



HAL
open science

Numerical methods for differential linear matrix equations via Krylov subspace methods

M. Hached, Khalide Jbilou

► **To cite this version:**

M. Hached, Khalide Jbilou. Numerical methods for differential linear matrix equations via Krylov subspace methods. *Journal of Computational and Applied Mathematics*, 2020, 370, pp.112674. 10.1016/j.cam.2019.112674 . hal-04392156

HAL Id: hal-04392156

<https://hal.science/hal-04392156v1>

Submitted on 13 Jan 2024

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.

NUMERICAL METHODS FOR DIFFERENTIAL LINEAR MATRIX EQUATIONS VIA KRYLOV SUBSPACE METHODS

M. HACHED * AND K. JBILOU†

Abstract. In the present paper, we present some numerical methods for computing approximate solutions to some large differential linear matrix equations. In the first part of this work, we deal with differential generalized Sylvester matrix equations with full rank right-hand sides using a global Galerkin and a norm-minimization approaches. In the second part, we consider large differential Lyapunov matrix equations with low rank right-hand sides and use the extended global Arnoldi process to produce low rank approximate solutions. We give some theoretical results and present some numerical examples.

Sylvester equation, Lyapunov equation, Global Arnoldi, matrix Krylov subspace.

subclass MSC 65F10

1. Introduction. In this work, we are interested in computing a numerical solution of two kinds of differential linear matrix equations. First, we consider the linear matrix differential equation with a full right-hand side

$$\begin{cases} \dot{X}(t) = \sum_{i=1}^q A_i X(t) B_i + C, \\ X(t_0) = X_0, \quad t \in [t_0, T_f], \end{cases} \quad (1.1)$$

where $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{p \times p}$, $i = 1, \dots, q$, C and $X \in \mathbb{R}^{n \times p}$, and we assume that the right hand term C is full rank and $p \ll n$. Differential Sylvester and Lyapunov matrix equations are particular cases of (1.1). The next differential matrix equation that will be considered in this paper, is the well known differential Lyapunov matrix equation with a low rank right hand side

$$\begin{cases} \dot{X}(t) = AX(t) + X(t)A^T + BB^T; \quad (DLE) \\ X(t_0) = X_0, \quad t \in [t_0, T_f], \end{cases} \quad (1.2)$$

where the matrix $A \in \mathbb{R}^{n \times n}$ is assumed to be large, sparse and nonsingular and $B \in \mathbb{R}^{n \times p}$ is a full rank matrix, with $p \ll n$. The initial condition $X_0 = \tilde{Z}_0 \tilde{Z}_0^T$ is assumed to be a given symmetric and positive low-rank matrix.

The differential linear matrix equations (1.1) and (1.2) play an important role in many areas such as control, filter design theory, model reduction problems, differential equations and robust control problems [1, 7].

Notice that the two linear differential matrix equations above can be reformulated as

$$\dot{x}(t) = \mathcal{M}x(t) + b, \quad (1.3)$$

where $x(t) = \text{vec}(X(t))$, the matrix \mathcal{M} is given by $\mathcal{M} = \sum_{i=1}^q (B_i^T \otimes A_i)$ for problem (1.1) and $\mathcal{M} = I \otimes A + A \otimes I$ for (1.2), while the right hand side b is given by $b = \text{vec}(C)$ for (1.1) and $b = \text{vec}(BB^T)$ for (1.2) respectively, and $\text{vec}(C)$ is the long vector obtained by stacking the columns of the matrix C . For small dimensional problems, classical solvers can be used to approximate the exact solution of (1.3)

*Laboratoire P. Painlevé UMR 8524, UFR de Mathématiques, Université des Sciences et Technologies de Lille, IUT A, Rue de la Recherche, BP 179, 59653 Villeneuve d'Ascq Cedex, France; email: mustapha.hached@univ-lille.fr

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

which can be written as

$$x(t) = e^{t\mathcal{M}} x_0 + \int_{t_0}^t e^{(t-\tau)\mathcal{M}} b d\tau, \quad (1.4)$$

However, in the cases for which the matrix \mathcal{M} is very large, this approach would not be appropriate as computational time grows up very quickly as the dimension of the problem gets larger.

The rest of the paper is organized as follows. In Section 2, we recall the definitions of the Kronecker and the \diamond products with some of their properties that will be of use in this work. In Section 3, we give a numerical method for solving the problem (1.1) by using projections onto matrix Krylov subspaces, based on a Global-Galerkin orthogonality condition. In Section 4, we will be interested in the numerical solution of the Lyapunov differential matrix equation (1.2). The approximate solutions will be obtained via projection onto matrix Krylov subspaces using the extended global Arnoldi algorithm. The last section is devoted to some numerical examples.

2. Preliminaries.

2.1. Definitions. We begin by recalling some notations that will be used in the sequel. We define the inner product

$$\langle Y, Z \rangle_F = \text{tr}(Y^T Z),$$

where $\text{tr}(Y^T Z)$ denotes the trace of the matrix $Y^T Z$ such that $Y, Z \in \mathbb{R}^{n \times p}$. The associated norm is the Frobenius norm denoted by $\|Z\|_F = \sqrt{\langle Z, Z \rangle_F}$.

The matrix product $A \otimes B = [a_{i,j} B]$ denotes the well known Kronecker product of the matrices A and B which satisfies the following properties:

1. $(A \otimes B)(C \otimes D) = (AC \otimes BD)$.
2. $(A \otimes B)^T = A^T \otimes B^T$.
3. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, if A and B are nonsingular square matrices.

We also use the matrix product \diamond defined in [5] as follows.

DEFINITION 2.1. Let $\mathcal{Z} = [Z_1, \dots, Z_m]$ and $\mathcal{W} = [W_1, \dots, W_l]$ be matrices of dimension $n \times mp$ and $n \times lp$ respectively, where Z_i and W_j ($i = 1, \dots, m$, $j = 1, \dots, l$) are $\mathbb{R}^{n \times p}$. Then the $\mathbb{R}^{m \times l}$ matrix $\mathcal{Z}^T \diamond \mathcal{W}$ is defined as:

$$\mathcal{Z}^T \diamond \mathcal{W} = [\langle Z_i, W_j \rangle_F]_{1 \leq i \leq m, 1 \leq j \leq l}$$

A block matrix $\mathcal{W} = [W_1, \dots, W_l]$ is said to be F-orthonormal if

$$\langle W_i, W_j \rangle_F = \delta_{i,j} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad i, j = 1, \dots, l. \quad (2.1)$$

which is equivalent to

$$\mathcal{W}^T \diamond \mathcal{W} = I_l.$$

For more details about the properties of the \diamond matrix product, see [5].

Let A and V be $n \times n$ and $n \times p$ matrices, respectively, then the matrix (also called the global) Krylov subspace $\mathcal{K}_m(A, V)$ associated to the pair (A, V) is the subspace of $\mathbb{R}^{n \times p}$ generated by $V, AV, \dots, A^{m-1}V$, i.e.,

$$\mathcal{K}_m(A, V) = \text{span}\{V, AV, \dots, A^{m-1}V\}.$$

In the next proposition, we recall the global QR (gQR) factorisation of an $n \times mp$ matrix Z . The algorithm of such a matrix factorisation is given in [5].

PROPOSITION 2.2. [5] Let $\mathcal{Z} = [Z_1, Z_2, \dots, Z_m]$ be an $n \times mp$ matrix with $Z_i \in \mathbb{R}^{n \times p}$, $i = 1, \dots, m$. Then, the matrix \mathcal{Z} can be factored as

$$\mathcal{Z} = \mathcal{Q}(R \otimes I_p),$$

where $\mathcal{Q} = [Q_1, \dots, Q_m]$ is an $n \times mp$ F-orthonormal matrix satisfying $\mathcal{Q}^T \diamond \mathcal{Q} = I_m$ and R is an upper triangular matrix of dimension $m \times m$.

The following proposition will be useful later.

PROPOSITION 2.3. [20] Let $\mathcal{V}_m = [V_1, \dots, V_m]$, be an $n \times mp$ F-orthonormal matrix. Let $Z = [z_{i,j}]$ and $G = [g_{i,j}]$ be matrices of sizes $m \times r$ and $mp \times q$ respectively, where r and q are any integers. Then we have

$$\|\mathcal{V}_m(Z \otimes I_p)\|_F = \|Z\|_F$$

and

$$\|\mathcal{V}_m G\|_F \leq \|G\|_F.$$

3. The Global-Galerkin Krylov subspace method for linear matrix differential equations.

In this section, we consider the differential linear matrix equation (1.1) and will present an iterative projection method to get numerical approximate solutions. Let \mathcal{A} be the linear matrix operator defined as follows

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{n \times p} &\longrightarrow \mathbb{R}^{n \times p} \\ X &\longrightarrow \sum_{i=1}^q A_i X B_i. \end{aligned}$$

Notice that the transpose of the operator \mathcal{A} with respect to the inner product $\langle \cdot, \cdot \rangle_F$ is defined as the application mapping $X \in \mathbb{R}^{n \times p}$ $\mathcal{A}^T(X) = \sum_{i=1}^q A_i^T X B_i^T$.

Let V be any $n \times p$ matrix then we define the matrix Krylov subspace associated to the pair (\mathcal{A}, V) and an integer m defined by

$$\mathcal{K}_m(\mathcal{A}, V) = \text{span}\{V, \mathcal{A}(V), \dots, \mathcal{A}^{m-1}(V)\},$$

Where $\mathcal{A}^i(V)$ is defined recursively as $\mathcal{A}^i(V) = \mathcal{A}(\mathcal{A}^{i-1}(V))$. Notice that the matrix Krylov subspace $\mathcal{K}_m(\mathcal{A}, V)$ is a subspace of $\mathbb{R}^{n \times p}$, which means that if a matrix Y is in $\mathcal{K}_m(\mathcal{A}, V)$, then we have $Y = \sum_{i=1}^m \alpha_i \mathcal{A}^{i-1}(V)$ where $\alpha_i \in \mathbb{R}$, $i = 1, \dots, m$. Next, we remind the modified global Arnoldi algorithm [19] that allows us to construct an F-orthonormal basis V_1, V_2, \dots, V_m of the matrix Krylov subspace $\mathcal{K}_m(\mathcal{A}, V)$, i.e.

$$\langle V_i, V_j \rangle_F = \delta_{i,j}, \quad \text{for } i, j = 1, \dots, k,$$

Algorithm 1 The Modified Global Arnoldi algorithm

- Set $V_1 = V/\|V\|_F$.
 - For $j = 1, \dots, k$
 1. $\tilde{V} = \mathcal{A}(V_j)$,
 2. for $i = 1, \dots, j$.
 - (a) $h_{i,j} = \langle V_i, \tilde{V} \rangle_F$
 - (b) $\tilde{V} = \tilde{V} - h_{i,j}V_i$,
 - (c) EndFor
 3. $h_{j+1,j} = \|\tilde{V}\|_F$,
 4. $V_{j+1} = \tilde{V}/h_{j+1,j}$.
 - EndFor.
-

where $\delta_{i,j}$ denotes the classical Kronecker symbol. The algorithm is described as follows.

The matrix \tilde{H}_m denotes the $(m+1) \times m$ upper Hessenberg matrix whose nonzero entries $h_{i,j}$ are defined by Algorithm 1 and H_m is the $m \times m$ matrix obtained from \tilde{H}_m by deleting its last row. The $n \times mp$ block matrix $\mathcal{V}_m = [V_1, V_2, \dots, V_m]$ is F-orthonormal which means that the matrices V_1, \dots, V_m are orthonormal with respect to the scalar product $\langle \cdot, \cdot \rangle_F$ which is equivalent to

$$\mathcal{V}_m^T \diamond \mathcal{V}_m = I_p. \quad (3.1)$$

We have the following relations

$$[\mathcal{A}(V_1), \dots, \mathcal{A}(V_m)] = \mathcal{V}_m(H_m \otimes I_p) + E_{m+1}, \quad (3.2)$$

where $E_{m+1} = h_{m+1,m} [0_{n \times p}, \dots, 0_{n \times p}, V_{m+1}]$, and

$$[\mathcal{A}(V_1), \dots, \mathcal{A}(V_m)] = \mathcal{V}_{m+1}(\tilde{H}_m \otimes I_p). \quad (3.3)$$

Starting from an initial guess $X_0(t) \in \mathbb{R}^{n \times p}$ and the corresponding residual $\mathcal{R}_0(t) = \dot{X}_0(t) - \mathcal{A}(X_0(t)) - C$, at step m , we define the approximation X_m as follows

$$X_m(t) = X_0(t) + Z_m(t) \quad \text{with } Z_m(t) \in \mathcal{K}_m(\mathcal{A}, \mathcal{R}_0(t)) \quad (3.4)$$

and

$$\mathcal{R}_m(t) \perp_F \mathcal{K}_m(\mathcal{A}, \mathcal{R}_0(t)). \quad (3.5)$$

where

$$\mathcal{R}_m(t) = \dot{X}_m(t) - \mathcal{A}(X_m(t)) - C$$

The Galerkin condition (3.5) is equivalent to

$$\mathcal{V}_m^T \diamond \mathcal{R}_m(t) = 0. \quad (3.6)$$

The condition (3.4) can be written as

$$X_m(t) = X_0(t) + \sum_{i=1}^m y_m^{(i)}(t) V_i = X_0(t) + \mathcal{V}_m(y_m(t) \otimes I_p), \quad (3.7)$$

where $y_m(t)$ is a vector of \mathbb{R}^m and $y_m^{(i)}(t)$ is the i -th component of $y_m(t)$. Therefore, the residual $\mathcal{R}_m(t)$ can be expressed as

$$\begin{aligned}
\mathcal{R}_m(t) &= \dot{X}_m(t) - \mathcal{A}(X_m(t)) - C \\
&= \dot{X}_0(t) - \mathcal{A}(X_0(t)) + \mathcal{V}_m(\dot{y}_m(t) \otimes I_p) - \mathcal{A}(\mathcal{V}_m(y_m(t) \otimes I_p)) - C \\
&= \mathcal{V}_m(\dot{y}_m(t) \otimes I_p) - \mathcal{A}\left(\sum_{i=1}^p y_m^{(i)}(t)V_i\right) + \mathcal{R}_0(t) \\
&= \mathcal{V}_m(\dot{y}_m(t) \otimes I_p) - \left(\sum_{i=1}^p y_m^{(i)}(t)\mathcal{A}(V_i)\right) + \mathcal{R}_0(t) \\
&= \mathcal{V}_m(\dot{y}_m(t) \otimes I_p) - [\mathcal{A}(V_1), \mathcal{A}(V_2), \dots, \mathcal{A}(V_m)](y_m(t) \otimes I_p) + \mathcal{R}_0(t).
\end{aligned}$$

Using the relation (3.2), it follows that

$$\mathcal{R}_m(t) = \mathcal{V}_m(\dot{y}_m(t) \otimes I_p) - \mathcal{V}_m(H_m \otimes I_p)(y_m(t) \otimes I_p) - E_{m+1}(y_m(t) \otimes I_p) + \mathcal{R}_0(t).$$

On the other hand, E_{m+1} can be expressed as $E_{m+1} = h_{m+1,m}V_{m+1}[0, 0, \dots, I]$ which can be written as $E_{m+1} = h_{m+1,m}V_{m+1}(\tilde{E} \otimes I_p)$. Then we get a new expression of the residual given by

$$\mathcal{R}_m(t) = \mathcal{V}_m(\dot{y}_m(t) \otimes I_p) - \mathcal{V}_m(H_m y_m(t) \otimes I_p) - h_{m+1,m}V_{m+1}(\tilde{E} y_m(t) \otimes I_p) + \mathcal{R}_0(t). \quad (3.8)$$

Using the properties of the \diamond product given in [5] and the fact that $\mathcal{V}_m^T \diamond V_{m+1} = 0$, the F-orthogonality condition (3.6) reduces to the low dimensional linear differential system of equations

$$\dot{y}_m(t) = H_m y_m(t) + c_m(t), \quad (3.9)$$

where $c_m = -\mathcal{V}_m^T \diamond \mathcal{R}_0(t) \in \mathbb{R}^m$.

The solution of the ODE (3.9) is given by

$$y_m(t) = e^{tH_m} y_m(0) + \int_0^t e^{(t-\tau)H_m} c_m(\tau) d\tau. \quad (3.10)$$

The computation of the exponential form (3.10) can be done applying a quadrature method to compute the projected exponential. In our numerical tests, a scaling and squaring strategy, implemented in the MATLAB `expm` function was used, see [16, 21] for more details.

In the next algorithm, we summarize the main steps of the global-Galerkin

Algorithm 2 The global-Galerkin (GG) algorithm

1. Choose a tolerance ε and a maximum number of Arnoldi iteration m_{max} .
 2. Compute $\beta = \|\mathcal{R}_0(t)\|_F$, and $V_1 = \mathcal{R}_0(t)/\beta$.
 3. For $m = 1 \dots m_{max}$
 - (a) Construct the F -orthonormal basis V_1, V_2, \dots, V_m by Algorithm 1.
 - (b) Determine y_m as solution of problem (3.10).
 - (c) Compute the residual norm $\|\mathcal{R}_m(t)\|_F$.
 - (d) If $\|\mathcal{R}_m(t)\|_F < \varepsilon$ stop, else, Goto (a).
 4. EndFor.
 5. Compute the approximation $X_m = X_0 + \mathcal{V}_m(y_m \otimes I_p)$.
-

4. Global projection methods for large differential Lyapunov equations with low-rank right-hand sides. In this section, we consider the following large scale differential Lyapunov equation (1.2). Those differential matrix equations play a fundamental role in many topics such as control, model reduction problems, differential equations and robust control problems [1, 7]. When the size n of the problem is large, the global-Galerkin (GG) approach is no longer interesting as the right-hand side BB^T is a $n \times n$ matrix. In order to take advantage of the structure of the right-hand side, we introduce in this section an approach based on the computation of low-rank approximate solutions to the exact solution X using the extended global Arnoldi process [9, 13, 14, 25]. In [12], methods based on the projection onto extended block Krylov subspaces showed to be an efficient tool. Nevertheless, the block Krylov methods can suffer from breakdowns when numerical colinearities appear between the vectors generated by the block Arnoldi process, resulting in a degradation of the accuracy. As an alternative, we propose a global Arnoldi approach to solve (1.2)

The expression of the exact solution is given by

$$X(t) = e^{(t-t_0)A} X_0 e^{(t-t_0)A^T} + \int_{t_0}^t e^{(t-\tau)A} BB^T e^{(t-\tau)A^T} d\tau. \quad (4.1)$$

4.1. Projecting by using the extended global Arnoldi process. We will consider extended global Krylov subspaces associated to the pair (A, B) and defined as follows

$$\mathbb{K}_m(A, B) = \text{span}(A^{-m}, \dots, A^{-1}B, B, AB, \dots, A^{m-1}B). \quad (4.2)$$

Notice that

$$\mathbb{K}_m(A, B) = \mathcal{K}_m(A, B) + \mathcal{K}_m(A^{-1}, A^{-1}B),$$

where $\mathcal{K}_m(A, B)$ is the global Krylov subspace associated to the pair (A, B) . To compute an F-orthonormal basis of $\mathbb{K}_m(A, B)$, we can use the extended global Arnoldi algorithm defined as follows [13]

If the upper triangular 2×2 matrices $H_{j+1,j}$ ($j = 1, \dots, m$) are full rank, then Algorithm 3 computes an F-orthonormal basis of the global extended Krylov subspace $\mathbb{K}_m(A, B)$; the obtained $n \times 2mp$ matrix $\mathbb{V}_m = [V_1, \dots, V_m]$ is F-orthonormal

$$\mathbb{V}_m^T \diamond \mathbb{V}_m = I_{2m}.$$

Let $\mathbb{T}_m = \mathbb{V}_m^T \diamond (A\mathbb{V}_m) = [T_{i,j}]$ with $T_{i,j} = V_i^T \diamond (AV_j) \in \mathbb{R}^{2 \times 2}$, $i, j = 1, \dots, m$. Then it can be shown that \mathbb{T}_m is a $2m \times 2m$ upper block Hessenberg matrix whose elements can be obtained from the matrix-coefficients $H_{i,j}$ computed by the extended global Arnoldi algorithm. Let $\tilde{\mathbb{T}}_m = \mathbb{V}_{m+1}^T \diamond (A\mathbb{V}_m)$, then \mathbb{T}_m

Algorithm 3 The extended global Arnoldi algorithm

1. Compute the global QR decomposition: $[B, A^{-1}B] = V_1(R \otimes I_p)$
 2. For $j = 1, \dots, m$
 - (a) Set $V_j^{(1)}$: the first p columns of V_j and $V_j^{(2)}$: the second p columns of V_j ,
 - (b) Set $\mathbb{V}_j = [\mathbb{V}_{j-1}, V_j]$ and $U = [AV_j^{(1)}, A^{-1}V_j^{(2)}]$,
 - (c) F-orthogonalize U w.r. to \mathbb{V}_j to get V_{j+1} , i.e.
 - (d) For $i = 1, 2, \dots, j$
 - i. $H_{i,j} = V_i^T \diamond U$,
 - ii. $U = U - V_i(H_{i,j} \otimes I_p)$
 - (e) EndFor
 3. Compute the QR decomposition $U = V_{j+1}(H_{j+1,j} \otimes I_p)$
 4. EndFor
-

can be obtained from $\tilde{\mathbb{T}}_m$ by deleting the last 2 rows of $\tilde{\mathbb{T}}_m$.

We have the following algebraic relations [13].

$$A\mathbb{V}_m = \mathbb{V}_{m+1}(\tilde{\mathbb{T}}_m \otimes I_p) \quad (4.3)$$

$$= \mathbb{V}_m(\mathbb{T}_m \otimes I_p) + V_{m+1}(T_{m+1,m}E_m^T \otimes I_p), \quad (4.4)$$

where $E_m^T = [0, 0, \dots, I_2]$ is the matrix of the last two rows of the identity matrix I_{2m} .

Let $X_m(t)$ be the desired low-rank approximate solution given as

$$X_m(t) = \mathbb{V}_m(Y_m(t) \otimes I_p)\mathbb{V}_m^T, \quad t \in [t_0, T_f], \quad (4.5)$$

where $Y_m(t) \in \mathbb{R}^{2m \times 2m}$, solves the low dimensional differential Lyapunov equation

$$\dot{Y}_m(t) - \mathbb{T}_m Y_m(t) - Y_m(t)\mathbb{T}_m^T - B_m B_m^T = 0, \quad t \in [t_0, T_f], \quad (4.6)$$

with $B_m = \mathcal{V}_m^T \diamond B$. Notice that

$$[B, A^{-1}B] = V_1(R \otimes I_p), \quad \text{and } B_m = r_{1,1}e_1^{(2m)}, \quad (4.7)$$

with $R = [r_{i,j}]$, $1 \leq i, j \leq 2$ and $e_1^{(2m)}$ is the first vector of the canonical basis of \mathbb{R}^{2m} .

The low-dimensional differential Lyapunov equation (4.6) will be solved by using some classical linear differential equation solvers.

In order to limit the computational effort, we give an upper of the norm of the residual that will allow to stop the iterations without explicitly forming $X_m(t)$ which will be given only at the end of the process.

THEOREM 4.1. *Let $\mathcal{R}_m(t)$ be the residual obtained at step m , then we have*

$$\|\mathcal{R}_m(t)\|_F \leq \sqrt{2} \|T_{m+1,m}E_m^T Y_m(t)\|_F, \quad t \in [t_0, T_f]. \quad (4.8)$$

Proof. Using (4.3) and (4.5), the residual $\mathcal{R}_m(t) = \dot{X}_m(t) - AX_m(t) - X_m(t)A^T - BB^T$ is expressed as

$$\begin{aligned} \mathcal{R}_m(t) &= \mathbb{V}_m(\dot{Y}_m(t) \otimes I_p)\mathbb{V}_m^T - [\mathbb{V}_m(\mathbb{T}_m \otimes I_p) + V_{m+1}(T_{m+1,m}E_m^T \otimes I_p)](Y_m(t) \otimes I_p)\mathbb{V}_m^T \\ &\quad - \mathbb{V}_m(Y_m(t) \otimes I_p)[\mathbb{V}_m(\mathbb{T}_m \otimes I_p) + V_{m+1}(T_{m+1,m}E_m^T \otimes I_p)]^T - BB^T. \end{aligned}$$

Therefore, using the fact that Y_m is solution of the low dimensional differential problem (4.6), the residual can be expressed as follows

$$\mathcal{R}_m(t) = \mathbb{V}_{m+1} \left(\begin{bmatrix} 0 & Y_m(t)E_m T_{m+1,m}^T \\ T_{m+1,m}E_m^T Y_m(t) & 0 \end{bmatrix} \otimes I_p \right) \mathbb{V}_{m+1}^T.$$

Therefore, applying Proposition 2.3, we get for any $t \in [t_0, T_f]$ the following upper bound

$$\|\mathcal{R}_m(t)\|_F^2 \leq 2 \|T_{m+1,m}E_m^T Y_m(t)\|_F^2. \quad (4.9)$$

□

4.2. Solving the low dimensional differential Lyapunov equation. We have now to solve the low dimensional differential Lyapunov equation (4.6) by some integration method such as the well known Backward Differentiation Formula (BDF). We notice that BDF is especially used for the solution of stiff differential equations.

At each time t_k , let $Y_{m,k}$ denote the approximation of $Y_m(t_k)$, where Y_m is a solution of (4.6). Then, the new approximation $Y_{m,k+1}$ of $Y_m(t_{k+1})$ obtained at step $k+1$ by l -step BDF is defined by the implicit relation

$$Y_{m,k+1} = \sum_{i=0}^{l-1} \alpha_i Y_{m,k-i} + h_k \beta \mathcal{F}(Y_{m,k+1}), \quad (4.10)$$

where $h_k = t_{k+1} - t_k$ is the step size, α_i and β_i are the coefficients of the BDF method as listed in Table 4.1 and $\mathcal{F}(X)$ is given by

$$\mathcal{F}(Y) = \mathbb{T}_m Y + Y \mathbb{T}_m^T + B_m B_m^T.$$

| l | β | α_0 | α_1 | α_2 |
|-----|---------|------------|------------|------------|
| 1 | 1 | 1 | | |
| 2 | 2/3 | 4/3 | -1/3 | |
| 3 | 6/11 | 18/11 | -9/11 | 2/11 |

TABLE 4.1

Coefficients of the l -step BDF method with $l \leq 3$.

The approximate $Y_{m,k+1}$ solves the following matrix equation

$$-Y_{m,k+1} + h_k \beta (\mathbb{T}_m Y_{m,k+1} + Y_{m,k+1} \mathbb{T}_m^T + B_m B_m^T) + \sum_{i=0}^{l-1} \alpha_i Y_{m,k-i} = 0,$$

which can be written as the following algebraic Lyapunov matrix equation

$$\mathcal{T}_m Y_{m,k+1} + Y_{m,k+1} \mathcal{T}_m^T + \mathcal{B}_{m,k} \mathcal{B}_{m,k}^T = 0. \quad (4.11)$$

We assume that at each time t_k , the approximation $Y_{m,k}$ is factorised as a low rank product $Y_{m,k} \approx Z_{m,k} Z_{m,k}^T$, where $Z_{m,k} \in \mathbb{R}^{n \times m_k}$, with $m_k \ll n$. In that case, the coefficient matrices appearing in (4.11) are given by

$$\mathcal{T}_m = h_k \beta \mathbb{T}_m - \frac{1}{2} I \text{ and } \mathcal{B}_{m,k+1} = [\sqrt{h_k \beta} B_m, \sqrt{\alpha_0} Z_{m,k}, \dots, \sqrt{\alpha_{l-1}} Z_{m,k+1-l}].$$

The Lyapunov matrix equation (4.11) can be solved by applying direct methods based on Schur decomposition such as the Bartels-Stewart algorithm [2, 11].

In the following algorithm, we summarise the main steps of the extended global Arnoldi method for solving the differential Lyapunov matrix equation (1.2).

Algorithm 4 The extended global Arnoldi for differential Lyapunov equations (EGA-BDF)

1. Inputs: Coefficient matrices A, B , a maximum number of extended Arnoldi iteration m_{max} and a tolerance tol .
 2. For $m = 1, \dots, m_{max}$
 - (a) Apply the extended global Arnoldi Algorithm to the pair (A, B) to get an F-orthonormal matrix $\mathbb{V}_m = [V_1, \dots, V_m]$ and the upper block Hessenberg matrix \mathbb{T}_m .
 - (b) Solve the low dimensional problem (4.6) by the BDF method.
 - (c) If $\mathcal{R}_m(t) < tol$ stop and compute the obtained approximate solution
 3. EndFor.
-

4.3. Using the approximation of the exponential of a matrix. In this subsection, we will see how to use the expression (4.1) to get low rank approximate solutions. It is known [15, 18] that for any square matrix A , we have the Cauchy's integral representation

$$f(A) = \frac{1}{2\pi i} \int_{\Gamma} f(\lambda)(\lambda I - A)^{-1} d\lambda, \quad (4.12)$$

where f is an analytic function on and inside a closed contour $\Gamma \subset \mathbb{C}$ that encloses the spectrum $\sigma(A)$. Then

$$f(A)B = \frac{1}{2\pi i} \int_{\Gamma} f(\lambda)(\lambda I - A)^{-1} B d\lambda. \quad (4.13)$$

The approximation of the product $f(A)B$ is a very important issue; see [3, 10, 17, 24]. Using the extended global Arnoldi algorithm, we can show [19] that

$$(\lambda I - A)^{-1} B \approx \mathbb{V}_m ((\lambda I - \mathbb{T}_m)^{-1} r_{11} e_1^{2m} \otimes I_p),$$

where \mathbb{V}_m is the F-orthonormal matrix obtained from the extended global Arnoldi process applied to the pair (A, B) and $\mathbb{T}_m = \mathbb{V}_m^T \diamond (A \mathbb{V}_m)$. Then

$$(\lambda I - A)^{-1} B \approx r_{11} \mathbb{V}_m ((\lambda I - \mathbb{T}_m)^{-1} \otimes I_p) \tilde{E}_1, \quad (4.14)$$

where $\tilde{E}_1 = e_1^{2m} \otimes I_p$. Therefore, if the contour Γ also contains the spectrum of \mathbb{T}_m , (which is the case for example if we choose the contour of field of values of the matrix A) we get

$$f(A)B \approx r_{11} \mathbb{V}_m \frac{1}{2\pi i} \int_{\Gamma} f(\lambda) ((\lambda I - \mathbb{T}_m)^{-1} \otimes I_p) \tilde{E}_1 d\lambda, \quad (4.15)$$

which can be written as

$$f(A)B \approx r_{11} \mathbb{V}_m \frac{1}{2\pi i} \int_{\Gamma} f(\lambda) ((\lambda I - \mathbb{T}_m)^{-1} \otimes I_p) \tilde{E}_1 d\lambda = r_{11} \mathbb{V}_m (f(\mathbb{T}_m) \otimes I_p) \tilde{E}_1. \quad (4.16)$$

Using the fact that $\tilde{E}_1 = e_1^{2m} \otimes I_p$, we get

$$f(A)B \approx r_{11} \mathbb{V}_m (f(\mathbb{T}_m) e_1^{2m} \otimes I_p). \quad (4.17)$$

Notice that using some Kronecker product relations, the expression (4.15) can also be written

$$f(A)B \approx \frac{1}{2\pi i} r_{11} \mathbb{V}_m \int_{\Gamma} f(\lambda) \left(\lambda I - [\mathbb{T}_m \otimes I_p]^{-1} \right) \tilde{E}_1 d\lambda, \quad (4.18)$$

and then

$$f(A)B \approx r_{11} \mathbb{V}_m f(\mathbb{T}_m \otimes I_p) \tilde{E}_1. \quad (4.19)$$

The two expressions on the right hand sides in (4.17) and (4.19) are the same. Applying these results to the function $f(x) = e^x$, we get the approximation to the exponential appearing in the expression of the exact solution (4.1)

$$e^{(t-\tau)A}B \approx r_{11} \mathbb{V}_m (e^{(t-\tau)\mathbb{T}_m} e_1^{2m} \otimes I_p). \quad (4.20)$$

Assuming that $X_0 = 0$, we consider approximations $X_m(t)$ to the solution (4.1) as follows

$$X_m(t) = \int_{t_0}^t Z_m(\tau) Z_m(\tau)^T d\tau, \quad (4.21)$$

where

$$Z_m(\tau) = r_{11} \mathbb{V}_m (e^{(t-\tau)\mathbb{T}_m} e_1^{2m} \otimes I_p). \quad (4.22)$$

Hence, from (4.21) and (4.22), we get

$$X_m(t) = \mathbb{V}_m (G_m(t) \otimes I_p) \mathbb{V}_m^T, \quad (4.23)$$

where

$$G_m(t) = \int_{t_0}^t \tilde{G}_m(\tau) \tilde{G}_m(\tau)^T d\tau, \quad (4.24)$$

with $\tilde{G}_m(\tau) = r_{11} e^{(t-\tau)\mathbb{T}_m} e_1^{2m}$. So, to compute the approximation $X_m(t)$, we need to compute the integral (4.24) which will be done by using a quadrature formula.

In the sequel, we state some theoretical results on the residual norm and on the error of the method proposed in this section. These theorems are an adaptation of those stated in [12] to the extended global Arnoldi projection spaces. The next theorem states that the matrix function G_m is solution of a low dimensional differential Lyapunov equation.

THEOREM 4.2. *The function G_m defined by the relation (4.24) satisfies the following differential Lyapunov equation,*

$$\dot{G}_m(t) = \mathbb{T}_m G_m(t) + G_m(t) \mathbb{T}_m^T + r_{11}^2 e_1^{2m} (e_1^{2m})^T. \quad (4.25)$$

Proof. The proof can easily be obtained by deriving the expression (4.24). \square

Next, we give a result that allows us the computation of the norm of the residual.

THEOREM 4.3. *Let $X_m(t) = \mathbb{V}_m (G_m(t) \otimes I_p) \mathbb{V}_m^T$ be the approximation obtained at step m . Then the residual $\mathcal{R}_m(t)$ satisfies*

$$\| \mathcal{R}_m(t) \|_F \leq \| T_{m+1,m} \|_F \| \bar{G}_m(t) \|_2, \quad (4.26)$$

where $\bar{G}_m(t)$ is the $2 \times 2mp$ matrix obtained by extracting the last two rows of $G_m(t)$.

THEOREM 4.4. Let $X_m(t)$ be the approximate solution given by (4.23). Then we have

$$\dot{X}_m(t) = AX_m(t) + X_m(t)A^T + \mathcal{L}_m. \quad (4.27)$$

where $\mathcal{L}_m = BB^T - L_m - L_m^T$ with $L_m(t) = V_{m+1}T_{m+1,m}(\bar{G}_m(t) \otimes I_p)\nabla_m^T$. The error $\mathcal{E}_m(t) = X(t) - X_m(t)$ satisfies the following equation

$$\dot{\mathcal{E}}_m(t) = A\mathcal{E}_m(t) + \mathcal{E}_m(t)A^T - \mathcal{R}_m(t), \quad (4.28)$$

and then

$$\mathcal{E}_m(t) = e^{(t-t_0)A}\mathcal{E}_{m,0}e^{(t-t_0)A^T} + \int_{t_0}^t e^{(t-\tau)A}\mathcal{R}_m(\tau)e^{(t-\tau)A^T}d\tau, \quad t \in [t_0, T_f]. \quad (4.29)$$

where $\mathcal{E}_{m,0} = \mathcal{E}_m(t_0)$.

Proof. The proof of (4.28) is obtained by using the expression (4.23) of the approximate solution $X_m(t)$ and the relation (4.25). The expression (4.29) of the error is easily derived by extracting the initial problem (1.2) from the expression of the residual $\mathcal{R}_m(t)$. \square

THEOREM 4.5. Assume that $X(t_0) = X_m(t_0)$, then we have the following upper bound

$$\|\mathcal{E}_m(t)\|_F \leq \|T_{m+1,m}\|_F \|\bar{G}_m\|_\infty \frac{e^{2(t-t_0)\mu_2(A)} - 1}{2\mu_2(A)}, \quad \forall t \in [t_0, T_f], \quad (4.30)$$

with $\mu_2(A) = \frac{1}{2}\lambda_{\max}(A + A^T)$ is the 2-logarithmic norm and $\|\bar{G}_m\|_\infty = \max_{\tau \in [t_0, t]} \|\bar{G}_m(\tau)\|_2$ where $\bar{G}_m(\tau)$ is the $2 \times 2mp$ matrix obtained by extracting the last two rows of $G_m(\tau)$.

Proof. Using the expression (4.29) of $\mathcal{E}_m(t)$ and the fact that $\|e^{tA}\| \leq e^{\mu_2(A)t}$, we get

$$\begin{aligned} \|\mathcal{E}_m(t)\| &\leq \|T_{m+1,m}\|_F \|\bar{G}_m\|_\infty \int_{t_0}^t e^{2(t-\tau)\mu_2(A)}d\tau \\ &\leq \|T_{m+1,m}\|_F \|\bar{G}_m\|_\infty e^{2t\mu_2(A)} \int_{t_0}^t e^{-2\tau\mu_2(A)}d\tau \\ &= \|T_{m+1,m}\|_F \|\bar{G}_m\|_\infty \frac{e^{2(t-t_0)\mu_2(A)} - 1}{2\mu_2(A)}, \end{aligned}$$

which gives the desired result. \square

REMARK 1. Note that if the matrix A is not invertible, then we can replace the extended global Arnoldi algorithm by the global Arnoldi. In this case, all the relations stated in this section still hold with some modifications. From a numerical point of view, when feasible, the extended global Arnoldi method has to be preferred.

5. Numerical examples. In this section, we apply the approaches presented in this work to the two problems that are considered in this paper. All computations were performed on a laptop with an Intel Core i7 processor and 8GB of RAM. The algorithms were coded in Matlab R2014b.

Example 1. In the first case, we consider the generalized Lyapunov equation

$$\begin{cases} \dot{X}(t) = A_1 X(t) B_1 + A_2 X(t) B_2 + C, \\ X(t_0) = X_0, \end{cases} \quad (5.1)$$

where A_1 and A_2 are matrices obtained from the 5-point discretization on the unit square $[0, 1] \times [0, 1]$ of the operators

$$L_{A_1} = \Delta u - f_1(x, y) \frac{\partial u}{\partial x} + f_2(x, y) \frac{\partial u}{\partial y} + g_1(x, y),$$

and

$$L_{A_2} = \Delta u - f_3(x, y) \frac{\partial u}{\partial x} + f_4(x, y) \frac{\partial u}{\partial y} + g_2(x, y),$$

respectively, with homogeneous Dirichlet boundary conditions. The number of inner grid points in each direction is n_0 and the size of the matrix A was $n \times n$, where $n = n_0^2$. Here we set $f_1(x, y) = 10xy$, $f_2(x, y) = e^{x^2y}$, $f_3(x, y) = 100y$, $f_4(x, y) = x^2y$, $g_1(x, y) = 20y$ and $g_2(x, y) = xy$. The time interval considered was $[0, 2]$, with the initial condition $X_0 = 0_{n \times p}$. The coefficients of matrices $B_i \in \mathbb{R}^{p \times p}$, $i = 1, 2$, and $C \in \mathbb{R}^{n \times p}$ have been randomly generated. For our tests, we set $p = 2$. We applied the global-Galerkin algorithm (GG) described by Algorithm 2, which consists in projecting the differential problem onto a global Arnoldi subspace and then solve it by a quadrature method to compute the exponential form (3.10) of the solution. The Frobenius norm of the residual at final time is then computed and while the tolerance is not met, we repeat the process increasing the dimension of the projection subspace. In order to confirm that our approach produces reliable results, we compared the outputs obtained by the GG method to the one given by a direct computation of the exponential form (1.4). This was done by vectorizing our DLE, stacking the columns of X one on top of each other and using a scaling and squaring method as implemented in the MATLAB `expm` function. This method is not suited for large-scale problems. We chose a size of 100×100 for the matrices A_1 and A_2 . In Figure 5.1, we plotted the component X_{11} of the solution obtained by the GG method to the one provided by the direct computation of the exponential form (1.4), on the time interval $[0, 2]$, for $size(A_i) = 100 \times 100$, $i = 1, 2$.

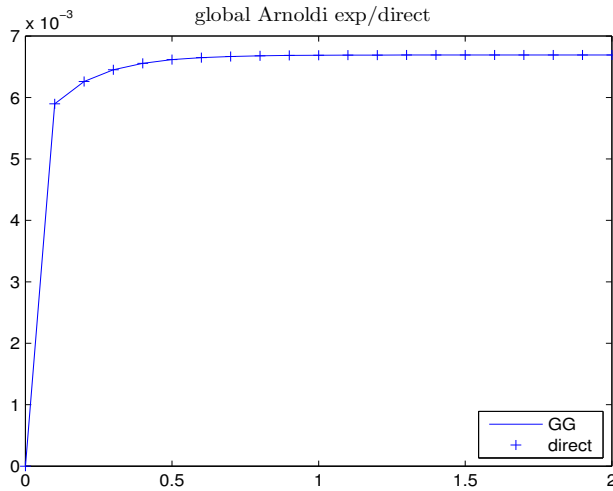


FIGURE 5.1. Values of $X_{11}(t)$ for $t \in [0, 2]$

We observe that both considered methods give similar results in terms of accuracy. The error norms $\|X_{GG}(t_f) - X_{direct}(t_f)\|$ at final time $t_f = 2$ was of order 10^{-8} and the Frobenius relative norm of the residual was of order 10^{-9} . The runtimes were respectively 2.3 s for the GG and 8.5 s for the direct method.

In Table 5.1, we give the obtained runtimes in seconds and the relative Frobenius residual norms at final time for the resolution of Equation (5.1) for $t \in [0, 2]$. In order to minimize the computational time, the resolution of the projected differential equation was done every 50 steps of the global Arnoldi algorithm.

| size(A_i) | GG | Direct method | Residual norm |
|--------------------|--------|---------------|--------------------------------------|
| 900×900 | 11.9 s | 398.2 s | $\mathcal{O}(10^{-9})$ ($m = 200$) |
| 2500×2500 | 35.4s | > 1000 s | $\mathcal{O}(10^{-9})$ ($m = 300$) |
| 6400×6400 | 78.2s | > 1000s | $\mathcal{O}(10^{-9})$ ($m = 350$) |

TABLE 5.1
runtimes and residual norms for GG and direct method

The results in Table 5.1 illustrate that the GG method allows the numerical resolution of equation (1.3) in cases for which direct methods are not suitable for being too slow or due to memory limitations. In Figure 5.2, we plotted the norm of the residual at final time $t = 2$ for the GG method for $\text{size}(A_i) = 2500 \times 2500$ in function of the number m of global Arnoldi iterations.

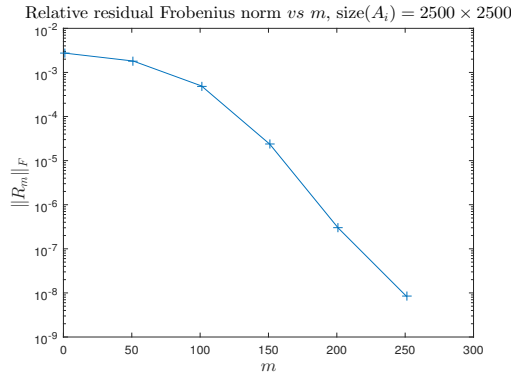


FIGURE 5.2. Residual norms vs the number of global Arnoldi iterations m

In the next examples, we consider the low-rank Lyapunov equation

$$\begin{cases} \dot{X}(t) = AX(t) + X(t)A^T + BB^T; (DLE) \\ X(t_0) = X_0, \quad t \in [t_0, T_f], \end{cases} \quad (5.2)$$

Provided that A is not singular, we used the Extended global Arnoldi BDF (EGA-BDF) and the EGA-exp methods as introduced in Sections 4.2 and 4.3 respectively. The steps of the EGA-exp method only differs from Algorithm 4 in the way the projected equation is solved, using the exponential method, described in Section 4.3 instead of the BDF method.

Example 2. The matrix A was obtained from the 5-point discretization of the operators

$$L_A = \Delta u - f_1(x, y) \frac{\partial u}{\partial x} + f_2(x, y) \frac{\partial u}{\partial y} + g_1(x, y),$$

on the unit square $[0, 1] \times [0, 1]$ with homogeneous Dirichlet boundary conditions. The number of inner grid points in each direction is n_0 and the dimension of the matrix A was $n = n_0^2$. Here we set $f_1(x, y) = 10xy$, $f_2(x, y) = e^{x^2y}$ and $g_1(x, y) = 20y$. The coefficients of $B \in \mathbb{R}^{n \times p}$ were randomly generated. The time interval considered was $[0, 2]$ and the initial condition $X_0 = X(0)$ was chosen as the low rank product $X_0 = Z_0 Z_0^T$, where $Z_0 = 0_{n \times p}$.

In Table 5.2, we give the obtained runtimes in seconds, for the resolution of Equation (1.2) for $t \in [0, 2]$, with a timestep $h = 0.001$ and the Frobenius norm of the residual at the final time. The rank of matrix B was set to $p = 2$. The projected differential equations were solved every 5 iterations of the extended Arnoldi algorithm.

| size(A) | EGA-exp | GA-BDF(1) | Residual norm |
|------------------------|---------|-----------|-------------------------------------|
| 40000×40000 | 14.3 s | 21.5 s | $\mathcal{O}(10^{-9})$ ($m = 35$) |
| 250000×250000 | 166 s | 180.5 s | $\mathcal{O}(10^{-8})$ ($m = 45$) |

TABLE 5.2
runtimes and residual norms for EGA-exp and EGA-BDF(1)

The figures in Table 5.2 illustrate the ability to numerically solve large scale low-rank differential Lyapunov equations. The EGA-exp method is slightly faster than the EGA-BDF(1) method and both methods performed similarly in terms of accuracy.

Example 3. In this last example, we applied the EGA-exp and EGA-BDF(1) methods to the well-known problem Optimal Cooling of Steel Profiles. The matrices were extracted from the IMTEK collection ¹. We compared both methods for problem sizes $n = 5177$ and $n = 20209$, on the time interval $[0, 1000]$. The initial value X_0 was chosen as $X_0 = 0$ and the timestep was set to $h = 1$ for the BDF(1) integration scheme. The tolerance for the Arnoldi stop test was set to 10^{-7} for both methods. The integration of the projected differential equations were done every 2 steps of the Extended global Arnoldi algorithm.

| size(A) | EGA-exp | EGA-BDF(1) | Residual norms |
|----------------------|---------|------------|-------------------------------------|
| 5177×5177 | 17.6 s | 19.4 s | $\mathcal{O}(10^{-7})$ ($m = 50$) |
| 20209×20209 | 70 s | 82.5 s | $\mathcal{O}(10^{-7})$ ($m = 70$) |

TABLE 5.3
Optimal Cooling of Steel Profiles: runtimes and residual norms for EGA-exp and EGA-BDF(1)

In Table 5.3, we listed the obtained runtimes which again showed that both methods are interesting in terms of execution time and are similar in terms of accuracy.

6. Conclusion. We presented in the present paper different new approaches for computing approximate solutions to large scale differential matrix equations. These approaches are based on projection onto matrix Krylov subspaces using the global and the extended global Arnoldi algorithms. For problems with full rank right hand sides, the problem reduces to the computation of solutions of differential linear systems of equations by classical methods. In the second part of this work, we considered a differential Lyapunov matrix equation with a decomposed low rank hand sides. The initial problem was projected onto matrix Krylov subspaces to get low dimensional differential Lyapunov equation that is solved by the classical BDF methods. The numerical examples provided in this work showed that both methods are promising for large-scale problems.

¹<https://portal.uni-freiburg.de/imteksimulation/downloads/benchmark>

References.

- [1] H. ABOU-KANDIL, G. FREILING, V. IONESCU, G. JANK, *Matrix Riccati Equations in Control and Systems Theory*, in Systems & Control Foundations & Applications, Birkhauser, (2003).
- [2] R.H. BARTELS, G.W. STEWART, *Algorithm 432: Solution of the matrix equation $AX+XB=C$* , Circ. Syst. Signal Proc., 13 (1972), 820–826.
- [3] M. BENZI, N. RAZOUK, *Decay rates and $O(n)$ algorithms for approximating functions of sparse matrices*, Electr. Trans. Numer. Anal., 28 (2007), 16–39.
- [4] A. BOUHAMIDI AND K. JBILOU, A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications, Applied Mathematics and Computation, 206(2), (2008), 687–694.
- [5] R. Bouyouli, K. Jbilou, R. Sadaka and H. Sadok, Convergence properties of some block Krylov subspace methods for multiple linear systems. J. Comput. Appl. Math., 196 (2006), 498–511.
- [6] C. BREZINSKI, Computational Aspects of Linear Control, Kluwer, Dordrecht, 2002.
- [7] M. J. CORLESS AND A. E. FRAZHO, Linear systems and control - An operator perspective, Pure and Applied Mathematics. Marcel Dekker, New York-Basel, 2003.
- [8] B.N. DATTA, Numerical Methods for Linear Control Systems Design and Analysis, Elsevier Academic Press, (2003).
- [9] V. DRUSKIN, L. KNIZHNERMAN, Extended Krylov subspaces: approximation of the matrix square root and related functions, SIAM J. Matrix Anal. Appl., 19(3) (1998), 755–771.
- [10] E. GALLOPOULOS AND Y. SAAD, Efficient solution of parabolic equations by Krylov approximation methods, SIAM J. Sci. Statist. Comput., 13 (1992), 1236–1264.
- [11] G. H. GOLUB AND S. NASH AND C. VAN LOAN A Hessenberg-Schur method for the problem $AX+XB=C$, IEEE Trans. Autom. Contr., AC-24(1979), 909-913.
- [12] M. HACHED AND K. JBILOU Numerical solutions to large-scale differential Lyapunov matrix equations, Numer. Algorithms, 79 (3) (2018), 747-757.
- [13] M. HEYOUNI, Extended Arnoldi methods for large Sylvester matrix equations, App. Num. Math., 60(11) (2010) 1171–1182.
- [14] M. HEYOUNI AND K. JBILOU, An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation, Elect. Trans. Num. Anal., 33(2009,) 53–62.
- [15] N. J. HIGHAM, Functions of Matrices, Theory and Computation, SIAM, Philadelphia, 2008.
- [16] N. J. HIGHAM, *The scaling and squaring method for the matrix exponential revisited*. SIAM J. Matrix Anal. Appl., 26(4) (2005), 1179–1193.
- [17] M. HOCHBRUCK AND C. LUBICH, On Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal., 34 (1997), 1911–1925.
- [18] R.A. HORN AND C.R. JOHNSON, Topics in matrix analysis, Cambridge University Press, 1991.
- [19] K. JBILOU A. MESSAOUDI H. SADOK *Global FOM and GMRES algorithms for matrix equations*, Appl. Num. Math 31(1999), 49–63.
- [20] K. JBILOU AND A. MESSAOUDI, A computational method for symmetric Stein matrix equations. In Numerical Linear Algebra in Signals, Systems and Control, 80(2011), 295–311.
- [21] C.B. MOLER, C.F. VAN LOAN, *Nineteen Dubious Ways to Compute the Exponential of a Matrix*, SIAM Review 20, 1978, pp. 801–836. Reprinted and updated as "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later," SIAM Review 45(2003), 3–49.
- [22] B. C. MOORE, *Principal component analysis in linear systems: controllability, observability and model reduction*, IEEE Trans. Automatic Contr., AC-26 (1981), 17–32.
- [23] Y. SAAD, *Numerical solution of large Lyapunov equations*, in Signal Processing, Scattering, Operator Theory and Numerical Methods. Proceedings of the international symposium MTNS-89, vol. 3, M.A. Kaashoek, J.H. van Schuppen and A.C. Ran, eds., Boston, 1990, Birkhauser, 503–511.

- [24] Y. SAAD, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), 209–228.
- [25] V. SIMONCINI, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comp., 29(3) (2007), 1268–1288.