



HAL
open science

An adaptive block tangential method for multi-input multi-output dynamical systems

A.H. Bentbib, Khalide Jbilou, Yassine Kaouane

► **To cite this version:**

A.H. Bentbib, Khalide Jbilou, Yassine Kaouane. An adaptive block tangential method for multi-input multi-output dynamical systems. *Journal of Computational and Applied Mathematics*, 2019, 358, pp.190 - 205. 10.1016/j.cam.2019.03.014 . hal-03484587

HAL Id: hal-03484587

<https://hal.science/hal-03484587>

Submitted on 20 Dec 2021

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.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

An adaptive block tangential method for multi-input multi-output dynamical systems

A. H. Bentbib^a, K. Jbilou^b, Y. Kaouane^{b,a}

^a*Faculté des Sciences et Techniques-Gueliz, Laboratoire de Mathématiques Appliquées et Informatique, Marrakech, Morocco.*

^b*Laboratoire de Mathématiques Pures et Appliquées, Université du Littoral Côte d'Opale, 50 Rue F. Buisson, BP 699 - 62228 Calais cedex, France.*

Abstract

In this paper, we present a new approach for model order reduction in large-scale dynamical systems, with multiple inputs and multiple outputs (MIMO). This approach will be named: Adaptive Block Tangential Arnoldi Algorithm (ABTAA) and is based on interpolation via block tangential Krylov subspaces requiring the selection of shifts and tangent directions via an adaptive procedure. We give some algebraic properties and present some numerical examples to show the effectiveness of the proposed method.

Keywords: Block Arnoldi, Krylov subspaces, Model reduction, Tangential directions.

subclass MSC 65F10, MSC 65F30

1. Introduction

A multi-input and multi-output (MIMO) linear time invariant (LTI) dynamical system can be expressed in the state-space form as

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{cases} \quad (1)$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t) \in \mathbb{R}^p$ and $y(t) \in \mathbb{R}^p$ are the input and output signals, respectively. The matrix $A \in \mathbb{R}^{n \times n}$ is assumed to be large, sparse and stable, $B \in \mathbb{R}^{n \times p}$ and $C \in \mathbb{R}^{p \times n}$. Applying the Laplace transform to (1) yields

$$\begin{cases} \omega X(\omega) &= AX(\omega) + BU(\omega) \\ Y(\omega) &= CX(\omega), \end{cases} \quad (2)$$

Email addresses: a.bentbib@uca.ac.ma (A. H. Bentbib), jbilou@univ-littoral.fr (K. Jbilou), yassine.kaouane@etu.univ-littoral.fr (Y. Kaouane)

where $X(\omega)$, $Y(\omega)$ and $U(\omega)$ are the Laplace transform of $x(t)$, $y(t)$ and $u(t)$, respectively. If we eliminate $X(\omega)$ in the previous two equations we obtain $Y(\omega) = H(\omega)U(\omega)$, where $H(\omega)$ is called the transfer function of the system (1) defined as

$$H(\omega) := C(\omega I_n - A)^{-1}B \in \mathbb{R}^{p \times p}. \quad (3)$$

The LTI dynamical system (1) is usually denoted as

$$\Sigma := \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array} \right]. \quad (4)$$

When the dimension n of the original system is very large, it is not practical to use the complete system for simulation or execution control. The goal of model reduction techniques is to produce a much smaller order system with the state-space form

$$\begin{cases} \dot{x}_m(t) &= A_m x_m(t) + B_m u(t) \\ y_m(t) &= C_m x_m(t), \end{cases} \quad (5)$$

and its transfer function

$$H_m(\omega) := C_m(\omega I_m - A_m)^{-1}B_m \in \mathbb{R}^{p \times p}, \quad (6)$$

where $A_m \in \mathbb{R}^{m \times m}$, $B_m \in \mathbb{R}^{m \times p}$ and $C_m \in \mathbb{R}^{p \times m}$, (with $m \ll n$), such that the reduced system (6) will have an output $y_m(t)$ as close as possible to the one of the original system to any given input $u(t)$, which means that for some chosen norm, $\|y - y_m\|$ should be small.

Various model reduction methods for MIMO systems, such as Padé approximation [13, 34], balanced truncation [27], optimal Hankel norm [17, 18] have been used for the reduction of large scales dynamical systems. The most popular techniques used for model reduction these last years are based on interpolation methods [7, 8, 24]. These methods use block Krylov subspace

$$\mathcal{K}_m(A, B) = \text{Range}\{B, AB, \dots, A^{m-1}B\},$$

or rational block Krylov subspace

$$\mathbb{K}_m(A, B) = \text{Range}\{(\sigma_1 I - A)^{-1}B, \dots, (\sigma_m I - A)^{-1}B\},$$

where $\sigma_1, \dots, \sigma_m$ are some selected complex shifts. The purpose of those methods is to produce a reduced order model with a moderate space dimension, by projecting the original problem onto $\mathcal{K}_m(A, B)$ or $\mathbb{K}_m(A, B)$, see [2, 14, 23]. In the present paper, we use a method that was first introduced in [28] where one has to interpolate the transfer function at some points and in directions $d_i \in \mathbb{R}^p$. The tangential Krylov subspace is defined as

$$\mathcal{W}_m = \text{span}\{(s_1 I - A)^{-1}Bd_1, \dots, (s_m I - A)^{-1}Bd_m\}.$$

In this work, the tangent directions are blocks of $p \times s$ with $s < p$, and the computation of the parameters (s_i, d_i) will be done in an adaptive way.

The paper is organized as follows: In Section 2, we introduce the tangential interpolation method. In Section 3 we present the Adaptive Block Tangential Arnoldi method, where an adaptive approach is used for the selection of the shifts and the tangential directions, that will be used in the construction of tangential Krylov subspaces. The last section is devoted to some numerical tests and comparisons with some well known model order reduction methods.

Throughout the paper we use the following notations: The field of values of A is defined by

$$\mathcal{W}(A) = \{x^T A x, x \in \mathbb{C}^n, \|x\| = 1\},$$

where $\|\cdot\|$ is the Euclidean vector norm. We assume that $\mathcal{W}(A)$ is strictly a subset of \mathbb{C}^- .

2. Tangential interpolation

2.1. Moments and interpolation

Given the LTI dynamical system defined by (1), then its associated transfer function $H(\omega) = C(\omega I - A)^{-1}B$ can be decomposed through a Laurent series expansion around a given $\sigma \in \mathbb{C}$ (shift point), as follows

$$H(\omega) = \eta_0^{(\sigma)} + \eta_1^{(\sigma)}(\omega - \sigma) + \eta_2^{(\sigma)}(\omega - \sigma)^2 + \dots \quad (7)$$

where $\eta_i^{(\sigma)} \in \mathbb{R}^{p \times p}$ for $i \geq 0$ are called the i -th moments at σ associated to the system (1) and given as follows

$$\eta_i^{(\sigma)} = C(\sigma I_n - A)^{-(i+1)}B = (-1)^i \frac{d^i}{d\omega^i} H(\omega)|_{\omega=\sigma}, \quad i = 0, 1, \dots \quad (8)$$

In the case where $\sigma = \infty$, the $\eta_i^{(\sigma)}$'s are called Markov parameters and are given by

$$\eta_i = CA^i B.$$

The aim of this paper is to produce a transfer function H_m corresponding to the low order model, that approximates the original transfer function H . Various model reduction methods for MIMO dynamical systems have been explored these last years. Some of them are based on Krylov subspace interpolation methods. the main idea is as follows:

Select a set of points $\{\sigma_i\}_{i=1}^m \subset \mathbb{C}$ and seek for a reduced order transfer function H_m such that $H_m(\sigma_i) = H(\sigma_i)$ for $i = 1, \dots, m$; see [3, 4, 10, 11] for more details.

The tangential interpolation is a more powerful method in which the interpolation conditions above are acting in specified directions. Assume that the following parameters are given: Right

complex interpolation points $\{\sigma_i\}_{i=1}^m$ and right tangent directions $\{r_i\}_{i=1}^m \subset \mathbb{C}^p$. The aim of the tangential interpolation is to produce a low-order dimensional LTI dynamical system (5) such that the associated transfer function, H_m in (6) is a tangential interpolant to H , i.e.

$$\bullet H_m(\sigma_i)r_i = H(\sigma_i)r_i, \quad \text{for } i = 1, \dots, m. \quad (9)$$

The interpolation points and tangent directions are selected to realize the model reduction goals described later. We want to interpolate H without ever computing explicitly the quantities in (9), since these parameters are numerically ill-conditioned, as provided in [13] for single-input/single-output dynamical systems. This can be achieved by using Petrov-Galerkin projections by carefully choosing the projection subspaces.

The model reduction interpolation projectors were first introduced in [9, 22]. Later, Grimme [19] modified this approach into a numerically framework by using the rational Krylov subspace method of Ruhe [37]. For MIMO dynamical systems, a rational tangential interpolation method has been developed in [1, 15].

In this paper we considered another approach based on a work of Druskin and Simoncini [11], as well as some theory in [1]. For this approach, we considered the tangential directions as blocks of $p \times s$ size with $s < p$ and we used the block Arnoldi procedure to generate orthogonal bases of the desired projection subspaces.

3. The adaptive block tangential Arnoldi method

Let the original transfer function $H(\omega) = C(\omega I - A)^{-1}B$ be expressed as $H(\omega) = CX$ where X is such that

$$(\omega I_n - A)X = B. \quad (10)$$

Hence, approximating $H(\omega)$, for a fixed ω such that $\omega I - A$ is nonsingular, is equivalent to approximate the solution X of the multiple linear systems (10). This will be done as follows: Given a system of matrices $\{V_1, \dots, V_m\}$ where $V_i \in \mathbb{R}^{n \times s}$, the approximate solution X_m of X is computed, at step m , such that

$$X_m^i \in \text{Range}\{V_1, \dots, V_m\}, \quad (11)$$

and

$$R_B^i(\omega) \perp \text{Range}\{V_1, \dots, V_m\}, \quad i = 1, \dots, p \quad (12)$$

where X_m^i and R_B^i are the i -th columns of X_m and $R_B = B - (\omega I_n - A)X_m$, respectively. If we set $\mathbb{V}_m = [V_1, \dots, V_m]$, then from (11) and (12), we obtain

$$X_m = \mathbb{V}_m(\omega I_{ms} - A_m)^{-1} \mathbb{V}_m^T B,$$

which gives the following approximate transfer function

$$H_m(\omega) = C_m(\omega I_{ms} - A_m)^{-1} B_m,$$

where $A_m = \mathbb{V}_m^T A \mathbb{V}_m$, $B_m = \mathbb{V}_m^T B$ and $C_m = C \mathbb{V}_m$. Notice that the residual can be expressed as

$$R_B(\omega) = B - (\omega I_n - A) \mathbb{V}_m (\omega I_{ms} - A_m)^{-1} \mathbb{V}_m^T B. \quad (13)$$

Next, we introduce the block tangential Arnoldi algorithm that allows us to compute an orthonormal basis of some specific matrix subspace and we derive some algebraic relations related to this algorithm.

3.1. The block tangential Arnoldi method

We present here the block tangential Arnoldi algorithm (BTAA) for computing an orthonormal matrix $\mathbb{V}_m = [V_1, \dots, V_m]$ such that

$$\text{Range}\{V_1, \dots, V_m\} = \text{Range}\{(\sigma_1 I_n - A)^{-1} B R_1, \dots, (\sigma_m I_n - A)^{-1} B R_m\}, \quad (14)$$

where $\sigma = \{\sigma_i\}_{i=1}^m$ is a set of interpolation points and $\{R_i\}_{i=1}^m$ is a set of tangential matrix directions, where $R_i \in \mathbb{R}^{p \times s}$. The algorithm is summarized as follows:

Algorithm 1 The Block Tangential Arnoldi Algorithm (BTAA)

– Inputs: $A, B, C, \sigma = \{\sigma_i\}_{i=1}^{m+1}, R = \{R_i\}_{i=1}^{m+1}, R_i \in \mathbb{R}^{p \times s}$.

– Output: $\mathbb{V}_{m+1} = \{V_1, \dots, V_{m+1}\}$.

- Set $\tilde{V}_1 = (\sigma_1 I_n - A)^{-1} B R_1$.
 - Compute $\tilde{V}_1 = V_1 H_{1,0}$, QR decomposition.
 - Initialize: $\mathbb{V}_1 = [V_1]$.
 - For $j = 1, \dots, m$
 1. If $\sigma_{j+1} \neq \infty$, $\tilde{V}_{j+1} = (\sigma_{j+1} I_n - A)^{-1} B R_{j+1}$, else $\tilde{V}_{j+1} = A B R_{j+1}$.
 2. For $i = 1, \dots, j$
 - $H_{i,j} = V_i^T \tilde{V}_{j+1}$,
 - $\tilde{V}_{j+1} = \tilde{V}_{j+1} - V_i H_{i,j}$,
 3. End.
 4. $\tilde{V}_{j+1} = V_{j+1} H_{j+1,j}$, QR Decomposition.
 5. $\mathbb{V}_{j+1} = [V_j, V_{j+1}]$,
 - End
-

In Algorithm 1, we assume that the interpolation points $\sigma = \{\sigma_i\}_{i=1}^{m+1}$ and tangential directions $\{R_i\}_{i=1}^{m+1}$ are given. At each iteration j , we use a new interpolation point σ_{j+1} and a new tangential direction R_{j+1} , $j = 1, \dots, m$ and we initialize the subsequent tangential subspace by setting $\tilde{V}_{j+1} = (\sigma_{j+1} I_n - A)^{-1} B R_{j+1}$ if σ_{j+1} is finite and $\tilde{V}_{j+1} = A B R_{j+1}$ if $\sigma_{j+1} = \infty$. The

matrices $H_{i,j}$ constructed in Step 2 are of size $s \times s$ and they are used to construct the block upper Hessenberg matrix $\tilde{\mathbb{H}}_m = [\tilde{\mathbb{H}}^{(1)}, \dots, \tilde{\mathbb{H}}^{(m)}] \in \mathbb{R}^{(m+1)s \times ms}$, where

$$\tilde{\mathbb{H}}^{(j)} = \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix}, \quad \text{for } j = 1, \dots, m,$$

and we define the $(m+1)s \times s$ matrix $\tilde{\mathbb{H}}^{(0)}$ as

$$\tilde{\mathbb{H}}^{(0)} = \begin{bmatrix} H_{1,0} \\ \mathbf{0} \end{bmatrix}.$$

where $\mathbf{0}$ is the zero matrix of size $(m-j) \times s$. The upper Hessenberg matrix \mathbb{H}_m is the $ms \times ms$ matrix obtained from $\tilde{\mathbb{H}}_m$ by deleting its last row

$$\tilde{\mathbb{H}}_m = \begin{bmatrix} \mathbb{H}_m \\ H_{m+1,m}(e_m^T \otimes I_s) \end{bmatrix}.$$

The next proposition gives some algebraic properties corresponding to the matrices derived from Algorithm 1.

Proposition 3.1. *Let \mathbb{V}_{m+1} be the orthonormal matrix of $\mathbb{R}^{n \times (m+1)s}$ constructed by Algorithm 1. Then we have the following relations*

$$A\mathbb{V}_{m+1}\tilde{\mathbb{H}}_m = \mathbb{V}_{m+1}\tilde{\mathbb{K}}_m - B\tilde{\mathbb{R}}_{m+1}, \quad (15)$$

$$A_m = \mathbb{V}_m^T A \mathbb{V}_m = [\mathbb{K}_m - B_m \tilde{\mathbb{R}}_{m+1} - \mathbb{V}_m^T A \mathbb{V}_{m+1} H_{m+1,m}(e_m^T \otimes I_s)] \mathbb{H}_m^{-1}, \quad (16)$$

and

$$\mathbb{T}_{m+1} = \mathbb{V}_{m+1} \mathbb{G}_{m+1}, \quad (17)$$

where $\tilde{\mathbb{K}}_m = \tilde{\mathbb{H}}_m(D_m \otimes I_s)$, $D_m = \text{Diag}\{\sigma_2, \dots, \sigma_{m+1}\}$, $\tilde{\mathbb{R}}_{m+1} = [R_2, \dots, R_{m+1}]$, $\mathbb{G}_{m+1} = [\tilde{\mathbb{H}}^{(0)} \quad \tilde{\mathbb{H}}_m]$ is a block upper triangular matrix and \mathbb{K}_m is the $ms \times ms$ matrix obtained from $\tilde{\mathbb{K}}_m$ by deleting its last row. The matrix \mathbb{H}_m is assumed to be non singular.

Proof From Algorithm 1, we have

$$V_{j+1}H_{j+1,j} = (\sigma_{j+1}I_n - A)^{-1}BR_{j+1} - \sum_{i=1}^j V_i H_{i,j} \quad j = 1, \dots, m. \quad (18)$$

Multiplying (18) on the left by $(\sigma_{j+1}I_n - A)$ and re-arranging terms, we get

$$A \sum_{i=1}^{j+1} V_i H_{i,j} = \sigma_{j+1} \sum_{i=1}^{j+1} V_i H_{i,j} - BR_{j+1} \quad j = 1, \dots, m,$$

which gives

$$A \mathbb{V}_{j+1} \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \end{bmatrix} = \sigma_{j+1} \mathbb{V}_{j+1} \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \end{bmatrix} - BR_{j+1}, \quad j = 1, \dots, m,$$

also be written as

$$A \mathbb{V}_{m+1} \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} = \sigma_{j+1} \mathbb{V}_{j+1} \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} - BR_{j+1}, \quad j = 1, \dots, m, \quad (19)$$

where $\mathbf{0}$ is the zero matrix of size $(m-j) \times s$. Then we have

$$A \mathbb{V}_{m+1} \tilde{\mathbb{H}}^{(j)} = \sigma_{j+1} \mathbb{V}_{j+1} \tilde{\mathbb{H}}^{(j)} - BR_{j+1}, \quad j = 1, \dots, m. \quad (20)$$

Therefore, we can deduce from (20), the following expression

$$A \mathbb{V}_{m+1} [\tilde{\mathbb{H}}^{(1)}, \dots, \tilde{\mathbb{H}}^{(m)}] = \mathbb{V}_{m+1} [\tilde{\mathbb{H}}^{(1)}, \dots, \tilde{\mathbb{H}}^{(m)}] (D_m \otimes I_s) - B \tilde{\mathbb{R}}_{m+1},$$

which ends the proof of (15).

For the relation (16), we have from (15),

$$A \mathbb{V}_m \mathbb{H}_m + A V_{m+1} H_{m+1,m} (e_m^T \otimes I_s) = \mathbb{V}_m \mathbb{K}_m + \sigma_{m+1} V_{m+1} H_{m+1,m} (e_m^T \otimes I_s) - B \tilde{\mathbb{R}}_{m+1}.$$

Multiplying on the left by \mathbb{V}_m^T gives

$$\mathbb{V}_m^T A \mathbb{V}_m \mathbb{H}_m = \mathbb{K}_m - \mathbb{V}_m^T B \tilde{\mathbb{R}}_{m+1} - \mathbb{V}_m^T A V_{m+1} H_{m+1,m} (e_m^T \otimes I_s).$$

Therefore

$$A_m = \mathbb{V}_m^T A \mathbb{V}_m = [\mathbb{K}_m - B_m \tilde{\mathbb{R}}_{m+1} - \mathbb{V}_m^T A V_{m+1} H_{m+1,m} (e_m^T \otimes I_s)] \mathbb{H}_m^{-1}.$$

For the proof of (17), we first use (18) to obtain

$$\sum_{i=1}^{j+1} V_i H_{i,j} = (\sigma_{j+1} I_n - A)^{-1} BR_{j+1} \quad j = 1, \dots, m,$$

which gives

$$\mathbb{V}_{m+1} \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} = (\sigma_{j+1}I_n - A)^{-1}BR_{j+1}, \quad j = 1, \dots, m.$$

It follows that

$$\mathbb{V}_{m+1} \left[\tilde{\mathbb{H}}^{(1)}, \dots, \tilde{\mathbb{H}}^{(m)} \right] = [(\sigma_2 I_n - A)^{-1}BR_2, \dots, (\sigma_{m+1} I_n - A)^{-1}BR_{m+1}],$$

Since $V_1 H_{1,0} = (\sigma_1 I_n - A)^{-1}BR_1$, we have

$$\mathbb{V}_{m+1} \left[\tilde{\mathbb{H}}^{(0)}, \tilde{\mathbb{H}}^{(1)}, \dots, \tilde{\mathbb{H}}^{(m)} \right] = [(\sigma_1 I_n - A)^{-1}BR_1, (\sigma_2 I_n - A)^{-1}BR_2, \dots, (\sigma_{m+1} I_n - A)^{-1}BR_{m+1}],$$

which ends the proof of (17). \square

The following theorem generalizes a result given in [1].

Theorem 3.1. *Let $\sigma \in \mathbb{C}$ be such that $(\sigma I - A)$ is invertible. Let $\mathbb{V}_m = [V_1, \dots, V_m]$ have full-rank, where the $V_i \in \mathbb{R}^{n \times s}$. Let $R = [r_1, \dots, r_s] \in \mathbb{R}^{p \times s}$ be a chosen tangential matrix direction. Then,*

1. *If $(\sigma I - A)^{-1}Br_i \in \text{Range}\{V_1, \dots, V_m\}$ for $i = 1, \dots, s$, then*

$$H_m(\sigma)R = H(\sigma)R.$$

2. *If in addition A is symmetric and $C = B^T$, then,*

$$R^T H'_m(\sigma)R = R^T H'(\sigma)R.$$

Proof 1) We follow the same techniques as those given in [1] for the non-block case. Define

$$\mathcal{P}_m(\omega) = \mathbb{V}_m(\omega I_m - A_m)^{-1}\mathbb{V}_m^T(\omega I - A),$$

and

$$\mathcal{Q}_m(\omega) = (\omega I - A)\mathcal{P}_m(\omega)(\omega I - A)^{-1} = (sI - A)\mathbb{V}_m(\omega I_m - A_m)^{-1}\mathbb{V}_m^T.$$

It is easy to verify that $\mathcal{P}_m(\omega)$ and $\mathcal{Q}_m(\omega)$ are projectors. Moreover, for all ω in a neighborhood of σ we have

$$\mathcal{V}_m = \text{Range}\{V_1, \dots, V_m\} = \text{Range}(\mathcal{P}_m(\omega)) = \text{Ker}(I - \mathcal{P}_m(\omega)).$$

Observe that

$$H(\omega) - H_m(\omega) = C(\omega I - A)^{-1}(I - \mathcal{Q}_m(\omega))(\omega I - A)(I - \mathcal{P}_m(\omega))(\omega I - A)^{-1}B. \quad (21)$$

Evaluating this expression at $\omega = \sigma$ and multiplying by r_i from the right, yields the first assertion.

2) If A is symmetric and $C = B^T$, we have $\mathcal{V}_m^\perp = \text{Ker}(\mathcal{Q}_m(\omega)) = \text{Range}(I - \mathcal{Q}_m(\omega))$. Notice that

$$((\sigma + \varepsilon)I - A)^{-1} = (\sigma I - A)^{-1} - \varepsilon(\sigma I - A)^{-2} + O(\varepsilon^2),$$

and

$$((\sigma + \varepsilon)I_m - A_m)^{-1} = (\sigma I_m - A_m)^{-1} - \varepsilon(\sigma I_m - A_m)^{-2} + O(\varepsilon^2).$$

Therefore, evaluating (21) at $s = \sigma + \varepsilon$, multiplying by r_j^T and r_i , from the left and the right respectively, for $i, j = 1, \dots, s$, we get

$$r_j^T H(\sigma + \varepsilon) r_i - r_j^T H_m(\sigma + \varepsilon) r_i = O(\varepsilon^2).$$

Now notice that since $r_j^T H(\sigma) r_i = r_j^T H_m(\sigma) r_i$, we have

$$\lim_{\varepsilon \rightarrow 0} \left[\frac{1}{\varepsilon} (r_j^T H(\sigma + \varepsilon) r_i - r_j^T H(\sigma) r_i) - \frac{1}{\varepsilon} (r_j^T H_m(\sigma + \varepsilon) r_i - r_j^T H_m(\sigma) r_i) \right] = 0,$$

which proves the second assertion. \square

Proposition 3.2. *Let $R_B(\omega)$ be the residual $R_B(\omega) = B - (\omega I_n - A) \mathbb{V}_m \mathcal{Q}_m(\omega)$ as given in (13), where $\mathcal{Q}_m(\omega) = (\omega I_{ms} - A_m)^{-1} \mathbb{V}_m^T B$. We have the following new expression given by*

$$R_B(\omega) = (I_n - \mathbb{V}_m \mathbb{V}_m^T) B + (A \mathbb{V}_m - \mathbb{V}_m A_m) \mathcal{Q}_m(\omega). \quad (22)$$

Proof We have

$$\begin{aligned} R_B(\omega) &= B - \omega \mathbb{V}_m \mathcal{Q}_m(\omega) + A \mathbb{V}_m \mathcal{Q}_m(\omega) \\ &= B + A \mathbb{V}_m \mathcal{Q}_m(\omega) - \mathbb{V}_m (\omega I_{ms} - A_m) (\omega I_{ms} - A_m)^{-1} \mathbb{V}_m^T B \\ &\quad - \mathbb{V}_m A_m (\omega I_{ms} - A_m)^{-1} \mathbb{V}_m^T B \\ &= B + A \mathbb{V}_m \mathcal{Q}_m(\omega) - \mathbb{V}_m \mathbb{V}_m^T B - \mathbb{V}_m A_m \mathcal{Q}_m(\omega) \\ &= (I_n - \mathbb{V}_m \mathbb{V}_m^T) B + (A \mathbb{V}_m - \mathbb{V}_m A_m) \mathcal{Q}_m(\omega), \end{aligned}$$

which proves (22). \square

Proposition 3.3. *Let $\mathbb{T}_m = [(A - \sigma_1 I)^{-1} B R_1, \dots, (A - \sigma_m I)^{-1} B R_m] = \mathbb{V}_m \mathbb{G}_m$, where \mathbb{G}_m and \mathbb{V}_m are obtained by the Block Tangential Arnoldi Algorithm (BTAA). Let $\mathbb{R}_m = [R_1, \dots, R_m]$, then*

$$A \mathbb{V}_m - \mathbb{V}_m A_m = - (I_n - \mathbb{V}_m \mathbb{V}_m^T) B \mathbb{R}_m \mathbb{G}_m^{-1}, \quad (23)$$

and

$$R_B(\omega) = (I_n - \mathbb{V}_m \mathbb{V}_m^T) B (I_p - \mathbb{R}_m \mathbb{G}_m^{-1} \mathcal{Q}_m(\omega)). \quad (24)$$

Proof Let $\Sigma_m = [\text{diag}(\sigma_1, \dots, \sigma_m) \otimes I_s]$, then from the fact that

$$A(\sigma_i I - A)^{-1} BR_i = -BR_i + \sigma_i(\sigma_i I - A)^{-1} BR_i,$$

it follows that

$$A \mathbb{V}_m = A \mathbb{T}_m \mathbb{G}_m^{-1} = (-B \mathbb{R}_m + \mathbb{T}_m \Sigma_m) \mathbb{G}_m^{-1}.$$

Since $A_m = \mathbb{V}_m^T A \mathbb{V}_m = \mathbb{G}_m^{-T} \mathbb{T}_m^T A \mathbb{T}_m \tilde{\mathbb{G}}_m^{-1}$, we have

$$\begin{aligned} A \mathbb{V}_m - \mathbb{V}_m A_m &= -B \mathbb{R}_m \mathbb{G}_m^{-1} + \mathbb{T}_m \Sigma_m \mathbb{G}_m^{-1} - \mathbb{V}_m \mathbb{V}_m^T A \mathbb{V}_m, \\ &= -B \mathbb{R}_m \mathbb{G}_m^{-1} + \mathbb{T}_m \Sigma_m \mathbb{G}_m^{-1} - \mathbb{V}_m \mathbb{V}_m^T (-B \mathbb{R}_m + \mathbb{T}_m \Sigma_m) \mathbb{G}_m^{-1}, \\ &= -B \mathbb{R}_m \mathbb{G}_m^{-1} + \mathbb{T}_m \Sigma_m \mathbb{G}_m^{-1} + \mathbb{V}_m \mathbb{V}_m^T B \mathbb{R}_m \mathbb{G}_m^{-1} - \mathbb{V}_m \mathbb{V}_m^T \mathbb{T}_m \Sigma_m \mathbb{G}_m^{-1}. \end{aligned}$$

Now, as $\mathbb{T}_m = \mathbb{V}_m \mathbb{G}_m$, we have $\mathbb{V}_m \mathbb{V}_m^T \mathbb{T}_m \Sigma_m \mathbb{G}_m^{-1} = \mathbb{T}_m \Sigma_m \mathbb{G}_m^{-1}$ and then

$$\begin{aligned} A \mathbb{V}_m - \mathbb{V}_m A_m &= -B \mathbb{R}_m \mathbb{G}_m^{-1} + \mathbb{V}_m \mathbb{V}_m^T B \mathbb{R}_m \mathbb{G}_m^{-1} \\ &= -(I_n - \mathbb{V}_m \mathbb{V}_m^T) B \mathbb{R}_m \mathbb{G}_m^{-1}. \end{aligned}$$

□

The expression in (24) will be used in the next section in order to reduce the cost when computing the residual.

3.2. An adaptive strategy for selecting the interpolation points and tangent directions

In this subsection we use an adaptive strategy for choosing the interpolation points and tangent directions. This technique was first proposed in [10] to choose the shifts for the rational Krylov subspaces. The iterative rational Krylov algorithm (IRKA) was proposed in [1], where an initial set of interpolation points is given and a new set of interpolation points is chosen as a set of the mirror images of the eigenvalues of A_m , i.e $\sigma_i = -\lambda_i(A_m)$, $i = 1, \dots, m$. In [35] the iterative tangential interpolation algorithm (ITIA) was also proposed, with the same strategy as the one of IRKA, and the tangential directions are selected as

$$l_i = C_m d_i, \quad r_i = B_m^T g_i, \quad (25)$$

where d_i and g_i , $i = 1, \dots, m$, are right and left eigenvectors respectively, of the reduced model, i.e.,

$$A_m d_i = \lambda_i d_i, \quad g_i^T A_m = \lambda_i g_i^T.$$

In this paper we use an adaptive approach, inspired by the work given in [12].

In the adaptive approach, we seek to extend our subspace

$$\mathcal{W}_m = \text{Range}\{(\sigma_1 I_n - A)^{-1} BR_1, \dots, (\sigma_m I_n - A)^{-1} BR_m\},$$

by a new block defined by

$$W_{m+1} = (\sigma_{m+1} I_n - A)^{-1} BR_{m+1}, \quad (26)$$

which means that, at each iteration, we seek to define a new interpolation point σ_{m+1} and a new tangent direction R_{m+1} .

They will be computed as follows

$$(R_{m+1}, \sigma_{m+1}) = \arg \max_{\substack{\omega \in S_m \\ R \in \mathbb{R}^{p \times s}, \|R\|_2 = 1}} \|R_B(\omega)R\|_2. \quad (27)$$

Here $S_m \subset \mathbb{C}^+$ is the convex hull of $\{-\lambda_1, \dots, -\lambda_m\}$ where $\{\lambda_i\}_{i=1}^m$ are the eigenvalues of A_m .

Now we explain how to solve the problem (27). First we compute the interpolation point σ_{m+1} , by maximizing the the residual norm on the convex hull S_m , i.e we solve the following problem,

$$\sigma_{m+1} = \arg \max_{\omega \in S_m} \|R_B(\omega)\|_2. \quad (28)$$

In the case of small to medium systems, this is done by computing the norm of $R_B(\omega)$ for each ω in S_m and the tangent direction R_{m+1} is computed by evaluating (27) at $\omega = \sigma_{m+1}$

$$R_{m+1} = \arg \max_{R \in \mathbb{R}^{p \times s}, \|R\|=1} \|R_B(\sigma_{m+1})R\|_2. \quad (29)$$

Notice that the tangential matrix direction $R_{m+1} = [r_1^{(m+1)}, \dots, r_s^{(m+1)}]$, can be determined such that $r_i^{(m+1)}$ are the right singular vectors corresponding to the s largest singular values of $R_B(\sigma_{m+1})$.

In the case where the problem is large, the expression (24) of the residual given in Proposition 3.3

$$R_B(\omega) = (I_n - \mathbb{V}_m \mathbb{V}_m^T) B (I_p - \mathbb{R}_m \mathbb{G}_m^{-1} \mathbb{Q}_m(\omega)),$$

allows us to reduce the computational cost, while seeking for the next interpolation point and tangent direction. Applying the skinny QR decomposition $(I_n - \mathbb{V}_m \mathbb{V}_m^T) B = QL$, we get

$$\|R_B(\omega)\|_2 = \|L (I_p - \mathbb{R}_m \mathbb{G}_m^{-1} \mathbb{Q}_m(\omega))\|_2. \quad (30)$$

This means that, solving (27) requires only the computation of matrices of size $ms \times ms$ for each value of ω .

Next, we present the adaptive block tangential Arnoldi algorithm (ABTAA). The algorithm is summarized as follows:

Algorithm 2 Adaptive block tangential Arnoldi algorithm (ABTAA)

– Inputs $A, B, C, m, \omega_0^{(1)} \in \mathbb{R}$ and $R_1 \in \mathbb{R}^{p \times s}$.

– Outputs: $A_m = \mathbb{V}_m^T A \mathbb{V}_m, B_m = \mathbb{V}_m^T B$ and $C_m = C \mathbb{V}_m$.

- Set $\sigma_1 = \omega_0^{(1)}, \tilde{V}_1 = (\tilde{\sigma}_1 I_n - A)^{-1} B R_1$.
 - Compute $\tilde{V}_1 = V_1 H_{1,0}$ (QR decomposition) and initialize: $\mathbb{V}_1 = [V_1]$.
 - For $k = 1 : m - 1$
 1. If $\tilde{\sigma}_{k-1} \neq \sigma_k \in \mathbb{C}$ then $\sigma_{k+1} = \tilde{\sigma}_k$ else compute $\{\lambda_1, \dots, \lambda_{ks}\}$ the eigenvalues of A_k .
 2. Determine S_k , the convex hull of $\{-\lambda_1, \dots, -\lambda_{ks}, \omega_0^{(1)}, \bar{\omega}_0^{(1)}\}$ and solve (28).
 3. Compute R_{k+1} by solving (29).
 4. If $\sigma_{k+1} \neq \infty, \tilde{V}_{k+1} = (\sigma_{k+1} I_n - A)^{-1} B R_{k+1}$ else $\tilde{V}_{k+1} = A B R_{k+1}$.
 5. For $i = 1, \dots, k$
 - $H_{i,k} = V_i^T \tilde{V}_{k+1}$,
 - $\tilde{V}_{k+1} = \tilde{V}_{k+1} - V_i H_{i,k}$,
 6. End.
 7. $\tilde{V}_{k+1} = V_{k+1} H_{k+1,k}$, (QR Decomposition).
 8. $\mathbb{V}_{k+1} = [\mathbb{V}_k, V_{k+1}]$.
 - End
-

Algorithm 2 allows us to compute a low dimensional dynamical system by computing the reduced matrices $A_m = \mathbb{V}_m^T A \mathbb{V}_m, B_m = \mathbb{V}_m^T B$ and $C_m = C \mathbb{V}_m$. The interpolation points and the tangent directions are computed in an adaptive way.

Proposition 3.4. Let $\mathbb{V}_k = [V_1, \dots, V_k]$, be the orthonormal matrix obtained by Algorithm 2 at the iteration k , then setting $\mathcal{M}_k = \text{Range}\{V_1, \dots, V_k, (\sigma_{k+1} I_n - A)^{-1} B R_{k+1}\}$, we have

$$\text{Range}(\mathcal{M}_k) = \text{Range}\{V_1, \dots, V_k, (\sigma_{k+1} I_n - A)^{-1} R_B(\sigma_{k+1}) R_{k+1}\},$$

and

$$\dim(\mathcal{M}_k) = k + 1 \quad \text{if and only if} \quad R_B(\sigma_{k+1}) R_{k+1} \neq 0.$$

Proof We have,

$$R_B(\sigma_{k+1}) R_{k+1} = B(\sigma_{k+1}) R_{k+1} - (\sigma_{k+1} I_n - A) \mathbb{V}_k (\sigma_{k+1} I_{js} - A_k)^{-1} B_k R_{k+1}.$$

Multiplying the last equality on the left by $(\sigma_{k+1} I_n - A)^{-1}$, gives

$$(\sigma_{k+1} I_n - A)^{-1} R_B(\sigma_{k+1}) R_{k+1} = (\sigma_{k+1} I_n - A)^{-1} B R_{k+1} - \mathbb{V}_k (\sigma_{k+1} I_{ks} - A_k)^{-1} B_k R_{k+1},$$

which proves the first assertion.

If $R_B(\sigma_{k+1})R_{k+1} = 0$, then $\dim(\{V_1, \dots, V_k, (\sigma_{k+1}I_n - A)^{-1}BR_{k+1}\}) = k$.
Now assume that $R_B(\sigma_{k+1})R_{k+1} \neq 0$, then we only need to prove that

$$Y = (I - \mathbb{V}_k \mathbb{V}_k^T) ((\sigma_{k+1}I_n - A)^{-1}R_B(\sigma_{k+1})R_{k+1}) \neq 0.$$

We observe that,

$$\begin{aligned} (R_B(\sigma_{k+1})R_{k+1})^T Y &= (R_B(\sigma_{k+1})R_{k+1})^T (\sigma_{k+1}I_n - A)^{-1}R_B(\sigma_{k+1})R_{k+1} \\ &\quad - (R_B(\sigma_{k+1})R_{k+1})^T \mathbb{V}_k \mathbb{V}_k^T ((\sigma_{k+1}I_n - A)^{-1}R_B(\sigma_{k+1})R_{k+1}). \end{aligned}$$

Using the fact that the residual $R_B(\sigma_{k+1})$ is orthogonal to $[V_1, \dots, V_k]$, we get

$$R_B(\sigma_{k+1})R_{k+1})^T \mathbb{V}_k \mathbb{V}_k^T ((\sigma_{k+1}I_n - A)^{-1}R_B(\sigma_{k+1})R_{k+1}) = 0,$$

and then

$$(R_B(\sigma_{k+1})R_{k+1})^T Y = (R_B(\sigma_{k+1})R_{k+1})^T (\sigma_{k+1}I_n - A)^{-1}R_B(\sigma_{k+1})R_{k+1}.$$

Now, as $\mathscr{W}[(\sigma_{k+1}I_n - A)^{-1}] \subset \mathbb{C}^+ / \{0\}$, we have $Y \neq 0$, which proves the second assertion.

□

Proposition 3.5. Let $A_{m+1} = \mathbb{V}_{m+1}^T A \mathbb{V}_{m+1} = [a_{:,1}, \dots, a_{:,m+1}]$, where $a_{:,i} \in \mathbb{R}^{(m+1)s \times s}$ are the i -th block column of the $(m+1)s \times (m+1)s$ matrix A_{m+1} , and $\tilde{\mathbb{H}}_m = [\tilde{\mathbb{H}}^{(1)}, \dots, \tilde{\mathbb{H}}^{(m)}]$ is the upper Hessenberg matrix obtained from Algorithm 2. The, for $j = 1, \dots, m$, we have

$$a_{:,j+1} = \left[\sigma_{j+1} \tilde{\mathbb{H}}^{(j)} - [a_{:,1}, \dots, a_{:,j}] \tilde{\mathbb{H}}_{1:j,s}^{(j)} - B_{m+1} R_{j+1} \right] H_{j+1,j}^{-1}. \quad (31)$$

Proof We have from Algorithm 2

$$V_{j+1} H_{j+1,j} = (\sigma_{j+1} I_n - A)^{-1} B R_{j+1} - \sum_{i=1}^j V_i H_{i,j} \quad j = 1, \dots, m.$$

Multiplying on the left by $(\sigma_{j+1} I_n - A)$, and re-arranging terms, we get

$$A V_{j+1} H_{j+1,j} = \sigma_{j+1} \sum_{i=1}^{j+1} V_i H_{i,j} - A \sum_{i=1}^j V_i H_{i,j} - B R_{j+1},$$

which gives the following relation

$$A V_{j+1} H_{j+1,j} = \sigma_{j+1} \mathbb{V}_{m+1} \tilde{\mathbb{H}}^{(j)} - A \mathbb{V}_j \tilde{\mathbb{H}}_{1:j,s}^{(j)} - B R_{j+1}.$$

Multiplying now on the left by \mathbb{V}_{m+1}^T , we obtain

$$a_{:,j+1} H_{j+1,j} = \sigma_{j+1} \tilde{\mathbb{H}}^{(j)} - [a_{:,1}, \dots, a_{:,j}] \tilde{\mathbb{H}}_{1:j,s}^{(j)} - B_{m+1} R_{j+1},$$

which gives the desired result

$$a_{:,j+1} = \left[\sigma_{j+1} \tilde{\mathbb{H}}^{(j)} - [a_{:,1}, \dots, a_{:,j}] \tilde{\mathbb{H}}_{1:j,s}^{(j)} - B_{m+1} R_{j+1} \right] H_{j+1,j}^{-1}.$$

□

Proposition 3.5 allows us to compute the matrix A_{m+1} without computing the inverse of the $(m+1)s \times (m+1)s$ matrix \mathbb{H}_{m+1} as in (16), we only need the inverse of small matrices $H_{j+1,j}$, $j = 1, \dots, m$.

4. Numerical experiments

In this section, we give some numerical examples to show the effectiveness of our adaptive block tangential Arnoldi method (ABTAA). All the experiments presented in this paper were carried out using the CALCULCO computing platform, supported by SCoSI/ULCO (Service Commun du Système d'Information de l'Université du Littoral Côte d'Opale). The algorithms were coded in Matlab R2017a. We used the following functions from LYAPACK [29]:

- `lp_lgfrq`: Generates a set of logarithmically distributed frequency sampling points.
- `lp_para`: Used for computing the initial first two shifts.
- `lp_gnorm`: Computes $\|H(j\omega) - H_m(j\omega)\|_2$.

We used various matrices from LYAPACK and from the Oberwolfach collection¹. These matrix tests are reported in Table 1 with different values of p and the used values of s .

Example 1: The model of the first experiment is a model of stage 1R of the International Space Station (ISS). It has 270 states, three inputs and three outputs; for more details on this system, see [22]. Figure 1 shows the singular values of the transfer function and its approximation. In Figure 2, we plotted the 2-norm of the errors $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-6}, 10^6]$ for $m = 15$.

¹Oberwolfach model reduction benchmark collection 2003. <http://www.imtek.de/simulation/benchmark>

Table 1: Matrix Tests

Model	n	p	s
CDplayer	$n = 120$	$p = 2$	$s = 1$
ISS	$n = 270$	$p = 3$	$s = 2$
RAIL3113	$n = 3113$	$p = 6$	$s = 2$
MNA ₂	$n = 9223$	$p = 18$	$s = 6$
FLOW	$n = 9669$	$p = 5$	$s = 3$
FDM10000	$n = 10\ 000$	$p = 9$	$s = 3$
MNA ₅	$n = 10\ 913$	$p = 9$	$s = 3$
RAIL20209	$n = 20\ 209$	$p = 7$	$s = 3$
RAIL79841	$n = 79\ 841$	$p = 7$	$s = 3$
FDM40000	$n = 40\ 000$	$p = 9$	$s = 3$
FDM90000	$n = 90\ 000$	$p = 9$	$s = 3$

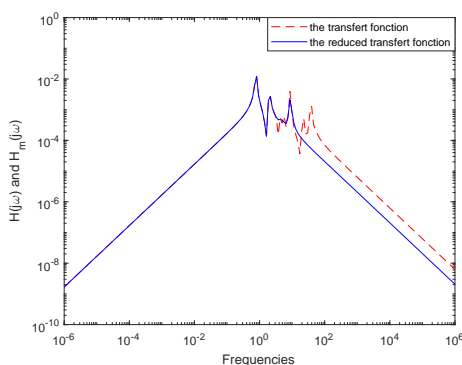


Figure 1: The ISS model: singular values vs frequencies.

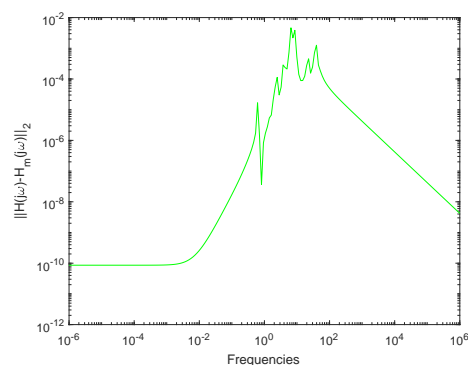


Figure 2: The ISS model: error-norms vs frequencies.

Example 2: In this example we used the CDplayer model, that describes the dynamics between a lens actuator and the radial arm position in a portable CD player. The model is relatively hard to reduce. For more details on this system, see [21]. Figure 3, represents the sigma-plot (the singular values of the transfer function) of the original system (dashed-dashed line) and the one of the reduced order system (solid line). In Figure 4, we plotted the error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-6}, 10^6]$.

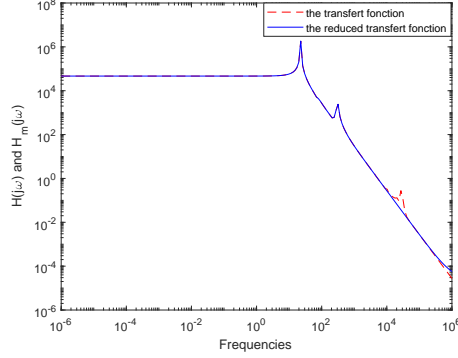


Figure 3: The CDplayer model: singular values vs frequencies.

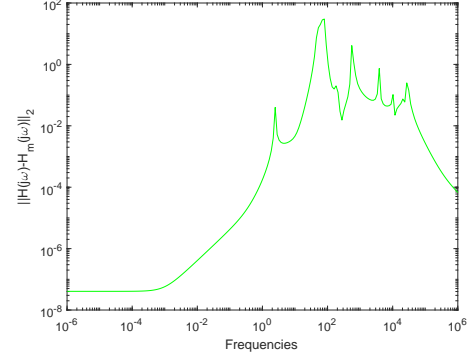


Figure 4: The CDplayer model: error-norms vs frequencies.

Example 3: In this example we compared the ABTAA algorithm with the Iterative Rational Krylov Algorithm (IRKA [20]) and the adaptive tangential method represented by Druskin and Simonsini (TRKSM) see for more details [11]. We used seven models: FDM, MNA₂, MNA₅, RAIL3113, RAIL20209, RAIL79841 and FLOW. The FDM 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 $f(x, y) = \log(x + 2y + 1)$, $g(x, y) = e^{x+y}$ and $h(x, y) = x + y$. The matrices B and C were random matrices with entries uniformly distributed in $[0, 1]$. The number of inner grid points in each direction was $n_0 = 100$ and the dimension of A is $n = n_0^2 = 10000$.

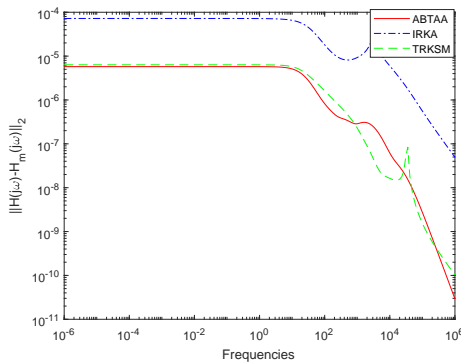


Figure 5: The FDM model: ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line), $m = 20$.

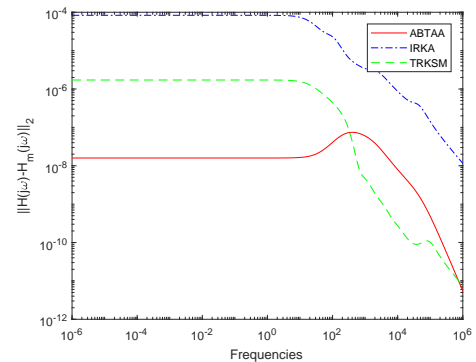


Figure 6: The FDM model: ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line), $m = 30$.

The MNA_2 and MNA_5 models were obtained from NICONET [26]. Figures 7 and 8 represent the exact error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies for ABTAA (solid line) and IRKA (dashed-dotted line) with $m = 20$.

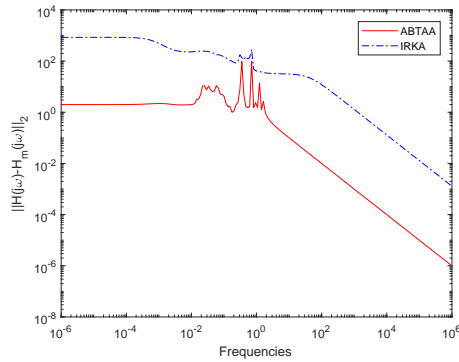


Figure 7: The MNA_2 model: ABTAA (solid line) & IRKA (dashed-dotted line), $m = 20$.

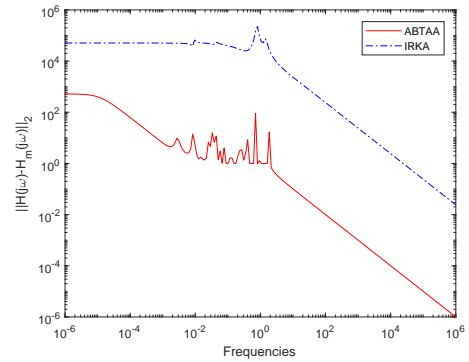


Figure 8: The MNA_5 model: ABTAA (solid line) & IRKA (dashed-dotted line), $m = 20$.

The models RAIL3113 ($n = 3113$, $p = 6$) and Flow ($n = 9669$, $p = 5$) are from the Oberwolfach collection. Figures 9 and 10 illustrate the error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies for $m = 20$. The execution time for the RAIL3113 is as follows: (ABTAA: 0.59 seconds, TRKSM: 2.17 seconds and IRKA: 15.21 seconds) and for the Flow model (ABTAA: 1.69 seconds, TRKSM: 7.39 seconds, IRKA: 42.50 seconds).

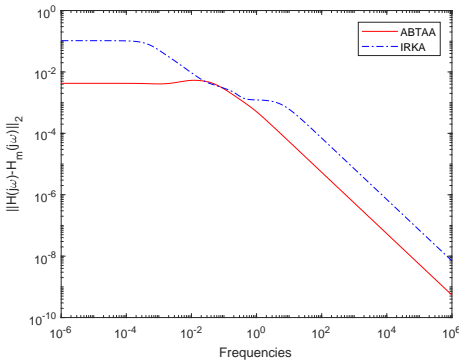


Figure 9: The RAIL3113 model: ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line), $m = 20$.

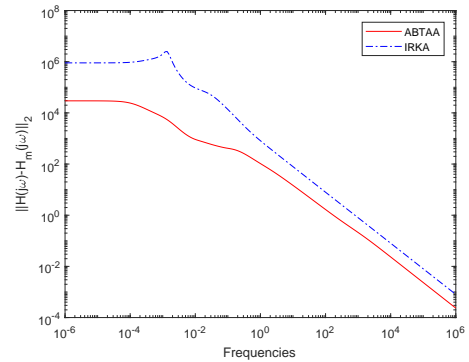


Figure 10: The Flow model: ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line), $m = 20$.

In the plots below, we used RAIL20209 ($n=20209$, $p=6$) and RAIL79841 ($n=79841$, $p=6$) models with a fixed $m = 12$, the matrices B and C were random. Figures 11 and 12 represent the exact error $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies of the tree methods ABTAA (solid line), IRKA (dashed-dotted line) and TRKSM (dashed-dashed line). The execution time for

the RAIL20209 example is the following: (ABTAA: 2.92 seconds, TRKSM: 9.92 seconds and IRKA: 44.08 seconds) and for RAIL79841 model is: (ABTAA: 32.69 seconds, TRKSM: 80.93 seconds, IRKA: 247.64 seconds).

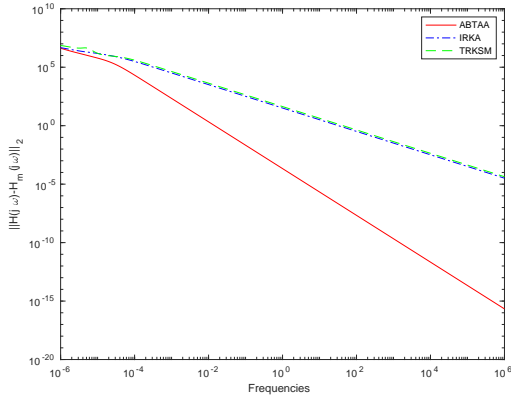


Figure 11: The RAIL20209 model: ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line).

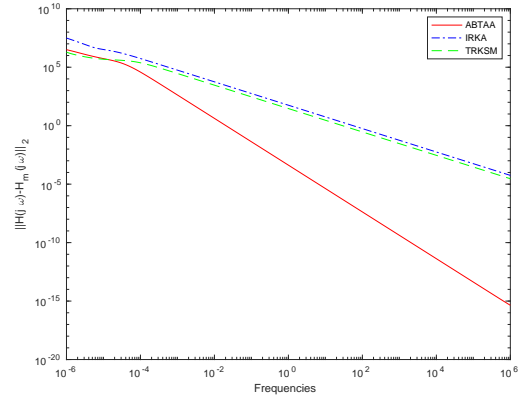


Figure 12: The RAIL79841 model: ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line).

Example 4: In this example, we used the FDM model: ($n = 40.000$ and $n = 90.000$ with $p = 9$). In Table 2, we compared the execution times and the \mathcal{H}_∞ norm $\|H - H_m\|_{\mathcal{H}_\infty}$ for ABTAA, IRKA and TRKSM algorithms with different values of m . We notice that the obtained timing didn't contain the execution times used to obtain the errors. As can be seen from the results in Table 2, the cost of IRKA and TRKSM methods is much higher than the cost required with the adaptive block tangential Arnoldi method.

Table 2: The computation time & the Err- \mathcal{H}_∞ error-norm

Model		ABTAA		IRKA		TRKSM	
		Time	Err- \mathcal{H}_∞	Time	Err- \mathcal{H}_∞	Time	Err- \mathcal{H}_∞
FDM40.000	m=10	9.30s	5.39×10^{-4}	126.28s	2.24×10^{-5}	34.89s	7.9×10^{-4}
	m=20	13.29s	3.87×10^{-5}	269.3s	1.06×10^{-4}	36.82s	1.93×10^{-5}
	m=30	19.15	3.08×10^{-7}	382.70s	3.30×10^{-4}	37.48s	7.84×10^{-7}
FDM90.000	m=10	43.29	6.49×10^{-4}	354.12s	1.55×10^{-4}	126.97s	1.25×10^{-4}
	m=20	52.72	1.46×10^{-4}	725.17s	1.44×10^{-4}	128.20s	9.83×10^{-5}
	m=30	64.24	1.90×10^{-5}	1025.68s	6.48×10^{-5}	127.88s	2.15×10^{-5}

5. Conclusion

In this paper, we proposed a new approach named block tangential Arnoldi method based on block tangential Krylov subspaces, to obtain reduced order dynamical systems, that approximate the initial large scale dynamical systems with multiple inputs and multiple outputs (MIMO). The method constructs sequences of orthogonal blocks from block tangential Krylov subspaces using the block Arnoldi approach. The interpolation shifts and the tangential directions are selected in an adaptive way by maximizing the residual norms. We gave some new algebraic properties and present some numerical experiments on some benchmark examples showing that the proposed method returns good results, as compared to some well known methods for large problems.

References

- [1] A. C. Antoulas, C. A. Beattie and S. Gugercin, Interpolatory Model Reduction of Large-scale Dynamical Systems. *Effi. Modeli. Contr. Larg. Sca. Syst.*, (2010), 3–58.
- [2] Z. Bai, Krylov subspace techniques for reduced-order modeling of large scale dynamical systems. *Appl. Numer. Math.*, 43(2002), 9–44.
- [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(2015), 221–236.
- [4] C. A. Beattie, S. Gugercin, Interpolation projection methods for structure-preserving model reduction. *Syst. Control Lett.*, 58(2009), 225–232.
- [5] 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.*, 75(2018), 1614–1632.
- [6] 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.
- [7] B. N. Datta, Large-Scale Matrix computations in Control. *Appl. Numer. Math.*, 30(1999), 53–63.
- [8] B. N. Datta, Krylov Subspace Methods for Large-Scale Matrix Problems in Control. *Future Gener. Comput. Syst.*, 19(2003), 1253–1263.
- [9] C. De Villemagne, R. Skelton, Model reductions using a projection formulation. *Internat. J. Contr.*, 46(1987), 2141–2169.

- [10] 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(2010), 2485–2496.
- [11] V. Druskin, V. Simoncini, Adaptive rational Krylov subspaces for large-scale dynamical systems. *Syst. Contr. Lett.*, 60(2011), 546–560.
- [12] V. Druskin, V. Simoncini, and M. Zaslavsky, Adaptive Tangential Interpolation in Rational Krylov Subspaces for MIMO Dynamical Systems. *SIAM. J. Matrix Anal. Appl.*, 35(2014), 476–498.
- [13] P. Feldmann, R. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Comput. Aid. Desi.*, 14(1995), 639–649.
- [14] M. Frangos, I.M. Jaimoukha, Adaptive rational interpolation: Arnoldi and Lanczos-like equations. *Eur. J. Control.*, 14(2008), 342–354.
- [15] K. Gallivan, A. Vandendorpe and P. Dooren, Model reduction of MIMO systems via tangential interpolation. *SIAM. J. Matrix. Anal. Appl.*, 26(2006), 328–349.
- [16] K. Gallivan, E. Grimme and P. Van Dooren, Asymptotic wave form evaluation via a Lanczos method. *Appl. Math. Lett.*, 7(1994), 75–80.
- [17] K. Glover, All optimal Hankel-norm approximations of linear multivariable systems and their L1-errors bounds. *Internat. J. Contr.*, 39(1984), 1115–1193.
- [18] 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. Contr. Optim.*, 29(1991), 283–324.
- [19] E. Grimme, Krylov projection methods for model reduction. Ph.D. thesis, Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, 1997.
- [20] S. Gugercin, A.C. Antoulas and C. Beattie, A rational Krylov iteration for optimal \mathcal{H}_2 model reduction. *J. Comp. Appl. Math.*, 53(2006), 1665–1667.
- [21] S. Gugercin, A. C. Antoulas, N. Bedrossian, Approximation of the international space station 1R and 12A models, in the Proceedings of the 40th CDC, 2001.
- [22] S. Guttel, L. Knizhnerman, Automated parameter selection for rational Arnoldi approximation of Markov functions. *Proc. Appl. Math. Mecha.*, 11(2011), 15–18.
- [23] M. Heyouni, K. Jbilou, Matrix Krylov subspace methods for large scale model reduction problems. *App. Math. Comput.*, 181(2006), 1215–1228.
- [24] I. M. Jaimoukha, E. M. Kasenally, Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 31(1994), 227–251.

- [25] A. J. Laub, M. T. Heath, C. C. Paige and R. C. Ward, Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Trans. Automat. Control.*, 32(1987), 115–122.
- [26] V. Mehrmann, T. Penzl, Benchmark collections in SLICOT. Technical Report SLWN 1998–5, SLICOT Working Note, ESAT, KU Leuven, K. Mercierlaan 94, Leuven-Heverlee 3100, Belgium, 1998. Available under <http://www.win.tue.nl/niconet/NIC2/reports.html>.
- [27] B. C. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Automat. Control.*, 26(1981), 17–32.
- [28] V. Olshevsky, A. Shokrollahi, A Super fast Algorithm for Confluent Rational Tangential Interpolation Problem via Matrix-vector Multiplication for Confluent Cauchy-like Matrices, In *Structured Matrices in Mathematics. Comput. Scie. Eng. Contempo. Math.*, 280(2001), 32–46.
- [29] T. Penzl, LYAPACK MATLAB toolbox for Large Lyapunov and Riccati Equations, Model Reduction Problems, and Linear-quadratic Optimal Control Problems, <http://www.tuchemintz.de/sfb393/lyapack>.
- [30] A. Ruhe, Rational Krylov algorithms for nonsymmetric eigenvalue problems. II. matrix pair. *Lin. Alg. Appl.*, (197-198)(1994), 283–295.
- [31] A. Ruhe, Rational Krylov sequence methods for eigenvalue computation. *Lin. Alg. Appl.*, 58(1984), 391–405.
- [32] A. Ruhe, The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: complex shifts for real matrices. *BIT Numer. Math.*, 34(1994), 165–176.
- [33] M. G. Safonov, R. Y. Chiang, A Schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control.*, 34(1989), 729–733.
- [34] Y. Shamash, Stable reduced-order models using Padé type approximations, *IEEE. Trans. Automa. Control.*, 19(1974), 615–616.
- [35] P. Van Dooren, K. A. Gallivan, and P. Absil, \mathcal{H}_2 -optimal model reduction with higher order poles. *SIAM. J. Matrix Anal. Appl.*, 31(2010), 2738–2753.
- [36] P. Vuillemin, C. Poussot-Vassal and D. Alazard, Proceedings of the "7eme Conférence Internationale Francophone d'Automatique", 2012.
- [37] A. Yousuff, R. Skelton, Covariance equivalent realizations with applications to model reduction of large-scale systems. *Contr Dynam. Syst.*, 22(1985), 273–348.