HAL
open science

NUMERICAL SOLUTION OF LARGE-SCALE LYAPUNOV EQUATIONS AND MODEL REDUCTION PROBLEMS USING AN EXTENDED-RATIONAL BLOCK ARNOLDI METHOD

O. Abidi, Khalide Jbilou, Mohamed Amine Hamadi

To cite this version:

O. Abidi, Khalide Jbilou, Mohamed Amine Hamadi. NUMERICAL SOLUTION OF LARGE-SCALE LYAPUNOV EQUATIONS AND MODEL REDUCTION PROBLEMS USING AN EXTENDED-RATIONAL BLOCK ARNOLDI METHOD. 2020. hal-02568939

HAL Id: hal-02568939

https://hal.science/hal-02568939v1

Preprint submitted on 10 May 2020

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 SOLUTION OF LARGE-SCALE LYAPUNOV EQUATIONS AND MODEL REDUCTION PROBLEMS USING AN EXTENDED-RATIONAL BLOCK ARNOLDI METHOD

O.ABIDI*, M.A. HAMADI *†, K. JBILOU *†, Y. KAOUANE†, AND A. RATNANI†

Abstract. In this paper, we propose a new block Krylov-type subspace method for model reduction in large scale dynamical systems. We also show how this method can be used to extract an approximate low rank solution of large-scale Lyapunov equations. We project the initial problem onto a new subspace, generated as combination of Rational and Polynomial block Krylov subspaces. Algebraic properties are given such as expressions of the error between the original and reduced transfer functions. Besides, a simplified expression of the residual, that allows us to compute the approximate solution in an efficient way, is established. Furthermore, we present an adaptive strategy of the interpolation points that will be used in the construction of our new block Krylov subspace. Numerical results are finally reported using some "benchmark examples" to confirm the performance of our method compared with other known methods.

Key words. dynamical systems, Krylov subspaces, transfer function, model order reduction, Lyapunov equations.

AMS subject classifications. 65F10, 65F30

1. Introduction.

In recent past, a various model reduction methods for Multiple Input Multiple Output system (MIMO, in short) have been explored. Some of them are based on the Krylov subspace methods (moment matching) while others use balanced truncation [1, 23, 28], Padé approximation [37], Optimal Hankel norm [19, 20].

We consider the following multi-input multi-output (MIMO) linear time-invariant (LTI, in short) system, described by the state-space equations

$$\left\{ \begin{array}{lcl} \dot{x}(t) & = & A\,x(t) + B\,u(t) \\ y(t) & = & C\,x(t), \end{array} \right. \tag{1.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t)$, $y(t) \in \mathbb{R}^p$ are the input and the output vectors of the system (1.1) respectively. The system matrices $B, C^T \in \mathbb{R}^{n\times p}$ and $A \in \mathbb{R}^{n\times n}$ are supposed to be large and sparse. The transfer function associated to the above system is given by :

$$F(s) = C\,(s\,I_n - A)^{-1}\,B. \tag{1.2}$$

The goal of model reduction is to produce a smaller order system with a state-space form :

$$\left\{ \begin{array}{lcl} \dot{x}_m(t) & = & A_m\,x_m(t) + B_m\,u(t) \\ y_m(t) & = & C_m\,x_m(t), \end{array} \right. \tag{1.3}$$

*Université du Littoral Côte d'Opale, 50 Rue F. Buisson, BP 699–62228 Calais cedex, France; abidi@lmpa.univ-littoral.fr and jbilou@univ-littoral.fr

†Université Mohammed VI polytechnique, Lot 660, Hay Moulay Rachid, Ben Guerir, 43150 Maroc; amine.hamadi@um6p.ma, kaouane.yassine@um6p.ma and ahmed.ratnani@um6p.ma

with the associated transfer function F_m given by

$$F_m(s) = C_m\,(s\,I_m - A_m)^{-1}\,B_m, \tag{1.4}$$

where $A_m \in \mathbb{R}^{m\times m}$ and B_m, $C_m^T \in \mathbb{R}^{m\times p}$, with $(m \ll n)$.

One of the most popular techniques used for model reduction methods, is based on interpolation and moment matching methods are treated e.g., in [4, 16, 21, 26, 29]. These methods use Polynomial block Krylov subspace

$$\mathbb{K}_m^{poly}(A, V) = \text{colspan}\{B, A\,B, A^2\,B, \ldots, A^{m-1}\,B\}.$$

Extended block Krylov subspace

$$\mathbb{K}_m^{poly}(A, V) = \text{colspan}\{A^{-m}\,B, \ldots, A^{-1}\,B, B, A\,B, A^2\,B, \ldots, A^{m-1}\,B\}.$$

or Rational block Krylov subspace

$$\mathbb{K}_m^{rat}(A, B) = \text{colspan}\{(s_1 I - A)^{-1}\,B, \ldots, \prod_{i=1}^{m}(s_i I - A)^{-1}\,B\},$$

where $\{s_1, \ldots, s_m\}$ are some selected complex shifts. The procedure of Rational Krylov subspace was originally proposed by Ruhe [35], in the context of approximating interior eigenvalues and has been used during the last years for model order reduction, see [21]. The aim of those methods is to build reduced models whose transfer function (1.4) interpolates the original system transfer function (1.2) at selected interpolation points. These methods enable us to construct a reduced order systems, at a short time and with appropriate computational requirements, via a projection using bases of a particular Krylov subspaces. In this paper, we are interested in the projection of the original problem (1.1) onto a new block Krylov subspace to obtain a low order model. The new block Krylov subspace called the "Extended-Rational" is defined as a combination of the rational and the polynomial block Krylov subspaces. One of the issues in this method is the selection of shifts that are used in the construction of the Extended-Rational block Krylov. Diverse methods have been proposed in the literature to construct the interpolation points see, e.g., [15, 21]. In [6, 24] authors proposed an Iterative Rational Krylov Algorithm (IRKA, in short) to compute a reduced order model satisfying the first order conditions for the $\mathcal{H}_2$ approximation.
The remainder of this paper is organised as follows. Section 2 describes the new block Krylov subspace projection framework and derives some algebraic properties. An adaptive strategy for the selection of the shifts, that are used in the construction of the Extended-Rational block Krylov subspace, and an expression of the error between the original transfer function and its approximation are presented in Section 3. In Section 4, we show how to extract a low rank approximate solution to Lyapunov equations. In Section 5 we treat the model reduction of second-order systems. Finally in Section 6, we provide some numerical examples to show the effectiveness of our method.

2. Block Extended-Rational method.

The general idea of this section is to provide a new subspace Krylov richer than the rational Krylov subspace. To this end, we introduce the Extended-Rational block Krylov subspace which is a subspace of $\mathbb{R}^n$, denoted by $\mathbb{K}_m(A, B)$, spanned by the columns of the matrices $A^k\,B$, $k = 0, \ldots, m-1$ and $(s_1 I - A)^{-1}\,B, (s_1 I - A)^{-1}(s_2 I - A)^{-1}\,B, \ldots, \prod_{i=1}^{m}(s_i I -$

$A)^{-1}\,B$. This subspace is defined by

$$\mathbb{K}_m(A,B) := \text{colspan}(\{\prod_{i=1}^{m}(s_i I - A)^{-1}\,B, \ldots, , (s_1 I - A)^{-1}\,B, B, A\,B, A^2\,B, \ldots, A^{m-1}\,B\}),$$

where $s_1, \ldots, s_m$ are some selected complex parameters. As in all projection methods we need to provide an orthonormal basis of the subspace where the projection occurs. For this, we introduce the Extended-Rational Arnoldi algorithm based on the Gram-Schmidt orthogonalization process. We start with the pair $\{B, (s_1 I - A)^{-1}\,B\}$ and we add simultaneously two vectors at a time, one multiplied by A and the other by $(s_j I - A)^{-1}\,B$ at each iteration. Here we need to mention that we can use the decomposition LU to compute the inverse of $(s_j I - A)$, but when we deal with large matrices we can use iterative solvers with preconditioners instead. The Extended-Rational block Arnoldi algorithm generates a sequence of blocks $\{V_1, \ldots, V_m\}$ of size $(n \times 2p)$, such that their columns form an orthonormal basis of the Extended-Rational block Krylov subspace $\mathbb{K}_m(A,B)$. The algorithm is defined as follows

Algorithm 1 The Extended-Rational (block) Arnoldi algorithm

- Inputs: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $\{s_1, \ldots, s_m\} \subset \mathbb{C}$ and m.
- Compute $[V_1, \Lambda] = QR([B, (s_1 I - A)^{-1}\,B])$, $\mathbb{V}_1 = [V_1]$.
- For $j = 1, \ldots, m$
  1. Set $V_j^{(1)}$: first p columns of V_j; $V_j^{(2)}$: second p columns of V_j.
  2. $\widetilde{V}_{j+1} = [A\,V_j^{(1)}, (s_j I - A)^{-1}\,V_j^{(2)}]$.
  3. Orthogonalize $\widetilde{V}_{j+1}$ with respect to to $\mathbb{V}_1, \ldots, \mathbb{V}_j$ to get V_{j+1}, i.e.,
     for $i = 1, 2, \ldots, j$
     $H_{i,j} = (V_i)^T\,\widetilde{V}_{j+1}$;
     $\widetilde{V}_{j+1} = \widetilde{V}_{j+1} - V_i\,H_{i,j}$;
     end for
  4. $[V_{j+1},\ H_{j+1,j}] = QR(\widetilde{V}_{j+1})$.
  5. $\mathbb{V}_{j+1} = [\mathbb{V}_j,\ V_{j+1}]$.

  End For.

The matrix $\mathbb{V}_m = [V_1, V_2, \ldots, V_m] \in \mathbb{R}^{n\times 2mp}$ with $V_i \in \mathbb{R}^{n\times 2p}$ have their columns mutually orthogonal provided that none of the upper triangular matrices $H_{j+1,j}$ are rank deficient. Hence, after m steps, Algorithm 1 builds an orthonormal basis $\mathbb{V}_m$ of the Extended-Rational block Krylov subspace $\mathbb{K}_m(A,B)$ and an upper block Hessenberg matrix $\mathbb{H}_m \in \mathbb{R}^{2mp\times 2mp}$ whose non zero blocks are the $H_{i,j}$. Note that each submatrix $H_{i,j}$ $(1 \le i \le j \le m)$ is of order $2p \times 2p$.
We consider the following notations :
$\mathbb{V}_m^o, \mathbb{V}_m^e$ are the matrices of $\mathbb{R}^{n\times mp}$ formed by the block columns of the odd and the even indices of the matrix $\mathbb{V}_m$.
$\bar{\mathbb{H}}_m^o, \bar{\mathbb{H}}_m^e$ are the matrices of $\mathbb{R}^{2(m+1)p\times mp}$ formed by the block columns of the odd and the even indices of the matrix $\bar{\mathbb{H}}_m \in \mathbb{R}^{2(m+1)p\times 2(m+1)p}$ an upper block Hessenberg matrix, defined as

$$\bar{\mathbb{H}}_m = \begin{bmatrix} \mathbb{H}_m \\ H_{m+1,m}E_m^T \end{bmatrix},$$

where $E_m = [0_{2p}, \ldots, 0_{2p}, I_{2p}] \in \mathbb{R}^{2p\times 2mp}$ and $H_{m+1,m} \in \mathbb{R}^{2p\times 2p}$ defined as

$$H_{m+1,m} = \begin{bmatrix} H^{(1,1)}_{m+1,m} & H^{(1,2)}_{m+1,m} \\ 0 & H^{(2,2)}_{m+1,m} \end{bmatrix}.$$

PROPOSITION 2.1. *Let $\mathbb{V}_{m+1} = [V_1, \ldots, V_{m+1}]$, by using the above notation we get*

$$A\mathbb{V}^o_m = \mathbb{V}_{m+1}\bar{\mathbb{H}}^o_m, \tag{2.1}$$

$$A\,\mathbb{V}_{m+1}\bar{\mathbb{H}}^e_m = \mathbb{V}_{m+1}\bar{\mathbb{H}}^e_m S_m - \mathbb{V}^e_m, \tag{2.2}$$

where $S_m = [s_1 I_p; \ldots; s_m I_p]$.

Proof. Assuming that $[V_1, \ldots, V_j]$ has been computed, we seek for $V_{j+1} = [V^{(1)}_{j+1}, V^{(2)}_{j+1}] \in \mathbb{R}^{n\times 2p}$. In Algorithm 1 a Gram-shmidt process is used in order to get V_{j+1}, we can sum up this process in the following result :

$$V_{j+1}H_{j+1,j} = [A\,V^{(1)}_j, (s_j I - A)^{-1}V^{(2)}_j] - \sum_{i=1}^{j} V_i\, H_{i,j}, \quad \text{for } j = 1, \ldots, m. \tag{2.3}$$

Let V_j, $H_{i,j}$ and $H_{j+1,j}$ be defined by the following partition form :

$$V_j = [V^{(1)}_j, V^{(2)}_j], \qquad H_{i,j} = \begin{bmatrix} H^{(1,1)}_{i,j} & H^{(1,2)}_{i,j} \\ H^{(2,1)}_{i,j} & H^{(2,2)}_{i,j} \end{bmatrix}, \qquad H_{j+1,j} = \begin{bmatrix} H^{(1,1)}_{j+1,j} & H^{(1,2)}_{j+1,j} \\ 0 & H^{(2,2)}_{j+1,j} \end{bmatrix},$$

we can rewrite (2.3) as

$$\left\{ \begin{array}{lcl} A\,V^{(1)}_j & = & \displaystyle\sum_{i=1}^{j+1} V_i \begin{bmatrix} H^{(1,1)}_{i,j} \\ H^{(2,1)}_{i,j} \end{bmatrix} \quad \text{with } H^{(21)}_{j+1,j} = 0 \\ (s_j I - A)^{-1}V^{(2)}_j & = & \displaystyle\sum_{i=1}^{j+1} V_i \begin{bmatrix} H^{(1,2)}_{i,j} \\ H^{(2,2)}_{i,j} \end{bmatrix} \end{array} \right\} \text{for } j = 1, \ldots, m \tag{2.4}$$

then as

$$\left\{ \begin{array}{lcl} A\,V^{(1)}_j & = & \displaystyle\sum_{i=1}^{j+1} V_i \begin{bmatrix} H^{(1,1)}_{i,j} \\ H^{(2,1)}_{i,j} \end{bmatrix} \quad \text{with } H^{(2,1)}_{j+1,j} = 0 \\ A\displaystyle\sum_{i=1}^{j+1} V_i \begin{bmatrix} H^{(1,2)}_{i,j} \\ H^{(2,2)}_{i,j} \end{bmatrix} & = & s_j \displaystyle\sum_{i=1}^{j+1} V_i \begin{bmatrix} H^{(1,2)}_{i,j} \\ H^{(2,2)}_{i,j} \end{bmatrix} - V^{(2)}_j \end{array} \right\} \text{for } j = 1, \ldots, m \tag{2.5}$$

finally, we get

$$A\mathbb{V}^o_m = \mathbb{V}_{m+1}\bar{\mathbb{H}}^o_m,$$

$$A\,\mathbb{V}_{m+1}\bar{\mathbb{H}}^e_m = \mathbb{V}_{m+1}\bar{\mathbb{H}}^e_m S_m - \mathbb{V}^e_m,$$

where $S_m = [s_1 I_p; \ldots; s_m I_p]$. □

Let $\mathbb{L}_m = [\mathbb{V}_m^T(s_1 I - A)^{-1}V_1, \mathbb{V}_m^T(s_2 I - A)^{-1}V_2, \ldots, \mathbb{V}_m^T(s_m I - A)^{-1}V_m] := [L_{i,j}]$, where $L_{i,j} = V_i^T(s_j I - A)^{-1}V_j \in \mathbb{R}^{2p\times 2p}$. Notice that we can verify that $\mathbb{L}_m$ has a block upper Hessenberg form, also the first p columns of $L_{i,j}$ with $i > j$, are zero i.e.,

$$L_{i,j} = V_i^T(s_j I - A)^{-1}V_j = \begin{bmatrix} 0 & L_{i,j}^{(12)} \\ 0 & L_{i,j}^{(22)} \end{bmatrix} \quad \text{with} \quad i > j.$$

The next result enable us to compute the block $(L_{i,j})_{i,j=1,\ldots,m}$ using just a few inner product and the blocks $(H_{i,j})$ which is already obtained form Algorithm 1.

PROPOSITION 2.2. *Let the blocks $L_{i,j}$ and $H_{i,j}$ as defined earlier for $i, j = 1, \ldots, m$ and consider the following partition*

$$L_{i,j} = [V_i^T(s_j I - A)^{-1}V_j^{(1)}, V_i^T(s_j I - A)^{-1}V_j^{(2)}],$$

where $V_j^{(1)}, V_j^{(2)}$ are the first and last p columns of V_j.
Then we have the following

1. $\mathbb{V}_m^T(s_1 I - A)^{-1}V_1^{(1)} = (\hat{e}_1\Lambda^{(1,2)} + \hat{e}_2\Lambda^{(2,2)})(\Lambda^{(1,1)})^{-1},$

2. $\mathbb{V}_m^T(s_j I - A)^{-1}V_j^{(2)} = \sum_{i=1}^{j+1} \tilde{E}_i \begin{bmatrix} H_{i,j}^{(1,2)} \\ H_{i,j}^{(2,2)} \end{bmatrix}. \quad for \quad j = 1, \ldots, m,$

3. $\mathbb{V}_{m+1}^T(s_{j+1} I - A)^{-1}V_{j+1}^{(1)} = (s_{j+1}\mathbb{V}_{m+1}^T(s_{j+1} I - A)^{-1}V_j^{(1)} - \hat{e}_j$
$- \sum_{i=1}^{j} \mathbb{V}_{m+1}^T(s_{j+1} I - A)^{-1}V_i \begin{bmatrix} H_{i,j}^{(1,1)} \\ H_{i,j}^{(2,1)} \end{bmatrix})(H_{j+1,j}^{(1,1)})^{-1},$

where $\hat{e}_j = \mathbb{V}_{m+1}^T V_j^{(1)} = e_j \otimes I_p$, e_i is the vectors of the canonical basis and $\tilde{E}_i = \mathbb{V}_m^T V_i = \tilde{e}_i \otimes I_{2p}$ with $\tilde{e}_i$ is the vectors of the canonical basis.

Proof. We know from Algorithm 1 that

$$[B, (s_1 I - A)^{-1}B] = V_1\Lambda = [V_1^{(1)},\, V_1^{(2)}] \begin{bmatrix} \Lambda^{(1,1)} & \Lambda^{(1,2)} \\ 0 & \Lambda^{(2,2)} \end{bmatrix},$$

then

$$B = V_1^{(1)}\Lambda^{(1,1)} \quad \text{and} \quad (s_1 I - A)^{-1}B = V_1^{(1)}\Lambda^{(1,2)} + V_1^{(2)}\Lambda^{(2,2)}.$$

Thus, if $\Lambda^{(1,1)}$ is non-singular, we obtain (1)

$$\mathbb{V}_m^T(s_1 I - A)^{-1}V_1^{(1)} = (\tilde{e}_1\Lambda^{(1,2)} + \tilde{e}_2\Lambda^{(2,2)})(\Lambda^{(1,1)})^{-1}.$$

In order to get (2), we use the partition (2.3)

$$(s_j I - A)^{-1}V_j^{(2)} = \sum_{i=1}^{j+1} V_i \begin{bmatrix} H_{i,j}^{(1,2)} \\ H_{i,j}^{(2,2)} \end{bmatrix},$$

then, we obtain

$$\mathbb{V}_m^T(s_j I - A)^{-1}V_j^{(2)} = \sum_{i=1}^{j+1} \tilde{E}_i \begin{bmatrix} H_{i,j}^{(1,2)} \\ H_{i,j}^{(2,2)} \end{bmatrix}.$$

where, $\tilde{E}_i = \mathbb{V}_m^T V_i = e_i \otimes I_{2p}$ and e_i is the vectors of the canonical basis.

For the last (3), we proceed as follow :

we know from the partition (1) that

$$A\,V_j^{(1)} = \sum_{i=1}^{j+1} V_i \begin{bmatrix} H_{i,j}^{(1,1)} \\ H_{i,j}^{(2,1)} \end{bmatrix} \quad \text{with} \quad H_{j+1,j}^{(2,1)} = 0,$$

$$V_{j+1}^{(1)} H_{j+1,j}^{(1,1)} = A\,V_j^{(1)} - \sum_{i=1}^{j} V_i \begin{bmatrix} H_{i,j}^{(1,1)} \\ H_{i,j}^{(2,1)} \end{bmatrix},$$

then we obtain

$$\begin{aligned} s_{j+1}V_j^{(1)} - V_{j+1}^{(1)} H_{j+1,j}^{(1,1)} &= s_{j+1}V_j^{(1)} - A\,V_j^{(1)} + \sum_{i=1}^{j} V_i \begin{bmatrix} H_{i,j}^{(1,1)} \\ H_{i,j}^{(2,1)} \end{bmatrix}, \\ V_{j+1}^{(1)} H_{j+1,j}^{(1,1)} &= s_{j+1}V_j^{(1)} - (s_{j+1}I - A)\,V_j^{(1)} - \sum_{i=1}^{j} V_i \begin{bmatrix} H_{i,j}^{(1,1)} \\ H_{i,j}^{(2,1)} \end{bmatrix}, \\ (s_{j+1}I - A)^{-1} V_{j+1}^{(1)} H_{j+1,j}^{(1,1)} &= s_{j+1}(s_{j+1}I - A)^{-1}V_j^{(1)} - V_j^{(1)} \\ &\quad - \sum_{i=1}^{j} (s_{j+1}I - A)^{-1} V_i \begin{bmatrix} H_{i,j}^{(1,1)} \\ H_{i,j}^{(2,1)} \end{bmatrix}. \end{aligned}$$

Then, if $(H_{j+1,j}^{(11)})^{-1}$ is non-singular we get

$$\begin{aligned} &\mathbb{V}_{m+1}^T (s_{j+1}I - A)^{-1} V_{j+1}^{(1)} = (s_{j+1}\mathbb{V}_{m+1}^T (s_{j+1}I - A)^{-1} V_j^{(1)} - \hat{e}_j \\ &- \sum_{i=1}^{j} \mathbb{V}_{m+1}^T (s_{j+1}I - A)^{-1} V_i \begin{bmatrix} H_{i,j}^{(1,1)} \\ H_{i,j}^{(2,1)} \end{bmatrix}) (H_{j+1,j}^{(1,1)})^{-1}, \end{aligned}$$

where $\hat{e}_j = \mathbb{V}_{m+1}^T V_j^{(1)} = e_j \otimes I_p$ and e_i is the vectors of the canonical basis.

Remark 1. *Notice that a solution of the shifted linear system given by $(s_{j+1}I - A)$ is needed in order to get the last relation for $j = 1, \ldots, m$. Fortunately, this shifted system has been already solved at each iteration j in the previous Algorithm 1.* ∎

Let $T_{i,j} = (V_i)^T\, A\, V_j \in \mathbb{R}^{2p\times 2p}$ and $\mathbb{T}_m = [T_{i,j}] \in \mathbb{R}^{2mp\times 2mp}$ be the restriction of the matrix A to the extended-rational Krylov subspace $\mathbb{K}_m(A, B)$, i.e.,

$$\mathbb{T}_m = \mathbb{V}_m^T\, A\, \mathbb{V}_m.$$

The matrix $\mathbb{T}_m$ is of great importance for the model reduction by projection methods. When we manipulate big size model, the direct calculus of the matrix $\mathbb{T}_m$ is of an elevated cost.

In the sequel, we give a recursion to compute $\mathbb{T}_m$ from $\mathbb{H}_m$ without requiring matrix-vector products with A and extra inner products of long vectors (block-vectors). First, we provide some notations:

- Let $[B,\ (s_1I - A)^{-1}B] := V_1\Lambda$ be the QR decomposition of $[B,\ (s_1I - A)^{-1}B]$ which can be written as

$$[B, (s_1I - A)^{-1}B] = V_1\Lambda = [V_1^{(1)},\ V_1^{(2)}] \begin{bmatrix} \Lambda^{(1,1)} & \Lambda^{(1,2)} \\ 0 & \Lambda^{(2,2)} \end{bmatrix}. \tag{2.6}$$

- For $j = 1, \ldots, m$, let the upper triangular matrix $H_{j+1,j}$ be under the form

$$H_{j+1,j} = \begin{bmatrix} H_{j+1,j}^{(1,1)} & H_{j+1,}^{(1,2)} \\ 0 & H_{j+1,j}^{(2,2)} \end{bmatrix}.$$

The following result enables us to compute $\overline{\mathbb{T}}_m$ directly from the columns of the upper block Hessenberg matrix $\overline{\mathbb{H}}_m$ obtained from Algorithm 1.

PROPOSITION 2.3. *Let $\overline{\mathbb{T}}_m$ and $\overline{\mathbb{H}}_m$ be the upper block Hessenberg matrices defined earlier. Then we have the following, for $j = 1, \ldots, m$*

$$\overline{\mathbb{T}}_m \, \widetilde{e}_{2j-1} = \begin{bmatrix} \overline{\mathbb{H}}_j \\ 0_{2(m-j)s \times 2js} \end{bmatrix} = \overline{\mathbb{H}}_m \, \widetilde{e}_{2j-1}, \tag{2.7}$$

$$\overline{\mathbb{T}}_m \, \widetilde{e}_2 = \Big[s_1 \begin{bmatrix} I_{2p} \\ 0_{2mp \times 2p} \end{bmatrix} \begin{bmatrix} \Lambda^{(1,2)} \\ \Lambda^{(2,2)} \end{bmatrix} - \overline{\mathbb{T}}_m \, \widetilde{e}_1 \, \Lambda^{(1,2)} - \widetilde{e}_1 \, \Lambda^{(1,1)} \Big] (\Lambda^{(2,2)})^{-1}, \tag{2.8}$$

and

$$\begin{aligned} \mathbb{T}_{m+1} \, \widetilde{e}_{2j+2} = \Big(s_j \overline{\mathbb{H}}_m \, \widetilde{e}_{2j} - \begin{bmatrix} \overline{\mathbb{T}}_j \\ 0_{2(m-j)s \times 2js} \end{bmatrix} \mathbb{H}_j \, \widetilde{e}_{2j} \\ - \mathbb{T}_{m+1} \, \widetilde{e}_{2j+1} \, H_{j+1,j}^{(1,2)} - \widetilde{e}_{2j} \Big) \, (H_{j+1,j}^{(2,2)})^{-1}, \end{aligned} \tag{2.9}$$

where $\widetilde{e}_i = e_i \otimes I_p$ and e_i is the vectors of the canonical basis.

Proof. Let $V_j = [V_j^{(1)}, V_j^{(2)}] \in \mathbb{R}^{n \times 2p}$ for $j \geq 1$. We notice from Algorithm 1 that

$$\widehat{V}_{j+1} = [A \, V_j^{(1)}, (s_j I - A)^{-1} \, V_j^{(2)}] - \mathbb{V}_j \, \mathbb{H}_j \, [\widetilde{e}_{2j-1}, \widetilde{e}_{2j}], \tag{2.10}$$

and

$$V_{j+1} \, H_{j+1,j} = \widehat{V}_{j+1}. \tag{2.11}$$

Using (2.10) and (2.11), we obtain

$$\begin{aligned} A \, V_j^{(1)} = \widehat{V}_{j+1} \, \widetilde{e}_1 + \mathbb{V}_j \, \mathbb{H}_j \, \widetilde{e}_{2j-1} &= V_{j+1} \, H_{j+1,j} \, \widetilde{e}_1 + \mathbb{V}_j \, \mathbb{H}_j \, \widetilde{e}_{2j-1} \\ &= \mathbb{V}_{j+1} \overline{\mathbb{H}}_j \, \widetilde{e}_{2j-1}, \end{aligned}$$

and by multiplying on the left by $\mathbb{V}_{m+1}^T$, we get

$$\mathbb{V}_{m+1}^T \, A \, V_j^{(1)} = \mathbb{V}_{m+1}^T \, \mathbb{V}_{j+1} \, \overline{\mathbb{H}}_j \, \widetilde{e}_{2j-1},$$

hence,

$$\mathbb{V}_{m+1}^T \, A \, \mathbb{V}_m \, \widetilde{e}_{2j-1} = \begin{bmatrix} I_{2(j+1)s} \\ 0_{2(m-j)s \times 2(j+1)s} \end{bmatrix} \overline{\mathbb{H}}_j \, \widetilde{e}_{2j-1},$$

and so

$$\overline{\mathbb{T}}_m \, \widetilde{e}_{2j-1} = \begin{bmatrix} \overline{\mathbb{H}}_j \\ 0_{2(m-j)s \times 2js} \end{bmatrix} = \overline{\mathbb{H}}_m \, \widetilde{e}_{2j-1}.$$

the (2.7) holds.
To prove (2.8), we start from the QR decomposition of $[B,\ (s_1 I - A)^{-1}B]$ given in (2.6)

$$[B,\ (s_1 I - A)^{-1}B] = [V_1^{(1)}\,\Lambda^{(1,1)}, V_1^{(1)}\,\Lambda^{(1,2)} + V_1^{(2)}\,\Lambda^{(2,2)}].$$

Then

$$(s_1 I - A)^{-1}B = V_1^{(1)}\,\Lambda^{(1,2)} + V_1^{(2)}\,\Lambda^{(2,2)}.$$

Now, multiplying on the left by $(s_1 I - A)$, we get

$$B = s_1[V_1^{(1)},\ V_1^{(2)}]\begin{bmatrix}\Lambda^{(1,2)}\\ \Lambda^{(2,2)}\end{bmatrix} - AV_1^{(1)}\,\Lambda^{(1,2)} - AV_1^{(2)}\,\Lambda^{(2,2)},$$

and since $B = V_1^{(1)}\,\Lambda^{(1,1)}$, we have

$$AV_1^{(2)}\,\Lambda^{(2,2)} = s_1 V_1\begin{bmatrix}\Lambda^{(1,2)}\\ \Lambda^{(2,2)}\end{bmatrix} - AV_1^{(1)}\,\Lambda^{(1,2)} - V_1^{(1)}\,\Lambda^{(1,1)}.$$

Then, if $\Lambda^{(2,2)}$ is nonsingular, assertion (2.8) holds by pre-multiplying the above equality on the left by $\mathbb{V}_{m+1}^T$ and using the facts that $\mathbb{V}_{m+1}^T V_1^{(1)} = (e_1 \otimes I_p) = \widetilde{e}_1$ and $\mathbb{V}_{m+1}^T A\, V_1^{(i)} = \overline{\mathbb{T}}_m\,(e_i \otimes I_p) = \overline{\mathbb{T}}_m\,\widetilde{e}_i$ for $i = 1,\ 2$.
Now, in order to prove (2.9), we use (2.10) and (2.11), to get

$$(s_j I - A)^{-1}\,V_j^{(2)} = \widehat{V}_{j+1}\,\widetilde{e}_2 + \mathbb{V}_j\,\mathbb{H}_j\,\widetilde{e}_{2j} = V_{j+1}\,H_{j+1,j}\,\widetilde{e}_2 + \mathbb{V}_j\,\mathbb{H}_j\,\widetilde{e}_{2j}.$$

Multiplying by $(s_j I - A)$ on the left, we get

$$V_j^{(2)} = s_j\left[V_{j+1}\,H_{j+1,j}\,\widetilde{e}_2 + \mathbb{V}_j\,\mathbb{H}_j\,\widetilde{e}_{2j}\right] - AV_{j+1}\,H_{j+1,j}\,\widetilde{e}_2 - A\mathbb{V}_j\,\mathbb{H}_j\,\widetilde{e}_{2j},$$

and since

$$AV_{j+1}\,H_{j+1,j}\,\widetilde{e}_2 = AV_{j+1}^{(1)}\,H_{j+1,j}^{(1,2)} + AV_{j+1}^{(2)}\,H_{j+1,j}^{(2,2)},$$

we deduce the following

$$\begin{aligned} AV_{j+1}^{(2)}\,H_{j+1,j}^{(2,2)} &= s_j\left[V_{j+1}\,H_{j+1,j}\,\widetilde{e}_2 + \mathbb{V}_j\,\mathbb{H}_j\,\widetilde{e}_{2j}\right] - A\mathbb{V}_j\,\mathbb{H}_j\,\widetilde{e}_{2j} - AV_{j+1}^{(1)}\,H_{j+1,j}^{(1,2)} - V_j^{(2)}\\ &= s_j\,\mathbb{V}_{j+1}\overline{\mathbb{H}}_j\,\widetilde{e}_{2j} - A\mathbb{V}_j\,\mathbb{H}_j\,\widetilde{e}_{2j} - AV_{j+1}^{(1)}\,H_{j+1,j}^{(1,2)} - V_j^{(2)}. \end{aligned}$$

Finally, multiplying on the left by $\mathbb{V}_{m+1}^T$, we get

$$\begin{aligned} \mathbb{T}_{m+1}\,\widetilde{e}_{2j+2} &= \Big[s_j\begin{bmatrix} I_{2(j+1)s}\\ 0_{2(m-j)s\times 2(j+1)s}\end{bmatrix}\overline{\mathbb{H}}_j\,\widetilde{e}_{2j} - \begin{bmatrix}\overline{\mathbb{T}}_j\\ 0_{2(m-j)s\times 2js}\end{bmatrix}\mathbb{H}_j\,\widetilde{e}_{2j}\\ &\quad - \mathbb{T}_{m+1}\,\widetilde{e}_{2j+1}\,H_{j+1,j}^{(1,2)} - \widetilde{e}_{2j}\Big]\,(H_{j+1,j}^{(2,2)})^{-1}\\ &= \left(s_j\overline{\mathbb{H}}_m\,\widetilde{e}_{2j} - \begin{bmatrix}\overline{\mathbb{T}}_j\\ 0_{2(m-j)s\times 2js}\end{bmatrix}\mathbb{H}_j\,\widetilde{e}_{2j} - \mathbb{T}_{m+1}\,\widetilde{e}_{2j+1}\,H_{j+1,j}^{(1,2)} - \widetilde{e}_{2j}\right)(H_{j+1,j}^{(2,2)})^{-1}. \end{aligned}$$

□

Remark 2. *Note that the matrix $\mathbb{T}_m$ has a block Hessenberg form.*

As for any Krylov subspace method, a set of identities known as the Arnoldi relations are satisfied and are used to compute error bounds, residuals, stop tests and to perform perturbation analysis. In the case of Rational Krylov subspace, some relations have been established in the literature, see, e.g., [14, 29, 36] and referenced therein. However these identities are much more complex in the Rational case when compared to the Polynomial Arnoldi equations.

PROPOSITION 2.4. *Assume that m steps of Algorithm 1 have been run and let $\overline{\mathbb{T}}_m = \mathbb{V}_{m+1}^T A\, \mathbb{V}_m$, then we have the following*

$$\begin{aligned} A\, \mathbb{V}_m &= \mathbb{V}_{m+1}\, \overline{\mathbb{T}}_m \\ &= \mathbb{V}_m\, \mathbb{T}_m + V_{m+1}\, T_{m+1,m}\, E_m^T, \end{aligned} \tag{2.12}$$

where $E_m = [0_{2p}; \cdots ; I_{2p}] = e_m \otimes I_{2p}$.

Proof. In order to prove the above proposition, we first need to show that

$$A\, \mathbb{K}_m(A, B) \subseteq \mathbb{K}_{m+1}(A, B).$$

Set $B = [B^{(1)}, \ldots, B^{(p)}] \in \mathbb{R}^{n\times p}$, where $B^{(k)}$ denotes the k-th column of B. In fact, for $j = 1, \ldots, m$ and $k = 1, \ldots, p$, we have

$$\begin{aligned} A \prod_{i=1}^{j} (s_i I - A)^{-1}\, B^{(k)} &= \prod_{i=1}^{j} (s_i I - A)^{-1}\, A\, B^{(k)} \\ &= \prod_{i=1}^{j} (s_i I - A)^{-1} \big[-(s_j I - A)\, B^{(k)} + s_j\, B^{(k)} \big] \\ &= \prod_{i=1}^{j-1} -(s_i I - A)^{-1} B^{(k)} + s_j \prod_{i=1}^{j} (s_i I - A)^{-1} B^{(k)} \in \mathbb{K}_{m+1}(A, B), \end{aligned}$$

thus $A \prod_{i=1}^{j} (s_i I - A)^{-1}\, B^{(k)} \in \mathbb{K}_{m+1}(A, B)$. It is easy to show that for $j = 1, \ldots, m$ and $k = 1, \ldots, p$, $A(A^j B^{(k)}) = A^{j+1} B^{(k)} \in \mathbb{K}_{m+1}(A, B)$. Hence, there exists a matrix T such that

$$A\, \mathbb{V}_m = \mathbb{V}_{m+1}\, T. \tag{2.13}$$

We have mentioned before that $\mathbb{T}_m$ is a block upper Hessenberg matrix with $(2p \times 2p)$ blocks and also it is a restriction matrix of A to $\mathbb{K}_m(A, B)$, i.e., $\mathbb{T}_m = \mathbb{V}_m A \mathbb{V}_m$. Since $\mathbb{V}_{m+1} = [\mathbb{V}_m, V_{m+1}]$, we have

$$\begin{aligned} \mathbb{T}_{m+1} &= \mathbb{V}_{m+1}^T A \mathbb{V}_{m+1} \\ &= \begin{bmatrix} \mathbb{V}_m^T A \mathbb{V}_m & \mathbb{V}_m^T A V_{m+1} \\ V_{m+1}^T A \mathbb{V}_m & V_{m+1}^T A V_{m+1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{T}_m & \mathbb{V}_m^T A V_{m+1} \\ V_{m+1}^T A \mathbb{V}_m & V_{m+1}^T A V_{m+1} \end{bmatrix}. \end{aligned}$$

We know that $\mathbb{T}_{m+1}$ is block upper Hessenberg, then we have $V_{m+1}^T A \mathbb{V}_m = T_{m+1,m} E_m^T$ and

$$\bar{\mathbb{T}}_m = \mathbb{V}_{m+1}^T A \mathbb{V}_m = \begin{bmatrix} \mathbb{T}_m \\ T_{m+1,m} E_m^T \end{bmatrix}.$$

Using (2.13) and multiplying by $\mathbb{V}_{m+1}$ on the left, we obtain $T = \bar{\mathbb{T}}_m$. As consequence

$$\begin{aligned} A\,\mathbb{V}_m &= \mathbb{V}_{m+1}\,\bar{\mathbb{T}}_m \\ &= [\mathbb{V}_m, V_{m+1}] \begin{bmatrix} \mathbb{T}_m \\ T_{m+1,m} E_m^T \end{bmatrix} \\ &= \mathbb{V}_m \mathbb{T}_m + V_{m+1} T_{m+1,m} E_m^T, \end{aligned}$$

where $E_m = [0_{2p}; \cdots ; I_{2p}]$=$e_m \otimes I_{2p}$ □

3. Application to model reduction problems.

The main purpose of this section is the computation of a reduced model order (1.3) approaching the original model (1.1), by using the block Extended-Rational method. The use of those large scale dynamical systems for either simulation or run-on-time control, can be very costly due to limitations of memory and/or computational capacity, hence the use of model reduction method.

We recall the linear time-invariant (LTI) multi-input and multi-output (MIMO) system (1.1)

$$\begin{cases} \dot{x}(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t). \end{cases}$$

To get the transfer function, we apply the transformation of Laplace

$$L(f)(s) := \int_0^\infty e^{-st} f(t),$$

to the original system (1.1), then we obtain :

$$\begin{cases} X(s) &= A\,X(s) + B\,U(s) \\ Y(s) &= C\,X(s), \end{cases}$$

where X(s),U(s) and Y(s) are the Laplace transforms of x(t), u(t) and y(t) respectively. If we eliminate X(s) in the two equations cited above, we get what we call the transfer function associated to the original system (1.1)

$$F(s) = C\,(s\,I_n - A)^{-1}\,B.$$

We recall that most of the model order reduction techniques, for example the moment-matching approaches, are based on the approximation of this transfer function, for more details, see, e.g., [2, 22, 17] and the references therein.

We apply the Galerkin projection method onto the Extended-Rational block Krylov. The process of this method is described as

1. Replacing $x(t) \approx \mathbb{V}_m x_m(t)$ on the original system (1.1) (where $\mathbb{V}_m$ is a orthonormal basis of $\mathbb{K}_m(A, B)$).

2. Using the fact that $\mathbb{V}_m^T \mathbb{V}_m = I_m$.

Following this process, we get the reduced system

$$\begin{cases} \dot{x}_m(t) &= \mathbb{T}_m\,x_m(t) + B_m\,u(t) \\ y_m(t) &= C_m\,x_m(t), \end{cases} \tag{3.1}$$

and we can easily verify that the original transfer function $F(s)$ can be approximated by

$$F_m(s) = C_m\,(s\,I_{2mp} - \mathbb{T}_m)^{-1}\,B_m, \tag{3.2}$$

where $\mathbb{T}_m = \mathbb{V}_m^T\,A\,\mathbb{V}_m$, $C_m = C\,\mathbb{V}_m$ and $B_m = \mathbb{V}_m^T\,B$.

The computation of the error between the exact transfer function and its approximation, can be used to verify how the response of our reduced model is sufficiently close to that of the original model. Unfortunately, we can't calculate directly the error due to the higher dimension of the original system. To remedy this problem, we propose the following proposition.

PROPOSITION 3.1. *Let F be the transfer function defined in (1.2) and let F_m (3.2) be its approximation obtained by the Galerkin projection method onto the extended-rational block Krylov subspace $\mathbb{K}_m(A, B)$. Then, under the condition $\|A\| < |s|$, we have the follwing upper bound*

$$F(s) - F_m(s) = C(sI_n - A)^{-1}V_{m+1}T_{m+1,m}\,\mathbb{E}_m^T(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T B, \tag{3.3}$$

and

$$\|F(s) - F_m(s)\| \leq \frac{\|B\|\|C\|\|T_{m+1,m}\|}{(|s| - \|A\|)}\|(sI_{2mp} - \mathbb{T}_m)^{-1}\|. \tag{3.4}$$

Proof. We have

$$\begin{aligned} F(s) - F_m(s) &= C(sI_n - A)^{-1}B - C_m(sI_{2mp} - \mathbb{T}_m)^{-1}B_m \\ &= C(sI_n - A)^{-1}B - C\mathbb{V}_m(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T B \\ &= C(sI_n - A)^{-1}\big[B - (sI_n - A)\mathbb{V}_m(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T B\big] \\ &= C(sI_n - A)^{-1}\big[B - (s\mathbb{V}_m - A\mathbb{V}_m)(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T B\big], \end{aligned}$$

using (2.12), we obtain

$$\begin{aligned} F(s) - F_m(s) &= C(sI_n - A)^{-1}\big[B - (s\mathbb{V}_m - \mathbb{V}_m\mathbb{T}_m - V_{m+1}T_{m+1,m}\,\mathbb{E}_m^T)(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T B\big] \\ &= C(sI_n - A)^{-1}\big[B - \mathbb{V}_m\mathbb{V}_m^* B + V_{m+1}T_{m+1,m}\,\mathbb{E}_m^T(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T B\big]. \end{aligned}$$

As B is in the extended-rational block Krylov subspace $\mathbb{K}_m(A, V)$, then we have $\mathbb{V}_m\mathbb{V}_m^T B = B$. It follows that

$$F(s) - F_m(s) = C(sI_n - A)^{-1}V_{m+1}T_{m+1,m}\,\mathbb{E}_m^T(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T B.$$

By applying the 2-norm and using the fact that for $\|A\| < |s|$, we have $\|(sI_n - A)^{-1}\| \leq \dfrac{1}{(|s| - \|A\|)}$, then (3.4) holds. □

3.1. An adaptive choice of interpolation points.

In the following, we introduce a new adaptive approach to select the interpolation points. The proposed choice is based on the residual expression given as follows:
We know that the transfer function of (1.1) is given by $F(s) = C\,(s\,I_n - A)^{-1}\,B$, which we can rewrite as $F(s) = CX$ where X is a solution of the following linear system of equations : $(s\,I_n - A)X = B$.
The approximation $X_m \in Range(\mathbb{V}_m)$ can be determined by imposing the Galerkin condition on the residual $R(s) \perp Range(\mathbb{V}_m)$, where

$$R(s) = B - (s\, I_n - A)X_m,$$

which gives

$$X_m = V_m(s\, I_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T\, B.$$

Then, the residual can be expressed as follows :

$$\begin{aligned}
R(s) &= B - (s\, I_n - A)X_m \\
&= B - (s\, I_n - A)\mathbb{V}_m(s\, I_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T\, B \\
&= B - (s\, \mathbb{V}_m - \mathbb{T}_m + \mathbb{T}_m - A\mathbb{V}_m)(s\, I_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T\, B \\
&= (s\, I_n - \mathbb{V}_m\mathbb{V}_m^T)B + (A\mathbb{V}_m - \mathbb{V}_m\mathbb{T}_m)(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T\, B \\
&= (A\mathbb{V}_m - \mathbb{V}_m\mathbb{T}_m)(sI_{2mp} - \mathbb{T}_m)^{-1}\mathbb{V}_m^T\, B.
\end{aligned}$$

We know form Algorithm 1 that B=$V_1 \begin{bmatrix} \Lambda_{1,1} \\ 0 \end{bmatrix}$, then $\mathbb{V}_m^T\, B$=$\begin{bmatrix} I_{2p} \\ 0_{2p} \\ \vdots \\ 0_{2p} \end{bmatrix} \begin{bmatrix} \Lambda_{1,1} \\ 0 \end{bmatrix}$.

Using Proposition 2.4 and the new expression of $\mathbb{V}_m^T\, B$, we get the new expression of the residual :

$$R(s) = V_{m+1}T_{m+1,m}\mathbb{E}_m^T(sI_{2mp} - \mathbb{T}_m)^{-1} \begin{bmatrix} I_{2p} \\ 0_{2p} \\ \vdots \\ 0_{2p} \end{bmatrix} \begin{bmatrix} \Lambda_{1,1} \\ 0 \end{bmatrix}.$$

The procedure to generate the set of interpolation points is defined as follows. We start by two initial shifts σ_1,σ_2 and we construct the next shift σ_{k+2} as :

$$\sigma_{k+2} = \arg\max_{\omega \in S} \|R(s)\|_2, \qquad k = 1 \cdots m-2, \tag{3.5}$$

where S is a set of frequency sample points.

Remark 3. *Notice that the expression of $R(s)$ allows us to reduce the computation cost, while searching for the next shift σ_{k+2} as it shown in the numerical test. This means that to solve problem 3.5, it requires only a computation of matrices of low size.*

In the following we present the block extended-rational Arnoldi algorithm. The algorithm is summarized as follows :

Algorithm 2 Adaptive Block Extended-Rational Arnoldi algorithm (ABERAA)

- Inputs: A, B, C,$\sigma_1, \sigma_2 \in \mathbb{R}$ and m.
- Outputs: The reduced system $(\mathbb{T}_m, B_m, C_m)$
- Compute : $V_1, V_2 \in \mathbb{R}^{n,2\times p}$ by the Gram-Schmith orthogonalization process
- For $k = 1, \ldots, m-2$
  1. Determine σ_{k+2} by solving (3.5)
  2. Set : $\widetilde{V}_{k+2} = [A\, V_{k+1}^{(1)}, (\sigma_{k+2}I - A)^{-1}\, V_{k+1}^{(2)}]$.
  3. Orthogonalize $\widetilde{V}_{k+2}$ with respect to to $\mathbb{V}_1, \ldots, \mathbb{V}_{k+1}$ to get V_{k+2},

  End For.
- Compute : $\mathbb{T}_m$ using proposition (2.3), B_m and C_m

Remark 4. *Note that a* `LU` *factorization is proposed to solve the shifted matrices* $(\sigma_{k+2}I - A)^{-1} V_{k+1}^{(2)}$ *or a solver such GMRES with pre-conditioner can be used instead, also* `qr` *decomposition is used to get* V_{k+2} *and after* m *iteration we end up by* $\mathcal{O}(p^2mn)$ *as total number of arithmetic operations.*

Beside Krylov techniques used to construct a reduced order dynamical system as it is shown in the above section, we can find what we called Lyapunov balanced truncation, one of the most known reduction techniques. This method was first introduced by Mullis and robert (1976) and later in the systems and control literature by Moore (1981). The method is based on the solutions of two large coupled Lyapunov matrix equation. Next, we give a brief description on how to solve such a matrix equation using the block Extended-Rational method.

4. Application to the Lyapunov equation.

We consider the following continuous-time algebraic Lyapunov equation :

$$AX + XA^T + BB^T = 0, \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}$ is large and sparse stable matrix. The matrix $B \in \mathbb{R}^{n\times p}$ is assumed to be of full rank with $p \ll n$.

In [5], a standard and widely direct method used for the solution of Lyapunov equations for small dimensional problems. In our case, the matrix A is large and sparse, then using a direct method would be unsuitable. Thus, iterative methods have to be used. We refer to [7, 27, 38], those are Krylov-type subspace methods based on the Arnoldi process. The alternating directional implicit (ADI) iterations could also be applied, see ([10]) and the references therein. In what follows, we use the block Extended-Rational Krylov method in order to solve the Lyapunov equation (4.1).

We seek for a low rank approximate solution to the Lyapunov matrix equation (4.1). We project the initial problem onto the block extended-rational Krylov subspace $\mathbb{K}_m(A, B)$. We consider the low-rank approximate solution under the following form

$$X_m = \mathbb{V}_m Y_m \mathbb{V}_m^T, \tag{4.2}$$

where $\mathbb{V}_m = [V_1, \cdots, V_m]$ an orthonormal basis of the block extended-rational Krylov subspace $\mathbb{K}_m(A, B)$ and $Y_m \in \mathbb{R}^{2mp\times 2mp}$. Note that rank$(X_m) \leq 2mp$.

We denote by $R(X_m)$ the residual corresponding to the approximation X_m given by

$$R_m := R(X_m) = AX_m + X_m A^T + BB^T. \tag{4.3}$$

Using the expression (4.2) in the residual equation, multiplying on the left by $\mathbb{V}_m^T$ and on the right by $\mathbb{V}_m$ (Galerkin condition), we get the low-dimensional continuous-time Lyapunov equation

$$\mathbb{T}_m Y_m + Y_m \mathbb{T}_m^T + (\mathbb{V}_m^T B)(\mathbb{V}_m^T B)^T = 0. \tag{4.4}$$

where $\mathbb{V}_m^T B = E_1 \Lambda^{(1,1)}$. In fact, using step 1 of algorithm 1, we get

$$V_1 \Lambda = [B, (s_1 I - A)^{-1} B].$$

Then

$$B = V_1 \begin{bmatrix} \Lambda^{(1,1)} \\ 0 \end{bmatrix}.$$

Multiplying on the left by $\mathbb{V}_m^T$, we get

$$\mathbb{V}_m^T B = E_1 \Lambda^{(1,1)},$$

where $E_1 = [I_p; 0_{(2m-1)p,p)}]$ is the first p columns of the identity matrix I_{2mp} and $\Lambda^{(1,1)}$ is the $p \times p$ matrix obtained from the QR decomposition

$$V_1 \, \Lambda = [B, (s_1 I - A)^{-1} \, B] \quad \text{with} \quad \Lambda = \begin{bmatrix} \Lambda^{(1,1)} & \Lambda^{(1,2)} \\ 0 & \Lambda^{(2,2)} \end{bmatrix}. \tag{4.5}$$

In order to stop iterations, we need to compute the residual R_m given by (4.2) without involving X_m, since it become expensive as m increases. The next result shows how to compute the residual norm R_m without involving the approximate solution, which is given only in a factored form at the end of the process.

THEOREM 4.1. *Let $\mathbb{V}_m = [V_1, \cdots, V_m]$ be an orthonormal basis of the extended-rational block Krylov subspace $\mathbb{K}_m(A, B)$. Let $X_m = \mathbb{V}_m Y_m \mathbb{V}_m^T$ be the approximate solution of the Lyapunov matrix equation (4.1), then the residual norm is given by*

$$\|R_m\| = \|T_{m+1,m} E_m^T Y_m\|, \tag{4.6}$$

where $E_m = [0_{2p \times 2(m-1)p}, I_{2p}]^T$.

Proof. According (2.12) and (4.2), we have

$$\begin{aligned} R_m(X_m) &= AX_m + X_m A^T + BB^T \\ &= A\mathbb{V}_m Y_m \mathbb{V}_m^T + \mathbb{V}_m Y_m \mathbb{V}_m^T A^T + BB^T \\ &= (\mathbb{V}_m \mathbb{T}_m + V_{m+1} T_{m+1,m} E_m^T) Y_m \mathbb{V}_m + \mathbb{V}_m Y_m (\mathbb{T}_m^T \mathbb{V}_m^T + E_m T_{m+1,m}^T V_{m+1}^T) + BB^T, \end{aligned}$$

using the fact that $B = V_1^1 \Lambda^{(1,1)}$, as it is described in 4.5, we get

$$R_m(X_m) = [\mathbb{V}_m, V_{m+1}] \begin{bmatrix} \mathbb{T}_m Y_m + Y_m \mathbb{T}_m^T + \, E_1 \Lambda^{(1,1)} (E_1 \Lambda^{(1,1)})^T & (T_{m+1,m} E_m^T Y_m)^T \\ T_{m+1,m} E_m^T Y_m & 0 \end{bmatrix} \begin{bmatrix} \mathbb{V}_m \\ V_{m+1} \end{bmatrix}.$$

Since $B_m^T = E_1 \Lambda^{(1,1)}$ and Y_m is the symmetric solution of the reduced Lyapunov matrix equation (4.4), we get

$$R_m(X_m) = \mathbb{V}_{m+1} \begin{bmatrix} 0 & (T_{m+1,m} E_m^T Y_m)^T \\ T_{m+1,m} E_m^T Y_m & 0 \end{bmatrix} \mathbb{V}_{m+1}^T,$$

and then

$$\|R_m\| = \|T_{m+1,m} E_m^T Y_m\|.$$

□

REMARK 5. *We can easily verify that the approximation X_m can be an exact solution of a perturbed continuous-time Lyapunov equation and that can be done by multiplying*

the reduced Lyapunov equation (4.4) on the left by $\mathbb{V}_m$ and on the right by $\mathbb{V}_m^T$ and using (2.12). This perturbed Lyapunov equation is given by

$$(A - G_m)^T X_m + X_m(A - G_m) + BB^T = 0,$$

where $G_m = V_m T_{m+1,m} V_{m+1}^T$.

It is important to note that the theorem above, allows us to achieve the convergence which can be verified by the test ($\|R_m\| < \epsilon$), without computing the approximate solution X_m. Using the fact that X_m is symmetric positive semidefinite, X_m can be decomposed into product of two matrices of low-rank as $X_m = ZZ^T$ where Z a matrix of rank smaller than or equal to $2m$, thus we need just to store Z in order to compute the approximate solution X_m. Let $Y_m = U\Sigma V^T$ be the SVD of Y_m where Σ is the matrix of the singular values of Y_m sorted in decreasing. Let `dtol` some tolerance, we define U_k, V_k as the first k columns respectively of U and V corresponding to the k singular values magnitude greater than `dtol`. Setting $\Sigma_k = [\sigma_1, \cdots, \sigma_k]$, we get the approximation $Y_m \approx U_k \Sigma_k V_k^T$, and it follows that

$$X_m \approx Z_m Z_m^T,$$

with $Z_m = \mathbb{V}_m U_k (\Sigma_k)^{1/2}$.

All these results are summarized in the following algorithm

Algorithm 3 Extended-Rational block Arnoldi Lyapunov algorithm (ERBALA)

- Inputs: A, B, tolerance ϵ, dtol, number of iteration m_{max}.
- Outputs: the approximate solution $X_m \approx Z_m Z_m^T$

- For $m = 1, \cdots, m_{max}$

- Using Algorithm 1 to compute $\mathbb{V}_m$ an orthonormal basis and compute $\mathbb{T}_m$ the block Hessenberg matrix.
- Solve the low-dimensional Lyapunov equation (4.4).
- Compute the residual norm

$$\|R_m\| = \|T_{m+1,m} E_m^T Y_m\|$$

- if $R_m < \epsilon$ then

  1. Compute the SVD of $Y_m = U\Sigma V$ where $\Sigma = diag[\sigma_1, \cdots, \sigma_{2m}]$
  2. Determine k such that $\sigma_{k+1} <$ dtol $\leq \sigma_k$, set $\Sigma_k = daig[\sigma_1, \cdots, \sigma_k]$ and compute $Z_m = \mathbb{V}_m U_k (\Sigma_k)^{1/2}$

  end if.

- End For

5. Model Reduction of Second-Order Systems.

Linear PDEs modeling structures in many areas of engineering (plates, shells, beams ...) are often second order in time see, e.g., [33, 34, 39]. The spatial semi-discretization of its models by a method of finite elements leads to systems written in the form:

$$\begin{cases} M\ddot{q}(t) + D\dot{q}(t) + Kq(t) &= Bu(t) \\ y(t) &= Cq(t), \end{cases} \tag{5.1}$$

where $M \in \mathbb{R}^{n\times n}$ is the mass matrix, $D \in \mathbb{R}^{n\times n}$ is the damping matrix and $K \in \mathbb{R}^{n\times n}$ the stiffness matrix. When the source term $Bu(t)$ is null, the system is said to be free, otherwise, it is said forced. If $D = 0$, the system is said to be undamped. We assume that the mass matrix M is invertible, then the system (5.1) can be written as

$$\left\{ \begin{array}{lcl} \ddot{q}(t) + D_M\dot{q}(t) + K_M q(t) & = & B_M u(t) \\ y(t) & = & Cq(t), \end{array} \right. \tag{5.2}$$

where $D_M = M^{-1}D$, $K_M = M^{-1}K$ and $B_M = M^{-1}B$, for simplicity we still denote K, D, and B instead of D_M, K_M and B_M respectively. The transfer function associated with the system (5.2) is given by using the Laplace transform as:

$$F(\omega) := C(\omega^2 I_n + \omega D + K)^{-1}B \in \mathbb{R}^{p\times p}. \tag{5.3}$$

It is usually difficult to get the efficient solution of various control or simulation tasks because the original system is too large to allow it. In order to solve this problem, methods that produce a reduced system of size $m \ll n$ that preserves the essential properties of the full order model have been developed. The reduced model have the following form:

$$\left\{ \begin{array}{lcl} \ddot{q}_m(t) + D_m\dot{q}_m(t) + K_m q_m(t) & = & B_m u(t) \\ y_m(t) & = & C_m q(t), \end{array} \right. \tag{5.4}$$

where $D_m,\ K_m \in \mathbb{R}^{m\times m}$, $B_m,\ C_m^T \in \mathbb{R}^{m\times p}$ and $q_m(t) \in \mathbb{R}^m$. The transfer function associated to the system (5.4) is given by:

$$F_m(\omega) := C_m(\omega^2 I_m + \omega D_m + K_m)^{-1}B_m \in \mathbb{R}^{p\times p}. \tag{5.5}$$

Second-order systems (5.2) can be written as a first order linear systems. In fact,

$$\left\{ \begin{array}{lcl} \begin{bmatrix} \dot{q}(t) \\ \ddot{q}(t) \end{bmatrix} & = & \begin{bmatrix} 0 & I_n \\ -K & -D \end{bmatrix} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} u(t) \\ y(t) & = & \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}, \end{array} \right. \tag{5.6}$$

which is equivalent to

$$\left\{ \begin{array}{lcl} \dot{x}(t) & = & \mathcal{A}x(t) + \mathcal{B}u(t) \\ y(t) & = & \mathcal{C}x(t), \end{array} \right. \tag{5.7}$$

with $x(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}$, $\mathcal{A} = \begin{bmatrix} 0 & I_n \\ -K & -D \end{bmatrix}$, $\mathcal{B} = \begin{bmatrix} 0 \\ B \end{bmatrix}$ and $\mathcal{C} = \begin{bmatrix} C & 0 \end{bmatrix}$.

Thus, the corresponding transfer function is defined as,

$$\mathcal{F}(\omega) := \mathcal{C}(\omega I_{2n} - \mathcal{A})^{-1}\mathcal{B} \in \mathbb{R}^{p\times p}. \tag{5.8}$$

We note that $\mathcal{F}(\omega) = F(\omega)$. In fact, setting

$$X = (\omega I_{2n} - \mathcal{A})^{-1}\mathcal{B} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix},$$

wich gives $\mathcal{F}(\omega) = \mathcal{C}X$, where X verifies $(\omega I_{2n} - \mathcal{A})X = \mathcal{B}$. Using the expressions of the matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, we get,

$$(\omega^2 I_n + \omega D + K)X_1 = B \text{ and } \mathcal{F}(\omega) = CX_1.$$

Hence

$$\mathcal{F}(\omega) = F(\omega) = C(\omega^2 I_n + \omega D + K)^{-1} B.$$

We can reduce the second-order system (5.2) by applying linear model reduction technique presented in the previous section to $(A,\ B,\ C)$ to yield a small linear system $(\mathbb{T}_m,\ B_m,\ C_m)$. Unfortunately, there is no guarantee that the matrices defining the reduced system have the necessary structure to preserve the second-order form of the original system. For that we follow the model reduction techniques of second-order structure-preserving, presented in [8, 11, 12].

5.1. The structure-preserving of the second-order reduced model. Using the Krylov subspace-based methods discussed in the previous section do not guaranty the second-order structure when applied to the linear system (5.7). The Authors in [8, 12] proposed a result, that gives a simple sufficient condition to satisfy the interpolation condition and produce a second order reduced system.

LEMMA 5.1. *Let $(\mathcal{A},\ \mathcal{B},\ \mathcal{C})$ be the state space realization defined in* (5.7). *If we project the state space realization with $2n \times 2mp$ bloc diagonal matrices*

$$\mathcal{V}_m = \begin{bmatrix} \mathcal{V}_m^1 & 0 \\ 0 & \mathcal{V}_m^2 \end{bmatrix},\ \mathcal{V}_m^T \mathcal{V}_m = I_{2mp},$$

where $\mathcal{V}_m^1$ and $\mathcal{V}_m^2 \in \mathbb{R}^{n \times mp}$, then the reduced transfer function

$$\mathcal{F}_m(\omega) := \mathcal{C}\mathcal{V}_m(\omega I_{2mp} - \mathcal{V}_m^T \mathcal{A} \mathcal{V}_m)^{-1} \mathcal{V}_m^T \mathcal{B},$$

is a second order transfer function, on condition that the matrix $(\mathcal{V}_m^2)^T \mathcal{V}_m^1$ is invertible.

THEOREM 5.2. *Let $\mathcal{F}(\omega) := \mathcal{C}(\omega I_{2n} - \mathcal{A})^{-1}\mathcal{B} = C(\omega^2 I_n + \omega D + K)^{-1} B$, with*

$$\mathcal{A} = \begin{bmatrix} 0 & I_n \\ -K & -D \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} 0 \\ B \end{bmatrix} \text{ and } \mathcal{C} = \begin{bmatrix} C & 0 \end{bmatrix},$$

be a second order transfer function. Let $\mathbb{V}_m \in \mathbb{R}^{2n \times mp}$ be defined as:

$$\mathbb{V}_m = \begin{bmatrix} \mathbb{V}_m^1 \\ \mathbb{V}_m^2 \end{bmatrix},$$

where $\mathbb{V}_m^1$, and $\mathbb{V}_m^2 \in \mathbb{R}^{n \times mp}$, with $(\mathbb{V}_m^1)^T \mathbb{V}_m^1 = (\mathbb{V}_m^2)^T \mathbb{V}_m^2 = I_{mp}$. Let us construct the $2n \times 2mp$ projecting matrices as

$$\mathcal{V}_m = \begin{bmatrix} \mathbb{V}_m^1 & 0 \\ 0 & \mathbb{V}_m^2 \end{bmatrix}.$$

Define the second order transfer function of order m by

$$\begin{aligned} \mathcal{F}_m(\omega) &= \mathcal{C}\mathcal{V}_m(\omega I_{2mp} - \mathcal{V}_m^T \mathcal{A} \mathcal{V}_m)^{-1} \mathcal{V}_m^T \mathcal{B} \\ &= \mathcal{C}_m(\omega I_{2mp} - \mathcal{A}_m)^{-1} \mathcal{B}_m. \end{aligned}$$

If we have

$$Range\{(s_1 I - \mathcal{A})^{-1}\mathcal{B}, ..., (s_m I - \mathcal{A})^{-1}\mathcal{B}\} \subseteq Range(\mathbb{V}_m),$$

where σ_i, for $i = 1, ...m$, are the interpolation points. Then the reduced order transfer function $\mathcal{F}_m(\omega)$ interpolates the values of the original transfer function $\mathcal{F}(\omega)$ and preserves the structures of the second-order model provided that the matrix $(\mathbb{V}_m^2)^T \mathbb{V}_m^1$ is non-singular.

6. Numerical examples.

In this section, we give some experimental results to show the effectiveness of the proposed approaches. All the experiments were performed on a computer of Intel Core i5 at 1.3GHz and 8GB of RAM. The algorithms were coded in Matlab R2014a. We used different known benchmark models listed in Table 6.1.

We use the following functions from LYAPACK [32]

- `lp_lgfrq` : Generates a set of logarithmically distributed frequency sampling points.
- `lp_gnorm` : Computes $\|F(s) - F_m(s)\|$.

TABLE 6.1
Test matrices

Matrix A	size n	size p	$\|A\|_F$	$cond(A)$
`CDplayer`	120	2	2.3095e+05	1.8149e+04
`ISS`	270	3	2.0594e+04	9.6800e+03
`FOM`	1006	6	1.8283e+04	1000
`RAIL5177`	5177	2	5.6430e+003	3.7420e+007
`Eady`	598	4	1.2660e+02	5.3727e+02
`MNA3`	4863	4	2.1168e+05	1.8156e+08
`MNA5`	10913	5	147.6687	6.2170e+04
`FDM`	40.000	5	2.8758e+08	9.7945e+04

The matrices for the benchmark problems `FDM`, `CDplayer`, `FOM`, `Eady`, `Beam`, `MNA3`, `MNA5` and `RAIL5177` are obtained from NICONET [31] while the matrices for the `Flow` model are obtained from the discretization of a 2D convective thermal flow problem (flow meter model v0.5) from the Oberwolfach collection [1]. Some informations on these matrices are reported in Table 6.1. For more details see [32].

Example 1 : In the first experiment, we consider the International Space Station (`ISS`). It has 270 states, three inputs and three outputs, for more details on this system see [25, 13]. The figure 6.1 shows the singular values of the transfer function and its approximation, and we plot the error norm $\|F(i\omega) - F_m(i\omega)\|_2$ versus the frequencies $\omega \in [10^{-1}, 10^5]$.

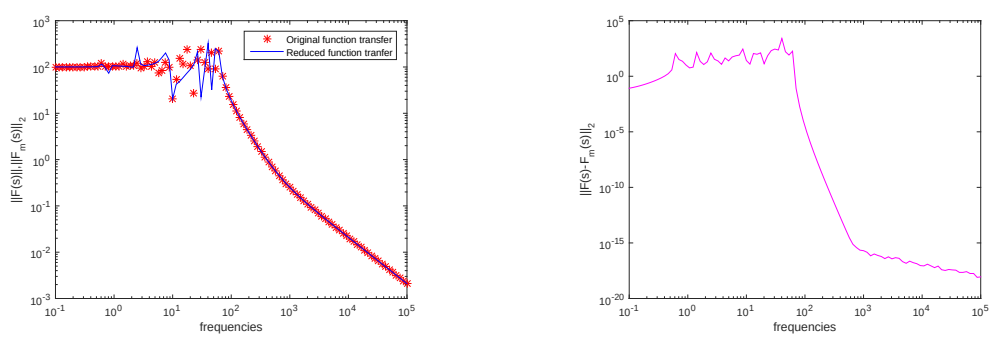

FIG. 6.1. *The `ISS` model. Left : singular values versus frequencies. Right : error norms vs frequencies.*

[1] Oberwolfach model reduction benchmark collection, 2003. http://www.imtek.de/simulation/benchmark

Example 2 : In this example, we use the `CDplayer` that describes the dynamics between a lens actuator and the radial arm position in a portable CD player. For more details on this system see [13]. The figure 6.2 shows the singular values of the transfer function and its approximation, and we plot the error norm $\|F(i\omega) - F_m(i\omega)\|_2$ versus the frequencies $\omega \in [10^{-1}, 10^5]$.

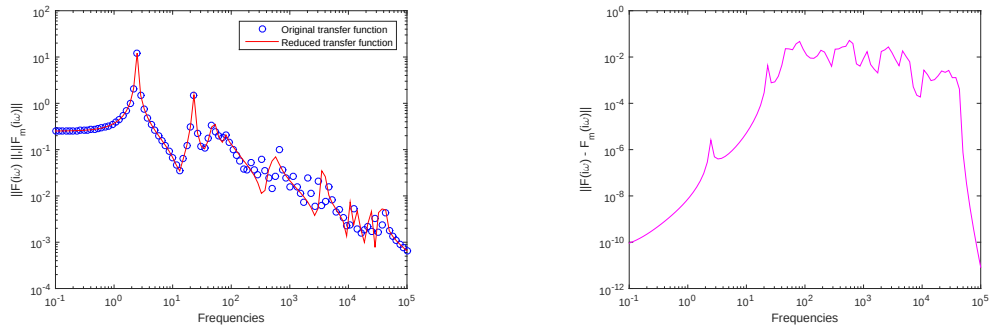

FIG. 6.2. *The* `CDplayer` *model. Left : singular values versus frequencies. Right : error norms vs frequencies.*

Example 3 : In this example we compare the performance of the Rational Krylov subspace method with that of the Extended-Rational Krylov subspace method. We use three models : BEAM, EADY, FOM. For more details about these models, see [13].

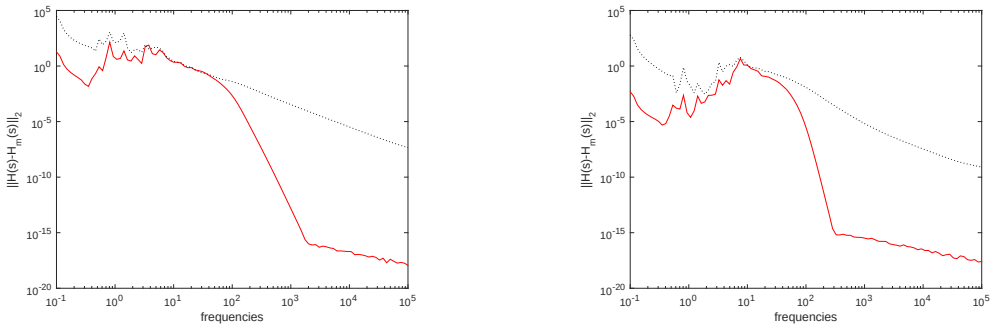

FIG. 6.3. *The* `BEAM` *model. Left : error norms vs frequencies for m=10. Extended-Rational Krylov(solid line), Rational Krylov (dotted line) Right : error norms vs frequencies for m=20. Extended-Rational Krylov(solid line), Rational Krylov (dotted line).*

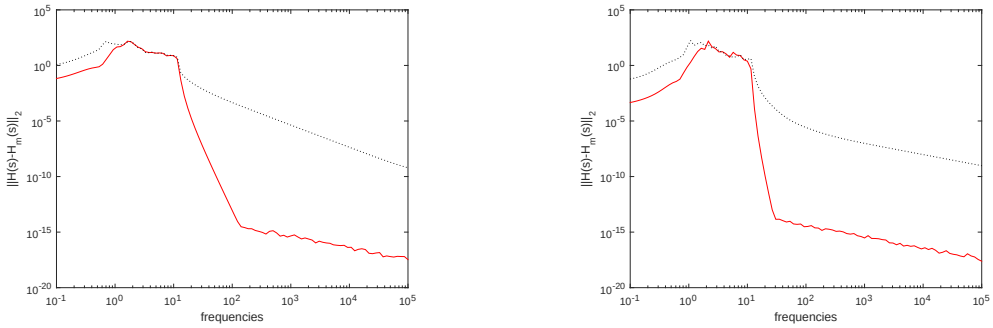

FIG. 6.4. *The* `EADY` *model. Left : error norms vs frequencies for m=10. Extended-Rational Krylov(solid line), Rational Krylov (dotted line) Right : error norms vs frequencies for m=20. Extended-Rational Krylov(solid line), Rational Krylov (dotted line).*

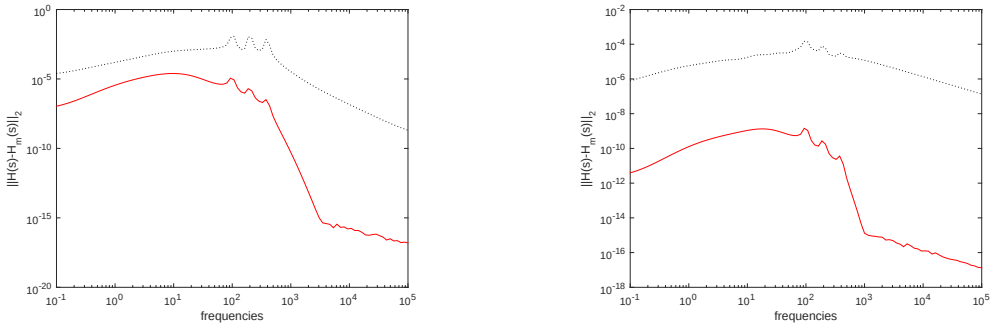

FIG. 6.5. *The* `FOM` *model. Left : error norms vs frequencies for m=10. Extended-Rational Krylov(solid line), Rational Krylov (dotted line) Right : error norms vs frequencies for m=20. Extended-Rational Krylov(solid line), Rational Krylov (dotted line).*

Example 4 In this example, we use the `FDM` model : (n =10.000 and n =40.000 with p =5). This model is obtained from the centred finite difference discretization of the operator

$$L_A(u) = \Delta u - f(x,y)\frac{\partial u}{\partial x} - g(x,y)\frac{\partial u}{\partial y} - h(x,y)u,$$

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

$$\begin{cases} f(x,y) &= \log(x + 2 * y + 1), \\ g(x,y) &= e^{x+y}, \\ h(x,y) &= x + y, \end{cases}$$

and the matrices B and C are random matrices with entries uniformly distributed in [0, 1]. The number of inner grid points in each direction is $n0 = 100$ and $n0 = 200$, besides the dimension of A is $n = n_0^2$. In the following table, you find a comparison results of the execution times and the H_∞ norm $\|F - F_m\|_{H_\infty}$ of the known `IRKA` algorithm suggested in [24] and our algorithm `ABERAA`.

TABLE 6.2
The calculation time and the Err-H_∞

Models		ABERAA		IRKA	
		time	Err-H_∞	time	Err-H_∞
FDM10.000	m=6	3.79 sec	8.2×10^{-3}	46.64 sec	1.02×10^{-3}
	m=12	7.93 sec	6.38×10^{-7}	164.42 sec	8.82×10^{-2}
	m=18	14.8 sec	1.9×10^{-10}	245.5 sec	2.37×10^{-2}
FDM40.000	m=6	11.83 sec	3.10×10^{-2}	257.26 sec	3.37×10^{-2}
	m=12	19.97 sec	2.30×10^{-5}	853.43 sec	1.76×10^{-2}
	m=18	30.95 sec	1.07×10^{-7}	976.69 sec	7×10^{-3}

We continue the comparison of the algorithm `IRKA` and `ABERAA` algorithm, here we use the `MNA3`, `MNA5`, `RAIL3113` and `RAIL5177` models from Table 6.1. We plot the curve corresponding to the error norm $\|F(i\omega) - F_m(i\omega)\|_2$ versus frequencies for `IRKA` (red line) and `ABERAA` (blue line) with $\omega \in [10^{-1}, 10^5]$.

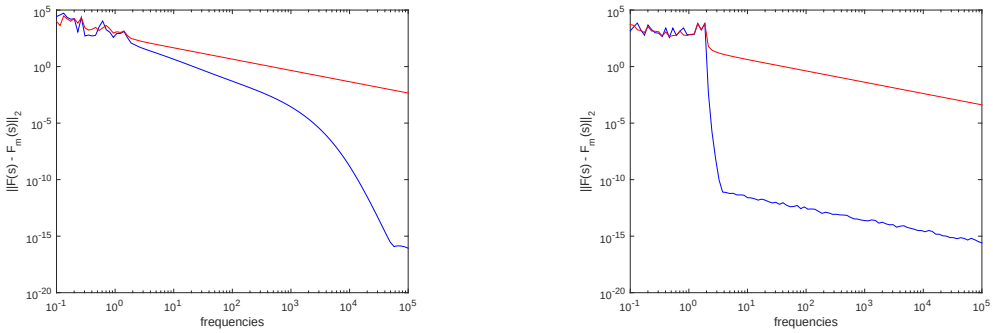

FIG. 6.6. *Left: The* `MNA3` *model :* `ABERAA`*(blue line) and* `IRKA`*(red line) with* $m = 10$ *and* $\omega \in [10^{-1}, 10^5]$. *Right: The* `MNA5` *model :* `ABERAA`*(blue line) and* `IRKA`*(red line) with* $m = 20$ *and* $\omega \in [10^{-1}, 10^5]$.

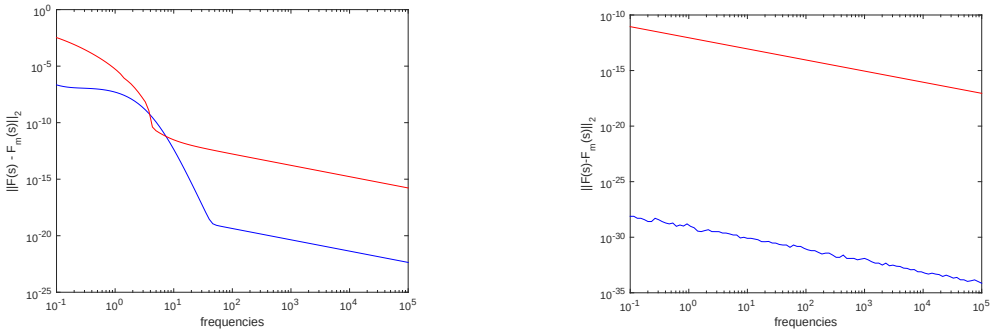

FIG. 6.7. *Left: The* `RAIL3113` *model :* `ABERAA`*(blue line) and* `IRKA`*(red line) with* $m = 10$ *and* $\omega \in [10^{-1}, 10^5]$. *Right: The* `RAIL5177` *model :* `ABERAA`*(blue line) and* `IRKA`*(red line) with* $m = 20$ *and* $\omega \in [10^{-1}, 10^5]$.

6.1. Examples of Lyapunov matrix equation. The results listed in the table below show the effectiveness of our method ERBALA as it is described in the Algorithm 3. We compare the results with the ones of the Rational method. For each example, we show the number iterations needed for convergence, the CPU-time in seconds and the rank of the obtained approximate solution. We use matrices from the table (6.1). The model `FDM` already described in the **Example 4**. For both models `FDM` and `FOM` we chose B as random matrix.

TABLE 6.3
Performance of the ERBALA.

Model / Method	`Extended-Rational method`	`Rational Method`
`FOM`, $n = 1006$, $p = 5$,		
Iteration	16	35
Residual norm	$4.05\,10^{-8}$	$9.85\,10^{-8}$
CPU-time	0.67	4.29
Rank	91	91
`FLOW` , $n = 9669$, $p = 3$,		
Iteration	41	70
Residual norm	$8.7\,10^{-8}$	$9\,10^{-6}$
CPU-time	16.44	68.23
Rank	238	210
`FDM` , $n = 12100$, $p = 4$,		
Iteration	16	53
Residual norm	$9.25\,10^{-8}$	$9.86\,10^{-8}$
CPU-time	14.11	> 200
Rank	85	85

6.2. Examples of Second order Systems.

Example 5: Linear 1D Beam

Moving structures are an essential part for many micro-system devices, among them fluidic components like pumps and electrically controllable valves, sensing cantilevers,

and optical structures. While the single component can easily be simulated on a usual desktop computer, the calculation of a system of many coupled devices still presents a challenge. This challenge is raised by the fact that many of these devices show a nonlinear behavior. This model describes a slender beam with four degrees of freedom per node: "x the axial displacement", "Θ_x the axial rotation", "y the flexural displacement" and "Θ_z the flexural rotation". The model is from the Oberwolfach collection. The matrices are obtained by using the finite element discretization presented in [40]. We used two examples of linear 1D Beam model:

The file name	Degrees of freedom	Num. nodes	Dimension n
1DBeam-LF100	flexural (Θ_z and y)	10000	n = 19998
1DBeam-LF5000	(Θ_z and y), (Θ_x and x)	50000	n = 19994

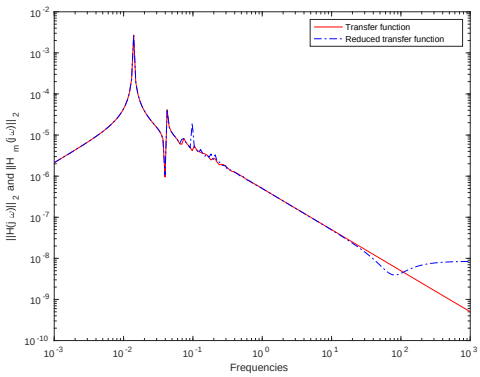

FIG. 6.8. *1DBeam-LF5000 model: m=20.*

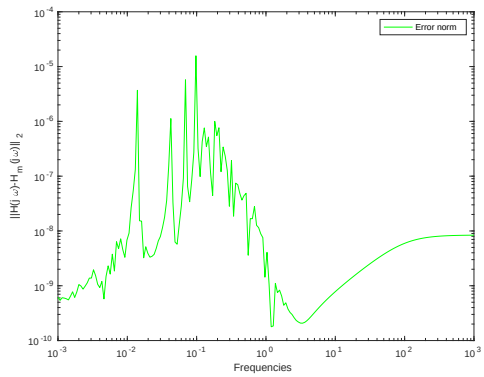

FIG. 6.9. *The error norm.*

Figures 6.8 above represent the norm of the original transfer function $\|H(j\omega)\|_2$ and the norm of the reduced transfer function $\|H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-3},\ 10^3]$ of the 1Dbeam-LF50000 model and it is a second-order model of dimension $2 \times n = 39988$ with one input and one output. Figure 6.9 represents the exact error $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies. The plots in Figures 6.10 and 6.11, represent the original transfer

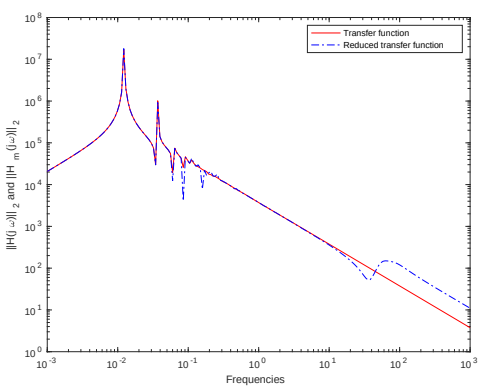

FIG. 6.10. *1DBeam-LF5000 model, m=20.*

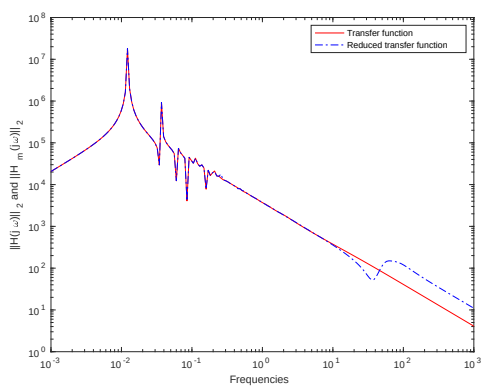

FIG. 6.11. *1DBeam-LF5000 mode, m=40.*

function $\|H(j\omega)\|_2$ and the norm of the reduced transfer function $\|H_m(j\omega)\|_2$ where we

modified the matrices B and C (random matrices) to get a MIMO system with four inputs and four outputs.

Example 6: Butterfly Gyroscope

The structural model of the gyroscope has been done in ANSYS (the global leader in engineering simulation) using quadratic tetrahedral elements. The model used here is a simplified one with a coarse mesh as it is designed to test the model reduction approaches. It only includes the pure structural mechanics problem. The load vector is composed from time-varying nodal forces applied at the centers of the excitation electrodes. The Dirichlet boundary conditions have been applied to all degrees of freedom of the nodes belonging to the top and the bottom surfaces of the frame. This benchmark is also part of the Oberwolfach Collection. It is a second-order model of dimension $n = 17361$, (then the matrix $\mathcal{A}$ is of size $2 \times n = 34722$) with the matrix $B = C^T$ to get a MIMO system with 12 inputs and 12 outputs.

The plots in Figure 6.12 represent $\|H(j\omega)\|_2$ and the norm of the reduced transfer function $\|H_m(j\omega)\|_2$. Figure 6.13 represent the exact error $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies. The dimension of the reduced model is $m = 40$. The execution time was 41.83 seconds with $\mathcal{H}_\infty$-err norms equal to 1.73×10^{-3}.

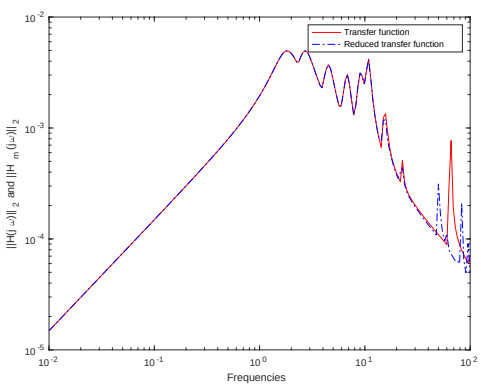

FIG. 6.12. *Butterfly model, m=40.*

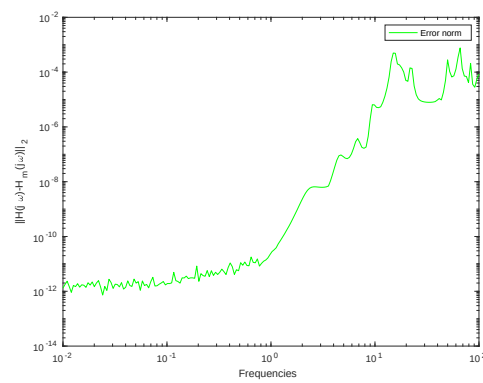

FIG. 6.13. *The error norm.*

Conclusion.

A Krylov-based method has been explored in this paper. This method is based on a projection technique onto a new proposed Krylov subspace named the block Extended-Rational Krylov subspace. A variety of applications of this method have been presented such as the construction of reduced dynamical system of first and second order, as well as the solution of Lyapunov matrix equation. An adaptive method to choose the interpolation points, that are used into the construction of the block Extended-rational Krylov subspace has been established. Moreover, we have derived from the block Extended-Rational Arnoldi process some new algebraic properties and a new error expression between the original transfer function and its approximation. Numerical results are finally reported to confirm the performance of our method compared with other known methods.

REFERENCES

[1] O. Abidi and K. Jbilou, *Balanced Truncation-Rational Krylov methods for model reduction in large scale dynamical systems.* Comput. Appl. Math., 37(1)525–540 (2018).

[2] Z. Bai, Q. Ye, *Error estimation of the Padé approximation of transfer functions via the Lanczos process.* Elect. Trans. Numer. Anal., 1–17 (1998).

[3] H. Barkouki, A. H. Bentbib and K. Jbilou, *An Adaptive Rational Block Lanczos-type Algorithm for Model Reduction of Large Scale Dynamical Systems.* SIAM J. Sci. Comput.67., 221–236 (2015).

[4] H. Barkouki, A. H. Bentbib and K. Jbilou, *An extended nonsymmetric block Lanczos method for model reduction in large scale dynamical systems.* Calcolo., 55(1) (2018).

[5] R. H. Bartels and G.W. Stewart, *Solution of the matrix equation* $AX+XB=C$. Comm. ACM., 15 820–826 (1972).

[6] C. A. Beattie S. Gugercin, *Krylov-based minimization for optimal* $\mathcal{H}_2$ *model reduction.* In: Proceedings of the 46th IEEE Conference on Decision and Control., 4385–4390 (2007).

[7] A. El Guennouni, K. Jbilou and A. J. Riquet, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl., 283-313 (1992).

[8] C. A. Beattie, S. Gugercin, *Krylov-based model reduction of second-order systems with proportional damping.* Proceedings of the 44th IEEE Conference on Decision and Control., 2278–2283 (2005).

[9] A. H. Bentbib, K. Jbilou and Y. Kaouane, A computational global tangential Krylov subspace method for model reduction of large-scale MIMO dynamical systems. J. Sci. Comput., 59, 1–19 (2017).

[10] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Linear Algebra Appl., 15 pp. 755777 (2008).

[11] Y. Chahlaoui, K. A. Gallivan, A. Vandendorpe and P. Van Dooren, *Model reduction of second-order system.* Comput. Sci. Eng., 45, 149–172 (2005).

[12] Y. Chahlaoui, D. Lemonnier, A. Vandendorpe and P. Van Dooren, *Second-order balanced truncation.* Lin. Alg. Appli., 415, 373–384 (2006).

[13] Y.Chahlaoui and P.Van Dooren, *A collection of benchmark examples for model reduction of linear time invariant dynamical systems.* MIMS EPrint., (2008).

[14] V. Druskin, C. Lieberman and M. Zaslavsky, *On adaptive choice of shifts in rational Krylov subspace reduction of evolutionary problems.* SIAM J. Sci. Comput., 32(5)2485–2496 (2010).

[15] V. Druskin and V. Simoncini, *Adaptive rational Krylov subspaces for large-scale dynamical systems.* Systems Control Lett., 60(8) 546–560 (2011).

[16] M. Frangos and I.M. Jaimoukha, *Adaptive rational Krylov algorithms for model reduction.* In Proceedings of the European Control Conference., 4179–4186 (2007).

[17] M. Frangos and I.M. Jaimoukha, *Adaptive rational interpolation: Arnoldi and Lanczos-like equations.* Eur. J. Control., 14(4) 342–354 (2008).

[18] K. Gallivan, E. Grimme and P. Van Dooren, *A rational Lanczos algorithm for model reduction.* Numer. Alg., 12 33–63 (1996).

[19] K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their L-infinity error bounds.* Inter. J. Cont., 39 1115–1193 (1984).

[20] K. Glover, D. J. N. Limebeer, J. C. Doyle, E. M. Kasenally and M.G. Safonov, *A characterization of all solutions to the four block general distance problem.* SIAM J. Control Optim., 29 283–324 (1991).

[21] E. Grimme, *Krylov projection methods for model reduction.* Ph.D. thesis., Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, (1997).

[22] E. Grimme, D. Sorensen and P. Van Dooren, *Model reduction of state space systems via an implicitly restarted Lanczos method.* Numer. Alg., 12 1–32 (1996).

[23] S. Gugercin and A. C. Antoulas, *A survey of model reduction by balanced truncation and some new resultsn.* Internat. J. Control., 27–55 (2003).

[24] S. Gugercin, A.C. Antoulas and C. Beattie, $\mathcal{H}_2$ *model reduction for large-scale linear dynamical systems.* SIAM J. Matrix Anal. Appl., 30 609–638 (2008).

[25] S. Gugercin, A.C. Antoulas and N. Bedrossian, *Approximation of the international space station 1R and 12A models.* In the Proceedings of the 40th CDC., (2001).

[26] M. Heyouni, K. Jbilou, A. Messaoudi and K. Tabaa, *Model reduction in large scale MIMO dynamical systems via the block Lanczos method.* Comput. Appl. Math., 27(2), 211-236 (2008).

[27] K. Jbilou, *Low rank approximate solutions to large Sylvester matrix equations*, Appl. Math. Comput., 365-376 (2006).

[28] Y. Kaouane, *A tangential method for the balanced truncation in model reduction.* Numer Algor

83, 629652. https://doi.org/10.1007/s11075-019-00696-9 (2020).

[29] L. Knizhnerman, V. Druskin and M. Zaslavsky, *On optimal convergence rate of the rational Krylov subspace reduction for electromagnetic problems in unbounded domains*. SIAM J. Numer. Anal., 47(2) 953–971 (2009).

[30] A. J. LAUB, *A Schur method for solving algebraic Riccati equations*. IEEE Trans. Automat. Control., 24, pp.913921 (1979).

[31] V. Merhrmann and T. Penzl, *Benchmark collections in SLICOT*. Technical Report SLWN 19985, SLICOT Working Note, ESAT, KU Leuven, K. Mercierlaan 94, Leuven-Heverlee 3100, Belgium, Available under http://www.win.tue.nl/niconet/NIC2/reports.html., (1998).

[32] T. Penzl, *LYAPACK A MATLAB toolbox for Large Lyapunov and Riccati Equations, Model Reduction Problem, and Linear-quadratic Optimal Control Problems*. http://www.tu-chemnitz.de/sfb393/lyapack.

[33] A. Preumont, *Vibration control of active structures*. An introduction, second ed., Kluwer, Dordrecht, (2002).

[34] M. F. Rubinstein, *Structural systems-statics, dynamics and stability*. Prentice Hall, Inc., (1970).

[35] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Lin. Alg. Appl., 58 391–405 (1984).

[36] A. Ruhe, *The rational Krylov algorithm for nonsymmetric eigenvalue problems III: Complex shifts for real matrices*, BIT., 34(1) 165–176 (1994).

[37] Y. Shamash, *Stable reduced-order models using Pad type approximations*. IEEE. Trans. Automatic Control., AC-19 615–616 (1974).

[38] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comp., 29(3):1268–1288, (2007).

[39] W. Weaver, P. Johnston, *Structural dynamics by finite elements*. Prentice Hall, Inc., (1987).

[40] W. Weaver, S. P. Timoshenko, D. H. Young, *Vibration problems in engineering*. 5th ed Wiley., (1990).