Numerical Algorithms (2020) 83:629–652
https://doi.org/10.1007/s11075-019-00696-9

ORIGINAL PAPER

A tangential method for the balanced truncation in model reduction

Y. Kaouane[1,2]

Received: 19 November 2018 / Accepted: 26 March 2019 / Published online: 11 April 2019

Abstract
In this paper, we present a new approach for large-scale Lyapunov matrix equations, where we present two algorithms named: Adaptive Block Tangential Lanczos-type and Arnoldi-type algorithms (ABTL and ABTA). This approach is based on the projection of the initial problem onto tangential Krylov subspaces to produce a low-rank approximate solution of large Lyapunov equations. These approximations are used in model reduction of large-scale dynamical systems with multiple inputs and multiple outputs (MIMO). We give some algebraic properties and present some numerical experiences to show the effectiveness of the proposed algorithms.

Keywords Balanced truncation · Krylov subspaces · Lyapunov · Model reduction · Tangential directions

1 Introduction

Consider a multi-input, multi-output linear time-invariant (LTI) dynamical system, described by the state-space equations as follows:

$$\Sigma := \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t), \end{cases} \tag{1.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t)$ and $y(t)$ are the input and the output signal vectors, respectively. The matrix $A \in \mathbb{R}^{n\times n}$ is assumed to be large and sparse

✉ Y. Kaouane
yassine.kaouane@etu.univ-littoral.fr

[1] Laboratoire de Mathématiques Pures et Appliquées, Université du Littoral, Côte d'Opale, Calais, France

[2] Laboratoire de Mathématiques Appliquées et Informatique, Université Cadi Ayyad, Marrakech, Morocco

and $B,\ C^T \in \mathbb{R}^{n\times p}$. The transfer function associated to the system in (1.1) is given as follows:

$$H(\omega) := C(\omega I_n - A)^{-1}B \in \mathbb{R}^{p\times p}. \tag{1.2}$$

The goal of our model reduction approach consists in defining two orthogonal matrices V_m and $W_m \in \mathbb{R}^{n\times m}$ (with $m \ll n$) to produce a much smaller order system Σ_m with the state-space form as follows:

$$\Sigma_m : \begin{cases} \dot{x}_m(t) &= A_m x_m(t) + B_m u(t) \\ y_m(t) &= C_m x_m(t), \end{cases} \tag{1.3}$$

and its transfer function is defined by the following:

$$H_m(\omega) := C_m(\omega I_m - A_m)^{-1}B_m \in \mathbb{R}^{p\times p}, \tag{1.4}$$

where $A_m = W_m^T A V_m \in \mathbb{R}^{m\times m}$, $B_m = W_m^T B \in \mathbb{R}^{m\times p}$ and $C_m = C V_m \in \mathbb{R}^{p\times m}$, such that the reduced system Σ_m 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, $\|y - y_m\|$ is small.

Assume that the matrix $(\omega I_n - A)$ is non-singular and define $X = (\omega I_n - A)^{-1}B$ and $Y = (\omega I_n - A)^{-T}C^T$. Then, we have $H(\omega) = CX = Y^T B$, and X and Y solves the multiple linear systems of equations as follows:

$$(\omega I_n - A)X = B, \quad \text{and} \quad (\omega I_n - A)^T Y = C^T. \tag{1.5}$$

An approximate solution $X_m \in \text{Range}(V_m)$ and $Y_m \in \text{Range}(W_m)$ can be determined by imposing the Galerkin condition on the residuals $R_B(\omega) \perp \text{Range}(W_m)$ and $R_C(\omega) \perp \text{Range}(V_m)$, where,

$$R_B = B - (\omega I_n - A)X_m,$$

and

$$R_C = C^T - (\omega I_n - A)^T Y_m,$$

which gives $X_m = V_m(\omega I_m - A_m)^{-1}W_m^T B$ and $Y_m = W_m(\omega I_m - A_m)^{-T}V_m^T C^T$, hence the (1.4) and the residuals can be expressed as follows:

$$R_B(\omega) = B - (\omega I_n - A)V_m(\omega I_m - A_m)^{-1}W_m^T B, \tag{1.6}$$

$$R_C(\omega) = C^T - (\omega I_n - A)^T W_m(\omega I_m - A_m)^{-T}V_m^T C^T. \tag{1.7}$$

Various model reduction techniques, such as Padé approximation [12, 19], balanced truncation [20], optimal Hankel norm [11], and Krylov subspace methods [5, 6, 9, 15], have been used for large multi-input multi-output (MIMO) dynamical systems (see [2, 4, 11, 14]). The balanced truncation model reduction (BTMR) method is a very popular method [1, 10]; the method preserves the stability and provides a bound for the approximation error. In the case of small to medium systems, BTMR can be implemented efficiently. However, for large-scale settings, the method is quite expensive to implement, because it requires the computation of two Lyapunov equations, and results in a computational complexity of $\mathcal{O}(n^3)$ and a storage requirement

of $\mathcal{O}(n^2)$ (see [1, 3, 13]). In this paper, we project the Lyapunov equations using the block tangential Krylov subspaces defined as follows:

$$\widetilde{\mathcal{K}}_m(A, B) = \text{Range}\{B, (\sigma_1 I_n - A)^{-1} B R_1, ..., (\sigma_m I_n - A)^{-1} B R_m\},$$
$$\widetilde{\mathcal{K}}_m(A^T, C^T) = \text{Range}\{C^T, (\mu_1 I_n - A)^{-T} C^T L_1, ..., (\mu_m I_n - A)^{-T} C^T L_m\},$$

in order to obtain small-scale Lyapunov equations. The $\{\sigma_i\}_{i=1}^m$ and $\{\mu_i\}_{i=1}^m$ are the right and left interpolation points and the $\{R_i\}_{i=1}^m$ and $\{L_i\}_{i=1}^m$ are the right and left blocks tangent directions with $R_i,\ L_i \in \mathbb{R}^{p\times s}$ with $s \leq p$. Later, we will show how to choose these tangent interpolation points and directions.

The paper is organized as follows: In Section 2, we give some definitions used later and we introduce the balanced truncation method. In Section 3, we present the tangential block Lanczos-type method and the corresponding algorithm to solve a large-scale Lyapunov equation. Section 4 is devoted to the selection of the interpolation points and the tangential directions that are used in the construction of block tangential Krylov subspaces, and we present briefly the tangential block Lyapunov Arnoldi-type algorithm. The last section is devoted to numerical tests and comparisons with some well-known model-order reduction methods.

2 The balanced truncation method

First, we recall some definitions that will be used in this paper.

Definition 2.1 Given a stable LTI dynamical system (1.1), the associated controllability Gramian, denoted by $\mathcal{P}$, is defined as follows:

$$\mathcal{P} = \int_0^\infty e^{tA} B B^T e^{tA^T} dt, \tag{2.1}$$

and the observability Gramian $\mathcal{Q}$ is defined as follows:

$$\mathcal{Q} = \int_0^\infty e^{tA^T} C^T C e^{tA} dt. \tag{2.2}$$

The two Gramians can be computed by solving two equations. In fact, $\mathcal{P}$ and $\mathcal{Q}$ are the unique solutions of the following Lyapunov matrix equations:

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0, \tag{2.3}$$

and

$$A^T\mathcal{Q} + \mathcal{Q}A + C^T C = 0. \tag{2.4}$$

Definition 2.2 The $\mathcal{H}_\infty$-norm of the transfer function H is defined by the following:

$$\| H \|_{\mathcal{H}_\infty} = \max_{\omega\in\mathbb{R}} \gamma_{\max}(H(j\omega)), \tag{2.5}$$

where $\gamma_{\max}$ denotes the maximum singular value.

The balanced truncation method for model reduction was first introduced by Mullis and Roberts [17] and later in systems and control theory by Moore and Glover

(see [10, 16]). If we assume that the LTI system (1.1) is stable, controllable, and observable, in this case, we call it also stable and minimal, then the controllability and observability Gramians are unique positive definite. The balanced truncation of a LTI dynamical model Σ is obtained by applying a non-singular matrix transformation $T \in \mathbb{R}^{n\times n}$ to get the following:

$$\widetilde{A} = T^{-1}AT, \qquad \widetilde{B} = T^{-1}B, \qquad \widetilde{C} = CT.$$

Hence, the associated controllability and observability Gramians $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$ are expressed as follows:

$$\widetilde{\mathcal{P}} = T^{-1}\mathcal{P}T^{-T}, \qquad \widetilde{\mathcal{Q}} = T^{T}\mathcal{Q}T.$$

The aim of the balanced truncation method is to find the transformation T such that the new Gramians $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$ are diagonal as follows:

$$\widetilde{\mathcal{P}} = \widetilde{\mathcal{Q}} = \mathbf{diag}(\gamma_1, ..., \gamma_n), \tag{2.6}$$

where the γ_i, $i = 1, ..., n$ are called the Hankel singular values. For controllable, observable, and stable systems, they can be computed as follows:

$$\gamma_i = \sqrt{\lambda_i(\mathcal{P}\mathcal{Q})}.$$

Notice that the Hankel singular values are invariant by transformation, contrary to the Gramians. Now, we show how to obtain the Gramians $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$ that verify (2.6). First, we compute directly the lower Cholesky factorization of the Gramians $\mathcal{P}$ and $\mathcal{Q}$ as follows:

$$\mathcal{P} = L_c L_c^T, \qquad \mathcal{Q} = L_o L_o^T,$$

then, we compute the singular value decomposition of the matrix $L_o^T L_c$,

$$L_o^T L_c = U\mathcal{D}V^T, \tag{2.7}$$

where $\mathcal{D}$ is the diagonal matrix containing the Hankel singular values of the system (1.1). The balanced transformation is given by the following:

$$T = L_c V \mathcal{D}^{-\frac{1}{2}}, \qquad T^{-1} = \mathcal{D}^{-\frac{1}{2}} U^T L_o^T.$$

It is proved in [1] that if the system (1.1) is stable and minimal, and having the new equivalent LTI dynamical system $\widetilde{\Sigma}$ as follows:

$$\widetilde{\Sigma} := \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & 0 \end{array}\right] \equiv \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & 0 \end{array}\right],$$

with $\widetilde{\mathcal{P}} = \widetilde{\mathcal{Q}} = \mathbf{diag}(\gamma_1, ..., \gamma_m, \gamma_{m+1}, ..., \gamma_n)$, then,

$$\| H(.) - H_m(.) \|_{\mathcal{H}_\infty} \leq 2(\gamma_{m+1} + ... + \gamma_n). \tag{2.8}$$

Inequality (2.8) shows that the dynamical system (1.1) can be represented by a reduced-order LTI system $\widetilde{\Sigma}_m$, if the singular values $\gamma_{m+1}, ..., \gamma_n$ are small enough as follows:

$$\widetilde{\Sigma}_m \equiv \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & 0 \end{array}\right].$$

Let us now construct the reduced model Σ_m. First, we define the following matrices as follows:

$$\mathcal{V}_m = L_c V_m \mathcal{D}_m^{-\frac{1}{2}}, \qquad \mathcal{W}_m = L_o U_m \mathcal{D}_m^{-\frac{1}{2}},$$

where $\mathcal{D}_m$, U_m, and V_m correspond to the first m columns of the matrices $\mathcal{D}$, U, and V in (2.7). Then, the reduced model Σ_m is given as follows:

$$\Sigma_m \equiv \left[\begin{array}{c|c} A_m & B_m \\ \hline C_m & 0 \end{array}\right],$$

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

Next, we apply the tangential block Lanczos-type algorithm for solving large Lyapunov matrix equations that are used in the construction of reduced-order models using the balanced truncation method.

3 Tangential block Lanczos-type method for Lyapunov matrix equations

Consider the following Lyapunov matrix equations as follows:

$$AX^{(1)} + X^{(1)}A^T + BB^T = 0, \tag{3.1}$$

and

$$A^T X^{(2)} + X^{(2)}A + C^T C = 0, \tag{3.2}$$

where $A \in \mathbb{R}^{n\times n}$ is non-singular, $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{p\times n}$. To extract low-rank approximate solutions to the Lyapunov (3.1) and (3.2), we project the initial problems onto the following tangential block Krylov subspaces as follows:

$$\widetilde{\mathcal{K}}_m(A, B) = \text{Range}\{B, (\sigma_1 I_n - A)^{-1} B R_1, ..., (\sigma_m I_n - A)^{-1} B R_m\}, \tag{3.3}$$

and

$$\widetilde{\mathcal{K}}_m(A^T, C^T) = \text{Range}\{C^T, (\mu_1 I_n - A)^{-T} C^T L_1, ..., (\mu_m I_n - A)^{-T} C^T L_m\}, \tag{3.4}$$

where $\{\sigma_i\}_{i=1}^m$ and $\{\mu_i\}_{i=1}^m$ are the right and left interpolation points respectively and $\{R_i\}_{i=1}^m$, $\{L_i\}_{i=1}^m$ are the right and left tangent directions with $R_i,\ L_i \in \mathbb{R}^{p\times s}$.

A tangential Lanczos-type method consists in constructing two bi-orthonormal bases, spanned by the columns of $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$, of the tangential Krylov subspaces $\widetilde{\mathcal{K}}_m(A^T, C^T)$, respectively.
Let $\mathbb{V}_m = [V_1, V_2, \ldots, V_m]$ and $\mathbb{W}_m = [W_1, W_2, \ldots, W_m]$. Then, we should have the bi-orthogonality conditions for $i, j = 1, \ldots, m$:

$$\begin{cases} W_i^T V_j = I, & i = j, \\ W_i^T V_j = 0, & i \neq j. \end{cases} \tag{3.5}$$

The low-rank approximate solutions $\mathcal{X}_m^{(1)}$ and $\mathcal{X}_m^{(2)}$ to the solutions of the Lyapunov matrix (3.1) and (3.2) are defined as follows:

$$\mathcal{X}_m^{(1)} = \mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{V}_m^T, \qquad \mathcal{X}_m^{(2)} = \mathbb{W}_m \mathcal{Y}_m^{(2)} \mathbb{W}_m^T, \tag{3.6}$$

such that the following Galerkin conditions are satisfied as follows:

$$\mathbb{W}_m^T \mathcal{R}_1(\mathcal{X}_m^{(1)})\mathbb{W}_m = 0, \qquad \mathbb{V}_m^T \mathcal{R}_2(\mathcal{X}_m^{(2)})\mathbb{V}_m = 0, \tag{3.7}$$

where the residuals are given by the following:

$$\mathcal{R}_1(\mathcal{X}_m^{(1)}) = A\mathcal{X}_m^{(1)} + \mathcal{X}_m^{(1)}A^T + BB^T, \tag{3.8}$$

and

$$\mathcal{R}_2(\mathcal{X}_m^{(2)}) = A^T\mathcal{X}_m^{(2)} + \mathcal{X}_m^{(2)}A + C^TC. \tag{3.9}$$

Replacing $\mathcal{X}_m^{(1)}$ and $\mathcal{X}_m^{(2)}$ in (3.7), we obtain the following:

$$\mathbb{W}_m^T A\mathbb{V}_m\mathcal{Y}_m^{(1)} + \mathcal{Y}_m^{(1)}\mathbb{V}_m^T A^T\mathbb{W}_m + \mathbb{W}_m^T BB^T\mathbb{W}_m = 0,$$

and

$$\mathbb{V}_m^T A^T\mathbb{W}_m\mathcal{Y}_m^{(2)} + \mathcal{Y}_m^{(2)}\mathbb{W}_m^T A\mathbb{V}_m + \mathbb{V}_m^T C^TC\mathbb{V}_m = 0,$$

which gives the low-dimensional Lyapunov matrix equations as follows:

$$A_m\mathcal{Y}_m^{(1)} + \mathcal{Y}_m^{(1)}A_m^T + B_mB_m^T = 0, \tag{3.10}$$

and

$$A_m^T\mathcal{Y}_m^{(2)} + \mathcal{Y}_m^{(2)}A_m + C_m^TC_m = 0, \tag{3.11}$$

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

The main problem now is the computation of the two bi-orthogonal bases $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$ of the tangential Krylov subspaces in (3.3) and (3.4). The following Block Tangential Lanczos (BTL) algorithm allows us to construct such bases. It is summarized in the following steps.

Here, we suppose that we already have the set of interpolation points $\sigma = \{\sigma_i\}_{i=1}^m$, $\mu = \{\mu_i\}_{i=1}^m$ and the tangential matrix directions $R = \{R_i\}_{i=1}^m$ and $L = \{L_i\}_{i=1}^m$. The upper block upper Hessenberg matrices $\widetilde{\mathbb{H}}_m = \left[\widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)}\right]$ and $\widetilde{\mathbb{F}}_m = \left[\widetilde{\mathbb{F}}^{(1)}, ..., \widetilde{\mathbb{F}}^{(m)}\right] \in \mathbb{R}^{(ms+p)\times ms}$ are obtained from the BTL algorithm, with the following:

$$\widetilde{\mathbb{H}}^{(j)} = \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} \text{ and } \widetilde{\mathbb{F}}^{(j)} = \begin{bmatrix} F_{1,j} \\ \vdots \\ F_{j,j} \\ F_{j+1,j} \\ \mathbf{0} \end{bmatrix}, \quad \text{for } j = 1, ..., m.$$

The matrices $H_{i,j}$ and $F_{i,j}$ constructed in step 3 of Algorithm 1 are of size $p \times s$ if $i = 1$ and are of size $s \times s$ otherwise. We define the $(ms + p) \times p$ matrices $\widetilde{\mathbb{H}}^{(0)}$ and $\widetilde{\mathbb{F}}^{(0)}$ as follows:

$$\widetilde{\mathbb{H}}^{(0)} = \begin{bmatrix} H_{1,0} \\ \mathbf{0} \end{bmatrix} \text{ and } \widetilde{\mathbb{F}}^{(0)} = \begin{bmatrix} F_{1,0} \\ \mathbf{0} \end{bmatrix},$$

where $\mathbf{0}$ is the zero matrix of size $(m - j) \times s$. We define also the following matrices as follows:

$$\widetilde{D}_m^{(1)} = \begin{bmatrix} O_p & O_{p,ms} \\ O_{ms,p} & D_m^{(1)} \otimes I_s \end{bmatrix}, \qquad \widetilde{D}_m^{(2)} = \begin{bmatrix} O_p & O_{p,ms} \\ O_{ms,p} & D_m^{(2)} \otimes I_s \end{bmatrix},$$

Algorithm 1 The Block Tangential Lanczos (BTL) algorithm.

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

– Output: $\mathbb{V}_{m+1} = \left[V_1, ..., V_{m+1}\right]$, $\mathbb{W}_{m+1} = \left[W_1, ..., W_{m+1}\right]$.

- Compute $B = V_1 H_{1,0}, \quad C^T = W_1 F_{1,0}$ (QR decomposition).
- Initialize: $\mathbb{V}_1 = [V_1]$, $\mathbb{W}_1 = [W_1]$.
- For j = 1,...,m
  1. If $\sigma_j \neq \infty$, $\widetilde{V}_{j+1} = (\sigma_j I_n - A)^{-1} B R_j$, else $\widetilde{V}_{j+1} = AB R_j$.
  2. If $\mu_j \neq \infty$, $\widetilde{W}_{j+1} = (\mu_j I_n - A)^{-T} C^T L_j$, else $\widetilde{W}_{j+1} = AC^T L_j$.
  3. For i = 1,...,j
     - $H_{i,j} = W_i^T \widetilde{V}_{j+1}$, – $F_{i,j} = V_i^T \widetilde{W}_{j+1}$,
     - $\widetilde{V}_{j+1} = \widetilde{V}_{j+1} - V_i H_{i,j}$, – $\widetilde{W}_{j+1} = \widetilde{W}_{j+1} - W_i F_{i,j}$,
  4. End.
  5. $\widetilde{V}_{j+1} = V_{j+1} H_{j+1,j}$, $\widetilde{W}_{j+1} = W_{j+1} F_{j+1,j}$, (QR decomposition).
  6. $W_{j+1}^T V_{j+1} = P_j D_j Q_j^T$, (singular value decomposition).
  7. $V_{j+1} = V_{j+1} Q_j D_j^{-1/2}$, $W_{j+1} = W_{j+1} P_j D_j^{-1/2}$.
  8. $H_{j+1,j} = D_j^{1/2} Q_j^T H_{j+1,j}$, $F_{j+1,j} = D_j^{1/2} P_j^T F_{j+1,j}$.
  9. $\mathbb{V}_{j+1} = \left[\mathbb{V}_j,\ V_{j+1}\right]$, $\mathbb{W}_{j+1} = \left[\mathbb{W}_j,\ W_{j+1}\right]$.
- End

where $D_m^{(1)} = \text{Diag}\{\sigma_1, ..., \sigma_m\}$ and $D_m^{(2)} = \text{Diag}\{\mu_1, ..., \mu_m\}$. With all those notations, we have the following theorem.

Theorem 3.1 *Let $\mathbb{V}_{m+1}$ and $\mathbb{W}_{m+1}$ be the bi-orthonormal matrices of $\mathbb{R}^{n\times(ms+p)}$ constructed by Algorithm 1. Then, we have the following relations as follows:*

$$A\mathbb{V}_{m+1} = \left[\mathbb{V}_{m+1}\mathbb{G}_{m+1}\widetilde{D}_m^{(1)} - \mathbb{K}_{m+1}^B\right]\mathbb{G}_{m+1}^{-1}, \tag{3.12}$$

and

$$A^T\mathbb{W}_{m+1} = \left[\mathbb{W}_{m+1}\mathbb{Q}_{m+1}\widetilde{D}_m^{(2)} - \mathbb{K}_{m+1}^C\right]\mathbb{Q}_{m+1}^{-1}. \tag{3.13}$$

Let $\mathbb{T}_{m+1}$ and $\mathbb{Y}_{m+1}$ be the matrices as follows:

$$\mathbb{T}_{m+1} = \left[B, (\sigma_1 I - A)^{-1} B R_1, ..., (\sigma_m I - A)^{-1} B R_m\right] \text{ and}$$

$$\mathbb{Y}_{m+1} = \left[C^T, (\mu_1 I - A)^{-T} C^T L_1, ..., (\mu_m I - A)^{-T} C^T L_m\right],$$

then, we have the following:

$$\mathbb{T}_{m+1} = \mathbb{V}_{m+1}\mathbb{G}_{m+1} \text{ and } \quad \mathbb{Y}_{m+1} = \mathbb{W}_{m+1}\mathbb{Q}_{m+1}, \tag{3.14}$$

where $\mathbb{K}_{m+1}^B = [-AB \ \ B\mathbb{R}_m]$, $\mathbb{K}_{m+1}^C = \left[-A^TC^T \ \ C^T\mathbb{L}_m\right]$, $\mathbb{R}_m = [R_1, ..., R_m]$, and $\mathbb{L}_m = [L_1, ..., L_m]$. $\mathbb{G}_{m+1} = \left[\widetilde{\mathbb{H}}^{(0)} \ \ \widetilde{\mathbb{H}}_m\right]$ and $\mathbb{Q}_{m+1} = \left[\widetilde{\mathbb{F}}^{(0)} \ \ \widetilde{\mathbb{F}}_m\right]$ are block upper triangular matrices of sizes $(ms+p)\times(ms+p)$ and are assumed to be non-singular.

Proof From Algorithm 1, we have the following:

$$V_{j+1}H_{j+1,j} = (\sigma_j I_n - A)^{-1}BR_j - \sum_{i=1}^{j} V_i H_{i,j} \qquad j = 1, ..., m. \tag{3.15}$$

Multiplying (3.15) on the left by $(\sigma_j I_n - A)$ and re-arranging terms, we get the following:

$$A\sum_{i=1}^{j+1} V_i H_{i,j} = \sigma_j \sum_{i=1}^{j+1} V_i H_{i,j} - BR_j \qquad j = 1, ..., m,$$

which gives the following:

$$A\mathbb{V}_{j+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \end{bmatrix} = \sigma_j \mathbb{V}_{j+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \end{bmatrix} - BR_j, \qquad j = 1, \ldots, m,$$

that written as follows:

$$A\mathbb{V}_{m+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} = \sigma_j \mathbb{V}_{j+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} - BR_j, \qquad j = 1, \ldots, m, \tag{3.16}$$

where $\mathbf{0}$ is the zero matrix of size $(m - j) \times s$. Then, for $j = 1, \ldots, m$, we have the following:

$$A\mathbb{V}_{m+1}\widetilde{\mathbb{H}}^{(j)} = \sigma_j \mathbb{V}_{j+1}\widetilde{\mathbb{H}}^{(j)} - BR_j; \tag{3.17}$$

Therefore, we can deduce from (3.17), the following expression is as follows:

$$A\mathbb{V}_{m+1}\left[\widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)}\right] = \mathbb{V}_{m+1}\left[\widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)}\right](D_m^{(1)} \otimes I_s) - B\mathbb{R}_m,$$

Now, since $V_1 H_{1,0} = B$, we get the following:

$$A\mathbb{V}_{m+1}\left[\widetilde{\mathbb{H}}^{(0)}, \widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)}\right] = \mathbb{V}_{m+1}\left[\widetilde{\mathbb{H}}^{(0)}, \widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)}\right]\widetilde{D}_m^{(1)} - [-AB \;\; B\mathbb{R}_m],$$

which ends the proof of (3.12). The same proof can be done for the relation (3.13).

For the proof of (3.14), we first use (3.15) to obtain the following:

$$\sum_{i=1}^{j+1} V_i H_{i,j} = (\sigma_j I_n - A)^{-1}BR_j \qquad j = 1, \ldots, m,$$

which gives the following:

$$\mathbb{V}_{m+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} = (\sigma_j I_n - A)^{-1}BR_j, \qquad j = 1, \ldots, m.$$

It follows that

$$\mathbb{V}_{m+1}\left[\widetilde{\mathbb{H}}^{(0)}, \widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)}\right] = \left[B, (\sigma_1 I_n - A)^{-1}BR_1, ..., (\sigma_m I_n - A)^{-1}BR_m\right],$$

which ends the proof of the first relation of (3.14). In the same manner, we can prove the second relation. □

In the following theorem, we give the exact expression of the residual norms in a simplified and economical computational form.

Theorem 3.2 *Let $\mathbb{V}_m = [V_1, ..., V_m]$ and $\mathbb{W}_m = [W_1, ..., W_m]$ be the bi-orthonormal matrices obtained by Algorithm 1. Let $\mathcal{X}_m^{(1)} = \mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{V}_m^T$, $\mathcal{X}_m^{(2)} = \mathbb{W}_m \mathcal{Y}_m^{(2)} \mathbb{W}_m^T$, be the approximate solutions of the Lyapunov matrix (3.1) and (3.2), then the residual norms are given as follows:*

$$\| \mathcal{R}_1(\mathcal{X}_m^{(1)}) \|_2 = \| \mathbb{S}_m^{(1)} J (\mathbb{S}_m^{(1)})^T \|_2 \text{ and } \| \mathcal{R}_2(\mathcal{X}_m^{(2)}) \|_2 = \| \mathbb{S}_m^{(2)} J (\mathbb{S}_m^{(2)})^T \|_2, \tag{3.18}$$

where $\mathbb{S}_m^{(1)}$ and $\mathbb{S}_m^{(2)}$ are the upper triangular matrices obtained from the skinny QR decomposition of the matrices $\mathbb{U}_m^{(1)}$ and $\mathbb{U}_m^{(2)}$ defined by the following:

$$\mathbb{U}_m^{(1)} = \left[\mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{G}_m^{-T} \quad (\mathbb{V}_m \mathbb{W}_m^T - I_n)\mathbb{K}_m^B\right] \text{ and } \mathbb{U}_m^{(2)} = \left[\mathbb{W}_m \mathcal{Y}_m^{(2)} \mathbb{Q}_m^{-T} \quad (\mathbb{W}_m \mathbb{V}_m^T - I_n)\mathbb{K}_m^C\right].$$

The matrix J is defined as $J = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$.

Proof We know that in the following:

$$\begin{aligned} \mathcal{R}_1(\mathcal{X}_m^{(1)}) &= A\mathcal{X}_m^{(1)} + \mathcal{X}_m^{(1)} A^T + BB^T \\ &= A\mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{V}_m^T + \mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{V}_m^T A^T + BB^T. \end{aligned}$$

Using the (3.12), we get the following:

$$A_m = \mathbb{W}_m^T A \mathbb{V}_m = \left[\mathbb{G}_m \widetilde{D}_{m-1} - \mathbb{W}_m^T \mathbb{K}_m^B\right] \mathbb{G}_m^{-1},$$

which gives

$$A\mathbb{V}_m = \mathbb{V}_m A_m + (\mathbb{V}_m \mathbb{W}_m^T - I_n)\mathbb{K}_m^B \mathbb{G}_m^{-1}. \tag{3.19}$$

It follows that

$$\begin{aligned} \mathcal{R}_1(\mathcal{X}_m^{(1)}) &= \left[\mathbb{V}_m A_m + (\mathbb{V}_m \mathbb{W}_m^T - I_n)\mathbb{K}_m^B \mathbb{G}_m^{-1}\right] \mathcal{Y}_m^{(1)} \mathbb{V}_m^T \\ &\quad + \mathbb{V}_m \mathcal{Y}_m^{(1)} \left[\mathbb{V}_m A_m + (\mathbb{V}_m \mathbb{W}_m^T - I_n)\mathbb{K}_m^B \mathbb{G}_m^{-1}\right]^T + BB^T. \end{aligned}$$

Using the fact that $\mathcal{Y}_m^{(1)}$ solves the low-dimensional Lyapunov (3.10), we get the following:

$$\begin{aligned} \mathcal{R}_1(\mathcal{X}_m^{(1)}) &= (\mathbb{V}_m \mathbb{W}_m^T - I_n)\mathbb{K}_m^B \mathbb{G}_m^{-1} \mathcal{Y}_m^{(1)} \mathbb{V}_m^T + \mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{G}_m^{-T} (\mathbb{K}_m^B)^T (\mathbb{V}_m \mathbb{W}_m^T - I_n) \\ &= \left[\mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{G}_m^{-T} \quad (\mathbb{V}_m \mathbb{W}_m^T - I_n)\mathbb{K}_m^B\right] \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} \mathbb{G}_m^{-1} \mathcal{Y}_m^{(1)} \mathbb{V}_m^T \\ (\mathbb{K}_m^B)^T (\mathbb{V}_m \mathbb{W}_m^T - I_n) \end{bmatrix} \\ &= \mathbb{U}_m^{(1)} J (\mathbb{U}_m^{(1)})^T. \end{aligned}$$

We proceed in the same way for the proof of the second relation. □

4 An adaptive choice of the interpolation points and tangent directions

In the section, we will see how to chose the interpolation points $\{\sigma_i\}_{i=1}^m$, $\{\mu_i\}_{i=1}^m$ and tangential directions $\{R_i\}_{i=1}^m$, $\{L_i\}_{i=1}^m$, where $R_i,\ L_i \in \mathbb{R}^{p\times s}$. In this paper, we adopted the adaptive approach, inspired by the work in [7]. For this approach, we extend our subspaces $\widetilde{\mathcal{K}}_m(A, B)$ and $\widetilde{\mathcal{K}}_m(A^T, C^T)$ by adding new blocks $\widetilde{V}_{m+1}$ and $\widetilde{W}_{m+1}$ defined as follows:

$$\widetilde{V}_{m+1} = (\sigma_{m+1} I_n - A)^{-1} B R_{m+1}, \text{ and } \widetilde{W}_{m+1} = (\sigma_{m+1} I_n - A)^{-T} C^T L_{m+1}, \quad (4.1)$$

where the new interpolation point σ_{m+1}, μ_{m+1} and the new tangent direction R_{m+1}, L_{m+1} are computed as follows:

$$(R_{m+1}, \sigma_{m+1}) = arg \max_{\substack{\omega \in S_m \\ R \in \mathbb{R}^{p\times s} \\ \|R\|_2 = 1}} \|R_B(\omega) R\|_2, \quad (4.2)$$

$$(L_{m+1}, \mu_{m+1}) = arg \max_{\substack{\omega \in S_m \\ L \in \mathbb{R}^{p\times s} \\ \|L\|_2 = 1}} \|R_C(\omega) L\|_2. \quad (4.3)$$

Here, $S_m \subset \mathbb{C}^+$ is defined as the convex hull of $\{-\lambda_1, ..., -\lambda_m\}$ where $\{\lambda_i\}_{i=1}^m$ are the eigenvalues of the matrix A_m.
For solving the problem (4.2), we proceed as follows. First, we compute the next interpolation point, by computing the norm of $R_B(\omega)$ for each ω in S_m, i.e., we solve the following problem:

$$\sigma_{m+1} = \arg\max_{\omega \in S_m} \|R_B(\omega)\|_2. \quad (4.4)$$

Then, the tangent direction R_{m+1} is computed by evaluating (4.2) at $\omega = \sigma_{m+1}$,

$$R_{m+1} = \arg \max_{\substack{R \in \mathbb{R}^{p\times s} \\ \|R\|_2 = 1}} \|R_B(\sigma_{m+1}) R\|_2 . \quad (4.5)$$

We can easily prove that the tangent matrix direction R_{m+1} is given as follows:

$$R_{m+1} = [r_1^{(m+1)}, ..., r_s^{(m+1)}],$$

where the $r_i^{(m+1)}$'s are the right singular vectors corresponding to the s largest singular values of the matrix $R_B(\sigma_{m+1})$. This approach of maximizing the residual norm, works efficiently for small to medium matrices, but cannot be used for large-scale systems. To overcome this problem, we give the following proposition.

Proposition 4.1 *Let $R_B(\omega) = B - (\omega I_n - A)\mathbb{V}_m U_m^B(\omega)$ and $R_C(\omega) = C^T - (\omega I_n - A)^T \mathbb{W}_m U_m^C(\omega)$ be the residuals given in (1.6) and (1.7), where $U_m^B(\omega) = (\omega I -$*

$A_m)^{-1}\mathbb{W}_m^T B$ *and* $U_m^C(\omega) = (\omega I - A_m)^{-T}\mathbb{V}_m^T C^T$. *Then, we have the following new expressions as follows:*

$$R_B(\omega) = (\mathbb{V}_m\mathbb{W}_m^T - I_n)\mathbb{K}_m^B\mathbb{G}_m^{-1}U_m^B(\omega), \tag{4.6}$$

and

$$R_C(\omega) = (\mathbb{W}_m\mathbb{V}_m^T - I_n)\mathbb{K}_m^C\mathbb{Q}_m^{-1}U_m^C(\omega). \tag{4.7}$$

Proof The residual $R_B(\omega)$ can be written as follows:

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

Since $B \in \text{Range}\{V_1, ...V_m\}$, then $(I_n - \mathbb{V}_m\mathbb{W}_m^T)B = 0$. Using (3.19), we get the following:

$$A\mathbb{V}_m - \mathbb{V}_m A_m = -\mathbb{K}_m^B\mathbb{G}_m^{-1} + \mathbb{V}_m\mathbb{W}_m^T\mathbb{K}_m^B\mathbb{G}_m^{-1},$$

which proves (4.6). In the same way, we can prove (4.7). □

The expression of $R_B(\omega)$ given in (4.6) allows to reduce the computational cost while computing the next pole and direction. In fact, applying the skinny QR decomposition is as follows:

$$(\mathbb{V}_m\mathbb{W}_m^T - I_n)\mathbb{K}_m\mathbb{G}_m^{-1} = \text{QL},$$

we get the new expression of the residual norm as follows:

$$\|R_B(\omega)\|_2 = \left\|LU_m^B(\omega)\right\|_2. \tag{4.8}$$

This means that, solving the problem (4.2) requires only the computation of matrices of size $ms \times ms$ for each value of ω.

The next algorithm, summarizes all the steps of the adaptive choice of tangent interpolation points and tangent directions.

In order to save memory, Algorithm 2 allows us to compute the approximations $\mathcal{X}_m^{(1)} = Z_m^{(1)}(Z_m^{(1)})^T$ and $\mathcal{X}_m^{(2)} = Z_m^{(2)}(Z_m^{(2)})^T$ in a factored form, where,

$$Z_m^{(1)} = \mathbb{V}_m\widetilde{U}_1\Lambda_1^{\frac{1}{2}} \qquad Z_m^{(2)} = \mathbb{W}_m\widetilde{V}_1\Gamma_1^{\frac{1}{2}}. \tag{4.9}$$

The matrices $\widetilde{U}_1$, Λ_1, $\widetilde{V}_1$, and Γ_1 are obtained via the eigenvalue decomposition of the low-rank solutions $\mathcal{Y}_m^{(1)} = \widetilde{U}\Lambda\widetilde{U}^T$, $\mathcal{Y}_m^{(2)} = \widetilde{V}\Gamma\widetilde{V}^T$, and $\widetilde{U} = [\widetilde{U}_1 \ \ \widetilde{U}_2]$, $\widetilde{V} = [\widetilde{V}_1 \ \ \widetilde{V}_2]$ such that $\Lambda = \text{diag}(\Lambda_1, \Lambda_2)$, $\Gamma = \text{diag}(\Gamma_1, \Gamma_2)$ verify $\max(\text{diag}(\Lambda_1)) > \text{dtol}$ and $\max(\text{diag}(\Gamma_1)) > \text{dtol}$ for some given tolerance dtol.

In this section, we presented the ABTL algorithm inspiring from [8], where the adaptive approach was used for the selection of the interpolation points and tangent directions, as mentioned before. The main differences of the new method and the existing ones in [7, 8] are the following: first was the oblique projection onto Krylov

subspaces and constructing two bi-orthonormal subspaces from those latter, using the Lanczos procedure. Secondly, the tangent directions in our method are blocks of size $p \times s$ where $s \leq p$ and it is fixed, when in [8] the s is depending on the iteration m. Moreover, the residual norms used for stopping criterion (Algorithm 2, line 14) are simplified to new expressions given in 3.2, which allows for faster time execution and smaller memory occupancy, as reported in the numerical experiments.

Algorithm 2 The Adaptive Block Tangential Lanczos (ABTL) algorithm

- Given A, B, C, $m_{\max}$, ϵ.
- Outputs: $Z^{(1)}_{m_{\max}}$, $Z^{(2)}_{m_{\max}}$.
- Set $B = H_{1,0}V_1$ and $C^T = F_{1,0}W_1$ such that $W_1^T V_1 = I_p$.
- Initialize: $\mathbb{V}_1 = [V_1]$, $\mathbb{W}_1 = [W_1]$.

1. For $m = 1 : m_{\max}$
2. Set $A_m = \mathbb{W}_m^T A \mathbb{V}_m$, $B_m = \mathbb{W}_m^T B$, $C_m = C\mathbb{V}_m$.
3. Compute σ_m, and μ_m
   - Compute $\{\lambda_1, ..., \lambda_m\}$ eigenvalues of A_m.
   - Determine S_m, convex hull of $\{-\lambda_1, ..., -\lambda_m\}$.
   - Solve (4.4). The same for μ_m.
4. Compute right and left vectors R_m, L_m.
5. $\widetilde{V}_m = (A - \sigma_m I_n)^{-1} B R_m, \quad \widetilde{W}_m = (A - \mu_m I_n)^{-T} C^T L_m.$
6. For i = 1,...,m
   - $H_{i,m} = W_i^T \widetilde{V}_{m+1}$, $\quad - F_{i,m} = V_i^T \widetilde{W}_{m+1}$,
   - $\widetilde{V}_{m+1} = \widetilde{V}_{m+1} - V_i H_{i,m}$, $\quad - \widetilde{W}_{m+1} = \widetilde{W}_{m+1} - W_i F_{i,m}$,
7. End.
8. $\widetilde{V}_{m+1} = V_{m+1} H_{m+1,m}$, $\widetilde{W}_{m+1} = W_{m+1} F_{m+1,m}$. (QR decomposition).
9. $W_{m+1}^T V_{m+1} = P_m D_m Q_m^T$. (singular value decomposition).
10. $V_{m+1} = V_{m+1} Q_m D_m^{-1/2}$, $W_{m+1} = W_{m+1} P_m D_m^{-1/2}$.
11. $H_{m+1,m} = D_m^{1/2} Q_m^T H_{m+1,m}$, $F_{m+1,m} = D_m^{1/2} P_m^T F_{m+1,m}$.
12. $\mathbb{V}_{m+1} = \left[\mathbb{V}_m, V_{m+1}\right]$, $\mathbb{W}_{m+1} = \left[\mathbb{W}_m, W_{m+1}\right]$.
13. Solve (3.10) and (3.11) to get $\mathcal{Y}_m^{(1)}$ and $\mathcal{Y}_m^{(2)}$.
14. If $\max(\| \mathcal{R}_1(\mathcal{X}_m^{(1)}) \|_2, \| \mathcal{R}_2(\mathcal{X}_m^{(2)}) \|_2) < \epsilon$ Stop.
15. End.

- Compute $Z^{(1)}_{m_{\max}}$, $Z^{(2)}_{m_{\max}}$ as in (4.9).

4.1 Adaptive Block Tangential Arnoldi (ABTA) algorithm

Consider the following Lyapunov equation as follows:

$$AX + XA^T + BB^T = 0. \tag{4.10}$$

The solution X is approximated by $\mathcal{X}_m$ such as the following:

$$\mathcal{X}_m = \mathbb{V}_m \mathcal{Y}_m \mathbb{V}_m^T, \tag{4.11}$$

and satisfying the Galerkin condition as follows:

$$\mathbb{V}_m^T \mathcal{R}(\mathcal{X}_m) \mathbb{V}_m = 0, \tag{4.12}$$

where the residual is given by $\mathcal{R}(\mathcal{X}_m) = A\mathcal{X}_m + \mathcal{X}_m A^T + BB^T$ and $\mathbb{V}_m = [V_1, V_2, \ldots, V_m]$ is a matrix obtained from the orthonormal basis $\mathcal{V}_m = \text{Range}\{V_1, V_2, \ldots, V_m\}$ constructed from the following tangential subspace $\text{Range}\{B, (\sigma_1 I_n - A)^{-1} B R_1, ..., (\sigma_m I_n - A)^{-1} B R_m\}$. From (4.11) and (4.12) , $\mathcal{Y}_m$ is obtained by solving the low-dimensional Lyapunov matrix equation as follows:

$$A_m \mathcal{Y}_m + \mathcal{Y}_m A_m^T + B_m B_m^T = 0,$$

where $A_m = \mathbb{V}_m^T A \mathbb{V}_m$, $B_m = \mathbb{V}_m^T B$. Notice that, we consider here only one tangential subspace. All the results obtained in the previous section can be adapted for the adaptive block Arnoldi method. For the computation of the residual norms, we have the following result.

Theorem 4.1 *Let $\mathbb{V}_m = [V_1, ..., V_m]$ obtained from BTAA. Let $\mathcal{X}_m = \mathbb{V}_m \mathcal{Y}_m \mathbb{V}_m^T$, be the approximate solution of the Lyapunov matrix equation, then the residual norm is given as follows:*

$$\| \mathcal{R}(\mathcal{X}_m) \|_2 = \| \mathbb{S}_m J \mathbb{S}_m \|_2, \tag{4.13}$$

where $\mathbb{S}_m$ is an upper triangular matrix obtained from the skinny QR decomposition of the matrix as follows:

$$\mathbb{U}_m = \begin{bmatrix} \mathbb{V}_m \mathcal{Y}_m \mathbb{G}_m^{-T} & (I_n - \mathbb{V}_m \mathbb{V}_m^T) \mathbb{K}_m^B \end{bmatrix}.$$

The matrix J is defined as $J = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$.

Proof The proof is similar to the one given in the proof of Theorem 3.2. □

The choice of the interpolation points and tangent directions is the same as in the previous section, as shown in the following:

$$(R_{m+1}, \sigma_{m+1}) = \arg \max_{\substack{\omega \in S_m \\ R \in \mathbb{R}^{p \times s} \\ \|R\|_2 = 1}} \| R_B(\omega) R \|_2. \tag{4.14}$$

where $R_B(\omega) = B - (\omega I_n - A) V_m (\omega I_m - A_m)^{-1} B_m$. The algorithm will be named Adaptive Block Tangential Arnoldi (ABTA) and is summarized as follows.

Algorithm 3 The Adaptive Block Tangential Arnoldi (ABTA) algorithm

- Inputs A, B, $m_{\max}$, ϵ and dtol.
- Set $B = H_{1,0}V_1$ and initialize: $\mathbb{V}_1 = [V_1]$.

  1. For $m = 1 : m_{\max}$
     - Set $A_m = \mathbb{V}_m^T A \mathbb{V}_m$, $B_m = \mathbb{V}_m^T B$.
     - Compute the interpolation points σ_m and the directions R_m.
     - $\widetilde{V}_m = (A - \sigma_m I_n)^{-1} B R_m$.
     - For i = 1,...,m
       - $H_{i,m} = V_i^T \widetilde{V}_{m+1}$,
       - $\widetilde{V}_{m+1} = \widetilde{V}_{m+1} - V_i H_{i,m}$,
     - End.
     - $\widetilde{V}_{m+1} = V_{m+1} H_{m+1,m}$. (QR decomposition).
     - $\mathbb{V}_{m+1} = \left[\mathbb{V}_m,\ V_{m+1}\right]$.
     - Compute the approximate $\mathcal{Y}_m$.
     - If $\| \mathcal{R}(\mathcal{X}_m) \|_2 < \epsilon$, stop.
     - End.

- Compute $Z_{m_{\max}}$.

5 Numerical experiments

In this section, we present some numerical examples to show the effectiveness of the adaptive block tangential Arnoldi and Lanczos-types algorithms (ABTA and ABTL). All the experiments 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 [18]:

- lp_lgfrq: Generates a set of logarithmically distributed frequency sampling points
- lp_gnorm: Computes $\|H(j\omega) - H_m(j\omega)\|_2$

Example 1 In this first experiment, we used the rail3113 model ($n = 3113, p = 6$). The model describes the steel rail cooling problem and is from the Oberwolfach collection.[1] Figures 1 and 2 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 $\omega \in [10^{-6},\ 10^{6}]$ for both methods ABTL (left) and ABTA (right). The dimension of the reduced model $m = 20$.

In this part, we compared the ABTA and ABTL algorithms with (RKSM) that solves a large-scale Lyapunov matrix equation by means of the adaptive Rational Krylov method with Galerkin condition; for more, see [8] and (TRKSM)

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

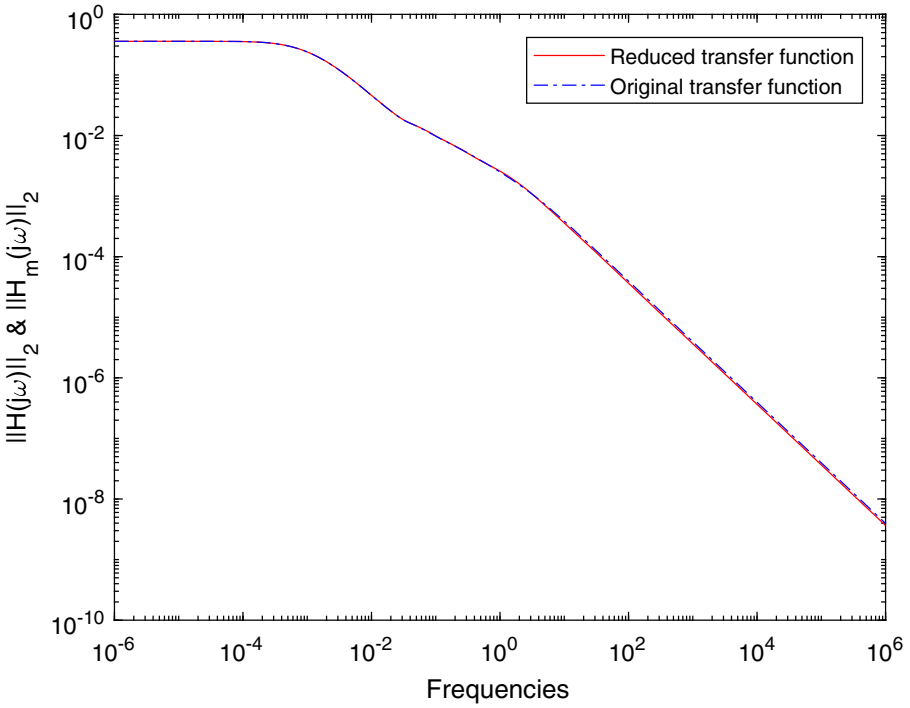

Fig. 1 The RAIL3113 model: ABTL

where the tangential approach is used [7]. The Matlab implementations of (RKSM) and (TRKSM) have been downloaded from Simoncini's web page.[2] We used the FDM model, where the matrix A is obtained from the centered finite difference discretization of the operator as follows:

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

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

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

Different choices of columns for B and C were performed. The number of inner grid points in each direction is n_0 and the dimension of A is $n = n_0^2$, various value of n_0 are used.

Example 2 Plots in Figs. 3 and 4 represent the exact error $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-6},\ 10^{6}]$ of the four methods, the ABTA method (solid

[2] http://www.dm.unibo.it/~simoncin/software.html

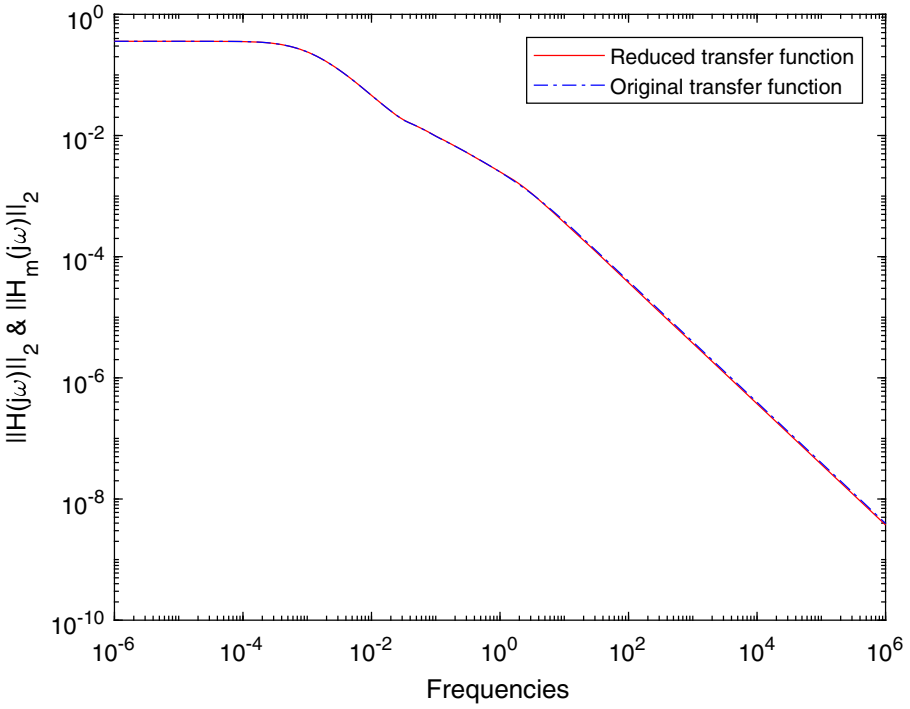

Fig. 2 The RAIL3113 model: ABTA.

line), the ABTL method (dashed-dotted line), the RKSM (dashed-dashed line), and TRKSM (dotted line). The matrices B and C were random, the stopping tolerance for the Frobenius norm of the Lyapunov equation residual was set to 10^{-8}. Plots in Fig. 4 represent the same thing but with the matrix $C = B^T$.

We present below Table 1 that gives more information about the plots in Figs. 3 and 4, where we present the execution time, the maximum space dimension, the rank dimension, and the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ error norms obtained by each method. The maximum space dimension is the dimension of the matrices obtained after the stopping tolerance tol $= 10^{-8}$ and the rank dimension is dimension obtained as in (4.9).

Example 3 In this example we compared, the Rank dimension (Fig. 5), the Err-$\mathcal{H}_\infty$ norm (Fig. 6) and the execution time (Fig. 7) as p the rank of the matrix B is grown. In all plots of Figs. 5, 6, and 7, the ABTL method (solid line), the ABTA method (dashed-dotted line), the RKSM (dashed-dashed line), and TRKSM (black dashed-dotted line) were seen. We used the FDM model of size $n = 10000$ and the matrix $B = A^{-1}I_p$ with p ranging from 4 to 24.

Figure 5 shows that the four methods have the same rank dimension and is growing with as p grows. In Fig. 6, we notice that our both methods and RKSM give better $\mathcal{H}_\infty$-err norm than TRKSM method, but in Fig. 7, we can see clearly that TRKSM have the best execution time, followed by our both methods, and finally, RKSM

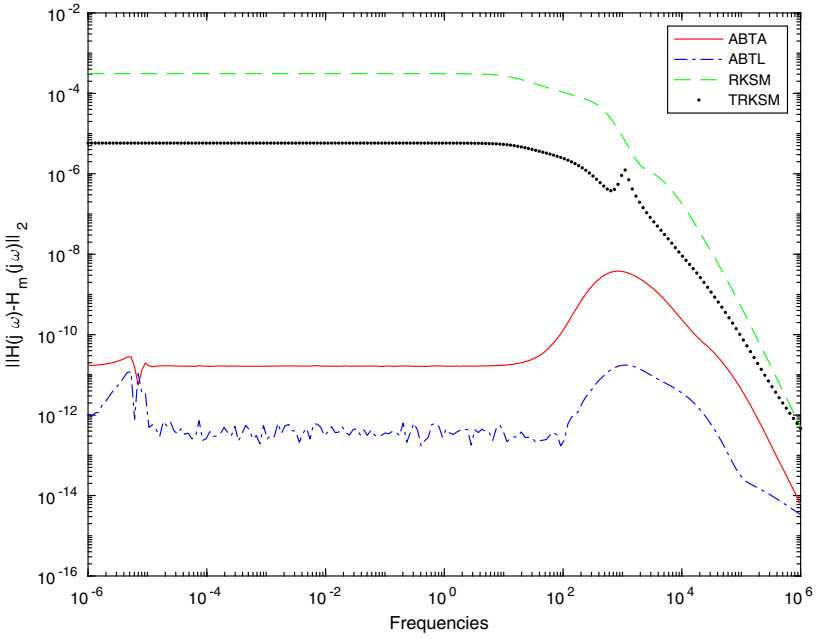

Fig. 3 $B = \text{rand}(n,p)$, $C = \text{rand}(p,n)$

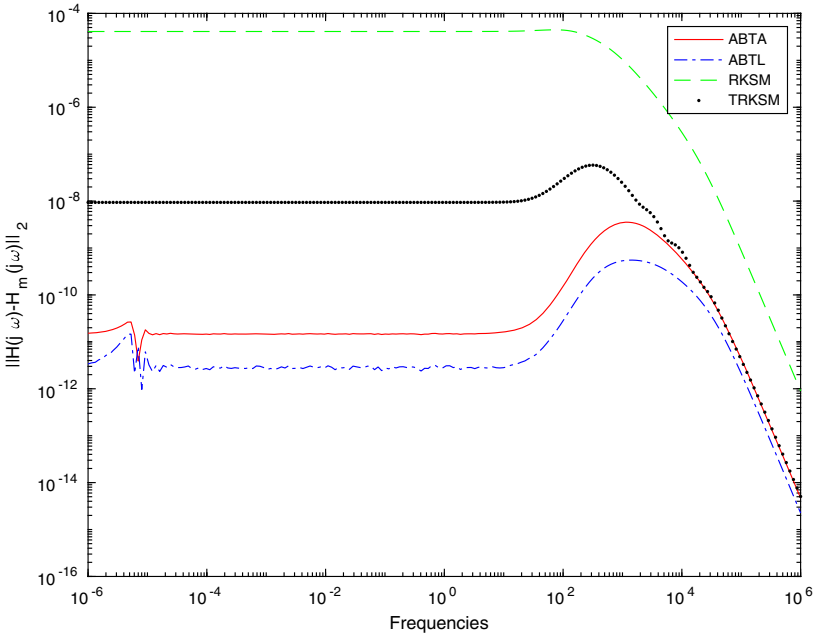

Fig. 4 $B = \text{rand}(n,p)$, $C = B^T$

Table 1 The calculation time and dimension of convergence

$n = 4900$	$p = 6$	Time	S. dim	Rank	Err-$\mathcal{H}_\infty$	Err-$\mathcal{H}_2$
	ABTA	3.90 s	96	96	2.5×10^{-9}	5.7×10^{-9}
$B = \text{rand}(n, p)$	ABTL	5.67 s	126	126	1.2×10^{-11}	2.5×10^{-11}
$C = \text{rand}(p, n)$	RKSM	73.28 s	66	60	3.1×10^{-4}	3.4×10^{-3}
	TRKSM	2.59 s	88	88	5.7×10^{-6}	6.4×10^{-5}
	ABTA	3.11 s	90	90	3.7×10^{-9}	1.3×10^{-8}
$B = \text{rand}(n, p)$	ABTL	3.12 s	96	96	5.5×10^{-10}	2.2×10^{-9}
$C = B^T$	RKSM	76.37 s	60	66	4.4×10^{-5}	4.9×10^{-4}
	TRKSM	2.56 s	89	89	5.8×10^{-8}	2.3×10^{-7}

method with very bad execution time as shown above. In short, we can say that, our both methods have the performance of the RKSM method with a execution time near to that of the TRSKM method.

Example 4 For this experiment, we used the FLOW matrix of size $n = 9669$, from the Oberwolfach collection, we compared the four methods, the results are reported

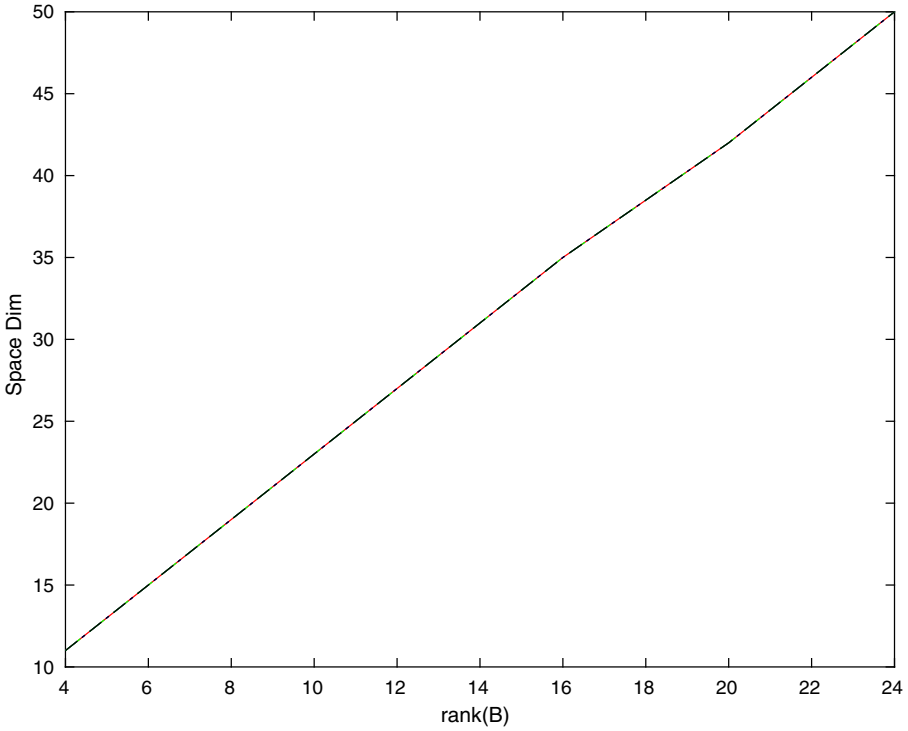

Fig. 5 FDM model: the rank dimension

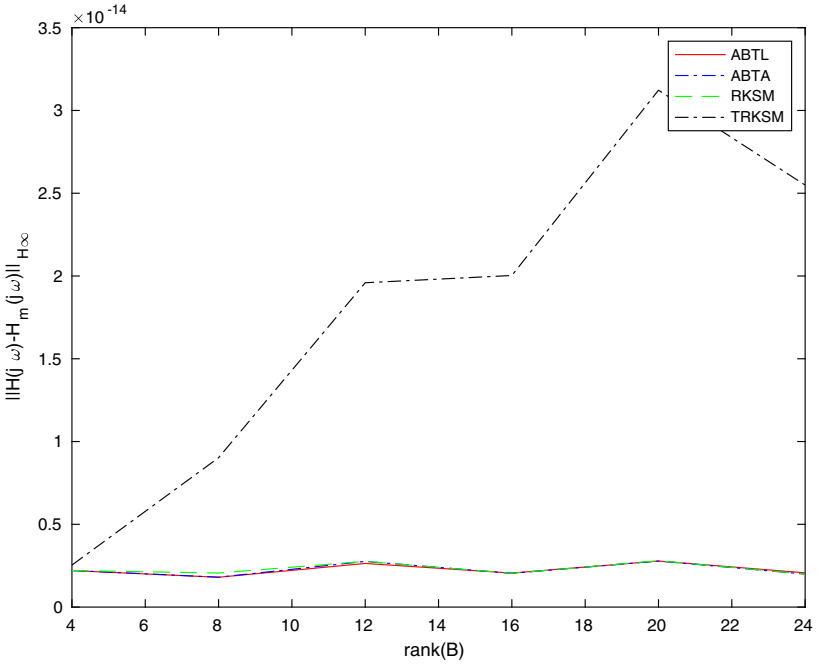

Fig. 6 FDM model: the $\mathcal{H}_\infty$-err norm

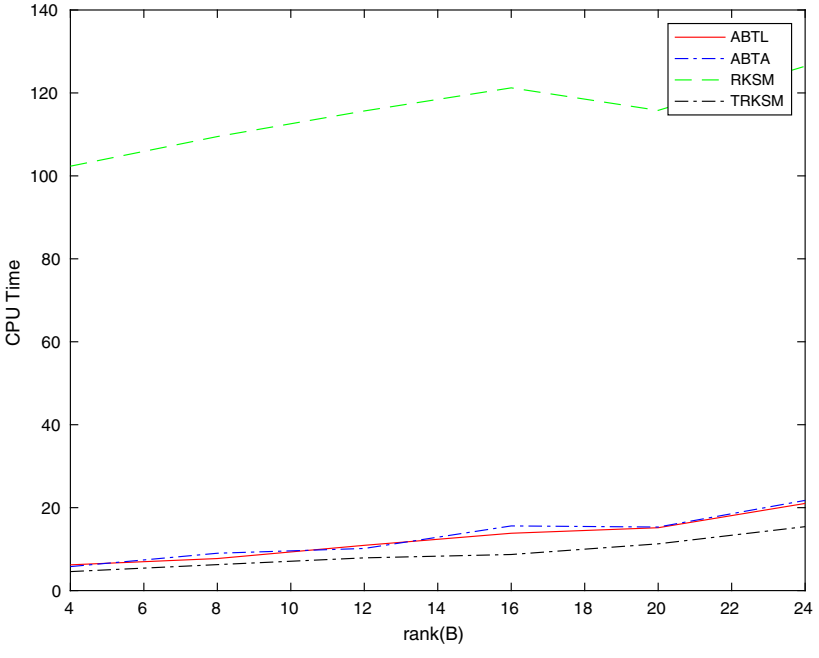

Fig. 7 FDM model: the execution time

Table 2 The execution time and dimension of convergence

FLOW	$p = 6, s = p$	Time	S. dim	Rank	Err-$\mathcal{H}_\infty$	Err-$\mathcal{H}_2$
	ABTA	13.94 s	132	132	3.07×10^{-5}	1.22×10^{-4}
$B = \text{rand}(n, p)$	ABTL	13.01 s	132	132	3.06×10^{-5}	1.22×10^{-4}
$C = \text{rand}(p, n)$	RKSM	20.43 s	12	8	$6.82\times10^{+5}$	$4.88\times10^{+6}$
	TRKSM	8.85 s	126	126	6.54×10^{-5}	2.54×10^{-4}
	ABTA	10.00 s	84	84	1.21×10^{-5}	7.42×10^{-5}
$B = \text{rand}(n, p)$	ABTL	8.73 s	84	84	1.21×10^{-5}	7.42×10^{-5}
$C{=}B^T$	RKSM	33.65 s	12	8	$4.08\times10^{+4}$	$1.99\times10^{+5}$
	TRKSM	6.38 s	73	73	2.05×10^{-4}	1.16×10^{-3}

in Table 2 and implemented in Figs. 8 and 9. We notice that for the RKSM method, the space dimension is so small compared to the other methods, because RKSM achieved the stopping tolerance after two iterations, which gives bad results as shown in the plots and the table and also still have the longest execution time.

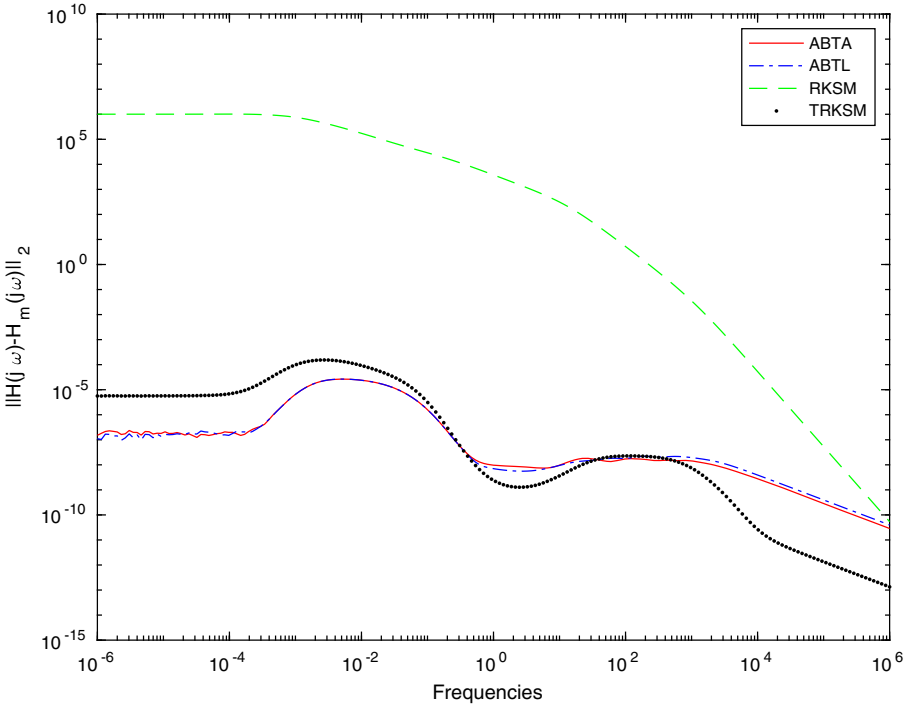

Fig. 8 $B = \text{rand}(n, p)$, $C = B^T$

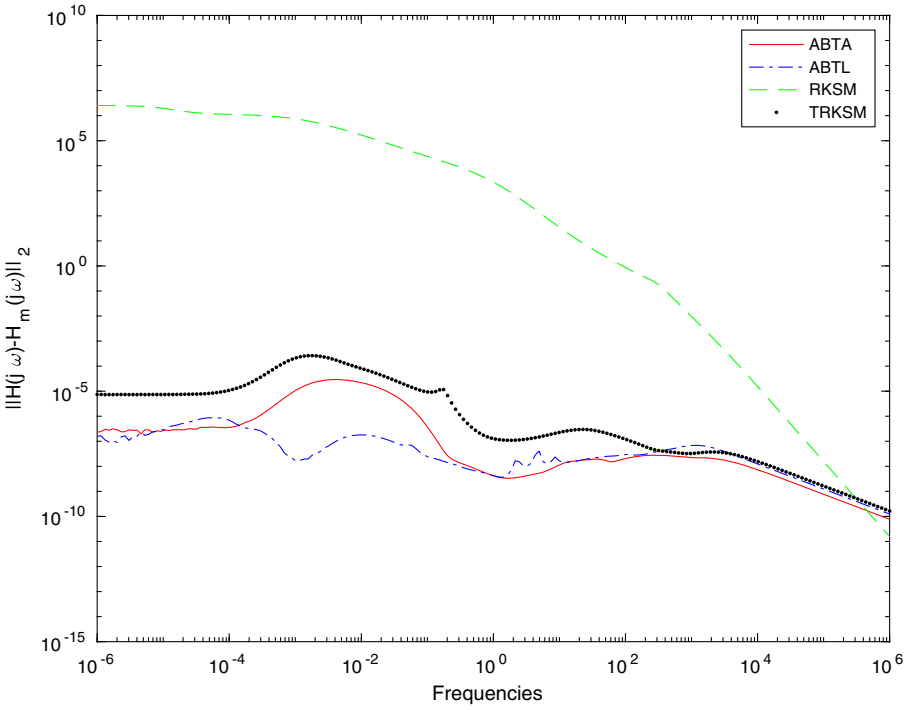

Fig. 9 $B = \text{rand}(n, p)$, $C = \text{rand}(p, n)$

Example 5 In this experiment, we used the rail20209 and rail79841 models with a fixed $m = 20$. These models describe the steel rail cooling problem and are also from the Oberwolfach collection. The plots below represent the exact error $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies of the tree methods ABTA (solid line), the ABTL (dashed-dotted line), and TRKSM (dashed-dashed line), the stopping tolerance was set to 10^{-6}.

Figure 10 represents the rail20209 model ($n = 20209$, $p = 6$), we notice that the tree methods coincide, with an execution time almost the same (TRKSM, 9.26 s; ABTL, 9.76 s; ABTA, 10.45 s). Figure 11 represents the rail79841 model ($n = 79841$, $p = 6$), the matrices B and C were random. The execution time for this example is as follows: (ABTL, 85.99 ; ABTA, 99.45 ; TRKSM, 119.76 s).

Example 6 In this last experiment, we present the Table 3 below, that contains the execution time, space dimension, rank dimension, the $\mathcal{H}_\infty$ and $\mathcal{H}_2$-err norms of the FDM model with a large dimension ($n = 122500$ and $n = 90000$), the stopping tolerance was set to 10^{-6}. We notice that for this large-scales system, our methods are faster and give better error norms.

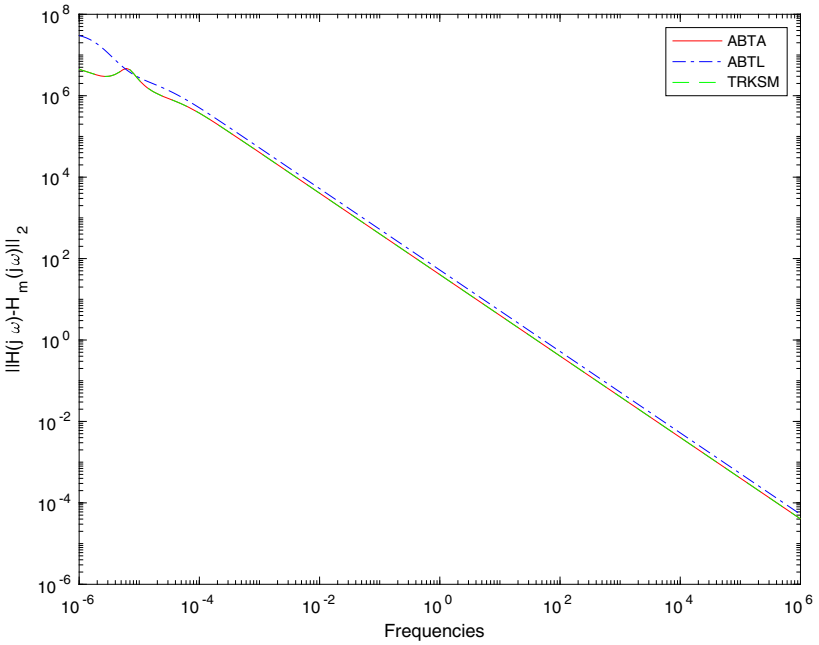

Fig. 10 The RAIL20209 model

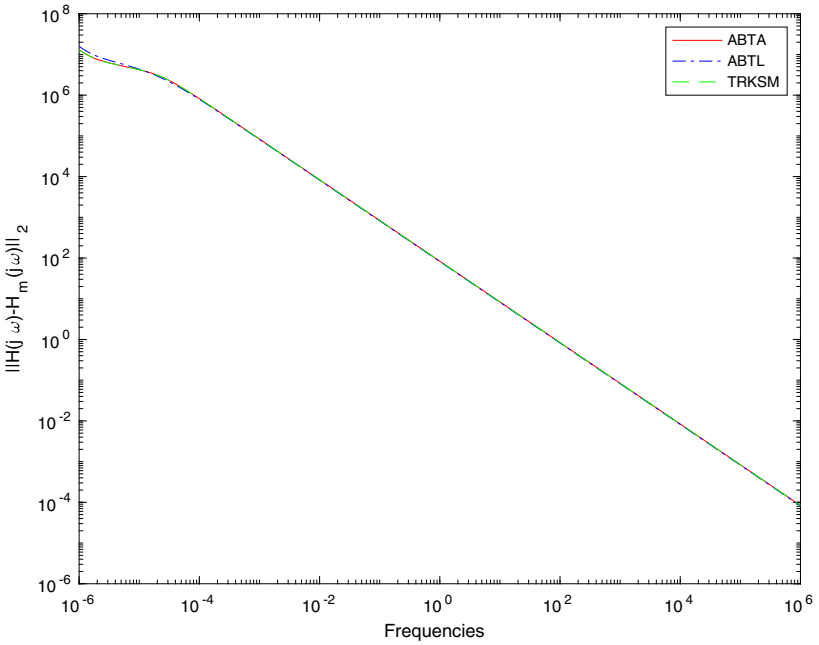

Fig. 11 The RAIL79841 model

Table 3 The execution time and dimension of convergence

$n = 122500$	$p = 6, s = 6$	Time	S. dim	Rank	Err-$\mathcal{H}_\infty$	Err-$\mathcal{H}_2$
	ABTA	151.57 s	108	108	7.64×10^{-7}	3.28×10^{-6}
$B = \text{rand}(n, p)$	ABTL	84.58 s	102	102	1.53×10^{-6}	7.36×10^{-6}
	TRKSM	596.04 s	92	92	2.53×10^{-4}	2.81×10^{-3}
$n = 90000$	$p = 6, s = 3$	Time	S. dim	Rank	Err-$\mathcal{H}_\infty$	Err-$\mathcal{H}_2$
	ABTA	108.13 s	66	3	2.21×10^{-15}	2.53×10^{-14}
$B = A^{-1} I_p$	ABTL	64.82 s	72	3	2.22×10^{-15}	2.53×10^{-14}
	TRKSM	150.95 s	60	3	2.03×10^{-15}	2.33×10^{-14}

6 Conclusion

In the present paper, we proposed a new approach based on block tangential Krylov subspaces to compute low-rank approximate solutions to large Lyapunov equations. These approximate solutions are given in factored forms and are used to build reduced-order models that approximate the initial large-scale dynamical systems with multiple inputs and multiple outputs (MIMO). The method constructs sequences of orthogonal blocks from matrix tangential Krylov subspaces using the block Lanczos-type or Arnoldi-type approaches. We construct approximate Gramians which are used in the balanced truncation method. The interpolation shifts and the tangential directions are selected in an adaptive way by maximizing the residual norms. We presented some new algebraic properties and gave some numerical experiments on some benchmark examples showing that the proposed methods return good results, as compared to some well-knowing methods, for large problems.

References

1. Antoulas, A.C.: Approximation of large-scale dynamical systems. SIAM Adv. Des. Contr. **29**, 181–190 (2005)
2. Bai, Z.: Krylov subspace techniques for reduced-order modeling of large scale dynamical systems. Appl. Numer. Math. **43**, 9–44 (2002)
3. Benner, P., Li, J., Penzl, T.: Numerical solution of large Lyapunov equations, Riccati equations and linear-quadratic optimal control problems. Numer. Linear Algebra Appl. **15**, 755–777 (2008)
4. Bentbib, A.H., Jbilou, K., Kaouane, Y.: A computational global tangential Krylov subspace method for model reduction of large-scale MIMO dynamical systems. J. Sci. Comput. **75**, 1614–1632 (2018)
5. Datta, B.N.: Large-scale matrix computations in control. Appl. Numer. Math. **30**, 53–63 (1999)
6. Datta, B.N.: Krylov subspace methods for large-scale matrix problems in control. Future Gener. Comput. Syst. **19**, 1253–1263 (2003)
7. Druskin, V., Simoncini, V., Zaslavsky, M.: Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems. SIAM J. Matrix Anal. Appl. **35**, 476–498 (2014)
8. Druskin, V., Simoncini, V.: Adaptive rational Krylov subspaces for large-scale dynamical systems. Syst. Contr. Lett. **60**, 546–560 (2011)
9. Frangos, M., Jaimoukha, I.M.: Adaptive rational interpolation: Arnoldi and Lanczos-like equations. Eur. J. Control. **14**, 342–354 (2008)
10. Glover, K.: All optimal Hankel-norm approximations of linear multivariable systems and their L, ∞-error bounds. Int. J. Control. **39**, 1115–1193 (1984)

11. Glover, K., Limebeer, D.J.N., Doyle, J.C., Kasenally, E.M., Safonov, M.G.: A characterization of all solutions to the four block general distance problem. SIAM J. Control Optim. **29**, 283–324 (1991)
12. Grimme, E.: Krylov projection methods for model reduction. Ph.D. thesis, Coordinated Science Laboratory University of Illinois at Urbana–Champaign (1997)
13. Gugercin, S., Antoulas, A.C.: A survey of model reduction by balanced truncation and some new results. Int. J. Control. **77**, 748–766 (2004)
14. Heyouni, M., Jbilou, K.: Matrix Krylov subspace methods for large scale model reduction problems. App. Math. Comput. **181**, 1215–1228 (2006)
15. Jaimoukha, I.M., Kasenally, E.M.: Krylov subspace methods for solving large Lyapunov equations. SIAM J. Matrix Anal. Appl. **31**, 227–251 (1994)
16. Moore, B.C.: Principal component analysis in linear systems: controllability, observability and model reduction. IEEE Trans. Automatic Contr. **26**, 17–32 (1981)
17. Mullis, C.T., Roberts, R.A.: Roundoff noise in digital filters: frequency transformations and invariants. IEEE Trans. Acoust. Speec Signal Process. **24**, 538–550 (1976)
18. Penzl, T.: Lyapack matlab toolbox for large Lyapunov and Riccati equations, model reduction problems and linear-quadratic optimal control problems, http://www.tuchemintz.de/sfb393/lyapack
19. Safonov, M.G., Chiang, R.Y.: A Schur method for balanced-truncation model reduction. IEEE Trans. Automat. Contr. **34**, 729–733 (1989)
20. Van Dooren, P., Gallivan, K.A., Absil, P.: H_2-optimal model reduction with higher order poles. SIAM J. Matrix Anal. Appl. **31**, 2738–2753 (2010)