Tangential methods for model reductions and applications

Yassine Kaouane

ULCO
UCA

École doctorale **SI & ED SPI**

Unité de recherche **LAMAI & LMPA**

Thèse présentée par **KAOUANE Yassine**

Soutenue le **31 décembre 2018**

En vue de l'obtention du grade de docteur de l'ULCO et de l'UCA

Discipline **Mathématiques**

Spécialité **Systèmes dynamiques**

Titre de la thèse

Méthodes tangentielles pour les réductions de modèles et applications

Thèse dirigée par	Abdeslem Hafid Bentbib	directeur
	Khalide Jbilou	co-directeur

Composition du jury

Rapporteurs	El Alaoui Talibi Mohamed	professeur à l'Université Cadi Ayyad Marrakech	
	El Bernoussi Souad	professeur à l'Université de Rabat	
	Guedda Mohammed	professeur à l'Université de Picardie Jules Verne	
Examinateur	Sadok Hassane	professeur à l'Université du Littoral Côte d'Opale	président du jury
Directeurs de thèse	Abdeslem Hafid Bentbib	professeur à l'UCA	
	Khalide Jbilou	professeur à l'ULCO	

ULCO
UCA

École doctorale **SI & ED SPI**

Unité de recherche **LAMAI & LMPA**

Thèse présentée par **KAOUANE Yassine**

Soutenue le **31 décembre 2018**

En vue de l'obtention du grade de docteur de l'ULCO et de l'UCA

Discipline **Mathématiques**

Spécialité **Systèmes dynamiques**

Titre de la thèse

Méthodes tangentielles pour les réductions de modèles et applications

Thèse dirigée par	Abdeslem Hafid Bentbib	directeur
	Khalide Jbilou	co-directeur

Composition du jury

Rapporteurs	El Alaoui Talibi Mohamed	professeur à l'Université Cadi Ayyad Marrakech	
	El Bernoussi Souad	professeur à l'Université de Rabat	
	Guedda Mohammed	professeur à l'Université de Picardie Jules Verne	
Examinateur	Sadok Hassane	professeur à l'Université du Littoral Côte d'Opale	président du jury
Directeurs de thèse	Abdeslem Hafid Bentbib	professeur à l'UCA	
	Khalide Jbilou	professeur à l'ULCO	

ULCO
UCA

Doctoral School **SI & ED SPI**

University Department **LAMAI & LMPA**

Thesis defended by **KAOUANE Yassine**

Defended on **31st December, 2018**

In order to become Doctor from ULCO and from UCA

Academic Field **Mathematics**

Speciality **Dynamical systems**

Thesis Title

Tangential methods for model reductions and applications

Thesis supervised by	Abdeslem Hafid Bentbib	Supervisor
	Khalide Jbilou	Co-Supervisor

Committee members

Referees	El Alaoui Talibi Mohamed	Professor at Université Cadi Ayyad Marrakech	
	El Bernoussi Souad	Professor at Université de Rabat	
	Guedda Mohammed	Professor at Université de Picardie Jules Verne	
Examiner	Sadok Hassane	Professor at Université du Littoral Côte d'Opale	Committee President
Supervisors	Abdeslem Hafid Bentbib	Professor at UCA	
	Khalide Jbilou	Professor at ULCO	

Je dédie ce travail
à tous ceux qui le méritent

À mon directeur bien-aimé !

À mon co-directeur bien-co-aimé !

Méthodes tangentielles pour les réductions de modèles et applications

Résumé

Les simulations à grande dimension jouent un rôle crucial dans l'étude d'une grande variété de phénomènes physiques complexes, entraînant souvent des demandes écrasantes sur les ressources informatiques. La gestion de ces demandes constitue la principale motivation pour la réduction du modèle : produire des modèles de commande réduite plus simples, qui permettent une simulation plus rapide et moins coûteuse tout en se rapprochant avec précision du comportement du modèle d'origine. La présence des systèmes avec multiple entrées et multiple sorties (MIMO) rend le processus de réduction encore plus difficile. Dans cette thèse, nous nous intéressons aux méthodes de réduction de modèles à grande dimension en utilisant la projection sur des sous-espaces de Krylov tangentielles. Nous nous penchons sur le développement de techniques qui utilisent l'interpolation tangentielle. Celles-ci présentent une alternative efficace et intéressante à la troncature équilibrée qui est considérée comme référence dans le domaine et tout particulièrement la réduction pour les systèmes linéaire à temps invariants. Enfin, une attention particulière sera portée sur l'élaboration de nouveaux algorithmes efficaces et sur l'application à des problèmes pratiques.

Mots clés : réduction de modèle, interpolation, sous-espace de krylov

Tangential methods for model reductions and applications

Abstract

Large-scale simulations play a crucial role in the study of a great variety of complex physical phenomena, leading often to overwhelming demands on computational resources. Managing these demands constitutes the main motivation for model reduction: produce simpler reduced-order models, which allow for faster and cheaper simulation while accurately approximating the behaviour of the original model. The presence of multiple inputs and outputs (MIMO) systems, makes the reduction process even more challenging. In this thesis we are interested in methods of reducing large-scale models, using projection on tangential Krylov subspaces. We are looking at the development of techniques using tangential interpolation. These present an effective and interesting alternative to the balanced truncation which is considered as a reference in the field and especially for the reduction of linear time invariant systems. Finally, special attention will be focused on the development of new efficient algorithms and application to practical problems.

Keywords: model reduction, interpolation, krylov subspace

LAMAI & LMPA
Maison de la Recherche Blaise Pascal – 50, rue Ferdinand Buisson – CS 80699 – 62228 Calais Cedex – France

Remerciements

Il me sera très difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme.

Je voudrais tout d'abord remercier grandement Monsieur Abdeslem Hafid Bentbib, Professeur à l'Université Cadi Ayyad et Monsieur Khalide Jbilou, Professeur à l'Université du Littoral Côte d'Opale, qui m'ont encadré tout au long de cette thèse et qui m'ont fait partager leurs brillantes intuitions. Qu'ils soient aussi remercié pour leur gentillesse, leur disponibilité permanente et pour les nombreux encouragements qu'il m'ont prodiguée.

J'adresse tous mes remerciements à Monsieur Mohamed El Alaoui Talibi, Professeur à l'Université Cadi Ayyad Marrakech, ainsi qu'à Madame Souad El Bernoussi, Professeur à l'Université de Rabat, et à Monsieur Mohammed Guedda, Professeur à l'Université Picardie Jules Verne, de l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse.

Je tiens aussi à remercier Monsieur Hassane Sadok, Président et Professeur à l'Université du Littoral Côte d'Opale, qui m'a accueilli pendant deux ans au sein de son laboratoire. C'est grâce à lui que j'ai pu concilier avec bonheur recherche théorique et appliquée pendant cette thèse.

Merci à mes collègues. les étudiants du département qui ont aidé à faire ces trois ans une période très agréable à travers les nombreuses pauses café et autres Activités.

Enfin, je tiens à remercier tous les membres de ma famille, qui ont tout fait pour m'aider, qui m'ont soutenu et surtout supporté dans tout ce que j'ai entrepris.

Liste des symboles

Mathematical symbols

$< . >_F$ Frobenius product

$\mathbb{C}$ Set of complex numbers

$\mathbb{C}^+$ Right half plane, $z \in \{\mathbb{C}^+, Real(z) > 0\}$

$\mathbb{C}^-$ Left half plane, $z \in \{\mathbb{C}^-, Real(z) < 0\}$

$\mathbb{R}$ Set of real numbers

$\mathbb{R}^{n \times m}$ Set of real $n \times m$ matrices

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

$\otimes$ Kronecker product

$\| . \|$ Euclidean vector norm

$\| . \|_{\mathcal{H}_2}$ $\mathcal{H}_2$ norm of a dynamical system

$\| . \|_{\mathcal{H}_\infty}$ $\mathcal{H}_\infty$ norm of a dynamical system

$\sigma_{max}(A)$ Largest singular value of matrix A

A^T Transpose of matrix A

A^{-1} Inverse of matrix A

e_i i-th unit vector

I_n Identity matrix of dimension $n \times n$

j $\sqrt{-1}$

$Rank(A)$ Rank of matrix A

$Re(z)$ Real part of z

$X \perp_F Y$ $< X, Y >_F = 0$

General Notation

$\mathbb{K}_m(.,.)$ The block Krylov subspace

$\mathbb{K}_m(.,.)^R$ The block rational Krylov subspace

$\mathbb{K}_m(.,.)^T$ The block tangential Krylov subspace

$\mathcal{K}_m(.,.)$ The Krylov subspace

$\mathcal{K}_m(.,.)^R$ The rational Krylov subspace

$\mathcal{K}_m(.,.)^T$ The tangential Krylov subspace

Σ n^{th} order original dynamical system

A, B, C State space matrices of the original MIMO state space system

n Order of the original state space model

$u(t)$ The input vector of the state space system

$x(t)$ The state variable

$y(t)$ The output vector of the state space system

Abbreviations and acronyms

ABTAA Adaptive Block Tangential Arnoldi Algorithm

ABTAA Adaptive Block Tangential Arnoldi-type Algorithm

ABTL Adaptive Block Tangential Lanczos

AGTAA Adaptive global tangential Arnoldi-type algorithm

ATLA Adaptive Tangential Lanczos-type Algorithm

IRKA Iterative Rational Krylov Method

ISTIA Iterative SVD Tangential Interpolation Algorithm

ITIA Iterative Tangential Interpolation Algorithm

LTI Linear Time Invariant

MIMO Multiple input Multiple Output

SISO Single input Single Output

Sommaire

Table des figures

Contributions de la thèse

L'objectif de cette thèse est la réduction des systèmes dynamiques linéaires à grande dimension, avec multiple-entrées et multiple-sorties, en se basant sur la projection sur les sous-espaces de Krylov tangentiels. La thèse peut être divisée en quatre parties :

Partie I On présente une introduction générale sur les différents objectifs autour de la réduction des modèles en décrivant les différents exemples des systèmes dynamiques utilisés dans cette thèse.

Partie II Compsée de deux chapitres :

- **Chapitre**2 : Ce chapitre est consacré au rappel de quelques outils mathématiques utilisés dans la théorie des systèmes dynamiques linéaires invariants dans le temps (LTI).
- **Chapitre**3 : Dans ce chapitre, on présente des techniques connues dans la littérature pour la réduction des modèles. La première est la méthode de la troncature équilibrée basée sur l'utilisation des Grammians et sur la méthode de la décomposition en valeurs singuliers (SVD). La deuxième est la méthode de l'interpolation qui est basée sur des sous-espaces de Krylov. La dernière technique combine les deux précédentes méthodes pour profiter des avantages des deux.

Partie III On s'intéresse ici aux méthodes basées sur l'orthogonalité de type Arnoldi. Cette partie se décompose en deux chapitres

- **Chapitre**4 : Dans ce chapitre, on propose une nouvelle technique pour la réduction de modèles, en utilisant les méthodes de type Arnoldi, pour

construire une base orthonormale, à partir d'un sous-espace de Krylov tangentiel. Ensuite on introduit des propriétés algébriques qui décrivent la relation entre la base construite et la matrice du système d'origine. Ces équations algébriques nous permettent aussi d'établir une nouvelle expression de l'erreur. Finalement, des exemples numériques ont été donnés pour montrer l'efficacité de la méthode proposée. Ce travail a donné lieu à un article accepté dans le journal "Computational and Applied Mathematics (CAM)", sous le titre : An adaptive block tangential method for multi-input multi-output dynamical systems.

- **Chapitre**5 : Dans ce chapitre, une autre approche basée sur les sous-espaces de Krylov tangentiels a été proposée. On a donné ensuite un nouvel algorithme utilisant la technique Globale. On dérive des équations algébriques intéressantes pour le calcul des matrices réduites, ainsi que pour le calcul de la norme de l'erreur. Ce chapitre se termine par des exemples numériques qui illustrent l'efficacité de cet algorithme. Ceci a donné un article publié sous la référence : "A Computational Global Tangential Krylov Subspace Method for Model Reduction of Large-Scale MIMO Dynamical Systems". Journal of Scientific Computing, 75(2018), pp. 1614–1632.

Partie IV Cette partie est consacrée à l'introduction et l'étude de nouvelles méthodes basées sur la bi-orthogonalité de type Lanczos. Elle se se décompose en deux chapitres :

- **Chapitre**6 : Dans ce chapitre, on introduit un algorithme qui nous permet de construire deux bases à partir d'un sous-espaces de Krylov tangentiel et son dual. La construction se fait en utilisant le procédure de type Lanczos classique pour rendre les bases bi-orthogonales. On donne des propriétés algébriques qui nous permettent la construction des matrices réduites. Les tests numériques montrent les avantages de notre algorithme en le comparant à d'autres bien connus dans la littérature. Ce chapitre a donné lieu à un papier accepté dans le journal " Applied and Computational Mathematics", sous le titre : An adaptive tangential Lanczos-type method for model reduction in large-scale dynamical system.

- **Chapitre**7 : On s'est intéressé à une nouvelle technique dans laquelle on projette les équations de lyapunov sur des sous-espaces de Krylov tangentiels afin d'obtenir des équations de petite dimension. Ces équations de lyapunov réduites sont maintenant faciles à résoudre, ce qui nous permet de calculer les matrices réduites en utilisant la méthode de troncature équilibrée. On a ensuite donné des équations algébriques qui nous ont permis d'établir de nouveaux résultats théoriques. Une approche adaptative est utilisée pour le choix des points d'interpolation et les directions tangentielles. Finalement les expériences numériques montrent l'efficacité de cette nouvelle technique.

Une conclusion générale a été donnée avec quelques perspectives pour le travail futur.

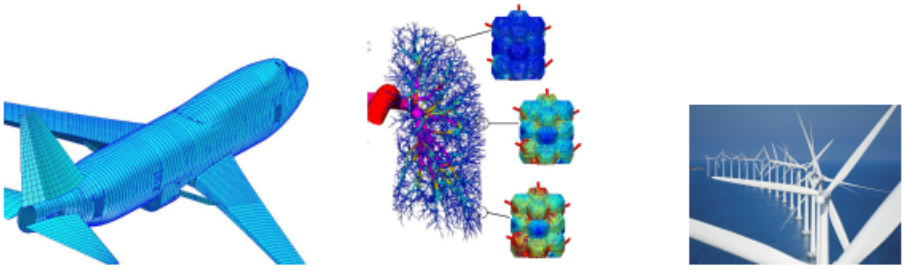
Première partie

General Introduction

Chapitre 1

Introduction

Large-scale systems are present in many engineering fields such as : aerospace, circuits, computational biology, building structure, automotive, weather forecasting ...



The determination of a numerical solution (even approximated) of a nonlinear and complex physical process (by its geometry or its multiphysical character for example) still requires significant computational resources today, either in computation time or in memory occupation. To set orders of magnitude for a realistic problem, Spalart et al [41] estimate that for an airplane wing at cruising flight conditions, it is necessary to employ about 1011 points and to integrate the Navier-Stokes equations for approximately 5.106 time steps. It seems difficult, with the numerical approaches currently used in fluid mechanics (finite ele-

ments, finite volumes, ...). Ideally, we would like to reduce the large number of degrees of freedom generally required to the dynamic description of the physical system (108 grid points, for example, in the case of a yet academic channel flow analyzed recently by Bewley, 2001) to some degrees of freedom in actual interaction. Thus, by agreeing to pay the cost of one (or more) resolutions of the detailed model, we could, for the same numerical cost, carry out a very large number of simulations of the reduced dynamic model.

Given the importance of the issues, model reduction is a research direction that has been, and remains, very active in many disciplinary fields. For example :

- In linear algebra (Antoulas, 2005) : Singular Value decomposition (SVD), Krylov methods, Hankel decomposition.
- In turbulence (Aubry & al, 1988 ; Ukeiley & al, 2001) : the modeling of turbulence consists precisely in replacing the Navier-Stokes equations by simpler models to solve, models k-ϵ or LES, weak order dynamical systems based on orthogonal decomposition to the Proper Orthogonal Decomposition (POD) values.
- In statistics (Karhunen, 1946, Love, 1955) : decomposition of Karhunen-Love.
- In Fluid Mechanics (Schlichting & Gersten, 2003) : boundary layer work in which a classical approach is to replace the Navier-Stokes equations with a combination of simpler models, Prandtl boundary layer equations near solid walls and a non-viscous model far from them.

The general problem associated with model reduction is described schematically in Figure (4.1a). Starting from any physical system, and from experimentally or numerically evaluated data, the modeling phase consists in determining a set of ordinary differential equations (ODEs) or partial differential equations (PDEs) representative of the physical system. In the case where an PDE system is obtained in the modeling phase, it is generally discretized in space in order to obtain an ODE system which is subsequently called $\mathcal{S}$. The model reduction phase consists in determining a dynamic system $\hat{\mathcal{S}}$ by appropriately reducing the number of ODEs necessary for the description of the system. Finally, the reduced

model $\hat{\mathcal{S}}$ is used to simulate or control the system $\mathcal{S}$. Eventually, the EDO system is also discretized in time, thus leading to a discrete dynamic system.

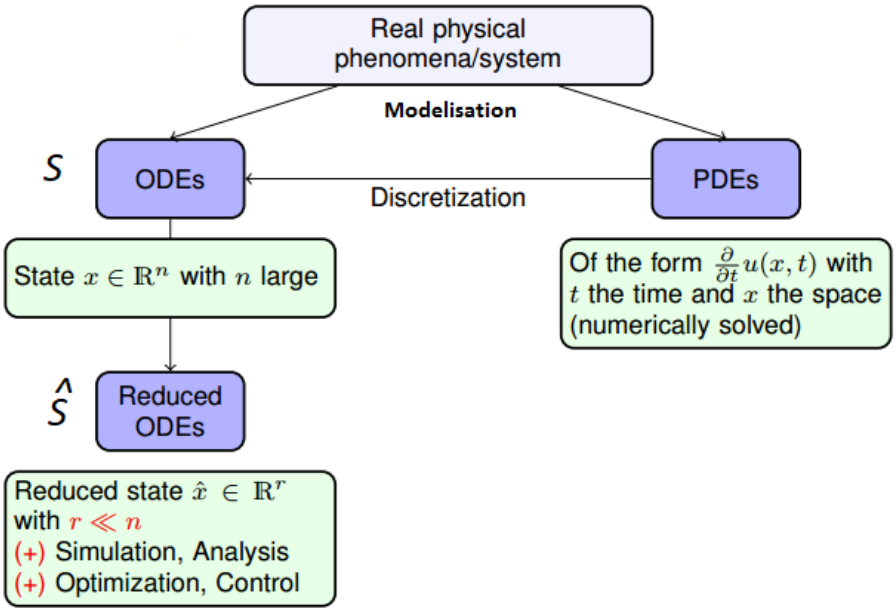


FIGURE 1.1 – General problematic of model reduction.

Let $\mathbb{T}$ be the set of values taken by time (depending on the case, we can consider $\mathbb{T} = \mathbb{R}^+$, $\mathbb{R}^-$ or $\mathbb{R}$ for a continuous system in time, or $\mathbb{T} = \mathbb{Z}^+$, $\mathbb{Z}^-$ or $\mathbb{Z}$ for a discrete system). Considers that the system $\mathcal{S}$ can be written generically as a system of first-order ODEs :

$$\mathcal{S} : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)) \\ y(t) &= g(t, x(t), u(t)) \end{cases} \tag{1.1}$$

where

- $\{x : \mathbb{T} \longrightarrow \mathbb{R}^n\}$ is the state vector.
- $\{u : \mathbb{T} \longrightarrow \mathbb{R}^m\}$ is the input or command vector.
- $\{y : \mathbb{T} \longrightarrow \mathbb{R}^p\}$ is the output or the observable vector.

where f and g are correctly scaled vector functions. The $\mathcal{S}$ system can therefore be represented by Figure 1.2 :

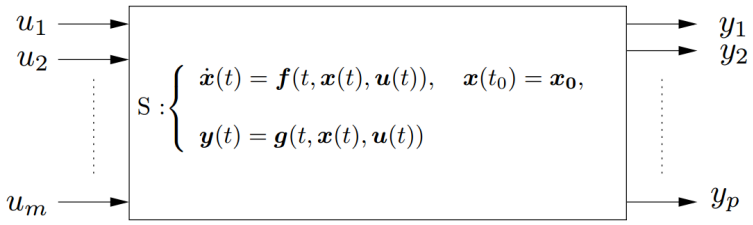


FIGURE 1.2 – Schematic representation of the dynamic system $\mathcal{S}$.

The main goal of the model reduction is to replace the system (1.1) with :

$$\hat{\mathcal{S}} : \begin{cases} \dot{\hat{x}}(t) &= \hat{f}(t, \hat{x}(t), u(t)) \\ \hat{y}(t) &= \hat{g}(t, \hat{x}(t), u(t)) \end{cases} \tag{1.2}$$

where,

- $\{\hat{x} : \mathbb{T} \longrightarrow \mathbb{R}^r\}$ $with$ $r \ll n$.
- $\{\hat{y} : \mathbb{T} \longrightarrow \mathbb{R}^p\}$.

Often in practice, the nonlinear system $\mathcal{S}$ is linearized around an equilibrium solution. We determine thus a linear system whose parameters are variable in time that we can still note $\mathcal{S}_{LPTV}$ where $LPTV$ means Linear, Parameter, Time-Varying. This system is given by :

$$\mathcal{S}_{LPTV} : \begin{cases} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{cases}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$.

Finally, if the system parameters do not depend on time, we obtain the system $\mathcal{S}_{LTI}$ where LTI means here Linear Time-Invariant. This system is then represented by :

$$\mathcal{S}_{LTI} : \begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{cases}$$

The $\mathcal{S}_{LPTV}$ and $\mathcal{S}_{LTI}$ systems have been extensively analyzed by automation specialists or mathematicians (Zhou et al., 1996, for example). In the case of linear

systems, many results exist on the reduction of models of large dynamic systems and research is still active in this field, as evidenced by many publications on the subject (Antoulas, 2005; Benneretal., 2005). According to *Antoulas* (2005), three large classes of approximation methods exist (see table below) :
1) Those based on Singular Value Decomposition (SVD).
2) Those based on a method of $Krylov$.
3) Finally, iterative methods combine some aspects from a Singular Values Decomposition method and others from a $Krylov$ method.

Approximation methods for dynamical systems		
SVD		**Krylov**
Nonlinear Systems	**Linear Systems**	
• POD methods • Empirical grammians	• Balanced truncation • Hankel approximation	• Realization • Interpolation • Lanczos • Arnoldi
SVD-Krylov		

FIGURE 1.3 – Classification of model reduction methods.

1.1 Formulation of the Problem

The systems considered in this thesis have the following form :

$$\Sigma := \begin{cases} x(0) &= x_0 \\ \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{cases} \tag{1.3}$$

the transfer function of the system (1.3) defined as

$$H(\omega) := C(\omega I_n - A)^{-1} B \in \mathbb{R}^{p \times q}. \tag{1.4}$$

In this mathematical formalism, the vector $x(t) \in \mathbb{R}^n$ represents the state of the system. he is called the state vector in theory of control. In mechanics, the state vector contains all the unknown physical quantities (pressure, displacement, speed, temperature, density ...). The matrix $A \in \mathbb{R}^{n\times n}$ represents in general the semi-discretisation of partial differential equations (PDEs) that model the physical phenomenon studied. The vector $u \in \mathbb{R}^p$ is called the input vector. The term $Bu(t)$ with $B = [b_1, ..., b_p] \in \mathbb{R}^{n\times p}$ is the source term or second member of the model. For example, it corresponds to external forces or forces control (imposed constraints, pressure fields, gravity, aerodynamic forces ...). The vector $y \in \mathbb{R}^q$ is called the vector of the outputs (output vector). The term $y = Cx(t)$ with $C \in \mathbb{R}^{q\times n}$ represents the physical quantities of interest that are observed. In mechanics, it can be the observation of the solution at the level of some sensors, particular characteristics of the flow or the structure. This system is said to be autonomous because the matrices A, B and C are assumed to be independent of time. It is first-rate and continuous in time. Finally x_0 is the initial condition. We talk about standard state space system or LTI system (Linear Time Invariant system). This type of system models the behavior of linear systems controlled by inputs/outputs. To specify the nature of the system, we speak respectively of SISO, SIMO, MISO or MIMO (S : Single, M : Multi, I : Input, O : Output) according to the number of inputs and outputs of the system. For example :

- if $p = q = 1$ the system is called SISO,
- if $p = 1$ and $q > 1$ the system is called SIMO,
- if $p > 1$ and $q = 1$ the system is called MISO,
- if $p > 1$ and $q > 1$ the system is called MIMO.

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

$$\Sigma_m := \begin{cases} \dot{x}_m(t) &= A_m x_m(t) + B_m u(t) \\ y_m(t) &= C_m x_m(t), \end{cases} \tag{1.5}$$

and its transfer function

$$H_m(\omega) := C_m(\omega I_m - A_m)^{-1} B_m \in \mathbb{R}^{p\times q}, \tag{1.6}$$

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 (1.5) 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. The reduced system (1.5) is always an LTI system, but much smaller in size than the original system (1.3). The problematic can also be formulated by working in the frequency space with the transfer function. The introduction of model reduction methods is in fact also motivated by the finding that it is not conceivable to calculate the transfer function H for problems large scale as this would solve an infinity of linear systems of very large large size (for each heartbeat). The formation of the transfer function is actually limited to models that can be expressed analytically. Given the matrix of transfer function $H(\omega)$, the goal is to construct a transfer function matrix $H_m(\omega)$, which is close to $H(\omega)$.

1.2 Other systems

The spatial semi-discretization of PDEs can lead to other forms of matrix system. The two examples that are often encountered and can be formally Brought back to the standard system, so to the whole theory as a result.

1.2.1 Generalized LTI system

A large number of spatially discretized physical models can be reduced to systems in the more general form :

$$\begin{cases} x(0) &= x_0 \\ \mathbf{E}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{cases}$$

The matrix $E \in \mathbb{R}^{n\times n}$ is usually called mass matrix. When the matrix E is regular,

it is enough to invert it to get back to the standard system (1.3). Sometimes E is singular and then we speak of a descriptor system or of generalized state space system. The descriptor systems certainly represent the most general form to describe linear physical models. The mathematical and numerical treatment of these systems, however, is more complicated.

1.2.2 Second order system

Linear PDEs modeling structures (plates, shells, beams ...) are often second order in time. The spatial semi-discretization of its models by a method of finite elements leads to systems that write in the form :

$$\begin{cases} q(0) & = & q_0 \\ \dot{q}(0) & = & q_1 \\ \mathbf{M}\ddot{q}(t) + \mathbf{D}\dot{q}(t) + \mathbf{K}q(t) & = & \mathbf{P}f(t) \\ y(t) & = & \mathbf{L}q(t), \end{cases}$$

where **M** is the mass matrix, **D** is the damping matrix and **K** the stiffness matrix. When the source term $\mathbf{P}f(t)$ is null, the system is said to be free, otherwise, it is said forced. The generalized LTI system is obtained by putting $x(t) = (\dot{q}(t), q(t))$, we will treat this later in details.
In the following of the thesis, we will always try to reduce to the standard system (1.3), to benefit from the theoretical justifications of the control formalism.

1.3 Motivating examples for model reduction

In this part we describe some benchmark examples for model reduction that we used in the numerical example linear time-invariant systems.

1.3.1 Compact Disc player example

It is the well known CD-Player example from the SLICOT [1] collection that has been frequently used as a test example in the literature. The control task is to

1. 5http ://www.slicot.org/index.php ?site=examples

follow the trace, which is to show Laser point to trace the pits on the spinning CD. The mechanism treated here, consists of a pivoting arm on which a lens is mounted by means of two horizontal leaf springs. The rotation of the arm in the horizontal plane allows the reading of spiral-shaped disc strips, and the suspended lens is used to focus the spot on the disc.

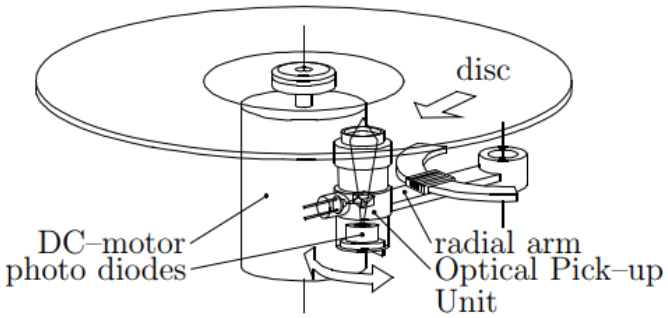


FIGURE 1.4 – Schematic view of CD mechanism.

Because of the fact that the disc is not perfectly flat, and because of the irregularities in the spiral of the pits on the disc, the challenge is to find a low cost controller that can make the servo system faster and less sensitive to external shocks [44]. The model contains 60 vibration modes.

1.3.2 The MNA example

The MNA model was obtained from NICONET [32]. To obtain the matrix of a multiport, the voltage sources are connected to the ports[36]. The multiport, with these sources, constitutes the equations of modified nodal analysis (MNA) :

$$\begin{cases} E\dot{x}_n &= Ax_n + Bu_p \\ i_p &= Cx_n \end{cases}$$

The i_p and u_p vectors denote the port currents and voltages, respectively, and

$$A = \begin{bmatrix} -N & -G \\ G^T & 0 \end{bmatrix}, \quad E = \begin{bmatrix} L & 0 \\ 0 & H \end{bmatrix}, \quad x_n = \begin{bmatrix} v \\ i \end{bmatrix},$$

where v and i are the MNA variables corresponding to the node voltages, inductance and voltage source currents, respectively. The matrices $n \times n$ A and E represent the conductance and susceptance matrices, while $-N$, L and H are the matrices stamps for resistors, capacitors and inductances, respectively. G consists of 1, -1 and 0, which represent the common variables in KCL equations. Provided the original $N\text{-}port$ is composed only of passive linear elements, L, H and $-N$ are symmetric non-negative defined matrices. This implies that E is also symmetrical and not negative. Since this is an N-port formulation, where the only sources are the N voltage sources port nodes, $B = C^T$. The matrices E all have several singular modes. We have five sparse examples :

The file name	Dimension n
MNA_1	n = 578
MNA_2	n = 9223
MNA_3	n = 4863
MNA_4	n = 980
MNA_5	n = 10913

1.3.3 FDM Semi-Discretized Heat Equation

The finite differences semi-discretized heat equation will serve as the most basic test example here. Its corresponding matrix A, is obtained from the centered finite difference discretization of the operator,

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

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

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

The matrices B and C were random matrices with entries uniformly distributed in $[0,1]$. The advantages of this model are :

• It's easy to understand.

• The discretization using the finite difference method (FDM) is easy to implement.

• It allows for simple generation of almost arbitrary size test problems.

1.3.4 The RAIL model

This example is a simplified linear model of a nonlinear problem obtained from a cooling process, which is part of the method of manufacturing steel rails[42]. The temperature of the rail is lowered by the water sprayed through several nozzles on its surface. Since the problem is "frozen" with respect to a spatial dimension and symmetric with respect to another, it suffices to consider the problem related to half of the cross-section of the rail, where the homogeneous limit of Neumann conditions are imposed on the artificial boundary segment Γ_7 (see Figure 1.5). The pressure of the the nozzles can be oriented independently for different sections Γ_1,..., Γ_6 of the surface. That corresponds the control of the limits of a two-dimensional unsteady heat equation in $x = x(\tau,\ \xi_1,\ \xi_2)$. The nozzle pressures provide the input signals $u_i = u_i(\tau)$, which form the right side of the boundary conditions of the third type (20). The output signals of this model are given by the temperature in several interior observation points marked by small circles in Figure 1.5. After an appropriate scaling of the physical quantities

we get the parabolic differential equation

$$\begin{cases} \dfrac{\partial}{\partial\tau}x = \dfrac{\partial^2}{\partial\xi_1^2}x + \dfrac{\partial^2}{\partial\xi_2^2}x, & (\xi_1,\xi_2)\in\Omega, \\ x + \dfrac{\partial^2}{\partial\vec{n}}x = u_i, & (\xi_1,\xi_2)\in\Gamma_i, \quad i=1,...,6 \\ \dfrac{\partial^2}{\partial\vec{n}}x = 0, & (\xi_1,\xi_2)\in\Gamma_7. \end{cases} \tag{1.7}$$

The finite element discretization of the problem is obtained by using the Matlab PDE Toolbox. Figure 1.5 shows the initial triangularization. The actual trian-

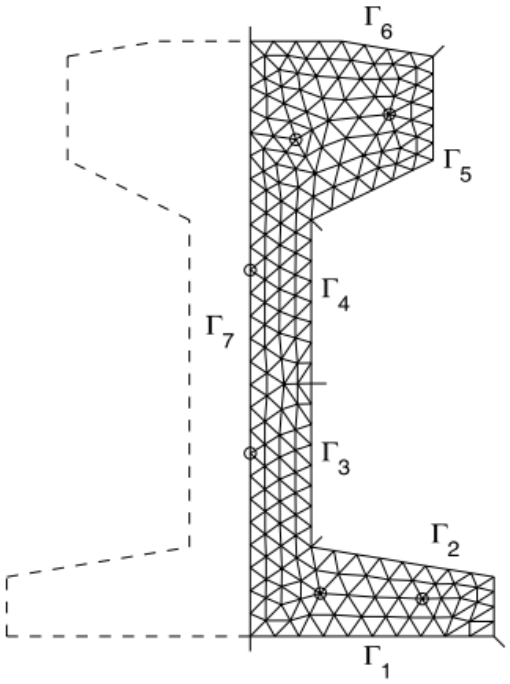


Figure 1.5 – Cross section of the steel rail and initial triangularization.

gularization is the result of two steps of regular mesh refinement, i.e., in each refinement step all triangles are split into four congruent triangles.

1.3.5 The ISS model : International Space Station.

The assembly and operation of the International Space Station (ISS) poses unique control challenges because of its complex, variable flexible structure as well as variety of operational modes and control systems. I (it is estimated that more than 40 Space Shuttle flights will be required to complete the assembly). In addition, each module is described in terms of $n \approx 10^3$ state variables. In this case, the controllers will be needed to reduce the oscillatory motion or to set the orientation of the space station relative to a desired direction. It is well known that generically a controller of a given plant has the same dimension as the plant. Since these controllers must be implemented on board, they must have low complexity due to hardware, radiation, throughput or test limits [4]. Thus, reduced-order models are very useful for the development of reduced-order controllers. A brief overview of structural flexibility during assembly is shown in Figure 1.6.

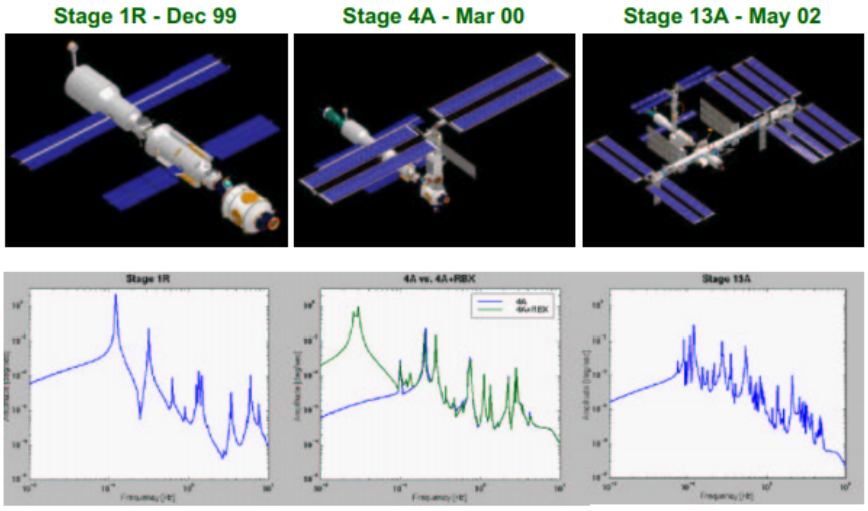


FIGURE 1.6 – ISS : Evolution of the frequency response as more components are added (Data : Draper Labs).

The figure shows particular steps of the assembly sequence of the space station and a frequency response curve thruster commands (Roll, Pitch, Yaw) to filtered Rate Gyro sensor. It is obvious that the complexity and size of the flex models

grows as new structural elements are added. Therefore, to perform controller flex structure dynamic interaction assessment, it becomes necessary to reduce the flex models in order to complete the analysis in a timely manner that meets the assembly schedule.

1.3.6 Chemical Reactors : Controling the Temperature of an Inflowing Reagent

The next example is a system appearing in the optimal heating/cooling of a fluid flow in a tube. An application could be the temperature regulation of certain reagent inflows in chemical reactors. The model equations are :

$$\begin{aligned} &\frac{\partial x}{\partial t} - \kappa \Delta x + v.\nabla x = 0, && in\ \Omega \\ &x = x_0, && on\ \Gamma_{in} \\ &\frac{\partial x}{\partial \vec{n}} = \sigma(u - x), && on\ \Gamma_{heat1} \cup \Gamma_{heat2} \\ &\frac{\partial x}{\partial \vec{n}} = 0, && on\ \Gamma_{out} \end{aligned} \tag{1.8}$$

Here Ω is the rectangular domain shown in Figure 1.7. The inflow Γ_{in} is at the left

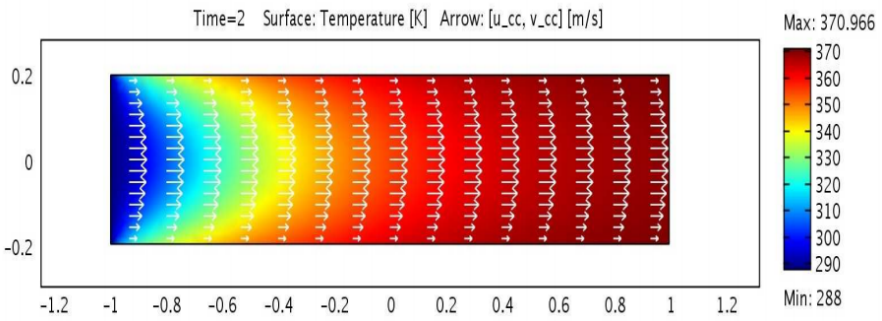


FIGURE 1.7 – A 2d cross-section of the liquid flow in a round tube.

part of the boundary and the outflow Γ_{out} the right one. The control is applied via the upper and lower boundaries. We can restrict ourselves to this 2d-domain assuming rotational symmetry, i.e., non-turbulent diffusion dominated flows.

The test matrices have been created using the COMSOL [2] Multiphysics software and have dimension 1090.

1.3.7 Eady example

This is a model of the atmospheric storm track (for example the region in the midlatitude Pacific). The mean flow is taken to be in a periodic channel in the zonal x-direction, $0 < x < 12\pi$, the channel is taken to be bounded with walls in the meridional y-direction located at $y = \pm\frac{\pi}{2}$ and at the ground, $z = 0$, and the tropopause, $z = 1$. The mean velocity is varying only with height and it is $U(z) = 0.2 + z$. Zonal and meridional lengths are non dimensionalized by $L = 1000km$, vertical scales by $H = 10km$, velocity by $U_0 = 30m/s$ and time is nondimensionalized advectively, i.e. $T = \frac{L}{U_0}$, so that a time unit is about $9h$.
In order to simulate the lack of coherence of the cyclone waves around the Earth's atmosphere, an observed characteristic of the Earth's atmosphere, we introduce linear damping at the storm track's entry and exit region. The perturbation variable is the perturbation geopotential height (i.e. the height at which surfaces of constant pressure are located).
The perturbation equations for single harmonic perturbations in the meridional (y) direction of the form $\phi(x,z,t)e^{ily}$ are :

$$\frac{\partial\phi}{\partial t} = \nabla^{-2}\Big[-z\nabla^2 D\phi - r(x)\nabla^2\phi\Big],$$

where ∇^2 is the Laplacian $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} - l^2$ and $D = \frac{\partial}{\partial x}$. The linear damping rate $r(x)$ is taken to be $r(x) = h\Big(2 - tanh[(x - \frac{\pi}{4})/\delta] + tanh[(x - \frac{7\pi}{2})/\delta]\Big)$ $(h = 2.5, \delta = 1.5)$. The boundary conditions are expressing the conservation of potential temperature (entropy) along the solid surfaces at the ground and tropopause :

$$\frac{\partial^2\phi}{\partial t\partial z} = -zD\frac{\partial\phi}{\partial z} + D\phi - r(x)\frac{\partial\phi}{\partial z} \quad at \quad z = 0,$$

$$\frac{\partial^2\phi}{\partial t\partial z} = -zD\frac{\partial\phi}{\partial z} + D\phi - r(x)\frac{\partial\phi}{\partial z} \quad at \quad z = 1.$$

2. https ://www.comsol.de

Note that these equations are the same for perturbation evolution in a Couette flow with free boundaries. We write the dynamical system in generalized velocity variables $\psi = (-\nabla^2)^{\frac{1}{2}}\phi$ so that the dynamical system is governed by the dynamical operator :

$$A = (-\nabla^2)^{\frac{1}{2}}\nabla^{-2}\left(-zD\nabla^2 + r(x)\nabla^2\right)(-\nabla^2)^{\frac{-1}{2}},$$

where the boundary equations have rendered the operators invertible. We refer to [15] for more details, including the type of discretization that was used.

Deuxième partie

Model reduction & LTI system theory

Chapitre 2

LTI systems theory

In this chapter, some general elements about LTI systems theory are recalled and the associated notations introduced. The material is standard and can be found in many books such as Zhou et al [45].

2.1 Representation of LTI dynamical models

Time-domain representation

We consider the multi-input and multi-output (MIMO) linear time invariant (LTI) dynamical system in (1.3) which represented in the time-domain by a state-space This form

$$\Sigma := \left\{ \begin{array}{rcl} x(0) & = & x_0 \\ \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t), \end{array} \right.$$

The LTI dynamical system above is usually denoted as

$$\Sigma := \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array} \right]. \tag{2.1}$$

The temporal solution of this system is established by applying the variation of constants method,

$$y(t) = Cx(t) = C\left[e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau\right],$$

where e^{tA} expresses the exponential of the matrix tA. It is defined by the absolutely convergent series $e^{tA} = I_n + \sum_{i=1}^{\infty} \frac{t^i}{i!}A^i$.
As a recall, the exponential matrix satisfies the following properties :

$$e^{t_1 A}e^{t_2 A} = e^{(t_1+t_2)A}.$$

$$\frac{d}{dt}(e^{tA}) = Ae^{tA} = e^{tA}A.$$

$$If\ AB = BA, \quad e^{tA}e^{tB} = e^{t(A+B)}.$$

$$e^{PAP^{-1}} = Pe^{A}P^{-1}.$$

where $t_1,\ t_2 \in \mathbb{R}$ and P is an invertible matrix. The solution of a problem with initials values not controlled ($Bu(t) = 0$) is simply expressed in the form :

$$y(t) = Ce^{(t-t_0)A}x_0.$$

Frequency-domain representation

The transfer function $H(\omega)$ represents the model Σ in the frequency-domain and it's obtained by applying the Laplace transform to (1.3) with a null initial conditions ($x_0 = 0$), which yields to :

$$\begin{cases} \omega X(\omega) &= AX(\omega) + BU(\omega) \\ Y(\omega) &= CX(\omega), \end{cases} \tag{2.2}$$

where $X(\omega)$, $Y(\omega)$ and $U(\omega)$ are the Laplace transform of $x(t)$, $y(t)$ and $u(t)$, respectively. If we re-arranging terms in the previous equation we obtain $Y(\omega) = H(\omega)U(\omega)$, where $H(\omega)$ is the transfer function of the system (1.3) defined as in

(1.4) by :

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

The main advantage of working directly with the transfer function is based on the fact that a distance ||.|| can be defined between the transfer function of the original model and that of the reduced model. Indeed, the matrices $H_m(\omega)$ and $H(\omega)$ are of the same size. The definition of such a norm for the transfer function is a way of estimating the error made when a reduced system is built. It is thus possible to evaluate and guarantee the quality of the model reduced with respect to the defined norm. Of course, the approximation quality of the reduced system also depends on the choice of the considered norm.

2.2 Basic properties of the LTI systems

The dynamic and numerical properties of the LTI system are closely related to the properties of the matrix A. We assume for simplicity that the matrix is with real coefficients.

2.2.1 Stability

Stability is defined in relation to the free response of the system (uncontrolled system). We say that the system is stable if the solution $x(t)$ for $u = 0$ and for any initial condition remains bounded when $t \to \infty$. The stability of the system can be analyzed directly from its modal representation. For matrix A to be stable, all the real parts of the eigenvalues need to be strictly negative. This remains valid if the real part is null on condition that any eigenvalue with real part null is simple. For example, if matrix A is negative definite then the system is stable. Stable matrices are also called Hurwitz matrices. The LTI dynamical system (1.3) is asymptotically stable, if and only if A is stable ($\Lambda(A) \subset \mathbb{C}^-$).

2.2.2 Passivity

The system is passive if it can not generate energy. In other words, the energy of the system can not increase. In theory, passivity is checked using one of the

following equivalent properties relating to the transfer function :

- H is analytic in $\mathbb{C}^+ := \{\omega \in \mathbb{C},\ \Re(\omega) > 0\}$.
- $H(\overline{\omega}) = \overline{H(\omega)}, \quad \forall \omega \in \mathbb{C}$.
- $H(\omega) + H(\omega)^T > 0, \quad \forall \omega \in \mathbb{C}^+$.

Passivity is a characteristic stronger than stability. The energy of a stable system can indeed grow very well on short time.

2.2.3 Preconditioning the system

It is sometimes possible to improve the numerical properties of the system by applying a regular matrix **P** such that,

$$\mathbf{P}\dot{x}(t) = \mathbf{P}Ax(t) + \mathbf{P}Bu(t),$$

The matrix **P** is called the preconditioner of the system. The construction of a pre-conditioner aims to improve the numerical conditioning of a matrix. Without going into the details, let's just say that the ultimate goal of a preconditioning is to get a **PA** matrix who :

• Be less sensitive to various numerical perturbations.

• Accelerate the convergence of numerical algorithms.

These improvements are largely due to the modification of the **PA** product spectrum compared to the spectrum of **A**. The application of such a preconditioner implies the work on the generalized LTI system. Unfortunately, the construction of such a matrix **P** is not easy in general.

2.2.4 Controllability and Observability Gramians

The concepts of controllability and observability are very important for the study of dynamic systems control and estimation problems. Indeed the notion of controllability is closely related to the existence of a "feasible" command, and the notion of observability is related to the possibility of estimating state variables from measurements.

Definition 2.1 (Controllability) *Given a LTI dynamic system,*

• *The state x_0 of a linear system is said to be controllable at a moment t_0, if there is*

an entry (i.e. a command) $u(t)$ that transfers the initial state $x_0 = x(t_0)$ to any other state in a finite time $t_1 > t_0$.

• *The system is controllable at the moment t_0 if this is true for any initial state $x(t_0)$.*

• *The system is said to be totally controllable if this is true for all moment t_0 and any initial state $x(t_0)$.*

Proposition 2.1 (Controllability) *The LTI dynamical system* (1.3) *is controllable if and only if the controllability matrix*

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix},$$

is of full rank, i.e $rank(\mathcal{C}) = n$. In this case the pair (A, B) is said to be controllable. If $rank(\mathcal{C}) = k < n$, then $n-k$ is the number of the uncontrollable modes (the eigenvalues of the matrix A satisfying $rank([\lambda I - AB]) < n$). If all uncontrollable modes are stable then the system is said to be stabilizable.

Example : Given the followig system,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u,$$

the controllability matrix is given by,

$$\mathcal{C} = \begin{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} & \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 1 & -2 \end{bmatrix},$$

$rank(\mathcal{C}) = 2$, the system is controllable.

Definition 2.2 (Observability) *Given a LTI dynamic system,*

• *The state $x_0 = x(t_0)$ of a linear system is said to be observable at a moment t_0 if it can be determined by the knowledge of $u(\tau)$ and $y(\tau)$ for $t_0 < \tau < t_1$ with t_1 is finite time.*

• *The system is observable at the moment t_0 if this is true for any initial state $x(t_0)$.*

• *The system is said to be totally observable if this is true for all moment t_0 and any initial state $x(t_0)$.*

Proposition 2.2 (observability) *The LTI dynamical system* (1.3) *is observable if and only if the observability matrix*

$$\mathcal{O} = \left[C^T,\ A^T C^T,\ (A^2)^T C^T,\ ...,\ (A^{n-1})^T C^T\right],$$

is of full rank, i.e $rank(\mathcal{O}) = n$. *In this case the pair* (A, C) *is said to be observable. If* $rank(\mathcal{O}) = l < n$, *then* $n - l$ *is the number of the unobservable modes (the eigenvalues of the matrix A satisfying* $rank([\lambda I - A^T C^T]) < n$ *).*

Example : Given the followig system,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

the observability matrix is given by,

$$\mathcal{O} = \left[\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right] = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix},$$

$rank(\mathcal{O}) = 1 \neq 2$, the system is not observable.

Definition 2.3 (Controllability & Observability Gramians) *Given a stable LTI dynamical system* (1.3). *The associated controllability Gramian, denoted by* $\mathcal{P}$, *is defined as*

$$\mathcal{P} = \int_0^\infty e^{tA} B B^T e^{A^T t} dt, \tag{2.3}$$

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

$$\mathcal{Q} = \int_0^\infty e^{A^T t} C^T C e^{tA} dt. \tag{2.4}$$

In the frequency domain, we get the formulations

$$\mathcal{P} = \frac{1}{2\pi} \int_{-\infty}^{\infty} (j\omega I_n - A)^{-1} B B^T (j\omega I_n - A)^{-T} d\omega, \tag{2.5}$$

and

$$\mathcal{Q} = \frac{1}{2\pi} \int_{-\infty}^{\infty} (j\omega I_n - A)^{-T} C^T C (j\omega I_n - A)^{-1} d\omega. \tag{2.6}$$

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

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0, \tag{2.7}$$

and

$$A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0. \tag{2.8}$$

Theorem 2.1 *The LTI system* (1.3) *is controllable if and only if the solution* $\mathcal{P}$ *of* (2.7) *is positive definite, it is observable if and only if the solution* $\mathcal{Q}$ *of* (2.8) *is positive definite.*

Lemma 2.2.1 *The LTI system in* (1.3) *is minimal if and only if* (A, C) *is observable and* (A, B) *is controllable.*

2.2.5 Norms of systems

Various norms can be used to verify the quality of the reduced order model. Here we recall the well known $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of a transfer function.

The $\mathcal{H}_2$ norm

Definition 2.4 *Given a LTI dynamical model as in* (1.3) *with the associated transfer function* H*, the* $\mathcal{H}_2$*-norm of* H *is defined as*

$$\| H \|^2_{\mathcal{H}_2} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \boldsymbol{tr}\Big(H(j\omega)H(-j\omega)^T\Big)\, d\omega. \tag{2.9}$$

Theorem 2.2 *Given a stable LTI dynamical system* Σ*, with the associated Gramians* $\mathcal{P}$ *and* $\mathcal{Q}$*, the* $\mathcal{H}_2$*-norm of* H *is given by*

$$\begin{aligned} \| H \|^2_{\mathcal{H}_2} &= \boldsymbol{tr}(C\mathcal{P}C^T), \\ &= \boldsymbol{tr}(B^T\mathcal{Q}B). \end{aligned} \tag{2.10}$$

The $\mathcal{H}_\infty$ norm

We present below the definition of the $\mathcal{H}_\infty$-norm of a LTI dynamical model. For MIMO models, it is the maximum singular value of the transfer function across all frequencies.

Definition 2.5 *Given an asymptotically stable LTI dynamical mod. The $\mathcal{H}_\infty$-norm of the transfer function H is defined by*

$$\| H \|_{\mathcal{H}_\infty} = \max_{\omega \in \mathbb{R}} \sigma_{max}(H(j\omega)), \tag{2.11}$$

where σ_{max} denotes the maximum singular values.

2.3 The Krylov subspaces

Before defining the Krylove subspaces, let us first recall the definition of the Singular Value Decomposition.

2.3.1 The Singular Value Decomposition

The singular value decomposition (SVD), has become since a few decades a fundamental tool for studying linear problems. The decomposition was discovered over a hundred years ago by Beltrami, but has only become a numerical tool since the late of 1960s, when G. Golub showed how it could be calculated stably and efficiently. We state the main theorem in the case of real matrices.

Theorem 2.3 *Let $A \in \mathbb{R}^{m\times n}$ be a matrix of rank r. There are two orthogonal matrices $U \in \mathbb{R}^{m\times m}$, $(U^T U = UU^T = I)$ et $V \in \mathbb{R}^{n\times n}$, $(V^T V = VV^T = I)$ such that :*

$$A = UDV^T \qquad , \qquad D = \begin{pmatrix} D_r & 0 \\ 0 & 0 \end{pmatrix} \tag{2.12}$$

$$\text{where} \quad D \in \mathbb{R}^{m\times n}, \quad D_r = Diag(\sigma_1, \sigma_2, ..., \sigma_r) \ \text{ et } \quad \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$$

2.3.2 Krylov subspaces

The concept is named after Russian applied mathematician and naval engineer Alexei Krylov, who published a paper about it in 1931[29]. The basis for its subspaces can be found in the Cayley-Hamilton theorem, which says that the inverse of a matrix A is expressed in terms of a linear combination of powers of A.

Definition 2.6 *Given $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, the m order Krylov $\mathcal{K}_m$ subspace is the linear subspace spanned by the images of bunder the first m-1 powers of A, that is :*

$$\mathcal{K}_m(A,B) = Range\{b,\ Ab,\ A^2b,\ ...,A^{m-1}b\}.$$

The Krylov subspaces form a croissant family of subspaces, necessarily limited. We will note m_{max}, the maximum dimension of Krylov subspaces, for a given vector $b \in \mathbb{R}^n$.

Lemma 2.3.1 *If $A^m b \in \mathcal{K}_m$, then $A^{m+k}b \in \mathcal{K}_m$ for all $k > 0$.*

Proof The demonstration is by recurrence. If for $k > 0$, $A^{m+k}b \in \mathcal{K}_m$, then :

$$A^{m+k}b = \sum_{l=0}^{m-1} \alpha_l A^l b$$

therefore :

$$\begin{aligned} A^{m+k+1}b &= \sum_{l=0}^{m-1} \alpha_l A^{l+1} b \\ &= \sum_{l=0}^{m-2} \alpha_l A^{l+1} b + \alpha_{m-1} \sum_{l=0}^{m-1} \beta_k A^l b \\ &= \sum_{l=0}^{m-1} \gamma_l A^l b \end{aligned}$$

Theorem 2.4 *The dimension of the Krylov subspace $\mathcal{K}_m$ is m, if and only if the grade of b associated to A is greater than $m-1$, which means :*

$$dim(\mathcal{K}_m) = m \Longleftrightarrow grade(b) \geq m,$$

$$dim(\mathcal{K}_m) = min\{m, grade(b)\}.$$

Where $grade(b)$ is the degree the minimal polynomial of A associated to the vector b i.e :

$$p(A)b = 0.$$

2.4 The Arnoldi & Lanczos methods

In this section we recall the definition of the Arnoldi and Lanczos methods using to construct the Krylov subspaces, with their properties.

2.4.1 The Arnoldi algorithm

The Arnoldi algorithm is a method to construct a $Krylov$ subspace through an orthonormal basis. The principle is simple : we start by a initial normalized vector, the vectors of the Krylov subspace are produced one after the other by multiplication by the system matrix and orthonormal by using the $GramSchmidt$ procedure.

Algorithm 1 .

1. *Choose a vector* v_1 *of an unitary norm.*
2. *For* $j = 1 : m$
3. *Set* $w := Av_j$.
4. *For* $i = 1 : j$
5. $h_{ij} =< w, v_i >$.
6. $w := w - h_{ij}v_i$.
7. *End*
8. $h_{j+1,j} =\| w \|$.
9. *If* $h_{j+1,j} = 0$, *then stop, else,*
10. $v_{j+1} = \dfrac{w}{h_{j+1,j}}$.
11. *End.*

12. Output $\{v_1, v_2, ..., v_{m+1}\}$.

1) Suppose that at the m^{th} step, the algorithm does not stop, the vectors $v_1, v_2, ..., v_m$ form an orthonormal basis of the Krylov subspace :

$$\mathcal{K}_m(A, v_1) = Range\left\{v_1,\ Av_1,\ A^2v_2, ...,\ A^{m-1}v_1\right\}.$$

2) Let V_m be the matrix of $(n \times m)$ whose columns are the vectors $v_1, v_2, ..., v_m$; let $\widetilde{H}_m$, be the Hessenberg matrix defined by the algorithm, and let H_m be the square matrix obtained from $\widetilde{H}_m$ by eliminating its last line, we have the following relations :

$$\begin{aligned} AV_m &= V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \\ &= V_{m+1} \widetilde{H}_m. \\ V_m^T A V_m &= H_m. \end{aligned} \tag{2.13}$$

In other words, this algorithm calculates the projection of the matrix A in the space defined by the Krylov vectors.
3) The *Arnoldi* algorithm stops at step j if and only if the minimal polynomial of v_1 is of degree j.

2.4.2 Block Arnoldi algorithm

In many cases, it is preferable to work with a block of vectors instead of a single vector, because the matrix A that's operates on a group of vectors, can be more efficient using block Krylov methods. There are several algorithm versions of block Arnoldi, here we will just recall one.

Algorithm 2 .

1. *Inputs* $A \in \mathbb{R}^{n \times n}$, $V_1 \in \mathbb{R}^{n \times p}$.
2. *For* $j = 1 : m$
3. *Set* $W := AV_j$.
4. *For* $i = 1 : j$.
5. $H_{ij} = V_i^T W$.

6. $W := W - V_i H_{ij}$.
7. *End.*
8. $W = V_{j+1} H_{j+1,j}$, *QR decomposition.*
9. *End.*
10. *Output* $\mathbb{V}_{m+1} = [V_1, V_2, ..., V_{m+1}] \in \mathbb{R}^{n \times mp}$.

The blocks $[V_1, V_2, ..., V_{m+1}]$ constructed in the algorithm 2 are orthogonal. The following relation, which is analogous to relation (2.13), can be easily proved :

$$A\mathbb{V}_m = \mathbb{V}_m \mathbb{H}_m + V_{m+1} H_{m+1,m} E_m^T.$$

where $\mathbb{H}_m$ is the Hessenberg matrix, and E_m is last $mp \times p$ block of the identity matrix I_{mp}.

2.4.3 Lanczos algorithm

Lanczos methods are iterative projection methods of the Krylov subspaces, that solve a large linear sparse systems. In 1950 and 1952, Lanczos [30, 31] proposed two processes for reduce a matrix A to a tridiagonal matrix T similar to A and build the bases of the desired Krylov subspaces. However, if the matrix A is symmetric the Lanczos process can be deduced from the Arnoldi process. A version equivalent to this algorithm has been introduced by M. Hestnes E. steifel [26] is the Conjugate gradient (CG) method. In the non-symmetric case of matrix A, Lanczos proposed a biorthogonalization process, which is distinct in principle from that of the Arnoldi algorithm. The algorithm proposed by Lanczos for non-symmetric matrices consists in transforming an $A \in \mathbb{R}^{n \times n}$ matrix into a similar tridiagonal matrix, and allows to construct a pair of biorthonormal bases $\{v_1, v_2, ..., v_m\}$ and $\{w_1, w_2, ..., w_m\}$ for the following two Krylov subspaces :

$$\mathcal{K}_m(A, v_1) = Range\left\{v_1,\ Av_1,\ A^2 v_2, ...,\ A^{m-1} v_1\right\}.$$

$$\mathcal{K}_m(A^T, w_1) = Range\left\{w_1,\ A^T w_1,\ (A^T)^2 w_2, ...,\ (A^T)^{m-1} w_1\right\}.$$

The biorthogonalization algorithm of Lanczos [31, 38] is given as follows :

Algorithm 3 .

1. *Choose two vectors* v_1 *and* w_1 *such as* $< v_1, w_1 >= 1$.
2. *Set* $\beta_1 = \gamma_1 = 0,\ w_0 \equiv v_0 \equiv 0$.
3. *For* $j = 1 : m$
4. $\alpha_j =< Av_i, wj >$.
5. $\widetilde{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$.
6. $\widetilde{w}_{j+1} = A^T w_j - \alpha_j w_j - \gamma_j w_{j-1}$.
7. $\gamma_{j+1} = | < \widetilde{w}_{j+1}, \widetilde{w}_{j+1} > |^{\frac{1}{2}}$, *if* $\gamma_{j+1} = 0$, *then stop, else*
8. $\beta_{j+1} = \dfrac{< \widetilde{w}_{j+1}, \widetilde{w}_{j+1} >}{\gamma_{j+1}}$.
9. $v_{j+1} = \dfrac{\widetilde{v}_{j+1}}{\gamma_{j+1}}, \qquad w_{j+1} = \dfrac{\widetilde{w}_{j+1}}{\beta_{j+1}}$
10. *end*
11. *Output* $\{v_1, v_2, ..., v_{m+1}\}, \qquad \{w_1, w_2, ..., w_{m+1}\}$.

Lanczos vectors can be generated by three recurrence terms. These recurrences can be indicated compactly in matrix form as follows :

$$AV_m = V_m T_m + \gamma_{m+1} v_{m+1} e_m^T.$$

$$A^T W_m = W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T.$$

$$W_m^T A V_m = T_m, \qquad W_m^T V_m = I_n,$$

where I_m is the identity matrix, V_m, W_m and T_m are matrices given by :

$$V_m = [v_1, v_2, ..., v_m], \qquad W_m = [w_1, w_2, ..., w_m],$$

$$\begin{pmatrix} \alpha_1 & \beta_2 & & \\ \gamma_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_m \\ & & \gamma_m & \alpha_m \end{pmatrix}.$$

Chapitre 3

Different model reduction techniques

Introduction

The model reduction methods of large dynamical systems are mainly divided into two families, methods that do not use projection and those that use it. Only the second family of methods is considered here because it responses better to the numerical requirements of large systems. The problem of projection reduction is expressed as follows :

Problem 1

Given the system Σ in (1.3) to reduce, the main goal here is to find two projection matrices $V_m, W_m \in \mathbb{R}^{n\times m}$ $(m \ll n)$ biorthogonal $(W_m^T V_m = I_m)$ such as the matrices of the Σ_m in (1.5) model are written :

$$A_m = W_m^T A V_m, \quad B_m = W_m^T B \ \ and \quad C_m = C V_m.$$

In the following, we present some well knowing methods for model reduction using projection.

3.1 Balanced truncation method

The balanced truncation method for model reduction was first introduced by Mullis & Roberts [35] and later in systems and control theory by Moore and

Glover, see [33, 20]. If we assume that the LTI system (1.3) 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 nonsingular matrix transformation $T \in \mathbb{R}^{n\times n}$ to get,

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

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

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

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,

$$\widetilde{\mathcal{P}} = \widetilde{\mathcal{Q}} = \mathbf{diag}(\sigma_1, ..., \sigma_n), \tag{3.1}$$

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

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

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 (3.1). First we compute directly the lower Cholesky factorization of the Gramians $\mathcal{P}$ and $\mathcal{Q}$,

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

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

$$L_o^T L_c = U\mathcal{D}V^T, \tag{3.2}$$

where $\mathcal{D}$ is the diagonal matrix containing the Hankel singular values of the system (1.3). The balanced transformation is given by,

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

It is proved in [1] that if the system (1.3) is stable and minimal, and having the new equivalent LTI dynamical system $\widetilde{\Sigma}$,

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

with $\widetilde{\mathcal{P}} = \widetilde{\mathcal{Q}} = \mathbf{diag}(\sigma_1, ..., \sigma_m, \sigma_{m+1}, ..., \sigma_n)$, then,

$$\| H(.) - H_m(.) \|_{\mathcal{H}_\infty} \leq 2(\sigma_{m+1} + ... + \sigma_n). \tag{3.3}$$

Inequality (3.3) shows that the dynamical system (1.3) can be represented by a reduced order LTI system $\widetilde{\Sigma}_m$, if the singular values $\sigma_{m+1}, ..., \sigma_n$ are small enough,

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

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

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

where $\mathcal{D}_m$, U_m and V_m correspond to the first m columns of the matrices $\mathcal{D}$, U and V in (3.2). Then the reduced model Σ_m is given as

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

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

Algorithm 4 .

1. *Inputs : A, B, C, m.*

2. *Solve* $\quad A\mathcal{P} + \mathcal{P}A^T + BB^T = 0.$
3. *Solve* $\quad A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0.$
4. $\mathcal{P} = L_c L_c^T$, $\mathcal{Q} = L_o L_o^T$. *(lower Cholesky factorizations)*
5. $L_o^T L_c = U\mathcal{D}V^T$. *(SVD decomposition)*
6. *Set* $T = L_c V \mathcal{D}^{-\frac{1}{2}}$, $\quad T^{-1} = \mathcal{D}^{-\frac{1}{2}} U^T L_o^T$.
7. *Apply projector* $\widetilde{\Sigma} := \left(\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & \end{array}\right)$.
8. *Truncation :* $A_m = \mathcal{W}_m^T A \mathcal{V}_m$, $B_m = \mathcal{W}_m^T B$ *and* $C_m = C\mathcal{V}_m$.

The transformation T constructed in Algorithm 4 (step 6), verify the condition in (3.1), in fact :

$$\begin{aligned} T^{-1}\mathcal{P}T^{-T} &= \mathcal{D}^{-\frac{1}{2}} U^T L_o^T \mathcal{P} L_o U \mathcal{D}^{-\frac{1}{2}} \\ &= \mathcal{D}^{-\frac{1}{2}} U^T L_o^T L_c L_c^T L_o U \mathcal{D}^{-\frac{1}{2}} \\ &= \mathcal{D}. \end{aligned}$$

Similarly,

$$\begin{aligned} T^T\mathcal{Q}T &= \mathcal{D}^{-\frac{1}{2}} V^T L_c^T \mathcal{Q} L_c V \mathcal{D}^{-\frac{1}{2}} \\ &= \mathcal{D}^{-\frac{1}{2}} V^T L_C^T L_o L_o^T L_c V \mathcal{D}^{-\frac{1}{2}} \\ &= \mathcal{D}. \end{aligned}$$

3.2 Interpolation or (moment matching) methods

We first give the following definition.

Definition 3.1 (Moments) *Given the system* Σ*, its associated transfer function* $H(\omega) = C(A - \omega I)^{-1}B$ *can be decomposed through a Laurent series expansion around a given* $\sigma \in \mathbb{C}$ *(shift point), as follows*

$$H(\omega) = \sum_{i=0}^{\infty} \eta_i^{(\sigma)} \frac{(\omega - \sigma)^i}{i!}, \tag{3.4}$$

where $\eta_i^{(\sigma)} \in \mathbb{R}^{p\times p}$ *is called the i-th moments at* σ *associated to the system and defined*

as follows

$$\eta_i^{(\sigma)} = C(A - \sigma I_n)^{-(i+1)}B = (-1)^i \frac{d^i}{d\omega^i} H(\omega)|_{\omega=\sigma}. \tag{3.5}$$

In the case where $\sigma = \infty$ the moments are called Markov parameters and are given by

$$\eta_i = CA^iB.$$

3.2.1 Rational interpolation

Given Σ as in (1.3), the interpolation problem is to find a reduced model Σ_m, such that $H_m(s)$ interpolates $H(s)$ in certain number of its moments at σ in the complex plane. A more general definition of approximation by moment matching is related to rational interpolation. By rational interpolation we mean that the reduced order system matches the moments of the original system at multiple interpolation points. The goal here is to produce a low order transfer function, $H_m(s)$, that approximates the large order transfer function, we want $H_m(\omega) \approx H(\omega)$ with $m \ll n$. Various model reduction methods for MIMO systems have been explored these last years. Some of them are based on rational Krylov subspace Interpolation methods. One could select a set of points $\{\sigma_i\}_{i=1}^m \subset \mathbb{C}$ and then seek a reduced order transfer function, $H_m(\omega)$, such that

$$H_m(\sigma_i) = H(\sigma_i), \qquad for \quad i = 1, ..., m$$

see [6, 7, 12, 13]. The rational Krylov subspace is defined as :

$$\mathcal{K}_m(A, b)^R = Range\{(\sigma_1 I - A)^{-1}b, ..., (\sigma_m I - A)^{-1}b\},$$

where b is a vector in $\mathbb{R}^n$ and $\sigma_1, ..., \sigma_m$ are some selected complex shifts. The block rational Krylov subspace is defined as :

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

where B is block of size $n \times p$. In the following we give a theorem that was presented in [22] for SISO systems, and is extended to the MIMO case in [18]. It shows how to construct the bi-orthogonal bases $\mathbb{V}_m$ and $\mathbb{W}_m$ so that the transfer

function $H_m(\omega)$ of the reduced order model Σ_m in (1.5) has to interpolate the transfer function $H(\omega)$ of the original system Σ in (1.3) and its first derivative at the interpolation points $\{\sigma_i\}_{i=1}^m$.

Theorem 3.1 *Let $\{\sigma_i\}_{i=1}^m$ be such that $(\sigma_i I - A)$ are invertible for $i = 1,...,m$. Let $\mathbb{V}_m = [V_1,...,V_m]$, $\mathbb{W}_m = [W_1,...,W_m] \in \mathbb{R}^{n\times mp}$ be two matrices with full-rank obtained as follows :*

$$Range\{V_1,...,V_m\} = Range\Big\{(\sigma_1 I - A)^{-1}B,...,(\sigma_m I - A)^{-1}B\Big\},$$

$$Range\{W_1,...,W_m\} = Range\{(\sigma_1 I - A)^{-T}C^T,...,(\sigma_m I - A)^{-T}C^T\},$$

where $\mathbb{W}_m^T\mathbb{V}_m = I_{mp}$. Then, the reduced order transfer function $H_m(\omega) = C_m(\omega I_m - A_m)^{-1}B_m$ obtained in (1.6) *interpolates $H(\omega)$ and its first derivative at $\{\sigma_i\}_{i=1}^m$.*

The question then is how to choose the interpolation points σ_i so as to have a good approximant of the initial model (in other words the reduced order model that solve the following problem) :

$$H_m(\omega) = \mathbf{arg}\left(\min_{\widetilde{H}(\omega)\ stable} \mathcal{J}_{\mathcal{H}_2}(\widetilde{H}(\omega))\right),$$

where,

$$\mathcal{J}_{\mathcal{H}_2}(\widetilde{H}(\omega)) = \|H(\omega) - \tilde{H}(\omega)\|_{\mathcal{H}_2}.$$

This problem has been partially solved in [24] by proposing an iterative procedure to achieve a reduced model satisfying the optimal approximation. The idea is to select the interpolation points as being the mirror images of the eigenvalues of the matrix A_m at previous iteration (see Algorithm 5, step 7).

Algorithm 5 (Iterative Rational Krylov Algorithm IRKA) .

1. *Inputs : $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{p\times n}$, an initial selection of interpolation points $\sigma^{(0)} = \{\sigma_1^{(0)},..,\sigma_m^{(0)}\} \in \mathbb{C}^m$.*
2. *Construct $\mathbb{V}_m = [(\sigma_1^{(0)}I_n - A)^{-1}B,...,(\sigma_m^{(0)}I_n - A)^{-1}B]$.*
3. *Construct $\mathbb{W}_m = [(\sigma_1^{(0)}I_n - A^T)^{-1}C^T,...,(\sigma_m^{(0)}I_n - A^T)^{-1}C^T]$.*
4. *Set $\mathbb{W}_m = \mathbb{W}_m(\mathbb{W}_m^T\mathbb{V}_m)^{-T}$.*

5. *While the algorithm has not converged.*
6. $A_m = \mathbb{W}_m^T A \mathbb{V}_m$.
7. $i \longleftarrow i+1$
8. $\sigma_j^{(i)} = -\lambda_j(A_m)$.
9. *Construct* $\mathbb{V}_m = [(\sigma_1^{(i)} I_n - A)^{-1} B, ..., (\sigma_m^{(i)} I_n - A)^{-1} B]$.
10. *Construct* $\mathbb{W}_m = [(\sigma_1^{(i)} I_n - A^T)^{-1} C^T, ..., (\sigma_m^{(i)} I_n - A^T)^{-1} C^T]$.
11. *Set* $\mathbb{W}_m = \mathbb{W}_m (\mathbb{W}_m^T \mathbb{V}_m)^{-T}$.
12. *End.*
13. *Output :* $A_m = \mathbb{W}_m^T A \mathbb{V}_m$, $B_m = \mathbb{W}_m^T B$, $C_m = C \mathbb{V}_m$.

- Steps 4 and 11 serve to make $\mathbb{V}_m$ and $\mathbb{W}_m$ biorthogonal, i.e. $\mathbb{W}_m^T \mathbb{W}_m = I_{mp}$.
- A possible stopping test may be the stagnation of interpolation points $\sigma_j^{(i)}$.

Although it is extremely rare to obtain unstable models, this technique does not guarantee the stability of the reduced model, and no bound on the approximation error has yet been proposed. Although the theory of rational interpolation applies indifferently to the cases of SISO and MIMO systems, the latter pose problems in practice insofar as the construction of the projection matrices $\mathbb{V}_m$ and $\mathbb{W}_m$ must be done by block and in a numerically stable way.This is very complex in practice where rank losses are observed, it is then preferable to consider the framework of the tangential interpolation which is more adapted to the case of MIMO systems.

3.2.2 Tangential interpolation

The difficulty of applying rational interpolation to the case of MIMO systems comes from the fact that the interpolation conditions are not restrictive enough. With tangential interpolation, in addition to the interpolation points, interpolation directions are provided to avoid this problem. Hence the problem of the tangential interpolation :

Problem 2

Given a full-order model (1.3) and the system of matrices A, B, and C and given :

- Left interpolation points $\{\mu_i\}_{i=1}^m \subset \mathbb{C}$ and left tangent directions $\{l_i\}_{i=1}^m \subset \mathbb{C}^p$.
- Right interpolation points $\{\sigma_i\}_{i=1}^m \subset \mathbb{C}$ and right tangent directions $\{r_i\}_{i=1}^m \subset \mathbb{C}^p$.

The problem is to find a reduced-order model (1.5) through identification of reduced system matrices A_m, B_m, and C_m such that the associated transfer function, H_m in (1.6) is a tangential interpolant to H, i.e.

$$l_i^T H_m(\mu_i) = l_i^T H(\mu_i) \quad \text{and} \quad H_m(\sigma_i) r_i = H(\sigma_i) r_i, \qquad for\ i = 1, ..., m. \tag{3.6}$$

The interpolation points and tangent directions are selected to realize the conditions of optimality of the first order that are expressed here through Theorem 3.2. They were formulated by Van Dooren and al[24].

Theorem 3.2 *Let λ_i, d_i et $g_i, i = 1, ..., m$, be the eigenvalues, the right and left eigenvectors of the reduced model Σ_m respectively :*

$$A_m d_i = \lambda_i d_i \qquad g_i^T A_m = \lambda_i g_i^T. \tag{3.7}$$

The left and right tangential directions l_i and r_i can then be defined as :

$$l_i = C_m d_i \qquad r_i^T = g_i^T B_m. \tag{3.8}$$

The following equations are satisfied, if and only if $H_m(s)$ has only single poles and minimizes $\mathcal{J}_{\mathcal{H}_2}$:

$$\begin{aligned} l_i^T H(-\lambda_i) &= l_i^T H_m(-\lambda_i) \\ H(-\lambda_i) r_i &= H_m(-\lambda_i) r_i \\ l_i^T H'(-\lambda_i) r_i &= l_i^T H_m'(-\lambda_i) r_i \end{aligned} \tag{3.9}$$

We want to interpolate H without ever computing the quantities to be matched since these numbers are numerically ill-conditioned, as provided in [16] for single-input/single-output dynamical systems. This can be reached by using Petrov-Galerkin projections with carefully choosing the projection subspaces. The tangential Krylov subspace is defined as follows :

$$\mathcal{K}_m^T(A, B) = Range\{(\sigma_1 I - A)^{-1} B r_1, ..., (\sigma_m I - A)^{-1} B r_m\}. \tag{3.10}$$

We give the following result presented in [2].

Theorem 3.3 *Let $\sigma, \mu \in \mathbb{R}$ be such that $A_s = (A - sI)$ is invertible for $s = \sigma$ and $s = \mu$. Let $V_m, W_m \in \mathbb{R}^{n\times m}$ be two matrices with full-rank. Let r, $l \in \mathbb{R}^p$ be some chosen tangential vectors. Then we have the following properties :*

1. *If $(A - \sigma I)^{-1}Br \in Range\{V_m\}$, then $\quad H(\sigma)r = H_m(\sigma)r$.*
2. *If $(A - \mu I)^{-T}C^T l \in Range\{W_m\}$, then $\quad l^T H(\mu) = l^T H_m(\mu)$.*
3. *If both (1) and (2) hold, and if $\sigma = \mu$, then $\quad l^T H'(\sigma)r = l^T H'_m(\sigma)r$.*

In the following we present, the $Iterative\ Tangential\ Interpolation\ Algorithm$ ($ITIA$) suggested by Van Dooren and al[24]. It takes the structure of Algorithm 5 a few points close :

Algorithm 6 (Iterative Tangential Interpolation Algorithm ITIA) .

1. *Inputs : $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{p\times n}$, an initial selection of interpolation points and tangent directions, $r^{(0)} = \{r_1^{(0)}, ..., r_m^{(0)}\} \in \mathbb{R}^{p\times m}$, $l^{(0)} = \{l_1^{(0)}, ..., l_m^{(0)}\} \in \mathbb{R}^{p\times m}$ $\sigma^{(0)} = \{\sigma_1^{(0)}, ..., \sigma_m^{(0)}\} \in \mathbb{R}^m$.*
2. *Construct $V_m = \left[(\sigma_1^{(0)} I_n - A)^{-1}Br_1^{(0)}, ..., (\sigma_m^{(0)} I_n - A)^{-1}Br_m^{(0)}\right]$.*
3. *Construct $W_m = \left[(\sigma_1^{(I)} I_n - A^T)^{-1}C^T l_1^{(0)}, ..., (\sigma_m^{(0)} I_n - A^T)^{-1}C^T l_m^{(0)}\right]$.*
4. *Set $W_m = W_m(W_m^T V_m)^{-T}$.*
5. *While the algorithm has not converged.*
6. *$A_m = W_m^T A V_m$.*
7. *$i \longleftarrow i + 1$*
8. *$\sigma_j^{(i)} = -\lambda_j(A_m)$.*
9. *Compute tangential directions as in (3.8).*
10. *Construct $V_m = \left[(\sigma_1^{(i)} I_n - A)^{-1}r_1^{(i)}, ..., (\sigma_r^{(i)} I_n - A)^{-1}Br_m^{(i)}\right]$.*
11. *Construct $W_m = \left[(\sigma_1^{(i)} I_n - A^T)^{-1}C^T l_1^{(i)}, ..., (\sigma_m^{(i)} I_n - A^T)^{-1}C^T l_m^{(i)}\right]$.*
12. *Set $W_m = W_m(W_m^T V_m)^{-T}$.*
13. *End.*

14. Outputs : $A_m = W_m^T A V_m$,$B_m = W_m^T B$, $C_m = C V_m$.

• Steps 2, 3, 10 and 11 of constructing the bases V_m and W_m now include the tangential directions.
• At each iteration, the new tangential directions are calculated as in the equations (3.7) and (3.8).

Tangential interpolation applies well to MIMO systems, but it has the same drawbacks as rational interpolation, namely the absence of a guarantee on the stability of the reduced model and the absence of a bound on the norm of the error. To overcome this, work has been done to combine approaches based on interpolation and approaches based on Lyapunov equations (such as balanced truncation) to extract the benefits. These are the mixed methods or SVD-Krylov methods (see e.g.Gugercin [3]).

3.3 Mixed methods or (SVD-Krylov methods)

Many procedures attempt to couple the advantages of balanced truncation with those of the Krylov subspaces (see Antoulas [7]), but here only those directly derived from interpolation methods are discussed. They are based on using one of the Gramians ($\mathcal{P}$ or $\mathcal{Q}$) for the construction of one of the projectors. The other is generated in the same way as before, i.e by using a Krylov subspace. The first algorithm using this approach was proposed by Gugercin[4] and takes the form of the IRKA algorithm, it is called *Iterative SVD-Rational Krylov Algorithm* (ISRKA). The only difference with IRKA lies in the construction of one of the projectors. For example, if the Gramian $\mathcal{Q}$ is used, the projector W_m is constructed as follows :

$$W_m = \mathcal{Q} V_m (V_m^T \mathcal{Q} V_m)^{-1} \tag{3.11}$$

The use of a Gramians can guarantee the stability of the reduced model and also to make the algorithm directly applicable to $MISO$ and $SIMO$ systems according to the Gramians used. For MIMO systems, on the other hand, it is better to use tangential interpolation to generate the first projector, it is for this purpose that the algorithm *Iterative SVD Tangential Interpolation Algorithm*

(ISTIA) has been proposed by Poussot-Vassal [15] (see Algorithm 7). Because of the use of a Gramian to form the second projector, only a portion of the first order optimality conditions can be satisfied. Thus, the SVD-Krylov methods are in theory less efficient than the methods based only on interpolation, but they guarantee the stability of the reduced model and are more robust compared to the choice of the initial interpolation points $\sigma^{(0)}$.

Algorithm 7 (Iterative SVD Tangential Interpolation Algorithm ISTIA) .

1. *Inputs :* $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{p\times n}$, $\sigma^{(0)} = \{\sigma_1^{(0)}, ..., \sigma_m^{(0)}\} \in \mathbb{C}^m$, $\{r_1, ..., r_m\} \in \mathbb{R}^{p\times m}$.
2. *Compute the observability Gramians* $\mathcal{Q}$.
3. *Construct* $V_m = [(\sigma_1^{(0)} I_n - A)^{-1} B r_1^{(0)}, ..., (\sigma_m^{(0)} I_n - A)^{-1} B r_m^{(0)}]$.
4. *Set* $W_m = \mathcal{Q} V_m (V_m^T \mathcal{Q} V_m)^{-1}$.
5. *While the algorithm has not converged.*
6. $A_m = W_m^T A V_m$,$B_m = W_m^T B$, $C_m = C V_m$.
7. $i \longleftarrow i + 1$
8. *Compute the interpolation points* $\sigma_j^{(i)}$ *and tangential directions* $r_j^{(i)}$.
9. $V_m = [(\sigma_1^{(i)} I_n - A)^{-1} B r_1^{(i)}, ..., (\sigma_m^{(i)} I_n - A)^{-1} B r_m^{(i)}]$.
10. *Set* $W_m = \mathcal{Q} V_m (V_m^T \mathcal{Q} V_m)^{-1}$.
11. *End*
12. *Outputs :* $A_m = W_m^T A V_m$, $B_m = W_m^T B$, $C_m = C V_m$.

Troisième partie

Tangential Arnoldi methods for model reduction

Chapitre 4

An adaptive block tangential method for MIMO dynamical systems

In this Chapter, 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.

4.1 Introduction

Various model reduction methods for MIMO systems, such as Padé approximation [16, 40], balanced truncation [34], optimal Hankel norm [19, 21] 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 [11, 10, 28]. Those methods use block Krylov subspace

$$\mathbb{K}_m(A, B) = Range\{B, AB, ..., A^{m-1}B\},$$

or rational block Krylov subspace defined in Chapter 3,

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

where $\sigma_1, ..., \sigma_m$ 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 $\mathbb{K}_m(A,B)$ or $\mathbb{K}_m^R(A,B)$, see [5, 17, 27].

In the present chapter we considered another approach based on a work of Druskin and Simoncini [13], as well as some theory in [2], by using the tangential Krylov subspaces. For this approach, we considered the tangential directions as blocks of $p \times s$ size with $s \leq p$ and we used the block Arnoldi procedure to generate orthogonormal bases of the desired projection subspaces,

$$\mathbb{K}_m^T = Range\{(\sigma_1 I - A)^{-1}BR_1, ..., (\sigma_m I - A)^{-1}BR_m\}. \tag{4.1}$$

The computation of the parameters (σ_i, R_i) will be done in an adaptive way. Throughout the thesis we use the following notations : The field of values of A is defined by

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

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

4.2 The block tangential Arnoldi-like 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

$$(\omega I_n - A)X = B. \tag{4.2}$$

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 (4.2). This will be done as follows : Given a system of matrices $\{V_1, \ldots, V_m\}$ where $V_i \in \mathbb{R}^{n \times s}$, the approximate solution X_m of X is computed, at step m, such that

$$X_m^i \in Range\{V_1, ..., V_m\}, \tag{4.3}$$

and

$$R_B^i(\omega) = \perp Range\{V_1, ..., V_m\},\ i = 1, ..., p \tag{4.4}$$

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 $\mathbb{V}_m = [V_1, \ldots, V_m]$, then from (4.3) and (4.4) , 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 expressd as

$$R_B(\omega) = B - (\omega I_n - A)\mathbb{V}_m(\omega I_{ms} - A_m)^{-1}\mathbb{V}_m^T B. \tag{4.5}$$

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.

4.2.1 Block tangential Arnoldi algorithm

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

$$Range\{V_1, ..., V_m\} = Range\Big\{(\sigma_1 I_n - A)^{-1}BR_1, ..., (\sigma_m I_n - A)^{-1}BR_m\Big\}, \tag{4.6}$$

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 8 (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 :* $\mathbb{V}_{m+1} = \{V_1, ..., V_{m+1}\}$.

1. *Set* $\widetilde{V}_1 = (\sigma_1 I_n - A)^{-1}BR_1$.
2. *Compute* $\widetilde{V}_1 = V_1 H_{1,0}$, *QR decomposition.*

3. *Initialize :* $\mathbb{V}_1 = [V_1]$.
4. *For* $j = 1,...,m$
5. *If* $\sigma_{j+1} \neq \infty$, $\widetilde{V}_{j+1} = (\sigma_{j+1} I_n - A)^{-1} B R_{j+1}$, *else* $\widetilde{V}_{j+1} = ABR_{j+1}$.
6. *For* $i = 1,...,j$
   - $H_{i,j} = V_i^T \widetilde{V}_{j+1}$,
   - $\widetilde{V}_{j+1} = \widetilde{V}_{j+1} - V_i H_{i,j}$,
7. *End.*
8. $\widetilde{V}_{j+1} = V_{j+1} H_{j+1,j}$, *QR Decomposition.*
9. $\mathbb{V}_{j+1} = \left[\mathbb{V}_j,\ V_{j+1}\right]$,
10. *End*

In Algorithm 8, 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}, $j = 1,...,m$ and we initialize the subsequent tangential subspace by setting $\widetilde{V}_{j+1} = (\sigma_{j+1} I_n - A)^{-1} B R_{j+1}$ if σ_{j+1} is finite and $\widetilde{V}_{j+1} = ABR_{j+1}$ if $\sigma_{j+1} = \infty$. The matrices $H_{i,j}$ constructed in Step 6 are of size $s \times s$ and they are used to construct the block upper Hessenberg matrix $\widetilde{\mathbb{H}}_m = \left[\widetilde{\mathbb{H}}^{(1)},...,\widetilde{\mathbb{H}}^{(m)}\right] \in \mathbb{R}^{(m+1)s \times ms}$, where

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

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

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

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 $\widetilde{\mathbb{H}}_m$ by deleting its last row

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

The next proposition gives some algebraic properties corresponding to the matrices derived from Algorithm 8.

Proposition 4.1 *Let $\mathbb{V}_{m+1}$ be the orthonormal matrix of $\mathbb{R}^{n\times(m+1)s}$ constructed by Algorithm 8. Then we have the following relations*

$$A\mathbb{V}_{m+1}\widetilde{\mathbb{H}}_m = \mathbb{V}_{m+1}\widetilde{\mathbb{K}}_m - B\widetilde{\mathbb{R}}_{m+1}, \tag{4.7}$$

$$A_m = \mathbb{V}_m^T A\mathbb{V}_m = \left[\mathbb{K}_m - B_m\widetilde{\mathbb{R}}_{m+1} - \mathbb{V}_m^T AV_{m+1}H_{m+1,m}(e_m^T \otimes I_s)\right]\mathbb{H}_m^{-1}, \tag{4.8}$$

and

$$\mathbb{T}_{m+1} = \mathbb{V}_{m+1}\mathbb{G}_{m+1}, \tag{4.9}$$

where $\mathbb{K}_m$ is the $ms \times ms$ matrix obtained from $\widetilde{\mathbb{K}}_m = \widetilde{\mathbb{H}}_m(D_m \otimes I_s)$, by deleting its last row, $D_m = Diag\{\sigma_2, ..., \sigma_{m+1}\}$, $\mathbb{T}_{m+1} = \left[(\sigma_1 I - A)^{-1}BR_1, ..., (\sigma_{m+1}I - A)^{-1}BR_{m+1}\right]$, $\widetilde{\mathbb{R}}_{m+1} = [R_2, ..., R_{m+1}]$, and $\mathbb{G}_{m+1} = \left[\widetilde{\mathbb{H}}^{(0)} \;\; \widetilde{\mathbb{H}}_m\right]$ is a block upper triangular matrix of $(m+1)s \times (m+1)s$. The matrix $\mathbb{H}_m$ is assumed to be non singular.

Proof : From Algorithm 8, we have

$$V_{j+1}H_{j+1,j} = (\sigma_{j+1}I_n - A)^{-1}BR_{j+1} - \sum_{i=1}^{j} V_i H_{i,j} \qquad j = 1, ..., m. \tag{4.10}$$

Multiplying (4.10) on the left by $(\sigma_{j+1}I_n - A)$ and re-arranging terms, we get

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

which gives

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

also be written as

$$A\mathbb{V}_{m+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} = \sigma_{j+1}\mathbb{V}_{j+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} - BR_{j+1}, \qquad j = 1,...,m, \tag{4.11}$$

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

$$A\mathbb{V}_{m+1}\widetilde{\mathbb{H}}^{(j)} = \sigma_{j+1}\mathbb{V}_{j+1}\widetilde{\mathbb{H}}^{(j)} - BR_{j+1}, \qquad j = 1,...,m. \tag{4.12}$$

Therefore, we can deduce from (4.12), the following expression

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

which ends the proof of (4.7).
For the relation (4.8), we have from (4.7),

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

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

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

Therefore

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

For the proof of (4.9), we first use (4.10) to obtain

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

which gives

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

It follows that

$$\mathbb{V}_{m+1} \left[\widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)} \right] = \left[(\sigma_2 I_n - A)^{-1} B R_2, ..., (\sigma_{m+1} I_n - A)^{-1} B R_{m+1} \right],$$

Since $V_1 H_{1,0} = (\sigma_1 I_n - A)^{-1} B R_1$, we have

$$\mathbb{V}_{m+1} \left[\widetilde{\mathbb{H}}^{(0)}, \widetilde{\mathbb{H}}^{(1)}, ..., \widetilde{\mathbb{H}}^{(m)} \right] = \left[(\sigma_1 I_n - A)^{-1} B R_1, (\sigma_2 I_n - A)^{-1} B R_2, ..., (\sigma_{m+1} I_n - A)^{-1} B R_{m+1} \right],$$

which ends the proof of (4.9).

Theorem 4.1 *Let $\sigma \in \mathbb{C}$ be such that $(\sigma I - A)$ is invertible. Let $\mathbb{V}_m = [V_1, ..., V_m]$ have full-rank, where the $V_i \in \mathbb{R}^{n \times s}$. Let $R = [r_1, ..., r_s] \in \mathbb{R}^{p \times s}$ be a chosen tangential matrix direction. Then,*

1. *If $(\sigma I - A)^{-1} B r_i \in Range\{V_1, ..., V_m\}$ for $i = 1, ..., s$, 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 [2] 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

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

Observe that

$$H(\omega) - H_m(\omega) = C(\omega I - A)^{-1}(I - \mathcal{Q}_m(\omega))(\omega I - A)(I - \mathcal{P}_m(\omega))(\omega I - A)^{-1}B. \quad (4.13)$$

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 $\mathcal{V}_m^{\perp} = Ker(\mathcal{Q}_m(\omega)) = Range(I - \mathcal{Q}_m(\omega))$. 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 (4.13) at $s = \sigma + \varepsilon$, multiplying by r_j^T and r_i, from the left and the right respectively, for $i, j = 1, ..., s$, 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

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

which proves the second assertion.

Proposition 4.2 *Let $R_B(\omega)$ be the residual $R_B(\omega) = B - (\omega I_n - A)\mathbb{V}_m Q_m(\omega)$ as given in (4.5), where $Q_m(\omega) = (\omega I_{ms} - A_m)^{-1}\mathbb{V}_m^T B$. We have the following new expression given by*

$$R_B(\omega) = (I_n - \mathbb{V}_m\mathbb{V}_m^T)B + (A\mathbb{V}_m - \mathbb{V}_m A_m)Q_m(\omega). \tag{4.14}$$

Proof : We have

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

which proves (4.14).

Proposition 4.3 *Let $\mathbb{T}_m = \left[(A-\sigma_1 I)^{-1}BR_1, ..., (A-\sigma_m I)^{-1}BR_m\right] = \mathbb{V}_m\mathbb{G}_m$, where $\mathbb{G}_m$ and $\mathbb{V}_m$ are obtained by the Block Tangential Arnoldi Algorithm (BTAA). Let $\mathbb{R}_m = [R_1, ..., R_m]$, then*

$$A\mathbb{V}_m - \mathbb{V}_m A_m = -\left(I_n - \mathbb{V}_m\mathbb{V}_m^T\right)B\mathbb{R}_m\mathbb{G}_m^{-1}, \tag{4.15}$$

and

$$R_B(\omega) = \left(I_n - \mathbb{V}_m\mathbb{V}_m^T\right)B\left(I_p - \mathbb{R}_m\mathbb{G}_m^{-1}Q_m(\omega)\right). \tag{4.16}$$

Proof : Let $\Sigma_m = [diag(\sigma_1, ..., \sigma_m) \otimes I_s]$, 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 $A_m = \mathbb{V}_m^T A\mathbb{V}_m = \mathbb{G}_m^{-T}\mathbb{T}_m^T A\mathbb{T}_m\tilde{G}_m^{-1}$, 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 (4.16) will be used in the next section in order to reduce the cost when computing the residual.

4.3 Choice of the interpolation points and tangent directions

In this section we use an adaptive strategy for choosing the interpolation points and tangent directions. This technique was first proposed in [12] to choose the shifts for the rational Krylov subspaces. The iterative rational Krylov algorithm (IRKA) was proposed in [2], 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 $\sigma_i = -\lambda_i(A_m)$, $i = 1, ..., m$. In [43] 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 defined in (3.8). In this chapter we use an adaptive approach, inspired by the work given in [14]. Our choice of this approach, is justified by the main disadvantage of the iterative methods, that requires the construction of many Krylov subspaces, which will not be used in the final model, only the last subspace is used.
In the adaptive approach, we seek to extend our subspace

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

by a new block defined by

$$V_{m+1} = (\sigma_{m+1} I_n - A)^{-1} BR_{m+1}, \tag{4.17}$$

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

$$(R_{m+1}, \sigma_{m+1}) = arg \max_{\substack{\omega \in S_m \\ R \in \mathbb{R}^{s\times s}, \|R\|_2 = 1}} \|R_B(\omega)R\|_2. \tag{4.18}$$

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

Now we explain how to solve the problem (4.18). First we compute the interpolation point σ_{m+1}, by maximizing the residual norm on the convex hull S_m, i.e we solve the following problem,

$$\sigma_{m+1} \quad = arg \max_{\omega \in S_m} \|R_B(\omega)\|_2. \tag{4.19}$$

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 (4.18) at $\omega = \sigma_{m+1}$

$$R_{m+1} = arg \max_{R \in \mathbb{R}^{s\times s}, \|R\|=1} \|R_B(\sigma_{m+1})R\|_2 . \tag{4.20}$$

Notice that the tangential matrix direction $R_{m+1} = [r_1^{(m+1)}, ..., r_s^{(m+1)}]$, 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 (4.5) of the residual given in Proposition 4.2

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

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

$$\|R_B(\omega)\|_2 = \left\| L \left(I_p - \mathbb{R}_m \mathbb{G}_m^{-1} Q_m(\omega)\right) \right\|_2 . \tag{4.21}$$

This means that, solving (4.18) 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 9 (Adaptive Block Tangential Arnoldi Algorithm ABTAA) .
– *Given A, B, C m,* $\omega_0^{(1)} \in \mathbb{R}$, $R_1 \in \mathbb{R}^{s\times s}$.
– *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* $\sigma_1 = \omega_0^{(1)}$, $\widetilde{V}_1 = (\sigma_1 I_n - A)^{-1} B R_1$.
- *Compute* $\widetilde{V}_1 = V_1 H_{1,0}$ *(QR decomposition), initialize :* $\mathbb{V}_1 = [V_1]$.
- *For* $k = 1 : m-1$
  1. *If* $\bar{\sigma}_{k-1} \neq \sigma_k \in \mathbb{C}$ *then* $\sigma_{k+1} = \bar{\sigma}_k$ *else compute* $\{\lambda_1, ..., \lambda_{ks}\}$ *eigenvalues of* A_k.
  2. *Determine* S_k, *the convex hull of* $\{-\lambda_1, ..., -\lambda_{ks}, \omega_0^{(1)}, \bar{\omega}_0^{(1)}\}$ *and solve (4.19).*
  3. *Compute the right vector* R_{k+1} *by solving (4.20).*
  4. *If* $\sigma_{k+1} \neq \infty$, $\widetilde{V}_{k+1} = (\sigma_{k+1} I_n - A)^{-1} B R_{k+1}$ *else* $\widetilde{V}_{k+1} = ABR_{k+1}$.
  5. *For i = 1,...,k*
     - $H_{i,k} = V_i^T \widetilde{V}_{k+1}$,
     - $\widetilde{V}_{k+1} = \widetilde{V}_{k+1} - V_i H_{i,k}$,
  6. *End.*
  7. $\widetilde{V}_{k+1} = V_{k+1} H_{k+1,k}$, *(QR Decomposition).*
  8. $\mathbb{V}_{k+1} = [\mathbb{V}_k,\ V_{k+1}]$.
- *End*

Algorithm 9, 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 4.4 *Let* $\mathbb{V}_k = [V_1, ..., V_k]$, *be the orthonormal matrix obtained by Algorithm 9 at the iteration k, then setting* $\mathcal{M}_k = Range\{V_1, ..., V_k, (\sigma_{k+1} I_n - A)^{-1} B R_{k+1}\}$, *we have*

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

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,

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

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

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

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

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

We observe that,

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

Using the fact that the residual $R_B(\omega)$ is orthogonal to $[V_1,..,V_k]$, 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

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

which gives for $1 \leq i, j \leq s$,

$$(R_B(\sigma_{k+1})R_{k+1})^T Y = \left[(R_B(\sigma_{k+1})r_i^{(k+1)})^T(\sigma_{k+1}I_n - A)^{-1}R_B(\sigma_{k+1})r_j^{(k+1)}\right] \neq 0.$$

In fact we know that $\mathcal{W}(\sigma_{k+1}I_n - A)^{-1} \subset \mathbb{C}^+$. Hence $Y \neq 0$, which proves the second assertion.

Proposition 4.5 *Let* $A_{m+1} = \mathbb{V}_{m+1}^T A\mathbb{V}_{m+1} = [a_{:,1},...,a_{:,m+1}]$, *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* $\widetilde{\mathbb{H}}_m = \left[\widetilde{\mathbb{H}}^{(1)},...,\widetilde{\mathbb{H}}^{(m)}\right]$ *is the upper Hessenberg matrix obtained from Algorithm 9. The, for* $j = 1,...,m$, *we have*

$$a_{:,j+1} = \left[\sigma_{j+1}\widetilde{\mathbb{H}}^{(j)} - \left[a_{:,1},...a_{:,j}\right]\widetilde{\mathbb{H}}^{(j)}_{1:js,:} - B_{m+1}R_{j+1}\right]H^{-1}_{j+1,j}. \tag{4.22}$$

Proof : We have from Algorithm 9

$$V_{j+1}H_{j+1,j} = (\sigma_{j+1}I_n - A)^{-1}BR_{j+1} - \sum_{i=1}^{j} V_iH_{i,j} \qquad j = 1,...,m.$$

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

$$AV_{j+1}H_{j+1,j} = \sigma_{j+1}\sum_{i=1}^{j+1} V_iH_{i,j} - A\sum_{i=1}^{j} V_iH_{i,j} - BR_{j+1},$$

which gives the following relation

$$AV_{j+1}H_{j+1,j} = \sigma_{j+1}\mathbb{V}_{m+1}\widetilde{\mathbb{H}}^{(j)} - A\mathbb{V}_j\widetilde{\mathbb{H}}^{(j)}_{1:js,:} - BR_{j+1}.$$

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

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

which gives the desired result

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

Proposition 4.5 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 (4.8), we only need the inverse of small matrices $H_{j+1,j}$, $j = 1,...,m$.

4.4 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 [37] :

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

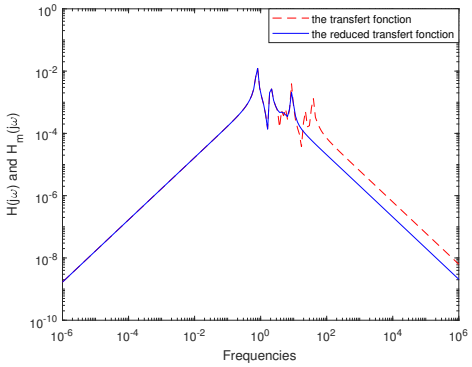
We used various matrices from LYAPACK and from the Oberwolfach collection [1]. These matrix tests are reported in Table 4.1 with different values of p and the used values of s.

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

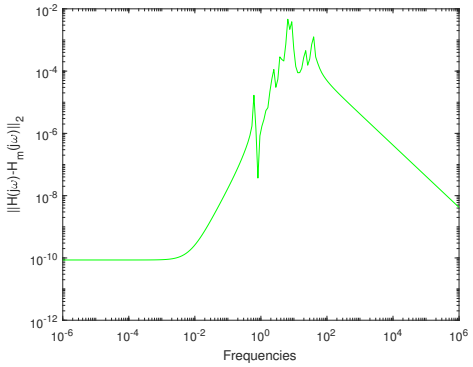
Tableau 4.1 – Matrix Tests

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 [35]. Figure 4.1a shows the singular values of the transfer function and its approximation. In Figure 4.1b, 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 $m = 15$.

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



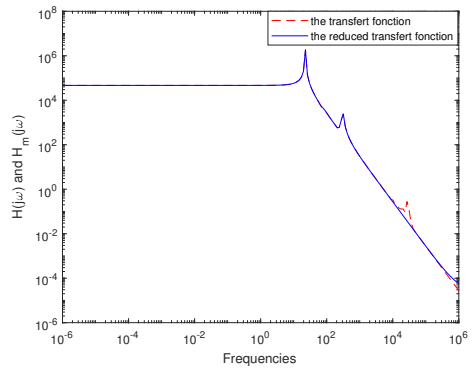
(a) Singular values vs frequencies.



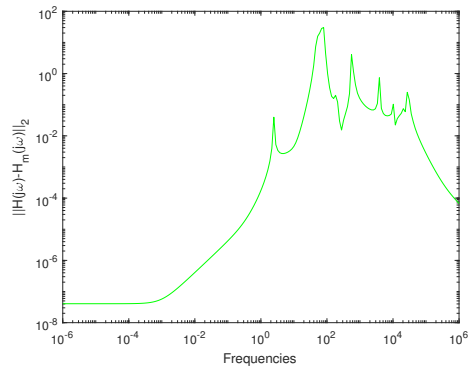
(b) Error-norms vs frequencies.

FIGURE 4.1 – The ISS model.

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 [26]. Figure 4.2a, 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 4.2b, we plotted the error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-6}, 10^{6}]$.



(a) Singular values vs frequencies.

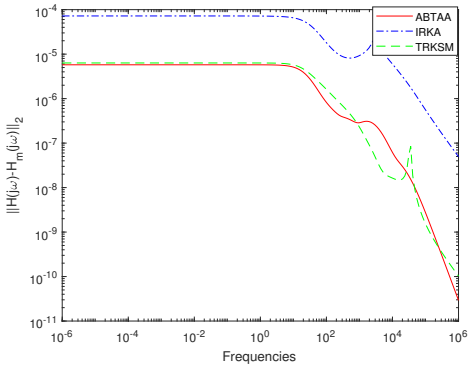


(b) Error-norms vs frequencies.

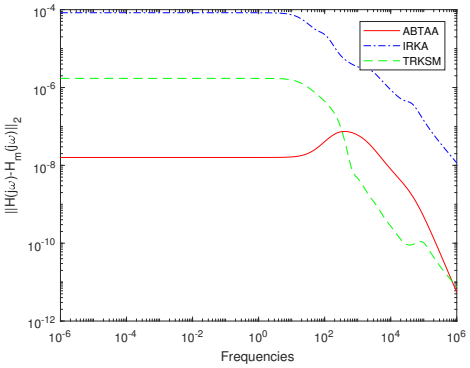
FIGURE 4.2 – The CDplayer model.

Example 3 : In this example we compared the ABTAA algorithm with the Iterative Rational Krylov Algorithm (IRKA [24]) and the adaptive tangential method represented by Druskin and Simonsini (TRKSM) see for more details [13]. We

used seven models : FDM, MNA_2, MNA_5, RAIL3113, RAIL20209, RAIL79841 and FLOW. The FDM model is described in paragraph(1.3.3) where in this example, the number of inner grid points in each direction was $n_0 = 100$ i.e the dimension of A is $n = n_0^2 = 10000$.



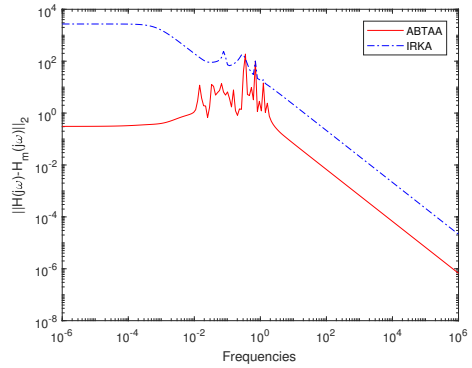
(a) ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line), $m = 20$.



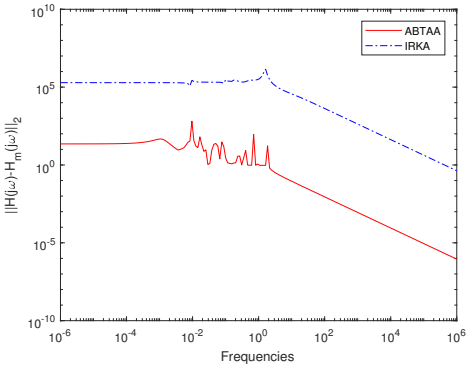
(b) ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line), $m = 30$.

FIGURE 4.3 – The FDM model.

The MNA_2 and MNA_5 models were obtained from NICONET (for more details see paragraph (1.3.2). Figures 4.4a and 4.4b 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 $m = 20$.



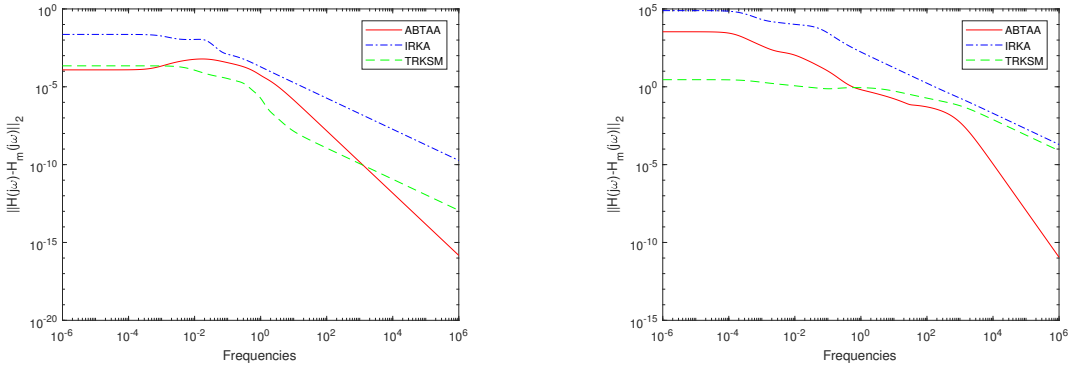
(a) The MNA_2 model, $m = 20$.



(b) The MNA_5 model, $m = 20$.

FIGURE 4.4 – ABTAA (solid line) & IRKA (dashed-dotted line).

The models RAIL3113 ($n = 3113$, $p = 6$) and Flow ($n = 9669$, $p = 5$) are from the Oberwolfach collection (see paragraph(1.3.4)). Figures 4.5a and 4.5b 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, IRKA : 15.21 seconds) and for the Flow model (ABTAA : 1.69 seconds, TRKSM : 7.39 seconds, IRKA : 42.50 seconds).



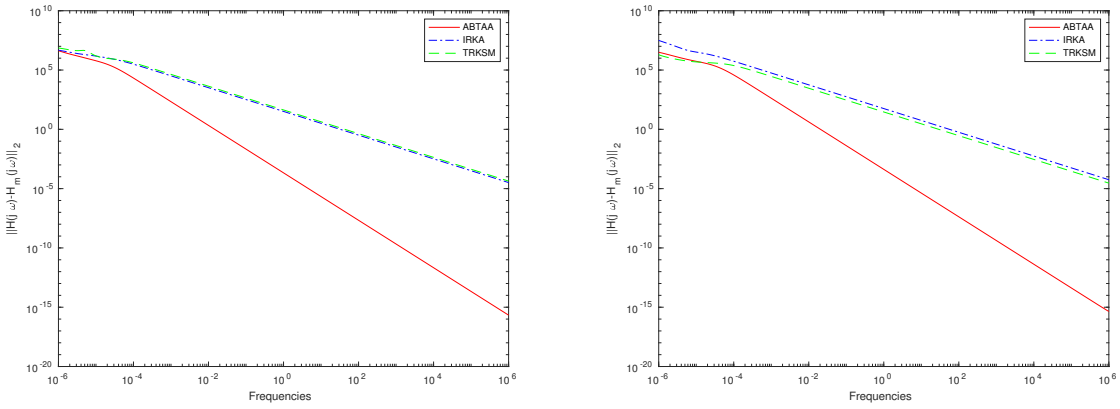
(a) The RAIL3113 model, $m = 20$.

(b) The Flow model, $m = 20$.

FIGURE 4.5 – ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line).

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 4.6a and 4.6b 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, IRKA : 44.08 seconds) and for RAIL79841 model is : (ABTAA : 32.69 seconds, TRKSM : 80.93 seconds, IRKA : 247.64 seconds).

Example 4 : In this example, we used the FDM model : ($n = 40.000$ and $n = 90.000$ with $p = 9$). In Table 4.2, we compared the execution times and the $\mathcal{H}_\infty$ 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 4.2, the cost of IRKA and TRKSM methods is much higher than the cost required with



(a) The RAIL20209 model. (b) The RAIL79841 model.

FIGURE 4.6 – ABTAA (solid line), IRKA (dashed-dotted line) & TRKSM (dashed-dashed line).

the adaptive block tangential Arnoldi method.

Model		ABTAA		IRKA		TRKSM	
		Time	Err-$\mathcal{H}_\infty$	Time	Err-$\mathcal{H}_\infty$	Time	Err-$\mathcal{H}_\infty$
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}

TABLEAU 4.2 – The computation time & the Err-$\mathcal{H}_\infty$ error-norm

4.5 Conclusion

In the present 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 matrix tangential Krylov subspaces using the block Arnoldi approach. The interpolation shifts and the tangential direc-

tions 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.

Chapitre 5

A computational global tangential Krylov subspace method

In this paper, we present another approach for model order reduction problems, with multiple inputs and multiple outputs (MIMO), named : Adaptive Global Tangential Arnoldi Algorithm (AGTAA). This method is based on a generalization of the global Arnoldi algorithm. The selection of the shifts and the tangent directions are done with an adaptive procedure as before. We give some algebraic properties and present some numerical examples to show the effectiveness of the proposed algorithm.

5.1 The global method

5.1.1 Definitions

We begin by recalling some notations that will be used later. We define the inner product $\langle Y, Z\rangle_F = tr(Y^T Z)$, where $tr(Y^T Z)$ denotes the trace of the matrix $Y^T Z$ such that $Y, Z \in \mathbb{R}^{n\times p}$. The associated norm is the Frobenius norm denoted by $\|.\|_F$. The matrix product $A \otimes B = [a_{i,j}B]$ denotes the well known Kronecker product of the matrices A and B which verifies the following properties :

1. $(A \otimes B)(C \otimes D) = (AC \otimes BD)$.
2. $(A \otimes B)^T = A^T \otimes B^T$.

3. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, if A and B are invertible.

We also use the matrix product $\diamond$ defined in [9] as follows.

Definition 5.1 *Let $A = [A_1, ..., A_m]$ and $B = [B_1, ..., B_l]$ be matrices of dimension $n \times mp$ and $n \times lp$, respectively, where A_i and B_j $(i = 1, ..., m\ j = 1, ..., l)$ are $\mathbb{R}^{n \times p}$. Then the $\mathbb{R}^{m \times l}$ matrix $A^T \diamond B$ is defined by :*

$$A^T \diamond B = \begin{pmatrix} \langle A_1, B_1 \rangle_F & \langle A_1, B_2 \rangle_F & \dots & \langle A_1, B_l \rangle_F \\ \langle A_2, B_1 \rangle_F & \langle A_2, B_2 \rangle_F & \dots & \langle A_2, B_l \rangle_F \\ \vdots & \vdots & \ddots & \vdots \\ \langle A_m, B_1 \rangle_F & \langle A_m, B_2 \rangle_F & \dots & \langle A_m, B_l \rangle_F \end{pmatrix}$$

Remark

1. If $p = 1$, then $A^T \diamond B = A^T B$.
2. If $p = 1$, $m = 1$ and $l = 1$, then setting $A = u \in \mathbb{R}^n$ and $B = v \in \mathbb{R}^n$, we have $A^T \diamond B = u^T v \in \mathbb{R}$.
3. The matrices $A = [A_1, ..., A_m]$ and $B = [B_1, ..., B_m]$ are F-biorthonormal if and only if $A^T \diamond B = I_m$, i.e.,

$$tr(A_i^T B_j) = \delta_{i,j} = \begin{cases} 0 & if \quad i \neq j \\ 1 & if \quad i = j \end{cases} \quad i, j = 1, ..., m. \tag{5.1}$$

4. If $X \in \mathbb{R}^{n \times p}$, then $X^T \diamond X = \|X\|_F^2$.

The following proposition gives some properties satisfied by the above product.

Proposition 5.1 *[9] Let $A,\ B,\ C \in \mathbb{R}^{n \times ps}$, $D \in \mathbb{R}^{n \times n}$, $L \in \mathbb{R}^{p \times p}$ and $\in \mathbb{R}$. Then we have,*

1. $(A + B)^T \diamond C = A^T \diamond C + B^T \diamond C$.
2. $A^T \diamond (B + C) = A^T \diamond B + A^T \diamond C$.
3. $(\alpha A)^T \diamond C = \alpha (A^T \diamond C)$.
4. $(A^T \diamond B)^T = B^T \diamond A$.
5. $(DA)^T \diamond B = A^T \diamond (D^T B)$.
6. $A^T \diamond (B(L \otimes I_s)) = (A^T \diamond B)L$.
7. $\|A^T \diamond B\|_F \leq \|A\|_F \|B\|_F$.

5.2 The global tangential Arnoldi method

In this section, we introduce the global tangential Arnoldi algorithm that allows us to compute an F-orthonormal basis of some specific matrix subspace and we derive some algebraic relations related to this algorithm.

5.2.1 The global tangential Arnoldi algorithm

The global tangential Arnoldi algorithm allows us to construct an F-orthonormal basis of the matrix subspace

$$\mathbb{V}_m = Span\{(\sigma_1 I_n - A)^{-1} BR_1, ..., (\sigma_m I_n - A)^{-1} BR_m\}, \tag{5.2}$$

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

Algorithm 10 (Global Tangential Arnoldi Algorithm GTAA) .
– *Inputs :* A, B, C, $\sigma = \{\sigma_i\}_{i=1}^{m+1}$, $R = \{R_i\}_{i=1}^{m+1}$, $R_i \in \mathbb{R}^{s\times s}$.
– *Output :* $\mathcal{V}_{m+1} = \{V_1, ..., V_{m+1}\}$.

- *Set* $\widetilde{V}_1 = (\sigma_1 I_n - A)^{-1} BR_1$.
- *Construct* V_1 *such that* $\|V_1\|_F = 1$.
- *Initialize :* $\mathcal{V}_1 = [V_1]$.
- *For* $j = 1,...,m$
  1. *If* $\sigma_{j+1} \neq \infty$, $\widetilde{V}_{j+1} = (\sigma_{j+1} I_n - A)^{-1} BR_{j+1}$, *else* $\widetilde{V}_{j+1} = ABR_{j+1}$.
  2. *For* $i = 1,...,j$
     - $h_{i,j} = tr(V_i^T \widetilde{V}_{j+1})$,
     - $\widetilde{V}_{j+1} = \widetilde{V}_{j+1} - h_{i,j} V_i$,

     End.
  3. $h_{j+1,j} = \|\widetilde{V}_{j+1}\|_F$,
  4. $V_{j+1} = \frac{\widetilde{V}_{j+1}}{h_{j+1,j}}$,
  5. $\mathcal{V}_{j+1} = \left[\mathcal{V}_j,\ V_{j+1}\right]$,
- *End*

In Algorithm 10, 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}, $j = 1,...,m$ and we initialize the subsequent tangential subspace by setting $\widetilde{V}_{j+1} = (\sigma_{j+1}I_n - A)^{-1}BR_{j+1}$ if σ_{j+1} is finite and $\widetilde{V}_{j+1} = ABR_{j+1}$ if $\sigma_{j+1} = \infty$. We notice that if at some step j, $h_{j+1,j} = 0$, then $(\sigma_{j+1}I_n - A)^{-1}BR_{j+1}$ is written as a linear combination of the computed blocks V_i, $i = 1,...,j$,

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

and in this case, we drop the interpolation point σ_{j+1} and tangent direction R_{j+1}, replace them by the new ones σ_{j+2} and R_{j+2} respectively, and go to the following step.

Algorithm 10 constructs also an upper Hessenberg matrix $\widetilde{H}_m$ whose elements are the $h_{i,j}$'s. The upper Hessenberg matrix H_m is the $m \times m$ matrix obtained from $\widetilde{H}_m$ by deleting its last row :

$$\widetilde{H}_m = \begin{pmatrix} H_m \\ h_{m+1,m}e_m^T \end{pmatrix}.$$

Let $\widetilde{K}_m$ be the matrix defined as follows

$$\widetilde{K}_m = \begin{pmatrix} H_m D_m \\ \sigma_{m+1}h_{m+1,m}e_m^T \end{pmatrix},$$

where D_m is the diagonal matrix $D_m = diag(\sigma_1,...,\sigma_m)$ containing the interpolation points. The following theorem derives some useful results that will be used later.

Theorem 5.1 *Let $\mathcal{V}_{m+1}$ be the F-orthonormal matrix of $\mathbb{R}^{n\times(m+1)s}$ constructed by Algorithm 10. Then, for $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times s}$, we have the following relations*

$$A\mathcal{V}_{m+1}(\widetilde{H}_m \otimes I_s) = \mathcal{V}_{m+1}(\widetilde{K}_m \otimes I_s) - B\widetilde{\mathcal{R}}_{m+1}, \tag{5.3}$$

and

$$T_m = \mathcal{V}_m^T \diamond A\mathcal{V}_m = \left[H_m D_m - \mathcal{V}_m^T \diamond B\widetilde{\mathcal{R}}_{m+1} - h_{m+1,m}(\mathcal{V}_m^T \diamond AV_{m+1})e_m^T\right] H_m^{-1}, \qquad (5.4)$$

where

$$\widetilde{\mathcal{R}}_{m+1} = \mathcal{R}_{m+1}\begin{pmatrix} I_m \\ e_m^T \end{pmatrix} \otimes I_s, \qquad (5.5)$$

and $\mathcal{R}_{m+1} = [R_1, ..., R_{m+1}]$.

Proof : From Algorithm10, we have

$$h_{j+1,j}V_{j+1} = (\sigma_{j+1}I_n - A)^{-1}BR_{j+1} - \sum_{i=1}^{j} h_{i,j}V_i \qquad j = 1, ..., m. \qquad (5.6)$$

Multiplying (5.6) on the left by $(\sigma_{j+1}I_n - A)$ and re-arranging terms, we get

$$A\sum_{i=1}^{j+1} h_{i,j}V_i = \sigma_{j+1}\sum_{i=1}^{j+1} h_{i,j}V_i - BR_{j+1} \qquad j = 1, ..., m, \qquad (5.7)$$

which gives the following relation

$$A\mathcal{V}_{m+1}(\widetilde{H}_m \otimes I_s) = \mathcal{V}_m(H_m D_m \otimes I_s) + \sigma_{m+1}h_{m+1,m}V_{m+1}(e_m^T \otimes I_s) - B\widetilde{\mathcal{R}}_{m+1}.$$

Hence

$$A\mathcal{V}_{m+1}(\widetilde{H}_m \otimes I_s) = \mathcal{V}_{m+1}(\widetilde{K}_m \otimes I_s) - B\widetilde{\mathcal{R}}_{m+1}.$$

For the proof of the expression (5.4), we have

$$\begin{aligned} A\mathcal{V}_{m+1}(\widetilde{H}_m \otimes I_s) = \quad & \mathcal{V}_m(H_m D_m \otimes I_s) + \sigma_{m+1}h_{m+1,m}V_{m+1}(e_m^T \otimes I_s) \\ & -BR_{m+1}(e_m^T \otimes I_s) - B\mathcal{R}_m(I_m \otimes I_s), \end{aligned}$$

which gives

$$A\mathcal{V}_m(H_m \otimes I_s) = \mathcal{V}_m(H_m D_m \otimes I_s) + \sigma_{m+1}h_{m+1,m}V_{m+1}(e_m^T \otimes I_s) - B\mathcal{R}_m(I_m \otimes I_s)$$

$$-BR_{m+1}(e_m^T \otimes I_s) - h_{m+1,m}AV_{m+1}(e_m^T \otimes I_s).$$

This implies that

$$\begin{aligned} A\mathcal{V}_m = \; & \Big[\mathcal{V}_m(H_m D_m \otimes I_s) + \sigma_{m+1} h_{m+1,m} V_{m+1}(e_m^T \otimes I_s) - B\mathcal{R}_m(I_m \otimes I_s) \\ & -h_{m+1,m} A V_{m+1}(e_m^T \otimes I_s)\Big](H_m \otimes I_s)^{-1}. \end{aligned}$$

Multiplying on the left by $\mathcal{V}_m^T$ with the diamond product, we get

$$\begin{aligned} \mathcal{V}_m^T \diamond A\mathcal{V}_m \; = \; & \Big[\mathcal{V}_m^T \diamond \mathcal{V}_m(H_m D_m \otimes I_s) + h_{m+1,m}\mathcal{V}_m^T \diamond V_{m+1}(e_m^T D_m \otimes I_s) \\ & -\mathcal{V}_m^T \diamond B\widetilde{R}_{m+1} - h_{m+1,m}\mathcal{V}_m^T \diamond A V_{m+1} e_m^T)\Big] H_m^{-1}. \end{aligned}$$

Finally, we obtain

$$\mathcal{V}_m^T \diamond A\mathcal{V}_m \; = \; \Big[H_m D_m - \mathcal{V}_m^T \diamond B\widetilde{R}_{m+1} - h_{m+1,m}\mathcal{V}_m^T \diamond A V_{m+1} e_m^T\Big] H_m^{-1}.$$

The following result gives a recursion for computing $T_{m+1} = \mathcal{V}_{m+1}^T \diamond A\mathcal{V}_{m+1}$ without requiring additional matrix-matrix products with A.

Proposition 5.2 *Let $T_{m+1} = \mathcal{V}_{m+1}^T \diamond A\mathcal{V}_{m+1} = [t_{:,1}, ..., t_{:,m+1}]$ and $\widetilde{H}_m = [h_{:,1}, ..., h_{:,m}]$, where $t_{:,i}$, $h_{:,i} \in \mathbb{R}^{m+1}$ are the i-th column of the $(m+1)\times(m+1)$ matrix T_{m+1} and the $(m+1)\times m$ upper Hessenberg matrix $\widetilde{H}_m$ obtained from Algrorithm 10, respectively. Then for $j = 1, ..., m$*

$$t_{:,j+1} = \frac{1}{h_{j+1,j}}\left[\sigma_{j+1}\left(e_1^{(m+1)}, ..., e_{j+1}^{(m+1)}\right)h_{1:j+1,j} - t_{:,1:j}h_{1:j,j} - \mathcal{V}_{m+1}^T \diamond (BR_{j+1})\right].$$

Proof : Using (5.6) and re-arranging terms, we get

$$h_{j+1,j} A V_{j+1} = \sigma_{j+1}\sum_{i=1}^{j+1} h_{i,j} V_i - A\sum_{i=1}^{j} h_{i,j} V_i - BR_{j+1} \qquad j = 1, ..., m,$$

which gives the following relation,

$$AV_{j+1} = \frac{1}{h_{j+1,j}}\left[\sigma_{j+1}[V_1, ..., V_{j+1}](h_{1:j+1,j} \otimes I_s) - A[V_1, ..., V_j](h_{1:j,j} \otimes I_s) - BR_{j+1}\right],$$

Multiplying on the left by $\mathcal{V}_{m+1}^T$ and using the properties of the $\diamond$-product, we

obtain

$$\mathcal{V}_{m+1}^T \diamond AV_{j+1} = \frac{1}{h_{j+1,j}}\Big[\sigma_{j+1}\left(\mathcal{V}_{m+1}^T \diamond \mathcal{V}_{j+1}\right)h_{1:j+1,j} - t_{:,1:j}h_{1:j,j} - \mathcal{V}_{m+1}^T \diamond (BR_{j+1})\Big],$$

hence,

$$t_{:,j+1} = \frac{1}{h_{j+1,j}}\left[\sigma_{j+1}\left(e_1^{(m+1)},...,e_{j+1}^{(m+1)}\right)h_{1:j+1,j} - t_{:,1:j}h_{1:j,j} - \mathcal{V}_{m+1}^T \diamond (BR_{j+1})\right].$$

5.3 The adaptive global Arnoldi method

The performance of the tangential approach depends on the choice of the poles and the tangent directions. The authors in [43] proposed an adaptive method for choosing the interpolation points and the tangential directions. The main objective of this method is the fact that at each iteration, new interpolation points and tangent directions are computed by minimizing the norm of a certain approximation of the error. Here, we will use a similar approach for our proposed global tangential-based method.

Next, let us see how to approximate the transfer function $H(\omega) = C(\omega I_n - A)^{-1}B$ when using global-type methods.

Setting $H(\omega) = CX$, where $X \in \mathbb{R}^{n\times s}$ is the solution of the following linear system

$$(\omega I_n - A)X = B. \tag{5.8}$$

An approximate solution $X_m \in Span\{V_1,...,V_m\}$ of the solution of (5.8) (assuming that $\omega I_n - A$ is nonsingular), can be determined by imposing the Petrov-Galerkin condition

$$R_B(\omega) \perp_F Span\{V_1,...,V_m\}$$

where the residual $R_B(\omega)$ is given by

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

Therefore,

$$X_m = \mathcal{V}_m\Big[(\omega I_m - T_m)^{-1}(\mathcal{V}_m^T \diamond B) \otimes I_s\Big].$$

Using the properties of the Kronecker product, we get

$$X_m = \mathcal{V}_m(\omega I_{ms} - (T_m \otimes I_s))^{-1}((\mathcal{V}_m^T \diamond B) \otimes I_s),$$

and then the approximated transfer function $H_m(\omega)$ is defined by

$$H_m(\omega) = C\,X_m,$$

which gives

$$\begin{aligned} H_m(\omega) &= C\mathcal{V}_m(\omega I_{ms} - (T_m \otimes I_s))^{-1}((\mathcal{V}_m^T \diamond B) \otimes I_s) \\ &= C_m(\omega I_{ms} - A_m)^{-1} B_m, \end{aligned} \tag{5.9}$$

where

$$\begin{aligned} A_m &= ((\mathcal{V}_m^T \diamond A\mathcal{V}_m) \otimes I_s) \in \mathbb{R}^{ms \times ms}, \quad & T_m &= \mathcal{V}_m^T \diamond A\mathcal{V}_m \in \mathbb{R}^{m \times m}, \\ B_m &= ((\mathcal{V}_m^T \diamond B) \otimes I_s) \in \mathbb{R}^{ms \times s}, & C_m &= C\mathcal{V}_m \in \mathbb{R}^{s \times ms}. \end{aligned}$$

The residual can be written as

$$R_B(\omega) = B - (\omega I_n - A)\mathcal{V}_m(\omega I_{ms} - A_m)^{-1} B_m. \tag{5.10}$$

In the following, we describe an adaptive approach for choosing interpolation points and tangent directions.

Using the expression of the transfer function (4.9), where $\mathcal{V}_m = [V_1, ..., V_m]$ has F-orthonormal block columns, we want to extend the subspace $Span\{V_1, ..., V_m\}$ by

$$V_{m+1} = (\sigma_{m+1} I_n - A)^{-1} B R_{m+1},$$

with the tangent direction $R_{m+1} \in \mathbb{R}^{s \times s}$, $\|R_{m+1}\|_2 = 1$ and with the new interpolation point $\sigma_{m+1} \in S_m$. Here $S_m \subset \mathbb{R}^+$ is a set defined similarly as in [12, 13], i.e., the convex hull of $\{-\lambda_1, ..., -\lambda_m\}$ where $\{\lambda_i\}_{i=1}^m$ are the eigenvalues of A_m.
The interpolation point σ_{m+1} and tangent direction R_{m+1} are computed as follows

$$(R_{m+1}, \sigma_{m+1}) = arg \max_{\substack{\omega \in S_m \\ R \in \mathbb{R}^{s \times s}, \|R\|_2 = 1}} \|R_B(\omega) R\|_2, \tag{5.11}$$

where the $R_B(\omega)$ is residual given by (5.10). The idea of this choice of poles and directions comes from the result of the following proposition.

Proposition 5.3 *Let $\mathcal{V}_m \in \mathbb{R}^{n\times ms}$ be the F-orthogonal matrix generated by the global tangential Arnoldi algorithm (GTAA), $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times s}$ and $C \in \mathbb{R}^{s\times n}$. The following relation holds*

$$\|H(\omega) - H_m(\omega)\|_2 \leq \|C(\omega I_n - A)^{-1}\|_2 \|R_B(\omega)\|_2. \tag{5.12}$$

Proof : Using the expression of transfer functions $H(\omega)$ and $H_m(\omega)$, we have

$$\begin{aligned}
\|H(\omega) - H_m(\omega)\|_2 &= \|C(\omega I_n - A)^{-1}B - C_m(\omega I_{ms} - A_m)^{-1}B_m\|_2, \\
&= \|C(\omega I_n - A)^{-1}(B - (\omega I_n - A)\mathcal{V}_m(\omega I_{ms} - A_m)^{-1}B_m)\|_2, \\
&= \|C(\omega I_n - A)^{-1}R_B(\omega)\|_2, \\
&\leq \|C(\omega I_n - A)^{-1}\|_2 \|R_B(\omega)\|_2.
\end{aligned}$$

The parameters σ_{m+1} and tangential direction block R_{m+1} are computed as follows. We first compute σ_{m+1} by maximizing the norm of the residual on the convex hull S_m,

$$\sigma_{m+1} = arg\max_{\omega\in S_m} \|R_B(\omega)\|_2, \tag{5.13}$$

For small to medium problems, this is done by computing the norm of $R_B(\omega)$ for each ω in S_m. The tangent direction R_{m+1} is computed by solving the problem

$$R_{m+1} = arg \max_{R\in\mathbb{R}^{s\times s}, \|R\|=1} \|R_B(\sigma_{m+1})R\|_2. \tag{5.14}$$

The tagential matrix direction $R_{m+1} = [r_{m+1}^{(1)}, ..., r_{m+1}^{(s)}]$, can be determined such that $r_{m+1}^{(i)}$ are the right singular vectors corresponding to the s largest singular values of $R_B(\sigma_{m+1})$.
Notice that as the adaptive tangential interpolation approximation becomes exact when

$$\max_{\omega\in S_m} \|R_B(\omega)\|_2 = 0,$$

it is reasonable to stop the process when $\|R_B(\sigma_{m+1})\|_2$ is small enough.

Proposition 5.4 *Let $A_m = (\mathcal{V}_m^T \diamond A\mathcal{V}_m) \otimes I_s$, then the residual $R_B(\omega)$ is expressed as*

$$R_B(\omega) = (I - \mathcal{P}_m)(B) + (A\mathcal{V}_m - \mathcal{V}_m A_m)Y_m(\omega), \tag{5.15}$$

where $Y_m(\omega) = (\omega I_{ms} - A_m)^{-1} B_m$, $\mathcal{V}_m = [V_1, ..., V_m]$ and $\mathcal{P}_m(B) = \mathcal{V}_m \Big[(\mathcal{V}_m^T \diamond B) \otimes I_s\Big]$.

Proof : We have

$$\begin{aligned}
R_B(\omega) &= B + A\mathcal{V}_m Y_m(\omega) - \omega \mathcal{V}_m Y_m(\omega), \\
&= B + A\mathcal{V}_m Y_m(\omega) - \mathcal{V}_m(\omega I_{ms} - A_m)(\omega I_{ms} - A_m)^{-1} B_m, \\
&\quad -\mathcal{V}_m A_m (\omega I_{ms} - A_m)^{-1} B_m, \\
&= B + A\mathcal{V}_m Y_m(\omega) - \mathcal{V}_m B_m - \mathcal{V}_m A_m Y_m(\omega), \\
&= B - \mathcal{P}_m(B) + (A\mathcal{V}_m - \mathcal{V}_m A_m)Y_m(\omega), \\
&= (I - \mathcal{P}_m)(B) + (A\mathcal{V}_m - \mathcal{V}_m A_m)Y_m(\omega).
\end{aligned}$$

Next we give a result obtained, by using the global tangential Arnoldi algorithm (GTAA).

Proposition 5.5 *Let $\mathcal{R}_m = [R_1, ..., R_m]$ and let $\mathcal{V}_m$ and H_m be the F-orthonormal and the upper Hessenberg matrices obtained at step m of the global tangential Arnoldi algorithm. Then*

$$A\mathcal{V}_m - \mathcal{V}_m A_m = -(I - \mathcal{P}_m)(B\mathcal{R}_m) \times (H_m^{-1} \otimes I_s), \tag{5.16}$$

and

$$R_B(\omega) = (I - \mathcal{P}_m)(B) - (I - \mathcal{P}_m)(B\mathcal{R}_m) \times (H_m^{-1} \otimes I_s) Y_m(\omega), \tag{5.17}$$

where $\mathcal{P}_m(B\mathcal{R}_m) = \mathcal{V}_m \Big[(\mathcal{V}_m^T \diamond B\mathcal{R}_m) \otimes I_s\Big]$.

Proof : Let $K_m = [(\sigma_1 I - A)^{-1} BR_1, ..., (\sigma_m I - A)^{-1} BR_m]$. Then from the global tangential Arnoldi algorithm, we can show that $K_m = \mathcal{V}_m(H_m \otimes I_s)$. Setting $D_m = diag(\sigma_1, ..., \sigma_m)$, and using 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

$$A\mathcal{V}_m = AK_m(H_m \otimes I_s)^{-1} = [-B\mathcal{R}_m + K_m(D_m \otimes I_s)](H_m \otimes I_s)^{-1}. \tag{5.18}$$

Therefore replacing $A_m = (\mathcal{V}_m^T \diamond A\mathcal{V}_m) \otimes I_s$ and $K_m = \mathcal{V}_m(H_m \otimes I_s)$ in (5.18) , we get

$$\begin{aligned} A\mathcal{V}_m - \mathcal{V}_m A_m &= [-B\mathcal{R}_m + \mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)](H_m \otimes I_s)^{-1} - \mathcal{V}_m\Big[(\mathcal{V}_m^T \diamond A\mathcal{V}_m) \otimes I_s\Big], \\ &= -B\mathcal{R}_m(H_m \otimes I_s)^{-1} + \mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)(H_m \otimes I_s)^{-1} \\ &\quad -\mathcal{V}_m\Big[(\mathcal{V}_m^T \diamond (-B\mathcal{R}_m + \mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s))(H_m \otimes I_s)^{-1}) \otimes I_s\Big]. \end{aligned}$$

It follows that

$$\begin{aligned} A\mathcal{V}_m - \mathcal{V}_m A_m &= -B\mathcal{R}_m(H_m \otimes I_s)^{-1} + \mathcal{V}_m\Big[(\mathcal{V}_m^T \diamond (B\mathcal{R}_m)(H_m \otimes I_s)^{-1}) \otimes I_s\Big] \\ &\quad +\mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)(H_m \otimes I_s)^{-1} \\ &\quad -\mathcal{V}_m\Big[(\mathcal{V}_m^T \diamond (\mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)(H_m \otimes I_s)^{-1})) \otimes I_s\Big], \\ &= -B\mathcal{R}_m(H_m \otimes I_s)^{-1} + \mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)(H_m \otimes I_s)^{-1} \\ &\quad -\mathcal{V}_m\Big[(\mathcal{V}_m^T \diamond (\mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)(H_m^{-1} \otimes I_s))) \otimes I_s\Big] \\ &\quad +\mathcal{V}_m\Big[((\mathcal{V}_m^T \diamond (B\mathcal{R}_m(H_m^{-1} \otimes I_s)) \otimes I_s\Big]. \end{aligned}$$

Using the relation

$$\begin{aligned} \mathcal{P}_m(B\mathcal{R}_m(H_m^{-1} \otimes I_s)) &= \Big[(\mathcal{V}_m^T \diamond (B\mathcal{R}_m(H_m^{-1} \otimes I_s)) \otimes I_s\Big] \\ &= \Big[((\mathcal{V}_m^T \diamond B\mathcal{R}_m) \times H_m^{-1}) \otimes I_s\Big] \\ &= \Big[(\mathcal{V}_m^T \diamond B\mathcal{R}_m) \otimes I_s\Big] \times (H_m^{-1} \otimes I_s) \\ &= \mathcal{P}_m(B\mathcal{R}_m) \times (H_m \otimes I_s)^{-1}, \end{aligned}$$

we obtain

$$\begin{aligned} A\mathcal{V}_m - \mathcal{V}_m A_m &= -(I - \mathcal{P}_m)(B\mathcal{R}_m)(H_m \otimes I_s)^{-1} + \mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)(H_m \otimes I_s)^{-1} \\ &- \mathcal{V}_m\Big[(\mathcal{V}_m^T \diamond (\mathcal{V}_m(H_m D_m H_m^{-1} \otimes I_s)) \otimes I_s\Big], \\ &= -(I - \mathcal{P}_m)(B\mathcal{R}_m)(H_m \otimes I_s)^{-1} + \mathcal{V}_m(H_m \otimes I_s)(D_m \otimes I_s)(H_m \otimes I_s)^{-1} \\ &- \mathcal{V}_m\Big[((\mathcal{V}_m^T \diamond \mathcal{V}_m)H_m D_m H_m^{-1}) \otimes I_s\Big], \\ &= -(I - \mathcal{P}_m)(B\mathcal{R}_m) \times (H_m^{-1} \otimes I_s). \end{aligned}$$

The expression of $R_B(\omega)$ given in (5.17) allows us to reduce the computational

cost while seeking for the next pole and direction. Solving $\max_{\omega \in S_m} \|R_B(\omega)\|$ for a large sample of values for ω requires the computation of the tall matrix $R_B(\omega)$ at each value of ω. Proposition 5.5 shows that computational cost can decrease, as ω varies. In fact, consider the skiny QR decomposition of $(I - \mathcal{P}_m)(B) = Q_1 L_1$ and $(I - \mathcal{P}_m)(B\mathcal{R}_m) = Q_2 L_2$, it follows that

$$\|R_B(\omega)\|_2 \leq \|L_1\|_2 + \|L_2(H_m^{-1} \otimes I_s) Y_m(\omega)\|_2.$$

We notice that, if we consider the case where the tangential subspace includes B as a first block, then $(I - \mathcal{P}_m)(B) = 0$ and hence

$$R_B(\omega) = -(I - \mathcal{P}_m)(B\mathcal{R}_m)(H_m^{-1} \otimes I_s) Y_m(\omega),$$

which shows that

$$\|R_B(\omega)\|_2 = \|L_2(H_m^{-1} \otimes I_s) Y_m(\omega)\|_2.$$

this means that, for each value of ω, one has to only compute norms of matrices of size $ms \times ms$. In the following, we present the adaptive global tangential Arnoldi algorithm (AGTAA). This algorithm allows us to compute a low dimentional dynamical system by computing the matrices $A_{m_{max}} = ((\mathcal{V}_{m_{max}}^T \diamond A\mathcal{V}_{m_{max}}) \otimes I_s)$, $B_{m_{max}} = ((\mathcal{V}_{m_{max}}^T \diamond B) \otimes I_s)$ and $C_{m_{max}} = C\mathcal{V}_{m_{max}}$ for a fixed value of m_{max} where $\mathcal{V}_{m_{max}}$ is the F-orthonormal matrix obtained by applying Algorithm 10 . The algorithm is summarized as follows :

Algorithm 11 (Adaptive global tangential Arnoldi algorithm AGTAA) .
– *Given* A, B, C m_{max}, $\omega_0^{(1)} \in \mathbb{R}$, $R_1 \in \mathbb{R}^{s \times s}$, m_{max}
– *Outputs :* $A_{m_{max}} = ((\mathcal{V}_{m_{max}}^T \diamond A\mathcal{V}_{m_{max}}) \otimes I_s)$, $B_{m_{max}} = ((\mathcal{V}_{m_{max}}^T \diamond B) \otimes I_s)$ *and* $C_{m_{max}} = C\mathcal{V}_{m_{max}}$.

- *Set* $\sigma_1 = \omega_0^{(1)}$, $\widetilde{V_1} = (\sigma_1 I_n - A)^{-1} B R_1$.
- *Construct* V_1 *such that* $\|V_1\|_F = 1$, *initialize :* $\mathcal{V}_1 = [V_1]$.
- *For* $m = 1 : m_{max}$
  1. *If* $\bar{\sigma}_{m-1} \neq \sigma_m \in \mathbb{C}$ *then* $\sigma_{m+1} = \bar{\sigma}_m$ *else compute* $\{\lambda_1, ..., \lambda_{ms}\}$ *eigenvalues of* A_m.
  2. *Determine* S_m, *the convex hull of* $\{-\lambda_1, ..., -\lambda_{ms}, \omega_0^{(1)}, \bar{\omega}_0^{(1)}\}$ *and solve (5.13).*

3. *Compute the right vector* R_{m+1} *by solving (5.14).*
4. *If* $\sigma_{m+1} \neq \infty$, $\widetilde{V}_{m+1} = (\sigma_{m+1} I_n - A)^{-1} B R_{m+1}$ *else* $\widetilde{V}_{m+1} = AB R_{m+1}$.
5. *For* $i = 1,...,m$
   - $h_{i,m} = tr(W_i^T \widetilde{V}_{m+1})$.
   - $\widetilde{V}_{m+1} = \widetilde{V}_{m+1} - h_{i,m} V_i$.

   End
6. $h_{m+1,m} = \|\widetilde{V}_{m+1}\|_F$,
7. $V_{m+1} = \frac{\widetilde{V}_{m+1}}{h_{m+1,m}}$.
8. $\mathcal{V}_{m+1} = [\mathcal{V}_m, V_{m+1}]$.

- *End*

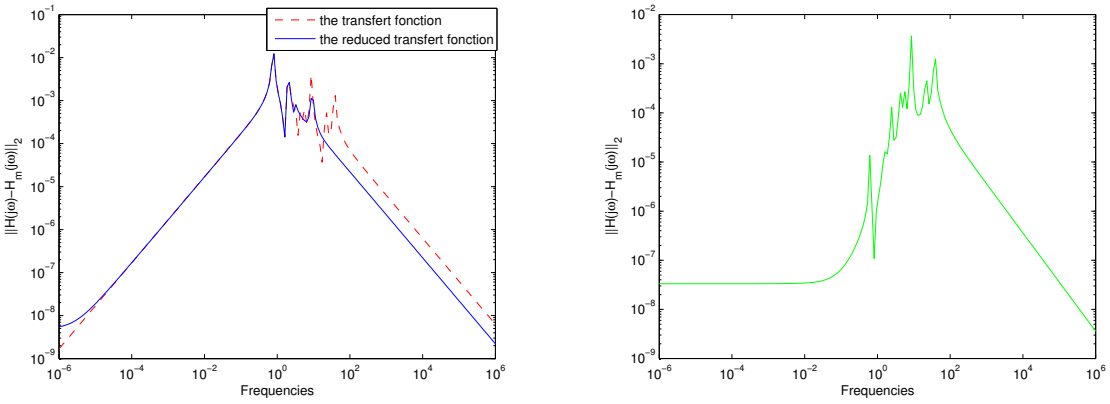
At each iteration m, the total number of arithmetic operations is dominated by the computation of the matrix $\widetilde{V}_{m+1}$ at Step 4 of Algorithm 11 by solving the multiple shifted linear system $(\sigma_{m+1} I_n - A)V = BR_{m+1}$. For medium or large structured problems, this is done by using the LU factorization of $(\sigma_{m+1} I_n - A)$. For large problems, one can also use a Krylov solver such as the block or standard GMRES with a suitable preconditionner. We notice that the well known Iterative Rational Krylov Algorithm (IRKA) algorithm needs, at each iteration, two LU factorization. In all our numerical tests, we solved those shifted linear systems using the LU factorization. We also need to compute, for each iteration m, the block direction R_{m+1} by solving the problem (5.14) which is done by computing the leading s right singular vectors of the $n \times s$ matrix $R_B(\sigma_{m+1})$.

5.4 Numerical experiments

In this section, we give some experimental results to show the effectiveness of the proposed approach. All the experiments were performed on a computer of Intel Core i5 at 3.00GHz and 8Go of RAM. The algorithms were coded in Matlab 8.0.

Example 1 : The model of the first experiment is the model of stage 1R of the International Space Station (ISS). Figure 5.1a shows the singular values vs frequencies of the transfer function and its approximation. In Figure 5.1b, we

plotted the 2-norm of the errors $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies for $m = 15$. As can be shown from these plots, the AGTAA algorithm gives good results with a small value of m.

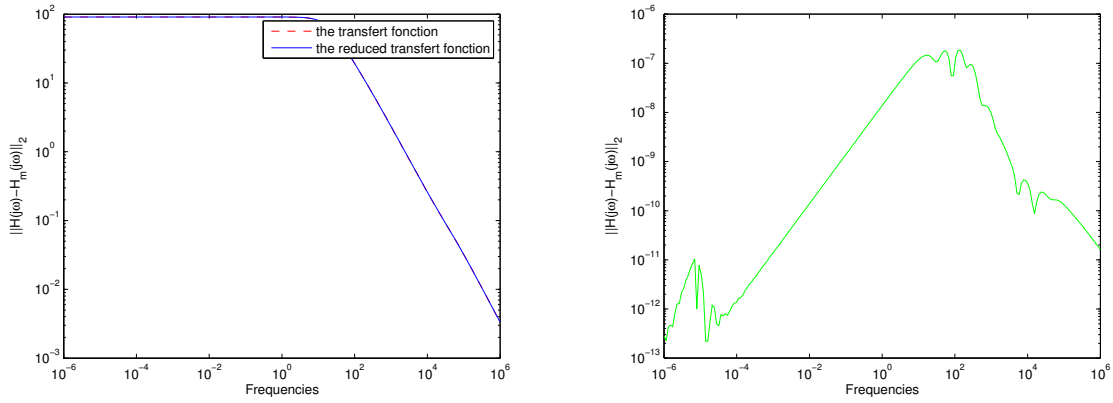


(a) Singular values vs frequencies. (b) Error-norms vs frequencies.

FIGURE 5.1 – The ISS model : n=270, p=3.

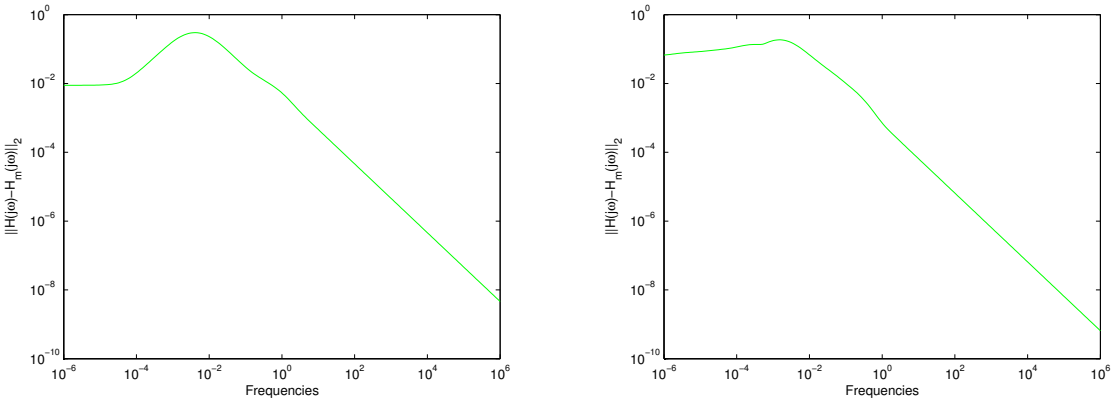
Example 2 : In this example, we applied the AGTAA method on 3 models : FDM, RAIL3113 and RAIL821. The plots in Figure 5.2a, represent the sigma-plot (the singular values of the transfer function) of the original system (dashed-dashed line) and the one of the reduced order system (solid line) of the FDM model. In Figure 5.2b, we plotted the error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-6}, 10^{6}]$. For this experiment, the value of m was $m = 20$.
In Figure 5.3a and Figure 5.3b, we considered the two models : RAIL821 and RAIL3113. These models describe the steel rail cooling problem (order $n = 821$ and $n = 3113$, respectively). The plots in these figures represent the error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies.

Example 3 : In this example we compared the AGTAA algorithm with the Iterative Rational Krylov Algorithm (IRKA)[2]. We used five models : CDplayer, MNA_1, MNA_3 and Flow.
The figures : Figure 5.4a–Figure 5.5b illustrate the exact error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies for AGTAA (solid line) and IRKA (dashed-dotted line) with $m = 20$. In Figure 5.4b, the AGTAA returns good results for small frequencies with a little advantage for IRKA for large frequencies. It is



(a) Singular values vs frequencies. (b) Error-norms vs frequencies.

FIGURE 5.2 – The FDM model : n=10000, p=4.

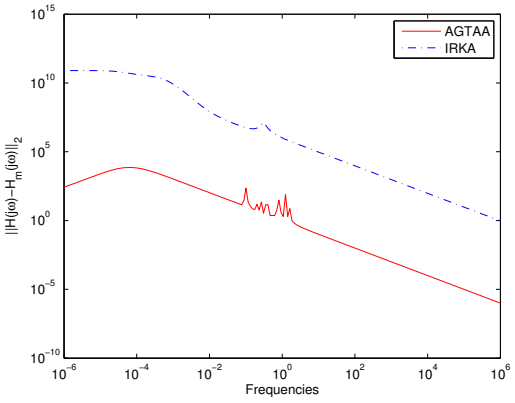


(a) The RAILl821 model : n=821, p=6 (b) The RAIL3113 model : n=3113, p=6

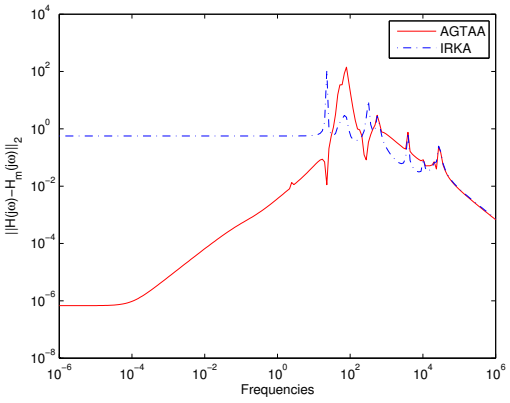
FIGURE 5.3 – Error-norms vs frequencies.

clear that in all the presented plots, the AGTAA algorithm returns generally good results as compared to the IRKA algorithm. We also compared the execusion times of the two algorithms and we reported the obtained timings in Table 5.1.

The results of Table 5.1 show that the cost of IRKA method is much higher than the cost of the adaptive global tangential Arnoldi method.

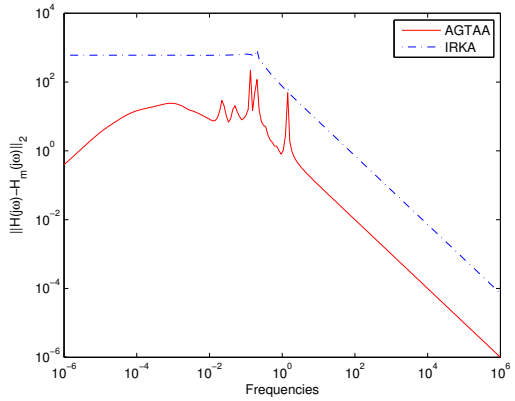


(a) The MNA_1 model : n=578, p=9

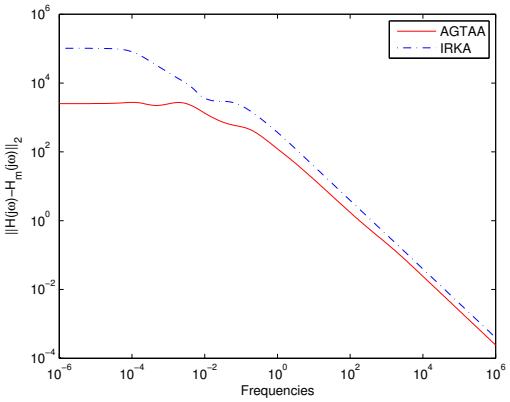


(b) The CDplayer model : n=120, p=2

FIGURE 5.4 – AGTAA (solid line) and the IRKA (dashed-dotted line).



(a) The MNA_3 model : n=4863, p=22.



(b) The Flow model : n=9669, p=5.

FIGURE 5.5 – AGTAA (solid line) and the IRKA (dashed-dotted line).

Model	AGTAA	IRKA
MNA_1	0.15s	1.45s
MNA_3	1.79s	19.2s
CDplayer	0.04s	0.12s
Flow	15.20s	51.82

TABLEAU 5.1 – The calculation time

5.5 Conclusion

In this chapter, we proposed a new global tangential Arnoldi method to get reduced order dynamical systems that approximate the initial large scale multiple inputs and multiple outputs (MIMO) ones. The method generates sequences of matrix tangential rational Krylov subspaces by selecting, in an adaptive way, the interpolation shifts (poles) and tangential directions by maximizing the residual norms. We gave some new algebraic properties and present some numerical experiments on some benchmark examples.

Quatrième partie

Tangential Lanczos methods for model reduction

Chapitre 6

An adaptive tangential Lanczos-type method for model reduction

6.1 Introduction

In this chapter, we present a new approach for model reduction of MIMO systems, named : Adaptive Tangential Lanczos Algorithm (ATLA). This method based on tangential interpolation using two tangential Krylov subspaces for the oblique projection of the original system, in order to solve the problem 2. We used also the Lanczos-type method for generating two bi-orthonormal bases. We give some algebraic properties and present some numerical examples to show the effectiveness of the proposed method.

6.2 The tangential Lanczos-type algorithm

We present here a tangential Lanczos-type algorithm for constructing bi-orthogonal bases $\{v_1, v_2, \ldots, v_m\}$ and $\{w_1, w_2, \ldots, w_m\}$ of the following tangential Krylov subspaces :

$$Range\{(A-\sigma_1 I_n)^{-1}Br_1, ..., (A-\sigma_m I_n)^{-1}Br_m\},$$

and

$$Range\{(A-\mu_1 I_n)^{-T}C^T l_1, ..., (A-\mu_m I_n)^{-T}C^T l_m\},$$

respectively so that the tangential interpolation problem 2 is solved, i.e., the reduced order model has to interpolate the original transfer function $H(s)$ and its first derivative as in (3.6).
The tangential Lanczos algorithm is summarized as follows :

Algorithm 12 (The Tangential Lanczos Algorithm TLA) .
– *Inputs :* $A, B, C, \sigma = \{\sigma_i\}_{i=1}^{m+1}, \mu = \{\mu_i\}_{i=1}^{m+1}, r = \{r_i\}_{i=1}^{m+1}$, *and* $l = \{l_i\}_{i=1}^{m+1}, r_i, l_i \in \mathbb{R}^p$.
– *Output :* $V_m = \{v_1, ..., v_m\}, \quad W_m = \{w_1, ..., w_m\}$.

1. *Set* $\widetilde{v_1} = (A - \sigma_1 I_n)^{-1} B r_1$ *and* $\widetilde{w_1} = (A - \mu_1 I_n)^{-T} C^T l_1$.
2. *Set* $\widetilde{v_1} = h_{1,1} v_1$ *and* $\widetilde{w_1} = g_{1,1} w_1$ *such that* $w_1^T v_1 = 1$.
3. *Initialize :* $V_1 = [v_1]$, $W_1 = [w_1]$.
4. *For* $k = 1 : m$
5. *if* $\{\sigma_{k+1} = \infty\}$, $\widetilde{v_k} = A B r_{k+1}$ *else,* $\widetilde{v_k} = (A - \sigma_{k+1} I_n)^{-1} B r_{k+1}$, *End.*
6. *if* $\{\mu_{k+1} = \infty\}$, $\widetilde{w_k} = A C^T l_{k+1}$ *else,* $\widetilde{w_k} = (A - \mu_{k+1} I_n)^{-T} C^T l_{k+1}$, *End.*
7. $h_k = W_k^T \widetilde{v_k}$ *and* $g_k = V_k^T \widetilde{w_k}$.
8. $\widetilde{v_k} = \widetilde{v_k} - V_k h_k, \quad \widetilde{w_k} = \widetilde{w_k} - W_k g_k$.
9. $\widetilde{v_k} = h_{k+1,k} v_{k+1}$ *and* $\widetilde{w_k} = g_{k+1,k} w_{k+1}$, *such that* $w_{k+1}^T v_{k+1} = 1$.
10. $V_{k+1} = [V_k, v_{k+1}], \quad W_{k+1} = [W_k, w_{k+1}]$.
11. $W_{k+1}^T V_{k+1} = P_k D_k Q_k^T$ *(Singular Value Decomposition).*
12. $V_{k+1} = V_{k+1} Q_k D_k^{-1/2}$ *and* $W_{k+1} = W_{k+1} P_k D_k^{-1/2}$.
13. *End*

In our setting we assume that we are given the sequences of shifts $\sigma = \{\sigma_i\}_{i=1}^{m+1}$, $\mu = \{\mu_i\}_{i=1}^{m+1}$ and the tangents $r = \{r_i\}_{i=1}^{m+1}$, $l = \{l_i\}_{i=1}^{m+1}$. In the next section a procedure will be defined to generate those sequences similarly to previous chapters.
In the tangential Lanczos-type algorithm (TLA), steps 5-6 are used to generate the next Lanczos vectors. At each iteration k, we used a new interpolation point σ_{k+1} and the tangent direction r_{k+1}, $k = 1, ..., m$ and we initialize the subsequent tangential Krylov subspace corresponding to this shift by $\widetilde{v_k} = (A - \sigma_{k+1} I_n)^{-1} B r_{k+1}$ if σ_{k+1} is finite and $\widetilde{v_k} = A B r_{k+1}$ if $\sigma_{k+1} = \infty$, and we do the same for the next left shift μ_{k+1} and tangent direction l_{k+1}. To insure that the matrices V_{k+1} and W_{k+1}

generated in each iteration are bi-orthogonal, the SVD decomposition is used (steps 11 – 12). The vectors h_k and g_k constructed in step 7 are in $\mathbb{R}^k$ and they are used to construct the upper Hessenberg matrices $\widetilde{H}_m$, $\widetilde{G}_m$, $\widetilde{K}_m$, and $\widetilde{T}_m$ where $\widetilde{H}_m = \left[\widetilde{h}_1, ..., \widetilde{h}_m\right]$, $\widetilde{G}_m = [\widetilde{g}_1, ..., \widetilde{g}_m] \in \mathbb{R}^{(m+1)\times m}$ respectively, where

$$\widetilde{h}_k = \begin{bmatrix} h_k \\ h_{k+1,k} \\ \mathbf{0} \end{bmatrix} \text{ and } \widetilde{g}_k = \begin{bmatrix} g_k \\ g_{k+1,k} \\ \mathbf{0} \end{bmatrix}, \quad \text{for } k = 1, ..., m,$$

where $\mathbf{0}$ is the zero vector having $m-k$ rows. $\widetilde{K}_m$ and $\widetilde{T}_m$ are defined as $\widetilde{K}_m = \widetilde{H}_m D_m$, $\widetilde{T}_m = \widetilde{G}_m S_m$ where $D_m = Diag\{\sigma_2, ..., \sigma_{m+1}\}$ and $S_m = Diag\{\mu_2, ..., \mu_{m+1}\}$. We define $\widetilde{h}_0$ and $\widetilde{g}_0$ as

$$\widetilde{h}_0 = \begin{bmatrix} h_{1,1} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \widetilde{g}_0 = \begin{bmatrix} g_{1,1} \\ \mathbf{0} \end{bmatrix}. \tag{6.1}$$

The upper Hessenberg matrices H_m and G_m are the $m \times m$ matrices obtained from $\widetilde{H}_m$ and $\widetilde{G}_m$, by deleting their last rows. The Theorem 6.1 gives an equations that relate A, V_{m+1}, W_{m+1} and the Hessenberg matrices constructed by Algorithm 12.

Theorem 6.1 *Let V_{m+1} and W_{m+1} be the matrices generated by Algorithm 12. Then, for $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times p}$, we have the following relations*

$$AV_{m+1}\widetilde{H}_m = V_{m+1}\widetilde{K}_m + B\widetilde{R}_{m+1}, \text{ and}$$

$$A^T W_{m+1}\widetilde{G}_m = W_{m+1}\widetilde{T}_m + C^T\widetilde{L}_{m+1}.$$

where $\widetilde{R}_{m+1} = [r_2, ..., r_{m+1}]$ and $\widetilde{L}_{m+1} = [l_2, ..., l_{m+1}]$.

Proof : Replacing the expression of $\widetilde{v}_k$ into the expressions of step 8 and step 9 yields the following relation

$$h_{k+1,k}v_{k+1} = (A - \sigma_{k+1}I)^{-1}Br_{k+1} - V_k h_k,$$

which can be written as

$$[V_k\ v_{k+1}]\begin{bmatrix} h_k \\ h_{k+1,k} \end{bmatrix} = (A-\sigma_{k+1}I)^{-1}Br_{k+1}, \tag{6.2}$$

multiplying (6.2) on the left by $(A-\sigma_{j+1}I_n)$ and re-arranging terms, we get

$$AV_{k+1}\begin{bmatrix} h_k \\ h_{k+1,k} \end{bmatrix} = \sigma_{k+1}V_{k+1}\begin{bmatrix} h_k \\ h_{k+1,k} \end{bmatrix} + Br_{k+1}.$$

On the other hand, for $k = 1,...,m$, we have

$$AV_{m+1} = [AV_{k+1}, Av_{k+2},..., Av_{m+1}],$$

which gives,

$$AV_{m+1}\begin{bmatrix} h_k \\ h_{k+1,k} \\ \mathbf{0} \end{bmatrix} = \sigma_{k+1}V_m\begin{bmatrix} h_k \\ h_{k+1,k} \\ \mathbf{0} \end{bmatrix} + Br_{k+1},$$

where $\mathbf{0}$ is the zero vector having $m-k$ rows. then,

$$AV_{m+1}\widetilde{H}_m = V_{m+1}\widetilde{K}_m + B\widetilde{R}_m.$$

The same proof for the second relation.

Proposition 6.1 *Let V_{m+1} and W_{m+1} be the matrices generated by tangential Lanczos algorithm (TLA), and let*

$$\mathcal{K}_{m+1} = \left[(A-\sigma_1 I)^{-1}Br_1,...,(A-\sigma_{m+1}I)^{-1}Br_{m+1}\right],$$

$$\mathcal{T}_{m+1} = \left[(A-\mu_1 I)^{-T}C^T l_1,...,(A-\mu_{m+1}I)^{-T}C^T l_{m+1}\right],$$

then we have

$$\mathcal{K}_{m+1} = V_{m+1}\mathcal{H}_{m+1}, \tag{6.3}$$

$$\mathcal{T}_{m+1} = W_{m+1}\mathcal{G}_{m+1}, \tag{6.4}$$

where $\mathcal{H}_m = \left[\widetilde{h}_0\ \ \widetilde{H}_m\right]$ and $\mathcal{G}_m = \left[\widetilde{g}_0\ \ \widetilde{G}_m\right]$ are upper triangular matrices of $(m+1)\times$

$(m+1)$.

Using the equation (6.2)

$$[V_k\ v_{k+1}]\begin{bmatrix} h_k \\ h_{k+1,k} \end{bmatrix} = (A-\sigma_{k+1}I)^{-1}Br_{k+1}, \qquad k=1,...,m,$$

which gives

$$V_{m+1}\begin{bmatrix} h_k \\ h_{k+1,k} \\ \mathbf{0} \end{bmatrix} = (A-\sigma_{k+1}I_n)^{-1}Br_{k+1}, \qquad k=1,...,m,$$

it follows that

$$V_{m+1}\left[\widetilde{h}_1,...,\widetilde{h}_m\right] = \left[(A-\sigma_2 I_n)^{-1}Br_2,...,(A-\sigma_{m+1}I_n)^{-1}Br_{m+1}\right],$$

Since $h_{1,1}v_1 = (A-\sigma_1 I_n)^{-1}Br_1$, we have

$$V_{m+1}\left[\widetilde{h}_0,\widetilde{h}_1,...,\widetilde{h}_m\right] = \left[(A-\sigma_1 I_n)^{-1}Br_1,(A-\sigma_2 I_n)^{-1}Br_2,...,(A-\sigma_{m+1}I_n)^{-1}Br_{m+1}\right],$$

which ends the proof of (6.3). In the same manner we prove (6.4).

6.3 Adaptive choice of interpolation points and directions

Setting $H(\omega) = CX(\omega) = Y(\omega)B$, where $X(\omega),\ Y(\omega)^T \in \mathbb{R}^{n\times p}$ are the solutions of the following linear systems

$$(A-\omega I_n)X(\omega) = B \quad and \quad (A-\omega I_n)^T Y(\omega)^T = C^T. \tag{6.5}$$

An approximate solutions $X_m \in Range\{v_1,...,v_m\}$ and $Y_m^T \in Range\{w_1,...,w_m\}$ of the solution of (6.5) ($A-\omega I_n$ is assumed to be nonsingular), can be determined

by imposing the Galerkin condition

$$R_B(\omega) \perp Span\{w_1,...,w_m\} \quad and \quad R_C(\omega) \perp Span\{v_1,...,v_m\},$$

where the residuals $R_B(\omega)$ and $R_C(\omega)$ are given by

$$R_B(\omega) = (A-\omega I_n)X_m - B \quad and \quad R_C(\omega) = (A-\omega I_n)^T Y_m^T - C^T.$$

Therefore,

$$X_m(\omega) = V_m(A_m-\omega I_m)^{-1}W_m^T B \quad and \quad Y_m(\omega)^T = W_m(A_m-\omega I_m)^{-T}V_m^T C^T,$$

wich gives this expression of the residuals :

$$R_B(\omega) = (A-\omega I_n)V_m(A_m-\omega I_m)^{-1}W_m^T B - B, \tag{6.6}$$

and

$$R_C(\omega) = (A-\omega I_n)^T W_m(A_m-\omega I_m)^{-T}V_m^T C^T - C^T. \tag{6.7}$$

Which amounts to approximating $H(\omega)$ by $H_m(\omega)$ as

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

where $A_m = W_m^T A V_m$, $B_m = W_m^T B$ and $C_m = C V_m$. The matrices $V_m = [v_1,...,v_m]$ and $W_m = [w_1,...,w_m]$ are bi-orthonormal. The $v_i,\ w_i \in \mathbb{R}^n$ are a bases of m-dimensional subspaces in $\mathbb{R}^{n\times p}$. In the adaptive approach, we seek to extend our subspaces

$$Range(v_1,...,v_m) = Range\Big((A-\sigma_1 I_n)^{-1}Br_1,...,(A-\sigma_m I_n)^{-1}Br_m\Big),$$

$$Range(w_1,...,w_m) = Range\Big((A-\mu_1 I_n)^{-T}C^T l_1,...,(A-\sigma_m I_n)^{-T}C^T l_m\Big),$$

by a new vectors defined as

$$(A-\sigma_{m+1}I_n)^{-1}Br_{m+1},$$

and

$$(A - \mu_{m+1} I_n)^{-T} C^T l_{m+1},$$

respectively, which means at each iteration, we seek to calculate a new interpolation points σ_{m+1}, μ_{m+1} and a new tangent directions r_{m+1}, l_{m+1}. Our unknowns are computed as follows

$$(r_{m+1}, \sigma_{m+1}) = arg \max_{\substack{s \in S_m \\ d \in \mathbb{R}^p, \|d\| = 1}} \|R_B(s)d\|. \tag{6.8}$$

$$(l_{m+1}, \mu_{m+1}) = arg \max_{\substack{s \in S_m \\ d \in \mathbb{R}^p, \|d\| = 1}} \|R_C(s)d\|, \tag{6.9}$$

respectively, with the direction vectors r_{m+1}, $l_{m+1} \in \mathbb{R}^p$, $\|r_{m+1}\| = \|l_{m+1}\| = 1$ and with the poles $\sigma_{m+1}, \mu_{m+1} \in S_m$. Here $S_m \subset \mathbb{R}^+$ is a set defined similarly as in Chapter 4 and 5, i.e the convex hull of $\{-\lambda_1, ..., -\lambda_m\}$ where $\{\lambda_i\}_{i=1}^m$ are the eigenvalues of A_m. We have the following results.

Proposition 6.2 *Let $R_B(\omega) = (A - \omega I_n)V_m N_m(\omega) - B$, $R_C(\omega) = (A - \omega I_n)^T W m Z_m(\omega) C^T$, where $N_m(\omega) = (A_m - \omega I_m)^{-1} W_m^T B$ and $Z_m(\omega) = (A_m - \omega I_m)^{-T} V_m^T C^T$, then we have the following new expressions for $R_B(\omega)$ and $R_C(\omega)$:*

$$R_B(\omega) = (AV_m - V_m A_m)N_m(\omega) - (I - V_m W_m^T)B, \tag{6.10}$$

and

$$R_C(\omega) = (A^T W_m - W_m A_m^T)Z_m(\omega) - (I - W_m V_m^T)C^T. \tag{6.11}$$

Proof : We have

$$\begin{aligned} R_B(\omega) &= AV_m N_m(\omega) - \omega V_m N_m(\omega) - B \\ &= AV_m N_m(\omega) + V_m(A_m - \omega I)(A_m - \omega I)^{-1} W_m^T B - V_m A_m (A_m - \omega I)^{-1} W_m^T B - B \\ &= AV_m N_m(\omega) + V_m W_m^T B - V_m A_m N_m(\omega) - B \\ &= (AV_m - V_m A_m)N_m(\omega) - (I - V_m W_m^T)B, \end{aligned}$$

which proves (6.10). The expression (6.11) for $R_C(s)$ can be proved in the same way. The result above is valid, for any space spanned by the columns of V_m and W_m. If we use the approximation space given by our rational Krylov space, the

following result holds.

Proposition 6.3 *Let* $\mathcal{K}_m = \left[(A-\sigma_1 I)^{-1}Br_1,...,(A-\sigma_m I)^{-1}Br_m\right]$ *and* $\mathcal{T}_m = \left[(A-\mu_1 I)^{-T}C^T l_1,...,(A-\mu_m I)^{-T}C^T l_m\right]$, *and set* $\mathcal{K}_m = V_m\mathcal{H}_m$, $\mathcal{T}_m = W_m\mathcal{G}_m$ *where* V_m, W_m, $\mathcal{H}_m$ *and* $\mathcal{G}_m$ *are obtained by the Tangential Lanczos Algorithm (TLA). Let* $R_m = [r_1,...,r_m]$, $L_m = [l_1,...,l_m]$, *and we assume that* $\mathcal{H}_m$, $\mathcal{G}_m$ *are nonsingular, then*

$$AV_m - V_m A_m = (I_n - V_m W_m^T)BR_m\mathcal{H}_m^{-1}, \tag{6.12}$$

and

$$A^T W_m - W_m A_m^T = (I_n - W_m V_m^T)C^T L_m\mathcal{G}_m^{-1}. \tag{6.13}$$

We also have

$$R_B(\omega) = (I - V_m W_m^T)B(R_m\mathcal{H}_m^{-1}N_m(\omega) - I), \tag{6.14}$$

and

$$R_C(\omega) = (I - W_m V_m^T)C^T(L_m\mathcal{G}_m^{-1}Z_m(\omega) - I). \tag{6.15}$$

Let $D_m = diag(\sigma_1,...,\sigma_m)$, then from the fact that $A(A-\sigma_i I)^{-1}Br_i = Br_i + \sigma_i(A-\sigma_i I)^{-1}Br_i$, it follows that $AV_m = A\mathcal{K}_m\mathcal{H}_m^{-1} = (BR_m + \mathcal{K}_m D_m)\mathcal{H}_m^{-1}$. Using $A_m = W_m^T AV_m = \mathcal{G}_m^{-T}\mathcal{T}_m^T A\mathcal{K}_m\mathcal{H}_m^{-1}$ we have

$$\begin{aligned}
AV_m - V_m A_m &= BR_m\mathcal{H}_m^{-1} + \mathcal{K}_m D_m\mathcal{H}_m^{-1} - \mathcal{K}_m\mathcal{H}_m^{-1}\mathcal{G}_m^{-T}\mathcal{T}_m^T AV_m \\
&= BR_m\mathcal{H}_m^{-1} + \mathcal{K}_m D_m\mathcal{H}_m^{-1} - \mathcal{K}_m\mathcal{H}_m^{-1}\mathcal{G}_m^{-T}\mathcal{T}_m^T(BR_m + \mathcal{K}_m D_m)\mathcal{H}_m^{-1} \\
&= BR_m\mathcal{H}_m^{-1} + \mathcal{K}_m D_m\mathcal{H}_m^{-1} - \mathcal{K}_m\mathcal{H}_m^{-1}\mathcal{G}_m^{-T}\mathcal{T}_m^T BR_m\mathcal{H}_m^{-1} \\
&\quad -\mathcal{K}_m\mathcal{H}_m^{-1}\mathcal{G}_m^{-T}\mathcal{T}_m^T\mathcal{K}_m D_m\mathcal{H}_m^{-1} \\
&= BR_m\mathcal{H}_m^{-1} - \mathcal{K}_m\mathcal{H}_m^{-1}\mathcal{G}_m^{-T}\mathcal{T}_m^T BR_m\mathcal{H}_m^{-1} \\
&= (I_n - V_m W_m^T)BR_m\mathcal{H}_m^{-1}.
\end{aligned}$$

This proves the relation (6.12). The expression (6.13) can be obtained in the same way. The relations (6.14) and (6.15) are directly derived from (6.10), (6.11) and the expressions (6.12) and (6.13). These expressions allows us to reduce the computational cost while looking for the next pole and direction. The problems (6.8) and (6.9) are solved in the same way as in chapter 4 and 5. 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,

$$\sigma_{m+1} = arg\max_{\omega \in S_m} \|R_B(\omega)\|_2, \tag{6.16}$$

this is done by computing the norm of $R_B(\omega)$ for each s in S_m. Then the tangent direction r_{m+1} is computed by evaluating (6.16) at $s = \sigma_{m+1}$ and solve the problem

$$r_{m+1} = arg \max_{d \in \mathbb{R}^p, \|d\|=1} \|R_B(\sigma_{m+1})d\|_2 . \tag{6.17}$$

The tangential direction r_{m+1}, can also be determined as the largest right singular vector of the matrix $R_B(\sigma_{m+1})$.

Solving $\max_{\omega \in S_m} \|R_B(\omega)\|$ for a large sample of values for ω requires the computation of the tall matrix $R_B(\omega)$ at each value of ω. Proposition 6.3 shows that computational can be with small matrices, as ω varies. In fact, let $Q_1 L_1 = (I - V_m W_m^T)B$ be the skinny QR decomposition of $(I - V_m W_m^T)B$. Then

$$\begin{aligned} \sigma_{m+1} &= arg\max_{\omega \in S_m} \|R_B(\omega)\| \\ &= arg\max_{\omega \in S_m} \|L_1(R_m \mathcal{H}_m^{-1} N_m(\omega) - I)\|, \end{aligned} \tag{6.18}$$

where the computation of σ_{m+1} here, requires an $m \times m$ matrix for each value of ω. Then r_{m+1} is computed as

$$r_{m+1} = arg\max_{\|d\|=1} \|\|L_1(R_m \mathcal{H}_m^{-1} N_m(s) - I)d\|. \tag{6.19}$$

With the same procedure let $Q_2 L_2 = (I - W_m V_m^T)C^T$ be the skinny QR decomposition of $(I - W_m V_m^T)C^T$. Then

$$\begin{aligned} \mu_{m+1} &= arg\max_{\omega \in S_m} \|R_C(\omega)\| \\ &= arg\max_{\omega \in S_m} \|L_2(L_m \mathcal{G}_m^{-1} Z_m(\omega) - I)\|, \end{aligned} \tag{6.20}$$

and l_{m+1} is computed as

$$l_{m+1} = arg\max_{\|d\|=1} \|L_2(L_m \mathcal{G}_m^{-1} Z_m(s) - I)d\|. \tag{6.21}$$

In the following, we present an algorithm, that allows us to compute a low dimensional dynamical system by computing the matrices $A_{m_{max}} = W_{m_{max}}^T A V_{m_{max}}$, $B_{m_{max}} = W_{m_{max}}^T B$ and $C_{m_{max}} = C V_{m_{max}}$, for a fixed value of m_{max}, where $V_{m_{max}}$ and $W_{m_{max}}$ are the bi-orthonormal matrices obtained by applying Algorithm 12. This algorithm will be called Adaptive Tangential Lanczos Algorithm (ATLA).

Algorithm 13 (Adaptive Tangential Lanczos-type Algorithm ATLA) .

– *Given* A, B, C m_{max}, $s_0^{(1)}$, $s_0^{(2)} \in \mathbb{C}$, r_1, $l_1 \in \mathbb{R}^p$, m_{max}.

– *Outputs :* $A_{m_{max}} = W_{m_{max}}^T A V_{m_{max}}$, $B_{m_{max}} = W_m^T B$, $C_{m_{max}} = C V_{m_{max}}$.

1. *Set* $\sigma_1 = s_0^{(1)}$, $\sigma_2 = s_0^{(2)}$ *and set* $\widetilde{v}_1 = (A - \sigma_1 I_n)^{-1} B r_1$ *and* $\widetilde{w}_1 = (A - \mu_1 I_n)^{-T} C^T l_1$.
2. *Set* $\widetilde{v}_1 = h_{1,1} v_1$ *and* $\widetilde{w}_1 = g_{1,1} w_1$ *such that* $w_1^T v_1 = 1$.
3. *Initialize :* $V_1 = [v_1]$, $W_1 = [w_1]$.
4. *For* $m = 1 : m_{max}$
5. *Set* $A_m = W_m^T A V_m$.
6. *Compute* σ_{m+1}, *and* μ_{m+1}
   - *if* $\bar{\sigma}_{m-1} \neq \sigma_m \in \mathbb{C}$ *then* $\sigma_{m+1} = \bar{\sigma}_m$ *else Compute* $\{\lambda_1, ..., \lambda_m\}$ *eigenvalues of* A_m.
   - *Determine* S_m, *convex hull of* $\{-\lambda_1, ..., -\lambda_m, s_0^{(1)}, s_0^{(2)}, \bar{s}_0^{(1)}, \bar{s}_0^{(2)}\}$.
   - *Solve (6.18). The same for* μ_{m+1}.
7. *Compute right and left vectors* r_{m+1}, l_{m+1} *: Solve (6.19) and (6.21).*
8. *if* $\{\sigma_{m+1} = \infty\}$, $\widetilde{v}_m = A B r_{m+1}$ *else,* $\widetilde{v}_m = (A - \sigma_{m+1} I_n)^{-1} B r_{k+1}$, *End.*
9. *if* $\{\mu_{m+1} = \infty\}$, $\widetilde{w}_m = A C^T l_{m+1}$ *else,* $\widetilde{w}_m = (A - \mu_{m+1} I_n)^{-T} C^T l_{m+1}$, *End.*
10. $h_m = W_m^T \widetilde{v}_m$ *and* $g_m = V_m^T \widetilde{w}_m$.
11. $\widetilde{v}_m = \widetilde{v}_m - V_m h_m$, $\widetilde{w}_m = \widetilde{w}_m - W_m g_m$.
12. $\widetilde{v}_m = h_{m+1,k} v_{m+1}$ *and* $\widetilde{w}_m = g_{m+1,k} w_{m+1}$, *such that* $w_{m+1}^T v_{m+1} = 1$.
13. $V_{m+1} = [V_m, v_{m+1}]$, $W_{m+1} = [W_m, w_{m+1}]$.
14. $W_{m+1}^T V_{m+1} = P_m D_m Q_m^T$ *(Singular Value Decomposition).*
15. $V_{m+1} = V_{m+1} Q_m D_m^{-1/2}$ *and* $W_{m+1} = W_{m+1} P_m D_m^{-1/2}$.
16. *End.*

Remark : For large problems, the total number of arithmetic operations after m_{max} iterations is dominated by $\mathcal{O}(\frac{4}{3}m_{max}n^3)$. This cost comes from LU factorization for solving shifted linear systems with the shifted matrices $(A-\sigma_i I_n)$, $(A-\mu_i I_n)^T$ (Lines 8 and 9 of Algorithm 13). We also need about $\mathcal{O}(2m_{max}pn^2)$ arithmetic operations for computing an SVD decomposition at each iteration $m = 1 : m_{max}$ (Line 7 of Algorithm 13). Compared to the Iterative Rational Krylov Algorithm (IRKA), whose computational complexity is of $\mathcal{O}(\frac{4}{3}m_{max}^2n^3)$, our method is numerically more reliable.

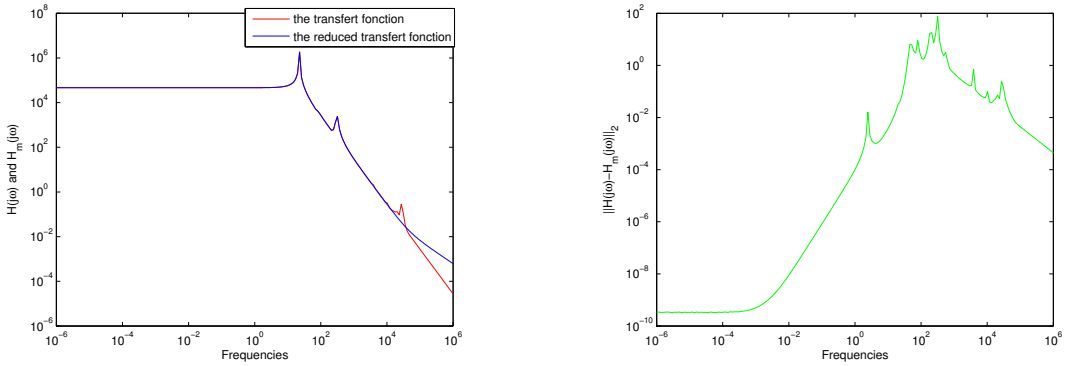
6.4 Numerical experiments

In this section, we give some experimental results to show the effectiveness of the proposed approach. All the experiments were performed on a computer of Intel Core i3 at 1.3GHz and 3GB of RAM. The algorithms were coded in Matlab 8.0. We give some numerical tests to show the performance of the Adaptive Tangential Lanczoc Algorithm (ATLA). These matrix tests used are reported in Table 6.1.

Model	n	p
CD-player	n = 120	p = 2
ISS	n = 277	p = 3
rail821	n = 821	p = 6
MNA4	n = 980	p = 4
rail3113	n = 3113	p = 6
FDM	n = 6400	p = 2

Tableau 6.1 – Matrix Tests

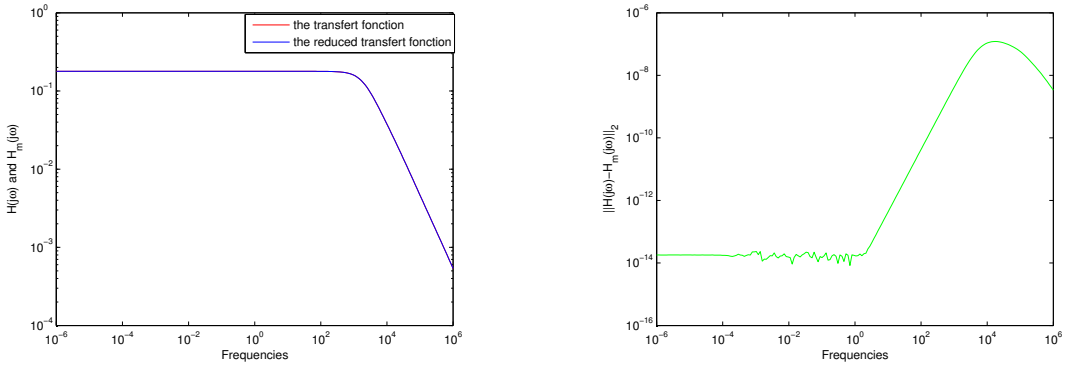
Example 1 : The Figure 6.1a represent the sigma-plot (the singular values of the transfer function) of the original system of a CD-player[25] (red line) and for the reduced order system (blue line). In the Figure 6.1b, we plotted the error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-6}\ 10^6]$. For this experiment, the value of m was $m = 15$.



(a) Singular values vs frequencies. (b) Error-norms vs frequencies.

FIGURE 6.1 – The CDplayer model.

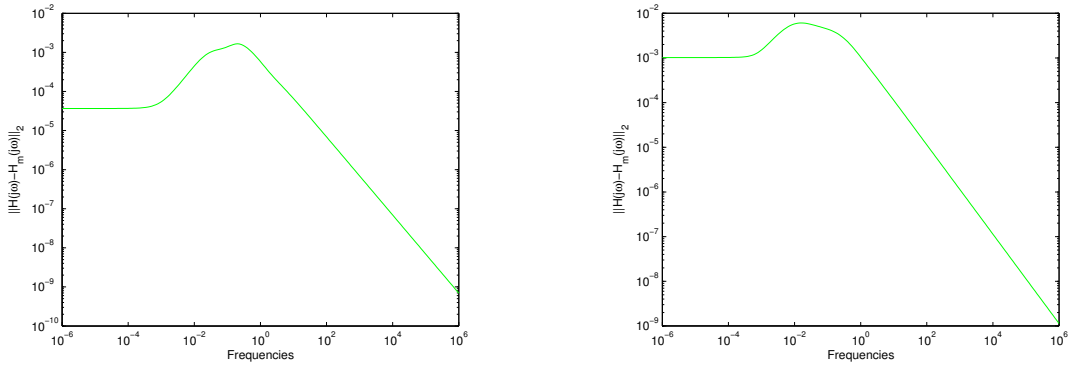
Example 2 : In this example, we applied the ATLA method on the FDM model. The plots in Figure 6.2a, represent the sigma-plot (the singular values of the transfer function) of the original system (red line) and the one of the reduced order system (blue line). In Figure 6.2b, we plotted the error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-6}, 10^{6}]$. For this experiment, the value of m was $m = 20$.



(a) Singular values vs frequencies. (b) Error-norms vs frequencies.

FIGURE 6.2 – The FDM model.

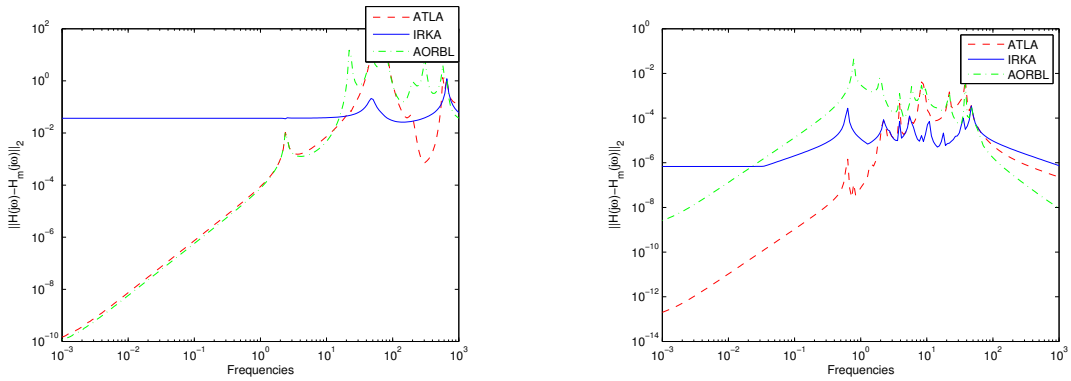
Example 3 : In the third experiment, we used the rail3113 and rail821 models with $m = 20$. The Figures 6.3a and 6.3b represent the exact error-norm $\|H(j\omega) - H_m(j\omega)\|_2$ for different frequencies.



(a) The RAILl821 model. (b) The RAIL3113 model.

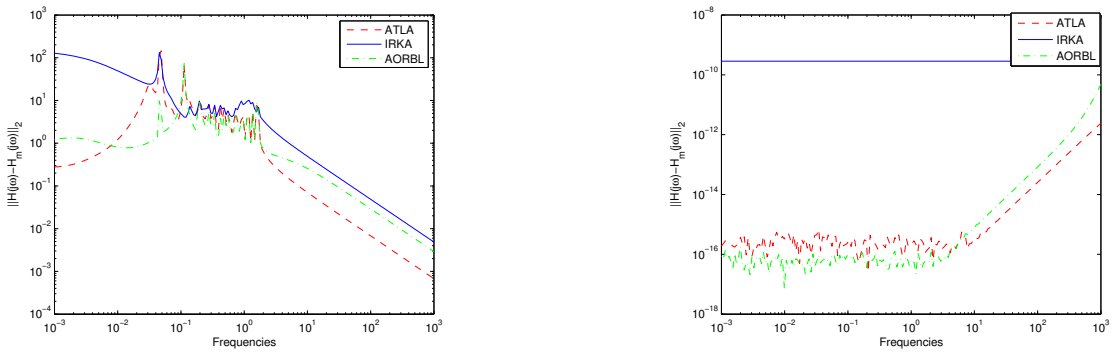
FIGURE 6.3 – Error-norms vs frequencies.

Example 4 : In the last example we compared the ATLA algorithm with AORBL (Adaptive Order Rational Block Lanczos [6]) and IRKA (Iterative Rational Krylov Algorithm [2]). We used four models : MNA_4, ISS, FDM and CDplayer. We compared the exact error $\|H(j\omega) - H_m(j\omega)\|_2$ versus the frequencies $\omega \in [10^{-3}, 10^3]$, for $m = 25$. In all figures, the ATLA method (dashed-dashed line), the AORBL method (dashed-dotted line) and the IRKA (solid line). It is clear that in the all plots, the proposed algorithm gives good approximation as compared to the other algorithms.



(a) The CD-player model. (b) The ISS model.

FIGURE 6.4 – ATLA method (dashed-dashed line), the AORBL method (dashed-dotted line) and the IRKA (solid line).



(a) The MNA_4 model.

(b) The FDM model.

FIGURE 6.5 – ATLA method (dashed-dashed line), the AORBL method (dashed-dotted line) and the IRKA (solid line).

6.5 Conclusion

We have introduced the tangential method, for generating tangential Krylov-type subspace for the reduction of dynamical linear systems, with multiple inputs and multiple outputs (MIMO). The existence of many inputs and outputs, changes the theoretical, and made the computational more difficult. The tangential interpolation, gives sufficient flexibility to overcome the inconvenient, of dimensionality in MIMO problems, where we generate sequences of tangential Krylov subspace by selecting the next interpolation point and tangential direction by maximizing the residual norm, and in the same time we are using the Lanczos approach to bi-orthonormalize the subspaces. The numerical experiments show that the proposed algorithm give good results compared to other known methods.

Chapitre 7

A tangential method for the Balanced Truncation in model reduction

7.1 Introduction

In this chapter, we present a new approach for model reduction in large scale dynamical systems with multiple inputs and multiple outputs (MIMO). This approach is based on the projection of the initial problem onto tangential Krylov subspaces to produce a simpler reduced-order model that approximates well the behaviour of the original model. We present two algorithms named : Adaptive Block Tangential Lanczos-type & Arnoldi-type algorithms (ABTL & ABTA). We give some algebraic properties and present some numerical experiences to show the effectiveness of the proposed algorithms.

Various model reduction techniques, such as Padé approximation [22, 39], balanced truncation [43], optimal Hankel norm [21] and Krylov subspace methods, [11, 10, 17, 28] have been used for large multi-input multi-output (MIMO) dynamical systems, see [5, 21, 27]. Balanced Truncation Model Reduction (BTMR) method is a very popular method [1, 19], 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, 8, 23]. In this paper, we project the Lyapunov equations using the block tangential Krylov subspaces almost similar to the one defined in (4.1),

$$Range\{B, (\sigma_1 I_n - A)^{-1} BR_1, ..., (\sigma_m I_n - A)^{-1} BR_m\},$$

$$Range\{C^T, (\mu_1 I_n - A)^{-T} C^T L_1, ..., (\mu_m I_n - A)^{-T} C^T L_m\},$$

in order to obtain a small scale Lyapunov equations. The $\{\sigma_i\}_{i=1}^m$ and $\{\mu_i\}_{i=1}^m$ are the right and left interpolation points, 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.

7.2 Tangential block Lanczos-type method for Lyapunov matrix equations

Consider the following Lyapunov matrix equations,

$$AX^{(1)} + X^{(1)}A^T + BB^T = 0, \tag{7.1}$$

and

$$A^T X^{(2)} + X^{(2)}A + C^T C = 0, \tag{7.2}$$

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 equations (7.1) and (7.2), we project the initial problems onto the following tangential block Krylov subspaces

$$\widetilde{\mathcal{K}}_m(A, B) = Range\{B, (\sigma_1 I_n - A)^{-1} BR_1, ..., (\sigma_m I_n - A)^{-1} BR_m\}, \tag{7.3}$$

and

$$\widetilde{\mathcal{K}}_m(A^T, C^T) = Range\{C^T, (\mu_1 I_n - A)^{-T} C^T L_1, ..., (\mu_m I_n - A)^{-T} C^T L_m\}, \tag{7.4}$$

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 $\widetilde{\mathcal{K}}_m(A, B)$ and $\widetilde{\mathcal{K}}_m(A^T, C^T)$, 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 $i, j = 1, \ldots, m$:

$$\begin{cases} W_i^T V_j = I, & i = j, \\ W_i^T V_j = 0, & i \neq j. \end{cases} \tag{7.5}$$

The low rank approximate solutions $\mathcal{X}_m^{(1)}$ and $\mathcal{X}_m^{(2)}$ to the solutions of the Lyapunov matrix equations (7.1) and (7.2), are defined as follows

$$\mathcal{X}_m^{(1)} = \mathbb{V}_m \mathcal{Y}_m^{(1)} \mathbb{V}_m^T, \qquad \mathcal{X}_m^{(2)} = \mathbb{W}_m \mathcal{Y}_m^{(2)} \mathbb{W}_m^T, \tag{7.6}$$

such that the following Galerkin conditions are satisfied

$$\mathbb{W}_m^T \mathcal{R}_1(\mathcal{X}_m^{(1)}) \mathbb{W}_m = 0, \qquad \mathbb{V}_m^T \mathcal{R}_2(\mathcal{X}_m^{(2)}) \mathbb{V}_m = 0, \tag{7.7}$$

where the residuals are given by

$$\mathcal{R}_1(\mathcal{X}_m^{(1)}) = A\mathcal{X}_m^{(1)} + \mathcal{X}_m^{(1)} A^T + BB^T, \tag{7.8}$$

and

$$\mathcal{R}_2(\mathcal{X}_m^{(2)}) = A^T \mathcal{X}_m^{(2)} + \mathcal{X}_m^{(2)} A + C^T C. \tag{7.9}$$

Replacing $\mathcal{X}_m^{(1)}$ and $\mathcal{X}_m^{(2)}$ in (7.7), we obtain

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

and

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

which gives the low-dimensional Lyapunov matrix equations

$$A_m \mathcal{Y}_m^{(1)} + \mathcal{Y}_m^{(1)} A_m^T + B_m B_m^T = 0, \tag{7.10}$$

and

$$A_m^T \mathcal{Y}_m^{(2)} + \mathcal{Y}_m^{(2)} A_m + C_m^T C_m = 0, \tag{7.11}$$

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 (7.3) and (7.4). The following Block Tangential Lanczos (BTL) algorithm allows us to construct such bases. It is summarized in the following steps.

Algorithm 14 (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 :* $\mathbb{V}_{m+1} = [V_1, ..., V_{m+1}]$, $\mathbb{W}_{m+1} = [W_1, ..., W_{m+1}]$.

- *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* $\widetilde{V}_{j+1} = ABR_j$.
  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* $i = 1,...,j$
     - $H_{i,j} = W_i^T \widetilde{V}_{j+1}$, $\quad - F_{i,j} = V_i^T \widetilde{W}_{j+1}$,
     - $\widetilde{V}_{j+1} = \widetilde{V}_{j+1} - V_i H_{i,j}$, $\quad - \widetilde{W}_{j+1} = \widetilde{W}_{j+1} - W_i F_{i,j}$,
  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. $\mathbb{V}_{j+1} = \left[\mathbb{V}_j,\ V_{j+1}\right]$, $\quad \mathbb{W}_{j+1} = \left[\mathbb{W}_j,\ W_{j+1}\right]$.
- *End*

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

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

The matrices $H_{i,j}$ and $F_{i,j}$ constructed in Step 3 of Algorithm 14 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

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

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

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

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

Theorem 7.1 *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 14. Then we have the following relations*

$$A\mathbb{V}_{m+1} = \left[\mathbb{V}_{m+1}\mathbb{G}_{m+1}\widetilde{D}_m^{(1)} - \mathbb{K}_{m+1}^B\right]\mathbb{G}_{m+1}^{-1}, \tag{7.12}$$

and

$$A^T\mathbb{W}_{m+1} = \left[\mathbb{W}_{m+1}\mathbb{Q}_{m+1}\widetilde{D}_m^{(2)} - \mathbb{K}_{m+1}^C\right]\mathbb{Q}_{m+1}^{-1}. \tag{7.13}$$

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

$$\mathbb{T}_{m+1} = \left[B, (\sigma_1 I - A)^{-1}BR_1, ..., (\sigma_m I - A)^{-1}BR_m\right] \text{ and}$$

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

then we have

$$\mathbb{T}_{m+1} = \mathbb{V}_{m+1}\mathbb{G}_{m+1} \quad \text{and} \quad \mathbb{Y}_{m+1} = \mathbb{W}_{m+1}\mathbb{Q}_{m+1}, \tag{7.14}$$

where $\mathbb{K}^B_{m+1} = [-AB \;\; B\mathbb{R}_m]$, $\mathbb{K}^C_{m+1} = \left[-A^T C^T \;\; C^T \mathbb{L}_m\right]$, $\mathbb{R}_m = [R_1, ..., R_m]$ *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 nonsingular.*

Proof : From Algorithm 14, we have

$$V_{j+1}H_{j+1,j} = (\sigma_j I_n - A)^{-1} BR_j - \sum_{i=1}^{j} V_i H_{i,j} \qquad j = 1, ..., m. \tag{7.15}$$

Multiplying (7.15) on the left by $(\sigma_j I_n - A)$ and re-arranging terms, we get

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

which gives

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

that written as

$$A\mathbb{V}_{m+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} = \sigma_j \mathbb{V}_{j+1}\begin{bmatrix} H_{1,j} \\ \vdots \\ H_{j,j} \\ H_{j+1,j} \\ \mathbf{0} \end{bmatrix} - BR_j, \qquad j = 1, \ldots, m, \tag{7.16}$$

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

$$A\mathbb{V}_{m+1}\widetilde{\mathbb{H}}^{(j)} = \sigma_j \mathbb{V}_{j+1}\widetilde{\mathbb{H}}^{(j)} - BR_j, \tag{7.17}$$

Therefore, we can deduce from (7.17), the following expression

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

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

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

which ends the proof of (7.12). The same proof can be done for the relation (7.13).

For the proof of (7.14), we first use (7.15) to obtain

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

which gives

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

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 (7.14). 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 7.2 *Let* $\mathbb{V}_m = [V_1, ..., V_m]$ *and* $\mathbb{W}_m = [W_1, ..., W_m]$ *be the bi-orthonormal matrices obtained by Algorithm 14. 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 equations* (7.1) *and* (7.2), *then the residuals norms are given as*

$$\| \mathcal{R}_1(\mathcal{X}_m^{(1)}) \|_2 = \| \mathbb{S}_m^{(1)} J (\mathbb{S}_m^{(1)})^T \|_2 \text{ and } \| \mathcal{R}_2(\mathcal{X}_m^{(2)}) \|_2 = \| \mathbb{S}_m^{(2)} J (\mathbb{S}_m^{(2)})^T \|_2, \tag{7.18}$$

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*

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

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

Proof : We know that

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

Using the equation (7.12), we get

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

which gives

$$A\mathbb{V}_m = \mathbb{V}_m A_m + (\mathbb{V}_m \mathbb{W}_m^T - I_n) \mathbb{K}_m^B \mathbb{G}_m^{-1}. \tag{7.19}$$

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 equation (7.10),

we get

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

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

7.3 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, $L_i \in \mathbb{R}^{p\times s}$. here we adopted the same approach used in the previous chapters, inspired by the work in [14]. We recall that 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

$$\widetilde{V}_{m+1} = (\sigma_{m+1}I_n - A)^{-1}BR_{m+1}, \quad \text{and} \quad \widetilde{W}_{m+1} = (\sigma_{m+1}I_n - A)^{-T}C^TL_{m+1}, \tag{7.20}$$

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

$$(R_{m+1},\sigma_{m+1}) = arg \max_{\substack{\omega\in S_m\\ R\in\mathbb{R}^{p\times s}\\ \|R\|_2=1}} \|R_B(\omega)R\|_2, \tag{7.21}$$

$$(L_{m+1},\mu_{m+1}) = arg \max_{\substack{\omega\in S_m\\ L\in\mathbb{R}^{p\times s}\\ \|L\|_2=1}} \|R_C(\omega)L\|_2. \tag{7.22}$$

Where

$$R_B(\omega) = B - (\omega I_n - A)V_m(\omega I_m - A_m)^{-1}W_m^TB, \tag{7.23}$$

$$R_C(\omega) = C^T - (\omega I_n - A)^TW_m(\omega I_m - A_m)^{-T}V_m^TC^T. \tag{7.24}$$

For solving the problem (7.21), we proceed in same way as in the previous chapters, i.e we solve first the following problem,

$$\sigma_{m+1} \quad = arg\max_{\omega \in S_m} \|R_B(\omega)\|_2. \tag{7.25}$$

Then the tangent direction R_{m+1} is computed by evaluating (7.21) at $\omega = \sigma_{m+1}$,

$$R_{m+1} = arg \max_{\substack{R \in \mathbb{R}^{p \times s} \\ \|R\|_2 = 1}} \|R_B(\sigma_{m+1})R\|_2. \tag{7.26}$$

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

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

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 7.1 *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 (7.23) and (7.24), 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*

$$R_B(\omega) = (\mathbb{V}_m \mathbb{W}_m^T - I_n)\mathbb{K}_m^B \mathbb{G}_m^{-1} U_m^B(\omega), \tag{7.27}$$

and

$$R_C(\omega) = (\mathbb{W}_m \mathbb{V}_m^T - I_n)\mathbb{K}_m^C \mathbb{Q}_m^{-1} U_m^C(\omega). \tag{7.28}$$

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

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

Since $B \in Range\{V_1, ...V_m\}$, then $(I_n - \mathbb{V}_m \mathbb{W}_m^T)B = 0$. Using Equation (7.19), we get

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

which proves (7.27). In the same way we can prove (7.28).
The expression of $R_B(\omega)$ given in Proposition (7.1) allows us to significantly reduce the computational cost while seeking the next pole and direction, by applying the skinny QR decomposition

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

we get the simplified residual norm

$$\|R_B(\omega)\|_2 = \left\|LU_m^B(\omega)\right\|_2. \tag{7.29}$$

This means that, solving the problem (7.21) 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.

Algorithm 15 (The Adaptive Block Tangential Lanczos (ABTL) algorithm) .
– *Given* A, B, C, m_{max}, ϵ.
– *Outputs :* $Z_m^{(1)}$, $Z_m^{(2)}$.

1. *Set* $B = H_{1,0}V_1$ *and* $C^T = F_{1,0}W_1$ *such that* $W_1^T V_1 = I_p$.
2. *Initialize :* $\mathbb{V}_1 = [V_1]$, $\mathbb{W}_1 = [W_1]$.
3. *For* $m = 1 : m_{max}$
4. *Set* $A_m = \mathbb{W}_m^T A \mathbb{V}_m$, $B_m = \mathbb{W}_m^T B$, $C_m = C\mathbb{V}_m$.
5. *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 (7.25). The same for* μ_m.
6. *Compute right and left vectors* R_m, L_m.
7. $\widetilde{V}_m = (\sigma_m I_n - A)^{-1} B R_m, \quad \widetilde{W}_m = (\mu_m I_n - A)^{-T} C^T L_m.$

8. *For i = 1,...,m*
   - $H_{i,m} = W_i^T \widetilde{V}_{m+1}$, $\quad - F_{i,m} = V_i^T \widetilde{W}_{m+1}$,
   - $\widetilde{V}_{m+1} = \widetilde{V}_{m+1} - V_i H_{i,m}$, $\quad - \widetilde{W}_{m+1} = \widetilde{W}_{m+1} - W_i F_{i,m}$,
9. *End.*
10. $\widetilde{V}_{m+1} = V_{m+1} H_{m+1,m}$, $\widetilde{W}_{m+1} = W_{m+1} F_{m+1,m}$. *(QR decomposition).*
11. $W_{m+1}^T V_{m+1} = P_m D_m Q_m^T$. *(Singular Value Decomposition).*
12. $V_{m+1} = V_{m+1} Q_m D_m^{-1/2}$, $W_{m+1} = W_{m+1} P_m D_m^{-1/2}$.
13. $H_{m+1,m} = D_m^{1/2} Q_m^T H_{m+1,m}$, $F_{m+1,m} = D_m^{1/2} P_m^T F_{m+1,m}$.
14. $\mathbb{V}_{m+1} = [\mathbb{V}_m,\ V_{m+1}]$, $\quad \mathbb{W}_{m+1} = [\mathbb{W}_m,\ W_{m+1}]$.
15. *Solve* (7.10) *and* (7.11) *to get* $\mathcal{Y}_m^{(1)}$ *and* $\mathcal{Y}_m^{(2)}$.
16. *If* $max(\| \mathcal{R}_1(\mathcal{X}_m^{(1)}) \|_2, \| \mathcal{R}_2(\mathcal{X}_m^{(2)}) \|_2) < \epsilon$ *Stop.*
17. *End.*
18. *Compute* $Z_m^{(1)}$, $Z_m^{(2)}$ *as in* (7.30).

In order to save memory, Algorithm 15 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

$$Z_m^{(1)} = \mathbb{V}_m \widetilde{U}_1 \Lambda_1^{\frac{1}{2}} \qquad Z_m^{(2)} = \mathbb{W}_m \widetilde{V}_1 \Gamma_1^{\frac{1}{2}}. \tag{7.30}$$

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 $\Lambda = diag(\Lambda_1, \Lambda_2)$, $\Gamma = diag(\Gamma_1, \Gamma_2)$ verify $max(diag(\Lambda_1)) > dtol$ and $max(diag(\Gamma_1)) > dtol$ for some given tolerance $dtol$.

7.4 Adaptive Block Tangential Arnoldi (ABTA) algorithm

Consider the following Lyapunov equation,

$$AX + XA^T + BB^T = 0. \tag{7.31}$$

The solution X is approximated by $\mathcal{X}_m$ such that

$$\mathcal{X}_m = \mathbb{V}_m \mathcal{Y}_m \mathbb{V}_m^T, \tag{7.32}$$

and satisfying the Galerkin condition,

$$\mathbb{V}_m^T \mathcal{R}(\mathcal{X}_m) \mathbb{V}_m = 0, \tag{7.33}$$

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 $\mathcal{V}_m = Range\{V_1, V_2, \ldots, V_m\}$ constructed from the following tangential subspace $Range\{B, (\sigma_1 I_n - A)^{-1} BR_1, ..., (\sigma_m I_n - A)^{-1} BR_m\}$. From (7.32) and (7.33) , $\mathcal{Y}_m$ is obtained by solving the low-dimensional Lyapunov matrix equation

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

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 7.3 *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 Lyupunov matrix equation, then the residual norm is given as*

$$\| \mathcal{R}(\mathcal{X}_m) \|_2 = \| \mathbb{S}_m J \mathbb{S}_m \|_2, \tag{7.34}$$

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

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

Proof : The proof is similar to the one given in the proof of Theorem 7.3.
The choice of the interpolation points and tangent directions is the same as in the previous section,

$$(R_{m+1}, \sigma_{m+1}) = arg \max_{\substack{\omega \in S_m \\ R \in \mathbb{R}^{p \times s} \\ \|R\|_2 = 1}} \|R_B(\omega) R\|_2. \tag{7.35}$$

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.

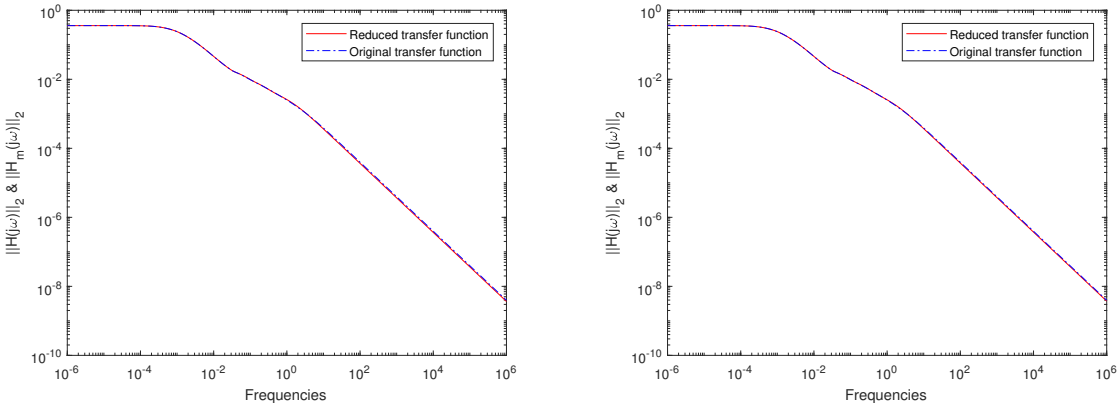
Algorithm 16 (The Adaptive Block Tangential Arnoldi (ABTA) algorithm) .
– Inputs A, B, m_{max}, ϵ and $dtol$.

1. *Set $B = H_{1,0}V_1$ and initialize : $\mathbb{V}_1 = [V_1]$.*
2. *For $m = 1 : m_{max}$*
3. *Set $A_m = \mathbb{V}_m^T A \mathbb{V}_m$, $B_m = \mathbb{V}_m^T B$.*
4. *Compute the interpolation points σ_m and the directions R_m.*
5. *$\widetilde{V}_m = (\sigma_m I_n - A)^{-1} B R_m$.*
6. *For $i = 1,...,m$*
   — *$H_{i,m} = V_i^T \widetilde{V}_{m+1}$,*
   — *$\widetilde{V}_{m+1} = \widetilde{V}_{m+1} - V_i H_{i,m}$,*
7. *End.*
8. *$\widetilde{V}_{m+1} = V_{m+1}H_{m+1,m}$. (QR decomposition).*
9. *$\mathbb{V}_{m+1} = [\mathbb{V}_m,\ V_{m+1}]$.*
10. *Compute the approximate $\mathcal{Y}_m$.*
11. *If $\| \mathcal{R}(\mathcal{X}_m) \|_2 < \epsilon$, stop.*
12. *End.*
13. *Compute $Z_{m_{max}}$.*

7.5 Numerical experiments

In this section, we present some numerical examples to show the effectiveness of the adaptive block tangential Arnoldi & Lanczos-types algorithms (ABTA & 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.

Example 1 : In this first experiment, we used the rail3113 model (n=3113, p=6). Figures 7.1a and 7.1b 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.

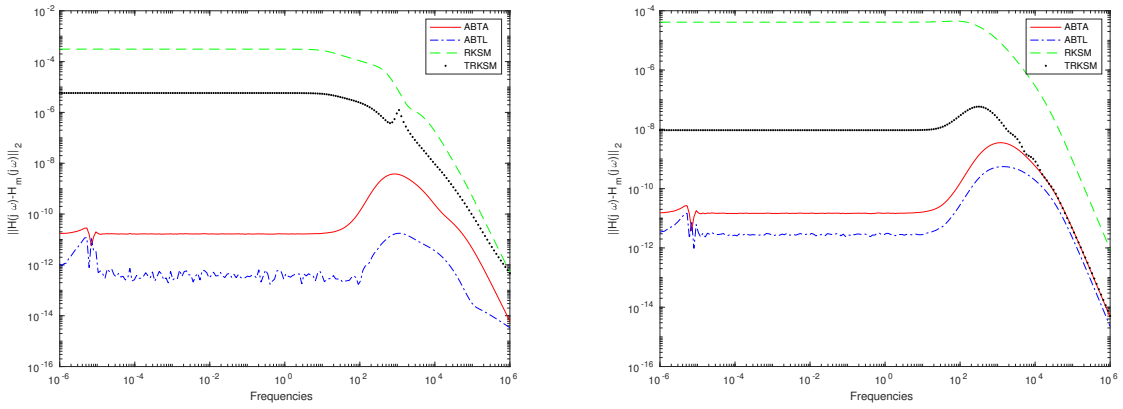


(a) ABTL method. (b) ABTA method.

FIGURE 7.1 – The RAIL3113 model : n=3113, p=6.

Example 2 : In this part, we compared the ABTA & 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 [13] and (TRKSM) where the tangential approach is used [14]. The Matlab implementations of (RKSM) and (TRKSM) have been downloaded from Simoncini's web page [1]. We used the FDM model. 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. Plots in Figures 7.2a and 7.2b, 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 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 Figure 7.2b represent the same thing but with the matrix $C = B^T$.

1. http ://www.dm.unibo.it/ simoncin/software.html



(a) B=rand(n,p), C=rand(p,n). (b) B=rand(n,p), $C = B^T$.

FIGURE 7.2 – The FDM model : n=4900, p=6.

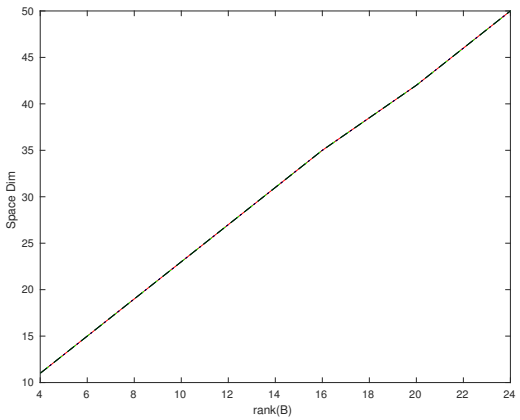
We present below Table 7.1 that gives more information about the plots in Figures 7.2a and 7.2b, where we present the execution time, the maximum space dimension, the rank dimension, the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ error norms obtained by each method. The maximum space dimension is the the dimension of the matrices obtained after the stopping tolerance $tol = 10^{-8}$ and the rank dimension is dimension obtained as in (7.30).

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

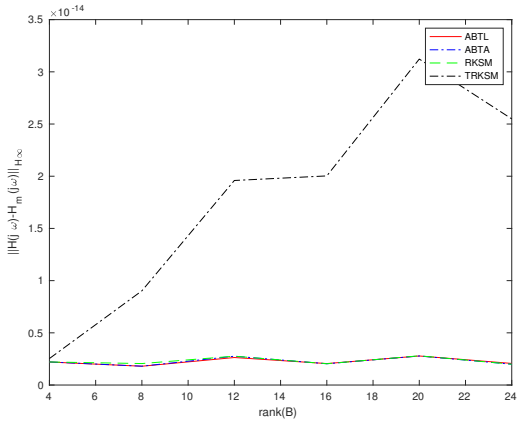
TABLEAU 7.1 – The calculation time and dimension of convergence

Example 3 : In this example we compared, the Rank dimension (Figure 7.3a), the Err-$\mathcal{H}_\infty$ norm (Figure 7.3b) and the execution time (Figure 7.4) as p the rank of the matrix B is grown. In all plots of Figures (7.3a) to (7.4), the ABTL method (solid line), the ABTA method (dashed-dotted line), the RKSM (dashed-dashed

line) and TRKSM (black dashed-dotted line). We used the FDM model of size n=10000 and the matrix $B = A^{-1}I_p$ with p ranging from 4 to 24.



(a) The rank dimension.



(b) The $\mathcal{H}_\infty$-err norm.

FIGURE 7.3 – The FDM model : n=22500, p=6.

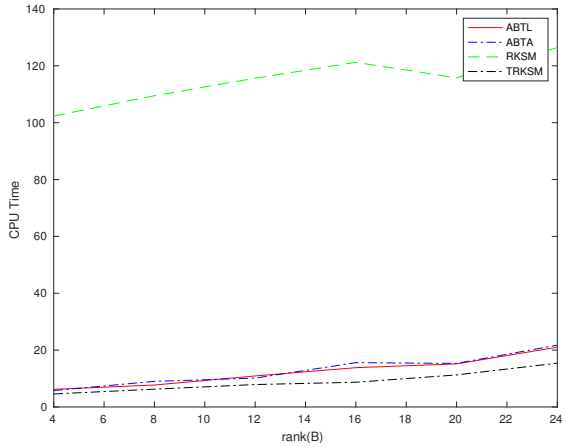


FIGURE 7.4 – FDM model : The execution time

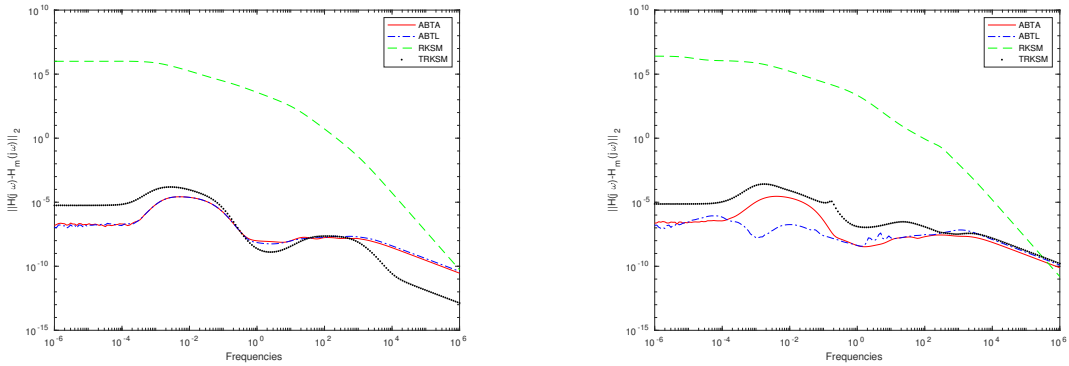
Figure 7.3a shows that the four methods have the same rank dimension and is growing with as p grows, in Figure 7.3b we notice that our both methods and RKSM give better $\mathcal{H}_\infty$-err norm than TRKSM method, but in Figure 7.4 we can see clearly that TRKSM have the best execution time, followed by our both methods, and finally RKSM 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 in table 7.2 and implemented in Figure 7.5a and 7.5b. 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 the longest execution time.

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

Tableau 7.2 – The execution time and dimension of convergence

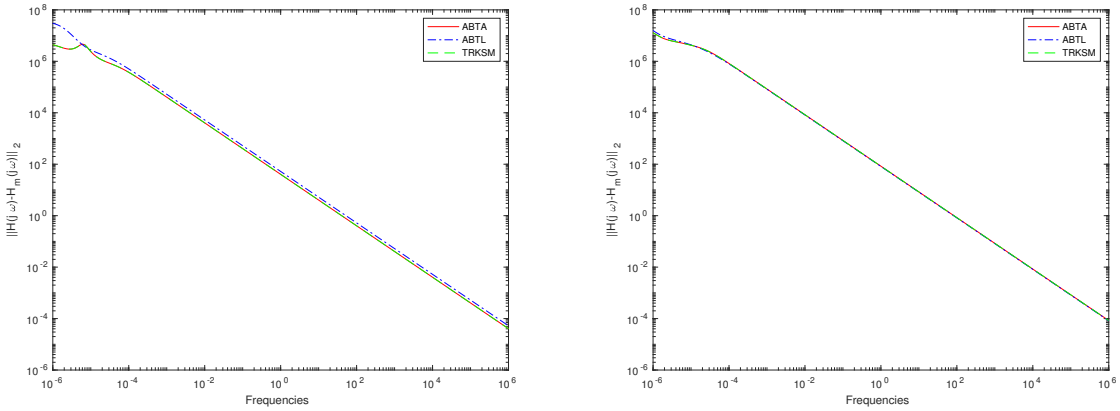


(a) B=rand(n,p), $C=B^T$. (b) B=rand(n,p), C=rand(p,n).

Figure 7.5 – The Flow model : n=9669, p=6.

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}.



(a) The RAIL20209 model. (b) The RAIL79841 model.

FIGURE 7.6 – $\|H(j\omega) - H_m(j\omega)\|_2$ vs the frequencies.

Figure 7.6a 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 seconds, ABTL : 9.76 seconds, ABTA : 10.45 seconds). Figure 7.6b 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 seconds, ABTA : 99.45 seconds, TRKSM : 119.76 seconds).

Example 6 : In this last experiment, we present the Table 7.3 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 & 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.

7.6 Conclusion

In the present chapter, we proposed a new approach based on block tangential Krylov subspaces to compute low rank approximate solutions to large Lyapunov

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

Tableau 7.3 – The execution time and dimension of convergence

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 gave some new algebraic properties and compared our algorithms with well knowing methods to show the effectiveness of this latter.

Chapitre 8

Conclusions and perspectives

In this chapter, we provide a brief review of the main results on model order reduction of large-scale dynamical systems described in the previous chapters. Moreover we give some suggestions on future works that can develop new techniques for model reduction.

8.1 Summary of result

The major contributions of this thesis are the development of new interpolation techniques for the reduction of dynamical linear systems, with multiple inputs and multiple outputs (MIMO), using projection methods onto tangential Krylov subspaces. The different techniques developed in this thesis are summarized as the following :

- In chapter 4, we proposed the adaptive block tangential Arnoldi method, where we used the projection onto the block tangential krylov subspace, this latter contain tangential matrix directions and interpolation points obtained using the adaptive technique, and we generate an orthonormal subspace from the tangential krylov subspace using the Arnoldi-type algorithm.
- In chapter 5, we have based on the global approach that used the Kronecker and the diamond products. This technique allows us the devellope a new algorithm called : Adaptive Global Tangential Arnoldi Algorithm

(AGTAA) . New theorems and alegebriac properties are presented, and the numerical experiments show the efectivness of this new approch.

- In chapter 6, we used the oblique projection for the original system onto the tangential krylov subspaces and his dual. Then Lanczos-type procedure is used to generate two bi-orthonormal subspaces. Tha adaptive approach is used for the selecting tof he next interpolation point and tangential direction by maximizing the residual norm. The method proposed here named : Adaptive Tangential Lanczos-type Algorithm (ATLA).
- In chapter 7, we projected the large Lyapunov equations onto block tangential Krylov subspaces, to compute low rank approximate solutions. These latter are used to build reduced order models that approximate the initial large scale dynamical systems with multiple inputs and multiple outputs (MIMO). We used the adaptive for the choice of the interpolation points and the matrix tangential directions, and we used two approaches for generating the projection bases. We present two algorithms named : Adaptive Block Tangential Lanczos-type & Arnoldi-type algorithms (ABTL & ABTA).

8.2 Perspectives

The main of this thesis was to provide efficient algorithms using the tangential Krylov subspaces for the reduction of Linear Time-Invariant (LTI) large-scale dynamical systems. One of the perspectives is to apply the techniques developed in this dissertation, on a linear in state and parameterized dynamical systems, with d parameters $p = [p_1, ..., p_d]^T \in \Omega \subset \mathbb{R}^d$ (usually, Ω is a bounded domain) defined as :

$$\begin{cases} E(p)\dot{x}(t,p) &= A(p)x(t,p) + B(p)u(t) \\ y(t,p) &= C(p)x(t,p) \end{cases} \quad with \quad x(0,p) = 0,$$

where $t \in [0, \infty)$. The state-vector is denoted by $x(t,p) \in \mathbb{R}^n$. $u(t) \in \mathbb{R}^m$ and $y(t,p) \in \mathbb{R}^q$ denote, respectively, the inputs (excitations) and outputs (observations or measurements) of the model. The state-space matrices, then, have the dimensions

$E(p), A(p) \in \mathbb{R}^{n\times n}$, $B(p) \in \mathbb{R}^{n\times m}$, and $C(p) \in \mathbb{R}^{q\times n}$. Another futures possibility, is reducing or searching for a low-dim of non-linear dynamical system defined in (1.1) as :

$$\begin{cases} \dot{x}(t) &= f(t, x(t), u(t)) \\ y(t) &= g(t, x(t), u(t)) \end{cases}$$

by using the projection onto the tangential and the rational Krylov subspaces.

Bibliographie

[1] A. C. Antoulas. « Approximation of large-scale dynamical systems ». In : *SIAM Adv. Des. Contr.* 29 (2005), p. 181–190.

[2] C. A. Antoulas et C. A. Beattie. « Interpolatory model reduction of large-scale dynamical systems ». In : *Effic. Mod. Contr. Syst.* (2010), p. 3–58.

[3] C. A. Antoulas, D. C. Sorensen et S. Gugercin. « A survey of model reduction methods for large scale systems, Contemporary Mathematic ». In : *AMS Publi.* (2001), p. 193–219.

[4] C. Athanasios, C. A. Antoulas et D. C. Sorensen. « Approximation of large-scale dynamical systems ». In : *J. Appl. Math. Comput. Sci.* 11 (2001), p. 1093–1121.

[5] Z. Bai. « Krylov subspace techniques for reduced-order modeling of large scale dynamical systems ». In : *Appl. Numer. Math.* 43 (2002), p. 9–44.

[6] H. Barkouki, A. H. Bentbib et K. Jbilou. « An adaptive rational block lanczos-type algorithm for model reduction oflarge scale dynamical systems ». In : *Journ. of Scien. Compu.* 67 (2015), p. 221–236.

[7] C. A. Beattie et S. Gugercin. « Interpolation projection methods for structure preserving model reduction ». In : *Syst. Contr. Lett.* 58 (2009), p. 225–232.

[8] P. Benner, J. Li et T. Penzl. « Numerical solution of large Lyapunov equations, Riccati equations and linear-quadratic optimal control problems ». In : *Numer. Lin. Alg. Appl.* 15 (2008), p. 755–777.

[9] R. Bouyouli et al. « Convergence properties of some block Krylov subspace methods for multiple linear systems ». In : *J. Comput. Appl. Math.* 196 (2006), p. 498–511.

[10] B. N. Datta. « Krylov subspace methods for large-scale matrix problems in control ». In : *Future Gener. Comput. Syst.* 19 (2003), p. 1253–1263.

[11] B. N. Datta. « Large-Scale Matrix computations in Control ». In : *Appl. Numer. Math.* 30 (1999), p. 53–63.

[12] V. Druskin, C. Lieberman et M. Zaslavsky. « On adaptive choice of shifts in rational Krylov subspace reduction of evolutionary problems ». In : *SIAM J. Sci. Comput.* 32 (2010), p. 2485–2496.

[13] V. Druskin et V. Simoncini. « Adaptive rational Krylov subspaces for large-scale dynamical systems ». In : *Syst. Contr. Lett.* 60 (2011), p. 546–560.

[14] V. Druskin, V. Simoncini et M. Zaslavsky. « Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems ». In : *SIAM. J. Matrix Anal. Appl.* 35 (2014), p. 476–498.

[15] B. F. Farrell et P. J. Ioannou. « Stochastic dynamics of the mid-latitude atmospheric jet ». In : *J. Atmo. Sci.* 52 (1995), p. 1642–1656.

[16] P. Feldmann et R. Freund. « Efficient linear circuit analysis by Pade approximation via the Lanczos process ». In : *IEEE Trans. Comp. Desi. Integr. Circ. Sys.* 14 (1995), p. 639–649.

[17] M. Frangos et I. M. Jaimoukha. « Adaptive rational interpolation : Arnoldi and Lanczos-like equations ». In : *Appl. Numer. Math.* 14 (2008), p. 342–354.

[18] K. Gallivan, A. Vandendorpe et P. VanDooren. « Sylvester equations and projection-based model reduction ». In : *J. Comp. Appl. Math.* 162 (2004), p. 213–229.

[19] K. Glover. « All optimal Hankel-norm approximations of linear multivariable systems and their L_1-errors bounds ». In : *Internat. J. Contr.* 39 (1984), p. 1115–1193.

[20] K. Glover et R. A. Roberts. « All optimal Hankel-norm approximations of linear multi-variable systems and their L,∞-error bounds ». In : *Inter. Jour. Cont.* 39 (1984), p. 1115–1193.

[21] K. Glover et al. « A characterization of all solutions to the four block general distance problem ». In : *SIAM J. Contr. Optim.* 29 (1991), p. 283–324.

[22] G. Grimme. « Krylov Projection methods for model reduction ». Thèse de doct. University of Illinois, 1997.

[23] S. Gugercin et A. C. Antoulas. « A survey of model reduction by balanced truncation and some new results ». In : *Inter. Jour. Cont.* 77 (2004), p. 748–766.

[24] S. Gugercin, A. Antoulas et C. Beattie. « A rational Krylov iteration for optimal $\mathcal{H}_2$ model reduction ». In : *J. Comp. Appl. Math.* 53 (2006), p. 1665–1667.

[25] S. Guttel et L. Knizhnerman. « Automated parameter selection for rational Arnoldi approximation of Markov functions ». In : *Proc. Appl. Math. Mecha.* 11 (2011), p. 15–18.

[26] K. Hestenes et E. Stiefel. « Methods of conjugate gradients for solving linear systems ». In : *J. Res. National Bur. Stand.* 49 (1952), p. 409–435.

[27] M. Heyouni et K. Jbilou. « Matrix Krylov subspace methods for large scale model reduction problems ». In : *App. Math. Comput.* 181 (2006), p. 1215–1228.

[28] I. M. Jaimoukha et E. M. Kasenally. « Krylov subspace methods for solving large Lyapunov equations ». In : *SIAM J. Matrix Anal. Appl.* 31 (1994), p. 227–251.

[29] A. N. Krylov. « On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined ». In : *Izvestija AN SSSR (News of Academy of Sciences of the USSR), Otdel. mat. i estest. nauk* 4 (1931), p. 491–539.

[30] C. Lanczos. « An iteration method for the solution of the eigenvalue problem of linear differential and integral operators ». In : *J. Rex Nat. Bur. Stand.* 45 (1950), p. 255–282.

[31] C. Lanczos. « Solution of systems of linear equations by minimized iterations ». In : *J. Rex Nat. Bur. Stand.* 49 (1952), p. 33–53.

[32] V. Merhrmann et T. Penzl. « Benchmark collections in SLICOT. Technical report SLWN, SLICOT working note, ESAT, KU Leuven, K. Mercierlaan 94, Leuven-Heverlee 3100, Belgium, 1998. » In : (1847). url : http://www.win.tue.nl/niconet/NIC2/reports.html.

[33] B. C. Moore. « Principal component analysis in linear systems : controllability, observability and model reduction ». In : *IEEE Trans. Automatic Contr.* 26 (1981), p. 17–32.

[34] B. C. Moore. « Principal component analysis in linear systems : controllability, observability, and model reduction ». In : *IEEE. Trans. Automatic Control.* 26 (1981), p. 17–32.

[35] C. T. Mullis et R. A. Roberts. « Round off noise in digital filters : Frequency transformations and invariants ». In : *IEEE Trans. Acoust. Speec Signal Process.* 24 (1976), p. 538–550.

[36] A. Odabasioglu, M. Celik et L. T. Pileggi. « PRIMA : passive reduced-order interconnect macromodeling algorithm ». In : *IEEE Trans. Compu. Desi. Integ. Circ. Syst.* 17 (1998), p. 645–654.

[37] T. Penzl. « LYAPACK MATLAB toolbox for Large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems ». In : (2000). url : http://www.tuchemintz.de/sfb393/lyapack.

[38] Y. Saad. *Iterative methods for solving sparse linear systems*. Boston : PWS Publishing Compagny, 1996.

[39] K. G. Safonov et R. Y. Chiang. « Schur method for balanced-truncation model reduction ». In : *IEEE Trans. Automat. Contr.* 34 (1989), p. 729–733.

[40] Y. Shamash. « Stable reduced-order models using Padé type approximations ». In : *IEEE. Trans. Automatic Control.* 19 (1974), p. 615–616.

[41] P. Spalart et al. « Comments on the feasibility of LES for wings, and on a hybrid ». In : *Conference : Advances in DNS/LES* (1997).

[42] F. Tröltzsch et A. Unger. « Fast solution of optimal control problems in the selective cooling of steel ». In : *Z. Angew. Math. Mech.* 81 (2001), p. 447–456.

[43] P. VanDooren, K. A. Gallivan et P. Absil. « $\mathcal{H}_2$-optimal model reduction with higher order poles ». In : *SIAM. J. Matrix Anal. Appl.* 31 (2010), p. 2738–2753.

[44] P. Wortelboer, M. Steinbuch et O. Bosgra. « Closed-Loop balanced reduction with application to a compact disc mechanism ». In : *Selec. Topi. Identi. Mod. Contr.* 9 (1996), p. 47–58.

[45] K. Zhou, J. C. Doyle et K. Glover. *Robust and optimal control*. Pentice Hall, 1995.

Published and submitted papers

List of papers :

- A. H. Bentbib, K. Jbilou and Y. Kaouane, "A Computational Global Tangential Krylov Subspace Method for Model Reduction of Large-Scale MIMO Dynamical Systems". Journal of Scientific Computing, 75(2018), pp. 1614–1632.

- A. H. Bentbib, K. Jbilou and Y. Kaouane, An adaptive tangential Lanczos-type method for model reduction in large-scale dynamical systems. Accepted in Applied and Computational Mathematics (2018).

- A. H. Bentbib, K. Jbilou and Y. Kaouane, An adaptive block tangential method for multi-input multi-output dynamical systems. Accepted in Journal of Computational and Applied Mathematics (CAM).

- A. H. Bentbib, K. Jbilou and Y. Kaouane, A tangential method for the Balanced Truncation in model reduction. Submitted in Linear and Multilinear Algebra.

List of talks :

- Y. Kaouane, An adaptive block tangential method for multi-input multi-output dynamical systems. 44^{me} Congrès d'Analyse Numérique CANUM 2018, 28 mai au 1er juin 2018, Cap d'Agde, France.

- A. H. Bentbib, K. Jbilou and Y. Kaouane, Adaptive block tangential Arnoldi method for modelreduction of large-scale MIMO dynamical systems. Journée commemorant le 60^{me} anniversaire de l'Université de Med 5 de rabat, 28 Decembre 2018, ENS, Rabat, Morocco.

- A. H. Bentbib, K. Jbilou and Y. Kaouane, An adaptive tangential Lanczos-type algorithm for model reduction in large-scale dynamical systems. Congès international Modélisation et Calcul Scientifique pour l'Ingénierie Mathématiques, 17–20 Avril 2017, Marrakech, Morocco.

- A. H. Bentbib, K. Jbilou and Y. Kaouane, An adaptive tangential interpolation rational Krylov subspaces for MIMO model reduction. Journée Doctoriale MOCASIM, 30 Novembre 2016, Marrakech, Morocco.

- A. H. Bentbib, K. Jbilou and Y. Kaouane, Model reduction methods based on rational interpolation : rational Krylov subspaces. Ecole de recherche CIMPA Workshop sous le thème : modelisation, analyse mathématique et numérique pour les problème aux dérivée partielles, 09–21 Mai 2016, Nador, Morocco.

- An adaptive block tangential method for multi-input multi-output dynamical systems, seminar in Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville (LMPA), 09 Fevrier 2018, Calais, France.

- Adaptive tangential Computational Krylov subspaces methods for model reduction in large-scale dynamical systems, seminar in Laboratoire de Mathématiques Pures et Appliquées Joseph Liouville (LMPA), 02 Juin 2017, Calais, France.

Table des matières

Méthodes tangentielles pour les réductions de modèles et applications

Résumé

Les simulations à grande dimension jouent un rôle crucial dans l'étude d'une grande variété de phénomènes physiques complexes, entraînant souvent des demandes écrasantes sur les ressources informatiques. La gestion de ces demandes constitue la principale motivation pour la réduction du modèle : produire des modèles de commande réduite plus simples, qui permettent une simulation plus rapide et moins coûteuse tout en se rapprochant avec précision du comportement du modèle d'origine. La présence des systèmes avec multiple entrées et multiple sorties (MIMO) rend le processus de réduction encore plus difficile. Dans cette thèse, nous nous intéressons aux méthodes de réduction de modèles à grande dimension en utilisant la projection sur des sous-espaces de Krylov tangentielles. Nous nous penchons sur le développement de techniques qui utilisent l'interpolation tangentielle. Celles-ci présentent une alternative efficace et intéressante à la troncature équilibrée qui est considérée comme référence dans le domaine et tout particulièrement la réduction pour les systèmes linéaire à temps invariants. Enfin, une attention particulière sera portée sur l'élaboration de nouveaux algorithmes efficaces et sur l'application à des problèmes pratiques.

Mots clés : réduction de modèle, interpolation, sous-espace de krylov

---

Tangential methods for model reductions and applications

Abstract

Large-scale simulations play a crucial role in the study of a great variety of complex physical phenomena, leading often to overwhelming demands on computational resources. Managing these demands constitutes the main motivation for model reduction: produce simpler reduced-order models, which allow for faster and cheaper simulation while accurately approximating the behaviour of the original model. The presence of multiple inputs and outputs (MIMO) systems, makes the reduction process even more challenging. In this thesis we are interested in methods of reducing large-scale models, using projection on tangential Krylov subspaces. We are looking at the development of techniques using tangential interpolation. These present an effective and interesting alternative to the balanced truncation which is considered as a reference in the field and especially for the reduction of linear time invariant systems. Finally, special attention will be focused on the development of new efficient algorithms and application to practical problems.

Keywords: model reduction, interpolation, krylov subspace

LAMAI & LMPA
Maison de la Recherche Blaise Pascal – 50, rue Ferdinand Buisson – CS 80699 – 62228 Calais Cedex – France