0 A} (X_0 - \tilde{X}^*) e^{-t_0 B}$ and expression (2.1) follows
159 immediately. \square

160 In the remainder of this paper, we assume that the matrices A and B in (1.1)
161 satisfy the condition

$$162 \quad \sigma(A) \cap \sigma(-B) = \emptyset.$$

163 The following property shows the behavior of the matrix solution $X^*(t)$ as the interval
164 $[t_0, T]$ becomes very more and more large, namely, as the final time T goes to $+\infty$.

PROPOSITION 2.2. [29, Chapter 8] *Suppose that the coefficients A and B in the system (1.1) are stable matrices, then the unique solution $X^*(t)$ of the differential system (1.1) satisfies*

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

165 where \tilde{X}^* is the unique solution of the corresponding algebraic Sylvester equation (1.2).

166 In the remainder of this section, we suppose that the matrix coefficients A and B
167 are of moderate size. In this case, an approximate solution to the algebraic Sylvester
168 equation (1.2) may be obtained by a direct solver such as the Bartels-Stewart al-
169 gorithm, the Schur decomposition, or the Hammarling method [4, 18, 21, 30, 40]. A
170 common point to all these methods is first the computation of the real Schur forms
171 of the coefficient matrices using the QR algorithm. Then, the original equation is
172 transformed into an equivalent form that is easier to solve by a forward substitution.
173 Now, suppose that \tilde{X}_a is an approximate solution to \tilde{X}^* the exact solution of the
174 Sylvester algebraic equation (1.2), it follows that an approximate solution $X_a(t)$ to
175 $X^*(t)$ the exact solution of the Sylvester differential equation (1.1) can be expressed
176 in the following form.

$$177 \quad (2.2) \quad X_a(t) = e^{(t-t_0)A} (X_0 - \tilde{X}_a) e^{(t-t_0)B} + \tilde{X}_a.$$

178 Here, as A and B are assumed to be of moderate size, we also assume that both
179 exponential $e^{(t-t_0)A}$ and $e^{(t-t_0)B}$ are computed exactly. To establish an upper bound

180 for the error norm, let us introduce the algebraic error \tilde{E} and the differential error
181 $E(t)$ given by

$$182 \quad \tilde{E} = \tilde{X}^* - \tilde{X}_a, \quad \text{and} \quad E(t) = X^*(t) - X_a(t), \quad \forall t \in [t_0, T],$$

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

184 have the following result

185 **PROPOSITION 2.3.** *In the case where the matrix exponential is computed exactly,*
186 *we have*

$$187 \quad \|E\|_\infty \leq \left(1 + e^{\Delta_T(\|A\| + \|B\|)}\right) \|\tilde{E}\|,$$

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

Proof. Subtracting (2.2) from (2.1), 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 $\|AB\| \leq \|A\| \|B\|$), this implies that $\|e^{sM}\| \leq e^{s\|M\|}$ for all $s \geq 0$ and for any square matrix M . Thus,

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

190 As E is a continuous matrix function on the interval $[t_0, T]$, $(t - t_0) \leq \Delta_T$ and
191 $\|E\|_\infty = \sup_{t \in [t_0, T]} \|E(t)\|$, then the desired result follows obviously. \square

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

$$194 \quad (2.3) \quad \begin{cases} R(t) &= \dot{X}_a(t) - (A X_a(t) + X_a(t) B - C), \quad t \in [t_0, T], \\ \tilde{R} &= C - (A \tilde{X}_a + \tilde{X}_a B), \end{cases}$$

195 and satisfy the following proposition.

PROPOSITION 2.4. *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 (2.2), we have

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

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

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

196

□

197

198

199

200

Before ending this section, we sketch in Algorithm 2.1 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 (1.1) at different nodes t_k , ($k = 1, \dots, N$) of a suitable discretization of the time interval $[t_0, T]$.

Algorithm 2.1 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**
-

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

3. 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 (2.1) 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 (1.2), 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 [9–11, 16, 22, 23, 26–28]. 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, 18], 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, 13, 39] 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 [19, 20] by exploiting the integral formula (1.5) 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) [12].

229 As said before, the approach we follow in this work is different from the one
 230 proposed in [19, 20]. It consists of exploiting formula (2.1), instead of the integral
 231 formula (1.5), which is less expensive. To have at hand an adequate basis of the
 232 considered Krylov subspace, we will use the block Arnoldi process described in the
 233 next subsection.

234 **3.1. The block Arnoldi process.** Let M be an $l \times l$ matrix and V an $l \times s$
 235 block vector. We consider the classical block Krylov subspace

$$236 \quad \mathbb{K}_m(M, V) = \text{Range}([V, M V, \dots, M^{m-1} V])$$

$$237 \quad = \left\{ \sum_{k=0}^{m-1} M^k V \Omega_k, \Omega_k \in \mathbb{R}^{s \times s}, 0 \leq k \leq m-1 \right\}.$$

238 The block Arnoldi process, described in Algorithm 3.1, generates an orthonormal basis
 239 \mathbb{V}_m^M of the block Krylov subspace $\mathbb{K}_m(M, V)$.

Algorithm 3.1 The block Arnoldi process (BA)

- 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 $\overline{\mathbb{H}}_m^M$ satisfying (3.1)–(3.3).
 - 3: Get V_1 by computing the QR decomposition of V , *i.e.*, $V = V_1 \Lambda_1$;
 - 4: **for** $j = 1, \dots, m$ **do**
 - 5: Compute $U = M V_j$;
 - 6: **for** $i = 1, 2, \dots, j$ **do**
 - 7: $H_{i,j} = V_i^T U$;
 - 8: $U = U - V_i H_{i,j}$;
 - 9: **end for**
 - 10: Get V_{j+1} and $H_{j+1,j}$ by computing the QR decomposition of U ,
i.e., $U = V_{j+1} H_{j+1,j}$.
 - 11: Set $H_{i,j} = 0$ for $i > j+1$
 - 12: Define $\mathbb{V}_{j+1}^M = [V_1, \dots, V_j, V_{j+1}]$ and $\overline{\mathbb{H}}_j^M = (H_{k,\ell})_{1 \leq k \leq j+1, 1 \leq \ell \leq j}$
 - 13: **end for**
-

240 Suppose that the upper triangular matrices $H_{j+1,j}$ are full rank then, since the
 241 above algorithm involves a Gram-Schmidt procedure, the obtained block vectors
 242 V_1, V_2, \dots, V_m ($V_i \in \mathbb{R}^{l \times s}$) have their columns mutually orthogonal. Hence, after
 243 m steps, Algorithm 3.1 generates an orthonormal basis $\mathbb{V}_m^M = [V_1, V_2, \dots, V_m]$ of the
 244 block Krylov subspace $\mathbb{K}_m(M, V)$ and a block upper Hessenberg matrix \mathbb{H}_m^M whose
 245 non zeros blocks are the $H_{i,j} \in \mathbb{R}^{s \times s}$. We have the following and useful algebraic
 246 relations [17, 36].

$$247 \quad (3.1) \quad M \mathbb{V}_m^M = \mathbb{V}_{m+1}^M \overline{\mathbb{H}}_m^M = \mathbb{V}_m^M \mathbb{H}_m^M + V_{m+1}^M H_{m+1,m}^M (\mathbb{E}_m^{(s)})^T,$$

$$248 \quad (3.2) \quad (\mathbb{V}_m^M)^T M \mathbb{V}_m^M = \mathbb{H}_m^M,$$

$$249 \quad (3.3) \quad (\mathbb{V}_m^M)^T \mathbb{V}_m^M = I_{m s},$$

250 where $\overline{\mathbb{H}}_m^M = (\mathbb{V}_{m+1}^M)^T M \mathbb{V}_m^M \in \mathbb{R}^{(m+1)s \times m s}$, $H_{i,j} \in \mathbb{R}^{s \times s}$ is the (i, j) block of \mathbb{H}_m^M
 251 and $\mathbb{E}_m^{(s)}$ is the matrix of the last s columns of the $m s \times m s$ identity matrix $I_{m s}$, *i.e.*
 252 $\mathbb{E}_m^{(s)} = [0_{s \times (m-1)s}, I_s]^T$. In the following, we will use the notation

$$253 \quad (3.4) \quad V_{m,s}^M = V_{m+1}^M H_{m+1,m}^M (\mathbb{E}_m^{(s)})^T.$$

254 **3.2. Full rank case.** Here, we suppose that A is a large matrix while B is
 255 relatively smaller, i.e., $s \ll n$. We also assume that the right-hand side C is full rank,
 256 i.e. $\text{rank}(C) = s$. Also, for simplicity reasons, we took $X_0 = 0$ in the initial condition,
 257 since, as mentioned in the introduction, it is not restrictive.

258 To obtain approximate solutions to the algebraic Sylvester equation (1.2), one
 259 can use the block Arnoldi method in which we consider approximate solutions that
 260 have the following form

$$261 \quad (3.5) \quad \tilde{X}_m = \mathbb{V}_m^A \tilde{Y}_m,$$

262 where \mathbb{V}_m^A is the orthonormal Krylov basis generated by applying m iterations of
 263 Algorithm 3.1 to the pair (A, C) . Let \tilde{R}_m be the algebraic residual given by

$$264 \quad (3.6) \quad \tilde{R}_m = C - (A \tilde{X}_m + \tilde{X}_m B).$$

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

$$266 \quad (\mathbb{V}_m^A)^T \tilde{R}_m = 0_{m \times s}.$$

267 Thus, taking into account the relations (3.1)–(3.3) and (3.5), it follows that \tilde{Y}_m is the
 268 solution of the reduced Sylvester equation

$$269 \quad \mathbb{H}_m^A Y + Y B = C_m,$$

270 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 3.1, we
 271 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
 272 equation admits a unique solution which can be obtained by a direct method [4, 18]. In
 273 addition, from the relations (3.1)–(3.3), the residual \tilde{R}_m satisfies the following relation

$$274 \quad (3.7) \quad \tilde{R}_m = -V_{m,s}^A \tilde{Y}_m.$$

275 According to [32, 34], the following approximation to $e^{(t-t_0)A} \tilde{X}^*$ holds

$$276 \quad e^{(t-t_0)A} \tilde{X}^* \simeq \mathbb{V}_m^A e^{(t-t_0) \mathbb{H}_m^A} (\mathbb{V}_m^A)^T \tilde{X}_m.$$

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 (1.1) may be obtained by

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

277 Taking into account (3.5), it follows that

$$278 \quad (3.8) \quad X_m(t) = \mathbb{V}_m^A Y_m(t), \quad t \in [t_0, T],$$

279 where

$$280 \quad (3.9) \quad Y_m(t) = -e^{(t-t_0) \mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0)B} + \tilde{Y}_m, \quad t \in [t_0, T].$$

281 This matrix function satisfies the following result.

282 **PROPOSITION 3.1.** *The matrix function $Y_m(t)$ given by (3.9) is the unique solu-*
 283 *tion of the reduced differential Sylvester matrix equation*

$$284 \quad (3.10) \quad \begin{cases} \dot{Y}(t) &= \mathbb{H}_m^A Y(t) + Y(t) B - C_m, & t \in [t_0, T] \\ Y(t_0) &= 0. \end{cases}$$

Proof. The derivative of the matrix function $Y_m(t)$ as given by (3.9) is

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

285 On the other hand, we have

286

$$\begin{aligned} 287 \quad \mathbb{H}_m^A Y_m(t) + Y_m(t) B - C_m &= -e^{(t-t_0) \mathbb{H}_m^A} \left(\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m B \right) e^{(t-t_0) B} \\ 288 \quad &+ \mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m B - C_m, \quad t \in [t_0, T]. \end{aligned}$$

Thus, it follows that

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

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

291

Remark 3.2. Proposition 3.1 shows that another way to obtain an approximation
292 of $Y_m(t)$ can be the resolution of the projected and reduced differential equation (3.9)
293 by using an adequate numerical ODE solver such as Runge-Kutta or BDF solvers.
294 We recall that such technique was used in [19, 20]. In our proposed method, we don't
295 use such approach, but instead, we solve the reduced algebraic equation and take into
296 account approximations (3.8) and (3.10) to get an approximate solution to the low
297 dimensional differential matrix equation (3.9).

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

$$299 \quad (3.11) \quad R_m(t) = \dot{X}_m(t) - (A X_m(t) + X_m(t) B - C), \quad t \in [t_0, T].$$

300 The following proposition gives an expression for this residual.

301 PROPOSITION 3.3. *The residual for the differential equation is given by*

$$302 \quad (3.12) \quad R_m(t) = -V_{m,s}^A Y_m(t)$$

$$303 \quad (3.13) \quad = V_{m,s}^A e^{(t-t_0) \mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0) B} + \tilde{R}_m,$$

304 where \tilde{R}_m is the algebraic residual given in (3.6). Moreover

$$305 \quad (3.14) \quad (\mathbb{V}_m^A)^T R_m(t) = 0_{m \times s}, \quad \forall t \in [t_0, T].$$

Proof. Replacing, in (3.11), $X_m(t)$ by its expression given by (3.8), we get

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

306 Then, using (3.1), we obtain

$$307 \quad R_m(t) = \mathbb{V}_m^A \dot{Y}_m(t) - \left(\mathbb{V}_m^A \mathbb{H}_m^A + V_{m,s}^A \right) Y_m(t) - \mathbb{V}_m^A Y_m(t) B + \mathbb{V}_m^A C_m.$$

$$308 \quad = \mathbb{V}_m^A \left(\dot{Y}_m(t) - \left[\mathbb{H}_m^A Y_m(t) + Y_m(t) B - C_m \right] \right) - V_{m,s}^A Y_m(t).$$

As $Y_m(t)$ is the solution of (3.10), we then get (3.12). Now, according to (3.9), we obtain

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

309 Finally, from (3.7), we get (3.13) and the relation (3.14) follows immediately. □

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,

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

310 Then, using (3.13), we get

$$311 \quad \lim_{T \rightarrow +\infty} R_m(T) = \tilde{R}_m.$$

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

$$314 \quad E_m(t) = X^*(t) - X_m(t), \quad \forall t \in [t_0, T].$$

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

THEOREM 3.4. *At step m , the following error estimate holds*

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

318 where

$$319 \quad r_m^A = \|\tilde{R}_m\| = \|H_{m+1,m}^A \bar{Y}_m\| \quad \text{and} \quad z_m^A = \|H_{m+1,m}^A \bar{Z}_m\|,$$

320 with \bar{Y}_m, \bar{Z}_m are the matrices of size $s \times s$ formed by the s last rows of \tilde{Y}_m and
321 $Z_m := e^{\Delta T} \mathbb{H}_m^A \tilde{Y}_m e^{\Delta T B}$ respectively.

Proof. From (2.3) and the differential Sylvester matrix equation (1.1), 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

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

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 (3.13) and the triangular inequality, we get

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

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

323 Now, we point out that (3.12) provides a cheap formula for computing at each node
324 t_k the norm $r_{m,k} := \|R_m(t_k)\|$ of the residual associated to the approximate solution
325 $X_{m,k} := X_m(t_k)$. This formula avoids computing matrix vector products with the
326 large coefficient matrix A since we have

$$327 \quad (3.15) \quad r_{m,k} := \|R_m(t_k)\| = \|H_{m+1,m}^A (\mathbb{E}_m^{(s)})^T Y_{m,k}\| = \|H_{m+1,m}^A \bar{Y}_{m,k}\|,$$

328 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
329 matrix $Y_{m,k} := Y_m(t_k)$.

330 Finally, we end this section by summarizing in Algorithm 3.2 our proposed method
331 that is the block Arnoldi combined with the constant solution method (BA-CSM)
332 applied for full-rank differential Sylvester equations

333 **3.3. Low-rank case.** Now, we consider the case where both A and B are large
334 matrices. In addition, the coefficient C appearing in the right-hand side is assumed
335 to be low rank and is given under the factored form $C = E F^T$ where $E \in \mathbb{R}^{n \times r}$
336 and $F \in \mathbb{R}^{s \times r}$. We also, assume for simplicity reasons that the initial condition
337 is such that $X_0 = 0$. To obtain approximate solutions to the low-rank algebraic
338 Sylvester equation (1.2), we can use the block Arnoldi method in which we consider
339 approximate solutions that have the form

$$340 \quad (3.16) \quad \tilde{X}_m = \mathbb{V}_m^A \tilde{Y}_m (\mathbb{V}_m^B)^T,$$

341 where $\mathbb{V}_m^A, \mathbb{V}_m^B$ are the orthonormal matrices obtained by running m iterations of
342 Algorithm 3.1 applied to the pairs (A, E) and (B^T, F) respectively. Enforcing the
343 following Petrov-Galerkin condition

$$344 \quad (\mathbb{V}_m^A)^T \tilde{R}_m \mathbb{V}_m^B = 0_{mr \times mr},$$

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

$$346 \quad (3.17) \quad \tilde{R}_m = E F^T - (A \tilde{X}_m + \tilde{X}_m B).$$

347 Multiplying (3.17) on the left by $(\mathbb{V}_m^A)^T$ and on the right by \mathbb{V}_m^B and taking into
348 account relations (3.1)–(3.3) and (3.16), it follows immediately, that \tilde{Y}_m is the solution
349 of the reduced projected Sylvester equation

$$350 \quad (3.18) \quad \mathbb{H}_m^A Y + Y (\mathbb{H}_m^B)^T = E_m F_m^T,$$

351 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
352 Hessenberg matrices generated by the block Arnoldi process and $E_m = (\mathbb{V}_m^A)^T E$,

Algorithm 3.2 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 3.1 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 $r_{m,k} = \|\mathbb{H}_{m+1,m}^A \bar{Y}_{m,k}\|$.

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**

353 $F_m = (\mathbb{V}_m^B)^T F$. Note that from Algorithm 3.1, we also get that $E = \mathbb{V}_m^A E_m$ and
 354 $F = \mathbb{V}_m^B F_m$. Here also, if $\sigma(\mathbb{H}_m^A) \cap \sigma(-\mathbb{H}_m^B) = \emptyset$, then equation (3.18) admits a
 355 unique solution which can be computed using a standard direct method such as those
 356 described in [4,18]. Using the relation (3.1)–(3.2) and from the relations (3.16)–(3.18),
 357 we get

$$358 \quad (3.19) \quad \tilde{R}_m = -V_{m,r}^A \tilde{Y}_m (\mathbb{V}_m^B)^T - \mathbb{V}_m^A \tilde{Y}_m (V_{m,r}^B)^T,$$

359 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
 360 that, according to [32,34], an approximation to $e^{(t-t_0)A} \tilde{X}^* e^{(t-t_0)B}$ may be obtained
 361 as

$$362 \quad e^{(t-t_0)A} \tilde{X}^* e^{(t-t_0)B} \simeq \mathbb{V}_m^A e^{(t-t_0) \mathbb{H}_m^A} (\mathbb{V}_m^A)^T \tilde{X}_m \mathbb{V}_m^B e^{(t-t_0) (\mathbb{H}_m^B)^T} (\mathbb{V}_m^B)^T.$$

Then, it follows, that an approximate solution $X_m(t)$ to the exact solution $X^*(t)$ of the differential Sylvester matrix equation (1.1) may be given by

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

363 Taking into account (3.16) gives that

$$364 \quad (3.20) \quad X_m(t) = \mathbb{V}_m^A Y_m(t) (\mathbb{V}_m^B)^T, \quad t \in [t_0, T],$$

365 where

$$366 \quad (3.21) \quad Y_m(t) = -e^{(t-t_0) \mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0) (\mathbb{H}_m^B)^T} + \tilde{Y}_m, \quad t \in [t_0, T].$$

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

368 PROPOSITION 3.5. *The matrix function $Y_m(t)$ given by (3.21) is the unique solu-*
 369 *tion of the reduced differential Sylvester matrix equation*

$$370 \quad (3.22) \quad \begin{cases} \dot{Y}(t) = \mathbb{H}_m^A Y(t) + Y(t) (\mathbb{H}_m^B)^T - E_m F_m^T, & t \in [t_0, T] \\ Y(t_0) = 0. \end{cases}$$

Proof. The derivative of the matrix function $Y_m(t)$ as given by (3.21) is

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

371 On the other hand, we have

$$372 \quad \mathbb{H}_m^A Y_m(t) + Y_m(t) (\mathbb{H}_m^B)^T - E_m F_m^T =$$

$$373 \quad -e^{(t-t_0) \mathbb{H}_m^A} \left(\mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m (\mathbb{H}_m^B)^T \right) e^{(t-t_0) (\mathbb{H}_m^B)^T}$$

$$374 \quad + \mathbb{H}_m^A \tilde{Y}_m + \tilde{Y}_m (\mathbb{H}_m^B)^T - E_m F_m^T.$$

375

Thus, it follows that

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

377 Moreover, $Y_m(t)$ satisfies the initial condition $Y_m(t_0) = 0$. \square

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

$$380 \quad (3.23) \quad R_m(t) = \dot{X}_m(t) - (A X_m(t) + X_m(t) B - E F^T), \quad t \in [t_0, T].$$

381

382 PROPOSITION 3.6. *The residual for the differential equation is given by*

$$383 \quad (3.24) \quad R_m(t) = -V_{m,r}^A Y_m(t) (\mathbb{V}_m^B)^T - \mathbb{V}_m^A Y_m(t) (V_{m,r}^B)^T,$$

$$384 \quad (3.25) \quad = V_{m,r}^A \tilde{F}_m(t) (\mathbb{V}_m^B)^T + \mathbb{V}_m^A \tilde{F}_m(t) (V_{m,r}^B)^T + \tilde{R}_m,$$

385 where $\tilde{F}_m(t) = e^{(t-t_0) \mathbb{H}_m^A} \tilde{Y}_m e^{(t-t_0) (\mathbb{H}_m^B)^T}$ and \tilde{R}_m is the algebraic residual given by
 386 (3.17). In addition,

$$387 \quad (3.26) \quad \left(\mathbb{V}_m^A \right)^T R_m(t) \mathbb{V}_m^B = 0_{m_r \times m_r}, \quad \forall t \in [t_0, T].$$

Proof. Using the definition (3.23) of the residual $R_m(t)$ and replacing $X_m(t)$ by
 its expression given in (3.20), we get

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

388 Now, using the algebraic relation (3.1) in which M is replaced either by A or by B ,
 389 we obtain

$$390 \quad R_m(t) = \mathbb{V}_m^A \dot{Y}_m(t) (\mathbb{V}_m^B)^T - \left(\mathbb{V}_m^A \mathbb{H}_m^A + V_{m,r}^A \right) Y_m(t) (\mathbb{V}_m^B)^T$$

$$391 \quad - \mathbb{V}_m^A Y_m(t) \left((\mathbb{H}_m^B)^T (\mathbb{V}_m^B)^T + (V_{m,r}^B)^T \right) + \mathbb{V}_m^A E_m F_m^T (\mathbb{V}_m^B)^T.$$

392 This may be arranged as following

$$393 \quad R_m(t) = \mathbb{V}_m^A \left(\dot{Y}_m(t) - \mathbb{H}_m^A Y_m(t) - Y_m(t)(\mathbb{H}_m^B)^T + E_m F_m^T \right) (\mathbb{V}_m^B)^T$$

$$394 \quad -V_{m,r}^A Y_m(t) (\mathbb{V}_m^B)^T - \mathbb{V}_m^A Y_m(t) (V_{m,r}^B)^T.$$

395 Taking into account (3.22), we get (3.24). The relation (3.25) follows by replac-
396 ing $Y_m(t)$ by its expression (3.21) and taking into account (3.19). Finally, (3.26) is
397 straightforward since \mathbb{V}_m^A and \mathbb{V}_m^B are orthogonal matrices. \square

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

$$399 \quad \lim_{T \rightarrow +\infty} R_m(T) = \tilde{R}_m.$$

400 As in the previous subsection, we have the following error estimates.

THEOREM 3.7. *Let $X_m(t)$, for $t \in [t_0, T]$, be the approximate solution at a step m given by (3.20) and (3.21) and let $E_m(t) = X^*(t) - X_m(t)$ be the error. Then, we have the following error estimate:*

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

401 where $r_m^A = \|H_{m+1,m}^A \bar{Y}_m\|$, $r_m^B = \|H_{m+1,m}^B \bar{Y}_m\|$, $z_m^A = \|H_{m+1,m}^A \bar{Z}_m\|$ and $z_m^B =$
402 $\|H_{m+1,m}^B \bar{Z}_m\|$. The matrices \bar{Y}_m and \bar{Z}_m are of size $r \times r$ and formed by the r last
403 rows of \tilde{Y}_m and $Z_m := e^{\Delta t} \mathbb{H}_m^A \tilde{Y}_m e^{\Delta t} \mathbb{H}_m^B$ respectively.

Proof. As previously done in the proof of Theorem 3.4, 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 (3.25) and (3.19), we get

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

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

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

Therefore

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

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

$$405 \quad \left\| V_{m,r}^A \left(\tilde{F}_m(t) - \tilde{Y}_m \right) (\mathbb{V}_m^B)^T \right\| \leq r_m^A + z_m^A.$$

406 and similarly, we also have

$$407 \quad \left\| \mathbb{V}_m^A \left(\tilde{F}_m(t) - \tilde{Y}_m \right) (V_{m,r}^B)^T \right\| \leq r_m^B + z_m^B.$$

408 which completes the proof. \square

409 To continue the description of the present method, we notice that (3.24) enables us
 410 to check if $\|R_m(t)\| < \text{tol}$ -where tol is some fixed tolerance-, without having to
 411 compute extra products involving the large matrices A and B . More precisely, we
 412 have

$$413 \quad (3.27) \quad \|R_m(t)\| = \sqrt{\|H_{m+1,m}^A (\mathbb{E}_m^{(r)})^T Y_m(t)\|^2 + \|Y_m(t) \mathbb{E}_m^{(r)} (H_{m+1,m}^B)^T\|^2}.$$

We end this subsection by recalling that in the case of large scale problems, and as suggested in [23, 38], 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 $k = 1, \dots, N$

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

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

415

416 The block Arnoldi combined with the Constant Solution Method (BA-CSM) for
 417 solving the differential Sylvester matrix equation, in the case where C is low rank,
 418 i.e., $C = EF^T$, is summarized in Algorithm 3.3.

419 **4. Numerical experiments.** In this section, a series of numerical tests will be
 420 presented to examine the performance and potential of Algorithms 2.1, 3.2 and 3.3.
 421 We have compared our proposed method which is based on relation (2.1) with the one
 422 described in [20] and which is based on the integral formula (1.5). We recall that the
 423 algorithms described in [20] only provide an approximate solution at the final time T
 424 and moreover they only deal with the case of low-rank differential equations. Thus,
 425 we modified Algorithm 1 proposed in [20] so that it provides an approximate solution
 426 $X_{m,k} = X_m(t_k)$ at each node t_k of the discretization of the time interval $[0, T]$ as it
 427 is the case in Algorithm 3.3. Moreover, we have drafted two other codes based on the
 428 integral formula (1.5) and equivalent to Algorithm 2.1 and Algorithm 3.2

429 It should be noted that in all the examples given here, we suppose that X_0 the
 430 matrix appearing in the initial condition of (1.1) is equal to zero, i.e., $X_0 = 0_{n \times s}$.
 431 Furthermore, we consider different time intervals $[t_0, T]$ where $t_0 = 0$ is fixed once
 432 and for all, while T is indicated in each example. The time interval $[0, T]$ is divided
 433 into sub-intervals of constant length $\delta_T = \frac{T}{N}$ where N is the number of nodes. All
 434 the numerical experiments were performed using MATLAB and have been carried
 435 out on an Intel(R) Core(TM) i7 with 2.60 GHz processing speed and 16 GB memory.
 436 In order to implement the different algorithms described in this work, we used the
 437 following MATLAB functions:

- 438 - **expm**: it allows to calculate the exponential of a square matrix. This function
 439 is based on a scaling and squaring algorithm with a Padé approximation [24].
- 440 - **lyap**: it allows to solve Sylvester or Lyapunov matrix equations. For our
 441 purposes, the instruction **lyap**($A, B, -C$) delivers the matrix X solution of the
 442 algebraic Sylvester equation $AX + XB = C$.
- 443 - **integral**: it allows to calculate numerically an integral, using the arguments
 444 "ArrayValued" and "true".

445 Furthermore, we precise that when the constant solution or integral formula meth-
 446 ods are combined with the block Arnoldi process to obtain an approximate solution

Algorithm 3.3 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 $\text{tol} > 0$, 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** $m = 1, \dots, M_{max}$ **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 3.1 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 $r_{max} = \max\{r_{m,1}, \dots, r_{m,N}\}$
 - 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: 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**
-

447 to the differential equation, the iterations are stopped as soon as the dimension of the
448 Krylov subspace generated by the block Arnoldi process reaches a maximum value
449 $m = M_{max} = 110$ or as soon as the maximal norm r_{max} computed by the algo-
450 rithm is lower than $10^{-10} \mu$ where $\mu = \|A\| + \|B\| + \|C\|$ in the full rank case and
451 $\mu = \|A\| + \|B\| + \|E\| \|F\|$ in the low rank case. We also mention that in the numerical
452 examples, the right-hand side C or its factors E and F were generated randomly.

453

454 To compare the performances of the Constant Solution method (in short CS or CS-
455 BA when combined with the block Arnoldi process) with those of the Integral Formula
456 method (in short IF or IF-BA when combined with the block Arnoldi process), we
457 used the following comparison criteria:

- TR: the time ratio between the CPU-time of CS and IF methods or between

CS-BA and IF-BA methods which are defined by

$$\text{TR} = \frac{\text{cpu-time}(\text{CS})}{\text{cpu-time}(\text{IF})} \quad \text{or} \quad \text{TR} = \frac{\text{cpu-time}(\text{CS-BA})}{\text{cpu-time}(\text{IF-BA})}.$$

- **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.

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

458
459

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^{Exact} be the exact solution computed by (4.2), we define the following quantities :

$$\text{REN}^{\text{CS-BA}} = \max_{k=0,1,\dots,N} \frac{\|X_k^{\text{CS-BA}} - X_k^{\text{Exact}}\|}{\|X_k^{\text{Exact}}\|}$$

and

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

460
461
462

Before to start the numerical experiments and tests, we will show in the next subsection, how to construct a differential Sylvester equation which have a known exact benchmark solution.

463
464
465
466
467

4.1. A benchmark example. To the best of our knowledge, there is no Known benchmark explicit solution for large-scale differential and Sylvester matrix equations. Here, we show how to construct a benchmark differential Sylvester equation whose exact solution is known and with which we can confront the approximate solutions provided by the different compared methods.

468
469
470
471

Let $K, R \in \mathbb{R}^{p_0 \times p_0}$ be two nilpotent matrices, of index $p_0 \geq 3$ i.e., $K^{p_0} = R^{p_0} = 0_{p_0}$ and let $A_0 \in \mathbb{R}^{n_0 \times n_0}$, $B_0 \in \mathbb{R}^{s_0 \times s_0}$, $X_0, C \in \mathbb{R}^{n \times s}$ where $n = p_0 n_0$ and $s = p_0 s_0$. The integer p_0 is a small integer. We also choose two real numbers α, β and we consider the matrices

472

$$(4.1) \quad A = \alpha I_n + A_0 \otimes K, \quad B = \beta I_s + B_0 \otimes R,$$

We easily have, for any real t and for any matrix X of size $n \times s$:

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

where $L_{i,j}(X)$ is defined by $L_{i,j}(X) = \frac{1}{i!j!} (A_0^i \otimes K^i) X (B_0^j \otimes R^j)$. Assuming that $\alpha + \beta < 0$, then the unique solution \tilde{X}^* of the algebraic matrix Sylvester equation $AX + XB = C$ is given by the formula (see [31]),

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

A 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 $C_{i,j} = \frac{1}{i!j!} (A_0^i \otimes K^i) C (B_0^j \otimes R^j)$. Then, using the formula (2.1), the unique solution of the differential Sylvester matrix equation

$$X'(t) = A X(t) + X(t) B - C,$$

satisfying the initial condition $X(t_0) = X_0$, is the matrix function $X^*(t)$ given by

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

473 It follows that

$$474 \quad (4.2) \quad X^*(t) = \sum_{i=0}^{p_0-1} \sum_{j=0}^{p_0-1} \left[(t-t_0)^{i+j} e^{(\alpha+\beta)(t-t_0)} L_{i,j}(X_0 - \tilde{X}^*) + \frac{(-1)^{i+j}}{(\alpha + \beta)^{i+j+1}} L_{i,j}(C) \right].$$

We may also obtain the solution $X^*(t)$ of the differential Sylvester matrix equation by using the integral formula (1.5), since we have

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

475 It follows that,

$$476 \quad X^*(t) = \sum_{i=0}^{p_0-1} \sum_{j=0}^{p_0-1} \left[(t-t_0)^{i+j} e^{(\alpha+\beta)(t-t_0)} L_{i,j}(X_0) - I_{i+j}(t) L_{i,j}(C) \right],$$

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

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

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

In addition, to show the strengths and limitations of compared methods in various experimental settings when using this benchmark example, we choose $p_0 = 3$ and we considered different values for the parameters α and β as well as different matrices A_0 and B_0 . The matrices K and R are fixed once and for all, as follows

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

4.2. 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 2.1 with the one obtained using the integral formula (1.5) 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`. In the following two experiments, 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 $A_0 = \text{gallery}('leslie', n_0)$, $B_0 = \text{gallery}('minij', s_0)$ with $n_0 = 50$ and $s_0 = 10$ and the coefficient matrices A, B of the differential Sylvester equations are generated by (4.1), 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 4, 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 2.1 and those computed by the Algorithm based on integral formula or issued by the Matlab ODE solvers are compared to the exact one given in (4.2) which is considered as the reference solution X^{ref} . Thus, in the plots, we represent the behavior of the norm of the relative error

$$t_k \rightarrow \frac{\|X_k - X_k^{\text{ref}}\|}{\|X_k^{\text{ref}}\|}$$

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

TABLE 1
The obtained CPU times (in seconds) in Experiment 1.2.

(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

482 The analysis of results obtained in Experiments 1 shows on the one hand that
483 the CS and IF methods return the best results in terms of the error norm. The
484 `ode45` solver is the best among the other Matlab solvers, but its performance does
485 not match that of the CS and IF methods. On the other hand, by comparing the time
486 ratios between CS and IF which are $\text{TR} = \frac{12.578}{0.203} \simeq 61$ for $T = 1$ and $N = 10$ and
487 $\text{TR} = \frac{75.703}{0.718} \simeq 105$ for $T = 10$ and $N = 50$. We clearly see that CS is faster than IF
488 because the former avoids using a quadrature formula as it is the case for the later.

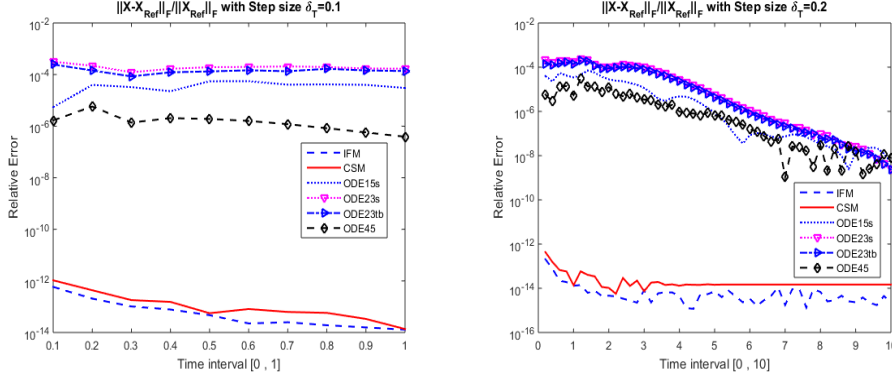


FIG. 1. *Experiment 1: comparison of the relative error norm. The reference solution is given by (4.2).*

4.3. 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 3.2 -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

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

and

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

489 To generate the coefficient matrices A and B , we used the `fdm_2d_matrix` function
 490 from the LYAPACK toolbox [33] as following $A = \text{fdm_2d_matrix}(n_0, f_A, g_A, h_A)$ and
 491 $B = \text{fdm_2d_matrix}(s_0, f_B, g_B, h_B)$ where n_0 and s_0 are the number of inner grid points
 492 in each direction when discretizing the operators L_A and L_B respectively. This gives
 493 $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{s \times s}$ with $n = n_0^2$ and $s = s_0^2$.

494 We examine the performances of CS-BA and IF-BA for four choices of n_0 and s_0
 495 which are $(n_0, s_0) = (30, 3)$, $(n_0, s_0) = (50, 3)$, $(n_0, s_0) = (30, 5)$ and $(n_0, s_0) = (50, 5)$.
 496 The considered time intervals are $[0, T]$ where $T = 1$ and $N = 10$ or $T = 2$ and
 497 $N = 20$. This means that the step time is always $\delta_T = 0.1$. In Table 2, we reported
 498 the time ratio (TR) and the relative difference norm (RDN) between the CPU-time of
 499 CS-BA and IF-BA.

500 **Experiment 2.2.** In this test, we took $A_0 = \text{gallery}('hanowa', 1500, -5)$ and
 501 $B_0 = \text{gallery}('leslie', 6)$ from the Matlab gallery and transform them into A
 502 et B of sizes $n = 4500$ and $s = 18$ respectively by using (4.1) in which we took
 503 $\alpha = -7$ and $\beta = -5$. The obtained results for different time intervals which are

TABLE 2

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	TR	RDN	TR	RDN
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

504 summarized in Table 3 include the time ratio TR and the relative error norms $\text{REN}^{\text{CS-BA}}$,
 505 $\text{REN}^{\text{IF-BA}}$ between the approximate solutions $X^{\text{CS-BA}}$, $X^{\text{IF-BA}}$ given by CS-BA and
 506 IF-BA respectively and the X^{Exact} the exact solution computed by (4.2).

TABLE 3

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

T	TR	$\text{REN}^{\text{CS-BA}}$	$\text{REN}^{\text{IF-BA}}$
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

507 **4.4. Experiment 3.** We describe and report here the results of numerical ex-
 508 periments carried out when solving large scale low-rank differential Sylvester or Lya-
 509 punov equations. The performance of CS-BA is compared with that of IF-BA. The
 510 test matrices come either from the centred finite difference discretization of the oper-
 511 ators L_A and L_B defined in the previous experiment, or from the Florida suite sparse
 512 matrix collection [14]. The invoked matrices for our tests from this collection are:
 513 **pde900**, **pde2961**, **cdde1**, **Chem97ZtZ**, **thermal**, **rdb5000**, **sstmodel**, **add32**
 514 and **rw5151**.

515 **Experiment 3.1 (a).** In this example, the numerical results are those obtained
 516 from solving differential Sylvester equations. The time interval is fixed to $[0, 1]$, ($T =$
 517 1). The number of nodes is $N = 10$ which gives a step time $\delta_T = 0.1$. The matrices
 518 $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{s \times s}$ come from the discretization of the operators L_A and L_B . As
 519 indicated previously, the coefficients of the right-hand side $E, F \in \mathbb{R}^{n \times r}$ are randomly
 520 generated. The obtained results for different sizes n, s and ranks r are summarized
 521 in Table 4.

522 **Experiment 3.1 (b).** Here, we consider two different time intervals $[0, T]$ for
 523 $T = 1$ and $T = 10$ in which the number of sub-intervals is always $N = 10$. The
 524 matrix A is from the Florida sparse matrix collection. We consider the particular
 525 case $B = A^T$ and $F = E$ and report the results obtained when solving low-rank
 526 differential Lyapunov equations. The obtained results for $r = 2$, $r = 5$ or $r = 10$ are
 527 displayed in Table 5.

528 **Experiment 3.2 (a).** Here, we consider $A_0 = \text{pde2961}$ and $B_0 = \text{pde900}$ and
 529 transform them into A et B of sizes $n = 8883$ and $s = 2700$ respectively by using (4.1).
 530 In order to confirm the influence of the rank r and/or length T of the time interval, on

TABLE 4

The obtained times ratio TR and relative difference norms RDN in Experiment 3.1 (a).

	Test Problems					
	$n_0 = 40, s_0 = 20$ $n = 1600, s = 400$		$n_0 = 30, s_0 = 30$ $n = 900, s = 900$		$n_0 = 50, s_0 = 50$ $n = 2500, s = 2500$	
r	TR	RDN	TR	RDN	TR	RDN
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 5

The obtained times ratio TR and relative difference norms RDN in Experiment 3.1 (b).

T	r	Test Problems									
		$A = -\text{cdde1}$ $n = 961$		$A = -\text{Chem97ZtZ}$ $n = 2541$		$A = -\text{pde2961}$ $n = 2961$		$A = \text{thermal}$ $n = 3456$		$A = \text{rdb5000}$ $n = 5000$	
		TR	RDN	TR	RDN	TR	RDN	TR	RDN	TR	RDN
1	2	6.8	3.988e-15	12	5.465e-15	1.6	2.784e-15	1.4	4.295e-15	14	7.863e-13
	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
10	2	47	2.815e-14	15	8.190e-15	12	1.002e-14	2.5	3.592e-15	57	3.482e-13
	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

531 the performances of the CS and IF methods, we report in Table 6 the results obtained
 532 for two cases : **case 1:** $(\alpha, \beta) = (-3, -1)$ and **case 2:** $(\alpha, \beta) = (-0.7, -0.4)$. For
 533 each case, we choose T from the set $\{2, 5, 10\}$ and took $N = 10$ for $T = 2$, $N = 20$ for
 534 $T = 5$ and $N = 40$ for $T = 10$. The rank r of the factors E and F is equal to $r = 5$,
 535 $r = 10$ or $r = 20$.

TABLE 6

The obtained times ratio TR and relative error norms norms $\text{REN}^{\text{CS-BA}}$ and $\text{REN}^{\text{IF-BA}}$ in Experiment 3.2 (b).

T	r	$\alpha = -3, \beta = -1$			$\alpha = -0.7, \beta = -0.4$		
		TR	$\text{REN}^{\text{CS-BA}}$	$\text{REN}^{\text{IF-BA}}$	TR	$\text{REN}^{\text{CS-BA}}$	$\text{REN}^{\text{IF-BA}}$
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	---

536

537

538

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

539 observed that for small values of α and β and when the values of r and T are large,
 540 the BA-IF method failed to converge within a reasonable time. The non-convergence
 541 is indicated by " ---".

542 **Experiment 3.2 (b).** In this last set of experiments, we compare the perfor-
 543 mances of the CS and IF methods when they are applied to the solution of low-rank
 544 differential Lyapunov equations. Unlike the previous series of tests, we did not gener-
 545 ate a discretization for the interval $[0, T]$ and only calculated the approximation $X(T)$
 546 at the final time, where $T = 10$. Similarly, the rank of $C = E E^T$ does not vary and
 547 is $r = 20$. For each experiment with a matrix A_0 -which is taken from the Florida
 548 sparse matrix collection [14]-, we considered four values for the scalar α that was used
 549 in the generation of the benchmark example. The size n_0 of each matrix A_0 , the size
 550 n of the benchmark matrix A as well as the obtained results are reported in Table 7.

TABLE 7
 Numerical results for Experiment 3.2 (b) with $T = 10$, $N = 1$ and $r = 20$.

α	Test Problems								
	$A_0 = \mathbf{cdde1}$ $n_0 = 961, n = 2883$			$A_0 = \mathbf{pde2961}$ $n_0 = 2961, n = 8883$			$A_0 = \mathbf{sstmodel}$ $n_0 = 3345, n = 10035$		
	TR	REN ^{CS-BA}	REN ^{IF-BA}	TR	REN ^{CS-BA}	REN ^{IF-BA}	TR	REN ^{CS-BA}	REN ^{IF-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	---

α	Test Problems								
	$A_0 = \mathbf{thermal}$ $n_0 = 3456, n = 10368$			$A_0 = \mathbf{add32}$ $n_0 = 4960, n = 14880$			$A_0 = \mathbf{rw5151}$ $n_0 = 5151, n = 15453$		
	TR	REN ^{CS-BA}	REN ^{IF-BA}	TR	REN ^{CS-BA}	REN ^{IF-BA}	TR	REN ^{CS-BA}	REN ^{IF-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	---

551 **5. Conclusion.** In this work, we have proposed new techniques for solving
 552 Sylvester and Lyapunov matrix differential equation. Unlike the recent method pro-
 553 posed in [20], our method avoid the integral formula which is very benefit and reduced
 554 the computational cost. The proposed method is very efficient for large scale problem
 555 by exploiting a projection on Krylov subspaces. Numerous numerical tests are used
 556 to show the effectiveness of such proposed method, we have reported some of them
 557 in a specific section. The convergence of such method is proved and a constructive
 558 benchmark example is given.

559

REFERENCES

- 560 [1] H. ABOU-KANDIL, G. FREILING, V. IONESCU, AND G. JANK, *Matrix Riccati equations in control*
 561 *and systems theory*, vol. 3, Basel, Switzerland: Birkhauser, 2003.
 562 [2] F. AMATO, R. AMBROSINO, M. ARIOLA, C. COSENTINO, AND G. DE TOMMASI, *Finite-Time*
 563 *Stability and Control*, Springer, 2014.
 564 [3] A. C. ANTOUNAS, *Approximation of Large-Scale Dynamical Systems*, vol. 6, Adv. Des. Control.
 565 SIAM Publications, Philadelphia, PA, 2005.
 566 [4] R. H. BARTELS AND G. W. STEWART, *Solution of the matrix equation $AX + XB = C$* , Commun.
 567 ACM, 15 (1972), pp. 820–826, <http://doi.acm.org/10.1145/361573.361582>.

- 568 [5] M. BEHR, P. BENNER, AND J. HEILAND, *Solution formulas for differential Sylvester and*
569 *Lyapunov equations*, *Calcolo*, (2019), pp. 56–51, [https://doi.org/https://doi.org/10.1007/](https://doi.org/https://doi.org/10.1007/s10092-019-0348-x)
570 [s10092-019-0348-x](https://doi.org/https://doi.org/10.1007/s10092-019-0348-x).
- 571 [6] R. BELLMAN, *Introduction to Matrix Analysis*, vol. 6, SIAM publications, Philadelphia, PA,
572 2nd edition, 1997, First edition published by McGraw-Hill in 1970.
- 573 [7] P. BENNER AND H. MENA, *BDF methods for large-scale differential Riccati equations*, (2009).
- 574 [8] P. BENNER AND H. MENA, *Rosenbrock methods for solving Riccati differential equations*, *IEEE*
575 *Transactions on Automatic Control*, 58 (2013), pp. 2950–2956, [https://doi.org/10.1109/](https://doi.org/10.1109/TAC.2013.2258495)
576 [TAC.2013.2258495](https://doi.org/10.1109/TAC.2013.2258495).
- 577 [9] A. BOUHAMIDI, M. HACHED, M. HEYOUNI, AND K. JBILOU, *A preconditioned block Arnoldi*
578 *method for large Sylvester matrix equations*, *Numerical Linear Algebra with Applications*,
579 20 (2013), pp. 208–219, <https://doi.org/10.1002/nla.831>.
- 580 [10] A. BOUHAMIDI AND K. JBILOU, *Sylvester Tikhonov-regularization methods in image restoration*,
581 *J. Comput. Appl. Math.*, 206 (2007), pp. 86–98, [https://doi.org/10.1016/j.cam.2006.05.](https://doi.org/10.1016/j.cam.2006.05.028)
582 [028](https://doi.org/10.1016/j.cam.2006.05.028).
- 583 [11] A. BOUHAMIDI AND K. JBILOU, *A note on the numerical approximate solutions for generalized*
584 *Sylvester matrix equations with applications*, *Appl. Math. Comput.*, 206 (2008), pp. 687–
585 694, <https://doi.org/10.1016/j.amc.2008.09.022>.
- 586 [12] C. F. CURTISS AND J. O. HIRSCHFELDER, *Integration of Stiff Equations*, vol. 38, National
587 Academy of Sciences of the United States of America, 1952, pp. 235–243, [https://doi.org/](https://doi.org/10.1073/pnas.38.3.235)
588 [10.1073/pnas.38.3.235](https://doi.org/10.1073/pnas.38.3.235), <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1063538/>.
- 589 [13] B. N. DATTA, *Numerical Methods for Linear Control Systems*, Academic Press, USA, 2004,
590 <https://doi.org/10.1016/B978-0-12-203590-6.X5000-9>.
- 591 [14] T. A. DAVIS AND Y. HU, *The university of florida sparse matrix collection*, *ACM Trans. Math.*
592 *Softw.*, 38 (2011), <https://doi.org/10.1145/2049662.2049663>.
- 593 [15] E. DAVISON, *The numerical solution of $X = A_1X + XA_2 + D$, $X(0) = C$* , *IEEE Trans. Autom.*
594 *Control*, 20 (1975), pp. 566–567.
- 595 [16] A. EL GUENNOUNI, K. JBILOU, AND A. RIQUET, *Block Krylov subspace methods for solving*
596 *large Sylvester equations*, *Numerical Algorithms*, 29 (2002), pp. 75–96, [https://doi.org/10.](https://doi.org/10.1023/A:1014807923223)
597 [1023/A:1014807923223](https://doi.org/10.1023/A:1014807923223).
- 598 [17] L. ELBOUYAHYAUI, M. HEYOUNI, K. JBILOU, AND A. MESSAOUDI, *A block Arnoldi method*
599 *for the solution of the Sylvester-Observer equation*, *Electronic Transactions on Numerical*
600 *Analysis*, 47 (2017), pp. 18–36, <https://doi.org/10.1553/etna.vol47s18>.
- 601 [18] G. GOLUB, S. NASH, AND C. V. LOAN, *A Hessenberg-Schur method for the problem $AX + XB =$*
602 *C* , *IEEE Transactions on Automatic Control*, 24 (1979), pp. 909–913, [https://doi.org/10.](https://doi.org/10.1109/TAC.1979.1102170)
603 [1109/TAC.1979.1102170](https://doi.org/10.1109/TAC.1979.1102170).
- 604 [19] M. HACHED AND K. JBILOU, *Numerical solutions to large-scale differential Lyapunov matrix*
605 *equations*, *Numerical Algorithms*, 79 (2017), pp. 741–757.
- 606 [20] M. HACHED AND K. JBILOU, *Computational Krylov-based methods for large-scale differen-*
607 *tial Sylvester matrix problems*, *Numerical Linear Algebra with Applications*, 25 (2018),
608 p. e2187, <https://doi.org/10.1002/nla.2187>.
- 609 [21] S. J. HAMMARLING, *Numerical Solution of the Stable, Non-negative Definite Lyapunov Equa-*
610 *tion*, *IMA Journal of Numerical Analysis*, 2 (1982), pp. 303–323, [https://doi.org/10.1093/](https://doi.org/10.1093/imanum/2.3.303)
611 [imanum/2.3.303](https://doi.org/10.1093/imanum/2.3.303).
- 612 [22] M. HEYOUNI, *Extended Arnoldi methods for large low-rank Sylvester matrix equations*, *Ap-*
613 *plied Numerical Mathematics*, 60 (2010), pp. 1171 – 1182, [http://www.sciencedirect.com/](http://www.sciencedirect.com/science/article/pii/S0168927410001224)
614 [science/article/pii/S0168927410001224](http://www.sciencedirect.com/science/article/pii/S0168927410001224).
- 615 [23] M. HEYOUNI AND K. JBILOU, *An extended block Arnoldi algorithm for large-scale solutions*
616 *of the continuous-time algebraic Riccati equation*, *Electronic Transactions on Numerical*
617 *Analysis*, 33 (2009), pp. 53–62, [http://etna.math.kent.edu/volumes/2001-2010/vol33/](http://etna.math.kent.edu/volumes/2001-2010/vol33/abstract.php?vol=33&pages=53-62)
618 [abstract.php?vol=33&pages=53-62](http://etna.math.kent.edu/volumes/2001-2010/vol33/abstract.php?vol=33&pages=53-62).
- 619 [24] N. J. HIGHAM, *The scaling and squaring method for the matrix exponential revised*, *SIAM J.*
620 *Matrix Anal Appl*, 26(4) (2005), pp. 1179–1193.
- 621 [25] R. A. HORN AND C. R. JOHNSON, *Topics in matrix analysis*, vol. (2nd edition), Cambridge
622 University Press, UK, 1994.
- 623 [26] D. HU AND L. REICHEL, *Krylov-subspace methods for the Sylvester equation*, *Linear Algebra and*
624 *its Applications*, 172 (1992), pp. 283–313, [https://doi.org/10.1016/0024-3795\(92\)90031-5](https://doi.org/10.1016/0024-3795(92)90031-5).
- 625 [27] I. M. JAIMOUKHA AND E. M. KASENALLY, *Krylov subspace methods for solving large Lyapunov*
626 *equations*, *SIAM Journal on Numerical Analysis*, 31 (1994), pp. 227–251, [http://www.](http://www.jstor.org/stable/2158160)
627 [jstor.org/stable/2158160](http://www.jstor.org/stable/2158160).
- 628 [28] K. JBILOU, *ADI preconditioned Krylov methods for large Lyapunov matrix equations*, *Linear*
629 *Algebra and its Applications*, 432 (2010), pp. 2473–2485, <https://doi.org/10.1016/j.laa>

- 630 [2009.12.025](#).
- 631 [29] M. KONSTANTINOV, D.-W. GU, V. MEHRMANN, AND P. PETKOV, *Perturbation Theory for*
632 *Matrix Equations*, vol. 49, 11 2004, <https://doi.org/10.1109/TAC.2004.835351>.
- 633 [30] D. KRESSNER, *Block variants of Hammarling's method for solving Lyapunov equations*, ACM
634 *Trans. Math. Softw.*, 34 (2008), <https://doi.org/10.1145/1322436.1322437>.
- 635 [31] P. LANCASTER, *Explicit solutions of linear matrix equations*, *SIAM Review*, 12 (1970), pp. 544–
636 566.
- 637 [32] C. MOLER AND C. V. LOAN, *Nineteen dubious ways to compute the exponential of a matrix,*
638 *twenty-five years later*, *SIAM Review*, 45(1) (2003), pp. 3–000.
- 639 [33] T. PENZL, *LYAPACK. A MATLAB Toolbox for large Lyapunov and Riccati equations, Model*
640 *Reduction Problems, and Linear-Quadratic Optimal Control Problems*, 2000.
- 641 [34] Y. SAAD, *Numerical solution of large Lyapunov equations*, in *Signal Processing, Scattering*
642 *and Operator Theory, and Numerical Methods*, Proc. MTNS-89, Birkhauser, 1990, pp. 503–
643 511.
- 644 [35] Y. SAAD, *Analysis of some Krylov subspace approximations to the matrix exponential operator*,
645 *SIAM J. Numer. Anal.*, 29(1) (1992), pp. 209–228.
- 646 [36] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied
647 *Mathematics*, second ed., 2003, <https://doi.org/10.1137/1.9780898718003>.
- 648 [37] E. M. SADEK, A. BENTBIB, L. SADEK, AND H. ALAOU, *Global extended Krylov subspace meth-*
649 *ods for large-scale differential Sylvester matrix equations*, *Journal of Applied Mathematics*
650 *and Computing*, 62 (2019), pp. 165–177, <https://doi.org/10.1007/s12190-019-01278-7>.
- 651 [38] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*,
652 *SIAM Journal on Scientific Computing*, 29 (2007), pp. 1268–1288, [https://doi.org/10.1137/](https://doi.org/10.1137/06066120X)
653 [06066120X](https://doi.org/10.1137/06066120X).
- 654 [39] V. SIMONCINI, *Computational methods for linear matrix equations*, *SIAM Review*, 58 (2016),
655 pp. 377–441, <https://doi.org/10.1137/130912839>.
- 656 [40] D. C. SORENSEN AND Y. ZHOU, *Direct methods for matrix Sylvester and Lyapunov equations*,
657 *Journal of Applied Mathematics*, 2003 (2003), pp. 277 – 303, [https://doi.org/10.1155/](https://doi.org/10.1155/S1110757X03212055)
658 [S1110757X03212055](https://doi.org/10.1155/S1110757X03212055).