Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aeroelastic predictions

Article · January 2010

1 author:

David Amsallem
Stanford University
46 PUBLICATIONS **2,127** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   Berkeley MAPP and VAPP View project

Project   Model order reduction using L1-norm minimization View project

INTERPOLATION ON MANIFOLDS OF

CFD-BASED FLUID AND

FINITE ELEMENT-BASED STRUCTURAL

REDUCED-ORDER MODELS

FOR ON-LINE AEROELASTIC PREDICTIONS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND

ASTRONAUTICS

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

David Amsallem

June 2010

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

(Charbel Farhat)   Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

(George Papanicolaou)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

(Sanjay Lall)

Approved for the University Committee on Graduate Studies

_____

Abstract

The critical impact of aeroelastic phenomena on the design and performance of aircraft calls for their accurate numerical prediction. Until the advent of modern computational capabilities, methods based on the linear theory of aeroelasticity were used, leading to reasonable predictions, except in the transonic flight regime. This regime being critical for high-performance jets, expensive wind tunnel testing remains the only option, before flight testing, for flutter clearance of new aircraft. The simultaneous development of advance Computational Fluid Dynamics (CFD) methods and high performance numerical algorithms has then suggested that CFD-based methods could become an alternative tool. While CFD-based aeroelastic computations have accurately predicted the correct behavior of full aircraft in the subsonic, transonic and supersonic regimes, the associated high computational cost has prevented these methods to be integrated in routine analysis. This is even the case for flutter which can be treated as a linearized problem, and therefore is less extensive to solve than nonlinear problems.

Reduced-Order Models (ROMs) have then seen a growing interest in the aeroelastic community because their lower dimensionality implies reduced computational costs. Unfortunately, routine analysis and flutter clearance involve parameter variations and most if not all ROMs lack robustness with respect to parameter changes.

Therefore, performing computations with ROMs calls for reconstructing a new ROM every time a new configuration is considered. However, such a reconstruction can be a computationally intensive process since the high-fidelity model is involved.

Together, these two issues underline the need for a new strategy for adapting pre-computed ROMs to new sets of physical or modeling parameters. In this dissertation a database of reduced-order information associated to fast interpolation-based techniques are considered. ROMs and their corresponding reduced-order bases are quantities that typically belong to nonlinear, matrix manifolds. As such, classical interpolation methods fail, as they are not able to enforce the constraints characterizing those manifolds. The first part of this thesis consists in first identifying the quantities of interest to be interpolated as well as their associated constraints and then designing a suitable interpolation method enforcing the constraints. Applications to the fast aeroelastic prediction of the behavior of two full aircraft configurations (F-16 Block 40 and F-18/A) are then presented. The contributions of this thesis also include a procedure for adapting structural ROMs to shape parameter variations. An approach based on a database of reduced-order fluid bases and reduced-order structural models coupled with this method of interpolation on a manifold is then shown to greatly reduce the computational cost for aeroelastic predictions of a full F-16 Block 40 aircraft while retaining good accuracy. The proposed method enables test operation calls for new, "last minute" flight configurations and thus paves the way for on-line, routine usage of reduced-order modeling including during flight testing.

Acknowledgments

I am very grateful to my advisor, Prof. Charbel Farhat for providing me with a wonderful research topic, encouragement, support and valuable guidance when I needed it as well as the freedom to explore new research paths. I would like to thank him for providing me with unique opportunities in terms of interaction with industry and the academic world.

I have been lucky enough to have a diverse and multidisciplinary thesis commitee. I would like to thank Professors George Papanicolaou and Sanjay Lall for accepting of forming my thesis reading committee, as well as Professors Juan Alonso and James Lambers for forming my oral presentation committee.

I would like to thank Dr. Thuan Lieu for his doctoral work and early stage support that provided me with a high-performance aeroelastic code I could use as a solid basis for the research conducted in this thesis. I am also grateful to Professors Jean-Frédéric Gerbeau, Michel Lesoinne and Ulrich Hetmaniuk for their precious help during their various visits at Stanford. Also, I am very thankful to my colleagues at FRG I have had the chance to collaborate with on some aspects of this dissertation research. I especially would like to thank Julien Cortial and Kevin Carlberg, not

Contents

List of Tables

xvii

List of Figures

Chapter 1

Introduction

1.1 Motivation

Aeroelastic analysis has a critical impact on the design and performance of an aircraft. The coupling between the aerodynamic loading due to the fluid surrounding the aircraft and its structural properties can lead to instabilities that cause important damage or failure. As such, predicting such dynamic instabilities is of prime interest for aircraft designers and engineers. Until the advent of modern computational capabilities, computational procedures based on the linear theory of aeroelasticity were the sole numerical tools available. These procedures included the doublet lattice method for subsonic analysis and the piston theory for the supersonic regime [1]. Many aircraft such as military high-performance jets are however critical to aeroelastic phenomena such as flutter in the transonic regime and the linear flow theory is unable to make accurate predictions in that regime. The reason is the presence of highly nonlinear phenomena such as shocks occurring in the flow at those speeds. Hence, wind tunnel testing has been the only method available for flutter testing in

the transonic regime for decades. However, designing, constructing and testing air-craft wind tunnel models for flutter is time demanding, leading to testing periods of more than a years time.

As such, the simultaneous development of high-order Computational Fluid Dynamics (CFD) methods and increased high-performance computing resources availability has suggested that CFD-based methods could become a new tool for the flutter analysis of full aircraft configurations. While CFD and Computational Structural Dynamics (CSD) coupled algorithms have been shown to accurately predict the in-flight aeroelastic behavior of the F-16 [2], CFD-based aeroelastic analysis procedures are still not integrated in the design loop due to the their high computational cost impact. Even aeroelastic perturbation phenomena such as flutter which lead to linear problems are relatively expensive to solve when based on CFD.

In order to tackle this CPU cost issue, Reduced-Order Modeling (ROM) methods have seen a growing interest in the aeroelasticity community since the end of the 90s. The high CPU burden associated with CFD-based techniques is indeed incurred by the large number of degrees of freedom (DOFs) resulting from the spatial discretization needed to accurately solve the underlying Partial Differential Equations (PDEs). As smaller number of degrees of freedom are involved, ROMs are however capable of being operated in near real-time. For this reason, and because they can be sufficiently accurate, ROMs are often sought-after for many applications pertaining to design [3], design optimization [4], control [5] and dynamic data-driven systems [6, 7], among others. Many approaches for constructing aeroelastic ROMs have been developed and shown to produce numerical results that compare well with those generated by high-fidelity nonlinear counterparts [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. Among these approaches, the Proper Orthogonal Decomposition (POD) method

[22, 23] is perhaps the most popular. For example, the POD was successfully applied to the CFD-based aeroelastic analysis of a transport aircraft model [24] and two complete fighter jet configurations [25, 26, 27, 28, 29, 30, 31].

Unfortunately, design optimization, control, data-driven systems and many other applications typically involve parameter changes, and most if not all ROM technologies lack robustness with respect to parameter variations. For example, a POD-based fluid ROM is very sensitive to the free-stream Mach number [15, 31]. In particular, it does not approximate well the dynamics of the fluid flow when the free-stream Mach number is different from that used for constructing the underlying POD basis. This is obviously an important limitation on reduced-order modeling since, due to this non-robustness, a ROM can only be useful at the point it has been built at. Therefore, performing "dynamic" computations using ROMs calls for constructing a new ROM each time a sensitive physical or modeling parameter is varied. Reconstructing a ROM is however in many cases a computationally intensive proposition as it requires generating solutions of the high-fidelity model, solving matrix equations or computing eigenmodes of the high-fidelity operator. All these methods involve the large-dimensional model and, as such, lead to expensive computations.

1.2 Strategy and Objectives

The main idea of this dissertation is to use databases of reduced-order information (ROI) and interpolation-based techniques in order to by-pass the expensive price associated with the use of the high-fidelity model. This idea is motivated by design-based strategies popular in the industry: the cost associated with previously created ROMs for some values of the parameters should be amortized by reusing them in the

interpolation process.

The development of such interpolation techniques is, however, not a straight-forward task. The main reason is that interpolating a reduced-order model should produce a viable reduced-order model, that is a ROM which shares the same characteristic properties as directly computed ROMs. Because, quantities involved in the definition of a ROM do not belong to a vector space, classical interpolation methods cannot be applied as is. Determining the properties characterizing ROMs and enforcing them in the interpolation process is one of the main contributions of this thesis.

The proposed interpolation-based ROM construction procedure that is proposed in this work is very general and can, as such, be applied to a variety of engineering problems. For instance, this approach can be used as a fast aeroelastic prediction tool for assisting flight-test. Predicting accurately and in real-time the aeroelastic behavior of the aircraft being tested at unexpected, "last minute" flight conditions is indeed a challenging problem in the flight-test community today.

For such applications, databases of previous test results conducted at various operating points are available to the engineer. Hence, being able to optimally exploit such a database in order to make accurate predictions at new operating points is critical. Figure 1.1 illustrates this concept of database: each point in the Mach number M_∞/angle of attack α parameter space represents a flight condition for which an F-16 Block 40 model has been tested for flutter.

Therefore, the main question remains the determination of the nature of the quantities to be stored in the database. In this dissertation, it will appear that reduced-order linear operators and reduced-order bases (ROBs) are those quantities of interest. Because these quantities belong to nonlinear matrix manifolds, a general high-order
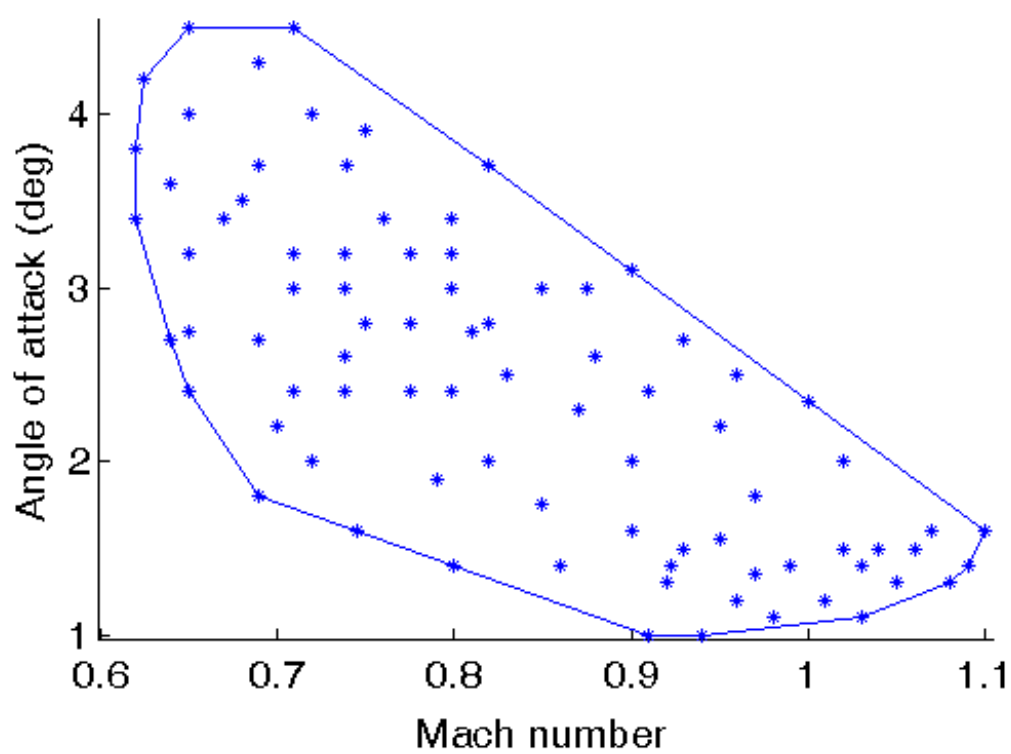
Figure 1.1: Example of database of reduced-order information for flutter testing of a full aircraft. Stars denote the points (M_∞, α) where ROI is stored.

multivariate interpolation method on such manifolds will be developed. This method is based on concepts from the field of differential geometry. The interpolation procedure will be subsequently coupled to a machine learning-based procedure, enabling real-time aeroelastic predictions using a database of ROMs and ROBs.

The ability of fast generation of ROMs for new values of parameters is also of great interest to the design and optimization communities, where many-queries-based methods are computationally demanding. As such, the scope of this thesis is not limited to routine analysis.

1.3   Thesis Accomplishments and Outline

The major contributions of this dissertation are as follows

- Construction and exploitation of a database of more than 80 reduced-order bases and models for the aeroelastic analysis of an F-16 Block 40 configuration over subsonic, transonic and supersonic flight regimes.

- Design of new, near-real time, interpolation algorithms on manifold for the model reduction of CFD-based fluid ROMs and Finite Element (FEM)-based structural ROMs.

- Assessment of the feasibility and practicality of the proposed methods for the aeroelastic analysis of two complete aircraft configurations.

- Comparison of meaningful CPU times demonstrating the feasibility and advantage of the proposed procedure over pre-existing methods.

This dissertation is organized as follows. Chapter 2 introduces a formal mathematical background on projection-based model reduction techniques for linear dynamical

systems. A review of the most common model reduction techniques is presented, with an emphasis on their qualitative properties. A rigorous analysis of the properties of projection-based model reduction then leads to the identification of matrix quantities of interest to be considered when interpolating linear ROMs. A new method for interpolating quantities belonging to matrix manifolds is then developed in chapter 3. It is based on notions and results from differential geometry that are reviewed as well. The general algorithm developed in this chapter is a contribution to applied mathematics and has potential applicability to fields such as imagery and data compression. The application of this general algorithm to matrix manifolds relevant to model reduction is then emphasized. Another contribution of this chapter is the rigorous sensitivity analysis of a model reduction technique, enabling the use of information on derivatives for compact interpolation of reduced-order bases. In chapter 4, model reduction of aeroelastic systems of two full aircraft configurations is presented, illustrating the potential of the new strategy by considering the interpolation of fluid reduced-order bases. Real-time interpolation technique for structural reduced systems is then developed in chapter 5 and shown to produce accurate structural ROMs when shape parameters are varied. Chapter 6 introduces techniques for the exploitation of a database of reduced-order bases and models for the F-16 Block 40, enabling real-time analyses. Interpolation of structural and fluid reduced-order operators is tackled in chapter 7. Finally, conclusions and indications for future research are provided in chapter 8.

Chapter 2

Projection-Based Model Reduction Techniques

2.1 Introduction

In this dissertation, high-fidelity nonlinear aeroelastic models are constituted of: (a) a CFD-based fluid model, (b) a FEM-based structural model. A discretization of the spatial domain has hence already been associated to each model. In each case, the governing equation of the subsystem can be written as an Ordinary Differential Equation (ODE) in time of dimension n_f and n_s respectively. As mentioned in the introduction to this thesis, flutter problems allow linearized formulations and the governing equations can be substituted by linear ODEs. Reduced-order modeling can then be performed onto the system by representing each subsystem by a much smaller number of degrees of freedom, $k_f \ll n_f$ for the fluid and $k_s \ll n_s$ for the structure.

This chapter introduces the class of model reduction techniques based on Petrov-Galerkin projection. These methods are characterized by the oblique projection of

high-dimensional systems of differential equations onto subspaces, thereby reducing both the number of equations and variables involved. Projection-based model reduction techniques are often categorized as modal based techniques [32]. These techniques rely on the determination of global, spatially distributed, modes describing the main behavior of the system. Alternatively, signal-based model reduction techniques such as Volterra series [19] rely on determining the relation between a small number of input and output quantities in order to approximate the transfer function of the full-order system. In this dissertation, only projection-based model reduction techniques are considered.

In section 2.2, general expressions for projection-based reduced linear dynamical systems are defined for both first- and second-order systems. In section 2.3 essential properties related to model reduction techniques are established. A careful analysis establishing classes of equivalence of ROMs is first conducted in section 2.3.3. This analysis will be essential to the determination of quantities to be interpolated when adapting ROMs.

Two analyses of basic properties of projection-based model reduction techniques are then conducted in sections 2.3.1 and 2.3.2. The error resulting from model reduction is first studied. The second analysis shows that the stability properties of a dynamical systems are not guaranteed to be preserved by model reduction. Both analyses will be of great importance to motivate the use of interpolation techniques for adapting ROMs in chapter 3. Numerical methods for solving the reduced systems are then reviewed in section 2.4, emphasizing the computational gains achieved by choosing a model reduction technique. In section 2.5, a review of common model reduction techniques is offered, establishing a framework for the model reduction of aeroelastic systems in chapter 4. Finally, the model reduction of a simple academic

dynamical system is presented as an illustrative example.

2.2 Petrov-Galerkin Projection-Based Model Reduction of a General Linear Dynamical System

2.2.1 Dynamical System of Interest

First-Order System

The following general linear dynamical system is considered

$$\begin{cases} \dfrac{d}{dt}\mathbf{x}(t;\boldsymbol{\mu}) &= \mathbf{A}(\boldsymbol{\mu})\mathbf{x}(t;\boldsymbol{\mu}) + \mathbf{B}(\boldsymbol{\mu})\mathbf{u}(t;\boldsymbol{\mu}) \\ \mathbf{y}(t;\boldsymbol{\mu}) &= \mathbf{C}(\boldsymbol{\mu})\mathbf{x}(t;\boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu})\mathbf{u}(t;\boldsymbol{\mu}). \end{cases} \tag{2.1}$$

All the variables are here assumed to be real-valued. $\boldsymbol{\mu} \in \mathbb{R}^{N_p}$ is a N_p-uplet containing the values of the parameters of the system. t denotes the time variable and typically belongs to a bounded interval $[t_0, T]$. $\mathbf{u} : [t_0, T] \times \mathbb{R}^{N_p} \to \mathbb{R}^{N_i}$ denotes the input variables and depend on the value of the parameter vector $\boldsymbol{\mu}$. $\mathbf{x} : [t_0, T] \times \mathbb{R}^{N_p} \to \mathbb{R}^{n}$ is the state variable vector of the system and $\mathbf{y} : [t_0, T] \times \mathbb{R}^{N_p} \to \mathbb{R}^{N_o}$ represents the output variables. These vectors depend on the parameter vector $\boldsymbol{\mu}$ as well. $\mathbf{A} : \mathbb{R}^{N_p} \to \mathbb{R}^{n \times n}$, $\mathbf{B} : \mathbb{R}^{N_p} \to \mathbb{R}^{n \times N_i}$, $\mathbf{C} : \mathbb{R}^{N_p} \to \mathbb{R}^{N_o \times n}$, $\mathbf{D} : \mathbb{R}^{N_p} \to \mathbb{R}^{N_o \times N_i}$ are assumed to be four continuous matrix-valued functions of $\boldsymbol{\mu}$. Initial conditions $\mathbf{x}(0;\boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu})$ are also associated to the first equation of the system.

A transfer function between inputs and outputs of the system can subsequently be written in the frequency domain by eliminating the state variable $\mathbf{x}$ as

$$\mathbf{G}(s;\boldsymbol{\mu}) = \mathbf{D}(\boldsymbol{\mu}) + \mathbf{C}(\boldsymbol{\mu})(s\mathbf{I}_n - \mathbf{A}(\boldsymbol{\mu}))^{-1}\mathbf{B}(\boldsymbol{\mu}), \quad s \in \mathbb{C}. \tag{2.2}$$

Second-Order System

Linear second-order dynamical systems can be written under a first-order form as well. The general dynamical system

$$\mathbf{A}_2(\boldsymbol{\mu})\frac{d^2}{dt^2}\mathbf{x}(t;\boldsymbol{\mu}) + \mathbf{A}_1(\boldsymbol{\mu})\frac{d}{dt}\mathbf{x}(t;\boldsymbol{\mu}) + \mathbf{A}_0(\boldsymbol{\mu})\mathbf{x}(t;\boldsymbol{\mu}) = \mathbf{B}(\boldsymbol{\mu})\mathbf{u}(t;\boldsymbol{\mu}) \tag{2.3}$$

can be written as

$$\frac{d}{dt}\hat{\mathbf{x}}(t;\boldsymbol{\mu}) = \hat{\mathbf{A}}(\boldsymbol{\mu})\hat{\mathbf{x}}(t;\boldsymbol{\mu}) + \hat{\mathbf{B}}(\boldsymbol{\mu})\mathbf{u}(t;\boldsymbol{\mu}), \tag{2.4}$$

assuming $\mathbf{A}_2(\boldsymbol{\mu})$ invertible, letting

$$\hat{\mathbf{x}}(t;\boldsymbol{\mu}) = \begin{bmatrix} \dfrac{d}{dt}\mathbf{x}(t;\boldsymbol{\mu}) \\ \mathbf{x}(t;\boldsymbol{\mu}) \end{bmatrix} \tag{2.5}$$

and

$$\hat{\mathbf{A}}(\boldsymbol{\mu}) = \begin{bmatrix} -\mathbf{A}_2(\boldsymbol{\mu})^{-1}\mathbf{A}_1(\boldsymbol{\mu}) & -\mathbf{A}_2(\boldsymbol{\mu})^{-1}\mathbf{A}_0(\boldsymbol{\mu}) \\ \mathbf{I}_n & \mathbf{0}_{n,n} \end{bmatrix}, \quad \hat{\mathbf{B}}(\boldsymbol{\mu}) = \begin{bmatrix} \mathbf{A}_2(\boldsymbol{\mu})^{-1}\mathbf{B}(\boldsymbol{\mu}) \\ \mathbf{0}_{n,N_i} \end{bmatrix}. \tag{2.6}$$

2.2.2 Model Reduction

Petrov-Galerkin Projection

The goal of model order reduction is to generate a dynamical system of much lower order k than the dimension n of the full-order model defined by Eq. (2.1), while retaining its main dynamical properties. A way to achieve this is to project the full-order set of equations using a well-suited Petrov-Galerkin projector in two steps as follows:

1. Define a trial basis $\mathbf{V}(\boldsymbol{\mu}) \in \mathbb{R}^{n \times k}$ having full-column rank and describing a trial subspace $\mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu})^1$. Each state vector $\mathbf{x}(t; \boldsymbol{\mu})$ in $\mathbb{R}^n$ can be decomposed as a sum of two orthogonal components, one lying in $\mathcal{V}(\boldsymbol{\mu})$ and another one in $\mathcal{V}^{\perp}(\boldsymbol{\mu})$, respectively spanned by two bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{V}^{\perp}(\boldsymbol{\mu})$ as

$$\mathbf{x}(t; \boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{V}^{\perp}(\boldsymbol{\mu})\check{\mathbf{x}}(t; \boldsymbol{\mu}) \tag{2.7}$$

with $\mathbf{x}_r(t; \boldsymbol{\mu}) \in \mathbb{R}^k$ and $\check{\mathbf{x}}(t; \boldsymbol{\mu}) \in \mathbb{R}^{n-k}$. The reduced-model approximation amounts to neglecting the components of $\mathbf{x}(t; \boldsymbol{\mu})$ that lie in the subspace $\mathcal{V}^{\perp}(\boldsymbol{\mu})$

$$\mathbf{x}(t; \boldsymbol{\mu}) \approx \mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}). \tag{2.8}$$

The trajectories of the solution $\mathbf{x}(t; \boldsymbol{\mu})$ are therefore constrained to evolve in the subspace $\mathcal{V}(\boldsymbol{\mu})$. The reduced size vector $\mathbf{x}_r \in \mathbb{R}^k$ defines the components of $\mathbf{x}$ in the trial basis $\mathbf{V}(\boldsymbol{\mu})$.

The linear system (2.1) becomes

$$\begin{cases} \mathbf{V}(\boldsymbol{\mu})\dfrac{d}{dt}\mathbf{x}_r(t; \boldsymbol{\mu}) &= \mathbf{A}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{B}(\boldsymbol{\mu})\mathbf{u}(t; \boldsymbol{\mu}) \\ \mathbf{y}_r(t; \boldsymbol{\mu}) &= \mathbf{C}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu})\mathbf{u}(t; \boldsymbol{\mu}). \end{cases} \tag{2.9}$$

2. Define a test basis $\mathbf{W}(\boldsymbol{\mu}) \in \mathbb{R}^{n \times k}$ having full-column rank and describing a test subspace $\mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu})$. The dynamical equation in (2.9) is then left-multiplied by the

---

[1]This subspace belongs to the Grassmann manifold, subsequently defined in section 3.3.2.

test basis as

$$\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \frac{d\mathbf{x}_r}{dt}(t; \boldsymbol{\mu}) = \mathbf{W}(\boldsymbol{\mu})^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{V}(\boldsymbol{\mu}) \mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{W}(\boldsymbol{\mu})^T \mathbf{B}(\boldsymbol{\mu}) \mathbf{u}(t; \boldsymbol{\mu}).$$
(2.10)

Assuming that $\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu})$ is non-singular, the reduced dynamical system is subsequently obtained under the same form as Eq. (2.1)

$$\begin{cases} \dfrac{d}{dt}\mathbf{x}_r(t; \boldsymbol{\mu}) & = \quad \mathbf{A}_r(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{B}_r(\boldsymbol{\mu})\mathbf{u}(t; \boldsymbol{\mu}) \\ \mathbf{y}_r(t; \boldsymbol{\mu}) & = \quad \mathbf{C}_r(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{D}_r(\boldsymbol{\mu})\mathbf{u}(t; \boldsymbol{\mu}), \end{cases}$$
(2.11)

where

$$\mathbf{A}_r(\boldsymbol{\mu}) = \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{V}(\boldsymbol{\mu}) \in \mathbb{R}^{k \times k} \tag{2.12}$$

$$\mathbf{B}_r(\boldsymbol{\mu}) = \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{B}(\boldsymbol{\mu}) \in \mathbb{R}^{k \times N_i} \tag{2.13}$$

$$\mathbf{C}_r(\boldsymbol{\mu}) = \mathbf{C}(\boldsymbol{\mu}) \mathbf{V}(\boldsymbol{\mu}) \in \mathbb{R}^{N_o \times k} \tag{2.14}$$

$$\mathbf{D}_r(\boldsymbol{\mu}) = \mathbf{D}(\boldsymbol{\mu}) \in \mathbb{R}^{N_o \times N_i}. \tag{2.15}$$

An initial condition for the reduced system can then be defined by projecting the initial condition for the full-order equations as

$$\mathbf{x}_r(0) = \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{x}(0) = \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{x}_0 \in \mathbb{R}^k. \quad (2.16)$$

The reduced system (2.11) corresponds to an oblique projection $\boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})}$ of the full-order system (2.1) onto $\mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu})$ orthogonally to $\mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu})$

$$\boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})} = \mathbf{V}(\boldsymbol{\mu}) \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T. \tag{2.17}$$

A transfer function $\mathbf{H}_r(s; \boldsymbol{\mu})$ can be defined for the reduced dynamical system as

$$\mathbf{H}_r(s; \boldsymbol{\mu}) = \mathbf{C}_r(\boldsymbol{\mu})\Big(s\mathbf{I}_k - \mathbf{A}_r(\boldsymbol{\mu})\Big)^{-1}\mathbf{B}_r(\boldsymbol{\mu}) + \mathbf{D}_r(\boldsymbol{\mu}), \quad s \in \mathbb{C}. \tag{2.18}$$

Second-Order Dynamical Systems

There are two procedures that can be followed when reducing a second-order linear dynamical systems.

- P1: First reduce the full-order system and then transform the second-order reduced system that is obtained into a first-order system. The reduced bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$ have as many rows as the state vector $\mathbf{x}(\boldsymbol{\mu})$. The second-order linear reduced dynamical systems takes the form

$$\begin{cases} \dfrac{d}{dt}\hat{\mathbf{x}}_r(t; \boldsymbol{\mu}) &= \hat{\mathbf{A}}_r(\boldsymbol{\mu})\hat{\mathbf{x}}_r(t; \boldsymbol{\mu}) + \hat{\mathbf{B}}_r(\boldsymbol{\mu})\mathbf{u}(t; \boldsymbol{\mu}) \\ \mathbf{y}_r(t; \boldsymbol{\mu}) &= \mathbf{C}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu})\mathbf{u}(t; \boldsymbol{\mu}), \end{cases} \tag{2.19}$$

letting

$$\hat{\mathbf{x}}_r(t; \boldsymbol{\mu}) = \begin{bmatrix} \dfrac{d}{dt}\mathbf{x}_r(t; \boldsymbol{\mu}) \\ \mathbf{x}_r(t; \boldsymbol{\mu}) \end{bmatrix}, \tag{2.20}$$

$$\hat{\mathbf{A}}_r(\boldsymbol{\mu}) = \begin{bmatrix} \hat{\mathbf{A}}_{r1}(\boldsymbol{\mu}) & \hat{\mathbf{A}}_{r0}(\boldsymbol{\mu}) \\ \mathbf{I}_k & \mathbf{0}_{k,k} \end{bmatrix} \tag{2.21}$$

with

$$\hat{\mathbf{A}}_{r1}(\boldsymbol{\mu}) = -\Big(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}_2(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\Big)^{-1}\mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}_1(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu}) \tag{2.22}$$

$$\hat{\mathbf{A}}_{r0}(\boldsymbol{\mu}) = -\Big(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}_2(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\Big)^{-1}\mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}_0(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu}), \tag{2.23}$$

and

$$\hat{\mathbf{B}}_r(\boldsymbol{\mu}) = \begin{bmatrix} \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{A}_2(\boldsymbol{\mu}) \mathbf{V}(\boldsymbol{\mu})\right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{B}(\boldsymbol{\mu}) \\ \mathbf{0}_{k,N_i} \end{bmatrix}. \tag{2.24}$$

- P2: First transform the full-order system into a first-order system (2.6), and then reduce the system that is obtained. The reduced bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$ have as many rows as $\hat{\mathbf{x}}(\boldsymbol{\mu})$, that is twice as many rows as the state vector $\mathbf{x}(\boldsymbol{\mu})$. In this case, the reduced-model approximation is

$$\hat{\mathbf{x}}(t; \boldsymbol{\mu}) \approx \mathbf{V}(\boldsymbol{\mu}) \hat{\mathbf{x}}_r(t; \boldsymbol{\mu}), \tag{2.25}$$

and hence

$$\mathbf{x}(t; \boldsymbol{\mu}) \approx \begin{bmatrix} \mathbf{0}_{n,n} & \mathbf{I}_n \end{bmatrix} \mathbf{V}(\boldsymbol{\mu}) \hat{\mathbf{x}}_r(t; \boldsymbol{\mu}) = \hat{\mathbf{V}}(\boldsymbol{\mu}) \hat{\mathbf{x}}_r(t; \boldsymbol{\mu}). \tag{2.26}$$

The second-order linear reduced dynamical system takes the form

$$\begin{cases} \dfrac{d}{dt} \hat{\mathbf{x}}_r(t; \boldsymbol{\mu}) &= \hat{\mathbf{A}}_r(\boldsymbol{\mu}) \hat{\mathbf{x}}_r(t; \boldsymbol{\mu}) + \hat{\mathbf{B}}_r(\boldsymbol{\mu}) \mathbf{u}(t; \boldsymbol{\mu}) \\ \mathbf{y}_r(t; \boldsymbol{\mu}) &= \mathbf{C}(\boldsymbol{\mu}) \hat{\mathbf{V}}(\boldsymbol{\mu}) \mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu}) \mathbf{u}(t; \boldsymbol{\mu}). \end{cases} \tag{2.27}$$

with

$$\hat{\mathbf{A}}_r(\boldsymbol{\mu}) = \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu})\right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \hat{\mathbf{A}}(\boldsymbol{\mu}) \mathbf{V}(\boldsymbol{\mu}) \tag{2.28}$$

and

$$\hat{\mathbf{B}}_r(\boldsymbol{\mu}) = \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu})\right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \hat{\mathbf{B}}(\boldsymbol{\mu}). \tag{2.29}$$

One should note that this method does not preserve the sparsity nature by blocks of $\hat{\mathbf{A}}(\boldsymbol{\mu})$ in Eq. (2.6).

2.3   Properties

2.3.1   Model Reduction Error

Galerkin Projection

Definition. A Galerkin projection is a Petrov-Galerkin projection where $\mathbf{W}(\boldsymbol{\mu})$ is chosen to be equal to $\mathbf{V}(\boldsymbol{\mu})$ and its columns are orthonormal as

$$\mathbf{V}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) = \mathbf{I}_k. \tag{2.30}$$

The corresponding projector is

$$\mathbf{\Pi}_{\mathbf{V}(\boldsymbol{\mu}),\mathbf{V}(\boldsymbol{\mu})} = \mathbf{V}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})^T. \tag{2.31}$$

The model reduction error $\mathcal{E}_{\text{ROM}}(t; \boldsymbol{\mu})$ is equal to

$$
\begin{aligned}
\mathcal{E}_{\text{ROM}}(t; \boldsymbol{\mu}) &= \mathbf{x}(t; \boldsymbol{\mu}) - \tilde{\mathbf{x}}(t; \boldsymbol{\mu}) \\
&= \mathbf{x}(t; \boldsymbol{\mu}) - \mathbf{\Pi}_{\mathbf{V}(\boldsymbol{\mu}),\mathbf{V}(\boldsymbol{\mu})}\mathbf{x}(t; \boldsymbol{\mu}) + \mathbf{\Pi}_{\mathbf{V}(\boldsymbol{\mu}),\mathbf{V}(\boldsymbol{\mu})}\mathbf{x}(t; \boldsymbol{\mu}) - \tilde{\mathbf{x}}(t; \boldsymbol{\mu}) \\
&= \left(\mathbf{I}_n - \mathbf{\Pi}_{\mathbf{V}(\boldsymbol{\mu}),\mathbf{V}(\boldsymbol{\mu})}\right)\mathbf{x}(t; \boldsymbol{\mu}) + \mathbf{V}(\boldsymbol{\mu})\left(\mathbf{V}(\boldsymbol{\mu})^T\mathbf{x}(t; \boldsymbol{\mu}) - \mathbf{x}_r(t; \boldsymbol{\mu})\right) \\
&= \mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})^\perp}(t; \boldsymbol{\mu}) + \mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})}(t; \boldsymbol{\mu}).
\end{aligned}
\tag{2.32}
$$

The first term $\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})^\perp}$ is the projection error that is a result of neglecting the components of $\mathbf{x}(t; \boldsymbol{\mu})$ that lie in the space orthogonal to $\mathbf{V}(\boldsymbol{\mu})$. The second term $\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})}(t; \boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})\left(\mathbf{V}(\boldsymbol{\mu})^T\mathbf{x}(t; \boldsymbol{\mu}) - \mathbf{x}_r(t; \boldsymbol{\mu})\right)$ is the reduced-order modeling error that results from solving a different dynamical system than the original one. This error vector lies in the subspace $\mathcal{V}(\boldsymbol{\mu})$ spanned by $\mathbf{V}(\boldsymbol{\mu})$. The two error vectors are orthogonal. This decomposition of the error between the Full-Order Model (FOM)

and ROM solutions is schematically represented in Fig. 2.1. Furthermore, the discrepancies in the trajectories of the original dynamical system, the projection of that trajectory onto the subspace $\mathcal{V}(\boldsymbol{\mu})$ and the trajectory of the reduced-system are schematically reported in Fig. 2.2. This figure refers directly to the schematic representation of the model reduction error that was first presented in [33].

Since the two components of the error are orthogonal, the following equality holds

$$\|\mathbf{x}(t;\boldsymbol{\mu}) - \tilde{\mathbf{x}}(t;\boldsymbol{\mu})\|_2^2 = \|\mathcal{E}_{\mathrm{ROM}}(t;\boldsymbol{\mu})\|_2^2 = \|\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})^\perp}(t;\boldsymbol{\mu})\|_2^2 + \|\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})}(t;\boldsymbol{\mu})\|_2^2. \qquad (2.33)$$

This is a very useful equality that can provide an a priori indication of the quality of a reduced basis $\mathbf{V}(\boldsymbol{\mu})$ before even solving the reduced-equations: $\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})^\perp}$ can indeed be computed knowing only the full-order solution $\mathbf{x}(t;\boldsymbol{\mu})$ and the reduced basis. A large error $\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})^\perp}$ would imply an even larger error $\mathcal{E}_{\mathrm{ROM}}$ and, as such, indicate that the basis $\mathbf{V}(\boldsymbol{\mu})$ is not suitable.

Petrov-Galerkin Projection

In the general case, the projector $\boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}),\mathbf{W}(\boldsymbol{\mu})}$ is oblique. The error vector $\mathcal{E}_{\mathrm{ROM}}(t;\boldsymbol{\mu})$ can still be decomposed in two components

$$\mathcal{E}_{\mathrm{ROM}}(t;\boldsymbol{\mu}) = \mathcal{E}_{\mathbf{W}(\boldsymbol{\mu})^\perp}(t;\boldsymbol{\mu}) + \mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})}(t;\boldsymbol{\mu}), \qquad (2.34)$$

with

$$\mathcal{E}_{\mathbf{W}(\boldsymbol{\mu})^\perp}(t;\boldsymbol{\mu}) = \left(\mathbf{I}_n - \boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}),\mathbf{W}(\boldsymbol{\mu})}\right)\mathbf{x}(t;\boldsymbol{\mu}) \in \mathcal{W}^\perp(\boldsymbol{\mu}) \qquad (2.35)$$

and

$$\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})}(t;\boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})\left(\left(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{V}(\boldsymbol{\mu})\right)^{-1}\mathbf{W}(\boldsymbol{\mu})^T\mathbf{x}(t;\boldsymbol{\mu}) - \mathbf{x}_r(t;\boldsymbol{\mu})\right) \in \mathcal{V}(\boldsymbol{\mu}). \qquad (2.36)$$
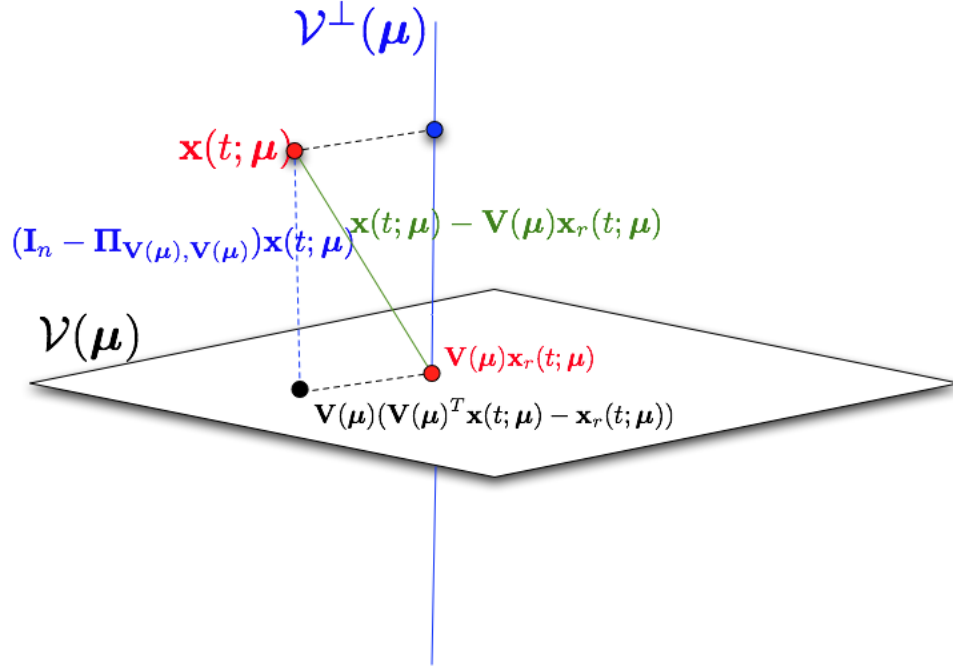
Figure 2.1: Orthogonal decomposition of the error between the full-order and reduced-order solutions after Galerkin projection
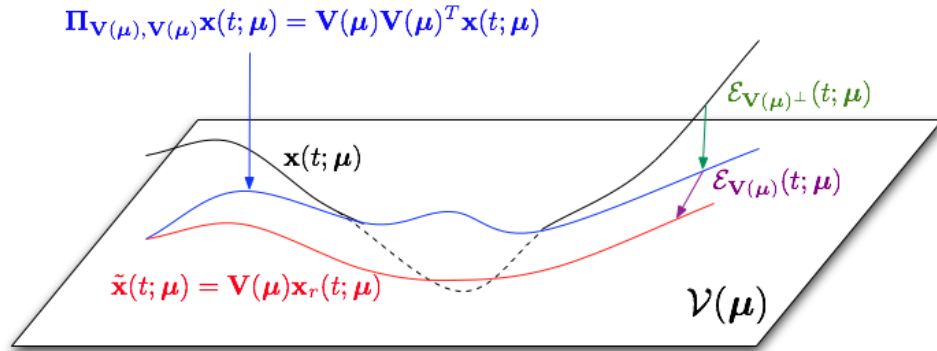
Figure 2.2: Discrepancy of the trajectories between the FOM and ROM solutions after Galerkin projection

The two components of the error are not orthogonal in this case. The triangular inequality can, however, provide a bound for $\mathcal{E}_{\mathrm{ROM}}(t; \boldsymbol{\mu})$ as

$$\|\mathbf{x}(t; \boldsymbol{\mu}) - \tilde{\mathbf{x}}(t; \boldsymbol{\mu})\|_2 = \|\mathcal{E}_{\mathrm{ROM}}(t; \boldsymbol{\mu})\|_2 \leq \|\mathcal{E}_{\mathbf{W}(\boldsymbol{\mu})^\perp}(t; \boldsymbol{\mu})\|_2 + \|\mathcal{E}_{\mathbf{V}(\boldsymbol{\mu})}(t; \boldsymbol{\mu})\|_2. \quad (2.37)$$

2.3.2 Stability

Preservation of stability is an essential property that characterizes a good reduced-order model. However, a general Petrov-Galerkin projection does not usually preserves stability. In order to emphasize this property, the following general full-order autonomous system is considered

$$\frac{d}{dt}\mathbf{x}(t; \boldsymbol{\mu}) = \mathbf{A}(\boldsymbol{\mu})\mathbf{x}(t; \boldsymbol{\mu}). \quad (2.38)$$

This system is assumed to be asymptotically stable, that is, all the eigenvalues of $\mathbf{A}$ have negative real part.

Its reduced-order counterpart is, after Petrov-Galerkin projection using the operator $\Pi_{\mathbf{V}(\boldsymbol{\mu}),\mathbf{W}(\boldsymbol{\mu})}$,

$$\frac{d}{dt}\mathbf{x}_r(t; \boldsymbol{\mu}) = \mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}), \quad (2.39)$$

where the two bases are chosen such that they are biorthogonal $(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{V}(\boldsymbol{\mu}) = \mathbf{I}_k)$.

Symmetric Case

Property. If the operator $\mathbf{A}(\boldsymbol{\mu})$ is symmetric and the full-order system asymptotically stable, then the reduced-order system is also asymptotically stable under Galerkin projection $(\mathbf{W}(\boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu}))$.

Proof. $\mathbf{A}_r(\boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})$ is also symmetric. Let λ_r denote an eigenvalue

of $\mathbf{A}_r(\boldsymbol{\mu})$. Then, since this is a symmetric matrix, there exists a vector $\boldsymbol{\phi} \neq \mathbf{0}$ with $\boldsymbol{\phi}^T \boldsymbol{\phi} = 1$ such that $\mathbf{A}_r(\boldsymbol{\mu})\boldsymbol{\phi} = \lambda_r \boldsymbol{\phi}$. Then,

$$\lambda_r = (\mathbf{V}(\boldsymbol{\mu})\boldsymbol{\phi})^T \mathbf{A}(\boldsymbol{\mu})(\mathbf{V}(\boldsymbol{\mu})\boldsymbol{\phi}) < 0 \tag{2.40}$$

since $\mathbf{V}(\boldsymbol{\mu})\boldsymbol{\phi} \neq \mathbf{0}$. All the eigenvalues of $\mathbf{A}_r(\boldsymbol{\mu})$ are negative. □

This result shows, in particular, that the stability of structural systems is always preserved under Galerkin projection.

Non-Symmetric Case

The property does not hold in the non-symmetric case as shown with the following counter-example. Considering an autonomous system with

$$\mathbf{A} = \begin{bmatrix} 1 & -3.5 \\ 0.6 & -2 \end{bmatrix}, \tag{2.41}$$

its spectrum is $\{-0.1127, -0.8873\}$. Let $\mathbf{V}$ be the reduced-order basis

$$\mathbf{V} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{2.42}$$

Then, $\mathbf{A}_r = \begin{bmatrix} 1 \end{bmatrix}$ and the reduced system after Galerkin projection is not asymptotically stable.

In particular, fluid operators which are non-symmetric and usually stable can become unstable under Galerkin projection, leading to unphysical systems. Hence, the choice of an appropriate reduced-order basis $\mathbf{V}(\boldsymbol{\mu})$ is crucial, not only in terms of

ROM accuracy but also in order to enforce basic stability properties of the underlying physical system.

Other definitions of stability exist for forced systems (systems with inputs and outputs) as well as non-linear systems (in the Lyapunov sense) [36]. The family of model reduction techniques based on the SVD – Balancing methods (section 2.5.1), Hankel-norm methods – preserves the stability of the full-order systems. However, the POD method (section 2.5.2) is not guaranteed to preserve the stability of the dynamical system.

2.3.3 Equivalence of Two Reduced-Order Models

Full-Order System Associated with a ROM

A full-order system can be associated with the reduced equations by reconstructing a full-order state vector $\tilde{\mathbf{x}}$ from the reduced coordinates. Doing so is meaningful in terms of analysis of the ROM since equivalence properties between different ROMs can be established. The present analysis goes beyond linear model reduction, and as such the following more generic notations are used, in order to emphasize the applicability of the present analysis to nonlinear ROMs as well.

$$\mathbf{f}(\mathbf{x}, \mathbf{u}; \boldsymbol{\mu}) = \mathbf{A}(\boldsymbol{\mu})\mathbf{x} + \mathbf{B}(\boldsymbol{\mu})\mathbf{u} \tag{2.43}$$

$$\mathbf{g}(\mathbf{x}, \mathbf{u}; \boldsymbol{\mu}) = \mathbf{C}(\boldsymbol{\mu})\mathbf{x} + \mathbf{D}(\boldsymbol{\mu})\mathbf{u}. \tag{2.44}$$

The reduced system (2.27) becomes

$$\begin{cases} \dfrac{d}{dt}\mathbf{x}_r(t; \boldsymbol{\mu}) = \left(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{V}(\boldsymbol{\mu})\right)^{-1}\mathbf{W}(\boldsymbol{\mu})^T\mathbf{f}\left(\mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}), \mathbf{u}(t; \boldsymbol{\mu}); \boldsymbol{\mu}\right) \\ \mathbf{y}_r(t; \boldsymbol{\mu}) = \mathbf{g}\left(\mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}), \mathbf{u}(t; \boldsymbol{\mu}); \boldsymbol{\mu}\right). \end{cases} \tag{2.45}$$

The initial condition for the reduced system $\mathbf{x}_r(0; \boldsymbol{\mu})$ is computed by projecting the initial condition for the full system $\mathbf{x}(0; \boldsymbol{\mu})$ using $\boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})}$ as

$$\tilde{\mathbf{x}}(0; \boldsymbol{\mu}) = \boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})} \mathbf{x}(0; \boldsymbol{\mu}) = \boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})} \mathbf{x}_0(\boldsymbol{\mu}), \tag{2.46}$$

that is

$$\mathbf{x}_r(0; \boldsymbol{\mu}) = \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{x}_0(\boldsymbol{\mu}). \tag{2.47}$$

$\tilde{\mathbf{x}}(t; \boldsymbol{\mu})$ denotes here the full state vector reconstructed from the reduced variable $\mathbf{x}_r(t; \boldsymbol{\mu})$ as

$$\tilde{\mathbf{x}}(t; \boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu}) \mathbf{x}_r(t; \boldsymbol{\mu}). \tag{2.48}$$

This vector is an approximation of the full state vector $\mathbf{x}(t; \boldsymbol{\mu})$ obtained by solving directly the full-order dynamical system. These two vectors are usually distinct, giving birth to an error due to the ROM approximation

$$\mathcal{E}_{\text{ROM}}(t; \boldsymbol{\mu}) = \mathbf{x}(t; \boldsymbol{\mu}) - \tilde{\mathbf{x}}(t; \boldsymbol{\mu}). \tag{2.49}$$

Furthermore, the vector $\tilde{\mathbf{x}}(t; \boldsymbol{\mu})$ satisfies a full-order dynamical system

$$\begin{cases} \dfrac{d}{dt} \tilde{\mathbf{x}}(t; \boldsymbol{\mu}) &= \mathbf{V}(\boldsymbol{\mu}) \left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}) \right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{f}\left(\tilde{\mathbf{x}}(t; \boldsymbol{\mu}), \mathbf{u}(t; \boldsymbol{\mu}); \boldsymbol{\mu} \right) \\ \mathbf{y}(t; \boldsymbol{\mu}) &= \mathbf{g}\left(\tilde{\mathbf{x}}(t; \boldsymbol{\mu}), \mathbf{u}(t; \boldsymbol{\mu}); \boldsymbol{\mu} \right), \end{cases} \tag{2.50}$$

which can be written in a more compact form as

$$\begin{cases} \dfrac{d}{dt} \tilde{\mathbf{x}}(t; \boldsymbol{\mu}) &= \boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})} \mathbf{f}\left(\tilde{\mathbf{x}}(t; \boldsymbol{\mu}), \mathbf{u}(t; \boldsymbol{\mu}); \boldsymbol{\mu} \right) \\ \mathbf{y}(t; \boldsymbol{\mu}) &= \mathbf{g}\left(\tilde{\mathbf{x}}(t; \boldsymbol{\mu}), \mathbf{u}(t; \boldsymbol{\mu}); \boldsymbol{\mu} \right), \end{cases} \tag{2.51}$$

with initial conditions $\tilde{\mathbf{x}}(0; \boldsymbol{\mu}) = \boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})} \mathbf{x}^0(\boldsymbol{\mu})$.

Reduced-Order Models Invariance

Equation (2.51) is a crucial identity that highlights the core of the reduced-order modeling approximation: solving the ROM dynamical systems amounts to solving a modified FOM dynamical system where the right-hand side of the dynamical equation as well as the initial conditions are respectively the original right-hand side and the original initial conditions projected using the Petrov-Garlerkin projector of interest. This remark also gives insight about the quantities of interest in model reduction.

Property. Choosing two different bases $\mathbf{V}'(\boldsymbol{\mu})$ and $\mathbf{W}'(\boldsymbol{\mu})$ that respectively span the same subspaces $\mathcal{V}(\boldsymbol{\mu})$ and $\mathcal{W}(\boldsymbol{\mu})$ results in the same reconstructed solution $\mathbf{x}(t; \boldsymbol{\mu})$.

Proof. Let $\mathbf{V}'(\boldsymbol{\mu})$ and $\mathbf{W}'(\boldsymbol{\mu})$ spanning the same respective subspaces as $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$. Then, there exists two invertible matrices $\mathbf{M}_{\mathbf{V}}$ and $\mathbf{M}_{\mathbf{W}}$ such that

$$\mathbf{V}'(\boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})\mathbf{M}_{\mathbf{V}}, \quad \mathbf{W}'(\boldsymbol{\mu}) = \mathbf{W}(\boldsymbol{\mu})\mathbf{M}_{\mathbf{W}}. \tag{2.52}$$

The corresponding Petrov-Galerkin projector is

$$
\begin{aligned}
\boldsymbol{\Pi}_{\mathbf{V}'(\boldsymbol{\mu}), \mathbf{W}'(\boldsymbol{\mu})} &= \mathbf{V}'(\boldsymbol{\mu})\left(\mathbf{W}'(\boldsymbol{\mu})^T \mathbf{V}'(\boldsymbol{\mu})\right)^{-1} \mathbf{W}'(\boldsymbol{\mu})^T & (2.53) \\
&= \mathbf{V}(\boldsymbol{\mu})\mathbf{M}_{\mathbf{V}}\left(\mathbf{M}_{\mathbf{W}}^T \mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu})\mathbf{M}_{\mathbf{V}}\right)^{-1} \mathbf{M}_{\mathbf{W}}^T \mathbf{W}(\boldsymbol{\mu})^T & (2.54) \\
&= \mathbf{V}(\boldsymbol{\mu})\left(\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu})\right)^{-1} \mathbf{W}(\boldsymbol{\mu})^T & (2.55) \\
&= \boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})}. & (2.56)
\end{aligned}
$$

This equality establishes that the reconstructed full-order state vector $\tilde{\mathbf{x}}'(t; \boldsymbol{\mu})$ corresponding to the projector $\boldsymbol{\Pi}_{\mathbf{V}'(\boldsymbol{\mu}), \mathbf{W}'(\boldsymbol{\mu})}$ will be identically equal to the state vector $\tilde{\mathbf{x}}'(t; \boldsymbol{\mu})$ corresponding to $\boldsymbol{\Pi}_{\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})}$, concluding the proof. $\square$

Application to the Interpolation of Reduced-Order Models

The above property is very important since it shows that the quantities of interest in model reduction are not the reduced bases but their corresponding projectors which are uniquely characterized by the subspaces the bases are spanning. Therefore, in this thesis, when interpolation of a reduced-order model is sought, interpolated quantities of interest will be subspaces and not reduced-order bases.

2.4 Numerical Resolution of the Reduced Dynamical Systems

In this section, the numerical resolution of the reduced linear dynamical systems is investigated. The ability to operate a linear ROM in real-time is shown.

The following linear dynamical system of equations is considered

$$\frac{d}{dt}\mathbf{x}_r(t; \boldsymbol{\mu}) = \mathbf{A}_r(\boldsymbol{\mu})\mathbf{x}_r(t; \boldsymbol{\mu}) + \mathbf{B}_r(\boldsymbol{\mu})\mathbf{u}(t; \boldsymbol{\mu}). \tag{2.57}$$

This set of equations can be subsequently discretized in time using the following m-step multistep method [34]:

$$\alpha_m \mathbf{x}_r^{n+m} + \cdots + \alpha_0 \mathbf{x}_r^n = \beta_m \Big(\mathbf{A}_r(\boldsymbol{\mu})\mathbf{x}_r^{n+m} + \mathbf{B}_r(\boldsymbol{\mu})\mathbf{u}^{n+m} \Big) + \cdots + \beta_0 \Big(\mathbf{A}_r(\boldsymbol{\mu})\mathbf{x}_r^n + \mathbf{B}_r(\boldsymbol{\mu})\mathbf{u}^n \Big). \tag{2.58}$$

$\mathbf{x}_r^n$ denotes here the numerical approximation of $\mathbf{x}_r(t_n; \boldsymbol{\mu})$ and $\mathbf{u}^n = \mathbf{u}(t_n; \boldsymbol{\mu})$. The unknown variable is $\mathbf{x}_r^{n+m}$ in the present case. Therefore, $\alpha_m \neq 0$. The scheme is said to be *implicit* if $\beta_m \neq 0$ and *explicit* otherwise.

Explicit Schemes

Explicit schemes can be simply solved for each iteration by direct substitutions of the variables $\{\mathbf{x}_r^{n+j}\}_{j=0}^{m-1}$ and $\{\mathbf{u}^{n+j}\}_{j=0}^{m}$ in the following equation

$$
\begin{aligned}
\mathbf{x}_r^{n+m} = \frac{1}{\alpha_m} & \left(\beta_m \Big(\mathbf{A}_r(\boldsymbol{\mu}) \mathbf{x}_r^{n+m} + \mathbf{B}_r(\boldsymbol{\mu}) \mathbf{u}^{n+m} \Big) + \cdots + \beta_0 \Big(\mathbf{A}_r(\boldsymbol{\mu}) \mathbf{x}_r^n + \mathbf{B}_r(\boldsymbol{\mu}) \mathbf{u}^n \Big) \right. \\
& \left. - \alpha_{m-1} \mathbf{x}_r^{n+m-1} - \cdots - \alpha_0 \mathbf{x}_r^n \right).
\end{aligned}
$$

$$(2.59)$$

The evaluations of the quantities in the right-hand side involve matrix-vector products of small-size matrices and vectors and, as such are inexpensive. Since the reduced operators are time-invariant they can be pre-computed once for all at the first step of the time marching process.

Implicit Schemes

The m-step scheme in the implicit case ($\beta_m \neq 0$) is

$$
\begin{aligned}
\Big(\alpha_m \mathbf{I}_k - \beta_m \mathbf{A}_r(\boldsymbol{\mu}) \Big) \mathbf{x}_r^{n+m} = & -\alpha_{m-1} \mathbf{x}_r^{n+m-1} - \cdots - \alpha_0 \mathbf{x}_r^n + \beta_m \mathbf{B}_r(\boldsymbol{\mu}) \mathbf{u}^{n+m} \\
& + \beta_{m-1} \Big(\mathbf{A}_r(\boldsymbol{\mu}) \mathbf{x}_r^{n+m-1} + \mathbf{B}_r(\boldsymbol{\mu}) \mathbf{u}^{n+m-1} \Big) + \cdots \quad (2.60) \\
& + \beta_0 \Big(\mathbf{A}_r(\boldsymbol{\mu}) \mathbf{x}_r^n + \mathbf{B}_r(\boldsymbol{\mu}) \mathbf{u}^n \Big).
\end{aligned}
$$

Computing $\mathbf{x}_r^{n+m}$ involves solving a small size k-by-k linear system and is inexpensive. Furthermore, if the time-steps are chosen to be uniform, the left-hand side matrix

is constant and, as such can be pre-factored using a LU decomposition [35]. The computation of $\mathbf{x}_r^{n+m}$ will require only forward and backward substitutions and will be quite inexpensive ($\mathcal{O}(k^2)$ operations). The dimension n of the full-order model is not part of the cost analysis, ensuring the real-time processing of the reduced-order model and giving the ability to obtain computational savings of several orders of magnitude both in terms of computational time and storage.

2.5 Review of Common Model Reduction Techniques

In this section, a linear system of the form of Eq. (2.1) is considered. This system is time-invariant. For simplification purposes, it is assumed that $\mathbf{D} = \mathbf{0}$.

2.5.1 Balanced Truncation

The *controllability* and *observability* Gramians are defined as the n-by-n symmetric positive semidefinite matrices [36]

$$\mathbf{W}_c = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^*t} dt \tag{2.61}$$

$$\mathbf{W}_o = \int_0^\infty e^{\mathbf{A}^*t}\mathbf{C}^*\mathbf{C} e^{\mathbf{A}t} dt. \tag{2.62}$$

Similarly, these two Gramians can be written in the frequency domain as [36, 37]

$$\mathbf{W}_c = \frac{1}{2\pi}\int_{-\infty}^\infty (j\omega\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^*(-j\omega\mathbf{I}_n - \mathbf{A}^*)^{-1} d\omega \tag{2.63}$$

$$\mathbf{W}_o = \frac{1}{2\pi}\int_{-\infty}^\infty (-j\omega\mathbf{I}_n - \mathbf{A}^*)^{-1}\mathbf{C}^*\mathbf{C}(j\omega\mathbf{I}_n - \mathbf{A})^{-1} d\omega. \tag{2.64}$$

$\mathbf{W}_c$ and $\mathbf{W}_o$ are respective bases for the controllable and observable subspaces. The controllable subspace characterizes to what degree a state $\mathbf{x}$ can be excited by an input without initial state. The measure of this degree is the inner product based on $\mathbf{W}_c$

$$\|\mathbf{x}\|_{\mathbf{W}_c}^2 = \mathbf{x}^*\mathbf{W}_c\mathbf{x}. \tag{2.65}$$

Similarly, the observable subspace indicates to what degree a given output can be excited by an initial state $\mathbf{x}_0$ with zero input. The measure of this degree is the inner product of that initial state based on $\mathbf{W}_o$

$$\|\mathbf{x}_0\|_{\mathbf{W}_o}^2 = \mathbf{x}_0^*\mathbf{W}_o\mathbf{x}_0. \tag{2.66}$$

$\| \ \|_{\mathbf{W}_c}$ and $\| \ \|_{\mathbf{W}_o}$ are semi-norms.

The two Gramians are solution of the Lyapunov equations

$$\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^* + \mathbf{B}\mathbf{B}^* = \mathbf{0}_n \tag{2.67}$$

$$\mathbf{A}^*\mathbf{W}_o + \mathbf{W}_o\mathbf{A} + \mathbf{C}^*\mathbf{C} = \mathbf{0}_n \tag{2.68}$$

Using one of this equation to compute a Gramian requires $\mathcal{O}(n^3)$ operations to compute the Gramian, and as such, is computationally intractable for large-scale systems.

An alternative to solving the Lyapunov equations consists of considering the state responses to unit impulses for the controllability Gramian, as the response to a unit input $\mathbf{u}(t) = \delta(t)\mathbf{e}_j$ is

$$\mathbf{x}_{j\delta}(t) = e^{\mathbf{A}t}\mathbf{B}\mathbf{e}_j = e^{\mathbf{A}t}\mathbf{b}_j, \quad j = 1, \cdots, N_i \tag{2.69}$$

where $\mathbf{b}_j$ is the j-th column vector of $\mathbf{B}$, and as such

$$
\int_0^\infty \sum_{j=1}^{N_i} \mathbf{x}_{j\delta}(t)\mathbf{x}_{j\delta}(t)^* dt = \int_0^\infty \sum_{j=1}^{N_i} e^{\mathbf{A}t}\mathbf{b}_j\mathbf{b}_j^* e^{\mathbf{A}^*t} dt = \int_0^\infty e^{\mathbf{A}t} \sum_{j=1}^{N_i} \mathbf{b}_j\mathbf{b}_j^* e^{\mathbf{A}^*t} dt
$$

$$
= \int_0^\infty e^{\mathbf{A}t} \sum_{j=1}^{N_i} \mathbf{B}\mathbf{e}_j\mathbf{e}_j^*\mathbf{B}^* e^{\mathbf{A}^*t} dt = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^*t} dt = \mathbf{W}_c.
$$

$$(2.70)$$

Computing $\mathbf{W}_c$ involves the simulation of N_i linear systems as well as the computation of an integral defined on $[0, \infty)$.

Similarly, the adjoint system can be considered

$$
\frac{d}{dt}\mathbf{z}(t; \boldsymbol{\mu}) = \mathbf{A}(\boldsymbol{\mu})^*\mathbf{z}(t; \boldsymbol{\mu}) + \mathbf{C}(\boldsymbol{\mu})^*\mathbf{v}(t; \boldsymbol{\mu}) \tag{2.71}
$$

and impulse responses computed as

$$
\mathbf{z}_j(t) = e^{\mathbf{A}^*t}\mathbf{c}_j, \quad j = 1, \cdots, N_o \tag{2.72}
$$

where $\mathbf{c}_j$ is the j-th column vector of $\mathbf{C}^*$, and then

$$
\int_0^\infty \sum_{j=1}^{N_o} \mathbf{z}_j(t)\mathbf{z}_j(t)^* dt = \int_0^\infty \sum_{j=1}^{N_o} e^{\mathbf{A}^*t}\mathbf{c}_j\mathbf{c}_j^* e^{\mathbf{A}t} dt = \int_0^\infty e^{\mathbf{A}^*t} \sum_{j=1}^{N_o} \mathbf{c}_j\mathbf{c}_j^* e^{\mathbf{A}t} dt = \mathbf{W}_o.
$$

$$(2.73)$$

Computing $\mathbf{W}_o$ involves the simulation of N_o linear systems, which can be expensive when the number of output is large, as in the case $\mathbf{C} = \mathbf{I}_n$ (entire state-vector).

Once the Gramians are known, two bases of $\mathbb{R}^n$, $\mathbf{T}$ and $\mathbf{S}$, can be computed when the system is both controllable and observable (that is $\mathbf{W}_c$ and $\mathbf{W}_o$ are positive

definite) as right-eigenvectors of the matrix product $\mathbf{W}_c\mathbf{W}_o$

$$\mathbf{W}_c\mathbf{W}_o\mathbf{T} = \mathbf{T}\mathbf{\Sigma}^2 \tag{2.74}$$

and left-eigenvectors

$$\mathbf{S} = \mathbf{T}^{-1}. \tag{2.75}$$

An overview of the various procedures and algorithms commonly used to compute the bases $\mathbf{T}$ and $\mathbf{S}$ is presented in [36].

By changing coordinates as $\mathbf{x} = \mathbf{T}\mathbf{z}$, it can be shown that the two Gramians are then transformed into equal and diagonal matrices

$$\mathbf{S}\mathbf{W}_c\mathbf{S}^* = \mathbf{T}^*\mathbf{W}_o\mathbf{T} = \mathbf{\Sigma}. \tag{2.76}$$

The coordinates are then said to be *balanced* [38]: a state that is less influenced by the input has also less influence on the output in those coordinates. Hence, this suggests truncating the least controllable and thus least observable states. This is achieved by a Petrov-Galerkin projection of the dynamical system of interest with a projector $\mathbf{\Pi}_{\mathbf{V},\mathbf{W}}$ where $\mathbf{V}$ and $\mathbf{W}$ denote respectively the k first columns of the matrices $\mathbf{T}$ and $\mathbf{S}$.

Denoting $\mathbf{\Sigma} = \mathbf{diag}(\sigma_1, \cdots, \sigma_n)$, σ_i, $i = 1, \cdots, n$ is the i-th Hankel singular value of the system. An error bound for the reduced-order model can be derived [39].

Property. The Balanced Truncation procedure yields the following error bound for the output of interest

$$\|\mathbf{y}(t) - \mathbf{y}_r(t)\|_2 \leq 2 \sum_{i=k+1}^{n} \sigma_i \|\mathbf{u}(t)\|_2. \tag{2.77}$$

Equivalently, the $\mathcal{H}_\infty$-norm of the error system is bounded by twice the sum of the neglected Hankel singular values:

$$\|\mathbf{G}(s) - \mathbf{G}_r(s)\|_{\mathcal{H}_\infty} \leq 2 \sum_{i=k+1}^{n} \sigma_i. \tag{2.78}$$

Equality holds when $\sigma_i = \sigma_{k+1}, \ \forall i = k+1, \cdots, n$.

Proof. A proof can be found in [39]. $\qquad\square$

The decay of the Hankel singular values of a given system can hence be used as a criteria for choosing the size of a ROM built by the Balanced Truncation procedure.

2.5.2 Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD) technique constructs an orthogonal basis of a fixed dimension k that represents the data in a certain least-squares optimal sense. The underlying data can be infinite dimensional (as for the properties of a fluid in a continuous domain) or finite dimensional. For simplicity, it is assumed here that the equations of interest have already been discretized, and, as such, the data has a finite dimension n. Reference [40] presents a derivation of the POD method in the infinite dimensional case. The strength of the POD method is that the method does not assume anything from the origin of the data: the data can come from a steady or unsteady, linear or nonlinear system. By looking for a linear subspace that approximates best the data, the method is computationally tractable, as linear algebra numerical methods can be directly applied. It should be noticed that the nonlinear nature of a dynamical system is preserved after model reduction, as shown in Eq. (2.51).

POD in the Time Domain

It is assumed in this section that a ROM is built for each value of the parameters $\boldsymbol{\mu}$. Each element of the data set is here represented by a vector $\mathbf{x}(t; \boldsymbol{\mu}) \in \mathbb{R}^n$. Continuous sets of data $\mathcal{X} = \{\mathbf{x}(t; \boldsymbol{\mu})\}_{0 \leq t \leq T} = \{\mathbf{x}(t)\}_{0 \leq t \leq T}$ are then considered (the parameter $\boldsymbol{\mu}$ is here dropped to alleviate the notations)[2]. An inner product can be defined between two such sets of data. Different choices for the inner product are possible. The optimal choice is usually linked to the problem of interest. In this section, the following general L_2 inner product is considered

$$(\mathbf{x}, \mathbf{y}) = \int_0^T \mathbf{x}(t)^T \mathbf{y}(t) dt. \tag{2.79}$$

The POD method seeks an orthogonal projector $\boldsymbol{\Pi}_{\mathbf{V}, \mathbf{V}}$ of fixed rank k such that

$$\tilde{J}(\boldsymbol{\Pi}_{\mathbf{V}, \mathbf{V}}) = \int_0^T \|\mathbf{x}(t) - \boldsymbol{\Pi}_{\mathbf{V}, \mathbf{V}} \mathbf{x}(t)\|_2^2 dt. \tag{2.80}$$

is minimized. In other terms, the goal is to build an orthogonal basis $\mathbf{V}$ minimizing $J(\mathbf{V}) = \tilde{J}(\boldsymbol{\Pi}_{\mathbf{V}, \mathbf{V}})$, the projection of the data onto the orthogonal complement to $\mathcal{V} = \text{span}(\mathbf{V})$.

Theorem. Let $\mathbf{K}$ denote the n-by-n real symmetric matrix

$$\mathbf{K} = \int_0^T \mathbf{x}(t) \mathbf{x}(t)^T dt. \tag{2.81}$$

---

[2]For simplicity it is assumed in this section that the mean value of the data is the zero vector $\mathbf{0}_n$. In the contrary case, the mean can be subtracted from the data and the subsequent analysis followed.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ be the eigenvalues of $\mathbf{K}$ and $\boldsymbol{\phi}_i \in \mathbb{R}^n$, $i = 1, \cdots, n$ their associated eigenvectors

$$\mathbf{K}\boldsymbol{\phi}_i = \lambda_i\boldsymbol{\phi}_i, \; i = 1, \cdots, n. \tag{2.82}$$

The subspace $\mathcal{V} = \text{span}(\mathbf{V})$ minimizing $J(\cdot)$ is the invariant subspace of $\mathbf{K}$ associated with the eigenvalues $\lambda_1, \cdots, \lambda_k{}^3$.

Proof. First, one can note that since $\mathbf{K}$ is a real symmetric matrix, it is diagonalizable and has a set of orthogonal eigenvectors associated to real, non-negative eigenvalues.

Furthermore, minimizing J is equivalent to maximizing $G(\mathbf{V})$, the integral of the norm of the projection of $\mathbf{x}(t)$ onto $\text{span}(\mathbf{V})$

$$G(\mathbf{V}) = \int_0^T \|\mathbf{\Pi}_{\mathbf{V},\mathbf{V}}\mathbf{x}(t)\|_2^2 dt = \int_0^T \|\mathbf{V}^T\mathbf{x}(t)\|_2^2 dt. \tag{2.83}$$

Letting $\mathbf{x}_r(t) = \mathbf{V}^T\mathbf{x}(t)$,

$$\|\mathbf{x}_r(t)\|_2^2 = \sum_{i=1}^k \mathbf{x}_{ri}(t)^2 = \text{tr}(\mathbf{x}_r(t)\mathbf{x}_r(t)^T) = \text{tr}\left(\mathbf{V}^T\mathbf{x}(t)\mathbf{x}(t)^T\mathbf{V}\right) \tag{2.84}$$

and

$$G(\mathbf{V}) = \int_0^T \text{tr}\left(\mathbf{V}^T\mathbf{x}(t)\mathbf{x}(t)^T\mathbf{V}dt\right) = \text{tr}\left(\mathbf{V}^T\int_0^T \mathbf{x}(t)\mathbf{x}(t)^T dt\mathbf{V}\right) = \text{tr}\left(\mathbf{V}^T\mathbf{K}\mathbf{V}\right) \tag{2.85}$$

with the constraint $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$, that is

$$\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}, \; 1 \leq i \leq j \leq k. \tag{2.86}$$

---

[3]It is assumed here that $\lambda_k > \lambda_{k+1}$

This constrained optimization problem can be solved introducing the Lagrangian

$$\mathcal{L}(\mathbf{V}, \boldsymbol{\alpha}) \;=\; G(\mathbf{V}) - \sum_{1 \leq i \leq j \leq k} \alpha_{i,j}(\mathbf{v}_i^T \mathbf{v}_j - \delta_{ij}) \tag{2.87}$$

$$=\; \mathrm{tr}\left(\mathbf{V}^T \mathbf{K} \mathbf{V}\right) - \mathrm{tr}(\mathbf{A}^T \mathbf{V}^T \mathbf{V}) + \mathrm{tr}(\mathbf{A}), \tag{2.88}$$

where $\mathbf{A}$ is a symmetric k-by-k matrix satisfying

$$\mathbf{A}_{i,i} = \alpha_{i,i}, \; i = 1, \cdots, k, \quad \mathbf{A}_{i,j} = \frac{1}{2}\alpha_{i,j}, \; 1 \leq i \neq j \leq k. \tag{2.89}$$

The first order optimality condition leads to

$$\nabla_{\mathbf{V}}\mathcal{L}(\mathbf{V}, \alpha) = 2\mathbf{K}^T \mathbf{V} - \mathbf{V}\mathbf{A} - \mathbf{V}\mathbf{A}^T = 2\mathbf{K}\mathbf{V} - 2\mathbf{V}\mathbf{A} = \mathbf{0}_{n,k} \Rightarrow \mathbf{K}\mathbf{V} = \mathbf{V}\mathbf{A} \tag{2.90}$$

since $\mathbf{A}$ is symmetric and

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\mathbf{V}, \alpha) = \mathbf{0}_{k,k} \Rightarrow \mathbf{V}^T \mathbf{V} = \mathbf{I}_k. \tag{2.91}$$

Eq. (2.90) shows that span($\mathbf{V}$) is an invariant subspace associated with $\mathbf{K}$. As such, since span($\boldsymbol{\Phi}_k$) = span($\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_k$) is such a subspace,

$$\sum_{i=1}^{k} \lambda_i = G(\boldsymbol{\Phi}_k) \leq G(\mathbf{V}). \tag{2.92}$$

Furthermore, since $\mathbf{V}^T \mathbf{V} = \mathbf{I}_k$, $\mathbf{A} = \mathbf{V}^T \mathbf{K} \mathbf{V} = \mathbf{V}^T \boldsymbol{\Phi}^T \boldsymbol{\Lambda} \boldsymbol{\Phi} \mathbf{V} = \boldsymbol{\Psi}^T \boldsymbol{\Lambda} \boldsymbol{\Psi}$, with $\boldsymbol{\Psi}^T \boldsymbol{\Psi} = \mathbf{I}_k$. Then

$$G(\mathbf{V}) = \mathrm{tr}\left(\mathbf{V}^T \mathbf{K} \mathbf{V}\right) = \mathrm{tr}(\boldsymbol{\Psi}^T \boldsymbol{\Lambda} \boldsymbol{\Psi}) = \sum_{j=1}^{k} \sum_{i=1}^{n} \boldsymbol{\Psi}_{ij}^2 \lambda_i \leq \sum_{i=1}^{k} \lambda_i. \tag{2.93}$$

Hence $G(\mathbf{V}) = \sum_{i=1}^{k} \lambda_i = G(\mathbf{\Phi}_k)$ and $\mathcal{V} = \text{span}(\mathbf{\Phi}_k)$, since it is assumed that $\lambda_k > \lambda_{k+1}$. $\qquad\qquad\square$

One can also note that

$$\max_{\mathbf{V}} G(\mathbf{V}) = \sum_{i=1}^{k} \lambda_i, \qquad (2.94)$$

$$\min_{\mathbf{V}} J(\mathbf{V}) = \sum_{i=k+1}^{n} \lambda_i. \qquad (2.95)$$

Hence, the eigenvalue λ_i associated to each POD vector $\mathbf{v}_i$ provides a quantitative measure of the contribution of each vector in the POD basis. This observation provides also a truncation criteria for choosing k:

$$E(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i}. \qquad (2.96)$$

$E(k)$ is the ratio of the total energy contained in the reduced subspace to the total energy in the full state-space. Its denomination takes its origin in turbulence applications. k is usually chosen such that $E(k)$ is very close to one, that is

$$\frac{\min_{\mathbf{V}} J(\mathbf{V})}{\min_{\mathbf{V}} J(\mathbf{V}) + \max_{\mathbf{V}} G(\mathbf{V})} = 1 - E(k) \qquad (2.97)$$

is close to zero.

Solving the eigenvalue problem (2.82) is usually computationally intractable due to the large dimension n of the underlying matrix $\mathbf{K}$. However, often, the data that is available under the form of "snapshots" that are state vectors $\{\mathbf{x}(t_i)\}_{i=1}^{N_{snap}}$ evaluated at instantaneous times $\{t_i\}_{i=1}^{N_{snap}}$. The integral (2.81) can be in that case approximated

by

$$\mathbf{K} = \sum_{i=1}^{N_{snap}} \mathbf{x}(t_i)\mathbf{x}(t_i)^T \delta_i, \tag{2.98}$$

where δ_i, $i = 1, \cdots, N_{snap}$ are quadrature weights. An n-by-N_{snap} snapshot matrix $\mathbf{X}$ can be defined as

$$\mathbf{X} = \left[\mathbf{x}(t_1)\sqrt{\delta_1} \quad \ldots \quad \mathbf{x}(t_{N_{snap}})\sqrt{\delta_{N_{snap}}}\right], \tag{2.99}$$

and the eigenvalue problem can be written, since $\mathbf{K} = \mathbf{X}\mathbf{X}^T$, as

$$\mathbf{X}\mathbf{X}^T \boldsymbol{\phi}_i = \lambda_i \boldsymbol{\phi}_i, \quad i = 1, \cdots, n. \tag{2.100}$$

Oftentimes, the number of snapshots N_{snap} is much smaller than the full-order dimension n. As such, the n-by-n eigenvalue problem (2.82) can be transformed into a small N_{snap}-by-N_{snap} eigenvalue problem [41], by noticing that the non-zero eigenvalues of the matrix $\mathbf{K}$ are the same as the ones for $\mathbf{R} = \mathbf{X}^T\mathbf{X}$. Hence, one can solve instead the cheaper eigenvalue problem

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i, \quad i = 1, \cdots, N_{snap}. \tag{2.101}$$

The first N_{snap} POD modes $\boldsymbol{\phi}_i$ are given by

$$\boldsymbol{\phi}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{X}\boldsymbol{\psi}_i, \quad i = 1, \cdots, N_{snap}. \tag{2.102}$$

Denoting $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \ldots & \boldsymbol{\phi}_{N_{snap}} \end{bmatrix}$ and $\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}_1 & \ldots & \boldsymbol{\psi}_{N_{snap}} \end{bmatrix}$, this is

$$\boldsymbol{\Phi} = \mathbf{X}\boldsymbol{\Psi}\Lambda^{-\frac{1}{2}}. \tag{2.103}$$

The optimal orthogonal basis is

$$\mathbf{V} = \mathbf{\Phi} \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0}_{n-k,k} \end{bmatrix} \tag{2.104}$$

which is the truncation of the first k columns of $\mathbf{\Phi}$.

$\mathbf{V}$ can alternatively be computed by performing a singular value decomposition of the snapshot matrix $\mathbf{X}$ and retaining the k first left singular vectors of the decomposition.

POD in the Frequency Domain

The POD kernel $\mathbf{K}$ presented in Eq. (2.83) can also be expressed in the frequency domain using Parseval's theorem by noting that [9]

$$\lim_{T \to \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \|\mathbf{V}^T \mathbf{x}(t)\|_2^2 dt = \lim_{T,\Omega \to \infty} \frac{1}{2\pi T} \int_{-\Omega}^{\Omega} \|\mathcal{F}\left[\mathbf{V}^T \mathbf{x}(t)\right]\|_2^2 d\omega. \tag{2.105}$$

Denoting by $\mathcal{X}(\omega) = \mathcal{F}[\mathbf{x}(t)]$ the Fourier transform of $\mathbf{x}(t)$,

$$\left\|\mathcal{F}\left[\mathbf{V}^T \mathbf{x}(t)\right]\right\|_2^2 = \left\|\mathbf{V}^T \mathcal{X}(\omega)\right\|_2^2 = \left\|\mathcal{X}(\omega)^T \mathbf{V}\right\|_2^2 = \mathbf{V}^T \mathcal{X}(\omega) \mathcal{X}(\omega)^* \mathbf{V} \tag{2.106}$$

and

$$\mathbf{V}^T \lim_{T \to \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \mathbf{x}(t)\mathbf{x}(t)^T dt \mathbf{V} = \mathbf{V}^T \lim_{T,\Omega \to \infty} \frac{1}{2\pi T} \int_{-\Omega}^{\Omega} \mathcal{X}(\omega)\mathcal{X}(\omega)^* d\omega \mathbf{V}. \tag{2.107}$$

This property justifies constructing an analog to $\mathbf{K}$ in the frequency domain

$$\tilde{\mathbf{K}} = \int_{-\Omega}^{\Omega} \mathcal{X}(\omega)\mathcal{X}(\omega)^* d\omega \tag{2.108}$$

and approximating this integral when snapshots $\{\mathcal{X}(\omega_i)\}_{i=1}^{N_{snap}^{\mathbb{C}}}$ in the frequency domain are available as

$$\tilde{\mathbf{K}} = \sum_{i=-N_{snap}^{\mathbb{C}}}^{N_{snap}^{\mathbb{C}}} \mathcal{X}(\omega_i)\mathcal{X}(\omega_i)^*\delta_i, \tag{2.109}$$

where $\delta_i,\ i = -N_{snap}^{\mathbb{C}},\cdots,N_{snap}^{\mathbb{C}}$ are quadrature weights corresponding to $\omega_i,\ i = -N_{snap}^{\mathbb{C}},\cdots,N_{snap}^{\mathbb{C}}$ [37]. The frequency sampling is chosen so that

$$\omega_{-i} = -\omega_i,\ i = 0,\cdots,N_{snap}^{\mathbb{C}} \tag{2.110}$$

hence $\omega_0 = 0$. As such, since

$$\mathcal{X}(-\omega) = \overline{\mathcal{X}(\omega)}, \tag{2.111}$$

where $\overline{(\ )}$ denotes the complex-conjugate, $\mathcal{X}(0) \in \mathbb{R}$ and one can write $\tilde{\mathbf{K}}$ as

$$
\begin{aligned}
\tilde{\mathbf{K}} &= \mathcal{X}(0)\mathcal{X}(0)^T\delta_0 + \sum_{i=1}^{N_{snap}^{\mathbb{C}}} \mathcal{X}(\omega_i)\mathcal{X}(\omega_i)^*\delta_i + \mathcal{X}(-\omega_i)\mathcal{X}(-\omega_i)^*\delta_i \\
&= \mathcal{X}(0)\mathcal{X}(0)^T\delta_0 + \sum_{i=1}^{N_{snap}^{\mathbb{C}}} \mathcal{X}(\omega_i)\mathcal{X}(\omega_i)^*\delta_i + \overline{\mathcal{X}(\omega_i)\mathcal{X}(\omega_i)^*}\delta_i \\
&= \mathcal{X}(0)\mathcal{X}(0)^T\delta_0 + 2\sum_{i=1}^{N_{snap}^{\mathbb{C}}} \left(\mathrm{Re}\left(\mathcal{X}(\omega_i)\right)\mathrm{Re}\left(\mathcal{X}(\omega_i)\right)^T + \mathrm{Im}\left(\mathcal{X}(\omega_i)\right)\mathrm{Im}\left(\mathcal{X}(\omega_i)\right)^T\right)\delta_i.
\end{aligned}
\tag{2.112}
$$

The complex valued snapshots can therefore be decomposed into real-valued snapshots by taking their real and imaginary parts and the following snapshot matrix is constructed

$$
\begin{aligned}
\tilde{\mathbf{X}} = \Big[&\mathcal{X}(0)\sqrt{\delta_0}\ \ \mathrm{Re}\left(\mathcal{X}(\omega_1)\right)\sqrt{2\delta_1}\ \ \ldots\ \ \mathrm{Re}\left(\mathcal{X}(\omega_{N_{snap}^{\mathbb{C}}})\right)\sqrt{2\delta_{N_{snap}^{\mathbb{C}}}} \\
&\mathrm{Im}\left(\mathcal{X}(\omega_1)\right)\sqrt{2\delta_1}\ \ \ldots\ \ \mathrm{Im}\left(\mathcal{X}(\omega_{N_{snap}^{\mathbb{C}}})\right)\sqrt{2\delta_{N_{snap}^{\mathbb{C}}}} \Big].
\end{aligned}
\tag{2.113}
$$

The rest of the procedure is the same as in the case of building a POD basis from snapshots generated in the time domain.

Link with Controllability Gramian

As noted in [37], when snapshots are generated in the time domain from impulse excitations, the corresponding matrix $\mathbf{K}$ in Eq (2.98) is an approximation of the controllability Gramian

$$\mathbf{W}_c = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^*t}dt = \int_0^\infty \sum_{i=1}^{N_i} \mathbf{x}_{i\delta}(t)\mathbf{x}_{i\delta}(t)^* dt \qquad (2.114)$$

where $\mathbf{x}_{i\delta}(t) = e^{\mathbf{A}t}\mathbf{B}\mathbf{e}_i$ is the i-th input impulse response.

Similarly, snapshots can be generated in the frequency domain, as

$$\mathcal{X}(\omega) = (j\omega\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}\mathcal{U}(\omega), \qquad (2.115)$$

$\mathcal{U}(\omega)$ denoting the Fourier transform of $\mathbf{u}(t)$.

2.5.3   Balanced Proper Orthogonal Decomposition

The Balanced POD method is based on the definition of empirical Gramians approximating the exact Gramians

$$\mathbf{W}_c = \int_0^\infty \sum_{j=1}^{N_i} \mathbf{x}_j(t)\mathbf{x}_j(t)^* dt \qquad (2.116)$$

$$\mathbf{W}_o = \int_0^\infty \sum_{j=1}^{N_o} \mathbf{z}_j(t)\mathbf{z}_j(t)^* dt. \qquad (2.117)$$

These empirical Gramians can be based on solutions snapshots analog to the ones for the POD method: N_{snap} snapshots can be considered for every input variable and stored in an n-by-$N_{snap}N_i$ primal snapshot matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1(t_1)\sqrt{\delta_1} & \dots & \mathbf{x}_1(t_{N_{snap}})\sqrt{\delta_{N_{snap}}} & \dots & \mathbf{x}_{N_i}(t_1)\sqrt{\delta_1} & \dots & \mathbf{x}_{N_i}(t_{N_{snap}})\sqrt{\delta_{N_{snap}}} \end{bmatrix}.$$
(2.118)

The exact controllability Gramian is then approximated as

$$\mathbf{W}_c \approx \mathbf{X}\mathbf{X}^*.$$
(2.119)

Similarly, N_{snap} snapshots can be generated for each output variable, that is for each input variable of the adjoint system, and stored in an n-by-$N_{snap} \times N_o$ primal snapshot matrix

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1(t_1)\sqrt{\delta_1} & \dots & \mathbf{z}_1(t_{N_{snap}})\sqrt{\delta_{N_{snap}}} & \dots & \mathbf{z}_{N_o}(t_1)\sqrt{\delta_1} & \dots & \mathbf{z}_{N_o}(t_{N_{snap}})\sqrt{\delta_{N_{snap}}} \end{bmatrix}.$$
(2.120)

The exact observability Gramian is then approximated as

$$\mathbf{W}_o \approx \mathbf{Z}\mathbf{Z}^*.$$
(2.121)

In the Balanced POD method, the balancing bases are computed without forming the approximated Gramians, which consumes lots of memory (these are dense, large n-by-n matrices). Instead, the inner product between both snapshot matrices is considered, and its compact singular value decomposition is computed

$$\mathbf{Z}^*\mathbf{X} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T.$$
(2.122)

$\tilde{\mathbf{\Sigma}} \in \mathbb{R}^r$, where r is the rank of $\mathbf{Z}^*\mathbf{X}$. Choosing $k \leq r$, $\mathbf{V}$ and $\mathbf{W}$ are respectively the first k columns of the matrices $\mathbf{T}$ and $\mathbf{S}$ defined as

$$\mathbf{T} = \mathbf{X}\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-\frac{1}{2}} \tag{2.123}$$

$$\mathbf{S} = \mathbf{Y}\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}^{-\frac{1}{2}} \tag{2.124}$$

More details on the Balanced POD method can be found in Refs. [12, 37, 42]

Similarly to the POD method, primal and dual snapshots can be generated in the frequency domain to approximate the controllability and observability Gramians written in the frequency domain (2.63)–(2.64). More details can be found in [37].

An extension of balanced truncation based on empirical Gramians to nonlinear systems has also been developed in [5].

2.5.4 Moment Matching Techniques

The transfer function $\mathbf{G}(s)$ defined in section 2.2.1 has moments η_j, $j \geq 0$ at an arbitrary point $s_0 \in \mathbb{C}$ defined as

$$\eta_j(s_0) = (-1)^j \frac{d^j}{ds^j}\mathbf{G}(s)\bigg|_{s=s_0}. \tag{2.125}$$

The goal of moment matching is to find a linear system of reduced-order k such that some moments $\hat{\eta}_j(\cdot)$, $j = 0, \cdots, l$ of this reduced system match the ones of the full system:

$$\eta_j(s_0) = \hat{\eta}_j(s_0), \ j = 0, \cdots, l. \tag{2.126}$$

Such a procedure is named *partial realization* when $s_0 = \infty$, *Padé approximation* when $s_0 = 0$ and *rational interpolation* when s_0 is general.

Moment matching can be achieved without direct computation of the moments. Numerical methods based on Arnoldi and Lanczos iterative procedures have been developed to compute such a ROM [36].

2.5.5 Techniques Comparison

As summarized by Antoulas in [36], two main classes of model reduction methods can be distinguished:

- The methods based on Balanced Approximation and Optimal Hankel norm approximation techniques [36]. These methods are based on the Singular Value Decomposition and preserve the stability of the FOM, provide global error bounds but are expensive. As such, these methods can only be applicable to moderately large full-order systems.

- The methods based on either moment matching techniques via Krylov iteration or Padé reconstruction that are computationally relatively inexpensive, and can therefore be applied to very large systems, but that lack global error bounds and stability preservation.

One should note that the POD method could be added to the second category since it shares all of its properties with moment matching techniques.

Two methods have been proposed in the literature to remediate this opposition and take advantage of the best properties of each class of methods:

- The Balancing techniques based on empirical Gramians, such as the Balanced POD method which provide error bounds, but are also computationally less expensive, as they are not based on solving Lyapunov equations. Stability

is preserved when the inner product of reference is based on the observability Gramian. However such a Gramian is only available for moderate size problems.

- The class of SVD-Krylov methods [36] is also promising since these methods combine stability preservation and inexpensive computations.

In this dissertation, the very large dimensionality of the full-order systems prevents the practical use of Balanced Approximation-type techniques for model reduction. As such, only techniques from the second category can be chosen in practice. The Proper-Orthogonal Decomposition method is used here to model the fluid subsystem because its effectivity in reducing aerodynamic and aeroelastic systems has been demonstrated in the past on a variety of examples. However, a drawback is the lack of error bound of this model reduction technique. Hence the recently developed balanced POD seems to be a method of choice for future investigation and, as such, an extension of the model reduction of aeroelastic system with POD to balanced POD is also developed in this thesis.

2.6 Reduced-Order Model Quality Criteria

The criteria presented by Antoulas et al. in Ref. [43] to determine the closeness between a linear Reduced-Order Model (ROM) and its corresponding linear Full-Order Model (FOM) are summarized in this section. Namely, these criteria can be classified into the following subcategories:

- C1: Dynamical properties of the dynamical matrices:
  - Comparison of the eigenvalues of the full-order dynamical matrix $\mathbf{A}(\boldsymbol{\mu})$ and its reduced-order counterpart $\mathbf{A}_r(\boldsymbol{\mu})$.

- C2: Frequency response:

  - Comparison of the Bode diagrams of the reduced and full-order systems $\mathbf{G}_r(s)$ and $\mathbf{G}(s)$.

  - Bode diagram of the error between the full-order and reduced-order systems

  $$\mathbf{E}(s) = \mathbf{G}(s) - \mathbf{G}_r(s). \tag{2.127}$$

  - Comparison of the Nyquist diagrams of the two systems.

- C3: Evaluation of the error between the FOM and ROM:

  - Relative $\mathcal{H}_\infty$-norm of the error system $\mathbf{E}(s)$.

  - Relative $\mathcal{H}_2$-norm of the error system $\mathbf{E}(s)$.

For more details about these criteria, the reader is referred to Reference [43].

These criteria are used in section 3.7.6 to assess the effectivity of the proposed reduced-order information interpolation techniques. For subsequent application of reduced-order modeling to aeroelastic systems, C1 and output-based comparison criteria are only used. The reason for this choice is that such criteria are in practice used to determine the aeroelastic behavior of a given aircraft and are, as such, more relevant to the application.

2.7 Application to a Mass-Spring-Damper System

2.7.1 Example System

A mass-damper-spring system shown in Figure 2.3 (and previously studied in Ref. [9]) is considered. The choice of this example is motivated by the fact that model reduction
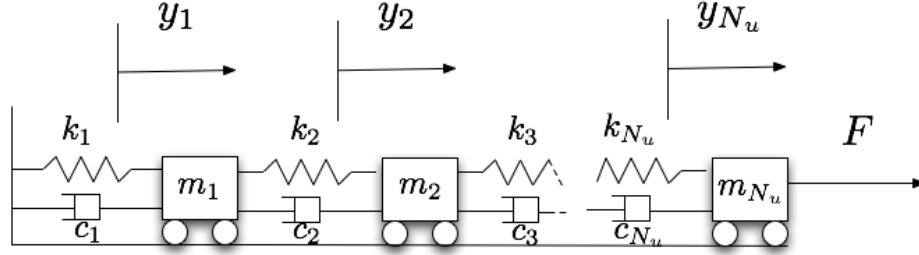
Figure 2.3: Mass-damper-spring system.

of structural subsystem is performed in chapters 4 and 5. Furthermore, studying such a system is computationally inexpensive and, as such, it can be considered as a good academic test example to assess the effectivity of new methods.

Each operating point of this mechanical system consists of $3N_u$ parameters corresponding to the $3N_u$ values of the masses $\{m_j\}_{j=1}^{N_u}$, dampers $\{c_j\}_{j=1}^{N_u}$ and springs $\{k_j\}_{j=1}^{N_u}$. For the sake of simplicity, it is assumed in this section that $m_j = m$, $c_j = c$, and $k_j = k$, $\forall j = 1, \cdots, N_u$, so that each operating point of the system is uniquely defined by the three parameters $\boldsymbol{\mu} = (m, c, k)$ only. Two systems are considered: System 1 is constituted of $N_u^{(1)} = 12$ mass-damper-spring units whereas the System 2 has $N_u^{(2)} = 48$ units.

The dependence of the dynamical properties of System 1 with respect to the operating point $\boldsymbol{\mu}$ is first illustrated. Figure 2.4 presents the Bode diagrams of the response of the $N_u^{(1)}$-th DOF of the system under an impulsive force applied to the corresponding mass. m and k are uniformly varied between 0.6 and 1.0 while c is kept constant at a value of 0.6. The various frequency response are quantitatively distinct.
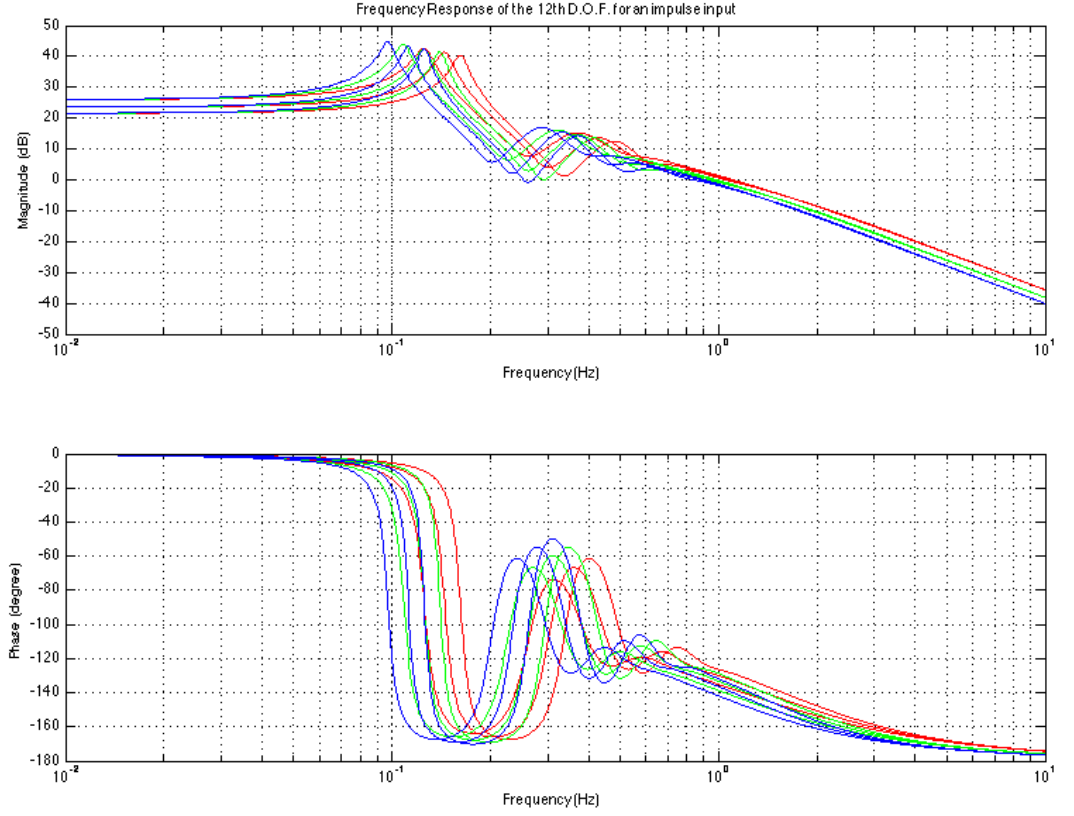
Figure 2.4: Frequency responses for an impulse input excitation. For varying values of k, the systems corresponding to $(m, c) = (0.6, 0.6)$ are reported in red, the ones for $(m, c) = (0.8, 0.6)$ in green and the ones for $(m, c) = (1.0, 0.6)$ in blue.

	$N_{snap}^{\mathbb{C}}$	ω_{min}	ω_{max}	N_{POD}
System 1	17	0	2.3	10
System 2	50	0	1.06	18

Table 2.1: Frequency sampling for snapshots generation.

2.7.2 Model Reduction

Procedure P2 (see section 2.2.2) is followed for reducing the system by a projection technique. As such, the second-order system is first transformed into a first-order system in state-space coordinates and then reduced by projection onto a subspace.

The POD method in the frequency domain is applied. Table 2.1 reports the frequency sampling used to generate the snapshots for both systems: $N_{snap}^{\mathbb{C}}$ complex-valued solutions are built by sampling uniformly the excitations frequencies between ω_{min} and ω_{max} rad/s. N_{POD} modes are then retained in the POD basis.

Criteria C2–C3 are then considered for assessing the quality of the ROM. The operating point $\boldsymbol{\mu} = (m, c, k) = (0.8, 0.6, 0.7)$ is chosen for both systems.

Figure 2.5 reports the Nyquist diagrams of the FOM and ROMs constructed for various values of N_{POD} for System 1. It can be observed that while the ROM constructed with 6 vectors in the basis is inaccurate, the one corresponding to 10 basis vectors is accurate enough. This ROM is then compared to the FOM by considering their two respective Bode diagrams in Fig. 2.6 as well as the Bode diagram of the error in Fig. 2.7. Very good agreements can be seen except for very high excitation frequencies.

Similarly, Fig. 2.8 reports the Nyquist diagrams of the FOM and ROMs constructed for various values of N_{POD} for the second system. The ROM constructed
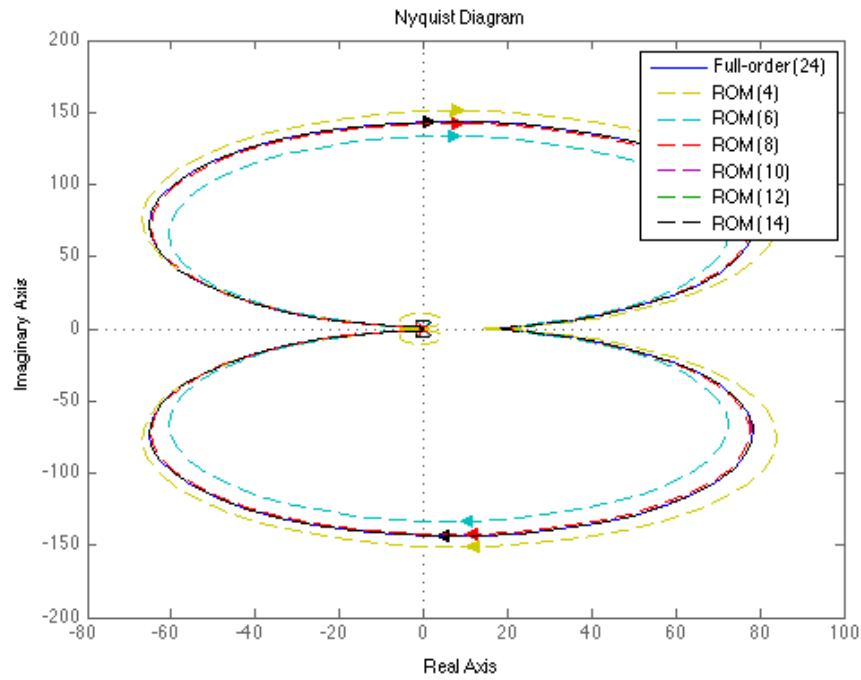
Figure 2.5: Comparison of the Nyquist diagrams of System 1 of the FOM and ROMs built with various numbers of POD vectors.
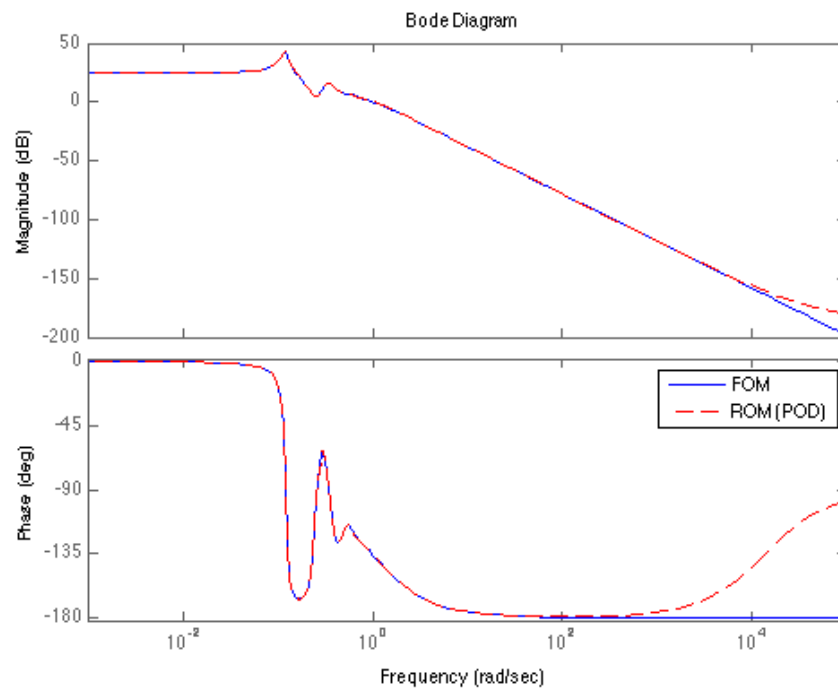
Figure 2.6: Comparison of the frequency responses of the FOM and the ROM of size 10.
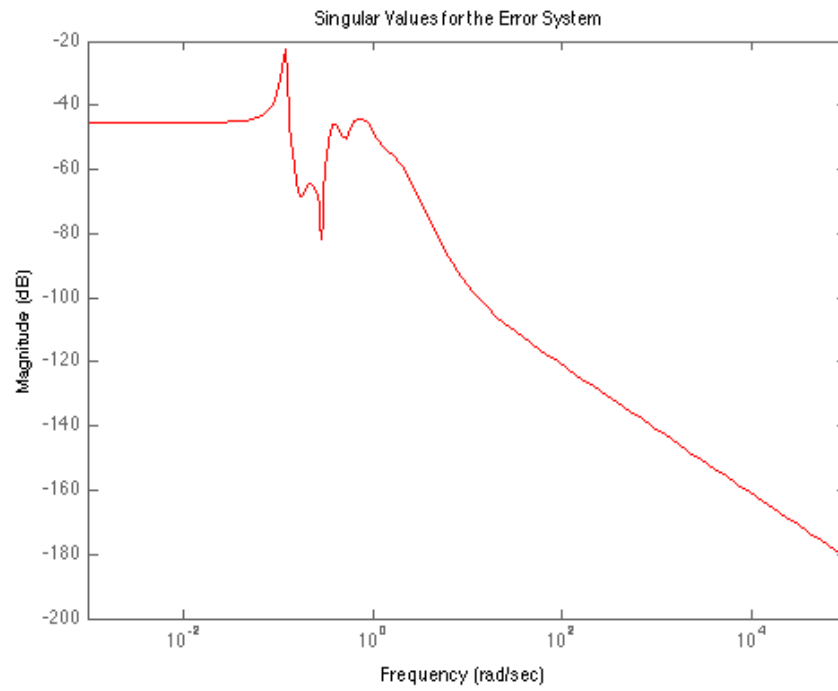
Figure 2.7: Frequency response of the error for System 1.
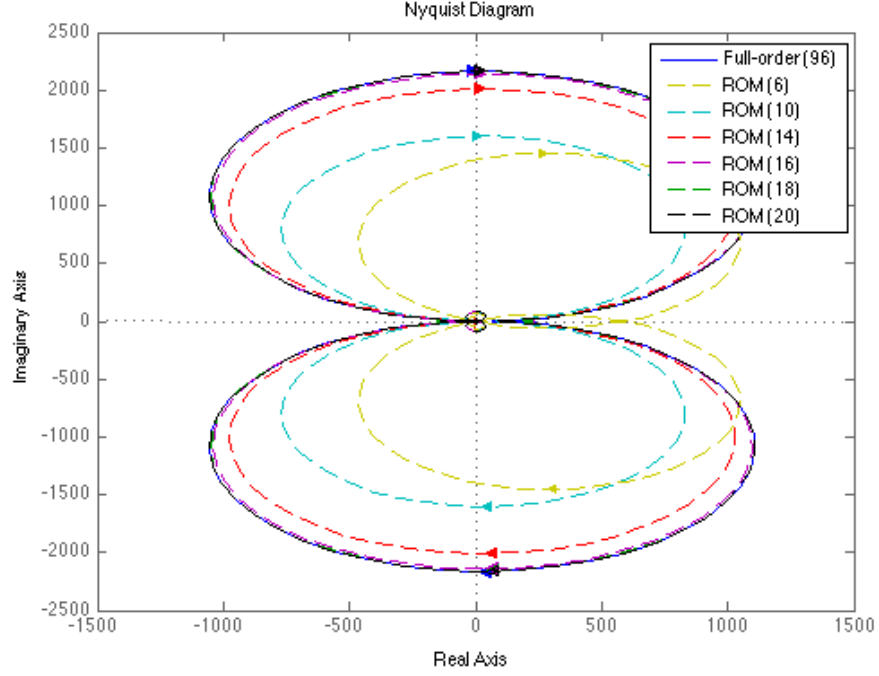
Figure 2.8: Comparison of the Nyquist diagrams of System 2 of the FOM and ROMs built with various numbers of POD vectors.

with 14 vectors in the basis is inaccurate but the one corresponding to 18 basis vectors is accurate enough. This ROM is then compared to the FOM by considering their respective Bode diagrams in Fig. 2.9 as well as the Bode diagram of the error in Fig. 2.10. As for System 1, very good agreements can be seen except for high excitation frequencies.

The relative errors for the $\mathcal{H}_\infty$- and $\mathcal{H}_2$-norms between the FOM and ROM of both systems are reported in Table 2.2, emphasizing the accuracy of the chosen ROMs.
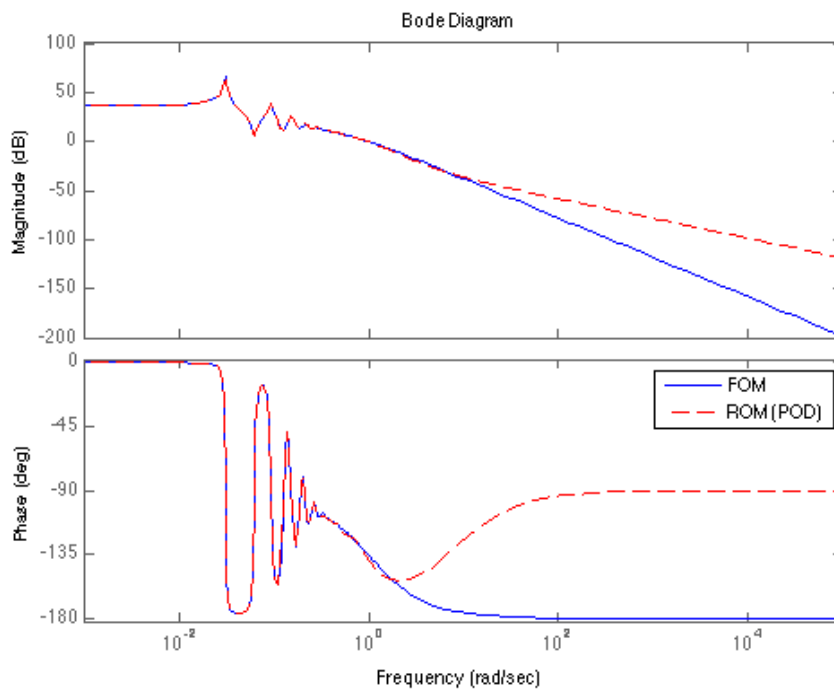
Figure 2.9: Comparison of the frequency responses of the FOM and the ROM of size 18.

	$\mathcal{H}_\infty$-norm	$\mathcal{H}_2$-norm
System 1	5.7×10^{-4}	4.9×10^{-4}
System 2	2.1×10^{-3}	1.7×10^{-3}

Table 2.2: Relative error norms for both systems.
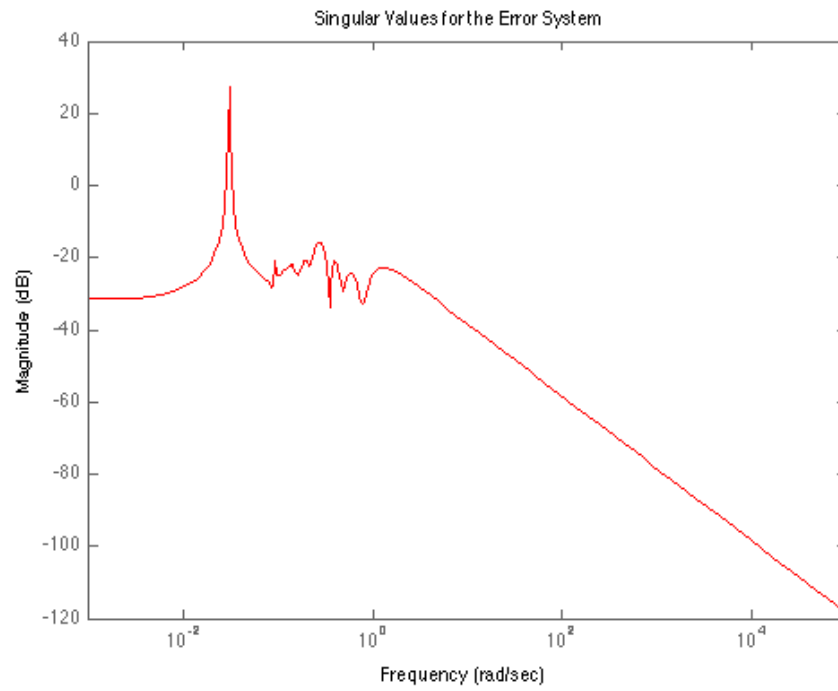
Figure 2.10: Frequency response of the error for System 2.

2.8 Parametric Reduced-Order Models

2.8.1 Sensitivity of the Reduced-Order Bases with Respect to the Parameters

In this thesis, full-order and reduced-order models are parameterized by a vector $\boldsymbol{\mu} \in \mathbb{R}^{N_p}$. This set of parameters is representative of physical variables characterizing an operating point. For instance, $\boldsymbol{\mu}$ can define far-field conditions of a fluid system, geometrical shape parameters or material properties of a structural system. Being able to compute sensitivities of the full and reduced-order quantities of interest is of great value to designers.

A derivation of the sensitivity of POD bases with respect to the operating point is presented. The resulting algorithm is subsequently used in the next chapter to develop sensitivity-based interpolation techniques. It will be shown that, for a same number of pre-computed information, such compact interpolation techniques can produce more accurate results than non-compact counterparts.

The algorithm for generating a POD basis from snapshots taken in the frequency domain (see section 2.5.2) can be differentiated step-by-step with respect to any parameter μ_l, $l = 1, \cdots, N_p$, as reported in Algorithm 1, and the sensitivities of the POD basis $\dfrac{\partial \mathbf{V}}{\partial \mu_l}$ obtained. The original linear system that is solved at $\boldsymbol{\mu} = \boldsymbol{\mu}^{(0)}$ when computing each snapshot is (for simplicity and without loss of generality, only a single output is considered)

$$\mathcal{X}_i = \left(j\omega_i \mathbf{I}_n - \mathbf{A}\left(\boldsymbol{\mu}^{(0)}\right)\right)^{-1} \mathbf{B}\left(\boldsymbol{\mu}^{(0)}\right) \mathcal{U}, \quad i = 0, \cdots, N_{snap}^{\mathbb{C}}. \qquad (2.128)$$

Note that the snapshot matrix $\tilde{\mathbf{X}}$ defined in Eq. (2.113) is assumed to have no repeated

singular values, so that the algorithm computing the sensitivity of the SVD with respect to the parameters presented in Appendix A can be used. This assumption is usually satisfied in practice due to numerical roundoff.

Algorithm 1 POD sensitivities computations at a configuration point $\boldsymbol{\mu}$

1: **for** $l = 1, \cdots, N_p$ **do**

2:    **for** $i = 0, \cdots, N_{snap}^{\mathbb{C}}$ **do**

3:

$$
\frac{\partial \mathcal{X}_i}{\partial \mu_l} = \left(j\omega_i \mathbf{I}_n - \mathbf{A}\left(\boldsymbol{\mu}^{(0)}\right)\right)^{-1} \frac{\partial \mathbf{A}}{\partial \mu_l}\left(\boldsymbol{\mu}^{(0)}\right)\left(j\omega_i \mathbf{I}_n - \mathbf{A}\left(\boldsymbol{\mu}^{(0)}\right)\right)^{-1} \mathbf{B}\left(\boldsymbol{\mu}^{(0)}\right)\mathcal{U}
$$
$$
+ \left(j\omega_i \mathbf{I}_n - \mathbf{A}\left(\boldsymbol{\mu}^{(0)}\right)\right)^{-1} \frac{\partial \mathbf{B}}{\partial \mu_l}\left(\boldsymbol{\mu}^{(0)}\right)\mathcal{U}
$$

$$(2.129)$$

4:    $\quad \dfrac{\partial \tilde{\mathbf{S}}}{\partial \mu_l} = \left[\sqrt{\delta_0}\dfrac{\partial \mathcal{X}_0}{\partial \mu_l} \quad \sqrt{2\delta_1}\dfrac{\partial \mathcal{X}_1}{\partial \mu_l} \quad \cdots \quad \sqrt{2\delta_{N_{snap}^{\mathbb{C}}}}\dfrac{\partial \mathcal{X}_{N_{snap}^{\mathbb{C}}}}{\partial \mu_l} \right]$

5:    $\quad \dfrac{\partial \tilde{\mathbf{X}}}{\partial \mu_l} = \left[\mathrm{Re}\left(\dfrac{\partial \tilde{\mathbf{X}}}{\partial \mu_l}\right) \quad \mathrm{Im}\left(\dfrac{\partial \tilde{\mathbf{X}}}{\partial \mu_l}\right) \right]$

6:    Compute the sensitivity of the singular values and vectors of $\tilde{\mathbf{X}}$ with respect to μ_l as shown in Appendix A

7:    Truncate the first columns of the sensitivity of the left singular vectors $\dfrac{\partial \mathbf{U}}{\partial \mu_l}$ to obtain the sensitivity of the POD basis $\dfrac{\partial \mathbf{V}}{\partial \mu_l}$ with respect to μ_l.

8:    **end for**

9: **end for**

This sensitivity analysis is local in the parameter space. When $\boldsymbol{\mu}$ is varied far enough from $\boldsymbol{\mu}^{(0)}$, large variations in the resulting Reduced-Order Bases may occur.

2.8.2 Non-Robustness of Parametric ROMs

ROMs are usually constructed as a low dimensional representation of the high-fidelity model for a given optimality criteria. For instance, the POD method based on snapshots constructs a ROM associated to a subspace so that the projection of the snapshots onto that subspace is maximized. This optimal low-order representation is therefore optimal for a given value $\boldsymbol{\mu}$ of the parameters and this optimality is lost when the parameters are varied.

This loss of optimality is illustrated in this thesis in section 4.5.2. A fluid reduced-order basis $\mathbf{V}^{(2)}$ is built for a flight condition parameterized as $\mu^{(2)}$. At that flight condition, the projection of the full-order model onto that basis produces an accurate ROM. However, when flight conditions $\mu^{(1)}$ and $\mu^{(3)}$ satisfying $\mu^{(1)} < \mu^{(2)} < \mu^{(3)}$ are considered, projecting the full-order models for these respective flight conditions onto $\mathbf{V}^{(2)}$ produces inaccurate ROMs, thereby illustrating the non-robustness of the reduced-order models. The same phenomena has been reported in the literature for a variety of systems [15, 44, 45, 46].

As mentioned in the introduction to this thesis, reconstructing a new reduced-order model, every time a new operating point $\boldsymbol{\mu}$ is considered, can be a computationally intensive task, especially for CFD-based high-fidelity models. As such, other approaches are sought to increase the range of application of already computed ROBs and ROMs. A first approach is based on the concept of a global basis which could be used for a whole range of parameters. The Global POD (GPOD) method presented in the next section is based on that idea. The second approach, based on interpolation of reduced-order data is the one developed in this dissertation.

2.8.3 Global Parametric Reduced-Order Bases and Models

In this section, several values $\boldsymbol{\mu}^{(1)}, \cdots, \boldsymbol{\mu}^{(N_R+1)}$ of the parameters are considered. Matrices of solution snapshots $\mathbf{X}\left(\boldsymbol{\mu}^{(i)}\right)$ for each system, defined as in Eq. (2.99), are first generated for each system defined by $\boldsymbol{\mu}^{(i)}, i = 1, \cdots, N_R$. A reduced-order basis is then sought for the system defined by

$$\boldsymbol{\mu}^{(N_R+1)} \neq \boldsymbol{\mu}^{(i)}, \ \forall i = 1, \cdots, N_R. \tag{2.130}$$

The basic idea behind the Global POD (GPOD) [47, 48, 49] approach – also called Lagrange subspace approach [50, 51] – is to enrich the snapshot matrix $\mathbf{X}$ with solutions corresponding to different values of the varied parameters, that is, with the present notations

$$\mathbf{X}\left(\boldsymbol{\mu}^{(N_R+1)}\right) = \left[\mathbf{X}\left(\boldsymbol{\mu}^{(1)}\right) \quad \cdots \quad \mathbf{X}\left(\boldsymbol{\mu}^{(N_R)}\right)\right]. \tag{2.131}$$

A global basis is then obtained by a singular value decomposition of that snapshot matrix. Alternatively, one may also add the sensitivities of the snapshots with respect to the parameters computed for each system $\boldsymbol{\mu}^{(i)}, i = 1, \cdots, N_R$ as additional snapshots. In that case,

$$\begin{aligned} \mathbf{X}\left(\boldsymbol{\mu}^{(N_R+1)}\right) = \Big[\mathbf{X}\left(\boldsymbol{\mu}^{(1)}\right) \ \ \frac{\partial \mathbf{X}}{\partial \mu_1}\left(\boldsymbol{\mu}^{(1)}\right) \ \ \cdots \ \ \frac{\partial \mathbf{X}}{\partial \mu_{N_p}}\left(\boldsymbol{\mu}^{(1)}\right) \ \ \cdots \\ \mathbf{X}\left(\boldsymbol{\mu}^{(N_R)}\right) \ \ \frac{\partial \mathbf{X}}{\partial \mu_1}\left(\boldsymbol{\mu}^{(N_R)}\right) \ \ \cdots \ \ \frac{\partial \mathbf{X}}{\partial \mu_{N_p}}\left(\boldsymbol{\mu}^{(N_R)}\right)\Big]. \end{aligned} \tag{2.132}$$

This method is sometimes referred to, in the literature, as the Hermite subspace [50, 51, 52, 53] or Compact POD method [54] when the snapshots are weighted.

The special case $N_R = 1$ has been treated heavily in the literature. This case

amounts to generate a reduced-order basis from a single pre-computed configuration. Oftentimes, higher-order sensitivities are also included as snapshots and the method is referred to as the Taylor subspace method [50, 55]. The POD basis at a single configuration is in that case extrapolated using the sensitivities information. Instead of adding the sensitivities of the snapshots, the sensitivities of the POD basis – as developed in section 2.8.1 – at one configuration are also sometimes added as additional snapshots [45, 46].

As noted by Degroote et al. [58], the GPOD approach is efficient computationally-wise when the matrices constituting the linear full-order model have an affine parameter dependence of the form

$$\mathbf{A}(\boldsymbol{\mu}) = \sum_{i=1}^{N_a} \mathbf{A}_i f_i(\boldsymbol{\mu}), \tag{2.133}$$

where the operators $\mathbf{A}_i$, $i = 1, \cdots, N_a$, are independent of the parameter vector $\boldsymbol{\mu}$ and the expressions for f_i, $i = 1, \cdots, N_a$, are easily available [59]. Such a form can also be obtained by a Taylor expansion of the operator $\mathbf{A}(\boldsymbol{\mu})$ in the vicinity of a point $\boldsymbol{\mu}^{(0)}$ [60]. Then, after Petrov-Galerkin projection of $\mathbf{A}(\boldsymbol{\mu})$ onto global left and right projection bases $(\mathbf{W}, \mathbf{V})$, a reduced matrix is obtained as

$$\mathbf{A}_r(\boldsymbol{\mu}) = \mathbf{W}^T \left(\sum_{i=1}^{N_a} \mathbf{A}_i f_i(\boldsymbol{\mu}) \right) \mathbf{V} = \sum_{i=1}^{N_a} \left(\mathbf{W}^T \mathbf{A}_i \mathbf{V} \right) f_i(\boldsymbol{\mu}). \tag{2.134}$$

As such, the quantities $\mathbf{W}^T \mathbf{A}_i \mathbf{V}$, $i = 1, \cdots, N_a$, can be pre-computed in an offline phase and the reduced-order model readily updated in an online phase at a new value $\boldsymbol{\mu}^\star$ of the parameter by a linear combination of the aforementioned quantities. This property is used in reduced basis frameworks [61, 62]. However, for many problems, such an affine dependency of the full-order model cannot be obtained. This is the

case in this dissertation for linearized fluid systems depending on flight parameters and structural systems depending on shape parameters.

For static systems, only one snapshot can be directly computed at a single configuration: the steady-state of the system. Hence, using a Global or Hermite subspace approach is effective for constructing a POD basis of moderate size at new configurations that can be used in optimization procedures [54] or to directly construct physics-based surrogate models of steady-state solutions of nonlinear systems [63]. However, using a Global POD approach for unsteady systems is usually ill-advised for a variety of reasons.

First, the method often leads to a large number of POD vectors in the basis: at each configuration, if N_{snap} snapshots are constructed, the method result in $N_{snap} \times (N_p+1) \times N_R$ snapshots, and in turn a large number of vectors in a given reduced-order basis, defeating the purpose of model reduction. Examples of non-linear systems for which a global basis approach leads to no reduction in the size of the full-order model have been provided [64].

Furthermore, it was demonstrated in the literature that using a global basis can alter the accuracy of unsteady ROMs. This is for instance the case when there is an optimal number of POD vectors that leads to maximal accuracy (see section 4.6.1). Hay et al. report in Ref. [45, 46] a numerical experiment of viscous flow around a cylinder in which adding sensitivities of the snapshots leads to an unphysical unsteady behavior of the ROM. The authors attribute this behavior by an incorrect energy transfer from the POD modes to the sensitivity components. Another drawback of global basis methods is the likelihood of correlations between low-energy modes across parameters, leading to the selection of these modes as high-energy modes in

the global basis and the truncation of other important modes that are, however, non-correlated [64]. These low-energy modes do not contain any physical information and often contain numerical noise, hence leading in the present case to an unphysical global basis.

To summarize, a global basis approach naturally loses the optimal approximation property of the POD method, and in many cases is not reliable for unsteady systems. For example, it also fails to properly adapt an aeroelastic ROM in the transonic regime to variations in the free-stream Mach number or angle of attack [27]. As such, methods based on interpolation of reduced-order information will appear to be the method of choice in this thesis, as these methods have the capability to: (a) generate new ROMs of the same small dimension as their pre-computed counterparts, (b) retain optimal approximation properties.

2.8.4 Interpolation of Parametric Reduced-Order Bases and Models

The main motivation behind such an approach is that, very often, reduced-order information varies continuously with respect to the operating parameters $\boldsymbol{\mu}$. As such, interpolation-based techniques can be applied to either (a) adapt the optimal projection subspaces to new values of the parameter and subsequently project full-order models constructed at these new operating points onto the interpolated subspaces, (b) interpolate the reduced-order models themselves, thereby by-passing the generation of a new full-order model altogether. Approaches for extrapolating ROBs in the vicinity of a pre-computed operating point $\boldsymbol{\mu}^{(0)}$ have been developed in References [45, 46, 53, 56, 57]. These methods require the computation of sensitivities of the POD bases as developed in section 2.8.1 and rely on a Taylor expansion of each

basis vector. As such, the method is limited to vicinity of pre-computed operating points. An interpolation method, on the contrary, can be effective in wider regions of the parameter space.

Optimal or sub-optimal approximations properties can be retained for the two proposed approaches – interpolation of pre-computed ROBs or ROMs – if the interpolated reduced-order quantities are sufficiently close to the target quantities. Furthermore, the dimension of the resulting ROM is the same as the dimension of the original precomputed ROMs, leading to an efficient model reduction at the new target operating point.

In both cases, reduced-order information is represented by matrix quantities that have intrinsic constraints such as full-column rank or non-singularity. These matrices belong to nonlinear manifolds. As such, classical interpolation techniques may fail to preserve these properties and new method for interpolating quantities belonging to matrix manifolds are sought. This is the focus of the next chapter of this thesis.

Chapter 3

Interpolation in the Tangent Space to a Matrix Manifold

3.1 Introduction

3.1.1 Motivation

The analysis conducted in the previous chapter has shown that the quantities characterizing linear reduced-order models of the form

$$
\begin{cases}
\dfrac{d}{dt}\mathbf{x}_r(t;\boldsymbol{\mu}) & = \ \Big(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\Big)\mathbf{x}_r(t;\boldsymbol{\mu}) + \Big(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{B}(\boldsymbol{\mu})\Big)\mathbf{u}(t;\boldsymbol{\mu}) \\[2mm]
\mathbf{y}(t;\boldsymbol{\mu}) & = \ \Big(\mathbf{C}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\Big)\mathbf{x}_r(t;\boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu})\mathbf{u}(t;\boldsymbol{\mu}) \\[2mm]
\mathbf{x}(t;\boldsymbol{\mu}) & = \ \mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t;\boldsymbol{\mu}),
\end{cases}
\tag{3.1}
$$

are the reduced-order linear operators $\mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})$, $\mathbf{W}(\boldsymbol{\mu})^T\mathbf{B}(\boldsymbol{\mu})$, $\mathbf{C}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})$, $\mathbf{D}(\boldsymbol{\mu})$, as well as the subspaces $\mathcal{V}(\boldsymbol{\mu})$ and $\mathcal{W}(\boldsymbol{\mu})$ characterizing oblique projections (here respectively represented by the reduced-order bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$). As

indicated in the introduction to this dissertation, a key objective of this work is to become able to adapt aeroelastic reduced-order models to new values of parameters. Interpolation is hence an attractive method for that purpose. However, reduced-order linear operators are characterized by intrinsic properties. For instance, structural reduced-order operators are symmetric positive definite matrices. Destroying this property in the interpolation process would lead to meaningless ROMs. Similarly, a subspace is characterized in practice by the definition of a basis whose column vectors span the subspace. It is hence necessary to prevent the loss of the full-column rank property of the basis in the interpolation process.

Subspaces and linear operators belong to matrix manifolds. The main challenge is therefore to be able to remain on those nonlinear manifold in the interpolation process. As mentioned in the introduction, classical interpolation method are designed to operate in vector spaces, and cannot be directly applied as such. In the present work, differential geometry properties of the Riemannian manifolds involved are exploited in order to produce meaningful interpolated quantities.

3.1.2   Problem of Interest

Let $\mathcal{M}$ be a matrix manifold and let $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_{N_p}) \in \mathbb{R}^{N_p}$ be a N_p-tuple defining a point in some parameter space. For each point $\boldsymbol{\mu}$, an element $\mathcal{Y}(\boldsymbol{\mu})$ of $\mathcal{M}$ is associated.

Problem: Given a set $Y = \{\mathcal{Y}_j\}_{j=1}^{N_R} = \{\mathcal{Y}(\boldsymbol{\mu}^{(j)})\}_{j=1}^{N_R}$ of such elements corresponding to N_R points $\{\boldsymbol{\mu}^{(j)}\}_{j=1}^{N_R}$ in the parameter space, construct by interpolation an element $\mathcal{Y}_{N_R+1}$ of $\mathcal{M}$ associated with a different point of the parameter space $\boldsymbol{\mu}^{(N_R+1)} \neq \boldsymbol{\mu}^{(j)}, \ \forall j = 1, \cdots, N_R$.

As mentioned in the introduction to this dissertation, the proposed method to solve this problem is based on concepts from the field of differential geometry. In order to make this chapter self-contained, some of those concepts and definitions are presented in the next three sections. The general interpolation algorithm is then presented in section 3.5. In sections 3.6 and 3.7, two matrix manifolds of great importance for the model reduction of aeroelastic systems are studied. Finally, numerical results of a simple academic mass-spring-damper system problems are presented to verify the interpolation method.

3.2 Definitions

3.2.1 Manifolds

A *manifold* is defined as a set of points such that any of these points has a neighborhood which is homeomorphic to an open set of an Euclidian space [65]. More rigorous definitions involving the concepts of charts and atlases between the manifold and the Euclidian space can be found in [65, 66, 67, 68, 69].

Such a manifold has a dimension which is equal to the dimension of the Euclidian space it is homeomorphic to at each point.

3.2.2 Embedded Manifolds and Quotient Manifolds of $\mathbb{R}^{n \times k}$

In the remainder of this dissertation, the manifolds of interest are matrix manifolds obtained as embedded and quotient manifolds of the manifold $\mathbb{R}^{n \times k}$. The reason for this choice is that these manifolds are very suitable for numerical calculation: their definitions involve concepts that are intrinsic to linear algebra, such as full column-rank, orthogonality or positivity of matrices. Hence, the methods developed in the

field of numerical linear algebra are particularly useful for such manifolds since they provide numerically tractable and stable algorithms. The textbook written by Absil et al. [66] is a very good reference on the topic of matrix manifolds and optimization techniques applied to these manifolds. Several concepts and notations used in this section refer directly to the ones presented in that reference.

Let $\mathcal{M}$ denote a manifold in $\mathbb{R}^{n \times k}$. $\mathcal{M}$ is said to be an immersed submanifold of $\mathbb{R}^{n \times k}$ if the function that associates each element of $\mathcal{M}$ with the same element in $\mathbb{R}^{n \times k}$ is an immersion. When this is the case, and the manifold topology of $\mathcal{M}$ coincides with the subspace topology induced from the manifold topology of $\mathbb{R}^{n \times k}$, $\mathcal{M}$ is called an *embedded manifold* of $\mathbb{R}^{n \times k}$.

The concept of quotient manifolds is defined when a manifold $\mathcal{M}$ can be provided with an equivalence relation $\sim$. In that case, for each element $\mathcal{Y}$ in $\mathcal{M}$, the set of all elements that are equivalent to $\mathcal{Y}$ is called the *equivalent class* containing $\mathcal{Y}$ and is denoted by $[\mathcal{Y}]$. The set of all equivalent classes of the equivalence relation in $\mathcal{M}$ is a manifold, called the *quotient manifold* of $\mathcal{M}$ by $\sim$. It is denoted as:

$$\mathcal{M}/\sim \ = \{[\mathcal{Y}] : \mathcal{Y} \in \mathcal{M}\}. \tag{3.2}$$

The mapping that associates each element $\mathcal{Y}$ in $\mathcal{M}$ with its equivalent class $[\mathcal{Y}]$ is called the *canonical projection.*

3.3 Matrix Manifolds of Interest

This section presents definitions and characterizations of the matrix manifolds that will be encountered in this dissertation. Some of these manifolds, such as the sphere provide intuitive illustrations of complex concepts. Others will appear as manifolds

of interest to represent reduced-order information.

3.3.1   Embedded Manifolds

The Sphere

The n dimensional sphere $\mathbb{S}^n$ is the set of vectors in $\mathbb{R}^n$ of unitary 2-norm.

$$\mathbb{S}^n = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_2 = 1\} \tag{3.3}$$

It is an embedded manifold of $\mathbb{R}^n$.

The Manifold of Non-Singular Matrices

GL(n) denotes the set of invertible matrices in $\mathbb{R}^{n \times n}$. It is an embedded manifold of $\mathbb{R}^{n \times n}$. It is also referred to as the General Linear Group.

The Manifold of Symmetric Definite Positive Matrices

SPD(n) denotes the set of symmetric definite positive matrices in $\mathbb{R}^{n \times n}$. It is also an embedded manifold of $\mathbb{R}^{n \times n}$.

The Manifold of Orthogonal Matrices

O(n) denotes the set of orthogonal matrices in $\mathbb{R}^{n \times n}$.

$$O(n) = \{\mathbf{Q} \in \mathbb{R}^{n \times n} \mid \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_n\} \tag{3.4}$$

It is also an embedded manifold of $\mathbb{R}^{n \times n}$.

The Non-Compact and Compact Stiefel Manifolds

$\mathbb{R}_*^{n \times k}$ denotes the *non-compact Stiefel manifold* which is defined as the set of full column-rank $n \times k$ real matrices.

$$\mathbb{R}_*^{n \times k} = \{\mathbf{Y} \in \mathbb{R}^{n \times k} \mid \mathrm{rank}(\mathbf{Y}) = k\} \tag{3.5}$$

$\mathcal{ST}(k, n)$ is the *compact Stiefel manifold* which is the set of matrices $\mathbf{Y} \in \mathbb{R}^{n \times k}$ satisfying $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}_k$.

$$\mathcal{ST}(k, n) = \{\mathbf{Y} \in \mathbb{R}^{n \times k} \mid \mathbf{Y}^T\mathbf{Y} = \mathbf{I}_k\} \tag{3.6}$$

$\mathbb{R}_*^{n \times k}$ and $\mathcal{ST}(k, n)$ are both embedded manifolds of $\mathbb{R}^{n \times k}$.

For some particular choices of k, $\mathbb{R}_*^{n \times k}$ and $\mathcal{ST}(k, n)$ are identical to some of the aforementioned matrix manifolds:

$$\mathbb{R}_*^{n \times 1} = \mathbb{R}^n - \{0\} \tag{3.7}$$

$$\mathcal{ST}(1, n) = \mathbb{S}^n \tag{3.8}$$

$$\mathbb{R}_*^{n \times n} = \mathrm{GL}(n) \tag{3.9}$$

$$\mathcal{ST}(n, n) = \mathrm{O}(n) \tag{3.10}$$

3.3.2   Quotient Manifold

The Grassmannian Manifold

$\mathcal{G}(k, n)$ denotes the *Grassmann manifold* which is defined as the set of k-dimensional linear subspaces of $\mathbb{R}^n$.

$$\mathcal{G}(k, n) = \{\mathcal{S} \subseteq \mathbb{R}^n \mid \exists \mathbf{Y} \in \mathbb{R}^{n \times k}, \ \mathrm{span}\{\mathbf{Y}\} = \mathcal{S}, \ \mathrm{rank}(\mathbf{Y}) = k\} \qquad (3.11)$$

The Grassmann manifold can be considered as a quotient manifold in the sense of section 3.2.2: for a given subspace in $\mathbb{R}^n$, one can indeed construct a class of bases spanning that subspace. $\mathcal{G}(k, n)$ is therefore a quotient manifold in one of the two following fashions.

1. For each subspace $\mathcal{S} \in \mathcal{G}(k, n)$ there exists an equivalent class of orthonormal matrices in $\mathcal{ST}(k, n)$ whose column space is $\mathcal{S}$. In terms of group action, it can be written that $\mathcal{G}(k, n) = \mathcal{ST}(k, n)/\mathrm{O}(k)$. $\mathcal{G}(k, n)$ is therefore a quotient manifold.

2. Similarly, one can write $\mathcal{G}(k, n) = \mathbb{R}_*^{n \times k}/\mathrm{GL}(k)$.

In a general form and in the remainder of this chapter, $\mathcal{G}(k, n) = \mathcal{M}/\mathcal{N}$ where $\mathcal{M}$ denotes either the compact or the non-compact Stiefel manifold and $\mathcal{N}$ the manifold of Orthogonal matrices or the General Linear group respectively. The equivalence relations can be written, for $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathcal{M}$, as

$$\mathbf{Y}_1 \sim \mathbf{Y}_2 \ \Leftrightarrow \ \mathrm{span}\{\mathbf{Y}_1\} = \mathrm{span}\{\mathbf{Y}_2\} \ \Leftrightarrow \ \exists \mathbf{Q} \in \mathcal{N} \mid \mathbf{Y}_1 = \mathbf{Y}_2 \mathbf{Q}. \qquad (3.12)$$

The underlying canonical projection is $\mathrm{span}\{\cdot\}$.

3.4   Background in Differential Geometry

The materials presented in this section are basic concepts in differential geometry. More details about these concepts can be found in many textbooks on the topic [65, 66, 67, 68, 69].

3.4.1   Riemannian Manifolds

Riemannian manifolds, that is manifolds that can be associated with a differentiable structure at each of their points, are considered in the remainder of this dissertation. An important property of Riemannian manifolds is that they admit a *tangent space* of the same dimension as the manifold at each of their points. By construction, at a given element $\mathcal{Y}$ of the manifold $\mathcal{M}$, the tangent space is the set of all tangent vectors to the manifold at that point and is denoted in this thesis as $T_{\mathcal{Y}}\mathcal{M}$. It is therefore a linear space, and this property will be of great importance in the remainder of this chapter. The union of all tangent spaces to a given differentiable manifold $\mathcal{M}$ has itself a manifold structure and is called the *tangent bundle* to the manifold. It is denoted as follows:

$$T\mathcal{M} = \bigcup_{\mathcal{Y} \in \mathcal{M}} T_{\mathcal{Y}}\mathcal{M}. \tag{3.13}$$

3.4.2   Geodesics on Manifolds

Considering a Riemannian manifold $\mathcal{M}$, a *geodesic* is defined, when it exists, as the shortest path between two points belonging to this manifold. The geodesic path is usually parameterized by a continuous parameter taking values on the interval $[0, 1]$ $\{\gamma(t),\ t \in [0, 1]\}$. Geodesics are defined by second-order differential equations and, as such, are characterized by an initial point γ_0 and an initial velocity $\dot{\gamma}_0 \in T_{\gamma_0}\mathcal{M}$.

A geodesic is defined as the unique path such that the tangent vector to this path is parallel transported along it [70]. In order to emphasize the dependence of the geodesic path on its initial conditions, it is denoted here as $\gamma(t; \gamma_0, \dot{\gamma}_0)$.

3.4.3 Exponential and Logarithm Mapping

Definitions

Let $\gamma(t; \gamma_0, \dot{\gamma}_0)$ be a geodesic path as defined in the previous section. One can define the *exponential mapping* as the final extremity point of the geodesic:

$$\mathrm{Exp}_{\gamma_0} : T_{\gamma_0}\mathcal{M} \quad \rightarrow \quad \mathcal{M} \tag{3.14}$$

$$\dot{\gamma}_0 \quad \longmapsto \quad \gamma(1; \gamma_0, \dot{\gamma}_0). \tag{3.15}$$

An exponential mapping can thus be defined at each point γ_0 of the manifold. Note that the exponential mapping is a particular case of a more general class of mappings called *retractions* [66].

Considering now two points $\mathcal{Y}_1, \mathcal{Y}_2 \in \mathcal{M}$ that are connected by a unique geodesic γ, the *logarithm mapping* $\mathrm{Log}_{\mathcal{Y}_1}$ maps $\mathcal{Y}_2$ to the unique velocity vector $\mathcal{V}$ satisfying $\gamma(1; \mathcal{Y}_1, \mathcal{V}) = \mathcal{Y}_2$. The logarithm mapping is the inverse of the exponential mapping when such an inverse can be defined[1]. $\mathrm{Log}_{\mathcal{Y}_1}(\mathcal{Y}_2)$ belongs to the tangent space $T_{\mathcal{Y}_1}\mathcal{M}$. When the element $\mathrm{Log}_{\mathcal{Y}_1}(\mathcal{Y}_2)$ exists, there is a unique geodesic joining the two elements $\mathcal{Y}_1$ and $\mathcal{Y}_2$ on the manifold. Therefore, this geodesic can be uniquely defined either by its two extremities $\mathcal{Y}_1$ and $\mathcal{Y}_2$ or by its initial point $\mathcal{Y}_1$ and initial velocity $\mathcal{V} =$

---

[1]The logarithm mapping is therefore defined in a neighborhood of $\mathcal{Y}_1$. This property can be explained by the fact that geodesics with different initial velocities and emitted from a same point can cross (consider for instance the meridians on a sphere emitted from the north pole that cross at the south pole). Hence, the logarithm mapping is not defined at the crossing points as different initial velocities lead to these points (see Fig. 3.2).

(a) Initial velocity   (b) Final point
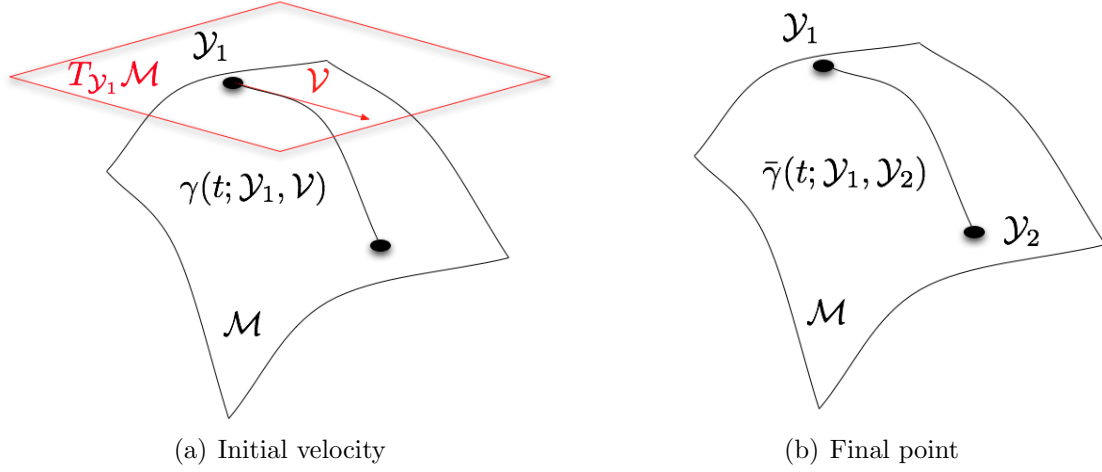
Figure 3.1: Dual notations for geodesic paths.

$\text{Log}_{\mathcal{Y}_1}(\mathcal{Y}_2)$. As a consequence, in the remainder of this paper, the following dual notations for such a geodesic curve are used (see Fig. 3.1):

$$\gamma(t; \mathcal{Y}_1, \mathcal{V}) = \bar{\gamma}(t; \mathcal{Y}_1, \mathcal{Y}_2). \tag{3.16}$$

γ denotes the geodesic when the initial conditions are provided and $\bar{\gamma}$ when its initial and final points are specified. The following property holds

Property.

$$\bar{\gamma}(t; \mathcal{Y}_1, \mathcal{Y}_2) = \gamma\left(t; \mathcal{Y}_1, \text{Log}_{\mathcal{Y}_1}(\mathcal{Y}_2)\right). \tag{3.17}$$
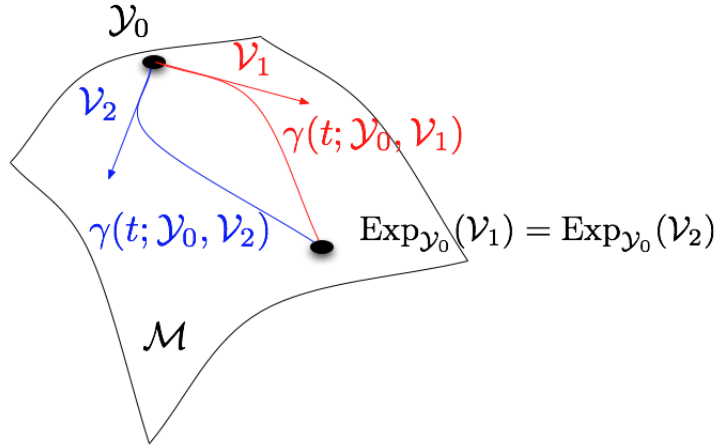
Figure 3.2: A point where the logarithm mapping is not defined

Analytic Formulae for Exponential and Logarithm Mappings of Common Matrix Manifolds

Tables 3.1, 3.2, 3.3 and 3.4 present analytic formulae for the Exponential mappings of the matrix manifolds mentioned in section 3.3. When the inverse of these mappings can be defined and analytic expressions exists, the logarithm mappings are reported as well. Finally, parameterizations of the tangent spaces at a point of these manifolds are also shown[2]. The reader can check that the logarithms effectively belong to that tangent space. The dimension of the tangent space also provides an insight on the dimension of the underlying manifold since the two are equal. These dimensions are also reported. The expressions of the mappings for the Grassmann manifold are derived in section 3.7.3, as they provide important insight on their geometrical properties.

---

[2]Tangent spaces are here defined as linear spaces. However, tangent spaces are rigorously affine spaces, their origin being the point of the manifold they are defined at. As such, the subsequent definitions of tangent spaces should be modified by a change of origin if parameterizations as affine spaces are sought.

Manifold	Sphere	Non-singular matrices
Matrix representation	$\mathbf{x} \in \mathbb{S}^n$	$\mathbf{Q} \in \mathrm{GL}(n)$
Dimension	$n-1$	n^2
Tangent space parameterization	$\mathbf{x}^\perp = \{\boldsymbol{\gamma} \in \mathbb{R}^n \mid \boldsymbol{\gamma}^T\mathbf{x} = 0\}$	$\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n}$
Exponential mapping	$\mathrm{Exp}_{\mathbf{x}}(\boldsymbol{\gamma}) = \cos(\|\boldsymbol{\gamma}\|_2)\mathbf{x} + \sin(\|\boldsymbol{\gamma}\|_2)\frac{\boldsymbol{\gamma}}{\|\boldsymbol{\gamma}\|_2}$	$\mathrm{Exp}_{\mathbf{Q}}(\boldsymbol{\Gamma}) = \texttt{expm}(\boldsymbol{\Gamma})\mathbf{Q}$
Logarithm mapping	$\mathrm{Log}_{\mathbf{x}}(\mathbf{y}) = \cos^{-1}(\mathbf{y}^T\mathbf{x})\frac{\mathbf{y}-(\mathbf{y}^T\mathbf{x})\mathbf{x}}{\|\mathbf{y}-(\mathbf{y}^T\mathbf{x})\mathbf{x}\|_2}$	$\mathrm{Log}_{\mathbf{Q}}(\mathbf{P}) = \texttt{logm}(\mathbf{P}\mathbf{Q}^{-1})$

Table 3.1: Exponential and Logarithm mappings for some matrix manifolds (1).

Manifold	SPD matrices	Orthogonal matrices
Matrix representation	$\mathbf{Q} \in \mathrm{SPD}(n)$	$\mathbf{Q} \in \mathrm{O}(n)$
Dimension	$\frac{n(n+1)}{2}$	$\frac{n(n-1)}{2}$
Tangent space parameterization	$\mathrm{S}(n) = \{\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n} \mid \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^T\}$	$\{\boldsymbol{\Gamma} = \mathbf{Q}\boldsymbol{\Omega} \mid \boldsymbol{\Omega} + \boldsymbol{\Omega}^T = \mathbf{0}_{n,n}\}$
Exponential mapping	$\mathrm{Exp}_{\mathbf{Q}}(\boldsymbol{\Gamma}) = \mathbf{Q}^{1/2}\texttt{expm}(\boldsymbol{\Gamma})\mathbf{Q}^{1/2}$	$\mathrm{Exp}_{\mathbf{Q}}(\boldsymbol{\Gamma}) = \mathbf{Q}\texttt{expm}(\boldsymbol{\Gamma})$
Logarithm mapping	$\mathrm{Log}_{\mathbf{Q}}(\mathbf{P}) = \texttt{logm}\left(\mathbf{Q}^{-1/2}\mathbf{P}\mathbf{Q}^{-1/2}\right)$	$\mathrm{Log}_{\mathbf{Q}}(\mathbf{P}) = \texttt{logm}(\mathbf{Q}^T\mathbf{P})$

Table 3.2: Exponential and Logarithm mappings for some matrix manifolds (2).

Manifold	Compact Stiefel manifold	Grassmann manifold
Matrix representation	$\mathbf{Y} \in \mathcal{ST}(k, n)$	$\mathcal{Y} \in \mathcal{G}(k, n) \leftrightarrow \mathbf{Y} \in \mathcal{ST}(k, n)$
Dimension	$\frac{k(k-1)}{2} + k(n-k)$	$k(n-k)$
Tangent space parameterization	$\{\mathbf{\Gamma} \in \mathbb{R}^{n \times k} \mid \mathbf{Y}^T\mathbf{\Gamma} + \mathbf{\Gamma}^T\mathbf{Y} = \mathbf{0}_{k,k}\}$	$\{\mathcal{V} \leftrightarrow \mathbf{\Gamma} \in \mathbb{R}^{n \times k} \mid \mathbf{Y}^T\mathbf{\Gamma} = \mathbf{0}_{k,k}\}$
Exponential mapping	$\mathbf{QR} = (\mathbf{I}_n - \mathbf{YY}^T)\mathbf{\Gamma}$ (compact QR decomposition) $\begin{bmatrix} \mathbf{M} \\ \mathbf{N} \end{bmatrix} = \texttt{expm} \begin{bmatrix} \mathbf{Y}^T\mathbf{\Gamma} & -\mathbf{R}^T \\ \mathbf{R} & \mathbf{0}_{k,k} \end{bmatrix} \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0}_{k,k} \end{bmatrix}$ $\mathrm{Exp}_{\mathbf{Y}}(\mathbf{\Gamma}) = \mathbf{YM} + \mathbf{QN}$	$\mathbf{\Gamma} = \mathbf{U\Sigma V}^T$ (thin SVD) $\mathrm{Exp}_{\mathcal{Y}}(\gamma) = \mathrm{span}\{\mathbf{YV}\mathbf{cos}(\mathbf{\Sigma}) + \mathbf{U}\mathbf{sin}(\mathbf{\Sigma})\}$
Logarithm mapping	?	$(\mathbf{I} - \mathbf{YY}^T)\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1} = \mathbf{U\Sigma V}^T$ (thin SVD) $\mathrm{Log}_{\mathcal{Y}}(\mathcal{Z}) \leftrightarrow \mathbf{\Gamma} = \mathbf{U}\mathbf{tan}^{-1}(\mathbf{\Sigma})\mathbf{V}^T$

Table 3.3: Exponential and Logarithm mappings for some matrix manifolds (3).

Manifold	Grassmann manifold
Matrix representation	$\mathcal{Y} \in \mathcal{G}(k,n) \leftrightarrow \mathbf{Y} \in \mathbb{R}_*^{n \times k}$
Dimension	$k(n-k)$
Tangent space parameterization	$\{\mathcal{V} \leftrightarrow \mathbf{\Gamma} \in \mathbb{R}^{n \times k} \mid \mathbf{Y}^T\mathbf{\Gamma} = \mathbf{0}_{k,k}\}$
Exponential mapping	$\mathbf{\Gamma} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ (thin SVD) $\operatorname{Exp}_{\mathcal{Y}}(\gamma) = \operatorname{span}\{\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}}\mathbf{V}\mathbf{cos}(\mathbf{\Sigma}) + \mathbf{U}\mathbf{sin}(\mathbf{\Sigma})\}$
Logarithm mapping	$(\mathbf{I} - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T)\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ (thin SVD) $\operatorname{Log}_{\mathcal{Y}}(\mathcal{Z}) \leftrightarrow \mathbf{\Gamma} = \mathbf{U}\mathbf{tan}^{-1}(\mathbf{\Sigma})\mathbf{V}^T$

Table 3.4: Exponential and Logarithm mappings for some matrix manifolds (4).

From Table 3.3, one can notice that although $\mathcal{ST}(k,n)$ and $\mathcal{G}(k,n)$ are represented by matrices belonging to the same manifold, their respective mappings differ. This is due to the fact that $\mathcal{ST}(k,n)$ is treated as an embedded manifold while $\mathcal{G}(k,n)$ is a quotient manifold.

A Shooting Technique for Computing Logarithm Mappings

As seen from Table 3.3, there is no known analytic formula for the logarithm mapping on the compact Stiefel manifold. While the Exponential mapping has an analytical formulation, this is not the case for the Logarithm mapping. The reason is that the matrices $\mathbf{M}$ and $\mathbf{N}$ in the formulation of the Exponential mapping are computed by truncating a few columns of a square matrix. This truncation suggests that this formula cannot be inverted in order to establish an analytic formula for the Logarithm

mapping.

However, this mapping can be approximated numerically using the method presented in the present section. It is based on a shooting method and is, to the best of the author's knowledge, the first method developed for computing such a mapping.

The method is based upon the observation that the Logarithm and Exponential mappings have an interpretation in terms of geodesics: given an initial position and velocity, the Exponential mapping computes the corresponding final point, whereas the Logarithm mapping takes as an input the initial and final points and outputs the corresponding initial velocity (see Fig. 3.1).

Therefore, the shooting method is a natural way to exploit these properties as follows: Given two elements $\mathbf{Y}_0, \mathbf{Y}_1 \in \mathcal{ST}(k, n)$, the method proceeds by computing $\mathrm{Log}_{\mathbf{Y}_0}(\mathbf{Y}_1)$ as

$$\mathrm{Log}_{\mathbf{Y}_0}(\mathbf{Y}_1) = \underset{\mathbf{\Gamma} \in T_{\mathbf{Y}_0}\mathcal{ST}(k,n)}{\arg \min} \|\mathrm{Exp}_{\mathbf{Y}_0}(\mathbf{\Gamma}) - \mathbf{Y}_1\|^2 \tag{3.18}$$

for some appropriate norm $\| \ \|$.

The Frobenius norm $\| \ \|_F$ for n-by-k real matrices is chosen here. It is defined as

$$\|\mathbf{\Delta}\|_F = \left[\mathrm{tr}(\mathbf{\Delta}^T \mathbf{\Delta})\right]^{\frac{1}{2}}. \tag{3.19}$$

Thus, denoting by F the following objective function

$$F(\mathbf{\Gamma}) = \|\mathrm{Exp}_{\mathbf{Y}_0}(\mathbf{\Gamma}) - \mathbf{Y}_1\|_F^2, \tag{3.20}$$

the global minimum for F corresponds to the Logarithm mapping.

Any vector $\mathbf{\Gamma}$ belonging to the tangent space at $\mathbf{Y}_0$ to the Stiefel manifold $\mathcal{ST}(k, n)$

can be decomposed into two orthogonal components:

$$\mathbf{\Gamma} = \mathbf{Y}_0\mathbf{A} + \mathbf{Z}\mathbf{B}, \tag{3.21}$$

where $\mathbf{Z} \in \mathcal{ST}(k, n-k)$ satisfies $\mathbf{Z}^T\mathbf{Y}_0 = \mathbf{0}_{k,k}$, $\mathbf{B} \in \mathbb{R}^{(n-k)\times k}$ and $\mathbf{A} \in \mathbb{R}^{k\times k}$ is a skew-symmetric matrix. Therefore, if a basis $\mathbf{Z}$ for the orthogonal subspace to $\mathbf{Y}_0$ is first pre-computed, the minimization problem becomes

$$\min_{\substack{\mathbf{A}\in\mathbb{R}^{k\times k} \\ \text{s.t. } \mathbf{A}+\mathbf{A}^T=\mathbf{0}_{k,k} \\ \mathbf{B}\in\mathbb{R}^{(n-k)\times k}}} \tilde{F}(\mathbf{A}, \mathbf{B}) = \|\text{Exp}_{\mathbf{Y}_0}(\mathbf{Y}_0\mathbf{A} + \mathbf{Z}\mathbf{B}) - \mathbf{Y}_1\|_F^2. \tag{3.22}$$

This reduces the problem to an unconstrained minimization problem of a univariate function with respect to $n_{DOF} = \dfrac{k(k-1)}{2} + (n-k)k$ variables.
A Quasi-Newton method is used here for solving this unconstrained non-linear minimization problem [71]. The resulting algorithm is reported as folllows.

The expressions for the derivatives in $\nabla_\mathbf{x}\tilde{F}$ are reported in Appendix B.

Applying a version of the algorithm implemented in Matlab to two randomly generated matrices $\mathbf{Y}_0, \mathbf{Y}_1 \in \mathcal{ST}(2, 5)$ defined as

$$\mathbf{Y}_0 = \begin{bmatrix} -0.4927 & -0.4806 \\ -0.5478 & -0.3583 \\ -0.0768 & 0.4754 \\ -0.5523 & 0.3391 \\ -0.3824 & 0.5473 \end{bmatrix} \tag{3.23}$$

Algorithm 2 Shooting method for computing the logarithm map on the compact Stiefel manifold

---

1: $k \leftarrow 0$
2: Choose $\mathbf{A}_0, \mathbf{B}_0, \eta$
3: Compute $\mathbf{x}_0$. {Retain for $\mathbf{x}$ the independent entries in $\mathbf{A}$ and $\mathbf{B}$.}
4: Compute $\tilde{F}(\mathbf{x}_0)$ and $\nabla \tilde{F}_0 = \nabla_{\mathbf{x}} \tilde{F}(\mathbf{x}_0)$
5: Let $\mathbf{H}_0 = \mathbf{I}_{N_{DOF}}$
6: **repeat**
7:     Let $\mathbf{p}_k = -\mathbf{H}_k \nabla \tilde{F}_k$.
8:     Choose α_k by performing a line-search along $\mathbf{p}_k$
9:     $\boldsymbol{\mu}_k = \alpha_k \mathbf{p}_k$
10:     $\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\mu}_k$
11:     Compute $\nabla \tilde{F}_{k+1} = \nabla_{\mathbf{x}} \tilde{F}(\mathbf{x}_{k+1})$
12:     $\mathbf{g}_k = \nabla \tilde{F}_{k+1} - \nabla \tilde{F}_k$
13:     Compute $\rho_k = 1/(\mathbf{g}_k^T \boldsymbol{\mu}_k)$
14:     Compute $\mathbf{H}_{k+1} = (\mathbf{I}_{N_{DOF}} - \rho_k \boldsymbol{\mu}_k \mathbf{g}_k^T) \mathbf{H}_k (\mathbf{I}_{N_{DOF}} - \rho_k \mathbf{g}_k \boldsymbol{\mu}_k^T) + \rho_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T$ {BFGS update for the inverse approximate Hessian}
15:     $k \leftarrow k + 1$.
16: **until** $\|\tilde{F}(\mathbf{x}_k)\|_F < \eta$ {Metric for convergence}
17: $\mathbf{x} \leftarrow \mathbf{x}_k$

---

and

$$
\mathbf{Y}_1 = \begin{bmatrix} -0.0949 & 0.0021 \\ -0.5843 & -0.7103 \\ -0.5762 & 0.0994 \\ -0.2922 & 0.5751 \\ -0.4818 & 0.3934 \end{bmatrix}, \tag{3.24}
$$

one obtains

$$
\boldsymbol{\Gamma} = \begin{bmatrix} 0.2485 & 0.2459 \\ -0.3700 & -0.6575 \\ -0.5011 & -0.2864 \\ 0.2743 & 0.2184 \\ -0.0857 & -0.1011 \end{bmatrix} \tag{3.25}
$$

and thus

$$\mathrm{Exp}_{\mathbf{Y}_0}(\mathbf{\Gamma}) = \begin{bmatrix} -0.0949 & 0.0021 \\ -0.5843 & -0.7103 \\ -0.5762 & 0.0994 \\ -0.2922 & 0.5751 \\ -0.4818 & 0.3934 \end{bmatrix}. \tag{3.26}$$

Hence, one can check that

$$\mathrm{Exp}_{\mathbf{Y}_0}(\mathrm{Log}_{\mathbf{Y}_0}(\mathbf{Y}_1)) = \mathbf{Y}_1. \tag{3.27}$$

The proposed method for interpolation of quantities belonging to a given matrix manifold is presented in the next section.

3.5 Interpolation of Multivariate Functions Belonging to a Matrix Manifold

3.5.1 Background

Several methods have been recently proposed for interpolation on matrix manifolds. A first approach based on using the linear properties of the tangent space to enable interpolation was proposed by Rahman et al. for midpoint iterative refinements [72]. That approach corresponds to the restriction of the method proposed in the following section to univariate interpolation between two points. Hüper and Leite have also proposed in Ref. [73] an interpolation method for points belonging to three specific matrix manifolds based on projections of data onto affine tangent spaces to the manifold using rolling maps. However that approach becomes computationally intractable

in high dimensions as it requires forming full-order square matrices, even when much cheaper representations are available, as in the case of subspaces. Other successful interpolation techniques based on differential geometry have been applied to tensor field interpolation where the matrix manifold of interest is the manifold of symmetric definite positive matrices [74]. The interpolation techniques used in that reference are linear interpolation between two points, points belonging to a lattice, and the Karcher mean. However none of these methods provides an inexpensive way to interpolate points that do not belong to a regular grid: the Karcher method is indeed an iterative method that requires evaluating a high number of logarithm and exponential maps, which can be very expensive for high-dimensional system. All these interpolation methods have in common the mapping of points belonging to a non-linear manifold to some affine tangent space. The new approach presented in the subsequent section generalizes these methods under an unified and rigorous framework.

The fact that the present algorithm is able to operate on matrix quantities built for values of the parameters that (a) are multidimensional, (b) do not belong to a regular grid, is essential for its applicability to practical problems. For aeroelastic applications, the parameters define flight conditions and are, as such, multivariate since both the speed and the attitude of the aircraft need to be considered. Previously explored flight conditions may not define a regular grid in this parametric space, and, as a consequence, the generality of the present algorithm is crucial.

3.5.2   Algorithm

Let $\left\{ \mathcal{Y}_i = \mathcal{Y}\big(\boldsymbol{\mu}^{(i)}\big) \right\}_{i=1}^{N_R}$ denote a set of elements of a manifold $\mathcal{M}$ associated with a set of different operating points $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^{N_R}$. Each element $\mathcal{Y}_i$ is represented by a matrix $\mathbf{Y}_i \in \mathbb{R}^{n \times k}$ that belongs to some matrix manifold $\mathcal{M}'$ and verifies one or more specific

properties that characterize $\mathcal{M}'^3$. The following four-step method is proposed to construct a new element $\mathcal{Y}_{N_R+1} \in \mathcal{M}$ associated with a new operating point $\boldsymbol{\mu}^{(N_R+1)}$ and its representative matrix $\mathbf{Y}_{N_R+1}$ — that is, an element $\mathcal{Y}_{N_R+1} \in \mathcal{M}$ and its representative matrix $\mathbf{Y}_{N_R+1} \in \mathcal{M}'$ which have the same properties as each element $\mathcal{Y}_i$ and its representative matrix $\mathbf{Y}_i$, $i = 1, \cdots, N_R$, respectively.

- Step 0. Choose an element $\mathcal{Y}_{i_0}$ in the data set $\{\mathcal{Y}_i\}_{i=1}^{N_R}$ as a reference element of the manifold $\mathcal{M}$.

- Step 1. Consider a few elements of the set $\{\mathcal{Y}_i\}_{i=1}^{N_R}$ that lie in a neighborhood $\mathcal{U}_{\mathcal{Y}_{i_0}}$ of $\mathcal{Y}_{i_0}{}^4$. Map each of them onto the tangent space to $\mathcal{M}$ at $\mathcal{Y}_{i_0}$ denoted by $T_{\mathcal{Y}_{i_0}}\mathcal{M}$. More specifically, map each element $\mathcal{Y}_i$ that is sufficiently close to $\mathcal{Y}_{i_0}$ to an element $\mathcal{V}_i \in T_{\mathcal{Y}_{i_0}}\mathcal{M}$ represented by a matrix $\boldsymbol{\Gamma}_i$, using the logarithm map $\mathrm{Log}_{\mathcal{Y}_{i_0}}$ which provides an appropriate continuous mapping to the tangent space of the manifold at $\mathcal{Y}_{i_0}$. This can be written as

$$\mathcal{V}_i = \mathrm{Log}_{\mathcal{Y}_{i_0}}(\mathcal{Y}_i). \tag{3.28}$$

- Step 2. Compute each entry of an $n \times k$ matrix $\boldsymbol{\Gamma}_{N_R+1}$ associated with the target operating point $\boldsymbol{\mu}^{(N_R+1)}$ by interpolating the corresponding entries of the $n \times k$ matrices $\{\boldsymbol{\Gamma}_i\}$ associated with the operating points $\{\boldsymbol{\mu}^{(i)}\}$ using any preferred multivariate interpolation algorithm.

- Step 3. Map the element $\mathcal{V}_{N_R+1} \in T_{\mathcal{Y}_{i_0}}\mathcal{M}$ represented by the matrix $\boldsymbol{\Gamma}_{N_R+1}$ to an element $\mathcal{Y}_{N_R+1} \in \mathcal{M}$ represented by a matrix $\mathbf{Y}_{N_R+1} \in \mathcal{M}'$ using the

---

³When $\mathcal{M}$ is an embedded manifold of $\mathbb{R}^{n \times k}$, $\mathcal{M}' = \mathcal{M}$ and $\mathcal{Y}_i = \mathbf{Y}_i$, $i = 1, \cdots, N_R$.

⁴The requirement is due to the fact that Logarithm mappings have to be computed. More specifically, logarithm mappings are only defined in a neighborhood of the reference point, as they are inverse mappings.

exponential map $\text{Exp}_{\mathcal{Y}_{i_0}}$. This can be written as

$$\mathcal{Y}_{N_R+1} = \text{Exp}_{\mathcal{Y}_{i_0}}(\mathcal{V}_{N_R+1}). \tag{3.29}$$

The specific algorithms for computing the logarithm and exponential mappings are dependent on the manifolds $\mathcal{M}$ and $\mathcal{M}'$.

A compact interpolation scheme can be established by adding information about the sensitivities of the matrix representations $\mathbf{Y}_i$ with respect to the parameters $\mu_1, \cdots, \mu_{N_p}$:

$$\nabla \mathbf{Y}_i = \begin{bmatrix} \dfrac{\partial \mathbf{Y}_i}{\partial \mu_1} & \cdots & \dfrac{\partial \mathbf{Y}_i}{\partial \mu_{N_p}} \end{bmatrix}. \tag{3.30}$$

Remark 1. This algorithm can be generalized to any mapping based on a retraction on the manifold $\mathcal{M}$ associated with a local inverse mapping. As noted by the authors in Ref. [66], retractions can be sometimes less computationally expensive to evaluate than their exponential mapping counterparts for a given matrix manifold. Hence, an analog interpolation algorithm based on a retraction map could be, in some case, computationally cost efficient, while still ensuring that the interpolated quantity belongs to the manifold.

Remark 2. Because the logarithmic map $\text{Log}_{\mathcal{Y}}$ is defined in a neighborhood of $\mathcal{Y} \in \mathcal{M}$, the interpolation method described above should not be sensitive, if the neighborhood is small, to the choice of the reference point $\mathcal{Y}_{i_0}$ made in Step 0. This is confirmed in practice as will be shown in section 4.6.2.
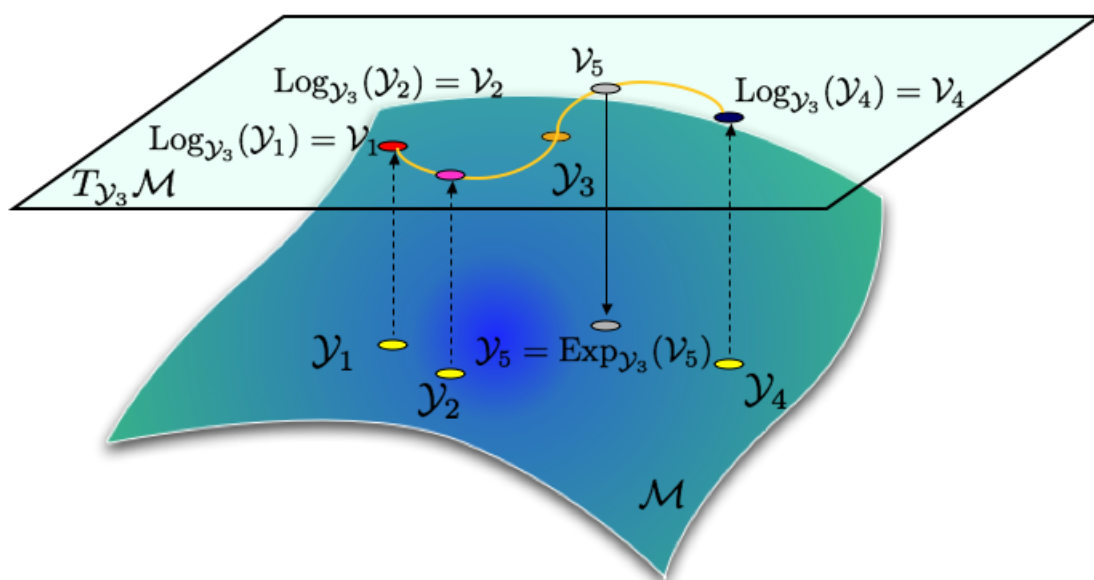
Figure 3.3: Graphical description of the generalized interpolation of the elements $\{\mathcal{Y}_i\}_{i=1}^4$ in a tangent space to a matrix manifold $\mathcal{M}$.

3.5.3 Geometric Interpretation

As presented in section 3.4.3, the exponential and logarithm mappings have a physical interpretation in term of geodesics: the exponential mapping defined at a point $\mathcal{Y}$ of a manifold $\mathcal{M}$ maps initial velocities $\mathcal{V}$ defined at $\mathcal{Y}$ to the final point $\gamma(1; \mathcal{Y}, \mathcal{V})$ of the corresponding geodesic. Its inverse mapping being the logarithm mapping, the proposed algorithm can be naturally interpreted as a method that proceeds by interpolating initial velocities defining geodesic curves emitted at a fixed point. The strength of the proposed algorithm is the fact that the nonlinearity properties of the manifolds of interest are dealt with by considering geodesic paths: each of these geodesics is guaranteed to lie on the manifold.

3.5.4 A Simple Example: Interpolation of Points on a Circle

In order to provide the reader with an intuitive application, a set of points $\{P(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R}$ belonging to a unitary one-dimensional sphere (circle) are considered. As such, each one of this points can be uniquely associated to an angle α_i, $i = 1, \cdots, N_R$ measuring the relative position of each point with respect to a reference radius of the sphere. Each point $P(\boldsymbol{\mu}^{(i)})$ can hence be represented by a two-dimensional vector

$$\mathbf{y}_i = \begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \end{bmatrix}, \ i = 1, \cdots, N_R. \tag{3.31}$$

The interpolation algorithm is then followed step-by-step

- Step 0. A point $P(\boldsymbol{\mu}^{(i_0)})$ represented by $\mathbf{y}_{i_0}$ is chosen as a reference point. Let

$$
\mathbf{z} = \begin{bmatrix} -\sin(\alpha_{i_0}) \\ \cos(\alpha_{i_0}) \end{bmatrix} \in \mathbf{y}_{i_0}^{\perp} \tag{3.32}
$$

- Step 1. Map each element $\mathbf{y}_i$ that lie in the neighborhood of $\mathbf{y}_{i_0}$ (the points that are not antipodal to $\mathbf{y}_{i_0}$) to $T_{\mathbf{y}_{i_0}}\mathbb{S}^2$ as

$$
\boldsymbol{\gamma}_i = \mathrm{Log}_{\mathbf{y}_{i_0}}(\mathbf{y}_i) = \cos^{-1}(\mathbf{y}_i^T \mathbf{y}_{i_0}) \frac{\mathbf{y}_i - (\mathbf{y}_i^T \mathbf{y}_{i_0})\mathbf{y}_{i_0}}{\|\mathbf{y}_i - (\mathbf{y}_i^T \mathbf{y}_{i_0})\mathbf{y}_{i_0}\|_2}. \tag{3.33}
$$

Since

$$
\mathbf{y}_i^T \mathbf{y}_{i_0} = \cos(\alpha_i)\cos(\alpha_{i_0}) + \sin(\alpha_i)\sin(\alpha_{i_0}) = \cos(\alpha_i - \alpha_{i_0}), \tag{3.34}
$$

and

$$
\frac{\mathbf{y}_i - (\mathbf{y}_i^T \mathbf{y}_{i_0})\mathbf{y}_{i_0}}{\|\mathbf{y}_i - (\mathbf{y}_i^T \mathbf{y}_{i_0})\mathbf{y}_{i_0}\|_2} = \frac{(\mathbf{y}_i^T \mathbf{z})\mathbf{z}}{\|(\mathbf{y}_i^T \mathbf{z})\mathbf{z}\|_2} = \frac{\sin(\alpha_i - \alpha_{i_0})}{|\sin(\alpha_i - \alpha_{i_0})|}\mathbf{z}, \tag{3.35}
$$

one obtains the simpler expression

$$
\boldsymbol{\gamma}_i = \cos^{-1}(\cos(\alpha_i - \alpha_{i_0})) \frac{\sin(\alpha_i - \alpha_{i_0})}{|\sin(\alpha_i - \alpha_{i_0})|}\mathbf{z} = |\alpha_i - \alpha_{i_0}| \frac{\sin(\alpha_i - \alpha_{i_0})}{|\sin(\alpha_i - \alpha_{i_0})|}\mathbf{z} = (\alpha_i - \alpha_{i_0})\mathbf{z} \tag{3.36}
$$

- Step 2. Compute $\boldsymbol{\gamma}_{N_R+1}$ associated with $\mu^{(N_R+1)}$ by interpolating the entries of the vectors $\boldsymbol{\gamma}_i$, $i = 1, \cdots, N_R$. Necessarily, the particular form of the vectors $\boldsymbol{\gamma}_i$ implies that $\boldsymbol{\gamma}_{N_R+1}$ will be of the form

$$
\boldsymbol{\gamma}_{N_R+1} = (\alpha_{N_R+1} - \alpha_{i_0})\mathbf{z}, \tag{3.37}
$$

where $\alpha_{N_R+1} - \alpha_{i_0}$ is interpolated from the quantities $\{\alpha_i - \alpha_{i_0}\}_{i=1}^{N_R}$.

- Step 3. $\boldsymbol{\gamma}_{N_R+1}$ is mapped to

$$
\begin{aligned}
\mathbf{y}_{N_R+1} &= \mathrm{Exp}_{\mathbf{y}_{i_0}}(\boldsymbol{\gamma}_{N_R+1}) \\
&= \cos(\|\boldsymbol{\gamma}_{N_R+1}\|_2)\mathbf{y}_{i_0} + \sin(\|\boldsymbol{\gamma}_{N_R+1}\|_2)\frac{\boldsymbol{\gamma}_{N_R+1}}{\|\boldsymbol{\gamma}_{N_R+1}\|_2} \\
&= \cos(|\alpha_{N_R+1} - \alpha_{i_0}|)\mathbf{y}_{i_0} + \sin(|\alpha_{N_R+1} - \alpha_{i_0}|)\mathrm{sign}(\alpha_{N_R+1} - \alpha_{i_0})\mathbf{z} \\
&= \cos(\alpha_{N_R+1} - \alpha_{i_0})\mathbf{y}_{i_0} + \sin(\alpha_{N_R+1} - \alpha_{i_0})\mathbf{z} \\
&= \begin{bmatrix} \cos(\alpha_{N_R+1} - \alpha_{i_0})\cos(\alpha_{i_0}) - \sin(\alpha_{N_R+1} - \alpha_{i_0})\sin(\alpha_{i_0}) \\ \cos(\alpha_{N_R+1} - \alpha_{i_0})\sin(\alpha_{i_0}) + \sin(\alpha_{N_R+1} - \alpha_{i_0})\cos(\alpha_{i_0}) \end{bmatrix} \\
&= \begin{bmatrix} \cos(\alpha_{N_R+1}) \\ \sin(\alpha_{N_R+1}) \end{bmatrix}
\end{aligned}
\tag{3.38}
$$

Hence, the algorithm has proceeded by naturally interpolating the angles measuring the distance of the pre-computed points $\{P(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R}$ to the reference pre-computed point $P(\boldsymbol{\mu}^{(i_0)})$.

3.6 Application to the Manifold of Symmetric Definite Positive Matrices

3.6.1 Motivation

As mentioned in the introduction, structural reduced-order operators have the property of symmetric positive definiteness. As such, the corresponding manifold is the focus of the present section. Furthermore, important properties characterizing structural ROM built by modal truncation are shown to be preserved by the proposed

interpolation algorithm.

3.6.2 Compact Interpolation

The sensitivities of the logarithm mapping for the SPD manifold with respect to the parameters $(\mu_1, \cdots, \mu_{N_p})$ can be computed by finite differences. The algorithm developed in section 3.5.2 can then be applied using the logarithm and exponential mappings for the SPD manifold presented in section 3.4.3.

3.6.3 Preservation Properties

Important propositions characterizing the proposed generalized interpolation method in a tangent space to a matrix manifold are here proven.

Let $n \in \mathbb{N}^*$. In this section, a method for interpolation in $\mathbb{R}^{n \times n}$ is considered. This method proceeds by interpolating independently each entry of a set of matrices $\{\mathbf{A}_i\}_{i=1}^{N_R}$, where $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $i = 1, \cdots, N_R$. The result of such an interpolation will be denoted by $\mathcal{I}_{\mathbb{R}^{n \times n}}[\mathbf{A}_i]$.

Furthermore, $\mathcal{I}_{\mathbb{R}^{n \times n}}$ is assumed to satisfy the very general property

$$\mathcal{I}_{\mathbb{R}^{n \times n}}[-\mathbf{A}_i] = -\mathcal{I}_{\mathbb{R}^{n \times n}}[\mathbf{A}_i]. \tag{3.39}$$

Let $\{\mathbf{C}_i\}_{i=1}^{N_R}$ be a set of N_R SPD matrices. The interpolation method developed in the previous section results in constructing a SPD matrix $\mathcal{I}_{\text{SPD}}[\mathbf{C}_i]$ interpolating the matrices $\{\mathbf{C}_i\}_{i=1}^{N_R}$ and satisfying

$$\mathcal{I}_{\text{SPD}}[\mathbf{C}_i] = \mathbf{C}_{i_0}^{1/2} \texttt{expm}\left(\mathcal{I}_{\mathbb{R}^{n \times n}}[\texttt{logm}(\mathbf{C}_{i_0}^{-1/2}\mathbf{C}_i\mathbf{C}_{i_0}^{-1/2})]\right)\mathbf{C}_{i_0}^{1/2}, \tag{3.40}$$

where `expm` and `logm` respectively denote the matrix exponential and logarithm and $i_0 \in 1, \cdots, N_R$.

Proposition 1. Let $\{\mathbf{C}_i\}_{i=1}^{N_R}$ be a set of N_R SPD matrices and $\{\mathbf{S}_i\}_{i=1}^{N_R}$ denote the set of their respective inverses, that is $\mathbf{S}_i = \mathbf{C}_i^{-1}$, $\forall i = 1, \cdots, N_R$. Then

$$\left(\mathcal{I}_{\text{SPD}}[\mathbf{C}_i]\right)^{-1} = \mathcal{I}_{\text{SPD}}[\mathbf{S}_i], \tag{3.41}$$

that is

$$\left(\mathcal{I}_{\text{SPD}}[\mathbf{C}_i]\right)^{-1} = \mathcal{I}_{\text{SPD}}[\mathbf{C}_i^{-1}]. \tag{3.42}$$

In other words, the inverse of the interpolation is the interpolation of the inverses.

Proof.

$$
\begin{aligned}
\left(\mathcal{I}_{\text{SPD}}[\mathbf{C}_i]\right)^{-1} &= \left(\mathbf{C}_{i_0}^{1/2}\texttt{expm}\left(\mathcal{I}_{\mathbb{R}^{n\times n}}[\texttt{logm}(\mathbf{C}_{i_0}^{-1/2}\mathbf{C}_i\mathbf{C}_{i_0}^{-1/2})]\right)\mathbf{C}_{i_0}^{1/2}\right)^{-1} \\
&= \mathbf{C}_{i_0}^{-1/2}\left(\texttt{expm}\left(\mathcal{I}_{\mathbb{R}^{n\times n}}[\texttt{logm}(\mathbf{C}_{i_0}^{-1/2}\mathbf{C_i}\mathbf{C}_{i_0}^{-1/2})]\right)\right)^{-1}\mathbf{C}_{i_0}^{-1/2} \\
&= \mathbf{C}_{i_0}^{-1/2}\texttt{expm}\left(-\mathcal{I}_{\mathbb{R}^{n\times n}}[\texttt{logm}(\mathbf{C}_{i_0}^{-1/2}\mathbf{C}_i\mathbf{C}_{i_0}^{-1/2})]\right)\mathbf{C}_{i_0}^{-1/2} \\
&= \mathbf{C}_{i_0}^{-1/2}\texttt{expm}\left(\mathcal{I}_{\mathbb{R}^{n\times n}}[-\texttt{logm}(\mathbf{C}_{i_0}^{-1/2}\mathbf{C}_i\mathbf{C}_{i_0}^{-1/2})]\right)\mathbf{C}_{i_0}^{-1/2} \\
&= \mathbf{C}_{i_0}^{-1/2}\texttt{expm}\left(\mathcal{I}_{\mathbb{R}^{n\times n}}\left[\texttt{logm}\left(\left(\mathbf{C}_{i_0}^{-1/2}\mathbf{C}_i\mathbf{C}_{i_0}^{-1/2}\right)^{-1}\right)\right]\right)\mathbf{C}_{i_0}^{-1/2} \\
&= \mathbf{C}_{i_0}^{-1/2}\texttt{expm}\left(\mathcal{I}_{\mathbb{R}^{n\times n}}\left[\texttt{logm}\left(\mathbf{C}_{i_0}^{1/2}\mathbf{C}_i^{-1}\mathbf{C}_{i_0}^{1/2}\right)\right]\right)\mathbf{C}_{i_0}^{-1/2} \\
&= \mathbf{S}_{i_0}^{1/2}\texttt{expm}\left(\mathcal{I}_{\mathbb{R}^{n\times n}}\left[\texttt{logm}\left(\mathbf{S}_{i_0}^{-1/2}\mathbf{S}_i\mathbf{S}_{i_0}^{-1/2}\right)\right]\right)\mathbf{S}_{i_0}^{1/2} \\
&= \mathcal{I}_{\text{SPD}}[\mathbf{S}_i].
\end{aligned}
\tag{3.43}
$$

$\square$

In Reference [75], elastic moduli of a material are interpolated across configurations corresponding to shape changes (size of a hole in the material). Instead of

interpolating individually each component of a tensor of elastic stiffnesses as in [75], an alternative method would be to consider a representation of the tensor as a symmetric positive definite matrix and to interpolate a set of such matrices on the SPD(6) manifold. Besides preserving the symmetric positive definiteness of the matrix, such a method presents an additional important advantage over the procedure proposed in [75]: considering the tensor of elastic stiffnesses or the tensor of elastic compliances for interpolation results in the same material elastic characteristics. This is a direct consequence of Proposition 1.

In the remainder of this section, it is assumed that the general interpolation method is equipped with a standard multi-variate interpolation algorithm that approximates constant functions exactly (for example, see [76]). With the notations defined above,

$$\mathcal{I}_{\mathbb{R}^{n \times n}}[\mathbf{A}] = \mathbf{A}. \tag{3.44}$$

Proposition 2. Let $M = \left(\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \cdots, \boldsymbol{\mu}^{(N_R)}\right)$ denote a set of N_R operating points and let $\{\mathbf{Q}_i = \mathbf{Q}\}_{i=1}^{N_R}$ denote a corresponding set of N_R identical symmetric positive definite matrices of dimension n. Given a new operating point $\boldsymbol{\mu}^{(N_R+1)} \notin M$, the interpolation method described in section 3.6.2 delivers a matrix $\mathbf{Q}_{N_R+1}$ which is identical to $\mathbf{Q}$ as well.

Proposition 3. Let $M = \left(\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \cdots, \boldsymbol{\mu}^{(N_R)}\right)$ denote a set of N_R operating points and let $\left\{\boldsymbol{\Omega}_{n,i}^2\right\}_{i=1}^{N_R}$ denote a corresponding set of N_R diagonal matrices with n

positive entries[5]

$$
\boldsymbol{\Omega}_{n,i}^2 = \begin{bmatrix} \omega_{1,i}^2 = \left(\omega_1\left(\boldsymbol{\mu}^{(i)}\right)\right)^2 & & & (0) \\ & \omega_{2,i}^2 = \left(\omega_2\left(\boldsymbol{\mu}^{(i)}\right)\right)^2 & & \\ & & \ddots & \\ (0) & & & \omega_{n,i}^2 = \left(\omega_n\left(\boldsymbol{\mu}^{(i)}\right)\right)^2 \end{bmatrix} . \quad (3.45)
$$

Given a new operating point $\boldsymbol{\mu}^{(N_R+1)} \notin M$, the interpolation method described in section 3.6.2 delivers a matrix $\boldsymbol{\Omega}_{n,N_R+1}^2$ which is a diagonal matrix with n positive entries. This means that the set of positive diagonal matrices is stable by interpolation.

Proof. Proofs are offered in Appendix C. $\qquad\qquad\square$

3.6.4 Application to Reduced-Order Models

An application of the interpolation on the tangent space to the SPD(n) manifold is presented in chapter 5. Structural reduced-order linear operators are indeed symmetric positive definite matrices, and as such the present method provides a rigorous method in order to preserve that property in the interpolation process.

---

[5]In the case of linear structural dynamics *modal* ROMs, the diagonal entries of $\boldsymbol{\Omega}_{n,i}^2$ are the squares of n natural circular frequencies. In this section, the notations for structural ROMs defined in chapter 5 are anticipated.

3.7 Application to the Grassmann Manifold

3.7.1 Motivation

The Grassmann manifold contains all the subspaces of a given dimension in $\mathbb{R}^n$. In chapter 2, it was shown that subspaces uniquely characterize a Petrov-Galerkin projection and thus a ROM. The Grassmann manifold is therefore the manifold of choice for reduced bases interpolation appearing in model reduction, as opposed to the Stiefel manifold. As such, this section is dedicated to the establishment of high-order multivariate interpolation techniques on the Grassmann manifold. In chapters 4 and 5 this algorithm will be used to interpolate fluid and structural reduced bases.

3.7.2 Geodesic Equation

Let $\mathcal{Y}_0 \in \mathcal{G}(k,n)$ be a subspace of dimension k in $\mathbb{R}^n$ and $\mathcal{V}_0 \in T_{\mathcal{Y}_0}\mathcal{G}(k,n)$ belonging to the tangent space to the Grassmann manifold $\mathcal{G}(k,n)$ at $\mathcal{Y}_0$. In this section, the equation describing the geodesic path $t \longmapsto \mathcal{Y}(t)$ on $\mathcal{G}(k,n)$ having $\mathcal{Y}_0$ as an initial position and $\mathcal{V}_0$ as an initial velocity, is recalled.

For that purpose, one considers $\mathbf{Y}_0 \in \mathcal{M}$ spanning the subspace $\mathcal{Y}_0$ and $\mathbf{\Gamma}_0 \in T_{\mathbf{Y}_0}\mathcal{M}$ representing $\mathcal{V}_0$. Then, it can be shown that the corresponding geodesic equation for $\mathcal{Y}(t) \in \mathcal{G}(k,n)$ written in terms of its representation $\mathbf{Y}(t)$ as a second-order matrix ordinary differential equation (ODE) is [66, 70]

$$\ddot{\mathbf{Y}} + \mathbf{Y}\left(\mathbf{Y}^T\mathbf{Y}\right)^{-1}\dot{\mathbf{Y}}^T\dot{\mathbf{Y}} = \mathbf{0}_{n,k}, \tag{3.46}$$

which simplifies when $\mathcal{M} = \mathcal{ST}(k, n)$ as

$$\ddot{\mathbf{Y}} + \mathbf{Y}\dot{\mathbf{Y}}^T\dot{\mathbf{Y}} = \mathbf{0}_{n,k}. \tag{3.47}$$

The general matrix ODE (3.46) associated with its initial conditions $\mathbf{Y}(0) = \mathbf{Y}_0$ and $\dot{\mathbf{Y}}(0)\left(\mathbf{Y}_0^T\mathbf{Y}_0\right)^{-\frac{1}{2}} = \mathbf{\Gamma}_0$ can be integrated as

$$\mathbf{Y}(t) = \left(\mathbf{Y}_0(\mathbf{Y}_0^T\mathbf{Y}_0)^{-\frac{1}{2}}\mathbf{V}\mathbf{cos}(\mathbf{\Sigma}t) + \mathbf{U}\mathbf{sin}(\mathbf{\Sigma}t)\right)\mathbf{V}^T\left(\mathbf{Y}_0^T\mathbf{Y}_0\right)^{\frac{1}{2}}, \tag{3.48}$$

where $\mathbf{\Gamma}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is a thin singular value decomposition (SVD) of $\mathbf{\Gamma}_0$ [70].
Since $\mathcal{Y}(t) = \mathrm{span}\{\mathbf{Y}(t)\}$, the geodesic on the Grassmann manifold has the following expression[6]:

$$\mathcal{Y}(t) = \gamma(t; \mathcal{Y}_0, \mathcal{V}_0) = \mathrm{span}\left\{\mathbf{Y}_0(\mathbf{Y}_0^T\mathbf{Y}_0)^{-\frac{1}{2}}\mathbf{V}\mathbf{cos}(\mathbf{\Sigma}t) + \mathbf{U}\mathbf{sin}(\mathbf{\Sigma}t)\right\}, \tag{3.49}$$

which simplifies further when $\mathcal{M} = \mathcal{ST}(k, n)$ as

$$\mathcal{Y}(t) = \gamma(t; \mathcal{Y}_0, \mathcal{V}_0) = \mathrm{span}\left\{\mathbf{Y}_0\mathbf{V}\mathbf{cos}(\mathbf{\Sigma}t) + \mathbf{U}\mathbf{sin}(\mathbf{\Sigma}t)\right\}. \tag{3.50}$$

---

[6]The identity span $\{\mathbf{XM}\} = \mathrm{span}\{\mathbf{X}\}$, with $\mathbf{M}$ a non-singular matrix, has been used.

3.7.3 Exponential and Logarithm Mappings for the Grassmann Manifold

Exponential Mapping

Using the same notations as in the previous subsection, an expression for the exponential mapping on the Grassmann manifold is

$$\text{Exp}_{\mathcal{Y}_0}(\mathcal{V}_0) = \gamma(1; \mathcal{Y}_0, \mathcal{V}_0) = \text{span}\left\{\mathbf{Y}_0(\mathbf{Y}_0^T\mathbf{Y}_0)^{-\frac{1}{2}}\mathbf{V}\cos(\mathbf{\Sigma}) + \mathbf{U}\sin(\mathbf{\Sigma})\right\} \quad (3.51)$$

with $\mathcal{Y}_0 = \text{span}\{\mathbf{Y}_0\}$ and $\mathbf{\Gamma}_0\left(\mathbf{Y}_0^T\mathbf{Y}_0\right)^{\frac{1}{2}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ representing $\mathcal{V}_0$.

Logarithm Mapping

Let $\mathcal{Y}, \mathcal{Z}$ be two subspaces belonging to $\mathcal{G}(k, n)$ and let $\mathbf{Y}, \mathbf{Z} \in \mathcal{ST}(k, n)$ be respective orthonormal bases for these subspaces.

Let $(\mathbf{I}_n - \mathbf{Y}\mathbf{Y}^T)\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$ be a thin SVD. Then $\text{Log}_{\mathcal{Y}}(\mathcal{Z})$ can be represented by an n-by-k real matrix $\mathbf{\Gamma} \in T_{\mathbf{Y}}\mathcal{ST}(k, n)$ defined as [79]

$$\mathbf{\Gamma} = \tilde{\mathbf{U}}\tan^{-1}(\tilde{\mathbf{\Sigma}})\tilde{\mathbf{V}}^T. \quad (3.52)$$

When $\mathcal{Y}$ and $\mathcal{Z}$ are represented by matrices $\mathbf{Y}$ and $\mathbf{Z}$ belonging to the non-compact Stiefel manifold, the formulae for the mapping are

$$(\mathbf{I}_n - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T)\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T \ \text{(thin SVD)} \quad (3.53)$$

$$\mathbf{\Gamma} = \tilde{\mathbf{U}}\tan^{-1}(\tilde{\mathbf{\Sigma}})\tilde{\mathbf{V}}^T. \quad (3.54)$$

These formulae are derived in Appendix D.

Remark 1. The logarithm mapping of $\mathcal{Z}$ can only be defined in a neighborhood of $\mathcal{Y}$. More specifically, the mapping exists if and only if $\mathbf{Y}^T\mathbf{Z}$ is invertible, that is $\mathcal{Z}$ does not contain any direction orthogonal to $\mathcal{Y}$.

Remark 2. The diagonal elements θ_i of the diagonal matrix $\mathbf{\Theta} = \mathbf{tan}^{-1}(\tilde{\mathbf{\Sigma}})$ are the *principal angles* between $\mathcal{Y}$ and $\mathcal{Z}$. When $\mathcal{M} = \mathcal{ST}(k,n)$, the columns of $(\mathbf{Y}\tilde{\mathbf{V}})$ and $(\mathbf{Y}\tilde{\mathbf{V}}\mathbf{cos}(\mathbf{\Theta}) + \tilde{\mathbf{U}}\mathbf{sin}(\mathbf{\Theta}))$ are the *principal vectors* corresponding to these principal angles [70]. The reader should note that these principal vectors naturally appear in the expression of the geodesic in Eq. (3.50) by considering $\mathcal{Y} = \gamma(0; \mathcal{Y}, \mathcal{V}_0)$ and $\mathcal{Z} = \gamma(1; \mathcal{Y}, \mathcal{V}_0)$ with $\mathcal{V}_0$ represented by $\mathbf{\Gamma}$ and therefore substituting $\mathbf{Y}$ for $\mathbf{Y}_0$ and $(\tilde{\mathbf{U}}, \mathbf{\Theta}, \tilde{\mathbf{V}})$ for $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V})$. It is subsequently shown in section 3.7.5 that these angles provide an intuitive interpretation for the following interpolation algorithm.

3.7.4   Compact Interpolation

Algorithm

The general interpolation algorithm for a quotient matrix manifold presented in section 3.5.2 can be applied to the Grassmann manifold $\mathcal{G}(k,n) = \mathcal{M}/\mathcal{N}$. Without any loss of generality, the case $\mathcal{M} = \mathcal{ST}(k,n)$ is considered.

The subspaces $\{\mathcal{Y}_i\}_{i=1}^{N_R} = \{\mathcal{Y}(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R} \in \mathcal{G}(k,n)^{N_R}$ are represented by their respective bases $\{\mathbf{Y}_i\}_{i=1}^{N_R} = \{\mathbf{Y}(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R} \in \mathcal{M}^{N_R}$ and $\mathcal{Y}_{N_R+1} = \mathcal{Y}(\boldsymbol{\mu}^{(N_R+1)})$ is sought.

Information about the sensitivities of the reduced bases can also be added in order to provide a higher-order compact interpolation scheme. As in section 3.5 , i_0 refers to the index of the reference element in the interpolation process. Let i be an integer in $\{1, \cdots, N_R\}$ different than i_0. Assuming the sensitivities $\{\nabla\mathbf{Y}_i\} =$

$\left\{ \dfrac{\partial \mathbf{Y}}{\partial \mu_1}(\boldsymbol{\mu}^{(i)}), \cdots, \dfrac{\partial \mathbf{Y}}{\partial \mu_{N_p}}(\boldsymbol{\mu}^{(i)}) \right\}$ are well-defined and known, a closed form for the gradient of the logarithm mapping $\{\nabla \boldsymbol{\Gamma}_i\}$ can be computed as follows:

Let μ_j, $j = 1, \cdots, N_p$ denote one of the parameters and consider the logarithm mapping $\mathrm{Log}_{\mathcal{Y}_{i_0}}(\mathcal{Y}_i) = \mathrm{span}\,(\boldsymbol{\Gamma}_i)$ that has already been computed using the following two equations

$$\mathbf{P}_i = \left(\mathbf{I}_n - \mathbf{Y}_{i_0}\mathbf{Y}_{i_0}^T\right)\mathbf{Y}_i\left(\mathbf{Y}_{i_0}^T\mathbf{Y}_i\right)^{-1} = \mathbf{U}_i\boldsymbol{\Sigma}_i\mathbf{V}_i^T \tag{3.55}$$

$$\boldsymbol{\Gamma}_i = \mathbf{U}_i\tan^{-1}\left(\boldsymbol{\Sigma}_i\right)\mathbf{V}_i^T, \tag{3.56}$$

1. Compute

$$\begin{aligned} \mathbf{M}_{i,j} = \frac{\partial \mathbf{P}_i}{\partial \mu_j} &= \left(\mathbf{I}_n - \mathbf{Y}_{i_0}\mathbf{Y}_{i_0}^T\right)\left(\frac{\partial \mathbf{Y}_i}{\partial \mu_j} - \mathbf{Y}_i\left(\mathbf{Y}_{i_0}^T\mathbf{Y}_i\right)^{-1}\mathbf{Y}_{i_0}^T\frac{\partial \mathbf{Y}_i}{\partial \mu_j}\right)\left(\mathbf{Y}_{i_0}^T\mathbf{Y}_i\right)^{-1} \\ &= \left(\mathbf{I}_n - \mathbf{Y}_{i_0}\mathbf{Y}_{i_0}^T\right)\left(\mathbf{I}_n - \mathbf{Y}_i\left(\mathbf{Y}_{i_0}^T\mathbf{Y}_i\right)^{-1}\mathbf{Y}_{i_0}^T\right)\frac{\partial \mathbf{Y}_i}{\partial \mu_j}\left(\mathbf{Y}_{i_0}^T\mathbf{Y}_i\right)^{-1} \\ &= \boldsymbol{\Pi}_{\mathbf{Y}_{i_0}^{\perp}, \mathbf{Y}_{i_0}^{\perp}}\left(\mathbf{I}_n - \boldsymbol{\Pi}_{\mathbf{Y}_i, \mathbf{Y}_{i_0}}\right)\frac{\partial \mathbf{Y}_i}{\partial \mu_j}\left(\mathbf{Y}_{i_0}^T\mathbf{Y}_i\right)^{-1}. \end{aligned}$$

$$\tag{3.57}$$

2. Compute $\dfrac{\partial \mathbf{U}_i}{\partial \mu_j}, \dfrac{\partial \boldsymbol{\Sigma}_i}{\partial \mu_j}$ and $\dfrac{\partial \mathbf{V}_i}{\partial \mu_j}$ using the algorithm described in Appendix A applied to the matrix $\mathbf{P}_i$.

3. Compute $\dfrac{\partial \boldsymbol{\Gamma}_i}{\partial \mu_j}$ as

$$\frac{\partial \boldsymbol{\Gamma}_i}{\partial \mu_j} = \frac{\partial \mathbf{U}_i}{\partial \mu_j}\tan^{-1}\left(\boldsymbol{\Sigma}_i\right)\mathbf{V}_i^T + \mathbf{U}_i\left(\mathbf{diag}\left(\frac{\frac{\partial \sigma_i}{\partial \mu_j}}{1 + \sigma_i^2}\right)\mathbf{V}_i^T + \tan^{-1}\left(\boldsymbol{\Sigma}_i\right)\frac{\partial \mathbf{V}_i^T}{\partial \mu_j}\right), \tag{3.58}$$

where $\boldsymbol{\Sigma}_i = \mathbf{diag}\,(\sigma_i)$.

As pointed in Appendix A, this approach is only applicable when the singular

values of the matrix $\mathbf{P}_i$ are distinct. When this is not the case, an approach based on finite differences shall be used to approximate $\dfrac{\partial \mathbf{\Gamma}_i}{\partial \mu_j}$. In particular, for $i = i_0$, $\mathbf{P}_{i_0} = \mathbf{0}_{n,k}$ and all its singular values are zero. In that case, for every parameter μ_j, $j = 1, \cdots, N_p$, a perturbed reduced basis $\mathbf{Y}_{i_0} + \nabla \mathbf{Y}_{i_0} \cdot \delta \boldsymbol{\mu}_j$ is computed, $\delta \boldsymbol{\mu}_j$ denoting an infinitesimal vector along the j-th direction in $\mathbb{R}^{N_p}$. Appendix E presents an algorithm for computing such a vector $\delta \boldsymbol{\mu}_j$ that preserves the full-rank property of the reduced basis. The algorithm for approximating $\dfrac{\partial \mathbf{\Gamma}_{i_0}}{\partial \mu_j}$ by finite differences is subsequently presented in Appendix F, as well as a proof showing that $\mathbf{\Gamma}(\boldsymbol{\mu})$ is differentiable at $\boldsymbol{\mu} = \boldsymbol{\mu}^{(i_0)}$, and an analysis showing that the quantity that is computed by the algorithm is a second-order approximation of $\dfrac{\partial \mathbf{\Gamma}_{i_0}}{\partial \mu_j}$.

Application to interpolation of elements of $\mathcal{G}(1,3)$

This special case corresponds to the interpolation of one dimensional linear spaces in $\mathbb{R}^3$. The bases $\{\mathbf{y}_i\}_{i=1}^{N_R} = \{\mathbf{y}(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R}$ representing these subspaces belong to $\mathcal{ST}(1,3)$ which is also the unit sphere $\mathbb{S}^3$ in $\mathbb{R}^3$. Hence, in the spherical set of coordinates, one can write

$$\mathbf{y}(\boldsymbol{\mu}) = \begin{bmatrix} \cos\phi(\boldsymbol{\mu})\sin\theta(\boldsymbol{\mu}) \\ \sin\phi(\boldsymbol{\mu})\sin\theta(\boldsymbol{\mu}) \\ \cos\theta(\boldsymbol{\mu}) \end{bmatrix}. \tag{3.59}$$

Following the interpolation procedure step-by-step, one obtains (see Appendix G)

$$\mathrm{Exp}_{\mathcal{Y}_{i_0}}(\mathcal{V}_{N_R+1}) = \mathrm{span}\left(\begin{bmatrix} \cos\phi_{N_R+1}\sin\theta_{N_R+1} \\ \sin\phi_{N_R+1}\sin\theta_{N_R+1} \\ \cos\theta_{N_R+1} \end{bmatrix} \right), \tag{3.60}$$

with $\phi_{N_R+1} \in [0, 2\pi]$, $\theta_{N_R+1} > 0$. The algorithm has proceeded by naturally interpolating the angles (in the spherical set of coordinates) between the pre-computed points $\{\mathcal{Y}(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R}$ and the reference pre-computed point $\mathcal{Y}(\boldsymbol{\mu}^{(i_0)})$.

Computational Complexity

In this section, the computational complexity of the method of interpolation in a tangent space to a Grassmann manifold is described. For simplicity, the non-compact case is considered.

- Step 1. $N_R - 1$ thin SVDs are performed for a total asymptotic computational cost equal to $(N_R - 1) \times \mathcal{O}(nk^2)$.

- Step 2. Interpolation in the tangent space is performed at an asymptotic computational cost equal to $N_R \times \mathcal{O}(nk)$ (Lagrange interpolation) or $N_R^2 \times \mathcal{O}(nk)$ (Radial basis function-based interpolation).

- Step 3. One thin SVD is performed at an asymptotic computational cost equal to $\mathcal{O}(nk^2)$.

Hence, the total asymptotic computational complexity of a univariate interpolation in a tangent space to a Grassmann manifold grows only as $N_R \times \mathcal{O}(nk^2)$. Therefore, this complexity is a linear function of the size of the underlying large dimension n, which suggests that the proposed interpolation method is computationally efficient.

3.7.5   Relationship to Principal Angles

The fundamental link between principal angles and geodesics has been established in Remark 2 in section 3.7.3. Björk and Golub provide in Ref. [80] another formula for

computing the principal angles and vectors based on the Singular Value Decomposition.

Letting $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathcal{ST}(k, n)$ be two matrices forming unitary bases of two subspaces $\mathcal{Y}_1, \mathcal{Y}_2 \in \mathcal{G}(k, n)$ respectively, the full SVD of the matrix $\mathbf{Y}_1^T \mathbf{Y}_2 \in \mathbb{R}^{k \times k}$ can be computed as

$$\mathbf{Y}_1^T \mathbf{Y}_2 = \mathbf{QCZ}^T. \tag{3.61}$$

The principal angles are then the arccosine of the diagonal elements of $\mathbf{C}$

$$\cos(\boldsymbol{\Theta}) = \mathbf{C} \tag{3.62}$$

and the principal vectors $\mathbf{U}_1, \mathbf{U}_2 \in \mathcal{ST}(k, n)$ for $\mathcal{Y}_1$ and $\mathcal{Y}_2$ respectively are

$$\mathbf{U}_1 = \mathbf{Y}_1 \mathbf{Q}, \quad \mathbf{U}_2 = \mathbf{Y}_2 \mathbf{Z}. \tag{3.63}$$

These important formulae have led to a large number of applications in domains such as numerical linear algebra [81], machine learning [82] and control [83]. Lieu and al. used the properties of the subspace principal angles and vectors to adapt two subspaces depending on a unique parameter $\boldsymbol{\mu} = \mu$ in References [17, 25, 26, 27, 28, 84] as follows.

Denoting these two subspaces by $\mathcal{Y}_1 = \mathcal{Y}(\mu^{(1)})$ and $\mathcal{Y}_2 = \mathcal{Y}(\mu^{(2)})$, the method constructs a new subspace $\mathcal{Y}_3$ defined at a new intermediary value of the parameter $\mu^{(3)} \in [\mu^{(1)}, \mu^{(2)}]$. Each principal vector in $\mathbf{U}_1$ is rotated towards its counterpart in $\mathbf{U}_2$ in order to obtain the interpolated corresponding principal vector. This algorithm corresponds to constructing an orthogonal basis $\mathbf{Y}_3$ corresponding to a subspace $\mathcal{Y}_3$ located on the geodesic $\{\bar{\gamma}(t; \mathcal{Y}_1, \mathcal{Y}_2), \ t \in [0, 1]\}$ on the Grassmann manifold $\mathcal{G}(k, n)$,

that is

$$\mathcal{Y}_3 = \bar{\gamma}\left(f\left(\mu^{(3)}\right); \mathcal{Y}_1, \mathcal{Y}_2\right). \tag{3.64}$$

with $f\left(\mu^{(3)}\right) \in [0,1]$ localizing the interpolated point on the geodesic. In the case of linear interpolation, $f\left(\mu^{(3)}\right) = \dfrac{\mu^{(3)} - \mu^{(1)}}{\mu^{(2)} - \mu^{(1)}}$ for instance.

Note that interpolation is here directly related to the homogeneity property of the geodesic (see Ref. [67]) as follows

$$\bar{\gamma}\left(f\left(\mu^{(3)}\right); \mathcal{Y}_1, \mathcal{Y}_2\right) = \gamma\left(f\left(\mu^{(3)}\right); \mathcal{Y}_1, \mathrm{Log}_{\mathcal{Y}_1}\left(\mathcal{Y}_2\right)\right) = \gamma\left(1; \mathcal{Y}_1, f\left(\mu^{(3)}\right)\mathrm{Log}_{\mathcal{Y}_1}\left(\mathcal{Y}_2\right)\right). \tag{3.65}$$

However, $\mathrm{Log}_{\mathcal{Y}_1}\left(\mathcal{Y}_1\right) \equiv 0$ and thus, in the linear interpolation case,

$$
\begin{aligned}
\mathcal{Y}_3 &= \gamma\left(1; \mathcal{Y}_1, \frac{\mu^{(3)} - \mu^{(1)}}{\mu^{(2)} - \mu^{(1)}}\mathrm{Log}_{\mathcal{Y}_1}\left(\mathcal{Y}_2\right)\right) && (3.66) \\
&= \gamma\left(1; \mathcal{Y}_1, \mathrm{Log}_{\mathcal{Y}_1}\left(\mathcal{Y}_1\right) + \frac{\mu^{(3)} - \mu^{(1)}}{\mu^{(2)} - \mu^{(1)}}\left(\mathrm{Log}_{\mathcal{Y}_1}\left(\mathcal{Y}_2\right) - \mathrm{Log}_{\mathcal{Y}_1}\left(\mathcal{Y}_1\right)\right)\right). && (3.67)
\end{aligned}
$$

This result shows that the interpolated quantity of interest is the logarithm mapping and therefore, the subspace angle interpolation method is exactly the restriction of the method proposed in section 3.7.4 to the case of one continuously varying parameter μ and two subspaces $\mathcal{Y}_1$ and $\mathcal{Y}_2$.

Furthermore, this property can provide another algorithm to compute the logarithm mapping as follows: let $\mathcal{X}, \mathcal{Y} \in \mathcal{G}(k,n)$ be two subspaces respectively spanned by two bases $\mathbf{X}, \mathbf{Y} \in \mathcal{ST}(k,n)$. Computing the SVD of $\mathbf{X}^T\mathbf{Y}$ as

$$\mathbf{X}^T\mathbf{Y} = \mathbf{QCZ}^T, \tag{3.68}$$

then defining the principal angles as $\mathbf{\Theta} = \mathbf{cos}^{-1}(\mathbf{C})$ and their sines as $\mathbf{S} = \mathbf{sin}(\mathbf{\Theta})$,

let's define $\hat{\mathbf{V}} = \mathbf{Q}$ and $\hat{\mathbf{U}} = (\mathbf{YZ} - \mathbf{XQC})\mathbf{S}^{-1}$. The logarithm mapping $\mathrm{Log}_{\mathcal{X}}(\mathcal{Y})$ is then represented by the matrix

$$\mathbf{\Gamma} = \hat{\mathbf{U}}\mathbf{\Theta}\hat{\mathbf{V}}^{T} = (\mathbf{YZ} - \mathbf{XQC})\mathbf{S}^{-1}\mathbf{\Theta}\mathbf{Q}^{T}. \tag{3.69}$$

However, as stated by Björck and Golub [80], the arccosine operation may produce losses of accuracy when the principal angles are small. In Ref. [85], a numerically accurate method that choses the arccosine formula for large angles and the arcsine formula for small angles is presented to remediate that issue. Similarly, in our case, the formula based on the arctangent operation would be particularly suitable when used for computing small principal angles.

3.7.6   Application to Reduced-Order Bases

The Petrov-Galerkin projection detailed in section 2.2 can be used as a framework for several methods in model reduction and in control, such as the Proper-Orthogonal Decomposition (POD) (section 2.5.2) and the balanced truncation methods (section 2.5.1). However, generally, repeatedly computing the reduced bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$ for each value of the parameter $\boldsymbol{\mu}$ is a computationally intensive process. The subspace interpolation method derived in section 3.7.4 can therefore be very useful in this context as it enables independently interpolating the two sets of subspaces represented by $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$ for new values of $\boldsymbol{\mu}$, at a much cheaper cost [88, 89].

Problem Formulation

In the remainder of this section, the reduced subspaces defining a Petrov-Galerkin projection are abstracted as a doublet of subspaces

$$\mathcal{S}(\boldsymbol{\mu}) = (\mathcal{V}(\boldsymbol{\mu}), \mathcal{W}(\boldsymbol{\mu})), \tag{3.70}$$

represented by their respective bases

$$\mathcal{B}(\boldsymbol{\mu}) = (\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})), \tag{3.71}$$

satisfying the orthogonality constraint

$$\mathbf{W}^T(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu}) = \mathbf{I}_k. \tag{3.72}$$

The following problem can be formulated.

Problem. Let $\{\mathcal{S}_j\}_{j=1}^{N_R} = \{\mathcal{S}(\boldsymbol{\mu}^{(j)})\}_{j=1}^{N_R}$ denote a set of N_R doublets of subspaces of the same dimension k constructed at the values of the parameters $\{\boldsymbol{\mu}^{(j)}\}_{j=1}^{N_R}$, and represented by their respective bases $\{\mathcal{B}_j\}_{j=1}^{N_R} = \{\mathcal{B}(\boldsymbol{\mu}^{(j)})\}_{j=1}^{N_R}$. Given new values of the parameters $\boldsymbol{\mu}^{(N_R+1)} \neq \boldsymbol{\mu}^{(j)}, \forall j = 1, \cdots, N_R$, compute $\mathcal{S}_{N_R+1} = \mathcal{S}(\boldsymbol{\mu}^{(N_R+1)})$ represented by $\mathcal{B}_{N_R+1} = \mathcal{B}(\boldsymbol{\mu}^{(N_R+1)})$.

Interpolation on the Grassmann Manifold

Two cases can be distinguished: the first one is the particular Galerkin projection case where $\mathcal{V}(\boldsymbol{\mu}) = \mathcal{W}(\boldsymbol{\mu})$ and $\mathbf{V}(\boldsymbol{\mu}) = \mathbf{W}(\boldsymbol{\mu})$. The second case is the more general case where these subspaces are distinct.

- Galerkin projection

  The algorithm for interpolation on quotient matrix manifold in section 3.5.2 applied to the Grassmann manifold seen as $\mathcal{G}(k, n) = \mathcal{ST}(k, n)/\mathrm{O}(k)$ is considered for computing $\mathcal{V}(\boldsymbol{\mu})$ at the new value $\boldsymbol{\mu}^{(N_R+1)}$. The orthogonality constraint (3.72) is here enforced by the choice of the manifold representation.

- General Petrov-Galerkin projection

  The algorithm proceeds in three steps as follows.

  - Step 1. Apply the algorithm presented in section 3.5.2 to the Grassmann manifold seen as $\mathcal{G}(k, n) = \mathbb{R}_*^{n \times k}/\mathrm{GL}(k)$ and interpolate $\{\mathcal{V}(\boldsymbol{\mu}^{(j)})\}_{j=1}^{N_R}$.

  - Step 2. Apply the algorithm considered in Step 1 to independently interpolate $\{\mathcal{W}(\boldsymbol{\mu}^{(j)})\}_{j=1}^{N_R}$ at the new value of the parameter $\boldsymbol{\mu}^{(N_R+1)}$.

  - Step 3. Enforce, the bi-orthogonality constraint (3.72) for the two interpolated bases. This can be achieved by a Gram-Schmidt-like procedure.

For a general Petrov-Galerkin projection method, the two bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$ are not bi-orthogonal and, as such, Step 3 of the algorithm is optional. For the BPOD and balanced truncation methods, the bi-orthogonality property is necessary.

POD Basis Interpolation and the Global POD Methods

There is a relation between the POD bases produced by the Grassmann interpolation and the GPOD methods (see section 2.8.3). For the sake of simplicity, it is assumed in the rest of this analysis that the POD bases that are built are not truncated, that is

$$\mathrm{span}\left\{\mathbf{V}(\boldsymbol{\mu}^{(i)})\right\} = \mathrm{span}\left\{\mathbf{X}(\boldsymbol{\mu}^{(i)})\right\}, \ i = 1, \cdots, N_R + 1. \tag{3.73}$$

In that case, the GPOD approach is equivalent to computing an orthogonal basis $\mathbf{V}^G = \mathbf{V}(\boldsymbol{\mu}^{(N_R+1)})$ satisfying

$$\text{span}\left\{\mathbf{V}(\boldsymbol{\mu}^{(N_R+1)})\right\} = \text{span}\left\{\left[\mathbf{V}(\boldsymbol{\mu}^{(1)}), \cdots, \mathbf{V}(\boldsymbol{\mu}^{(N_R)})\right]\right\}. \tag{3.74}$$

Property. Let $\mathbf{V}^I$ denote the POD basis solution of the Problem formulated above obtained by the Grassmann manifold interpolation method and $\mathbf{V}^G$ its counterpart produced by the GPOD approach. Then, span $\left\{\mathbf{V}^I\right\}$ is a subspace of span $\left\{\mathbf{V}^G\right\}$.

Proof. Since

$$\text{span}\left\{\mathbf{V}^G\right\} = \text{span}\left\{\left[\mathbf{V}(\boldsymbol{\mu}^{(1)}), \cdots, \mathbf{V}(\boldsymbol{\mu}^{(N_R)})\right]\right\}, \tag{3.75}$$

it is sufficient to prove that $\mathbf{V}^I$ can be written as a linear combination of the column vectors of $\left\{\mathbf{V}_i = \mathbf{V}(\boldsymbol{\mu}^{(i)})\right\}_{i=1}^{N_R}$.

Following the steps of the Algorithm in section 3.5.2:

- Step 0. $\mathbf{V}_{i_0} = \mathbf{V}(\boldsymbol{\mu}^{(i_0)})$ is chosen.

- Step 1. $\forall i = 1, \cdots, N_R$,

$$(\mathbf{I}_n - \mathbf{V}_{i_0}\mathbf{V}_{i_0}^T)\mathbf{V}_i(\mathbf{V}_{i_0}^T\mathbf{V}_i)^{-1} = \mathbf{U}_i\boldsymbol{\Sigma}_i\mathbf{W}_i^T \quad \text{(Thin SVD)} \tag{3.76}$$

$$\boldsymbol{\Gamma}_i = \mathbf{U}_i\mathbf{tan}^{-1}(\boldsymbol{\Sigma}_i)\mathbf{W}_i^T. \tag{3.77}$$

Hence

$$\text{span}\left\{\boldsymbol{\Gamma}_i\right\} \subseteq \text{span}\left\{\mathbf{U}_i\right\} \subseteq \text{span}\left\{\mathbf{V}_{i_0}, \mathbf{V}_i\right\} \subseteq \text{span}\left\{\mathbf{V}^G\right\}. \tag{3.78}$$

- Step 2. $\boldsymbol{\Gamma}_{N_R+1}$ is obtained by direct interpolation of the entries of $\left\{\boldsymbol{\Gamma}_i\right\}_{i=1}^{N_R}$.

Most interpolation methods proceeds by linear combinations of the precomputed entries, the weights of these combinations being independent of the values of the functions of interest. Therefore, one can write

$$\boldsymbol{\Gamma}_{N_R+1} = \sum_{i=1}^{N_R} \alpha_i \boldsymbol{\Gamma}_i \tag{3.79}$$

where $\alpha_i, \; i = 1, \cdots, N_R$ are the weights and, as such,

$$\text{span}\{\boldsymbol{\Gamma}_{N_R+1}\} \subseteq \text{span}\left\{\boldsymbol{\Gamma}_1, \cdots, \boldsymbol{\Gamma}_{N_R}\right\} \subseteq \text{span}\left\{\mathbf{V}^G\right\}. \tag{3.80}$$

- Step 3.

$$\boldsymbol{\Gamma}_{N_R+1} = \mathbf{U}_{N_R+1}\boldsymbol{\Sigma}_{N_R+1}\mathbf{W}_{N_R+1}^T \quad \text{(thin SVD)} \tag{3.81}$$

$$\mathbf{V}^I = \mathbf{V}_{i_0}\mathbf{W}_{N_R+1}\cos(\boldsymbol{\Sigma}_{N_R+1}) + \mathbf{U}_{N_R+1}\sin(\boldsymbol{\Sigma}_{N_R+1}), \tag{3.82}$$

therefore $\text{span}\left\{\mathbf{U}_{N_R+1}\right\} = \text{span}\left\{\boldsymbol{\Gamma}_{N_R+1}\right\} \subseteq \text{span}\left\{\mathbf{V}^G\right\}$ and

$$\text{span}\left\{\mathbf{V}^I\right\} \subseteq \text{span}\left\{\mathbf{V}_{i_0}, \mathbf{U}_{N_R+1}\right\} \subseteq \text{span}\left\{\mathbf{V}^G\right\}, \tag{3.83}$$

concluding the proof. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

This shows that in the cases where there is a property guarantying a decrease in the model reduction error $\mathcal{E}_{\text{ROM}}$ (defined in section 2.2.2) when vectors are added to the POD basis, the global POD method will be more accurate than the interpolation POD basis method. This property, however, rarely holds as illustrated by the model reduction of the F-16 and F-18/A in sections 4.5.1 and 4.6.1. Furthermore, the GPOD method typically leads to larger ROMs, which defeats the purpose of model reduction.

Hence, in this thesis, Grassmann manifold-based interpolation techniques will be the methods of choice for reduced-basis adaptation.

Application to the Mass-Spring-Damper System

System 1 defined in section 2.7.1 is considered here. The same snapshot sampling as the one presented in section 2.7.2 is followed for every value of the vector of parameters $\boldsymbol{\mu} = (m, c, k)$. Ten vectors are retained in the POD basis.

For this simple example, compact and non-compact interpolation methods on the tangent space to the Grassmann manifold are compared. Hence, the sensitivities of the POD basis $\mathbf{V}(\boldsymbol{\mu})$ with respect to each of the parameters $\mu_1 = m$, $\mu_2 = c$ and $\mu_3 = k$ are first computed. For that purpose, a simple inspection of the matrices $\hat{\mathbf{A}}(\boldsymbol{\mu})$ and $\hat{\mathbf{B}}(\boldsymbol{\mu})$ indicates that their dependency on the parameters m, c and k is as follows:

$$\hat{\mathbf{A}}(\boldsymbol{\mu}) = \hat{\mathbf{A}}(m, c, k), \quad \hat{\mathbf{B}}(\boldsymbol{\mu}) = \hat{\mathbf{B}}(m). \tag{3.84}$$

Therefore, the sensitivities of these two matrices can be computed analytically as:

$$\nabla \hat{\mathbf{A}}(\boldsymbol{\mu}) = \left[\frac{\partial \hat{\mathbf{A}}}{\partial m}(\boldsymbol{\mu}), \frac{\partial \hat{\mathbf{A}}}{\partial c}(\boldsymbol{\mu}), \frac{\partial \hat{\mathbf{A}}}{\partial k}(\boldsymbol{\mu}) \right], \quad \nabla \hat{\mathbf{B}}(\boldsymbol{\mu}) = \left[\frac{\partial \hat{\mathbf{B}}}{\partial m}(\boldsymbol{\mu}), \mathbf{0}_{2n}, \mathbf{0}_{2n} \right]. \tag{3.85}$$

Algorithm 1 developed in section 2.8.1 is used to compute these sensitivities.

The two subspace interpolation methods (compact and non-compact) are now compared at a fixed number of pre-computed information. A POD basis is here sought for $\boldsymbol{\mu}^{(10)} = (m^{(10)}, c^{(10)}, k^{(10)}) = (0.8, 0.6, 0.7)$. Three different bases at this point $\boldsymbol{\mu}^{(10)}$ are then computed using: (1) the method of snapshots presented in section 2.7.2. This basis will be referred to as the "direct POD basis". Two pre-computed databases are then considered. (2) A first database containing nine pre-computed POD bases

for other values of the parameter $\boldsymbol{\mu}^{(j)}$, $j = 1, \cdots, 9$. In this first database, the only information that is available is the POD basis $\mathbf{V}(\boldsymbol{\mu}^{(j)})$ at each point. A new POD basis is then constructed at $\boldsymbol{\mu}^{(10)}$ using the non-compact interpolation method presented in section 3.7.4. Hardy's bivariate spline method presented in Appendix H is used for the interpolation in the tangent space to the manifold. (3) A second database containing the POD bases as well as their sensitivities with respect to the parameters (m, c, k) is considered for three pre-computed points $\boldsymbol{\mu}^{(j)}$, $j = 2, 5, 8$. The new POD basis at $\boldsymbol{\mu}^{(10)}$ is then constructed using the compact subspace interpolation algorithm presented in section 3.7.4. Hardy's compact interpolation method presented in Appendix H is again chosen for the interpolation in the tangent space.

The parameter values (m, c, k) for the pre-computed elements of the first database are reported in Table 3.7.6. The distance $d^{(j)}$ of each pre-computed point $\boldsymbol{\mu}^{(j)}$ to the target point $\boldsymbol{\mu}^{(10)}$ in the parameter space (m, c, k) is also shown. Similarly, the parameter values for the pre-computed points in the second database as well as their distance to the target point are reported in Table 3.7.6. Since the parameter c is chosen to be constant, the sensitivities of the POD bases with respect to the parameters that are needed in the compact interpolation process are the one with respect to the mass m and stiffness coefficient k. As such, for each point $\boldsymbol{\mu}^{(j)}$of the second database, three quantities are computed, namely $\left\{ \mathbf{V}(\boldsymbol{\mu}^{(j)}), \dfrac{\partial \mathbf{V}}{\partial m}(\boldsymbol{\mu}^{(j)}), \dfrac{\partial \mathbf{V}}{\partial k}(\boldsymbol{\mu}^{(j)}) \right\}$. By inspection of the distances of the pre-computed points to the target point in both databases, the reader can notice that, in the two numerical experiments, for a given distance d in the parameter space, the number of pre-computed quantities is the same.

The eigenvalues of the full-order operator $\hat{\mathbf{A}}(\boldsymbol{\mu}^{(10)})$ and the operators for reduced operators $\hat{\mathbf{A}}_r(\boldsymbol{\mu}^{(10)})$ computed using the three POD bases are first compared (criterion C1 in section 2.6) and reported in Fig. 3.4. It can be noted that the eigenvalues of the

	m	c	k	d
$\boldsymbol{\mu}^{(1)}$	0.4	0.6	0.8	0.41
$\boldsymbol{\mu}^{(2)}$	0.5	0.6	1.2	0.58
$\boldsymbol{\mu}^{(3)}$	0.5	0.6	0.2	0.58
$\boldsymbol{\mu}^{(4)}$	0.6	0.6	0.6	0.22
$\boldsymbol{\mu}^{(5)}$	0.7	0.6	0.5	0.22
$\boldsymbol{\mu}^{(6)}$	0.9	0.6	1.1	0.41
$\boldsymbol{\mu}^{(7)}$	1.0	0.6	0.8	0.22
$\boldsymbol{\mu}^{(8)}$	1.2	0.6	0.6	0.41
$\boldsymbol{\mu}^{(9)}$	1.3	0.6	0.4	0.58
$\boldsymbol{\mu}^{(10)}$	0.8	0.6	0.7	0.0

Table 3.5: Operating points of the mass-damper-spring system for the non-compact POD basis interpolation.

	m	c	k	d
$\boldsymbol{\mu}^{(2)}$	0.5	0.6	1.2	0.58
$\boldsymbol{\mu}^{(5)}$	0.7	0.6	0.5	0.22
$\boldsymbol{\mu}^{(8)}$	1.2	0.6	0.6	0.41
$\boldsymbol{\mu}^{(10)}$	0.8	0.6	0.7	0.0

Table 3.6: Operating points of the mass-damper-spring system for the compact POD basis interpolation.

	$\mathcal{H}_\infty$-norm	$\mathcal{H}_2$-norm
Directly constructed ROM (10)	5.7×10^{-4}	4.9×10^{-4}
Interpolated ROM (10)	4.9×10^{-4}	4.6×10^{-4}
Compact-interpolated ROM (10)	4.6×10^{-4}	4.3×10^{-4}

Table 3.7: Relative error norms for the reduced systems.

reduced operators closely match the eigenvalues of the full operator having largest real part. Furthermore, the eigenvalues for the reduced operator built using the POD basis constructed by compact interpolation match more closely the eigenvalues of the reduced operator built using the direct POD basis. Criterion C2 is then considered. First, the Bode diagram of the four systems are computed and reported in Fig. 3.5. Very good agreements are apparent, except for high frequencies where the reduced system built using the compact-interpolated POD basis is the only reduced system to closely match the full one. This result is also noticeable on the Bode diagram of the error systems that is reported in Fig. 3.6. The comparison of the Nyquist diagrams is shown in Fig. 3.7 as well as a zoom on the region of the complex plane close to the origin in Fig 3.8. Very good agreements appear. Finally the relative errors between the full-order and reduced systems are investigated and reported in Table 3.7.6 (criterion C3). An improvement can be observed for both the $\mathcal{H}_\infty$- and $\mathcal{H}_2$-norm errors when the sensitivities of the POD bases are used in the interpolation process, in agreement with the results relative to the Bode diagrams.
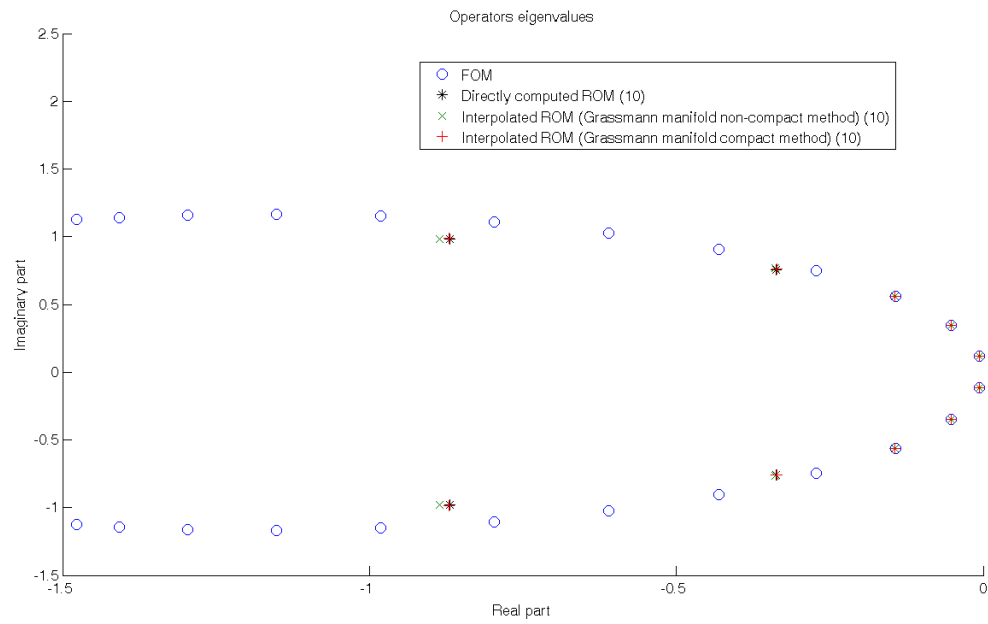
Figure 3.4: Eigenvalue comparison for the full-order and reduced systems
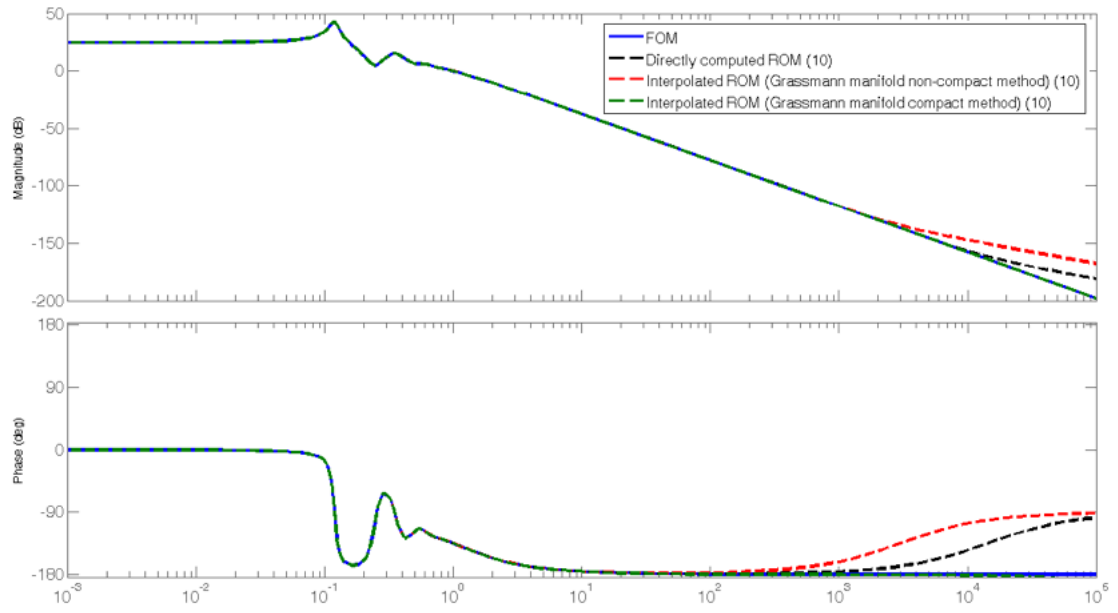
Figure 3.5: Bode diagrams comparison for the full-order and reduced systems
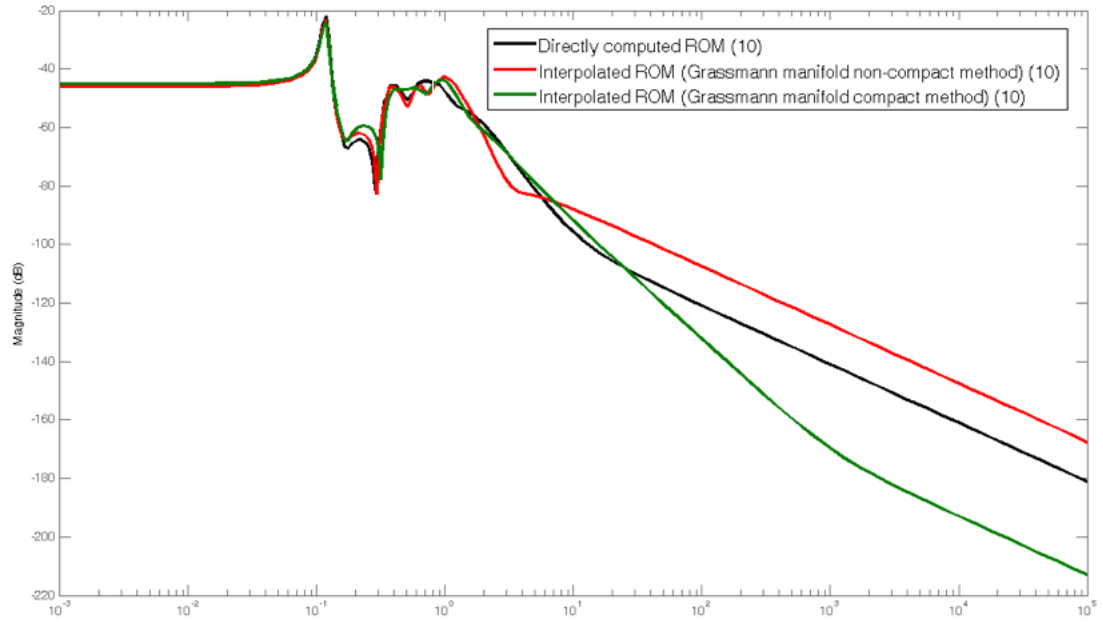
Figure 3.6: Error system frequency response comparison for the reduced systems
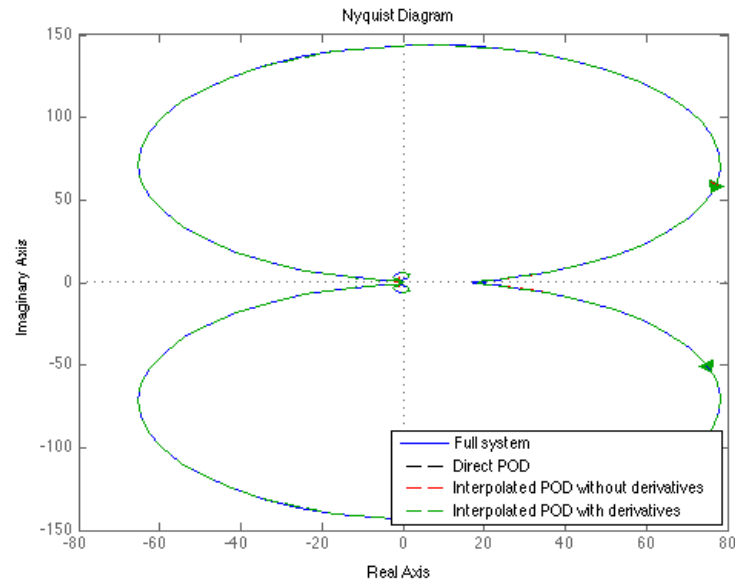
Figure 3.7: Nyquist diagram comparison for the full-order and reduced systems
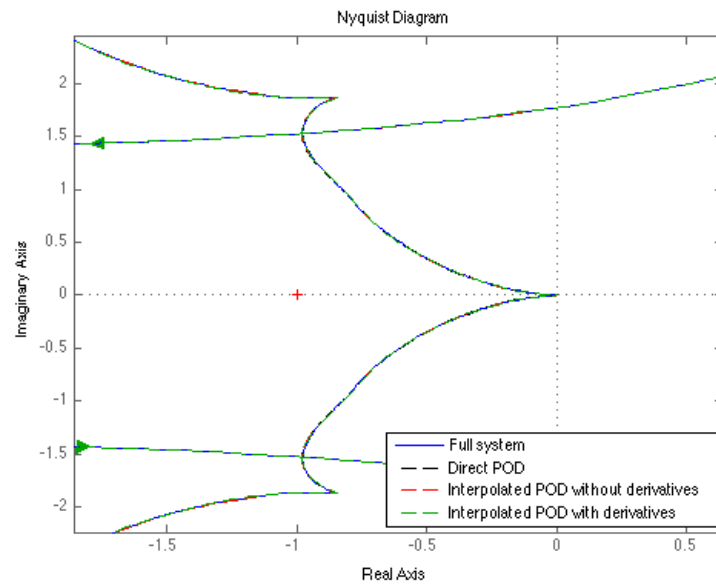
Figure 3.8: Zoom of the Nyquist diagram comparison for the full-order and reduced systems

Chapter 4

Interpolation of Fluid Reduced-Order Bases for Aeroelastic Problems

4.1 Introduction

In this thesis, model reduction of an aeroelastic system for flutter prediction is performed separately for each subsystem.

- The CFD-based linearized fluid subsystem is reduced by Petrov-Galerkin projection onto a reduced basis. By reducing the number of degrees of freedom involved in the analysis, the linear ODEs are of smaller size and can hence be integrated in real-time at a much reduced cost. Model reduction for the fluid subsystem of an aeroelastic system has been developed in the past using modal truncation [90], the Proper Orthogonal Decomposition method [8, 10, 16, 27]

and Volterra series [19]. The Proper Orthogonal Decomposition method is chosen here. The fluid ROM can be built in such a way that its elements only depend, before dimensionalization, on the Mach number of the freestream condition M_∞ and the corresponding angle of attack α (see section 4.4.5). As such, a fluid ROM can be exploited at a variety of altitudes (dynamic pressure and density) without any additional cost, drastically reducing the number of ROMs to be generated when studying the aeroelastic behavior of a given aircraft: only a two-dimensional space needs to be explored instead of a three-dimensional one.

- The Finite Element-based structural subsystem is reduced by modal truncation: the dry eigenmodes of the structure are computed and a few of these modes are retained to represent the structural subsystem. In this chapter, a unique structural configuration is chosen for each aircraft. Since the dry modes are independent of the flight configuration the aircraft operates at, a unique structural ROM has to be computed. In chapter 5, different structural configurations are explored for a given structural system, and, as such, algorithms for interpolation of reduced-order information will be developed.

The corresponding coupled reduced-order models then lead to real-time analysis of the aeroelastic system in one of the two following ways: (a) time integration of the reduced ODE which is fast and can be executed on a single processor system, as memory and CPU requirements are low, (b) stability analysis of the linear ROM operator which is computationally feasible through the computation of the entire eigenspectrum of a small-size matrix.

As mentioned in the introduction to this thesis, this ability to predict flutter in real-time is crucial for the flight-test community. However, the lack of robustness

of fluid reduced-order bases with respect to the flight conditions, as illustrated in section 4.5.2 for a full configuration F-16 aircraft, calls for a reconstruction of a reduced-order basis every time a new flight condition (M_∞, α) is considered. The high cost associated with such a reconstruction is highlighted in section 4.5.1. In this case, this cost is even greater than the cost associated with integrating in time the associated linear full-order model.

The availability of a database of reduced-order information can provide a strategy for rapidly constructing a new ROM. As mentioned in the introduction, interpolation of the fluid reduced-order bases is an alternative to an expensive reconstruction, provided that the interpolation is computationally inexpensive. This leads to the problem formulated in section 4.2. The interpolation-based strategy is then presented in section 4.3. For completeness purposes, the procedure leading to a CFD-based fluid ROM and FEM-based structural ROM is subsequently recalled in section 4.4. Applications to the aeroelastic analysis of two full aircraft configurations are then presented in sections 4.5 and 4.6 respectively, establishing the efficiency and accuracy of the proposed interpolation-based strategy.

4.2   Problem Formulation

Problem. Let $N_R > 1$ aeroelastic ROMs of the same dimension and their corresponding reduced-order bases constructed at N_R operating points $\boldsymbol{\mu}^{(j)} = (M_{\infty j}, \alpha_j)$, with $\boldsymbol{\mu}^{(j)} \neq \boldsymbol{\mu}^{(i)}$ for $j \neq i$, be given, and let S denote the set of respective operating points of these ROMs

$$S = \{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \cdots, \boldsymbol{\mu}^{(N_R)}\}. \tag{4.1}$$

Interpolate the given reduced-order fluid bases in order to rapidly construct a new ROM that can reliably operate at $\boldsymbol{\mu}^{(N_R+1)} \notin S$.

4.3 Reduced-Order Fluid Bases Adaptation

The interpolation algorithm presented in section 3.7.4 is applied here in its non-compact form to interpolate independently the reduced-order bases at the new operating point $\boldsymbol{\mu}^{(N_R+1)}$.

After reduced-order bases are constructed by interpolation as described in section 3.7.4, an adapted ROM can be built using the same procedure that was used in the first place for constructing the pre-computed ROMs from the pre-computed reduced-order bases. Given that, in most model reduction methods based on reduced-order bases, the phase of construction of the reduced-order basis dominates the total CPU time and is intrinsically expensive, and given the computational complexity results of section 3.7.4, the ROM adaptation method proposed in this section is a significantly more economical alternative to direct ROM reconstruction.

4.4 Model Reduction for Aeroelastic Problems

4.4.1 Arbitrary Lagrangian-Eulerian Formulation and Linearization

A high-fidelity, CFD-based, non-linear aeroelastic system can be represented by a three-field Arbitrary Lagrangian-Eulerian (ALE) formulation [91], which after semi-discretization in space leads to a set of three coupled matrix ODEs as

$$(\mathbf{A}(\mathbf{x})\mathbf{w})_{,t} + \mathbf{F}(\mathbf{w}, \mathbf{x}, \dot{\mathbf{x}}) = \mathbf{0}_{N_f} \tag{4.2}$$

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{f}^{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}) = \mathbf{f}^{\text{ext}}(\mathbf{u}, \mathbf{w}) \tag{4.3}$$

$$\tilde{\mathbf{K}}\mathbf{x} = \tilde{\mathbf{K}}_c\mathbf{u} \ . \tag{4.4}$$

Equation (4.2) corresponds generally to a finite volume discretization of the fluid equation in the ALE formulation, Eq. (4.3) to a finite element (FE) discretization of the structural equation and Eq. (4.4) models the deformation of the boundary of the fluid domain accordingly to the deformations of the structure: the fluid mesh is then modeled as a pseudo-structure with piecewise static behavior. In this work, all flows are assumed to be inviscid and therefore Eq. (4.2) is a discretization of the Euler equations.

$\mathbf{A}$ is a diagonal matrix containing the cell volumes, $\mathbf{F}$ denotes the numerical flux function and $\mathbf{w}$ the conservative state vector of the fluid subsystem. N_f denotes the dimension of the high-fidelity fluid subsystem model. $\mathbf{M}$ denotes the mass matrix resulting from the FE formulation, $\mathbf{f}^{\text{int}}$ the vector of internal forces and $\mathbf{f}^{\text{ext}}$ the vector of external forces applied on the structural subsystem. $\mathbf{u}$ denotes the vector of structural displacements. The boundary conditions associated with the subsystems are

not explicitly stated here for simplicity purposes. The fluid mesh motion is denoted
by $\mathbf{x}$, the fictitious stiffness matrix for the fluid mesh by $\tilde{\mathbf{K}}$ and the transfer matrix
describing the effect of the structural motion on the fluid mesh motion by $\tilde{\mathbf{K}}_c$. $(\dot{\ })$
and $(\ )_{,t}$ denote derivatives with respect to the dimensional time t.

This system of equations can be linearized around an equilibrium point denoted
by a subscript "o" and characterized by $\dot{\mathbf{w}}_o = \dot{\mathbf{x}}_o = \mathbf{0}$ [92, 93]. The perturbation of
the state vectors around that equilibrium point can be written as

$$\mathbf{w} = \mathbf{w}_o + \delta\mathbf{w}, \quad \dot{\mathbf{w}} = \dot{\mathbf{w}}_o + \delta\dot{\mathbf{w}}, \quad \mathbf{x} = \mathbf{x}_o + \delta\mathbf{x}, \quad \dot{\mathbf{x}} = \dot{\mathbf{x}}_o + \delta\dot{\mathbf{x}}. \tag{4.5}$$

The fluid equation is also non-dimensionalized [27, 84] to obtain the following form
of the semi-discrete system of equations governing its fluid subsystem (see, for exam-
ple [25, 26, 27, 28, 92])

$$\bar{\mathbf{A}}_o(\delta\bar{\mathbf{w}})_{,\tau} + \bar{\mathbf{H}}_o\delta\bar{\mathbf{w}} + (\bar{\mathbf{E}}_o + \bar{\mathbf{C}}_o)\delta\dot{\bar{\mathbf{x}}} + \bar{\mathbf{G}}_o\delta\bar{\mathbf{x}} = \mathbf{0}_{N_f}. \tag{4.6}$$

Here, the bar notation designates a non-dimensional quantity, $(\ )_{\tau}$ denotes the deriva-
tive with respect to the non-dimensional time τ and

$$\bar{\mathbf{A}}_o = \bar{\mathbf{A}}(\bar{\mathbf{x}}_o), \tag{4.7}$$

$$\bar{\mathbf{H}}_o = \frac{\partial\bar{\mathbf{F}}}{\partial\bar{\mathbf{w}}}(\bar{\mathbf{w}}_o, \bar{\mathbf{x}}_o, \dot{\bar{\mathbf{x}}}_o), \tag{4.8}$$

$$\bar{\mathbf{E}}_o = \frac{\partial\bar{\mathbf{A}}}{\partial\bar{\mathbf{x}}}(\bar{\mathbf{x}}_o)\bar{\mathbf{w}}_o, \tag{4.9}$$

$$\bar{\mathbf{C}}_o = \frac{\partial\bar{\mathbf{F}}}{\partial\dot{\bar{\mathbf{x}}}}(\bar{\mathbf{w}}_o, \bar{\mathbf{x}}_o, \dot{\bar{\mathbf{x}}}_o), \tag{4.10}$$

$$\bar{\mathbf{G}}_o = \frac{\partial\bar{\mathbf{F}}}{\partial\bar{\mathbf{x}}}(\bar{\mathbf{w}}_o, \bar{\mathbf{x}}_o, \dot{\bar{\mathbf{x}}}_o). \tag{4.11}$$

The matrices $\bar{\mathbf{H}}_o$, $\bar{\mathbf{E}}_o$, $\bar{\mathbf{C}}_o$ and $\bar{\mathbf{G}}_o$ are the first-order terms of a Taylor expansion of the fluid equation in the ALE formulation around the non-dimensional equilibrium point $(\bar{\mathbf{w}}_o, \bar{\mathbf{x}}_o, \dot{\bar{\mathbf{x}}}_o)$. $\bar{\mathbf{H}}_o \in \mathbb{R}^{N_f \times N_f}$ is the Jacobian matrix of the non-dimensional CFD flux function $\bar{\mathbf{F}}$ with respect to the non-dimensional fluid state vector $\bar{\mathbf{w}}$ and, as such, is generally a full-rank matrix. $\bar{\mathbf{C}}_o$, $\bar{\mathbf{G}}_o$ and $\bar{\mathbf{E}}_o$ are coupling matrices between the fluid and structural subsystems. $\bar{\mathbf{C}}_o$ and $\bar{\mathbf{G}}_o$ are the Jacobian matrices of the non-dimensional flux function with respect to the non-dimensional fluid mesh motion. $\bar{\mathbf{E}}_o$ denotes the Jacobian matrix of the non-dimensional cell volumes in the fluid mesh with respect to the non-dimensional fluid mesh motion.

Similarly, the structural subsystem can be linearized around an equilibrium state by perturbing the following state variables

$$\mathbf{w} = \mathbf{w}_o + \delta\mathbf{w}, \quad \mathbf{u} = \mathbf{u}_o + \delta\mathbf{u}, \quad \dot{\mathbf{u}} = \dot{\mathbf{u}}_o + \delta\dot{\mathbf{u}}, \quad \ddot{\mathbf{u}} = \ddot{\mathbf{u}}_o + \delta\ddot{\mathbf{u}}. \tag{4.12}$$

Eq. (4.3) becomes

$$\mathbf{M}\delta\ddot{\mathbf{u}} + \mathbf{D}_o\delta\dot{\mathbf{u}} + \mathbf{K}_s\delta\mathbf{u} = \hat{\mathbf{P}}_o\delta\bar{\mathbf{w}}, \tag{4.13}$$

where

$$\mathbf{K}_s = \mathbf{K}_o - \frac{\partial\mathbf{f}^{\text{ext}}}{\partial\mathbf{u}}(\mathbf{w}_o, \mathbf{u}_o), \tag{4.14}$$

$$\mathbf{K}_o = \frac{\partial\mathbf{f}^{\text{int}}}{\partial\mathbf{u}}(\mathbf{u}_o, \dot{\mathbf{u}}_o), \tag{4.15}$$

$$\mathbf{D}_o = \frac{\partial\mathbf{f}^{\text{int}}}{\partial\dot{\mathbf{u}}}(\mathbf{u}_o, \dot{\mathbf{u}}_o), \tag{4.16}$$

$$\hat{\mathbf{P}}_o = \frac{\partial\mathbf{f}^{\text{ext}}}{\partial\bar{\mathbf{w}}}(\mathbf{w}_o, \mathbf{u}_o). \tag{4.17}$$

$\mathbf{K}_o$ and $\mathbf{D}_o$ denote respectively the structural stiffness and damping matrices at the operating point $\mathbf{u}_o$. $\mathbf{K}_s$ is an adjusted structural stiffness matrix accounting for the

dependence of the external loading on the operating point, resulting from the coupled formulation. $\hat{\mathbf{P}}_o$ arises from the linearization of the external forces with respect to the non-dimensional fluid state vector. The hat notation emphasizes the fact that $\hat{\mathbf{P}}_o$ is neither a fully dimensionalized nor non-dimensionalized quantity.

For simplification purposes and to alleviate the notations, the subscripts "o" and the prefixes δ will be dropped. Hence, the perturbed fluid state, mesh motions and structural displacement vectors will be denoted as $\bar{\mathbf{w}}$, $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ respectively.

By inspection of Eq. (4.4), it can be seen that the fluid mesh state vector $\bar{\mathbf{x}}$ can be eliminated from the formulation. The matrix $\tilde{\mathbf{K}}$ being a stiffness matrix, it is definite positive, hence invertible, and a condensed deformation transfer matrix $\bar{\mathbf{K}}$ can be introduced as

$$\bar{\mathbf{K}} = \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{K}}_c. \tag{4.18}$$

The variables $\bar{\mathbf{x}}$ and $\dot{\bar{\mathbf{x}}}$ can then be eliminated from the formulation by

$$\bar{\mathbf{x}} = \bar{\mathbf{K}}\bar{\mathbf{u}}, \quad \dot{\bar{\mathbf{x}}} = \bar{\mathbf{K}}\dot{\bar{\mathbf{u}}}, \tag{4.19}$$

$\bar{\mathbf{u}}$ denoting the non-dimensional displacement of the structure at the equilibrium point.

The linearized aeroelastic formulation can therefore be written as a system of two coupled matrix ODEs

$$\bar{\mathbf{A}}\bar{\mathbf{w}}_{,\tau} + \bar{\mathbf{H}}\bar{\mathbf{w}} + (\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}}\dot{\bar{\mathbf{u}}} + \bar{\mathbf{G}}\bar{\mathbf{K}}\bar{\mathbf{u}} = \mathbf{0}_{N_f} \tag{4.20}$$

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}_s\mathbf{u} = \hat{\mathbf{P}}\bar{\mathbf{w}} \tag{4.21}$$

This system can be exploited after re-dimensionalizing the two equation as

$$\mathbf{A}\mathbf{w}_{,t} + \mathbf{H}\mathbf{w} + \mathbf{R}\dot{\mathbf{u}} + \mathbf{G}\mathbf{u} \;=\; \mathbf{0}_{N_f}, \tag{4.22}$$

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}_s\mathbf{u} \;=\; \mathbf{P}\mathbf{w} \tag{4.23}$$

leading to the following matrix ODE

$$\begin{bmatrix} \mathbf{A} & \mathbf{0}_{N_f,N_s} & \mathbf{0}_{N_f,N_s} \\ \mathbf{0}_{N_s,N_f} & \mathbf{M} & \mathbf{0}_{N_s,N_s} \\ \mathbf{0}_{N_s,N_f} & \mathbf{0}_{N_s,N_s} & \mathbf{I}_{N_s} \end{bmatrix} \dot{\mathbf{q}} + \begin{bmatrix} \mathbf{H} & \mathbf{R} & \mathbf{G} \\ -\mathbf{P} & \mathbf{D} & \mathbf{K}_s \\ \mathbf{0}_{N_s,N_f} & -\mathbf{I}_{N_s} & \mathbf{0}_{N_s,N_s} \end{bmatrix} \mathbf{q} = \mathbf{0}_{N_f+2N_s,N_f+2N_s}, \tag{4.24}$$

with

$$\mathbf{q} = \begin{bmatrix} \mathbf{w} \\ \dot{\mathbf{u}} \\ \mathbf{u} \end{bmatrix}. \tag{4.25}$$

The stability of the coupled system could be determined in theory by studying the eigenvalues of the full-order matrix

$$\mathbf{N}_{\text{FOM}} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{N_f,N_s} & \mathbf{0}_{N_f,N_s} \\ \mathbf{0}_{N_s,N_f} & \mathbf{M} & \mathbf{0}_{N_s,N_s} \\ \mathbf{0}_{N_s,N_f} & \mathbf{0}_{N_s,N_s} & \mathbf{I}_{N_s} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{H} & \mathbf{R} & \mathbf{G} \\ -\mathbf{P} & \mathbf{D} & \mathbf{K}_s \\ \mathbf{0}_{N_s,N_f} & -\mathbf{I}_{N_s} & \mathbf{0}_{N_s,N_s} \end{bmatrix}. \tag{4.26}$$

In practice, this is a large-scale matrix and determining its full eigenspectrum is an intractable task.

For a pure fluid problem, $\mathbf{R}$ and $\mathbf{G}$ are set to $\mathbf{0}_{N_f,N_s}$ in Eq. (4.22).

4.4.2   Model Reduction

Although computational aeroelasticity involves the coupling of fluid and structural subsystem models, model reduction is performed here for each of them separately following the approach advocated in [25, 27]. This decoupled reduced-order modeling approach has several advantages. First, it allows using the most suitable ROM technique for each individual discipline. Second, it allows formulating the fluid model in non-dimensional form, thereby making its reduced-order basis independent of the free-stream pressure and density. The latter property makes the adaptation of the resulting aeroelastic ROM to a variation of altitude a trivial task, as shown in section 4.4.5.

Fluid ROM

The fluid ROM associated with the higher-order computational fluid model described above is represented here by the following mathematical entities:

1. Two full column rank matrices $\mathbf{\Phi}$, $\mathbf{\Psi} \in \mathbb{R}^{N_f \times N_\Phi}$ whose respective columns represent the bases of two subspaces and which satisfy the orthogonality condition

$$\mathbf{\Psi}^T \mathbf{\Phi} = \mathbf{I}_{N_\Phi}, \tag{4.27}$$

where $N_\Phi << N_f$ and $\mathbf{I}_{N_\Phi}$ denotes the identity matrix of dimension N_Φ. Hence, $\mathbf{\Phi}$ and $\mathbf{\Psi}$ belong to the non-compact Stiefel manifold $\mathbb{R}_*^{N_f \times N_\Phi}$. In this work, $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are obtained by applying either: (a) the POD method to Eq. (4.22) in the frequency domain as described in the next sections as well as in many references including [9, 10, 11, 16, 27]. In that case $\mathbf{\Psi} = \mathbf{\Phi}$ belongs to the compact Stiefel manifold $\mathcal{ST}(N_\Phi, N_f)$. (b) the Balanced POD method to Eq. (4.22) and its

dual system in the frequency domain as described in [42, 37] and in the next sections.

2. The reduced-order form of Eq. (4.20) which can be written as

$$(\bar{\mathbf{w}}_r)_{,\tau} - \bar{\mathbf{H}}_r \bar{\mathbf{w}}_r + \bar{\mathbf{B}}_r \bar{\mathbf{y}} = \mathbf{0}_{N_\Phi} \tag{4.28}$$

where $\bar{\mathbf{w}}_r$ denotes the oblique projection by $\boldsymbol{\Pi}_{\boldsymbol{\Phi},\boldsymbol{\Psi}}$ of the underlying non-dimensional full-order fluid state vector — that is,

$$\bar{\mathbf{w}}_r = \boldsymbol{\Psi}^T \bar{\mathbf{w}}, \quad \bar{\mathbf{w}} \approx \boldsymbol{\Phi} \bar{\mathbf{w}}_r. \tag{4.29}$$

$\bar{\mathbf{H}}_r \in \mathbb{R}^{N_\Phi \times N_\Phi}$ and $\bar{\mathbf{B}}_r \in \mathbb{R}^{N_\Phi \times 2N_s}$ are reduced matrices obtained by the Petrov-Galerkin projection $\boldsymbol{\Pi}_{\boldsymbol{\Phi},\boldsymbol{\Psi}}$ of the linear full-order equation as

$$\bar{\mathbf{H}} = -\boldsymbol{\Psi}^T \bar{\mathbf{A}}^{-1} \bar{\mathbf{H}} \boldsymbol{\Phi} \tag{4.30}$$

$$\bar{\mathbf{B}}_r = \boldsymbol{\Psi}^T \bar{\mathbf{A}}^{-1} \left[(\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}} \quad \mathbf{G}\bar{\mathbf{K}} \right]. \tag{4.31}$$

$\bar{\mathbf{y}}$ denotes the structural subsystem non-dimensional state vector

$$\bar{\mathbf{y}} = \begin{bmatrix} \dot{\bar{\mathbf{u}}} \\ \bar{\mathbf{u}} \end{bmatrix}. \tag{4.32}$$

Structural ROM

The linear ROM chosen for modeling the dynamics of the structural subsystem is based on the classical model reduction. The effects of the structural damping $\mathbf{D}$ and the gradient of the external loading with respect to the structural displacements $\dfrac{\partial \mathbf{f}^{\text{ext}}}{\partial \mathbf{u}}$

are neglected in the following model. Hence, the structural subsystem is represented here by the following three mathematical entities:

1. A set of N_m natural modes of the "dry" structure represented by the matrix of eigenvectors $\mathbf{X} \in \mathbb{R}^{N_s \times N_m}$.

2. The corresponding matrix of eigenvalues (squares of natural circular frequencies) $\mathbf{\Omega}^2 \in \mathbb{R}^{N_m \times N_m}$.

3. The reduced-order form of the governing equation of dynamic equilibrium which can be written as [25, 26, 28]

$$\mathbf{I}_{N_m} \ddot{\mathbf{u}}_m + \mathbf{\Omega}^2 \mathbf{u}_m = \mathbf{X}^T \hat{\mathbf{P}} \bar{\mathbf{w}} \tag{4.33}$$

where $\mathbf{u}_m$ denotes the vector of generalized (modal) coordinates.

Aeroelastic ROM

Hence, the linear aeroelastic ROM chosen in this work can be fully described by:

1. The matrices $\mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{R}^{N_f \times N_\Phi}$, $\mathbf{X} \in \mathbb{R}^{N_s \times N_m}$ and $\mathbf{\Omega}^2 \in \mathbb{R}^{N_m \times N_m}$ defined above.

2. The coupled system of governing equations

$$
\begin{aligned}
(\bar{\mathbf{w}}_r)_{,\tau} - \bar{\mathbf{H}}_r \bar{\mathbf{w}}_r + \bar{\mathbf{B}}_r \bar{\mathbf{y}} &= \mathbf{0}_{N_\Phi} \\
\mathbf{I}_{N_m} \ddot{\mathbf{u}}_m + \mathbf{\Omega}^2 \mathbf{u}_m &= \hat{\mathbf{P}}_m \bar{\mathbf{w}}_r
\end{aligned}
\tag{4.34}
$$

where

$$\bar{\mathbf{y}} = \begin{bmatrix} \mathbf{X}\dot{\bar{\mathbf{u}}}_m \\ \mathbf{X}\bar{\mathbf{u}}_m \end{bmatrix} \tag{4.35}$$

and

$$\hat{\mathbf{P}}_m = \mathbf{X}^T \hat{\mathbf{P}} \mathbf{\Phi} \tag{4.36}$$

4.4.3   Snapshot Generation in the Frequency Domain

In this dissertation, solutions of the fluid subsystem are computed to build the reduced bases for the fluid subsystems. These so-called snapshots are chosen to be generated in the frequency domain. In his thesis [84], Lieu has shown that generating the snapshots in the frequency domain delivers better results than the generation in the time domain, both in terms of reduction of the number of degrees of freedom and accuracy and stability of the resulting reduced-order models.

Generation of Primal Snapshots

The POD and Balanced POD methods generate complex-valued snapshots $\bar{\mathbf{w}}_j(k)$ solutions of Eq. (4.20) in the frequency domain for a given reduced frequency k as

$$\bar{\mathbf{w}}_j(k) = -(ik\bar{\mathbf{A}} + \bar{\mathbf{H}})^{-1}(ik(\bar{\mathbf{E}} + \bar{\mathbf{C}}) + \bar{\mathbf{G}})\bar{\mathbf{K}}\bar{\mathbf{u}}_j. \tag{4.37}$$

This equation arises by assuming a periodic solution $\bar{\mathbf{w}} = \bar{\mathbf{w}}_j e^{ik\tau}$ as well as a periodic excitation $\bar{\mathbf{u}} = \bar{\mathbf{u}}_j e^{ik\tau}$, where $\bar{\mathbf{u}}_j$ is chosen as a dry natural mode of the structural subsystem. For each mode $\bar{\mathbf{u}}_j$ a sweep on the reduced frequency k is performed as

$$k_l = l\Delta k, \quad l = 0, \cdots, L, \tag{4.38}$$

and $N_{snap}^{\mathbb{C}} = (L+1)N_m$ complex valued snapshots $\bar{\mathbf{w}}_j$ are generated. By taking their real and imaginary parts, and noting that N_m of these snapshots are real, $N_{snap}^{\mathbb{R}} = (2L+1)N_m$ real-valued snapshots are generated.

Alternatively, Eq. (4.37) can also be differentiated with respect to the reduced frequency k as

$$(ik\bar{\mathbf{A}} + \bar{\mathbf{H}})\frac{\partial \bar{\mathbf{w}}_j}{\partial k}(k) = -i((\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}}\bar{\mathbf{u}}_j + \bar{\mathbf{A}}\bar{\mathbf{w}}_j(k)) \tag{4.39}$$

and

$$(ik\bar{\mathbf{A}} + \bar{\mathbf{H}})\frac{\partial^n \bar{\mathbf{w}}_j}{\partial k^n}(k) = -ni\bar{\mathbf{A}}\frac{\partial^{n-1} \bar{\mathbf{w}}_j}{\partial k^{n-1}}(k), \quad n \geq 2. \tag{4.40}$$

Hence, a second strategy for generating snapshots corresponding to a set of reduced frequencies $k \in \mathcal{K}$ can be considered as follows.

1. Choose $N \in \mathbb{N}$.

2. Choose a subset $\mathcal{K}_c$ of $\mathcal{K}$.

3. Compute $\left\{ \bar{\mathbf{w}}_j(k), \dfrac{\partial \bar{\mathbf{w}}_j}{\partial k}(k), \cdots, \dfrac{\partial^N \bar{\mathbf{w}}_j}{\partial k^N}(k) \right\}_{k \in \mathcal{K}_c} \quad \forall j \in 1, \cdots, N_m$.

4. Reconstruct $\{\bar{\mathbf{w}}_j(k)\}_{k \in \mathcal{K} - \mathcal{K}_c} \quad \forall j \in 1, \cdots, N_m$ using Padé approximants.

The snapshots and their derivatives computed at $k \in \mathcal{K}_c$ are used to reconstruct the snapshots at $k \in \mathcal{K} - \mathcal{K}_c$. One Padé approximation [94] is computed for each component of the snapshot vectors. This strategy is efficient whenever less linear systems are solved, that is

$$|\mathcal{K}_c|(N + 1) < |\mathcal{K}|, \tag{4.41}$$

where $|\ |$ denotes the cardinality of a set. Such a strategy was first applied to frequency sweep applications [95]. Ideally, the matrix $(ik\bar{\mathbf{A}} + \bar{\mathbf{H}})$ could be factored once for each value of k and the snapshots $\left\{ \bar{\mathbf{w}}_j(k), \dfrac{\partial \bar{\mathbf{w}}_j}{\partial k}(k), \cdots, \dfrac{\partial^N \bar{\mathbf{w}}_j}{\partial k^N}(k) \right\}_{j \in 1, \cdots, N_m}$ be computed by cheap forward and backward substitutions. However, the aforementioned matrix is very large and sparse which renders this method impractical. An

iterative procedure based on Krylov subspaces recycling can be considered instead. This method is presented in Appendix I.

Generation of Dual Snapshots

As presented in section 2.5.3, the Balanced POD method proceeds by generating snapshots of the dual dynamical system as well. The primal system is in the present case

$$\bar{\mathbf{w}}_{,\tau} = -\bar{\mathbf{A}}^{-1}\bar{\mathbf{H}}\bar{\mathbf{w}} - \bar{\mathbf{A}}^{-1}\Big((\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}}\dot{\mathbf{u}} + \bar{\mathbf{G}}\bar{\mathbf{K}}\bar{\mathbf{u}}\Big) \tag{4.42}$$

$$\boldsymbol{\eta} = \hat{\mathbf{P}}_m\bar{\mathbf{w}}. \tag{4.43}$$

The output quantity of interest is $\boldsymbol{\eta} \in \mathbb{R}^{N_m}$ and it represents the generalized (modal) external loading on the structure. Hence, the corresponding dual equation takes the following form

$$\bar{\mathbf{z}}_{,\tau} = -\left(\bar{\mathbf{A}}^{-1}\bar{\mathbf{H}}\right)^T \bar{\mathbf{z}} + \hat{\mathbf{P}}_m^T\mathbf{v}, \tag{4.44}$$

where $\mathbf{v} \in \mathbb{R}^{N_m}$.

Complex-valued dual snapshots $\bar{\mathbf{z}}_j(k)$ are then generated in the frequency domain for a given reduced frequency k as

$$\bar{\mathbf{z}}_j(k) = \left(-ik\mathbf{I}_{N_f} + \left(\bar{\mathbf{A}}^{-1}\bar{\mathbf{H}}\right)^T \right)^{-1} \hat{\mathbf{P}}_m^T\mathbf{v}_j, \tag{4.45}$$

that is, since $\bar{\mathbf{A}}$ is diagonal,

$$\bar{\mathbf{z}}_j(k) = \bar{\mathbf{A}}\left(-ik\bar{\mathbf{A}} + \bar{\mathbf{H}}^T\right)^{-1} \hat{\mathbf{P}}_m^T\mathbf{v}_j. \tag{4.46}$$

Hence, for a given frequency sweep, the same number of primal and dual snapshots

are generated.

Similarly to the case of primal snapshots, a strategy based on Padé reconstruction can be designed to alleviate the cost of solving a linear system for each snapshot.

4.4.4 Reduced Bases Construction

Once that $N_{snap}^{\mathbb{C}}$ snapshots are generated for each system, these vectors are stored as columns of two complex-valued snapshot matrices $\bar{\mathbf{W}}$, $\bar{\mathbf{Z}} \in \mathbb{C}^{N_f \times N_{snap}^{\mathbb{C}}}$ scaled appropriately (see section 2.5.2), from which two real-valued snapshot matrices are constructed

$$\mathbf{S}_{\bar{\mathbf{W}}} = \begin{bmatrix} \mathrm{Re}\left(\bar{\mathbf{W}}\right) & \mathrm{Im}\left(\bar{\mathbf{W}}\right) \end{bmatrix}, \quad \mathbf{S}_{\bar{\mathbf{Z}}} = \begin{bmatrix} \mathrm{Re}\left(\bar{\mathbf{Z}}\right) & \mathrm{Im}\left(\bar{\mathbf{Z}}\right) \end{bmatrix} \in \mathbb{R}^{N_f \times N_{snap}^{\mathbb{R}}}. \tag{4.47}$$

The following algorithm is suitable for both the POD and balanced POD methods. In the case of the POD basis construction, $\mathbf{S}_{\bar{\mathbf{Z}}} = \mathbf{S}_{\bar{\mathbf{W}}}$. The algorithm proceeds in four steps as follows;

1. Form the real-valued small size $N_{snap}^{\mathbb{R}}$-by-$N_{snap}^{\mathbb{R}}$ correlation matrix

$$\mathbf{R} = \mathbf{S}_{\bar{\mathbf{Z}}}^T \mathbf{S}_{\bar{\mathbf{W}}}. \tag{4.48}$$

2. Compute the compact singular value decomposition of $\mathbf{R}$

$$\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{4.49}$$

Thus, $\mathbf{\Sigma}$ is an invertible diagonal matrix of size r, where $r \leq N_{snap}^{\mathbb{R}}$ denotes the rank of $\mathbf{R}$.

3. Form the reduced bases

$$\tilde{\mathbf{\Phi}} = \mathbf{S}_{\bar{\mathbf{W}}} \mathbf{V} \mathbf{\Sigma}^{-1/2} \in \mathbb{R}^{N_f \times r} \tag{4.50}$$

$$\tilde{\mathbf{\Psi}} = \mathbf{S}_{\bar{\mathbf{Z}}} \mathbf{U} \mathbf{\Sigma}^{-1/2} \in \mathbb{R}^{N_f \times r}. \tag{4.51}$$

4. Truncate the reduced bases by retaining the first $N_\Phi \leq r$ columns of $\tilde{\mathbf{\Phi}}$ and $\tilde{\mathbf{\Psi}}$. $\mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{R}_*^{N_f \times N_\Phi}$ are constructed.

In the case of the POD basis construction, this method can be improved. The correlation matrix $\mathbf{R} = \mathbf{S}_{\bar{\mathbf{W}}}^T \mathbf{S}_{\bar{\mathbf{W}}}$ is indeed symmetric and its singular value decomposition is also an eigenvalue decomposition

$$\mathbf{R} = \mathbf{S}_{\bar{\mathbf{W}}}^T \mathbf{S}_{\bar{\mathbf{W}}} = \mathbf{V} \mathbf{\Sigma} \mathbf{V}^T. \tag{4.52}$$

Hence, $\tilde{\mathbf{\Phi}} = \mathbf{S}_{\bar{\mathbf{W}}} \mathbf{V} \mathbf{\Sigma}^{-1/2}$ can also be obtained by a direct singular value decomposition of the snapshot matrix $\mathbf{S}_{\bar{\mathbf{W}}}$ as

$$\mathbf{S}_{\bar{\mathbf{W}}} = \tilde{\mathbf{\Phi}} \mathbf{\Sigma}^{1/2} \mathbf{V}^T. \tag{4.53}$$

$\mathbf{\Phi}$ is then obtained by truncated the first N_Φ left singular vectors $\tilde{\mathbf{\Phi}}$ of the snapshot matrix $\mathbf{S}_{\bar{\mathbf{W}}}$. Although the singular value decomposition of an N_f-by-N_Φ matrix is more expensive to compute than the eigenvalue decomposition of a "small" N_Φ-by-N_Φ matrix, this approach is better suited in terms of numerical stability and roundoff properties. The condition number of $\mathbf{S}_{\bar{\mathbf{W}}}$ is indeed the square root of the one for $\mathbf{R} = \mathbf{S}_{\bar{\mathbf{W}}}^T \mathbf{S}_{\bar{\mathbf{W}}}$.

4.4.5   Reduced-Order Model Dimensionalization and Exploitation

From Eq. (4.34), the reduced coupled equation for the aeroelastic system can be written as

$$(\bar{\mathbf{w}}_r)_{,\tau} - \bar{\mathbf{H}}_r\bar{\mathbf{w}}_r + \bar{\mathbf{R}}_r\dot{\mathbf{u}}_m + \bar{\mathbf{G}}_r\mathbf{u}_m \;=\; \mathbf{0}_{N_\Phi} \tag{4.54}$$

$$\mathbf{I}_{N_m}\ddot{\mathbf{u}}_m + \mathbf{\Omega}^2\mathbf{u}_m \;=\; \hat{\mathbf{P}}_m\bar{\mathbf{w}}_r \tag{4.55}$$

which is the reduced-order counterpart of the system (4.20)–(4.21).

After redimensionalization of the fluid equation (4.54), the following system of matrix ODEs is obtained [84]

$$\dot{\mathbf{q}}_r + \begin{bmatrix} \sqrt{\dfrac{p_\infty}{\rho_\infty}}\hat{\mathbf{H}}_r & \mathbf{R}_r & \sqrt{\dfrac{p_\infty}{\rho_\infty}}\hat{\mathbf{G}}_r \\[2mm] -p_\infty\hat{\mathbf{P}}'_m & \mathbf{0}_{N_m,N_m} & \mathbf{\Omega}^2 \\[2mm] \mathbf{0}_{N_m,N_\Phi} & -\mathbf{I}_{N_m} & \mathbf{0}_{N_m,N_m} \end{bmatrix} \mathbf{q}_r = \mathbf{0}_{N_\Phi+2N_m,N_\Phi+2N_m}, \tag{4.56}$$

where p_∞ and ρ_∞ denote the free-stream pressure and density at the desired altitude, respectively and

$$\mathbf{q}_r = \begin{bmatrix} \mathbf{w}_r \\ \dot{\mathbf{u}}_m \\ \mathbf{u}_m \end{bmatrix}. \tag{4.57}$$

One can therefore define the dimensional 3-by-3 block, real matrix

$$\mathbf{N}_{\mathrm{ROM}}(p_\infty, \rho_\infty) = \begin{bmatrix} \sqrt{\dfrac{p_\infty}{\rho_\infty}}\hat{\mathbf{H}}_r & \mathbf{R}_r & \sqrt{\dfrac{p_\infty}{\rho_\infty}}\hat{\mathbf{G}}_r \\ -p_\infty\hat{\mathbf{P}}'_m & \mathbf{0}_{N_m,N_m} & \mathbf{\Omega}^2 \\ \mathbf{0}_{N_m,N_\Phi} & -\mathbf{I}_{N_m} & \mathbf{0}_{N_m,N_m} \end{bmatrix}. \tag{4.58}$$

The block matrices $\hat{\mathbf{H}}_r, \mathbf{R}_r, \hat{\mathbf{G}}_r, \hat{\mathbf{P}}'_m$ in $\mathbf{N}_{\mathrm{ROM}}(p_\infty, \rho_\infty)$ depend only on the flight conditions (Mach number M_∞ and angle of attack α) and not on the altitude parameters. Hence, a ROM that is generated for a given operating point (M_∞, α) by the present algorithm can be exploited at a variety of altitudes by specifying the corresponding values of (p_∞, ρ_∞) in Eq. (4.58) using an adequate atmospheric model.

The matrix $\mathbf{N}_{\mathrm{ROM}}$ can be exploited at least in the following ways:

- The eigenvalue analysis of $\mathbf{N}_{\mathrm{ROM}}$ determines the stability of the aeroelastic system for the flight conditions defined by p_∞ and ρ_∞. Altitude sweeps are simply performed by changing appropriately the values of the free-stream pressure and density in $\mathbf{N}_{\mathrm{ROM}}$.

- The matrix blocks of $\mathbf{N}_{\mathrm{ROM}}$ may be used to form the system of equations (4.56). The aeroelastic ROM can then be used for time-domain simulations provided that initial conditions are specified for $\mathbf{u}_m$ and $\mathbf{w}_r$, or to control the aircraft, as in aeroservoelasticity applications [96, 97].

4.4.6 Aeroelastic Parameters Extraction

A first method for determining the stability of the aeroelastic system is to compute the time histories of the full state vectors under initial perturbations of the steady-state aeroelastic equilibrium. When such histories are computed, (see section 4.4.7),

integrated quantities such as the lift or drag of the aircraft can be determined. The damping coefficients and eigenfrequencies of the aeroelastic modes in the coupled system can then be obtained by an analysis using the Eigensystem Realization Algorithm (ERA) [98]. These parameters are indicators of the aeroelastic stability of the system.

Another approach can be used when computing the critical flutter point. For a given flight condition (speed and angle of attack), the critical flutter point corresponds to the pressure p_∞ leading to a damping ratio of zero. Whenever the aeroelastic system is considered in a linearized analysis, this pressure can be computed by considering the eigenvalues of the matrix $\mathbf{N}_{\text{FOM}}$ in Eq. (4.26). However, computing the eigenvalues of this large matrix is impractical. Alternatively, computing the eigenvalues of the aeroelastic ROM operator $\mathbf{N}_{\text{ROM}}$ is cheap due to its small size.

Flutter analysis can also be conducted by computing a flutter speed index, which is defined as the ratio if the flutter velocity to an adimensionalizing constant [84] . The flutter velocity is found by computing the aeroelastic responses of the system for various pairs of Mach numbers and angle of attacks.

4.4.7 Time Integration

The generalized second-order backward difference (BDF) implicit scheme algorithm can be written as a $m = 2$-step method under the form of Eq. (4.59):

$$\alpha_m \mathbf{x}^{n+m} + \alpha_{m-1}\mathbf{x}^{n+m-1} + \cdots + \alpha_0 \mathbf{x}^n = \beta_m \mathbf{f}\left(\mathbf{x}^{n+m}, \mathbf{u}^{n+m}; \boldsymbol{\mu}\right) + \cdots + \beta_0 \mathbf{f}\left(\mathbf{x}^n, \mathbf{u}^n; \boldsymbol{\mu}\right),$$

$$(4.59)$$

with

$$\alpha_2 = \frac{1 + 2\tau_{n+1}}{1 + \tau_{n+1}}, \quad \alpha_1 = -1 - \tau_{n+1}, \quad \alpha_0 = \frac{\tau_{n+1}^2}{1 + \tau_{n+1}}, \quad \tau_{n+1} = \frac{\Delta t_{n+1}}{\Delta t_n}, \qquad (4.60)$$

and

$$\beta_2 = \Delta t_{n+1}, \quad \beta_1 = \beta_0 = 0. \qquad (4.61)$$

Here $\Delta t_n = t_n - t_{n-1}$, where t_i, $i = 0, \cdots, N_T$, is a sequence of strictly increasing times. For simplification purposes, it is assumed here that $t_0 = 0$ without loss of generality.

In order to define a second-order scheme for the coupled system of semi-discretized ODEs (4.20)–(4.21) with the non-damping assumption $\mathbf{D} = \mathbf{0}_{N_s,N_s}$, the three-point BDF scheme is applied to the fluid equation and a scheme based on the midpoint rule is applied to the structural equation. However, the initial conditions defined at $t_0 = 0$ cannot be used to initiate the three-point scheme. Therefore, an explicit two-point BDF (forward-Euler) scheme is chosen for the first iteration. The corresponding time step $\tilde{\Delta} t$ is chosen to be small in order to preserve accuracy and stability. As such, it is chosen as $\tilde{\Delta} t = \Delta t^2$, where $\Delta t = \Delta t_i$, $1 = 2, \cdots, N_T$, denotes the time step in the successive iterations.

A staggered algorithm is defined where the vectors of structural displacement and velocities $\mathbf{u}$ and $\dot{\mathbf{u}}$ are computed at times t_n, $n \in \mathbb{N}$ and the fluid state vector $\mathbf{w}$ as well as predictions for the structural vectors $\mathbf{u}^P$ and $\dot{\mathbf{u}}^P$ are computed at intermediary times $t_{n+1/2}$, $n \in \mathbb{N}$. A schematic representation of the scheme is sketched in Fig. 4.1.
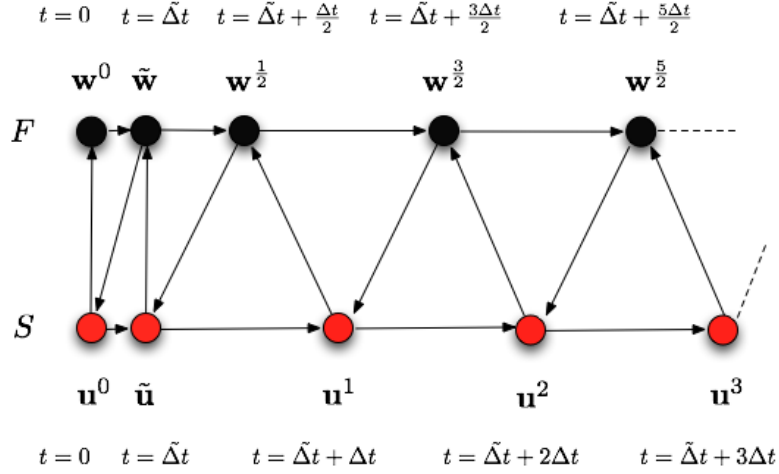
Figure 4.1: Staggered time integration scheme

The second-order BDF scheme for Eq. (4.20) is

$$\alpha_2 \bar{\mathbf{A}} \mathbf{w}^{n+\frac{5}{2}} + \alpha_1 \bar{\mathbf{A}} \mathbf{w}^{n+\frac{3}{2}} + \alpha_0 \bar{\mathbf{A}} \mathbf{w}^{n+\frac{1}{2}} + \Delta t_n \left(\bar{\mathbf{H}} \mathbf{w}^{n+\frac{5}{2}} + (\bar{\mathbf{E}} + \bar{\mathbf{C}}) \bar{\mathbf{K}} \dot{\mathbf{u}}^{n+\frac{5}{2}^P} + \bar{\mathbf{G}} \bar{\mathbf{K}} \mathbf{u}^{n+\frac{5}{2}^P} \right)$$

$$= \mathbf{0}_{N_f},$$

(4.62)

where the coefficients α_i, $i = 0, 1, 2$ depend on the chosen time step Δt_n by (4.60). Hence, the fact that the first step is explicit and has with a different time step modifies the corresponding τ's and α_i's in the first three time iterations. The values of the coefficients α_i at each iteration are reported in Appendix J.

4.5 Application to the F-16 Block 40

The methodology developed in the previous sections is here applied to an aeroelastic model of an F-16 Block 40 aircraft. After comparing the respective CPU timings for aeroelastic behavior predictions using (a) the high-fidelity linearized model, (b) a

reduced-order linear model, and observing the non-robustness of the ROM, the ROB interpolation procedure is applied.

4.5.1 Construction of F-16 ROMs

The considered high-fidelity aeroelastic computational model of an F-16 Block 40 aircraft in clean wing configuration consists of: (a) a FE structural model with 168,799 dofs built with bar, beam, solid, plate, and shell, metallic, and composite elements, and (b) an unstructured CFD (Euler) grid with 403,919 vertices and more than two million dofs (Figure 4.2).

Complex-valued snapshots are generated by solving Eq. (4.37) for $L + 1 = 6$ different values of k and $N_m = 9$ different exciting structural modes $\bar{\mathbf{u}}_j$. $(2L+1)N_m = 99$ real-valued snapshots are hence generated. The reduced frequencies are chosen as

$$k_l = l\Delta k, \quad l = 0, \cdots, 5, \tag{4.63}$$

where $\Delta k = 0.004$. The structural modes that are chosen are reported in Table 4.1.

Mode number	Frequency (Hz)	Description
1	4.72	Anti-symmetric bending
3	4.75	Symmetric bending
5	5.58	Pitching
9	7.60	Symmetric torsion
11	7.83	Fuselage bending
16	12.35	Tail torsion
26	16.14	Symmetric wing bending
36	19.47	
37	19.77	

Table 4.1: F-16 fighter retained modes for the structure
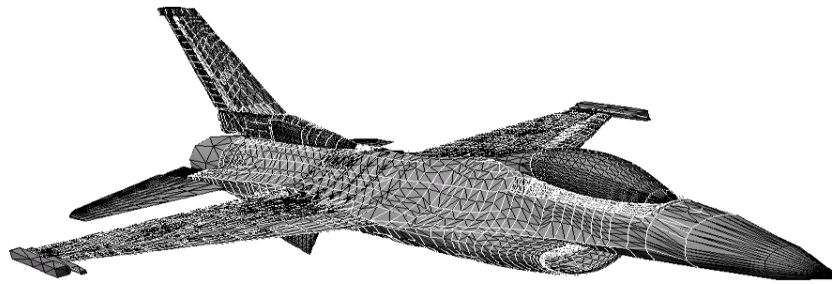
(a) FE structural model
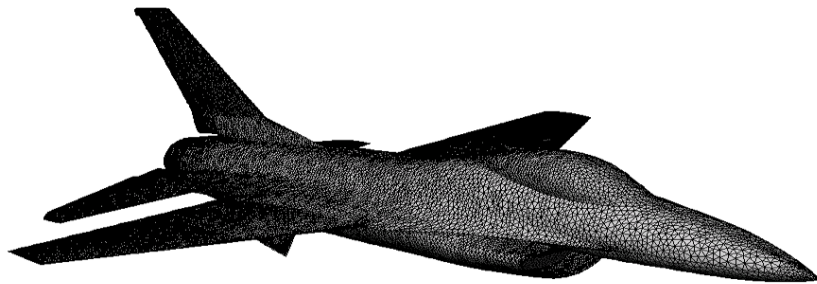
(b) CFD surface grid

Figure 4.2: High-fidelity aeroelastic model of an F-16 Block 40 configuration.
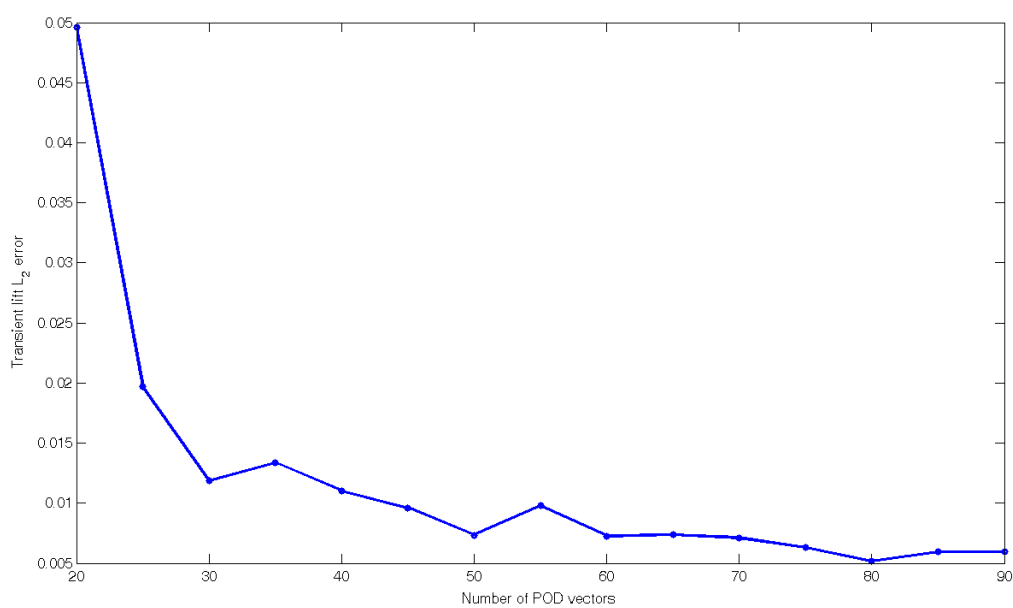
Figure 4.3: Transient lift L_2-norm of the relative error between the high-fidelity linearized full-order model and ROMs computed for various numbers of POD vectors at $(M_\infty, \alpha) = (0.799, 3.0°)$.
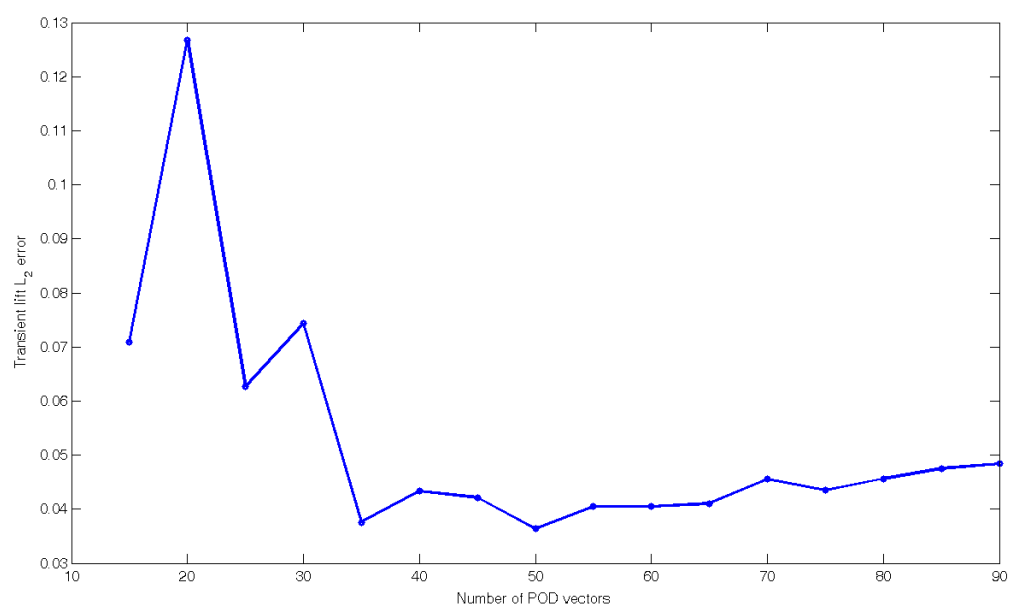
Figure 4.4: Transient lift L_2-norm of the relative error between the high-fidelity linearized full-order model and ROMs computed for various numbers of POD vectors at $(M_\infty, \alpha) = (1.030, 1.45°)$.

Figures 4.3 and 4.4 report the lift time-histories L_2-norm relative error between the full-order model and ROMs computed with various numbers of vectors in the POD basis for $(M_\infty, \alpha) = (0.799, 3.0°)$ and $(M_\infty, \alpha) = (1.030, 1.45°)$ respectively. While the error tends to decrease as the number of POD vectors increases, it can be seen, however, that in some cases, adding POD vectors in the basis can worsen slightly the precision of the ROM response. At $(M_\infty, \alpha) = (0.799, 3.0°)$, when more than 40 vectors are present in the POD basis, the relative error is less than one percent. At $(M_\infty, \alpha) = (1.030, 1.45°)$, when more than 35 POD vectors are used, it is less than 4%.

Computational step	N_p	CPU time	N_p	CPU time
Computation of steady-state equilibrium	32	7	64	3.5
Generation of 99 fluid snapshots and a fluid ROB (90)	32	25.5	64	13
Construction of fluid ROM	32	0.5	64	0.25
Assembly and post-processing of aeroelastic ROM	1	1	1	1
Total CPU time (ROM)		34		17.75
Total CPU time (FOM)	32	26	64	12.5

Table 4.2: Computational cost associated with the prediction of the first second of aeroelastic response using an aeroelastic ROM based on a fluid ROB computed directly at the desired flight point and $N_p = 32$ or $N_p = 64$ processors with a first-order numerical flux reconstruction (Time in minutes)

Table 4.2 reports the CPU timings associated with creating and exploiting an aeroelastic ROM built from a fluid model where the fluxes where obtained by a first-order reconstruction. More than 78% of the total CPU time of 32 minutes is consumed by the generation of the snapshots in the frequency-domain and the construction of the fluid ROB of dimension 90 using the POD method. All reported computations are shown to scale well when the number of processors is doubled. Predicting the

Computational step	N_p	CPU time	N_p	CPU time
Computation of steady-state equilibrium	32	12	64	6
Generation of 99 fluid snapshots and a fluid ROB (90)	32	90	64	50
Construction of fluid ROM	32	0.5	64	0.25
Assembly and post-processing of aeroelastic ROM	1	1	1	1
Total CPU time (ROM)		103.5		57.25
Total CPU time (FOM)	32	44	64	21.5

Table 4.3: Computational cost associated with the prediction of the first second of aeroelastic response using an aeroelastic ROM based on a fluid ROB computed directly at the desired flight point and $N_p = 32$ or $N_p = 64$ processors with a second-order numerical flux reconstruction. (Time in minutes)

counterpart aeroelastic response (same time-interval, same 500 time-steps) using the linearized high-dimensional computational model (FOM) produces almost the same results but requires 26 minutes CPU on 32 processors, and 12.5 minutes on 64 processors. When the fluxes are obtained by second-order reconstruction, the corresponding timings as indicated in Table 4.3. In that case, predicting the response using the linearized high-fidelity model requires 44 minutes CPU on 32 processors and 21.5 minutes on 64 processors.

Hence, generating and exploiting the ROM is computationally more expensive than using the FOM to obtain an aeroelastic response of the aircraft. This observation reinforces the use of an interpolation method for generating the ROBs in a way that bypasses the expensive computation of snapshots.

4.5.2 Adaptation of F-16 ROMs to New Free-Stream Data in the Transonic Regime

POD Basis Non-Robustness

First, the lack of robustness of a POD-based aeroelastic ROM of this aircraft with respect to a variation of the free-stream Mach number is highlighted. For this purpose, a single POD basis is constructed at the trimmed flight condition ($M_{\infty_2} = 0.710, \alpha_2 = 3.2°$). The projection on this reduced-order basis of the linearization of the fluid model at this flight condition is combined with a modal representation of the aforementioned high-fidelity FE structural model to obtain an aeroelastic ROM of dimension 69. Then, this ROM is used to predict the aeroelastic response of the F-16 aircraft to an initial excitation at three different trimmed flight conditions: ($M_{\infty_1} = 0.611, \alpha_1 = 4.5°$), ($M_{\infty_2} = 0.710, \alpha_2 = 3.2°$) and ($M_{\infty_3} = 0.799, \alpha_3 = 3.0°$). The obtained lift-time histories are reported in Figure 4.5 and compared to their counterparts predicted using the linearization of the high-fidelity aeroelastic model at the same considered flight conditions. The reader can observe that when applied at ($M_{\infty_1} = 0.611, \alpha_1 = 4.5°$), the aeroelastic ROM constructed at ($M_{\infty_2} = 0.710, \alpha_2 = 3.2°$) is unstable (the X symbol in Figure 4.5 is used to highlight the breaking of the time-dependent computation because of a numerical instability that causes some density and/or pressure values to become negative). When applied at ($M_{\infty_3} = 0.799, \alpha_3 = 3.0°$), the aforementioned aeroelastic ROM is weakly unstable and in any case totally inaccurate.
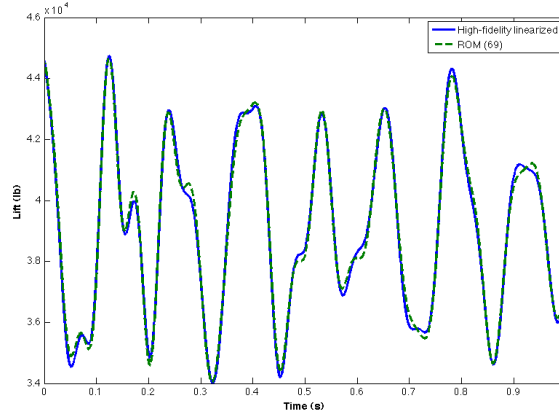
(a) $M_{\infty_2} = 0.710$, $\alpha_2 = 3.2°$
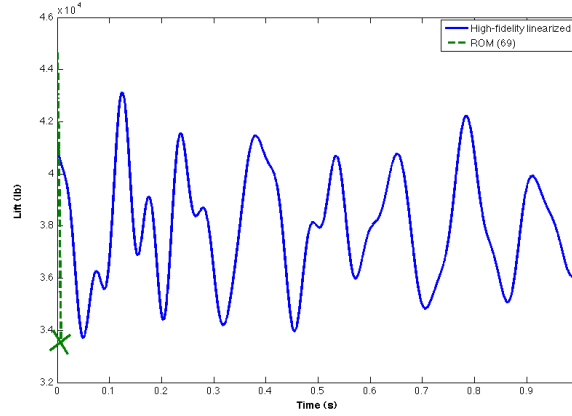
(b) $M_{\infty_1} = 0.611$, $\alpha_1 = 4.5°$
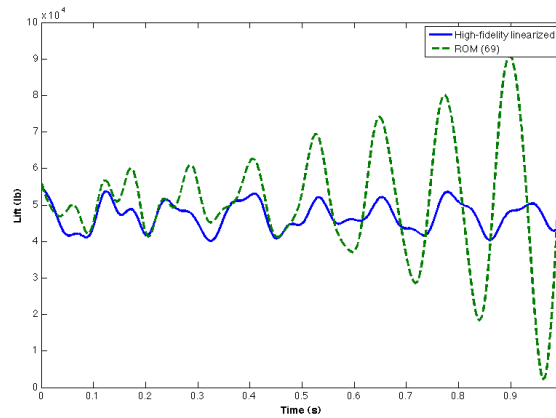
(c) $M_{\infty_3} = 0.799$, $\alpha_3 = 3.0°$

Figure 4.5: Comparison of the lift time-histories predicted for an F-16 configuration at $(M_{\infty_1} = 0.611, \alpha_1 = 4.5°)$, $(M_{\infty_2} = 0.710, \alpha_2 = 3.2°)$ and $(M_{\infty_3} = 0.799, \alpha_3 = 3.0°)$ using: (a) the linearization of the high-fidelity models at the considered flight-conditions, and (b) the ROM constructed at (M_{∞_2}, α_2).

POD Basis Interpolation in the Lower Transonic Regime

Next, the potential of the new interpolation method is assessed. To this effect, three different sets of numerical simulations are performed. In the first set, four POD bases of dimension 90 are pre-computed at the flight conditions ($M_{\infty_1} = 0.650, M_{\infty_2} = 0.710, M_{\infty_3} = 0.850$, and $M_{\infty_4} = 0.875$) and stored in a database. In all four cases, the free-stream angle of attack is set to $\alpha = 3.0°$. The method of interpolation in a tangent space to the Grassmann manifold is then applied to construct a new POD basis at the flight condition ($M_\infty = 0.799, \alpha = 3.0°$) by interpolating: (a) the pre-computed POD bases associated with the three operating points M_{∞_1}, M_{∞_2} and M_{∞_3}, and (b) the pre-computed POD bases associated with the four operating points M_{∞_1}, M_{∞_2}, M_{∞_3} and M_{∞_4}. Figure 4.6 reports the lift time-histories obtained using both interpolated ROMs, a POD-based aeroelastic ROM of the same dimension that is directly constructed at the new target flight condition, and the linearization at the same target flight condition ($M_\infty = 0.799$ and $\alpha = 3.0°$) of the high-fidelity nonlinear aeroelastic model. The reader can observe that while the aeroelastic ROM obtained by interpolating the three first pre-computed ROMs tracks well the lift time-history produced by the ROM directly constructed at the new target flight condition, there is an increasing offset between the two results after $t = 0.3$ s. However, the aeroelastic ROM obtained by interpolating all four pre-computed aeroelastic ROMs reproduces perfectly (less than 1% relative error) both lift time-histories produced by the ROM directly constructed at the new target flight condition, and that generated by the linearized high-fidelity aeroelastic model.

In a second experiment, four POD bases of dimension 90 are pre-computed at the flight conditions $(M_{\infty_1}, \alpha_1) = (0.775, 2.4°)$, $(M_{\infty_2}, \alpha_2) = (0.799, 3.0°)$, $(M_{\infty_3}, \alpha_3) = (0.81, 2.75°)$ and $(M_{\infty_4}, \alpha_4) = (0.83, 2.5°)$. A target flight condition $(M_{\infty_5}, \alpha_5) =$

Figure 4.6: Comparison of the lift time-histories of the F-16 configuration at ($M_\infty = 0.799$, $\alpha = 3.0°$) produced by: (a) a ROM obtained by interpolating in a tangent space to a Grassmann manifold three aeroelastic ROMs pre-computed at $M_{\infty_1} = 0.65$, $M_{\infty_2} = 0.71$ and $M_{\infty_3} = 0.85$, (b) a ROM obtained by adapting four aeroelastic ROMs pre-computed at $M_{\infty_1} = 0.65$, $M_{\infty_2} = 0.71$, $M_{\infty_3} = 0.85$ and $M_{\infty_4} = 0.875$ using the same interpolation method, (c) an aeroelastic ROM directly constructed at (M_∞, α), and (d) the linearization of a high-fidelity aeroelastic model at (M_∞, α).

(0.790, 2.5°) is considered. First each pre-computed POD basis is tested in order to reduce the full-order model computed at the target configuration. As in the first numerical experiment conducted above, non-robustness of the ROB can be observed from Fig. 4.7. A ROB obtained by interpolation of the pre-computed ones on the Grassmann manifold is then computed and an adapted ROM built. One can observe from Fig. 4.7 that the interpolated ROB is accurate enough to predict the aeroelastic behavior at the target flight condition.

POD Basis Interpolation for the Transonic Dip Prediction

In a third experiment, two POD bases of dimension 90 are first pre-computed at the trimmed flight conditions $(M_{\infty_1} = 0.923, \alpha_1 = 1.4°)$ and $(M_{\infty_2} = 1.114, \alpha_2 = 1.5°)$ and combined with the aforementioned structural modal representation. In [28], the subspace angle interpolation method was applied to interpolate these reduced-order bases in the interval $M_\infty \in [0.923, 1.114]$ and the corresponding trim angles of attack. It was found and reported that in this case, the subspace angle interpolation method fails to construct a stable interpolated POD at each of the two operating points $M_\infty = 1.072$ and $M_\infty = 1.031$. Consequently, an intermediate POD basis was pre-computed at $M_{\infty_3} = 1.031$ and its trim angle of attack, $\alpha_3 = 1.4°$, and piece-wise linear interpolations were reattempted in the same interval $M_\infty \in [0.923, 1.114]$ using the subspace angle interpolation method (see [28] for details). The obtained results, which were first reported in [28], are included here in Figure 4.8 as a reference point. The reader can observe that in this case, the piece-wise linearly interpolated ROMs using the subspace angle interpolation method track relatively well the variation of the first torsional aeroelastic damping ratio with the free-stream Mach number, except for $M_\infty = 0.990$.
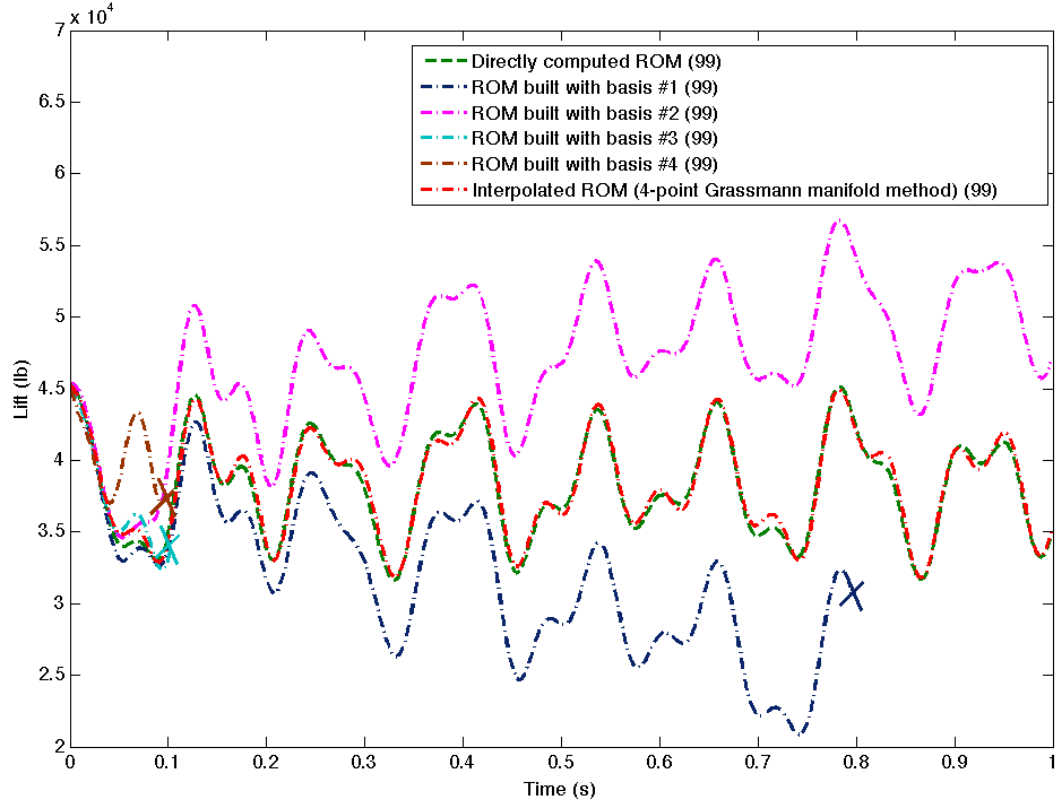
Figure 4.7: Comparison of the lift time-histories of the F-16 configuration at $(M_{\infty 5} = 0.790, \alpha_5 = 2.5°)$ produced by: (a) an aeroelastic ROM directly constructed at $(M_{\infty 5}, \alpha_5)$, (b) aeroelastic ROMs built with each one of the pre-computed bases (c) a ROM obtained by interpolating in a tangent space to a Grassmann manifold of the pre-computed four aeroelastic ROMs.
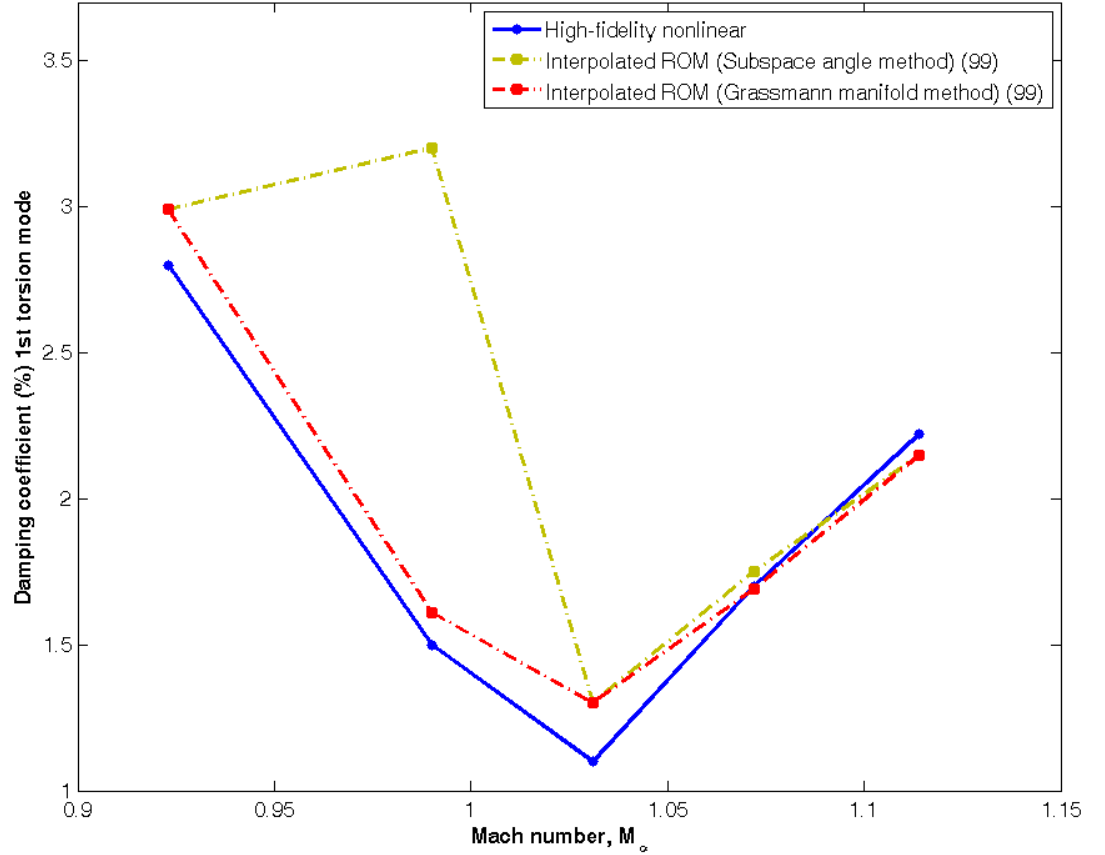
Figure 4.8: Comparison of the first torsional aeroelastic damping coefficients of the F-16 configuration predicted in the transonic interval $M_\infty \in [0.923, 1.114]$ using: (a) the subspace angle interpolation method applied to the piece-wise linear interpolation of three aeroelastic ROMs pre-computed at the three trimmed flight conditions $(M_{\infty_1} = 0.923, \alpha_1 = 1.4°)$, $(M_{\infty_2} = 1.031, \alpha_2 = 1.4°)$ and $(M_{\infty_3} = 1.114, \alpha_3 = 1.5°)$, (b) the higher-order method of interpolation in a tangent space to a Grassmann manifold applied to the same three pre-computed ROMs, and (c) a high-fidelity nonlinear aeroelastic model.

Here, a different attempt at interpolating the three aforementioned pre-computed POD bases in the same transonic interval $M_\infty \in [0.923, 1.114]$ is performed using the proposed method of interpolation in a tangent space to a Grassmann manifold. Indeed, unlike the subspace angle interpolation method, the more general method proposed in this paper allows higher-order approximations and can interpolate in one shot all three POD bases pre-computed at $(M_{\infty_1} = 0.923, \alpha_1 = 1.4°)$, $(M_{\infty_2} = 1.114, \alpha_2 = 1.5°)$ and $(M_{\infty_3} = 1.031, \alpha_3 = 1.4°)$. Figure 4.8 shows that in this case, the new ROM adaptation method proposed in section 4.3 delivers more accurate results in the entire transonic interval $M_\infty \in [0.923, 1.114]$.

4.6 Application to the F-18/A

4.6.1 Construction of F-18/A ROMs

The high-fidelity aeroelastic computational model of an F-18/A configuration with tip missiles considered here consists of an 11,290-dof FE structural model and an unstructured CFD (Euler) grid with 716,619 vertices and thus more than three and a half million dofs (Figure 4.9).

Complex-valued snapshots are generated by solving Eq. (4.37) for $L + 1 = 11$ different values of k and $N_m = 11$ different exciting structural modes $\bar{\mathbf{u}}_j$. $(2L+1)N_m = 231$ real-valued snapshots are hence computed. The reduced frequencies are chosen as

$$k_l = l\Delta k, \quad l = 0, \cdots, 11, \tag{4.64}$$

where $\Delta k = 0.001$. The modes that are chosen are reported in Table 4.4.
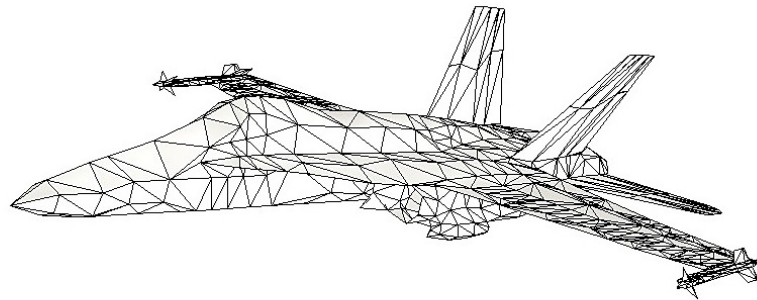
(a) FE structural model
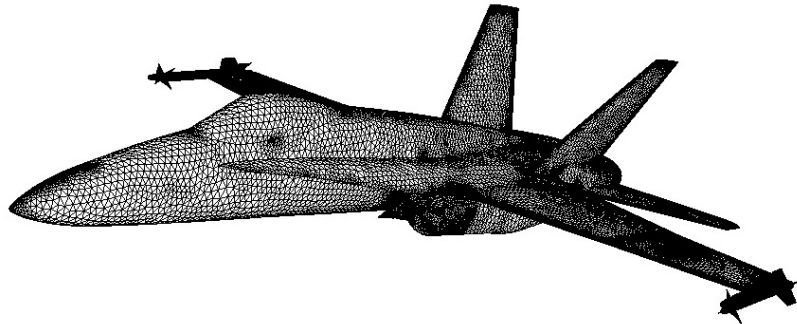
(b) CFD surface grid

Figure 4.9: High-fidelity aeroelastic model of an F-18/A configuration.

Mode number	Frequency (Hz)	Description
1	4.32	Symmetric bending
2	6.30	Anti-symmetric bending
3	7.96	Anti-symmetric tip missile pitching
4	8.47	Symmetric tip missile pitching
5	8.96	Wing torsion
6	9.54	Fuselage bending
7	10.62	Symmetric tip missile yawing
8	11.15	Anti-symmetric tip missile yawing
9	13.69	Second symmetric bending
10	13.97	Mixed symmetric bending and tip missile yawing
11	14.21	Mixed anti-symmetric bending and tip missile yawing

Table 4.4: F-18/A aircraft retained modes for the structure

Figure 4.10 reports the lift time-history L_2-norm relative error between the full-order model and ROMs computed with various numbers of vectors in the POD basis for $(M_\infty, \alpha) = (0.725, 4.85°)$. When 30 to 50 POD vectors are used in the basis, the relative error is less than 0.5%. However, this result also illustrates the fact that adding POD vectors in the basis can worsen the precision of the ROM response: when more than 55 POD vectors are used, negative pressures are obtained during the ROM time integration. This emphasizes the strength of the ROB interpolation method which produces POD bases with a fixed number of vectors in the basis, as opposed to the global POD method which would generate very large bases, likely to cause such instabilities in the present case.
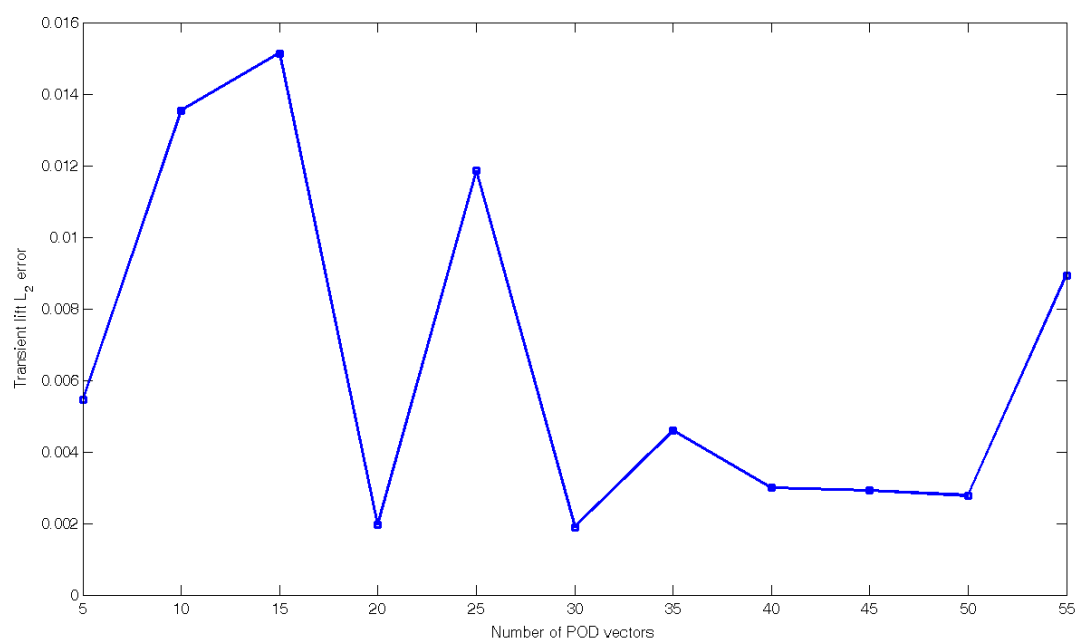
Figure 4.10: Transient lift L_2-norm of the relative error between the high-fidelity linearized full-order model and ROMs computed for various numbers of POD vectors at $(M_\infty, \alpha) = (0.725, 4.85°)$.

4.6.2 Adaptation of F/A-18 ROMs to New Free-Stream Data

POD Basis Interpolation

Four POD bases of dimension 40 are pre-computed at the following free-stream Mach numbers: $M_{\infty_1} = 0.5$, $M_{\infty_2} = 0.7$, $M_{\infty_3} = 0.75$ and $M_{\infty_4} = 0.8$. In all four cases, the angle of attack is set to $\alpha = 4.85°$. Then, each reduced-order POD basis is coupled with the chosen modal representation of dimension 11 of the high-fidelity FE structural model in order to construct at each considered flight condition an aeroelastic ROM of dimension 51.

First, both of the methods of interpolation in a tangent space to a Grassmann manifold and that of direct interpolation of basis vectors (by directly interpolating the entries of the vectors one-by-one and re-orthogonalizing by a Gram-Schmidt procedure) are applied to interpolate the four pre-computed aeroelastic ROMs and generate a new ROM capable of operating at ($M_\infty = 0.725$, $\alpha = 4.85°$). Figure 4.11 reports the lift time-histories obtained when applying the Mach-interpolated aeroelastic ROMs using both interpolation methods to the solution of a given aeroelastic response problem. The reader can observe that the interpolated ROM using the adaptation method proposed in this paper performs well, whereas that obtained by direct interpolation of the POD basis vectors diverges after a few time-steps.

Next, the method of interpolation in a tangent space to a Grassmann manifold is applied to interpolate: (a) two of the POD bases whose operating points are closest to that of the new target flight condition — that is, the POD bases corresponding to $M_{\infty_2} = 0.7$ and $M_{\infty_3} = 0.75$, and (b) all four pre-computed POD bases. Figure 4.12 reports the lift time-histories obtained using both interpolated ROMs, a POD-based aeroelastic ROM of the same dimension that is directly constructed at the new target flight condition, and the linearization at the same target flight condition ($M_\infty = 0.725$
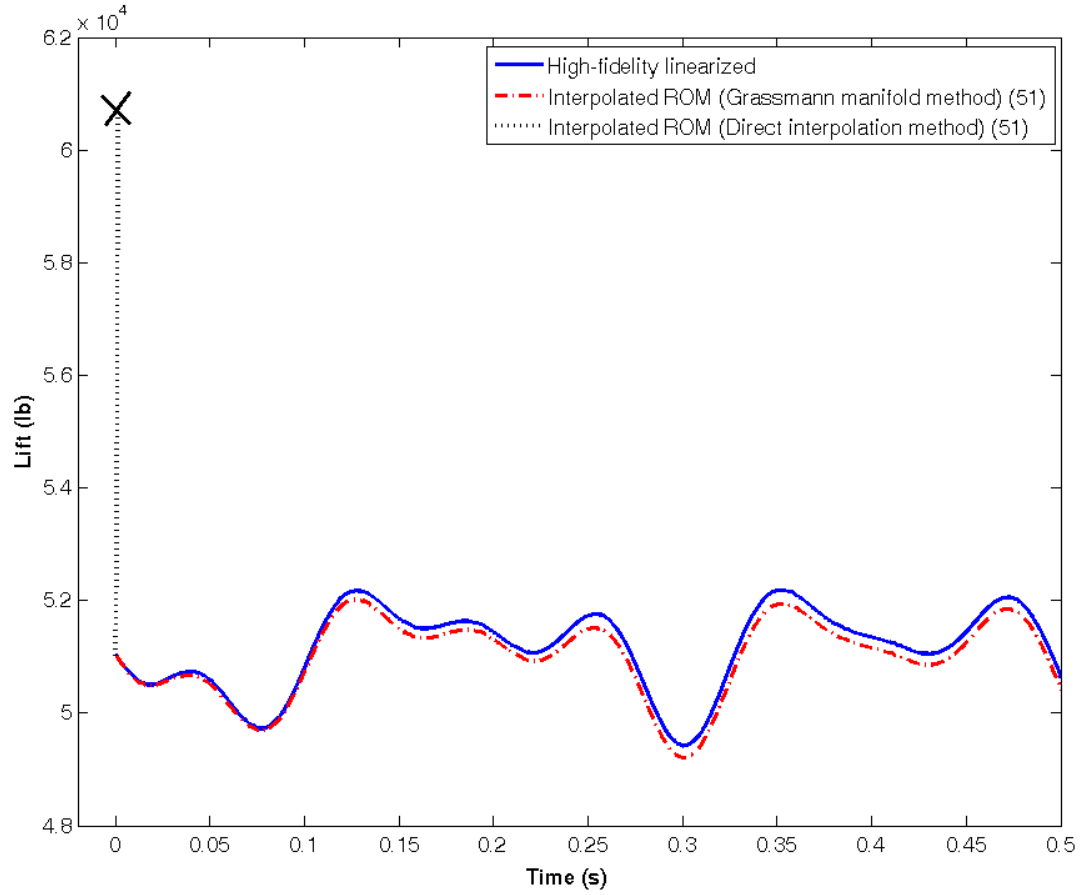
Figure 4.11: Comparison of the lift time-histories of an F/A-18 configuration at $(M_\infty = 0.725, \alpha = 4.85°)$ produced by: (a) the linearization of a high-fidelity aeroelastic model, (b) the ROM obtained by interpolating in a tangent space to a Grassmann manifold, four aeroelastic ROMs pre-computed at $M_{\infty_1} = 0.5$, $M_{\infty_2} = 0.7$, $M_{\infty_3} = 0.75$ and $M_{\infty_4} = 0.8$, (c) the ROM associated with the direct interpolation of the POD bases pre-computed at the same four different Mach numbers.

and $\alpha = 4.85°$) of the high-fidelity nonlinear aeroelastic model. The reader can observe that the aeroelastic ROM obtained by interpolation of all four pre-computed aeroelastic ROMs reproduces almost perfectly (less than 0.4% relative error) the lift time-history produced by the ROM directly constructed at the new target flight condition, and that this lift time-history correlates well with that of the linearized high-fidelity aeroelastic model. The reader can also observe that while the results obtained by interpolating only two of the pre-computed aeroelastic ROMs are less than perfect, they highlight the feasibility of applying adaptively the proposed ROM interpolation method. Indeed, as with most interpolation methods, the proposed ROM adaptation method can be applied to interpolate an increasing number of ROMs that are stored in a database until the target simulation result is converged.

Robustness with Respect to the Choice of Reference Reduced-Order Basis

In the Remark of section 3.5.2, it was stated that the proposed ROM adaptation method is robust with respect to the choice of the reference point where the tangent space to the relevant manifold is constructed, as long as all pre-computed ROMs to be interpolated are chosen in the neighborhood of the reference point. This statement is supported here by one example, namely, the problem of interpolating all four pre-computed F-18/A aeroelastic ROMs introduced in this section in order to construct a new aeroelastic ROM at the operating point ($M_\infty = 0.725$, $\alpha = 4.85°$).

For this purpose, the method of interpolation in a tangent space to the Grassmann manifold is applied four times to interpolate the four aforementioned pre-computed ROMs and produce an aeroelastic ROM that can operate at ($M_\infty = 0.725$, $\alpha = 4.85°$). Each time, one of the pre-computed POD bases is chosen as a different reference point. Then, each of the four different aeroelastic ROMs obtained by interpolation
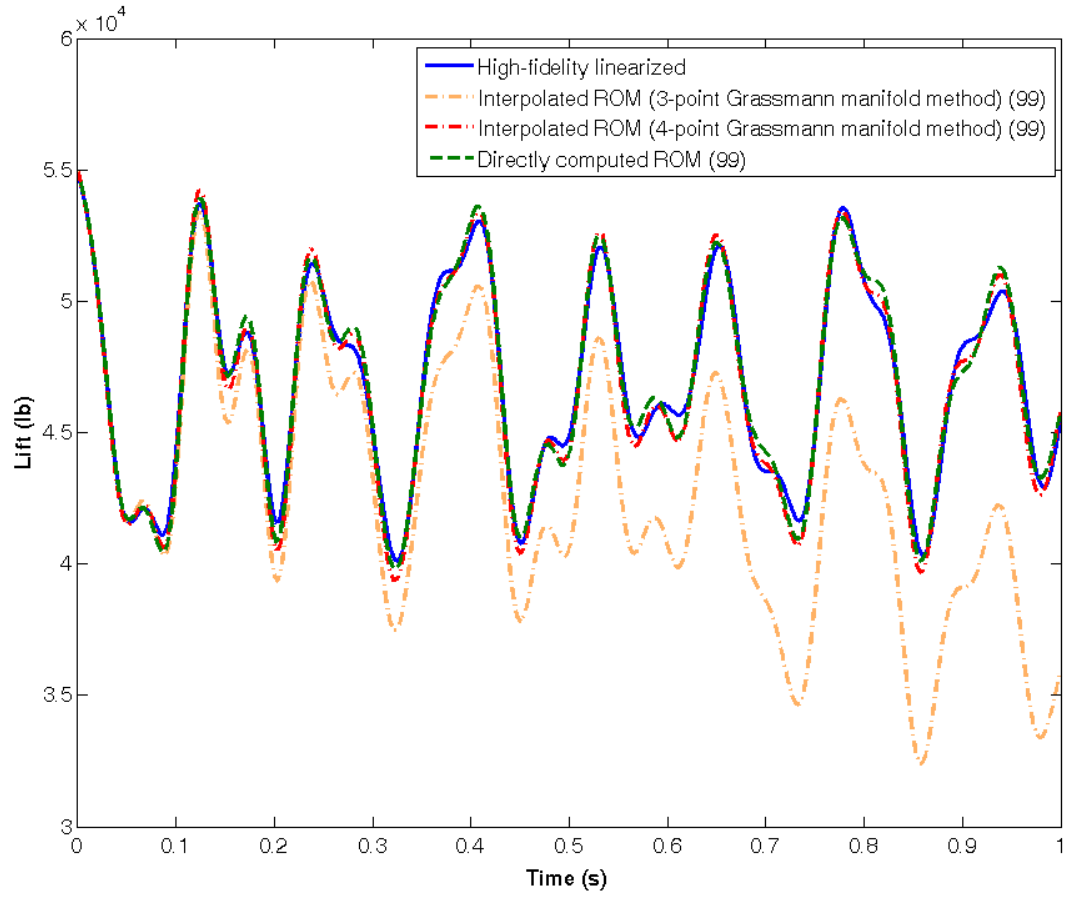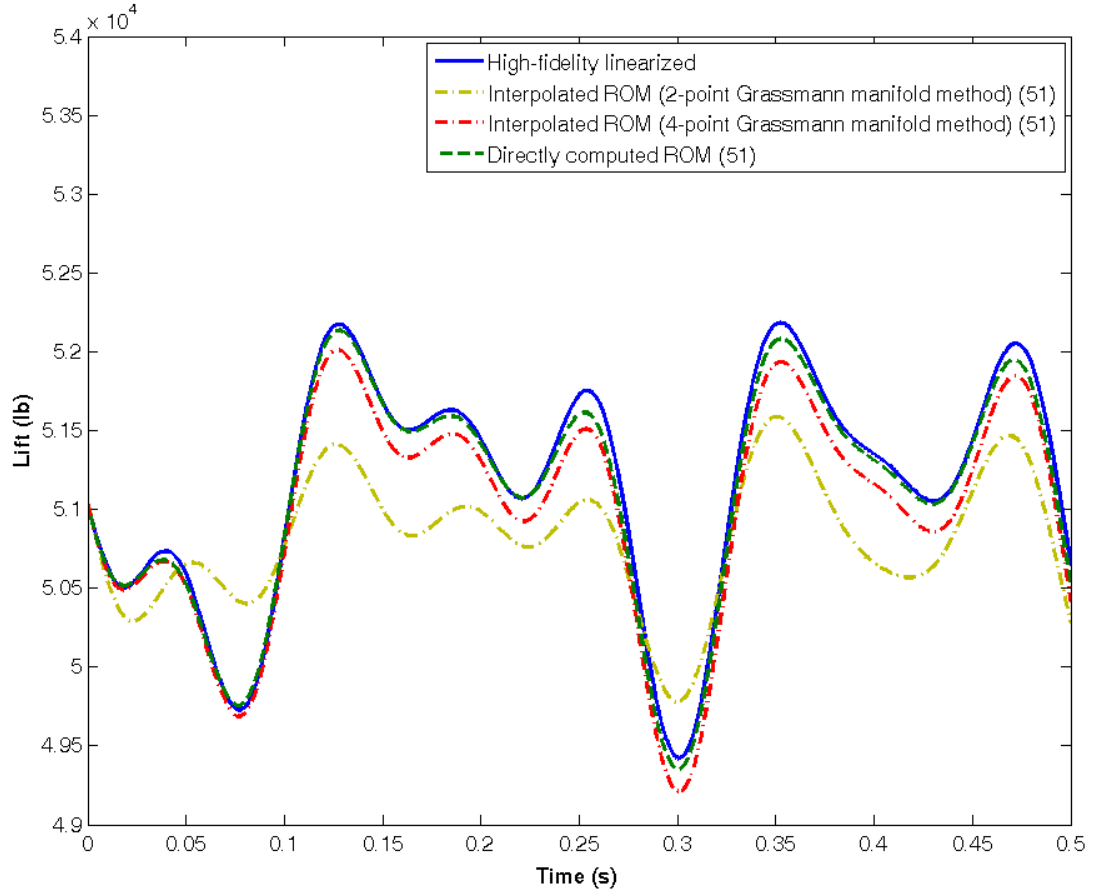
Figure 4.12: Comparison of the lift time-histories of an F/A-18 configuration at $(M_\infty = 0.725, \alpha = 4.85°)$ produced by: (a) a ROM obtained by interpolating in a tangent space to a Grassmann manifold two aeroelastic ROMs pre-computed at $M_{\infty_2} = 0.7$ and $M_{\infty_3} = 0.75$, (b) a ROM obtained by adapting four aeroelastic ROMs pre-computed at $M_{\infty_1} = 0.5$, $M_{\infty_2} = 0.7$, $M_{\infty_3} = 0.75$ and $M_{\infty_4} = 0.8$ using the same interpolation method, (c) an aeroelastic ROM directly constructed at (M_∞, α), and (d) the linearization of a high-fidelity aeroelastic model at (M_∞, α).

is used to solve the same aeroelastic response problem associated with an initial perturbation of an aeroelastic state of equilibrium. The four computed lift time-histories are reported in Figure 4.13 and contrasted with the lift time-history obtained using the linearization at the same operating point of the high-fidelity nonlinear aeroelastic model. The reader can observe that all four lift time-histories are nearly identical. This result supports the statement made about the robustness with respect to the chosen reference point of the proposed method of interpolation in a tangent space to a Grassmann manifold.

Relation to the Subspace Angle Interpolation Method

In section 3.7.5, a proof of the equivalence of the subspace angle interpolation method and the particular case of a two-point Grassmann interpolation method was offered. This proof is illustrated here by the following example. Two POD bases corresponding to $(M_{\infty_1} = 0.7, \alpha = 4.85°)$ and $(M_{\infty_2} = 0.75, \alpha = 4.85°)$ are adapted using both the subspace angle interpolation method and the method of interpolation in a tangent space to a Grassmann manifold, in order to generate a new POD basis for the operating point $(M_\infty = 0.725, \alpha = 4.85°)$. Figure 4.14 shows that the lift-time histories obtained by applying the Mach-interpolated aeroelastic ROMs produced by both interpolation methods to the solution of a given aeroelastic response problem are identical (0% relative error).
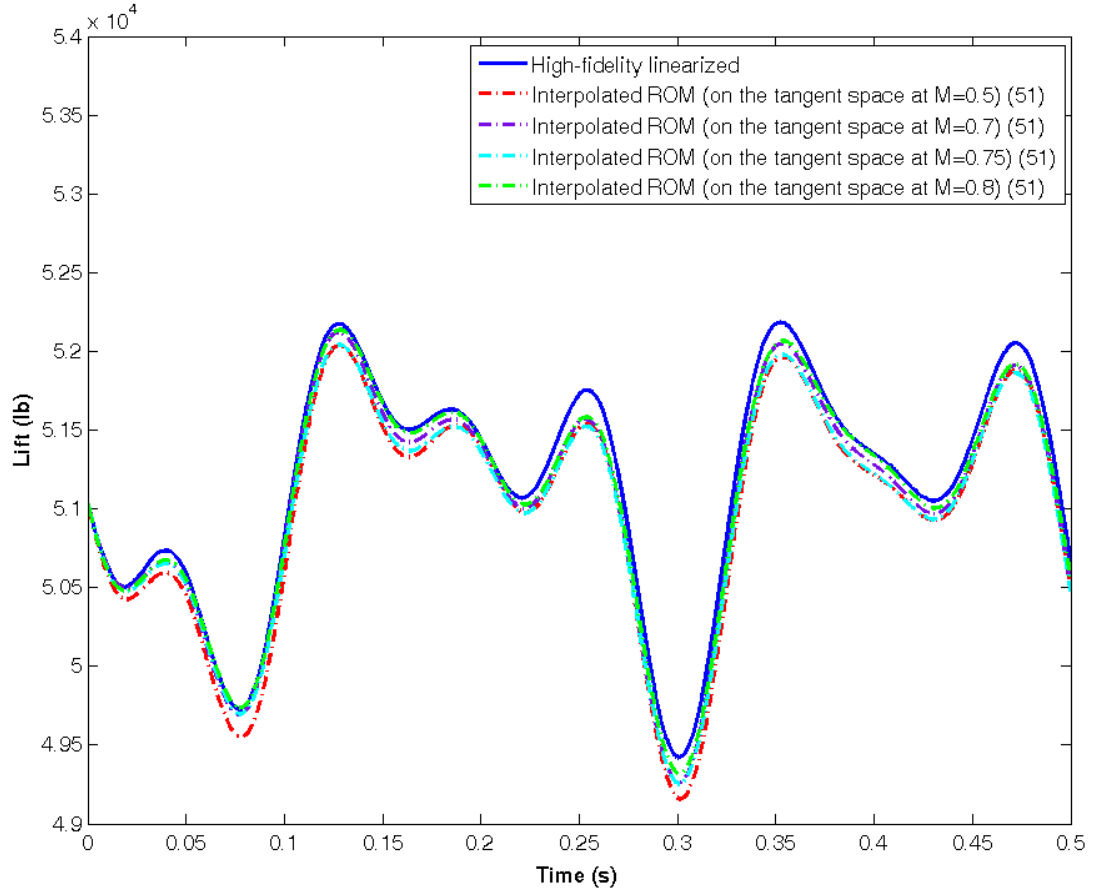
Figure 4.13: Comparison of the F-18/A lift time-histories produced by the four ROMs of dimension 51 obtained by interpolating at ($M_\infty = 0.725$, $\alpha = 4.85°$) the same four pre-computed POD bases but in four different tangent spaces to the same Grassmann manifold.
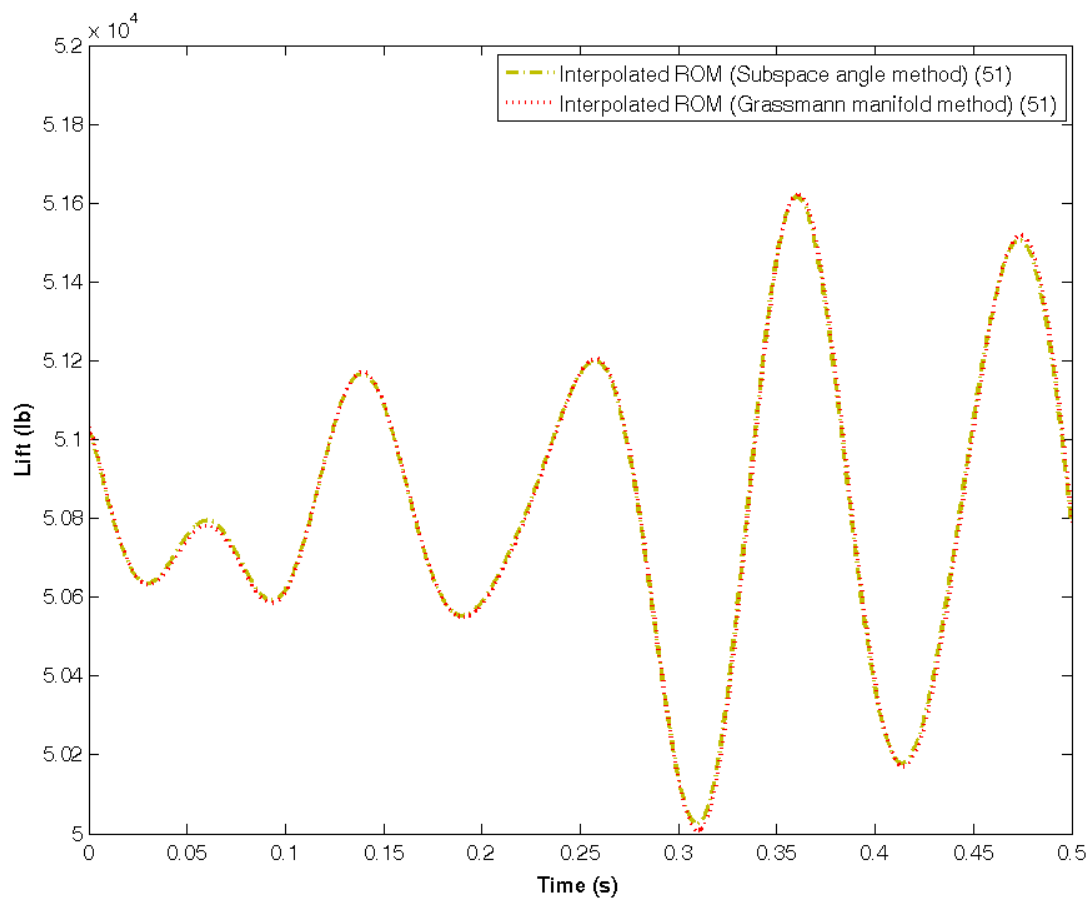
Figure 4.14: Comparison of the lift time-histories produced by the F-18/A aeroelastic ROMs at ($M_\infty = 0.725$, $\alpha = 4.85°$) obtained using both of the subspace and Grassmann interpolation methods.

Chapter 5

Interpolation of Linear Structural Reduced-Order Bases and Models

5.1 Introduction

In the previous chapter, the model reduction of the aeroelastic system was performed separately for the fluid and structural subsystem. This enabled the construction of a modal reduced-order model for the FEM-based structural system that was independent of the flight parameters of the aircraft. As such, variations of these flight conditions did not affect the structural ROM and a unique structural ROM was constructed for each aircraft. However, designers and engineers often explore a large variety of structural configurations in order to select the optimal one. For instance, these configurations can be defined by shape parameters that will have a direct influence on the aerodynamic properties of the structure. Exploring such a variety of configurations calls for smaller and cheaper representations of structural systems, such as ROMs.

The concept of model reduction in structural mechanics precedes its fluid counterpart. It has — and is still — been used for many purposes ranging from the design of a test-analysis model to provide a basis for comparing computational and experimental results, to the alleviation of the computational burden associated with large-scale finite element models. Among the many linear model reduction techniques that have been or remain popular in industry, one can mention Guyan's reduction method [99] and the related superelement dynamic reduction approaches, the IRS dynamic reduction method [100], and of course, the ubiquitous modal reduction method. More recently, the Proper Orthogonal Decomposition method, which can be used to generate a reduced-order model capable of accurately reproducing the dynamics of the underlying full-order model for a given set of input forces, has gained status in the linear structural dynamics community [101, 102, 103].

In all of these applications, structural dynamics ROMs are sought after because of their potential for operating in real-time. For the same reasons mentioned in introduction to this dissertation —non-robustness of the ROMs and expensive cost of generation of new ROMs— there is a pressing need for a fast structural ROM adaptation procedure which can operate on-line and in real-time. Here, on-line characterizes a procedure which does not operate on the full-order model at the origin of a ROM and therefore which avoids repeated manipulations of a large-scale complex simulation software (such as a parallelized Finite Element code for instance) and the associated computational cost. In other terms, such a procedure could be implemented and run on a single processor, using Matlab.

The proposed interpolation technique is developed in section 5.3 and subsequently applied to geometric variations of the AGARD 445.6 wing in section 5.4. Adapting reduced-order models to geometric changes is indeed of great interest in the model

reduction community, as pointed in Ref. [104]: "There are issues raised by electronic systems that are beyond the capabilities of current model reduction techniques. The most critical of these is geometric variations. Basis modes are computed for specific geometry, hence it is not clear how to effectively parameterize the change in basis functions with geometry changes so as to arrive at a reduced-order model that includes geometric variations". The interpolation method proposed in this chapter addresses this issue.

5.2 Problem Formulation

In this chapter, a parameterized linear structural dynamics (full-order) model of size N_s is abstracted as a triplet of the form

$$\big(\mathbf{M}(\boldsymbol{\mu}),\ \mathbf{D}(\boldsymbol{\mu}),\ \mathbf{K}(\boldsymbol{\mu})\big) \in \mathbb{R}^{N_s \times N_s} \times \mathbb{R}^{N_s \times N_s} \times \mathbb{R}^{N_s \times N_s}, \tag{5.1}$$

where $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ are symmetric positive definite mass, damping and stiffness matrices, respectively, and $\boldsymbol{\mu} = (\mu_1,\ \mu_2,\ \dots,\ \mu_{N_p})$ denotes a set of N_p model parameters. These parameters can be physical, non-physical, or a combination of both.

Similarly, a corresponding ROM of size $N_m << N_s$ is abstracted here as a quintuplet of the form

$$R(\boldsymbol{\mu}) = \big(\mathbf{M}_r(\boldsymbol{\mu}),\ \mathbf{D}_r(\boldsymbol{\mu}),\ \mathbf{K}_r(\boldsymbol{\mu}),\ \mathbf{X}(\boldsymbol{\mu}),\ \mathbf{Z}(\boldsymbol{\mu})\big)$$
$$\in \mathbb{R}^{N_m \times N_m} \times \mathbb{R}^{N_m \times N_m} \times \mathbb{R}^{n \times n} \times \mathbb{R}^{N_s \times N_m} \times \mathbb{R}^{N_s \times N_m}, \tag{5.2}$$

where
$$\mathbf{M}_r(\boldsymbol{\mu}) = \mathbf{X}^T(\boldsymbol{\mu})\mathbf{M}(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu}), \quad \mathbf{D}_r(\boldsymbol{\mu}) = \mathbf{X}^T(\boldsymbol{\mu})\mathbf{D}(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu}),$$
$$\mathbf{K}_r(\boldsymbol{\mu}) = \mathbf{X}^T(\boldsymbol{\mu})\mathbf{K}(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu}), \qquad \mathbf{Z}(\boldsymbol{\mu}) = \mathbf{A}(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu}), \tag{5.3}$$

$\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_s \times N_s}$ is a real symmetric positive definite matrix, $\mathbf{X}(\boldsymbol{\mu})$ denotes a projection matrix relating the full- and reduced-order displacement vectors $\mathbf{u} \in \mathbb{R}^{N_s}$ and $\mathbf{q} \in \mathbb{R}^{N_m}$ via

$$\mathbf{u}(t; \boldsymbol{\mu}) = \mathbf{X}(\boldsymbol{\mu})\mathbf{q}(t; \boldsymbol{\mu}) \tag{5.4}$$

and satisfying the orthogonality condition

$$\mathbf{X}^T(\boldsymbol{\mu})\mathbf{Z}(\boldsymbol{\mu}) = \mathbf{I}_{N_m}. \tag{5.5}$$

Indeed, the governing equations associated with a linear structural dynamics ROM can be written in general as

$$\mathbf{M}_r(\boldsymbol{\mu})\ddot{\mathbf{q}}(t; \boldsymbol{\mu}) + \mathbf{D}_r(\boldsymbol{\mu})\dot{\mathbf{q}}(t; \boldsymbol{\mu}) + \mathbf{K}_r(\boldsymbol{\mu})\mathbf{q}(t; \boldsymbol{\mu}) = \mathbf{F}_r(t; \boldsymbol{\mu}). \tag{5.6}$$

The present case covers most linear ROMs constructed by popular projection techniques where $\mathbf{X}(\boldsymbol{\mu}) \in \mathbb{R}^{N_s \times N_m}$ is a real rectangular matrix whose columns form a reduced-order basis and $\mathbf{A}(\boldsymbol{\mu})$ is associated with a metric. In this case, $\mathbf{F}_r(t; \boldsymbol{\mu}) = \mathbf{X}^T(\boldsymbol{\mu})\mathbf{F}(t)$ and $\mathbf{X}(\boldsymbol{\mu})$ satisfies an orthogonality constraint. For example, when $\mathbf{X}(\boldsymbol{\mu})$ is generated by the POD method, this matrix satisfies $\mathbf{X}^T(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu}) = \mathbf{I}_{N_m}$ and therefore $\mathbf{A}(\boldsymbol{\mu}) = \mathbf{I}_{N_s}$. Alternatively, $\mathbf{X}(\boldsymbol{\mu})$ can be a set of eigenvectors of the pencil $\big(\mathbf{M}(\boldsymbol{\mu}), \ \mathbf{K}(\boldsymbol{\mu})\big)$, in which case $\mathbf{A}(\boldsymbol{\mu}) = \mathbf{M}(\boldsymbol{\mu})$ and therefore the ROM is essentially a truncated modal representation of the structure. In all cases, $\mathbf{M}_r$, $\mathbf{D}_r$ and $\mathbf{K}_r$ are symmetric positive definite and therefore belong to the manifold $\text{SPD}(N_m)$.

Using the above nomenclature, the focus of this chapter is on solving the following problem.

Problem. Let $\boldsymbol{\mu}^{(i)} = \big(\mu_1^{(i)}, \mu_2^{(i)}, \ldots, \mu_{N_p}^{(i)}\big)$ denote a specific configuration of the set

of N_p parameters $\boldsymbol{\mu}$. In the remainder of this chapter, $\boldsymbol{\mu}^{(i)}$ is referred to as the i-th point of a set of operating points

$$S = \left(\boldsymbol{\mu}^{(1)},\, \boldsymbol{\mu}^{(2)},\, \ldots,\, \boldsymbol{\mu}^{(N_R)}\right). \tag{5.7}$$

Let also $\{R_i\}_{i=1}^{N_R} = \left\{R\left(\boldsymbol{\mu}^{(i)}\right)\right\}_{i=1}^{N_R}$ denote a set of N_R linear structural dynamics ROMs of the same dimension n constructed at the operating points $\left\{\boldsymbol{\mu}^{(i)}\right\}_{i=1}^{N_R}$. Given a new operating point $\boldsymbol{\mu}^{(N_R+1)} \notin S$, compute on-line and in real-time $R_{N_R+1} = R\left(\boldsymbol{\mu}^{(N_R+1)}\right)$.

5.3   ROM Adaptation Methods

Two related ROM adaptation methods are presented here. Both methods share the concept of interpolation in a tangent space to a manifold. Unlike any straightforward interpolation scheme, this concept enables all three methods to produce for any new operating point $\boldsymbol{\mu}^{(N_R+1)}$ a result $R\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ that is a genuine ROM — that is, a result R_{N_R+1} whose matrices $\mathbf{M}_r\left(\boldsymbol{\mu}^{(N_R+1)}\right)$, $\mathbf{D}_r\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ and $\mathbf{K}_r\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ are symmetric positive definite and whose matrix $\mathbf{X}\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ satisfies the constraints defined in Eq. (5.5). All of these methods are based on the approach presented in section 3.5 which can be summarized as follows: first, the data to be interpolated is mapped appropriately onto a tangent space to the appropriate manifold, then the mapped data is interpolated in this space and finally the interpolation result is mapped back to the same manifold (see Figure 3.3).

5.3.1   Reduced Operators Interpolation

Here, $\mathcal{M} = \mathcal{M}' = \mathrm{SPD}(N_m)$ as the triplets of interest

$$\left(\mathbf{M}_r(\boldsymbol{\mu}^{(i)}),\ \mathbf{D}_r(\boldsymbol{\mu}^{(i)}),\ \mathbf{K}_r(\boldsymbol{\mu}^{(i)})\right) \in \left(\mathbb{R}^{N_m \times N_m},\ \mathbb{R}^{N_m \times N_m},\ \mathbb{R}^{N_m \times N_m}\right),\quad i = 1, \cdots, N_R \tag{5.8}$$

are constituted of symmetric positive definite matrices. In this case, the matrix logarithm and exponential mappings are given by

$$\mathrm{Log}_{\mathbf{Y}_{i_0}}(\mathbf{Y}_i) = \texttt{logm}\left(\mathbf{Y}_{i_0}^{-1/2}\mathbf{Y}_i\mathbf{Y}_{i_0}^{-1/2}\right) \tag{5.9}$$

and

$$\mathrm{Exp}_{\mathbf{Y}_{i_0}}(\boldsymbol{\Gamma}) = \mathbf{Y}_{i_0}^{1/2}\texttt{expm}(\boldsymbol{\Gamma})\mathbf{Y}_{i_0}^{1/2}, \tag{5.10}$$

where $\texttt{logm}$ and $\texttt{expm}$ denote the matrix logarithm and exponential, respectively.

Remark. In the particular case where the interpolation data is based on the classical modal decomposition and truncation method, $\mathbf{M}_r\left(\boldsymbol{\mu}^{(i)}\right) = \mathbf{I}_{N_m}$ and $\mathbf{K}_r\left(\boldsymbol{\mu}^{(i)}\right) = \boldsymbol{\Omega}^2_{N_m}\left(\boldsymbol{\mu}^{(i)}\right)$, where $\boldsymbol{\Omega}^2_n\left(\boldsymbol{\mu}^{(i)}\right)$ is a diagonal matrix of squares of N_m natural circular frequencies. In this case, the generalized interpolation method described above preserves the structure of the interpolation triplets $\left(\mathbf{I}_{N_m},\ \mathbf{D}_r\left(\boldsymbol{\mu}^{(i)}\right),\ \boldsymbol{\Omega}^2_{N_m}\left(\boldsymbol{\mu}^{(i)}\right)\right)$ as it generates a pair of reduced-order matrices $\mathbf{M}_r\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ and $\mathbf{K}_r\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ that satisfy $\mathbf{M}_r\left(\boldsymbol{\mu}^{(N_R+1)}\right) = \mathbf{I}_{N_m}$ and $\mathbf{K}_r\left(\boldsymbol{\mu}^{(N_R+1)}\right) = \boldsymbol{\Omega}^2_{N_m}$, where $\boldsymbol{\Omega}^2_{N_m}$ is a diagonal matrix with N_m positive entries. This important property of the proposed generalized interpolation method in a tangent space to a matrix manifold is proven in section 3.6.3 in the general case $\mathbf{Q} = \mathbf{I}_{N_m}$.

5.3.2   Subspaces Interpolation

Here, the doublets of interest

$$\left(\mathbf{X}(\boldsymbol{\mu}^{(i)}),\ \mathbf{Z}(\boldsymbol{\mu}^{(i)})\right) \in \left(\mathbb{R}^{N_m \times N_m},\ \mathbb{R}^{N_m \times N_m},\ \mathbb{R}^{N_m \times N_m},\ \mathbb{R}^{N_s \times N_m},\ \mathbb{R}^{N_s \times N_m}\right), \quad (5.11)$$

for $i = 1, \cdots, N_R$, are constituted of two rectangular matrices representing two reduced-order bases and satisfying the constraint (5.5). The two subspaces spanned by the columns of $\mathbf{X}(\boldsymbol{\mu}^{(i)})$ and $\mathbf{Z}(\boldsymbol{\mu}^{(i)})$ belong to the Grassmann manifold $\mathcal{G}(N_m, N_s)$.

The computation of the two matrices $\mathbf{X}\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ and $\mathbf{Z}\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ requires the generalized interpolation of two different sets of matrices $\left\{\mathbf{X}(\boldsymbol{\mu}^{(i)})\right\}$ and $\left\{\mathbf{Z}(\boldsymbol{\mu}^{(i)})\right\}$ that are however connected by the orthogonality condition (5.5). Hence, this computation cannot be performed using the generalized interpolation method presented in section 3.5.2 as is. Instead, it is proposed to simultaneously interpolate the set of matrices $\mathbf{X}\left(\boldsymbol{\mu}^{(i)}\right)$ and set of matrices $\mathbf{Z}\left(\boldsymbol{\mu}^{(i)}\right)$ column-block per column-block while enforcing the orthogonality constraint (5.5). For this purpose, two sub-cases are distinguished: the sub-case where $\mathbf{X}\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ is associated with the classical modal decomposition and truncation method and that where $\mathbf{X}\left(\boldsymbol{\mu}^{(N_R+1)}\right)$ is associated with an arbitrary Galerkin projection method.

Modal Truncation

The modal truncation approach differs from the general Galerkin projection approach in that each individual column of a matrix $\mathbf{X}(\boldsymbol{\mu}^{(i)})$ has a specific meaning and importance that must be preserved during the interpolation process. More specifically, if the columns of the matrices $\left\{\mathbf{X}(\boldsymbol{\mu}^{(i)})\right\}$ to be interpolated are ordered so that the j-th column of each of them refers to the same eigenmode, then the j-th column of

the interpolated matrix $\mathbf{X}\big(\boldsymbol{\mu}^{(N_R+1)}\big)$ must refer to the same eigenmode. As such, the present approach enables the handling of mode crossing phenomena. Therefore, the generalized interpolation method proposed here loops on the eigensubspaces $\big\{\mathcal{S}_{ij}^X\big\}$ and subspaces $\big\{\mathcal{S}_{ij}^Z\big\}$ (where i refers to $\boldsymbol{\mu}^{(i)}$) of the parameterized system underlying all matrices $\big\{\mathbf{X}\big(\boldsymbol{\mu}^{(i)}\big)\big\}$ and $\big\{\mathbf{Z}\big(\boldsymbol{\mu}^{(i)}\big)\big\}$ and interpolates each of such set of matrices while enforcing the orthogonality constraint (5.5) as described below. For clarity, the proposed generalized interpolation method is first described in the simple case where each eigensubspace is of dimension 1. In this case, $\mathcal{S}_{ij}^X \in \mathcal{G}(1, N_s)$ and $\mathcal{S}_{ij}^Z \in \mathcal{G}(1, N_s)$.

For $j = 1, \cdots, N_m$

- Step 0. Interpolate the eigensubspaces $\big\{\mathcal{S}_{ij}^X \in \mathcal{G}(1, N_s)\big\}$ using the generalized interpolation algorithm described in section 3.5 with $\mathcal{M} = \mathcal{G}(1, N_s)$ and $\mathcal{M}'$ the non-compact Stiefel manifold $\mathbb{R}^*_{N_s \times 1}$ of non zero vectors of size N_s. The matrix logarithm and exponential mappings associated with $\mathcal{G}(1, N_s)$ are given by

$$\left(\mathbf{I}_{N_s} - \mathbf{X}_{i_0 j}\left(\mathbf{X}_{i_0 j}^T \mathbf{X}_{i_0 j}\right)^{-1} \mathbf{X}_{i_0 j}^T\right) \mathbf{X}_{ij}\left(\mathbf{X}_{i_0 j}^T \mathbf{X}_{ij}\right)^{-1}\left(\mathbf{X}_{i_0 j}^T \mathbf{X}_{i_0 j}\right)^{\frac{1}{2}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

$$\text{(Thin SVD)} \qquad\qquad (5.12)$$

$$\text{Log}_{\mathcal{S}_{i_0 j}}(\mathcal{S}_{ij}) \leftrightarrow \mathbf{U}\mathbf{tan}^{-1}(\boldsymbol{\Sigma})\mathbf{V}^T$$

and

$$\boldsymbol{\Gamma} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \qquad \text{(Thin SVD)}$$

$$(5.13)$$

$$\text{Exp}_{\mathcal{S}_{i_0 j}}(\chi) = \text{span}\left\{\mathbf{X}_{i_0 j}\left(\mathbf{X}_{i_0 j}^T \mathbf{X}_{i_0 j}\right)^{-\frac{1}{2}}\mathbf{V}\mathbf{cos}(\boldsymbol{\Sigma}) + \mathbf{U}\mathbf{sin}(\boldsymbol{\Sigma})\right\},$$

where $\mathbf{X}_{ij}$ denotes the j-th column of $\mathbf{X}\big(\boldsymbol{\mu}^{(i)}\big)$.

- Step 1. Perform a Gram-Schmidt procedure on $\mathbf{X}_{N_R+1 j}$ to enforce the orthogonality conditions $\mathbf{X}_{N_R+1 j}^T \mathbf{Z}_{N_R+1 l} = 0$, $l = 1, \cdots, j - 1$, where $\mathbf{Z}_{ij}$ denotes the

j-th column of $\mathbf{Z}\big(\boldsymbol{\mu}^{(i)}\big)$.

- Step 2. Interpolate the subspaces $\big\{\mathcal{S}_{ij}^Z \in \mathcal{G}(1, N_s)\big\}$ using the generalized interpolation algorithm described in section 3.5 with $\mathcal{M} = \mathcal{G}(1, N_s)$ and $\mathcal{M}'$ the non compact Stiefel manifold $\mathbb{R}^*_{N_s \times 1}$

- Step 4. Perform a Gram-Schmidt procedure on $\mathbf{Z}_{N_R+1j}$ to enforce the orthogonality conditions $\mathbf{X}_{N_R+1l}^T \mathbf{Z}_{N_R+1j} = 0, \;\; l = 1, \cdots, j-1$, and $\mathbf{X}_{N_R+1j}^T \mathbf{Z}_{N_R+1j} = 1$.

- Step 5. Scale $\mathbf{X}_{N_R+1j}$ and $\mathbf{Z}_{N_R+1j}$ so that their two-norms are comparable to the two-norms of $\{\mathbf{X}_{ij}\}$ and $\{\mathbf{Z}_{ij}\}$, respectively.

The extension of the above generalized interpolation algorithm to the case where each eigensubspace $\{\mathcal{S}_{ij}^X\}$ is of the same dimension $k_j > 1$ is straightforward by interpolating subspaces in $\mathcal{G}(k_j, N_s)$.

Arbitrary Galerkin Projection

In this case, the proposed generalized interpolation method is identical to that described in the previous section for $k_j = 1$, assuming that the reduced-order bases can be ordered in an identical fashion across configurations. Again, such an ordering solves issues associated with mode crossing.

5.4 Application

Here, the proposed interpolation method is illustrated with a simple application that highlight its potential for adapting on-line a structural dynamics ROM to a new operating point.

$c_{root}^{(min)}$	$c_{root}^{(max)}$	$c_{tip}^{(min)}$	$c_{tip}^{(max)}$	$\bar{s}^{(min)}/2$	$\bar{s}^{(max)}/2$	$\Lambda_{c/4}^{(min)}$	$\Lambda_{c/4}^{(max)}$
17.568 in	26.352 in	11.6 in	17.4 in	24 in	36 in	38.66°	50.16°

Table 5.1: AGARD Wing 445.6: bounds of the parametric domain of interest.

The AGARD Wing 445.6 [105] is considered here and represented by an undamped $(\mathbf{D} = \mathbf{0}_{N_s})$ finite element (FE) model composed of 800 shell elements that generate $N_s = 2646$ degrees of freedom. The geometry of the wing is parameterized by four shape parameters as shown in Figure 5.1: the root chord c_{root}, the tip chord c_{tip}, the half span length $\dfrac{\bar{s}}{2}$ and the quarter-chord sweep angle $\Lambda_{c/4}$.

A database of linear structural dynamics ROMs is constructed for this wing by precomputing a set of 17 quintuplets

$$\left\{ R(\boldsymbol{\mu}^{(i)}) = \left(\mathbf{M}_r(\boldsymbol{\mu}^{(i)}),\ \mathbf{D}_r(\boldsymbol{\mu}^{(i)}),\ \mathbf{K}_r(\boldsymbol{\mu}^{(i)}),\ \mathbf{X}(\boldsymbol{\mu}^{(i)}),\ \mathbf{Z}(\mathbf{s}^{(i)}) \right) \right\}_{i=1}^{17} \tag{5.14}$$

for 17 different design points $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^{17}$ that can be viewed as the vertices and center of a hypercube in a design space — that is, a subset of $\mathbb{R}^4$ — defined by

$$\begin{aligned} &\left(c_{root}, c_{tip}, \frac{\bar{s}}{2}, \Lambda_{c/4} \right) \in \\ &\left[c_{root}^{(min)}, c_{root}^{(max)} \right] \times \left[c_{tip}^{(min)}, c_{tip}^{(max)} \right] \times \left[\frac{\bar{s}^{(min)}}{2}, \frac{\bar{s}^{(max)}}{2} \right] \times \left[\Lambda_{c/4}^{(min)}, \Lambda_{c/4}^{(max)} \right], \end{aligned} \tag{5.15}$$

where the values of the upper and lower bounds are defined in Table 5.1.

For each of these 17 sample design points, a ROM is constructed by the method of modal decomposition and truncation based on the first five natural modes of this design. Hence, in this case,

$$\left\{ R(\boldsymbol{\mu}^{(i)}) = \left(\mathbf{I}_5,\ \mathbf{0}_{5,5},\ \boldsymbol{\Omega}_5^2(\boldsymbol{\mu}^{(i)}),\ \mathbf{X}(\boldsymbol{\mu}^{(i)}),\ \mathbf{Z}(\boldsymbol{\mu}^{(i)}) \right) \right\}_{i=1}^{17}, \tag{5.16}$$

where $\boldsymbol{\Omega}_5^2(\boldsymbol{\mu})$ denotes the diagonal matrix storing the squares of the first five natural circular frequencies of the structural model of the wing for the design point $\boldsymbol{\mu}$.

Here, the accuracy of the generalized interpolation method proposed in this chapter is assessed for three arbitrary design points (or configurations), $\boldsymbol{\mu}^{(a)}$, $\boldsymbol{\mu}^{(b)}$ and $\boldsymbol{\mu}^{(c)}$. These design configurations are shown in Figure 5.2, specified in Table 5.2 and referred to in the remainder of this section as "test" design points for the proposed generalized interpolation method. In particular, these test design points are sufficiently "far" from the other design points for which reduced-order data is precomputed and stored in the database. Indeed, Table 5.2 reports for each parameter of each test design configuration the distance $\delta_\mu(\boldsymbol{\mu}^{(l)})$, $l = a$, b, c, to the precomputed design points defined as

$$\delta_\mu(\boldsymbol{\mu}^{(l)}) = \min_{1 \leq j \leq 17} \frac{\left| \mu(\boldsymbol{\mu}^{(l)}) - \mu(\boldsymbol{\mu}^{(j)}) \right|}{\left| \max_{1 \leq i \leq 17} \mu(\boldsymbol{\mu}^{(i)}) - \min_{1 \leq i \leq 17} \mu(\boldsymbol{\mu}^{(i)}) \right|} \quad . \tag{5.17}$$

More specifically, the 17 precomputed quintuplets of reduced-order matrices and mode shape vectors (5.16) associated with the 17 precomputed design points are interpolated to generate three similar quintuplets of reduced-order matrices and mode shape vectors associated with the three design points $\boldsymbol{\mu}^{(a)}$, $\boldsymbol{\mu}^{(b)}$ and $\boldsymbol{\mu}^{(c)}$, respectively. To this effect, the reader is reminded that the proposed generalized interpolation method preserves the structure of a diagonal matrix (see section 3.6.3). Hence, it is guaranteed to deliver quintuplets of the form given in (5.16) when applied to precomputed quintuplets of the same form and as such is a viable alternative approach to the straightforward interpolation of scalar natural frequencies.

The accuracy of the interpolation is first assessed by comparing the natural frequencies and mode shapes of each interpolated linear structural dynamics ROM with
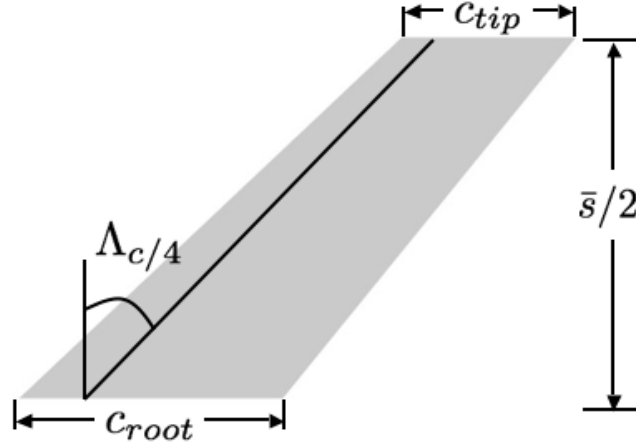
Figure 5.1: Geometrical parameterization of the AGARD Wing 445.6.

	c_{root}	c_{tip}	$\bar{s}/2$	$\Lambda_{c/4}$
$\boldsymbol{\mu}^{(a)}$	25.484 in	15.715 in	32.852 in	42.21°
	$\delta_{c_{root}}\big(\boldsymbol{\mu}^{(a)}\big) = 0.099$	$\delta_{c_{tip}}\big(\boldsymbol{\mu}^{(a)}\big) = 0.21$	$\delta_{\bar{s}/2}\big(\boldsymbol{\mu}^{(a)}\big) = 0.24$	$\delta_{\Lambda_{c/4}}\big(\boldsymbol{\mu}^{(a)}\big) = 0.19$
$\boldsymbol{\mu}^{(b)}$	23.414 in	13.549 in	24.200 in	47.76°
	$\delta_{c_{root}}\big(\boldsymbol{\mu}^{(b)}\big) = 0.17$	$\delta_{c_{tip}}\big(\boldsymbol{\mu}^{(b)}\big) = 0.16$	$\delta_{\bar{s}/2}\big(\boldsymbol{\mu}^{(b)}\big) = 0.02$	$\delta_{\Lambda_{c/4}}\big(\boldsymbol{\mu}^{(b)}\big) = 0.21$
$\boldsymbol{\mu}^{(c)}$	22.056 in	17.398 in	26.915 in	42.31°
	$\delta_{c_{root}}\big(\boldsymbol{\mu}^{(c)}\big) = 0.01$	$\delta_{c_{tip}}\big(\boldsymbol{\mu}^{(c)}\big) = 3 \times 10^{-4}$	$\delta_{\bar{s}/2}\big(\boldsymbol{\mu}^{(c)}\big) = 0.24$	$\delta_{\Lambda_{c/4}}\big(\boldsymbol{\mu}^{(c)}\big) = 0.18$

Table 5.2: "Test" design points.

those of its directly computed counterpart. For the mode shape vectors, the comparison is performed using the Modal Assurance Criterion (MAC) [106]. To this effect, the results reported in Table 5.3 and Table 5.4 reveal that the eigen characteristics of the interpolated ROMs are in good agreement (within 5%) with those of the directly computed ROMs, thereby illustrating the accuracy of the proposed generalized interpolation method.

Next, the accuracy of the proposed generalized interpolation method is assessed by computing the dynamic response of each test design point of the wing to a sudden and
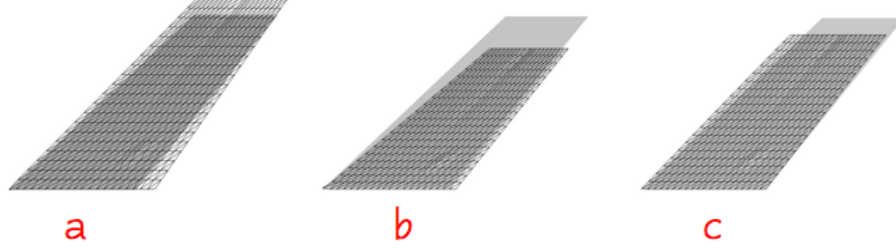
Figure 5.2: "Test" design points: shaded geometry corresponds to the wing configuration for the values of the shape parameters at the center of the hypercube and geometry shown in wireframe corresponds to the "test" wing configuration.

Mode	Test design point (a)			Test design point (b)			Test design point (c)		
	Direct ROM	Interp. ROM	Rel. discrep.	Direct ROM	Interp. ROM	Rel. discrep.	Direct ROM	Interp. ROM	Rel. discrep.
1	9.11	9.00	1.2 %	14.5	14.8	2.1 %	12.3	12.8	4.1 %
2	35.4	35.1	0.8 %	51.7	53.4	3.3 %	41.2	42.5	3.2 %
3	44.7	43.9	1.8 %	72.8	73.8	1.4 %	61.9	63.9	3.2 %
4	88.6	87.7	1.0 %	128	131	2.7 %	107	110	2.8 %
5	117	117	0.0 %	186	186	0.0 %	158	164	3.7 %

Table 5.3: Comparison of the first five natural frequencies (in Hz) of the three "test" design points delivered by the generalized interpolation method with their counterparts obtained from direct ROM constructions.

Mode	Test design point (a)	Test design point (b)	Test design point (c)
1	1.0000	1.0000	1.0000
2	0.9999	0.9994	0.9985
3	0.9999	0.9998	0.9993
4	0.9995	0.9995	0.9986
5	0.9702	0.9504	0.9962

Table 5.4: Modal Assurance Criterion (MAC) applied to the interpolated mode shapes and their counterparts obtained from direct ROM constructions.

Test design point (a)		Test design point (b)		Test design point (c)	
Direct ROM	Interp. ROM	Direct ROM	Interp. ROM	Direct ROM	Interp. ROM
1.87%	8.67%	2.77%	7.77%	3.30%	7.98%

Table 5.5: Relative discrepancies between the maximum amplitude of the vertical displacement at the trailing edge tip predicted by the complete FEM, the directly computed ROMs and the interpolated ROMs.

uniformly distributed vertical load. For this purpose, three simulations are performed for each test design configuration using: (a) the full-order FEM dynamic model, (b) a ROM counterpart directly built from the first five natural modes of this model and (c) the ROM counterpart computed by interpolating the 17 quintuplets stored in the ROM database. In all cases, the governing equations of dynamic equilibrium are time-integrated by the trapezoidal rule. Figure 5.3, Figure 5.4 and Figure 5.5 report for the test design points $\boldsymbol{\mu}^{(a)}$, $\boldsymbol{\mu}^{(b)}$ and $\boldsymbol{\mu}^{(c)}$, respectively, the computed time-histories of the vertical displacement at the trailing edge tip point of the wing using each of the three different computational models. The reader can observe that in each case, the interpolated ROM delivers good accuracy. Indeed, Table 5.5 shows that in each case, the relative discrepancy between the maximum amplitudes of the dynamic responses predicted by the full-order FEM and interpolated ROM is of the order of 8% only. For comparison, the relative discrepancy between the maximum amplitudes of the dynamic responses predicted by the full-order FEM and directly computed ROM is of the order of 3%.
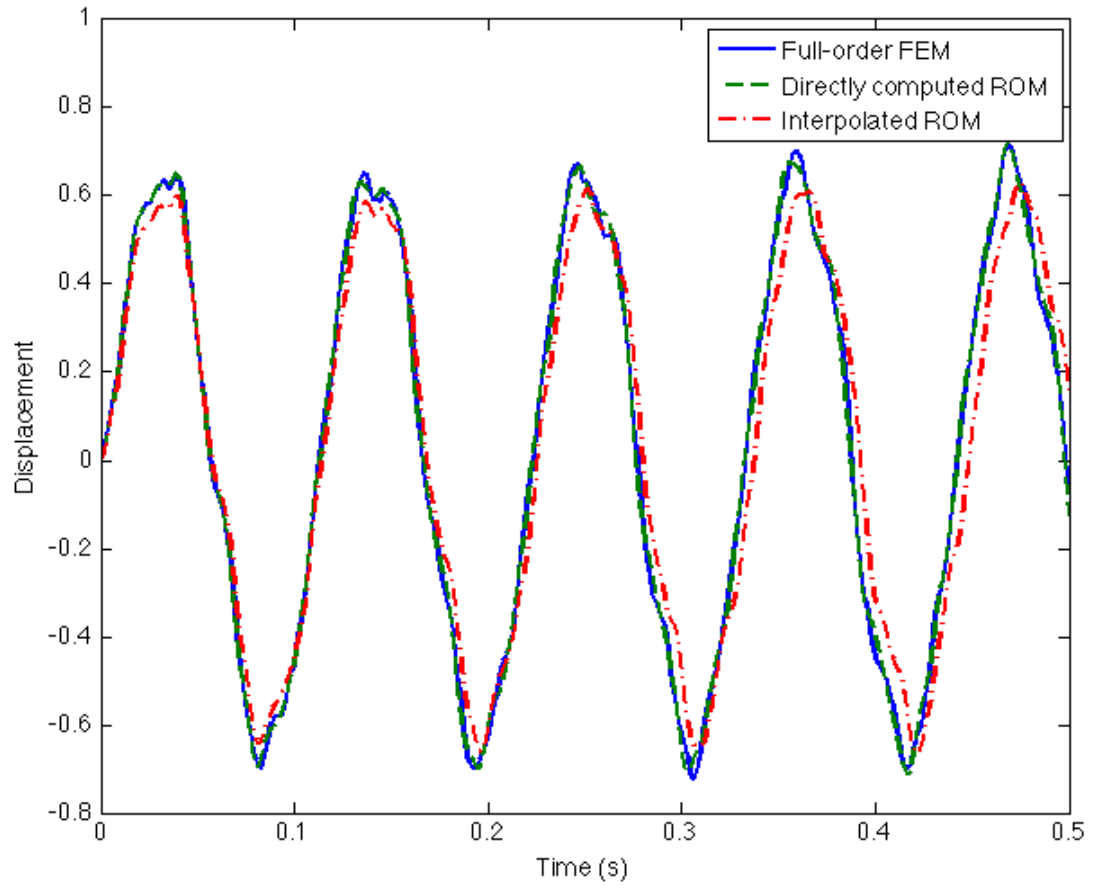
Figure 5.3: Test design point $\boldsymbol{\mu}^{(a)}$: transient responses predicted by the complete FEM model, the directly computed ROM and the interpolated ROM.
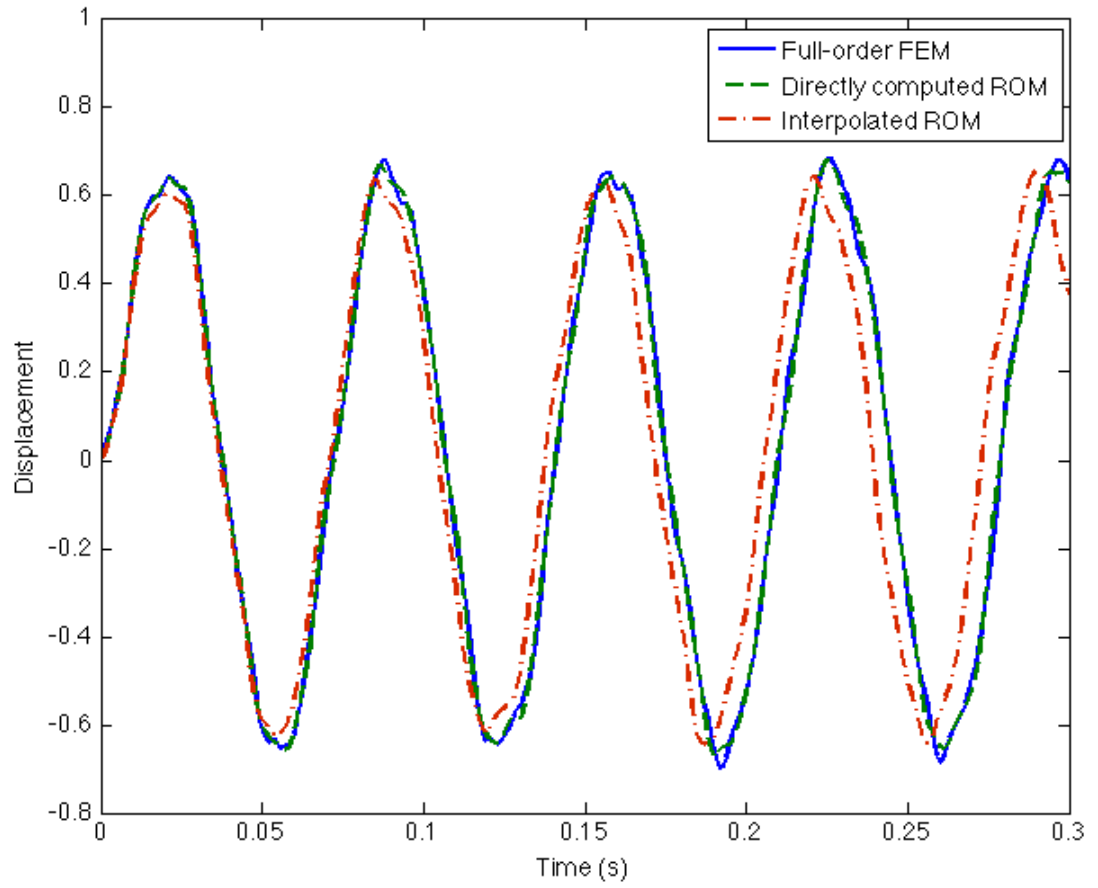
Figure 5.4: Test design point $\boldsymbol{\mu}^{(b)}$: transient responses predicted by the complete FEM model, the directly computed ROM and the interpolated ROM.
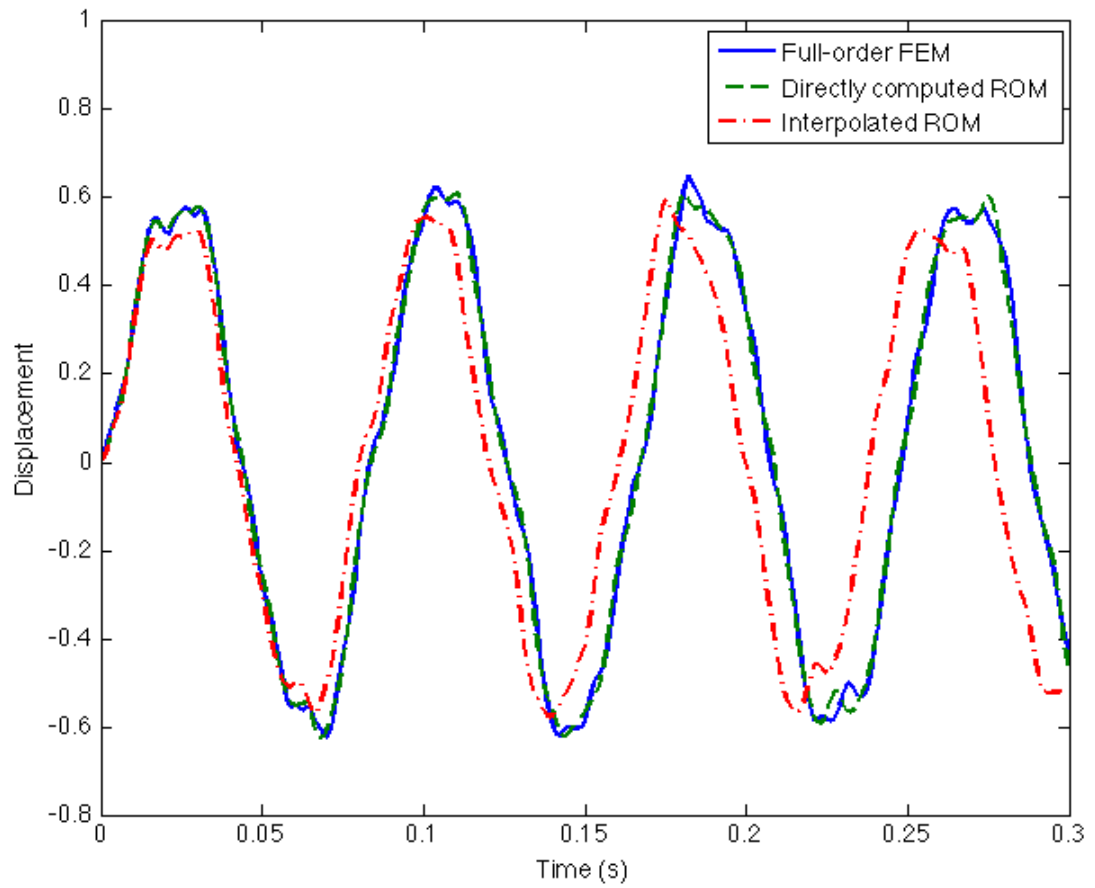
Figure 5.5: Test design point $\boldsymbol{\mu}^{(c)}$: transient responses predicted by the complete FEM model, the directly computed ROM and the interpolated ROM.

Chapter 6

Exploitation of a Database of Reduced-Order Bases and Models

6.1 Introduction

In chapter 4, a methodology for the analysis of aeroelastic systems based on reduced-order models was established. It was demonstrated that the reduced-order bases of the fluid subsystem are constructed for specific values of flight parameters (Mach number and angle of attack), but are not robust with respect to changes in these parameters. As such, a method based on the interpolation on the tangent space to the Grassmann manifold was developed in order to adapt ROMs to new, last-minute, flight conditions. This method was illustrated with aeroelastic analyses of an F-16 Block 40 and an F-18/A full aircraft configurations. In each case, two to five pre-computed ROMs were used in the interpolation process and, thus, a couple interpolation schemes were compared. It was shown that, in general, 4-point interpolation schemes produced better agreements with high-fidelity responses than 2-point schemes.

The present chapter focuses on the case where a large database of such pre-computed ROMs is available. As mentioned in introduction of this thesis, such a database can be the result, for instance, of the routine analysis of the flutter stability of a full aircraft for various pairs of flight parameters (M_{∞_i}, α_i). However, when several dozens pre-computed ROMs are present in a database, the number of possible interpolation schemes becomes very large, even when extrapolations are excluded. The goal of this chapter is to establish a cost effective machine learning-based procedure for exploiting a large database of reduced-order bases and models and determining optimal interpolation schemes for constructing ROMs at new flight conditions. All fluid ROBs stored in the database are constrained to have the same dimension in order to facilitate their interpolation as explained in the following section. The same F-16 Block 40 model studied in section 4.5 is considered.

After presenting the database of interest in section 6.2, an algorithm relying on concepts from machine learning is developed in section 6.3 for locally selecting an optimal interpolation scheme in the parameter space. Corresponding aeroelastic predictions are then studied in section 6.4. Numerical results as well as computational costs are reported. Extensions of the methods to parameter spaces of higher dimensions are then suggested in section 6.5. Finally, a solution to the problem of determining optimal points to be added to the database is exposed in section 6.6.

6.2 Database Setup

The space free-stream Mach number/angle of attack is sampled by eighty-three pairs of flight conditions (M_{∞_i}, α_i) that are chosen randomly but with the objective of covering the subsonic, transonic and supersonic flight regimes and a range of angles

(a) Parameter sampling and parametric domain of interest for aeroelastic predictions (interior of region delimited by blue line)

(b) Triangulation of the data points (blue lines) and construction of the dual cells (red lines)
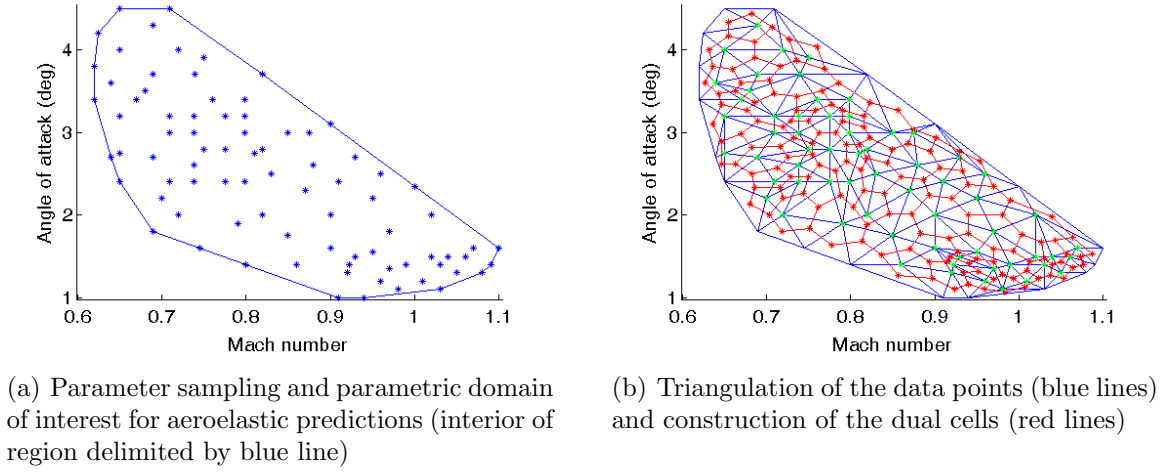
Figure 6.1: F-16 database of fluid ROBs and dual cell representation

of attack $\big($Fig. 6.1(a)$\big)$. For each chosen pair of flight conditions, a fluid ROB of dimension 90 is pre-computed as explained in section 4.4. The fluid ROBs are stored in a database together with a structural ROM constructed by projecting the mass and stiffness matrices of the detailed FE structural model of the F-16 aircraft on its first nine natural dry modes. Hence, all linearized aeroelastic ROMs based on this database are of dimension 99 (90 + 9).

To demonstrate the on-demand aeroelastic parametric identification of the chosen F-16 configuration using this database, the parametric domain of interest is set to the interior of the convex hull of the database points, as shown in Fig. 6.1(a).

6.3   Discretization, Decomposition, Training, and Reduction

Next, the space free-stream Mach number/angle of attack is triangulated using the data points introduced above $\big($Fig. 6.1(b)$\big)$. A set of dual cells is constructed from the

triangles and decomposed into overlapping clusters of cells. Within each cluster, two sets of data points are distinguished (Fig. 6.2): a set of interior data points whose enclosing cells belong to one cluster only, and a set of boundary data points that are shared by at least two clusters, except at the boundary of the convex hull of the database points. The latter set of data points results from the overlapping nature of the decomposition. Its role is to ensure that any interpolation within a cell never leads in practice to an extrapolation (because extrapolations can lead to numerical instabilities and/or inaccuracies). Finally, using concepts from machine learning [107, 108], the database is trained to identify within each dual cell an appropriate local interpolation scheme.
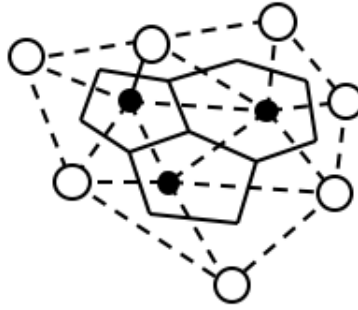
Figure 6.2: Schematic description of a cluster of dual cells of the database: the black and white circles denote the interior and boundary data points, respectively. The triangles resulting from the Delaunay triangulation are drawn using dashed lines, and the corresponding dual cells are delimited by full lines.

6.3.1 Triangulation and Dual Cell Construction

The discretization of the parametric domain of interest into dual cells containing each one and only one data point is performed in two steps as follows:

1. *Perform a Delaunay triangulation for the entire set of data points.* The resulting triangles are shown in blue in Fig. 6.1(b).

2. *Generate the associated dual cells.* For each "interior" data point, the corresponding dual cell is defined as the polygon whose vertices are the centers of gravity of the triangles connected to this data point, as shown in Fig. 6.2.

The dual cells associated with the triangulation performed in Step 1 above are highlighted in red in Fig. 6.1(b). Their union covers the interior of the regions of interest in the database.

6.3.2 Clustering and Cross-Validation

Next, the k-means algorithm [108] is applied to decompose the database — more specifically, its representation by dual cells — into overlapping clusters $\mathcal{C}_j$ with two to five dual cells and nine to 13 data points each $\big($Fig. 6.3(a)$\big)$. Within each cluster $\mathcal{C}_j$, a cross-validation process [108] is applied to select, among a few considered choices, the best local interpolation model. To this effect, and in order to minimize pre-processing time, two different $\mathcal{N}_k$-point interpolation model $\mathcal{M}_k$, $k = 1, 2$, are considered: model $\mathcal{M}_1$ using $\mathcal{N}_1 = 4$ interpolation points of the cluster $\mathcal{C}_j$, and model $\mathcal{M}_2$ using $\mathcal{N}_2 = 5$. Each interpolation model is tested as follows. The interior data points of each cluster $\mathcal{C}_j$ are treated as test points. At each test point, a dozen "surrogate" fluid ROBs are computed in real-time by interpolating fluid ROBs pre-computed at $\mathcal{N}_k$ randomly chosen other data points of the cluster $\mathcal{C}_j$ containing the test point, and

the corresponding dozen of linearized aeroelastic ROMs are constructed. Then, the numerical stability and accuracy of each constructed surrogate aeroelastic ROM is assessed by comparing the solution it delivers for a test aeroelastic response problem to that produced by the linearized aeroelastic ROM constructed directly from the fluid ROB available at this data (or test) point. Next, for each test point (M_{∞_i}, α_i) located within $\mathcal{C}_j$, the set of interpolation data points $\mathcal{P}_{ijk}$ (also located within $\mathcal{C}_j$) leading for each interpolation model $\mathcal{M}_k$ to the smallest L_2-norm of the discrepancy in the transient lift, ε_{ijk}, is identified. For each model $\mathcal{M}_k$, the generalized error at the test point (M_{∞_i}, α_i) is estimated as the mean value $\bar{\varepsilon}_{ijk}$ of all of the discrepancy values ε_{ijk}. Finally, in each dual cell identified by its center data point (M_{∞_i}, α_i), the interpolation model $\mathcal{M}_k$ with the lowest estimated generalized error $\bar{\varepsilon}_{ijk}$ is chosen $\big($Fig. 6.3(b)$\big)$ and assigned the set of interpolation data points $\mathcal{P}_{ijk}$ $\big($Fig. 6.3(c)$\big)$.

It is noted that a byproduct of the training described above is the effective reduction of the original database to the sets of points $\mathcal{P}_{ijk}$ chosen for the local interpolations.

6.4 On-Demand Aeroelastic Predictions

On-demand predictions of the transient aeroelastic responses of the considered aircraft to initial perturbations of its steady-state aeroelastic equilibrium states are illustrated here for five "flight" points $\big\{(M^\star_{\infty_l}, \alpha^\star_l)\big\}^5_{l=1}$ that are shown in red in Fig. 6.4. The $\star$ superscript emphasizes that no information is available in the database at these pairs of flight conditions. Using the aeroelastic computational strategy proposed in this thesis, a fast numerical aeroelastic prediction is performed for each of them as follows:
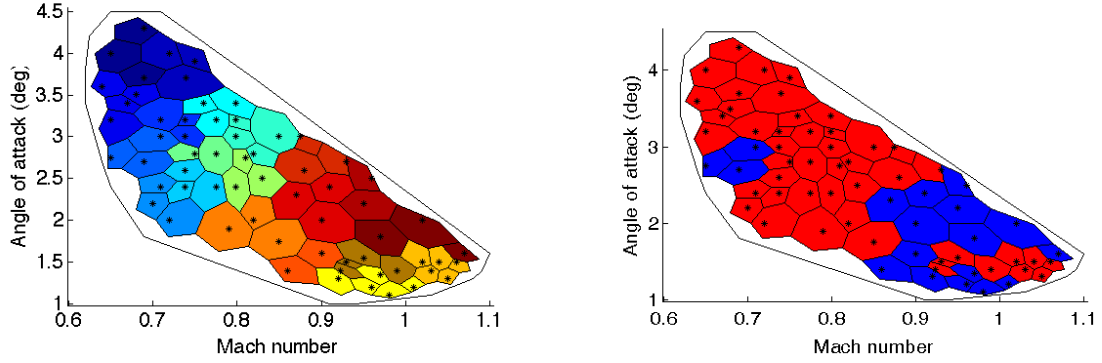
(a) Clustering of the dual cells (each color represents a cluster and each dot represents a test point)

(b) Retained local interpolation models (the dual cells where model $\mathcal{M}_1$ is retained are shown in blue and those where model $\mathcal{M}_2$ is retained are shown in red; each dot represents a test point)
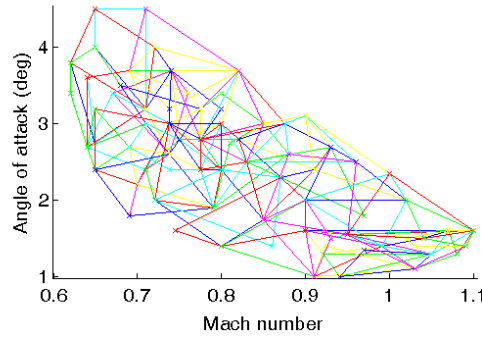
(c) Retained data points for local interpolations (for each dual cell, the corresponding set of interpolating data points is shown as a polygon enclosing it)

Figure 6.3: Decomposition, training, and reduction of the F-16 database of fluid ROBs

1. *Compute the steady-state equilibrium solution of the aerodynamic problem at $(M_{\infty_l}^\star, \alpha_l^\star)$.*

2. *Evaluate the linearized coupled fluid-structure operators at the above steady-state solution.*

3. *Determine to which dual cell the point $(M_{\infty_l}^\star, \alpha_l^\star)$ belongs to and select its local interpolation model.*

4. *Interpolate the corresponding pre-computed logarithm mappings in the tangent space to the Grassmann manifold at the cell center to obtain $\mathbf{\Gamma}(M_{\infty_l}^\star, \alpha_l^\star)$.*

5. *Map $\mathbf{\Gamma}(M_{\infty_l}^\star, \alpha_l^\star)$ to $\mathbf{\Phi}_l^\star = \mathbf{\Phi}(M_{\infty_l}^\star, \alpha_l^\star)$ using the exponential map. $\mathbf{\Phi}_l^\star$ is now the interpolated fluid ROB at $(M_{\infty i}^\star, \alpha_i^\star)$.*

6. *Construct the fluid ROM at $(M_{\infty_l}^\star, \alpha_l^\star)$ by projecting the linearized fluid-structure operators on the fluid ROB $\mathbf{\Phi}_l^\star$.*

7. *Assemble the partitioned aeroelastic ROM at $(M_{\infty_l}^\star, \alpha_l^\star)$ and post-process it in the time-domain.*

6.4.1   Numerical Results

Figs. 6.5-6.9 compare the lift time-histories predicted by the on-demand computational strategy proposed in this paper at the flight points $(M_{\infty_1}^\star, \alpha_1^\star)$, $(M_{\infty_2}^\star, \alpha_2^\star)$, $(M_{\infty_3}^\star, \alpha_3^\star)$, $(M_{\infty_4}^\star, \alpha_4^\star)$ and $(M_{\infty_5}^\star, \alpha_5^\star)$, respectively, to their counterparts obtained using the linearized aeroelastic ROMs directly computed at these flight points and the corresponding linearized high-dimensional computational models. Similar comparisons can be found in Reference [88] for other flight points. Table 6.1 reports for
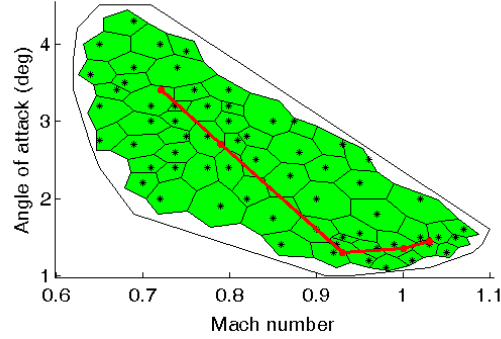
Figure 6.4: Flight points not represented in the database for which on-demand aeroelastic predictions are requested (shown in red; test points are shown in black)

each of the requested flight points $(M^\star_{\infty_l}, \alpha^\star_l)$ the L_2-norm of the relative difference between the transient lift predicted using the on-demand computational model and its counterpart associated with the corresponding high-dimensional linearized computational model, ε_l. All of Figs. 6.5-6.9 and Table 6.1 demonstrate that, except for the flight condition $(M^\star_{\infty_4}, \alpha^\star_4)$, the proposed on-demand computational strategy delivers excellent results. At the sonic point $(M^\star_{\infty_4}, \alpha^\star_4)$, it delivers acceptable results.

Flight point l	1	2	3	4	5
ε_l	0.41%	0.55%	6.94%	14.8%	3.59%

Table 6.1: Assessment of the numerical accuracy of the proposed on-demand computational strategy

6.4.2 Computational Costs

Table 6.2 reports the CPU timings associated with populating one data point of the F-16 database described above. About 89% of the total CPU time (102 mns) is consumed by the generation of 99 fluid snapshots in the frequency domain and the
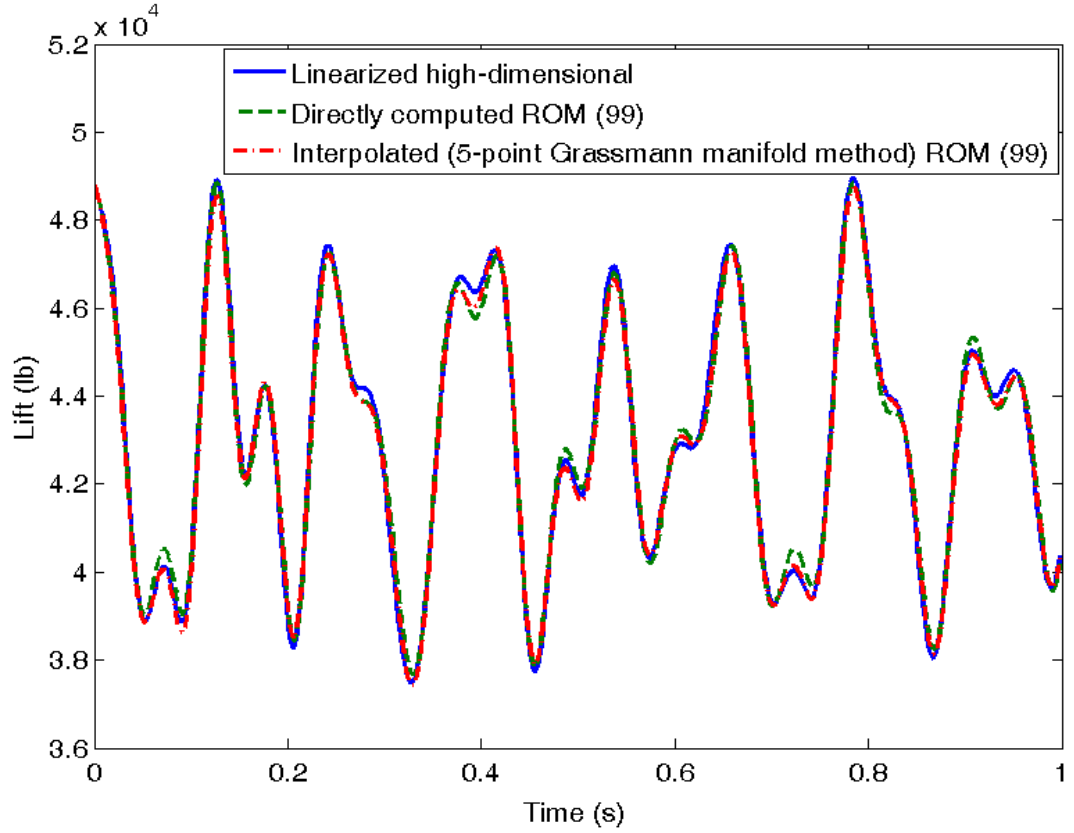
Figure 6.5: Aeroelastic predictions at $M^\star_{\infty_1} = 0.721$ and $\alpha^\star_1 = 3.4°$

construction of the fluid ROB of dimension 90 using the POD method. All reported computations are shown to scale well when the number of processors is doubled. From these results, the reader can infer that the total CPU time associated with setting up the entire F-16 database is 141.1 hrs on 32 processors (83×102 minutes), and 70.6 hrs on 64 processors.

Tables 6.3-6.4 report the computational cost associated with the prediction of the aeroelastic response of the considered F-16 configuration to some initial perturbation of its steady-state aeroelastic equilibrium at a flight point that is not available in the
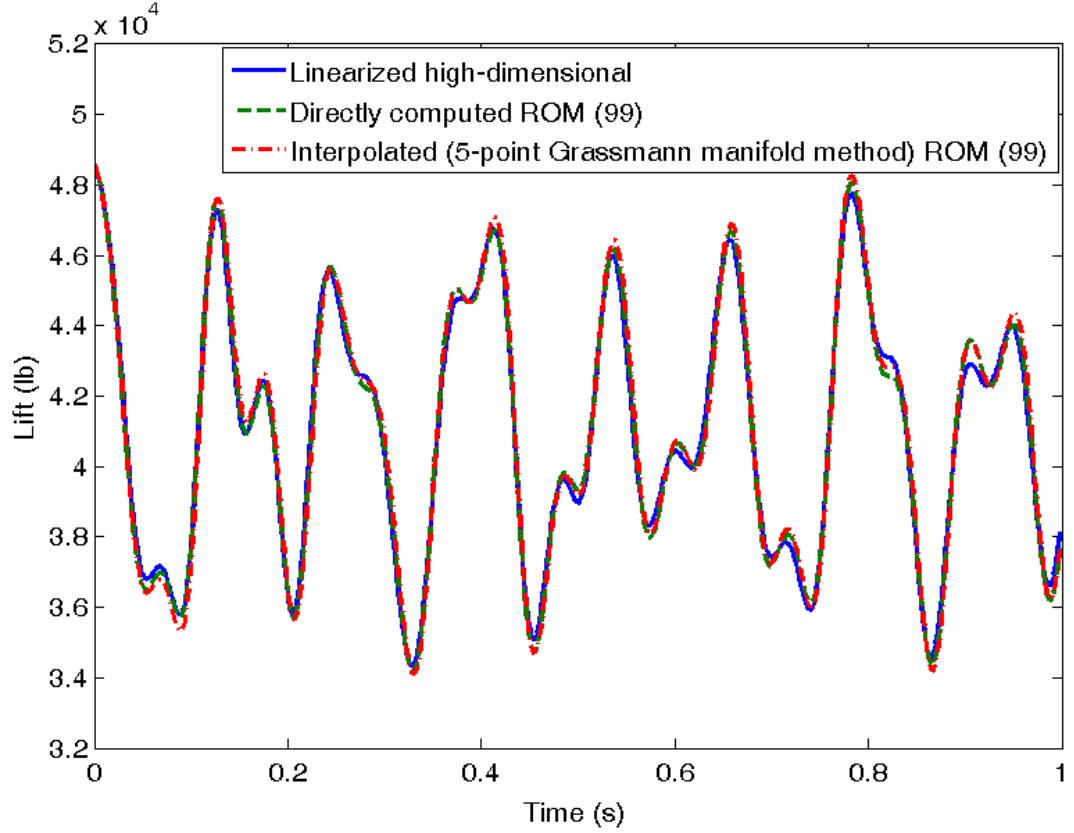
Figure 6.6: Aeroelastic predictions at $M^{\star}_{\infty_2} = 0.790$ and $\alpha^{\star}_2 = 2.7°$

database, for two different computational modeling strategies: (a) a linearized aeroe-lastic ROM of dimension 99 based on a fluid ROB of dimension 90 that is directly computed at the desired flight point by the POD method, and (b) the on-demand computational strategy proposed in this paper that is based on a database of fluid ROBs of dimension 90 and suitable local models of the interpolation method in a tan-gent space to the Grassmann manifold. In both cases, after the linearized aeroelastic ROM of dimension 99 is constructed, the aeroelastic response is predicted by time-integrating the initial boundary value problem associated with this ROM for the first one second using 500 time-steps and a single processor. As indicated by Figs. 6.5-6.9,
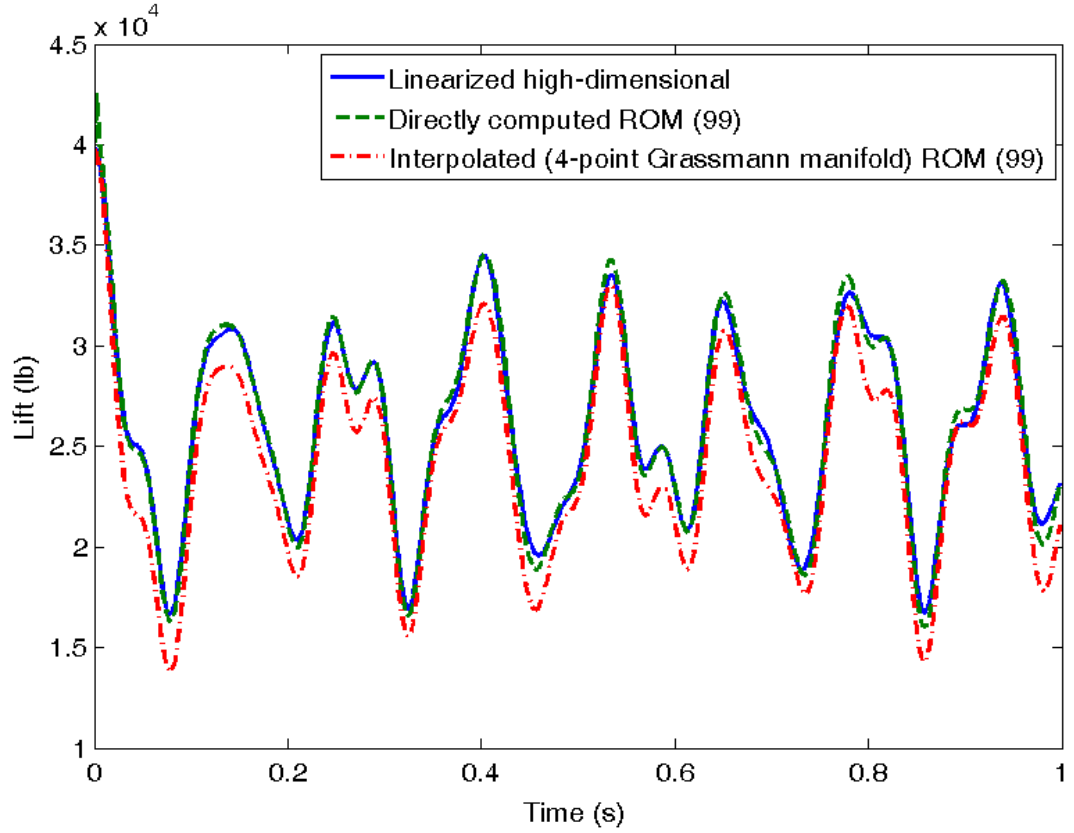
Figure 6.7: Aeroelastic predictions at $M^\star_{\infty_3} = 0.930$ and $\alpha^\star_3 = 1.3°$

the one second time-interval is sufficient for identifying the aeroelastic parameters of the considered F-16 configuration at each requested flight point. The reader can observe that the proposed interpolation method accelerates the generation of a fluid ROB by a factor greater than 50 $\big((12 + 90)$ mins/ 2 mins$\big)$. The reader can also observe that the total CPU time consumed by the proposed computational strategy for this on-demand aeroelastic prediction (not including the initial cost associated with setting up the database) is 15.5 mns. This cost is distributed as follows: 12 mns for computing a steady state flow at the desired flight point that is not represented in the database, 2.5 mns only for constructing a fluid ROM at the desired flight point by
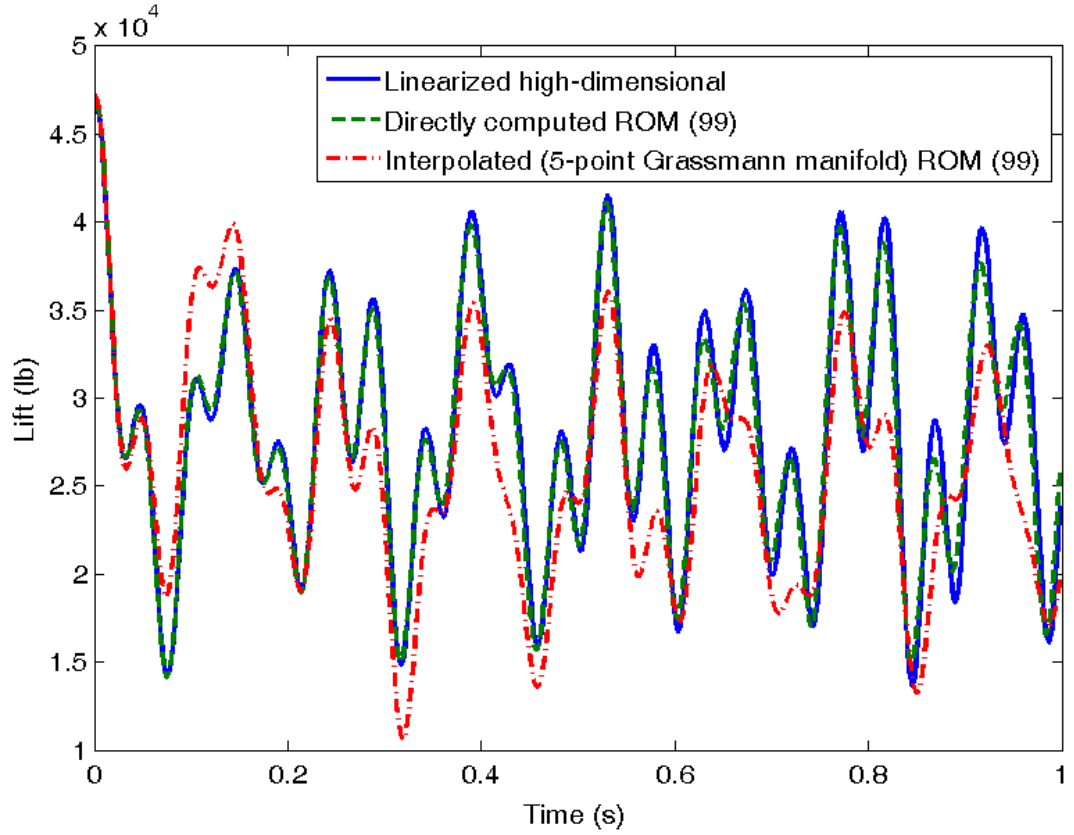
Figure 6.8: Aeroelastic predictions at $M^{\star}_{\infty_4} = 1.000$ and $\alpha^{\star}_4 = 1.35°$

interpolation, and one minute only for assembling and analyzing the corresponding aeroelastic ROM for performing the desired aeroelastic prediction. Hence, this total CPU cost is dominated by the cost associated with computing a steady-state flow at the desired flight point which is not represented in the database.

Furthermore, considering that predicting the aeroelastic response of the same F-16 configuration, to the same initial perturbation, at the same flight conditions, for the same time-interval, and using also 500 time-steps but the linearized high-dimensional computational model produces nearly the same result but requires 44 minutes CPU on 32 processors, and 21.5 minutes on 64 processors, the computational advantage of

Computational step	N_{CPU}	CPU time	N_{CPU}	CPU time
Computation of steady-state equilibrium	32	12	64	6
Generation of 99 fluid snapshots and a fluid ROB (90)	32	90	64	50
Total CPU time		102		56

Table 6.2: Computational cost associated with populating one data point of the database using $N_{CPU} = 32$ or $N_{CPU} = 64$ processors (Time in minutes)

Computational step	N_{CPU}	CPU time	N_{CPU}	CPU time
Computation of steady-state equilibrium	32	12	64	6
Generation of 99 fluid snapshots and a fluid ROB (90)	32	90	64	50
Construction of fluid ROM	32	0.5	64	0.25
Assembly and post-processing of aeroelastic ROM	1	1	1	1
Total CPU time		103.5		57.25

Table 6.3: Computational cost associated with the prediction of the first second of aeroelastic response of an F-16 configuration using an aeroelastic ROM based on a fluid ROB computed directly at the desired flight point and $N_{CPU} = 32$ or $N_{CPU} = 64$ processors (Time in minutes)
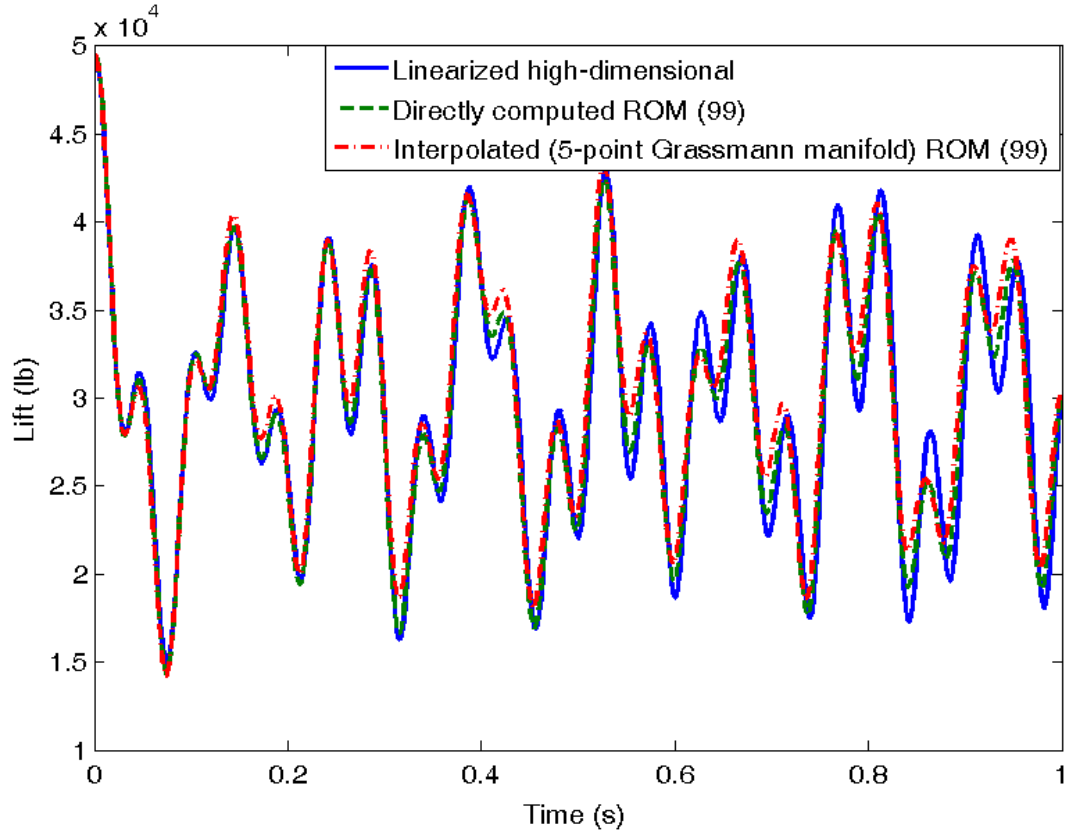
Figure 6.9: Aeroelastic predictions at $M^{\star}_{\infty_5} = 1.030$ and $\alpha^{\star}_5 = 1.45°$

the proposed on-demand computational strategy based on a pre-computed database of reduced-order information is impressive, and its potential for assisting flutter flight testing is promising.

6.5 Extension to Higher Dimensions

The methodology developed in section 6.3.1 relies on the Delaunay triangulation procedure, and as such, bears a cost which can grow exponentially as a power of

Computational step	N_{CPU}	CPU time	N_{CPU}	CPU time
Computation of steady-state equilibrium	32	12	64	6
Interpolation of fluid ROBs (5 points)	32	2	64	1
Construction of fluid ROM	32	0.5	64	0.25
Assembly and post-processing of aeroelastic ROM	1	1	1	1
Total CPU time		15.5		8.25

Table 6.4: Computational cost associated with the prediction of the first second of aeroelastic response of an F-16 configuration using the interpolation-based on-demand computational strategy and $N_{CPU} = 32$ or $N_{CPU} = 64$ processors (Time in minutes)

the dimension N_p of the parameter spaces, rendering the method infeasible in high-dimensional parameter spaces. As such, an method based on a regular grid of pre-computed points in the parameter space can be alternatively developed for high dimensions, as illustrated in section 5.4 for a database of structural reduced-order bases and models. This method is however also doomed by the "curse of dimensionality" as the number of required pre-computed points increases as a power of N_p. The recent development of methods based on sparse grids [109] could be used to alleviate that cost.

6.6 Note on Database Generation

In this chapter, it was assumed that the database was already generated, and as such there was no need to generate additional points. In this section, a heuristic, greedy methodology based on differential geometry is developed in order to cheaply determine at which locations in the parameter space new points should be added. The algorithm proceeds by computing distances between ROBs along the edges of the Delaunay triangles and finding the largest of these distances. A new ROB is then

generated for values of the parameters corresponding to the mid-point of that edge

6.6.1 Geodesic Distance Between Subspaces

Definition. Letting $\mathcal{S}_1$ and $\mathcal{S}_2$ be two subspaces belonging to $\mathcal{G}(N_\Phi, N_f)$ and $\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2 \in \mathcal{ST}(k, n)$ two respective bases, the geodesic distance between $\mathcal{S}_1$ and $\mathcal{S}_2$ on the Grassmann manifold $\mathcal{G}(N_\Phi, N_f)$ is defined as

$$d_\mathcal{G}(\mathcal{S}_1, \mathcal{S}_2) = \sum_{i=1}^{N_\Phi} \theta_i^2, \tag{6.1}$$

where $(\theta_1, \cdots, \theta_{N_\Phi})$ are the principal angles (see aection 3.7.3) between $\mathcal{S}_1$ and $\mathcal{S}_2$. These angles are the diagonal elements of $\boldsymbol{\Theta}$, computed as (see section 3.7.5)

$$\boldsymbol{\Phi}_1^T \boldsymbol{\Phi}_2 = \mathbf{QCZ}^T \quad \text{(Full SVD)} \tag{6.2}$$

$$\boldsymbol{\Theta} = \cos^{-1} \mathbf{C}. \tag{6.3}$$

6.6.2 Methodology

The algorithm proceeds in four steps as follows:

1. Consider the set of edges $\mathcal{E} = \{E_i = ((M_{\infty i_1}, \alpha_{i_1}), (M_{\infty i_2}, \alpha_{i_2}))\}_{i=1}^{N_E}$ in the Delaunay triangulation obtained in section 6.3.1, excluding the boundary edges.

2. Compute the geodesic distances between the corresponding doublets of subspaces

$$\{d_\mathcal{G}(\mathcal{S}(M_{\infty i_1}, \alpha_{i_1}), \mathcal{S}(M_{\infty i_2}, \alpha_{i_2}))\}_{i=1}^{N_E}, \tag{6.4}$$

where $\mathcal{S}(M_\infty, \alpha)$ denote the subspace spanned by the column of the POD basis constructed for the flight parameters (M_∞, α).

3. Find the edge i_0 such that

$$i_0 = \mathrm{argmin}_{i=1,\cdots,N_E} d_{\mathcal{G}} \left(\mathcal{S}(M_{\infty i_1}, \alpha_{i_1}), \mathcal{S}(M_{\infty i_2}, \alpha_{i_2}) \right). \tag{6.5}$$

4. Generate a new POD basis for the parameters

$$(M_\infty^\star, \alpha^\star) = \left(\frac{M_{\infty i_0 1} + M_{\infty i_0 2}}{2}, \frac{\alpha_{i_0 1} + \alpha_{i_0 2}}{2} \right). \tag{6.6}$$

Adding a new point results into splitting two pre-existing triangles in the decomposition into four smaller triangles as sketched in Fig. 6.10. A new local training can then be processed inside the cluster this new point belongs to.
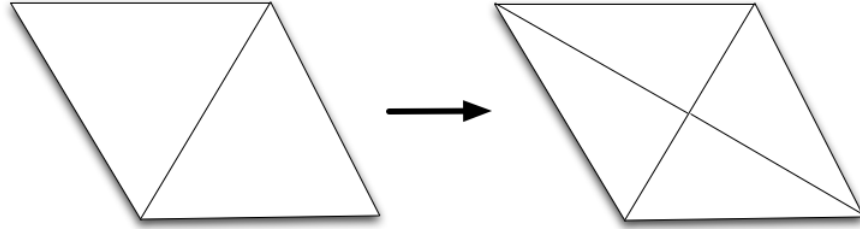
Figure 6.10: Geodesic distances between the pre-computed points in the database

6.6.3 Application to the F-16 ROM Database

Figure 6.11 reports the geodesic distances obtained after Step 2 of the proposed algorithm for the database considered in this chapter. It can be seen that a new point should be computed for values of $(M_\infty^\star, \alpha^\star) \approx (1.07, 2.6°)$.
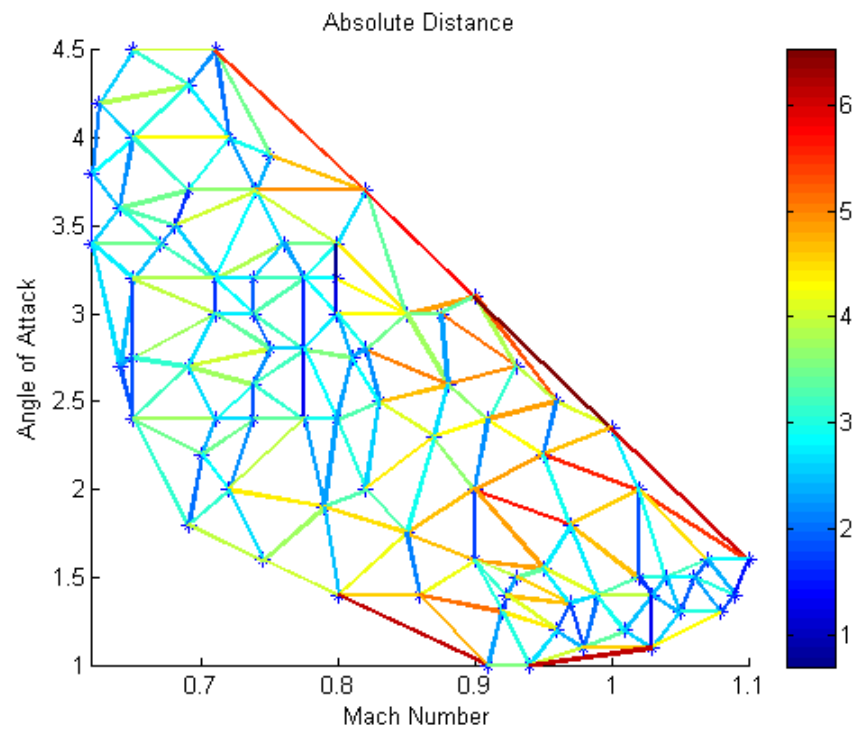
Figure 6.11: Geodesic distances between the pre-computed points in the database

Chapter 7

Interpolation of Linear Fluid and Structural Reduced-Order Models

7.1   Introduction

In the previous chapters, adaptation of reduced-order information involved the pre-computation and interpolation of the reduced-order bases (ROBs) themselves. The ROBs are essential as they enable mapping back the generalized coordinate vectors to the full state-space. As such, when quantities related to the full state vector need to be computed, the ROBs are required and hence these quantities have to be interpolated. Such an interpolation can induce a non-negligible computational burden, especially in terms of memory requirements, since the dimension of the ROBs scales with the size of the full-order model. As such, in many cases, computations on a parallel processor are needed.

In other cases, however, only the dynamics of the system is sought. For instance, the stability of a linear system can be determined directly from the inspection of the

eigenvalues of the underlying linear operator. As such, interpolation of the reduced-order basis is not necessary in those cases: only the adaptation of the reduced linear operators is required —these operators are also called by extension reduced-order models in this thesis.

On one hand, interpolating the ROM involves handling small-size matrices and the associated computations can be effected in real-time. On the other hand, the ROMs are matrices expressed in different sets of coordinates across the values of the parameters, and as such, these ROMs are often written in inconsistent sets of coordinates. This leads to a hard problem to solve. In this thesis, two different methods are proposed to solve this problem: the first one is based on a heuristic approach and the second one has connections to principal angles between subspaces. In both cases, it will be subsequently assumed that the subspaces defined by the ROBs vary continuously with the parameters.

This chapter is organized as follows. In the remainder of section 7.1, the role of the reduced-order bases in the expressions of the ROMs is emphasized. In section 7.2, the two problems of interest are formulated. Algorithms to solve these problems are subsequently proposed in section 7.3 and associated numerical methods developed in section 7.4. Applications to synthetic, fluid and structural dynamical systems are offered in section 7.6.

7.1.1 Equivalent Classes of LTI ROMs

The triplet of reduced-order linear operators $\left(\mathbf{A}_r(\boldsymbol{\mu}), \mathbf{B}_r(\boldsymbol{\mu}), \mathbf{C}_r(\boldsymbol{\mu}) \right)$ determined from Eq. (2.12)–(2.14) defines the ROM in the representative bases $\left(\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu}) \right)$. The generation of such a ROM through a projection process leads to invariance properties of the reduced system.

Proposition 1. The input-output behavior of a ROM does not depend on the choice of the trial and test basis defining the trial and test subspaces.

Proof. A proof is offered in Appendix K.1. □

This property justifies considering without any loss of generality, in the remainder of this chapter, trial and test bases that have orthogonality properties. Any basis can indeed be transformed into an orthogonal basis by, for instance, a Gram-Schmidt procedure. These orthogonal bases belong to the compact Stiefel manifold $\mathcal{ST}(k, n)$.

The following property demonstrates how the reduced-order operators are transformed when different reduced-order bases are chosen inside the trial and test subspaces.

Proposition 2. Considering the reduced-order system defined in Eq. (2.11)–(2.14), let $\left(\tilde{\mathbf{V}}(\boldsymbol{\mu}), \tilde{\mathbf{W}}(\boldsymbol{\mu})\right)$ denote another doublet of trial and test bases defining the same trial and test subspaces respectively. Since all the bases considered here belong to the Stiefel manifold, and as such have orthogonal columns, there exists two orthogonal matrices $\mathbf{P}$ and $\mathbf{Q}$ in the set $\mathrm{O}(k)$ of orthogonal matrices of size k, such that

$$\tilde{\mathbf{W}}(\boldsymbol{\mu}) = \mathbf{W}(\boldsymbol{\mu})\mathbf{P}, \ \ \tilde{\mathbf{V}}(\boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})\mathbf{Q}. \tag{7.1}$$

Let $\left(\tilde{\mathbf{A}}_r(\boldsymbol{\mu}), \tilde{\mathbf{B}}_r(\boldsymbol{\mu}), \tilde{\mathbf{C}}_r(\boldsymbol{\mu})\right)$ denote the reduced-order model defined by Petrov-Galerkin projection of the full-order model (2.1) using the bases $\left(\tilde{\mathbf{V}}(\boldsymbol{\mu}), \tilde{\mathbf{W}}(\boldsymbol{\mu})\right)$. Then

$$\tilde{\mathbf{A}}_r(\boldsymbol{\mu}) = \mathbf{Q}^T \mathbf{A}_r(\boldsymbol{\mu})\mathbf{Q}$$

$$\tilde{\mathbf{B}}_r(\boldsymbol{\mu}) = \mathbf{Q}^T \mathbf{B}_r(\boldsymbol{\mu}) \tag{7.2}$$

$$\tilde{\mathbf{C}}_r(\boldsymbol{\mu}) = \mathbf{C}_r(\boldsymbol{\mu})\mathbf{Q}.$$

Proof. The proof can be immediately established by substituting the new expressions (7.1) of the reduced-order bases in Eq. (2.12)-(2.14). □

This result is important since it demonstrates that transforming the reduced-order bases using the matrices $\mathbf{P}$ and $\mathbf{Q}$ results into a congruence transformation of the ROM that is only based on $\mathbf{Q}$. One can consider as a result the following group action α [112]

$$
\begin{aligned}
\alpha : \ \mathrm{O}(k) \times \mathcal{R}(p, k, q) &\longrightarrow \mathcal{R}(p, k, q) \\
\left(\mathbf{Q}, (\mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r)\right) &\longmapsto (\mathbf{Q}^T \mathbf{A}_r \mathbf{Q}, \mathbf{Q}^T \mathbf{B}_r, \mathbf{C}_r \mathbf{Q}).
\end{aligned}
\tag{7.3}
$$

The orbit of an element $\left(\bar{\mathbf{A}}_r, \bar{\mathbf{B}}_r, \bar{\mathbf{C}}_r, \right)$ belonging to $\mathcal{R}(p, k, q)$ is then

$$
\begin{aligned}
\mathcal{F}\left(\bar{\mathbf{A}}_r, \bar{\mathbf{B}}_r, \bar{\mathbf{C}}_r\right) &= \left\{ \alpha\left(\mathbf{Q}, (\bar{\mathbf{A}}_r, \bar{\mathbf{B}}_r, \bar{\mathbf{C}}_r)\right) \mid \mathbf{Q} \in \mathrm{O}(k) \right\} \\
&= \left\{ (\mathbf{Q}^T \bar{\mathbf{A}}_r \mathbf{Q}, \mathbf{Q}^T \bar{\mathbf{B}}_r, \bar{\mathbf{C}}_r \mathbf{Q}) \mid \mathbf{Q} \in \mathrm{O}(k) \right\},
\end{aligned}
\tag{7.4}
$$

and the corresponding equivalence relation $\sim$ between elements R_1 and R_2 of $\mathcal{R}(p, k, q)$ is

$$
R_1 \sim R_2 \Leftrightarrow \ \exists \mathbf{Q} \in \mathrm{O}(k) \mid \alpha(\mathbf{Q}, R_1) = R_2.
\tag{7.5}
$$

To summarize, any given ROM can be expressed in a variety of orthogonal bases. The resulting set of expressions of its equivalent ROMs forms its orbit $\mathcal{F}(\cdot)$. When several ROMs $\{R_i\}_{i=1}^{N_R}$ have to be compared, as it is the case in an interpolation process for instance, a real challenge is therefore to be able to express these ROMs in consistent sets of coordinates. Equivalently, this amounts to finding a consistent representative element for each ROM R_i inside its orbit $\mathcal{F}(R_i)$.

7.1.2 Role of the Reduced-Order Bases

Oftentimes, reduced operators of the form $\mathbf{A}_r(\boldsymbol{\mu}) = (\mathbf{W}(\boldsymbol{\mu})^T \mathbf{V}(\boldsymbol{\mu}))^{-1} \mathbf{W}(\boldsymbol{\mu})^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{V}(\boldsymbol{\mu})$ are computed and compared for various values of the parameter vector $\boldsymbol{\mu}$. When the reduced bases do not depend on $\boldsymbol{\mu}$

$$\mathbf{V}(\boldsymbol{\mu}) = \mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_k \end{bmatrix}, \ \mathbf{W}(\boldsymbol{\mu}) = \mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_k \end{bmatrix}. \tag{7.6}$$

The comparison of the linear ROMs is straightforward since all the operators are written in the same bases and, as such, a given entry of a reduced linear operator has the same interpretation across parameters. For instance, the (i, j)-the entry of $\mathbf{A}_r(\mu)$ is

$$a_{rij}(\boldsymbol{\mu}) = \mathbf{z}_i^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{v}_j, \tag{7.7}$$

$\mathbf{z}_i$ denoting the i-th column vector of $\mathbf{Z} = \mathbf{W}(\mathbf{V}^T \mathbf{W})^{-1}$.

When the bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$ do depend on the parameter $\boldsymbol{\mu}$, the reduced bases may not correspond to the same coordinate bases, and as such direct comparison of the linear ROMs may not correct. One should note that this may also occur in cases when the subspaces $\mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu})$ and $\mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu})$ respectively spanned by the column vectors of $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$ do not depend on the parameters $\boldsymbol{\mu}$, as demonstrated by the following example.

Example 1. Full-order operators $\mathbf{A}(\boldsymbol{\mu}_l) \in \mathbb{R}^{n \times n}$ depending continuously on $\boldsymbol{\mu}$ are considered for two values of the parameter vector $\boldsymbol{\mu}_l$, $l = 1, 2$. Let $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$ denote the canonical basis of $\mathbb{R}^n$. Each entry of $a_{ij}(\boldsymbol{\mu}_1) = \mathbf{e}_i^T \mathbf{A}(\boldsymbol{\mu}_1) \mathbf{e}_j$ at a given parameter value $\boldsymbol{\mu}_1$ can be compared to its counterpart entry $a_{ij}(\boldsymbol{\mu}_2)$ since both operators are expressed in the same coordinate basis. Let now the two sets of reduced-order bases

be defined as

$$\mathbf{W}(\boldsymbol{\mu}_1) = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}, \ \mathbf{V}(\boldsymbol{\mu}_1) = \begin{bmatrix} \mathbf{e}_1 & \frac{1}{\sqrt{2}}(\mathbf{e}_2 + \mathbf{e}_3) \end{bmatrix}, \tag{7.8}$$

and

$$\mathbf{W}(\boldsymbol{\mu}_2) = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}, \ \mathbf{V}(\boldsymbol{\mu}_2) = \begin{bmatrix} -\mathbf{e}_1 & \frac{1}{\sqrt{2}}(\mathbf{e}_2 + \mathbf{e}_3) \end{bmatrix}. \tag{7.9}$$

The projection subspaces are identical at the two values of the parameter: $\mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu}_1) = \mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu}_2) = \text{span}\{\mathbf{e}_1, \mathbf{e}_2\}$ and $\mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu}_1) = \mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu}_2) = \text{span}\{\mathbf{e}_1, \mathbf{e}_2 + \mathbf{e}_3\}$. Subsequently,

$$\mathbf{A}_r(\boldsymbol{\mu}_1) = \begin{bmatrix} a_{11}(\boldsymbol{\mu}_1) & \frac{1}{\sqrt{2}}(a_{12}(\boldsymbol{\mu}_1) + a_{13}(\boldsymbol{\mu}_1)) \\ a_{21}(\boldsymbol{\mu}_1) & \frac{1}{\sqrt{2}}(a_{22}(\boldsymbol{\mu}_1) + a_{23}(\boldsymbol{\mu}_1)) \end{bmatrix}, \tag{7.10}$$

and

$$\mathbf{A}_r(\boldsymbol{\mu}_2) = \begin{bmatrix} -a_{11}(\boldsymbol{\mu}_2) & \frac{1}{\sqrt{2}}(a_{12}(\boldsymbol{\mu}_2) + a_{13}(\boldsymbol{\mu}_2)) \\ -a_{21}(\boldsymbol{\mu}_2) & \frac{1}{\sqrt{2}}(a_{22}(\boldsymbol{\mu}_2) + a_{23}(\boldsymbol{\mu}_2)) \end{bmatrix}. \tag{7.11}$$

Therefore, this shows that a direct comparison of the entries of the first columns of $\mathbf{A}_r(\boldsymbol{\mu}_l)$, $l = 1, 2$ would be ill-advised, since these reduced operators are not written in consistent coordinate bases.

When the subspaces $\mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu})$ and $\mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu})$ depend continuously on the parameter $\boldsymbol{\mu}$ (i.e. are not invariant), it is also possible to interpolate the reduced operators as long as they are expressed in consistent coordinate bases as well. This is illustrated by the following example.

Example 2.

$$\begin{aligned} \mathbf{W}(\boldsymbol{\mu}) &= \begin{bmatrix} \mathbf{e}_1 & \cos(\|\boldsymbol{\mu}\|_2)\mathbf{e}_2 + \sin(\|\boldsymbol{\mu}\|_2)\mathbf{e}_3 \end{bmatrix} \\ \mathbf{V}(\boldsymbol{\mu}) &= \begin{bmatrix} \mathbf{e}_1 & \cos(\|\boldsymbol{\mu}\|_2 + \frac{\pi}{4})\mathbf{e}_2 + \sin(\|\boldsymbol{\mu}\|_2 + \frac{\pi}{4})\mathbf{e}_3 \end{bmatrix}. \end{aligned} \tag{7.12}$$

In this case,

$$\mathbf{A}_r(\boldsymbol{\mu}) = \begin{bmatrix} a_{r11}(\boldsymbol{\mu}) & a_{r12}(\boldsymbol{\mu}) \\ a_{r21}(\boldsymbol{\mu}) & a_{r22}(\boldsymbol{\mu}) \end{bmatrix}, \tag{7.13}$$

with $a_{r11}(\boldsymbol{\mu}) = a_{11}(\boldsymbol{\mu})$, $a_{r12}(\boldsymbol{\mu}) = \cos(\|\boldsymbol{\mu}\|_2 + \frac{\pi}{4})a_{12}(\boldsymbol{\mu}) + \sin(\|\boldsymbol{\mu}\|_2 + \frac{\pi}{4})a_{13}(\boldsymbol{\mu})$, $a_{r21}(\boldsymbol{\mu}) = \sqrt{2}\left(\cos(\|\boldsymbol{\mu}\|_2)a_{21}(\boldsymbol{\mu}) + \sin(\|\boldsymbol{\mu}\|_2)a_{31}(\boldsymbol{\mu})\right)$ and

$$\begin{aligned} a_{r22}(\boldsymbol{\mu}) = \sqrt{2}\Big[\cos(\|\boldsymbol{\mu}\|_2 + \frac{\pi}{4})\Big(\cos(\|\boldsymbol{\mu}\|_2)a_{22}(\boldsymbol{\mu}) + \sin(\|\boldsymbol{\mu}\|_2)a_{32}(\boldsymbol{\mu})\Big) \\ + \sin(\|\boldsymbol{\mu}\|_2 + \frac{\pi}{4})\Big(\cos(\|\boldsymbol{\mu}\|_2)a_{23}(\boldsymbol{\mu}) + \sin(\|\boldsymbol{\mu}\|_2)a_{33}(\boldsymbol{\mu})\Big)\Big]. \end{aligned} \tag{7.14}$$

Hence, $\mathbf{A}_r$ continuously depends on the parameter vector $\boldsymbol{\mu}$ when $\mathbf{A}$ also depends continuously on $\boldsymbol{\mu}$. This chapter focuses then on solving the following two problems.

7.2 Problems Formulation

Let

$$\left\{ \left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl}\right)\right\}_{l=1}^{N_R} = \left\{ \left(\mathbf{A}_r(\boldsymbol{\mu}_l), \mathbf{B}_r(\boldsymbol{\mu}_l), \mathbf{C}_r(\boldsymbol{\mu}_l)\right)\right\}_{l=1}^{N_R}$$

denote a sequence of triplets defining ROMs at multiple values of the parameter $\boldsymbol{\mu}$. These ROMs are constructed by Petrov-Galerkin projection of their full-order counterparts as derived in Eq. (2.12)–(2.14). It is assumed that the underlying full-order operators as well as the subspaces $\mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu})$ and $\mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu})$ depend continuously on $\boldsymbol{\mu}$. However, neither the full-order operators $\left(\mathbf{A}(\boldsymbol{\mu}_l), \mathbf{B}(\boldsymbol{\mu}_l), \mathbf{C}(\boldsymbol{\mu}_l)\right)$ nor the sets of reduced-order bases $\mathbf{W}(\boldsymbol{\mu}_l)$ and $\mathbf{V}(\boldsymbol{\mu}_l)$ are fully known[1].

Problem P1. Let $\boldsymbol{\mu}_{N_R+1} \neq \boldsymbol{\mu}_l$, $l = 1, \cdots, N_R$, define a new value of the parameter. Compute by interpolation a triplet $\left(\mathbf{A}_{rN_R+1}, \mathbf{B}_{rN_R+1}, \mathbf{C}_{rN_R+1}\right)$ defining a ROM

---

[1]Hence, the respective dimensions of the quantities of interest do not depend on the dimension of the full-order state space n, guarantying the real-time property of the proposed methods.

at the new operating point $\boldsymbol{\mu}_{N_R+1}$.

Problem P2. Let $\boldsymbol{\mu}_{N_R+1} \neq \boldsymbol{\mu}_l$, $l = 1, \cdots, N_R$, define a new value of the parameter. Assume also that the quantities $\{\mathbf{P}_{i,j}\}_{i,j=1}^{N_R} = \{\mathbf{V}(\boldsymbol{\mu}_i)^T \mathbf{V}(\boldsymbol{\mu}_j)\}_{i,j=1}^{N_R}$ have been precomputed. Compute by interpolation a triplet $\left(\mathbf{A}_{rN_R+1}, \mathbf{B}_{rN_R+1}, \mathbf{C}_{rN_R+1}\right)$ defining a ROM at the new operating point $\boldsymbol{\mu}_{N_R+1}$.[2]

7.3 Interpolation Method for First-Order Systems

The proposed interpolation method proceeds in two steps. First, for each pre-computed ROM $\left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl}\right)$, $l = 1, \cdots, N_R$, one element $R_l = R(\boldsymbol{\mu}_l)$, $l = 1, \cdots, N_R$ of its orbit $\mathcal{F}\left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl}\right)$ is selected using the method presented in section 7.3.1. This step amounts to congruence transformations of the pre-computed ROMs and will be denoted as Step A in the remainder of this paper. The resulting reduced-order models are then interpolated at the new point $\boldsymbol{\mu}_{N_R+1}$ using a suitable interpolation scheme. This interpolation step will be referred to as Step B and developed in section 7.3.2.

7.3.1 Step A: Congruence Transformations

As shown in section 7.1.1, a given ROM can be expressed in a variety of equivalent bases. However, it was demonstrated in section 7.1.2 that the quality of the interpolation depends crucially on the choice of the representative element inside each equivalent class. Hence, one must produce a discriminative criteria in order to determine this representative element.

---

[2]$\mathbf{P}_{i,j} \in \mathbb{R}^{k \times k}$ $\forall i, j \in 1, \cdots, N_R$ are small size matrices, and as such, the complexity of any algorithm for solving Problem P2 will not scale with the dimension n of the full state-space either.

Problem P1

A heuristic procedure is developed to solve this problem. First, an index $1 \leq l_0 \leq N_R$ is chosen and the corresponding pre-computed ROM $\left(\mathbf{A}_{rl_0}, \mathbf{B}_{rl_0}, \mathbf{C}_{rl_0} \right)$ considered as a reference representative element of the orbit $\mathcal{F}\left(\mathbf{A}_{rl_0}, \mathbf{B}_{rl_0}, \mathbf{C}_{rl_0} \right)$. This element will be chosen as being non-zero and denoted as R_{l_0} in the following.

Then, for $l = 1, \cdots, N_R$, $l \neq l_0$, among all elements of the orbit $\mathcal{F}\left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl} \right)$, a representative element R_l is chosen as being, in some well-chosen metric, the closest to the element R_{l_0}. In other words, the following sequence of problems is solved

$$\min_{R \in \mathcal{F}(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl})} \| R - R_{l_0} \|^2, \quad l = 1, \cdots, N_R. \tag{7.15}$$

The norm $\| \ \|$ is here chosen so that it characterizes the relative discrepancy between the reduced-order linear operators featured in R and R_{l_0}. As such, this norm is dependent on the ROM of reference R_{l_0} and is denoted as $\| \ \|_{R_{l_0}}$. It is subsequently defined.

Definition. Let $R_0 = \left(\mathbf{A}_{r0}, \mathbf{B}_{r0}, \mathbf{C}_{r0} \right) \in \mathcal{R}(p, k, q)$ such that $\mathbf{A}_{r0} \neq \mathbf{0}_{k,k}$, $\mathbf{B}_{r0} \neq \mathbf{0}_{k,p}$ and $\mathbf{C}_{r0} \neq \mathbf{0}_{q,k}$. $\| \ \|_{R_0}$ is defined as follows: $\forall R = \left(\mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r \right) \in \mathcal{R}(p, k, q)$,

$$\| R \|_{R_0}^2 = \alpha \| \mathbf{A}_r \|_F^2 + \beta \| \mathbf{B}_r \|_F^2 + \gamma \| \mathbf{C}_r \|_F^2, \tag{7.16}$$

where $\| \ \|_F$ denotes the Frobenius norm and

$$\alpha = \frac{1}{\| \mathbf{A}_{r0} \|_F^2}, \quad \beta = \frac{1}{\| \mathbf{B}_{r0} \|_F^2}, \quad \gamma = \frac{1}{\| \mathbf{C}_{r0} \|_F^2}. \tag{7.17}$$

From the definition of the orbit $\mathcal{F}\left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl} \right)$ in Eq. (7.4), the minimization

problem (7.15) is therefore

$$\min_{\mathbf{Q}\in\mathrm{O}(k)} \alpha\|\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q} - \mathbf{A}_{rl_0}\|_F^2 + \beta\|\mathbf{Q}^T\mathbf{B}_{rl} - \mathbf{B}_{rl_0}\|_F^2 + \gamma\|\mathbf{C}_{rl}\mathbf{Q} - \mathbf{C}_{rl_0}\|_F^2, \qquad (7.18)$$

for $l = 1, \cdots, N_R$. This minimization problem can be simplified further as shown by the following property.

Proposition 3. The minimization problem (7.18) is equivalent to

$$
\begin{aligned}
\max_{\mathbf{Q}\in\mathrm{O}(k)} F_2(\mathbf{Q}) &= \max_{\mathbf{Q}\in\mathrm{O}(k)} \alpha\,\mathrm{tr}\Big(\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T\Big) + \mathrm{tr}\Big((\beta\mathbf{B}_{rl_0}\mathbf{B}_{rl}^T + \gamma\mathbf{C}_{rl_0}^T\mathbf{C}_{rl})\mathbf{Q}\Big) \\
&= \max_{\mathbf{Q}\in\mathrm{O}(k)} \alpha\langle\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}, \mathbf{A}_{rl_0}\rangle + \langle\beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}, \mathbf{Q}\rangle,
\end{aligned}
\qquad (7.19)
$$

where

$$\langle\mathbf{M}, \mathbf{N}\rangle = \mathrm{tr}(\mathbf{M}^T\mathbf{N}), \quad \mathbf{M}, \mathbf{N} \in \mathbb{R}^{m\times n}, \qquad (7.20)$$

denotes the scalar product associated with the Frobenius norm.

Proof. A proof is offered in Appendix K.2. □

This leads to Algorithm 3.

Algorithm 3 Algorithm for solving Problem P1

1: **for** $l = 1, \cdots, N_R$ **do**
2:    Compute $\mathbf{Q}_l^\star = \arg\max_{\mathbf{Q}\in\mathrm{O}(k)} \alpha\langle\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}, \mathbf{A}_{rl_0}\rangle + \langle\beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}, \mathbf{Q}\rangle$.
3:    Set $\tilde{R}_l = \alpha\,(\mathbf{Q}_l^\star, (\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl}))$.
4: **end for**

In order to solve the constrained minimization problem (7.18), several approaches are possible. The first one is to use classical methods for constrained optimization —see for instance [71] for an overview of common methods. One should note that the orthogonal matrix $\mathbf{Q}$ belongs to $\mathbb{R}^{k\times k}$, and hence there are k^2 unknowns, while

the orthogonality condition on $\mathbf{Q}$ leads to $\dfrac{k(k+1)}{2}$ independent equality constraints.

The second approach is to use an optimization algorithm on the matrix manifold $\mathrm{O}(k)$. This approach has the advantage of transforming a minimization problem in which $\mathbf{Q} \in \mathrm{O}(k)$ is a constraint into an *unconstrained* optimization problem. Counterparts to algorithms for unconstrained optimization on $\mathbb{R}^{m \times n}$ can be derived for matrix manifolds. A theoretical framework and associated algorithms are presented in [66]. In the present case, an approach based on the nonlinear conjugate gradient method on the matrix manifold $\mathrm{O}(k)$ is chosen and developed in section 7.4.1.

The problem of maximizing the first term in Eq. (7.19) has been extensively studied in the literature [113, 114]. This first term defines a correlation criteria between the matrices $\mathbf{A}_{rl}$ and $\mathbf{A}_{rl_0}$. A solution to that problem developed in Ref. [113], consists of defining an iterative algorithm, the fixed points of this algorithm being the critical points of the maximization problem. This third approach is developed in section 7.4.2 and extended to solve the maximization problem (7.19).

Problem P2

In Problem P2, more information about the pre-computed reduced-order models is known, namely the quantities

$$\mathbf{P}_{i,j} = \mathbf{V}(\boldsymbol{\mu}_i)^T \mathbf{V}(\boldsymbol{\mu}_j). \tag{7.21}$$

These matrices provide information about the relative configurations of the pre-computed reduced-order bases and, as such the subspace they each define. As such, transforming the pre-computed ROMs to consistent sets of coordinates is an easier task. As in the previous section, a precomputed ROM $l_0 \in 1, \cdots, N_R$, is considered

as defining the reference set of coordinates. The following sequence of problems is then considered for $l \neq l_0$:

$$\min_{R \in \mathcal{F}(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl})} \|\mathbf{V}(R) - \mathbf{V}(R_{l_0})\|_F^2, \tag{7.22}$$

$\mathbf{V}(R_{l_0})$ denotes $\mathbf{V}(\boldsymbol{\mu}_{l_0})$. In other words, for a given ROM $R \in \mathcal{R}(p, k, q)$, $\mathbf{V}(R) \in \mathcal{ST}(k, n)$ denotes its associated test basis. As such, if R belongs to $\mathcal{F}(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl})$, there exists a matrix $\mathbf{Q} \in \mathrm{O}(k)$ such that $R = \alpha\left(\mathbf{Q}, (\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl})\right)$ and

$$\mathbf{V}(R) = \mathbf{V}\left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl}\right)\mathbf{Q} = \mathbf{V}(\boldsymbol{\mu}_l)\mathbf{Q}. \tag{7.23}$$

The minimization problem (7.22) can hence also be written as

$$\min_{\mathbf{Q} \in \mathrm{O}(k)} \|\mathbf{V}(\boldsymbol{\mu}_l)\mathbf{Q} - \mathbf{V}(\boldsymbol{\mu}_{l_0})\|_F^2, \tag{7.24}$$

which is equivalent to

$$\max_{\mathbf{Q} \in \mathrm{O}(k)} \mathrm{tr}\left(\mathbf{Q}^T\mathbf{V}(\mu_l)^T\mathbf{V}(\mu_{l_0})\right) = \max_{\mathbf{Q} \in \mathrm{O}(k)} \langle \mathbf{Q}, \mathbf{P}_{l,l_0} \rangle. \tag{7.25}$$

Proposition 4. The solution to the maximization problem (7.25) is given by

$$\mathbf{Q}_{l,l_0}^{\star} = \mathbf{U}_{l,l_0}\mathbf{Z}_{l,l_0}^T \tag{7.26}$$

where $\mathbf{U}_{l,l_0}\boldsymbol{\Sigma}_{l,l_0}\mathbf{Z}_{l,l_0}^T$ is a singular value decomposition of $\mathbf{P}_{l,l_0}$.

Proof. This is the classical orthogonal Procrustes problem. A derivation of the optimum for that problem can be found in [35]. □

This leads to Algorithm 4.

Algorithm 4 Algorithm for solving Problem P2

1: **for** $l = 1, \cdots, N_R$ **do**
2:    Compute $\mathbf{P}_{l,l_0} = \mathbf{U}_{l,l_0} \boldsymbol{\Sigma}_{l,l_0} \mathbf{Z}_{l,l_0}^T$ (SVD).
3:    Compute $\mathbf{Q}_{l,l_0}^\star = \mathbf{U}_{l,l_0} \mathbf{Z}_{l,l_0}^T$.
4:    Set $\tilde{R}_l = \alpha \left(\mathbf{Q}_{l,l_0}^\star, (\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl}) \right)$.
5: **end for**

This algorithm has deep connections with mode tracking procedures based on the Modal Assurance Criteria (MAC) [106]. The MAC between two modes $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ is

$$\text{MAC}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \frac{|\boldsymbol{\phi}^T \boldsymbol{\psi}|^2}{(\boldsymbol{\phi}^T \boldsymbol{\phi})(\boldsymbol{\psi}^T \boldsymbol{\psi})}. \tag{7.27}$$

When $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are normalized, $\text{MAC}(\boldsymbol{\phi}, \boldsymbol{\psi}) = |\boldsymbol{\phi}^T \boldsymbol{\psi}|^2$. As such, in the present case, $\mathbf{P}_{l,l_0}$ is the matrix of square roots of the MACs between the modes contained in $\mathbf{V}(\boldsymbol{\mu}_l)$ and $\mathbf{V}(\boldsymbol{\mu}_{l_0})$ respectively. In addition to being able to track modes, —which is necessary when mode crossing occurs [115, 116]— the present method has also the capacity to detect situations where reference modes are combined as it is the case with mode veering phenomena [117]. In Ref. [116], the authors choose the closeness to identity of the matrix $\mathbf{P}_{l,l_0}$ obtained after mode tracking as a measure of consistency between modes at the configurations l and l_0. In the present case, explicit mode tracking is not necessary since the closeness of $\boldsymbol{\Sigma}_{l,l_0}$ to the identity matrix is a natural measure of consistency between modes: $\boldsymbol{\Sigma}_{l,l_0}$ contains the cosines of the principal angles [35, 80] between the subspaces defined by $\mathbf{V}(\boldsymbol{\mu}_l)$ and $\mathbf{V}(\boldsymbol{\mu}_{l_0})$.

7.3.2   Step B: Interpolation on Matrix Manifolds

After congruence transformations of the pre-computed ROM in Step A, these transformed ROMs R_l, $l = 1, \cdots, N_R$ are interpolated. The interpolation method should

preserve certain properties of the elements of the triplet $R_l = (\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl})$. Among these properties, one could mention the non-singularity or the symmetry positive definiteness (SPD) of the operators $\mathbf{A}_{rl}$. The multivariate algorithm for interpolation on matrix manifolds developed in chapter 3 is then sequently applied to each set $\{\mathbf{A}_{rl}\}_{l=1}^{N_R}$, $\{\mathbf{B}_{rl}\}_{l=1}^{N_R}$ and $\{\mathbf{C}_{rl}\}_{l=1}^{N_R}$.

7.4 Numerical Algorithms

7.4.1 Optimization on a Matrix Manifold

The goal is to compute a local minimizer of the function

$$F_1(\mathbf{Q}) = \alpha\|\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q} - \mathbf{A}_{rl_0}\|_F^2 + \beta\|\mathbf{Q}^T\mathbf{B}_{rl} - \mathbf{B}_{rl_0}\|_F^2 + \gamma\|\mathbf{C}_{rl}\mathbf{Q} - \mathbf{C}_{rl_0}\|_F^2, \quad (7.28)$$

defined on the manifold $\mathrm{O}(k)$ of orthogonal matrices of size k.

An approach based on the nonlinear conjugate gradient-class of algorithm is proposed as follows. The advantage of such an approach is that it usually presents good convergence properties while requiring only first-order information about the cost function. For more details on optimization algorithms on matrix manifolds, the reader is referred to References [66, 71].

In order to be able to define an optimization algorithm on a matrix Riemannian manifold, a local inner product at any point $\mathbf{Q}$ of the manifold has to be defined. It is here chosen as the one inherited from the product in the embedding space $\mathbb{R}^{k \times k}$:

$$\langle \mathbf{X}, \mathbf{Z} \rangle_{\mathbf{Q}} = \mathrm{tr}(\mathbf{X}^T\mathbf{Z}), \quad (7.29)$$

where $\mathbf{X}$ and $\mathbf{Z}$ are two square matrices of size k belonging to the tangent space to

the manifold $O(k)$ at $\mathbf{Q}$. This tangent space is defined as

$$T_{\mathbf{Q}}O(k) = \{\mathbf{Q\Omega} \mid \mathbf{\Omega} + \mathbf{\Omega}^T = \mathbf{0}_{k,k}\}. \tag{7.30}$$

A cost function $\bar{F}_1$ can be defined on $\mathbb{R}^{k \times k}$ as an extension of F_1. The gradient of $\bar{F}_1$ is

$$\mathrm{grad}\bar{F}_1(\mathbf{Q}) = -2\alpha(\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}\mathbf{A}_{rl_0}) - 2(\beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}). \tag{7.31}$$

F_1 being the restriction of $\bar{F}_1$ to the $O(k)$, its gradient $\mathrm{grad}F_1(\mathbf{Q})$ on the submanifold is defined as [66]

$$\mathrm{grad}F_1(\mathbf{Q}) = \mathrm{P}_{\mathbf{Q}}\mathrm{grad}\bar{F}_1(\mathbf{Q}), \tag{7.32}$$

where $\mathrm{P}_{\mathbf{Q}}$ denotes the orthogonal projection operator onto the tangent space $T_{\mathbf{Q}}O(k)$. It is defined as

$$\mathrm{P}_{\mathbf{Q}}\mathbf{M} = \mathbf{Q}\ \mathrm{skew}(\mathbf{Q}^T\mathbf{M}),\ \forall \mathbf{M} \in \mathbb{R}^{k \times k}, \tag{7.33}$$

$\mathrm{skew}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} - \mathbf{A}^T)$ denoting the skew-symmetric part of a square matrix $\mathbf{A}$.

Furthermore, the manifold $O(k)$ being a non-linear manifold, given a search direction $\boldsymbol{\eta}_i \in T_{\mathbf{Q}_i}O(k)$ and a step size $\alpha_i \in \mathbb{R}$, updates of iterates $\mathbf{Q}_i \in O(k)$, $i = 0, \cdots$ cannot be effected by simple linear combinations of the type

$$\mathbf{Q}_{i+1} = \mathbf{Q}_i + \alpha_i\boldsymbol{\eta}_i, \tag{7.34}$$

since the linear combination $\mathbf{Q}_{i+1}$ is not guaranteed by construction to belong to the nonlinear manifold $O(k)$. Instead, the updates are based on *retractions*, which are mappings $R_{\mathbf{Q}}$, $\mathbf{Q} \in O(k)$ from the tangent space $T_{\mathbf{Q}}O(k)$ to $O(k)$ [66]. The update

formula analog to (7.34) is

$$\mathbf{Q}_{i+1} = R_{\mathbf{Q}_i}(\alpha_i \boldsymbol{\eta}_i). \tag{7.35}$$

In that case, $\mathbf{Q}_{i+1}$ belongs to O(k). There are several choices for $R_{\mathbf{Q}}$ [66]. In the present case, it is chosen as follows.

Definition. Let $\boldsymbol{\eta}$ be a vector of $T_{\mathbf{Q}}$O(k). By definition of the tangent space, $\boldsymbol{\eta}$ can be written under the form $\boldsymbol{\eta} = \mathbf{Q}\boldsymbol{\Omega}$ where $\boldsymbol{\Omega}$ is a skew-symmetric matrix. Denoting by $\mathbf{qf}$ the mapping that associates to every matrix the Q factor of its QR factorization, one can define a retraction $R_{\mathbf{Q}}$ as

$$R_{\mathbf{Q}}(\boldsymbol{\eta}) = R_{\mathbf{Q}}(\mathbf{Q}\boldsymbol{\Omega}) = \mathbf{Q} \ \mathbf{qf}(\mathbf{I}_k + \boldsymbol{\Omega}). \tag{7.36}$$

In the classical nonlinear conjugate gradient algorithms, updates for the search direction are of the form

$$\boldsymbol{\eta}_{i+1} = -\mathrm{grad}F_1(\mathbf{Q}_{i+1}) + \beta_{i+1}\boldsymbol{\eta}_i, \tag{7.37}$$

where $\beta_{i+1} \in \mathbb{R}$. However, one issue that arises when transposing this formula to matrix manifolds is the fact that $\mathrm{grad}F_1(\mathbf{Q}_{i+1})$ belongs to the tangent space to O(k) at $\mathbf{Q}_{i+1}$ while $\boldsymbol{\eta}_i$ belongs to the tangent space at $\mathbf{Q}_i$. $\mathbf{Q}_i$ and $\mathbf{Q}_{i+1}$ being in general distinct, their tangent spaces are distinct as well and the formula is ill-posed. To remediate this issue, Eq. (7.37) is replaced by the following update

$$\boldsymbol{\eta}_{i+1} = -\mathrm{grad}F_1(\mathbf{Q}_{i+1}) + \beta_{i+1}\mathcal{T}_{\alpha_i\boldsymbol{\eta}_i}(\boldsymbol{\eta}_i). \tag{7.38}$$

$\mathcal{T}$ is a *vector transport* associated to the retraction $R_{\mathbf{Q}}$ that takes two elements $\boldsymbol{\eta}, \boldsymbol{\xi}$ of $T_{\mathbf{Q}}$O(k) as arguments and returns an element of $T_{R_{\mathbf{Q}}(\boldsymbol{\eta})}$O($k$). This element can

be identified to the vector $\boldsymbol{\xi}$ transported along the path defined by the retraction $\{R_{\mathbf{Q}}(t\boldsymbol{\eta}),\ t \in [0,1]\}$. Hence, $\mathcal{T}_{\alpha_i\boldsymbol{\eta}_i}(\boldsymbol{\eta}_i)$ belongs to $T_{R_{\mathbf{Q}_i}(\alpha_i\boldsymbol{\eta}_i)}\mathrm{O}(k) = T_{\mathbf{Q}_{i+1}}\mathrm{O}(k)$ and the update formula (7.38) is well-posed. $\mathcal{T}$ is defined as follows:

Definition. Let $\boldsymbol{\eta}, \boldsymbol{\xi} \in T_{\mathbf{Q}}\mathrm{O}(k)$. Then

$$\mathcal{T}_{\boldsymbol{\eta}}(\boldsymbol{\xi}) = \mathrm{P}_{R_{\mathbf{Q}}(\boldsymbol{\eta})}\boldsymbol{\xi}, \tag{7.39}$$

where $R_{\mathbf{Q}}$ is the retraction defined above and $\mathrm{P}_{R_{\mathbf{Q}}(\boldsymbol{\eta})}$ is the orthogonal projection onto $T_{R_{\mathbf{Q}}(\boldsymbol{\eta})}\mathrm{O}(k)$. Hence

$$\mathcal{T}_{\boldsymbol{\eta}}(\boldsymbol{\xi}) = R_{\mathbf{Q}}(\boldsymbol{\eta})\ \mathrm{skew}(R_{\mathbf{Q}}(\boldsymbol{\eta})^T\boldsymbol{\xi}). \tag{7.40}$$

The formula for the parameter β_{i+1} is necessary to completely define the method. In the present work, an extension of the Fletcher-Reeves method [71] to the case of non-linear manifolds is considered [66]:

$$\beta_{i+1} = \frac{\langle \mathrm{grad}F_1(\mathbf{Q}_{i+1}), \mathrm{grad}F_1(\mathbf{Q}_{i+1})\rangle_{\mathbf{Q}_{i+1}}}{\langle \mathrm{grad}F_1(\mathbf{Q}_i), \mathrm{grad}F_1(\mathbf{Q}_i)\rangle_{\mathbf{Q}_i}}. \tag{7.41}$$

The following Fletcher-Reeves method on the matrix manifold $\mathrm{O}(k)$ is then defined.

Algorithm 5 Nonlinear Conjugate Gradient Algorithm

1: Choose an initial iterate $\mathbf{Q}_0 \in \mathrm{O}(k)$.
2: Compute $\boldsymbol{\eta}_0 = -\mathrm{grad}F_1(\mathbf{Q}_0)$.
3: **for** $i = 0, \cdots$ **do**
4:    Compute a step size α_i.
5:    $\mathbf{Q}_{i+1} = R_{\mathbf{Q}_i}(\alpha_i\boldsymbol{\eta}_i)$.
6:    Compute $\beta_{i+1} = \frac{\langle \mathrm{grad}F_1(\mathbf{Q}_{i+1}), \mathrm{grad}F_1(\mathbf{Q}_{i+1})\rangle_{\mathbf{Q}_{i+1}}}{\langle \mathrm{grad}F_1(\mathbf{Q}_i), \mathrm{grad}F_1(\mathbf{Q}_i)\rangle_{\mathbf{Q}_i}}$.
7:    $\boldsymbol{\eta}_{i+1} = -\mathrm{grad}F_1(\mathbf{Q}_{i+1}) + \beta_{i+1}\mathcal{T}_{\alpha_i\boldsymbol{\eta}_i}(\boldsymbol{\eta}_i)$.
8: **end for**

Since $O(k)$ has two connected components —the orthogonal matrices of determinant $+1$ and -1, respectively— the proposed algorithm is run twice, starting with an initial iterate from each connected component. The two final minimizing solutions are then compared and the minimizer over both components is considered as the solution of the minimization problem over $O(k)$.

7.4.2 Fixed Point Iteration Method

Adapting the algorithm developed in Ref. [113] to the present case, the iterative algorithm is here based on an affine map defined as

$$\mathbf{M}_s(\mathbf{Q}) = \alpha(\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}\mathbf{A}_{rl_0}) + s\mathbf{Q} + \beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}, \qquad (7.42)$$

where s is a fixed real parameter chosen such that $s > s_{\min}$ with

$$s_{\min} = 2\alpha\|\mathbf{A}_{rl}\|_2\|\mathbf{A}_{rl_0}\|_2 + \|\beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}\|_2. \qquad (7.43)$$

Defining the parameter s is necessary to prove that the fixed points of the proposed iterative algorithm are exactly the critical points of the maximization problem (see Theorem 1).

The algorithm then proceeds by solving at each iteration $i = 0, \cdots$, the maximization problem

$$\mathbf{Q}_{i+1} = \arg\max_{\mathbf{Q}}\langle\mathbf{Q}, \mathbf{M}_s(\mathbf{Q}_i)\rangle. \qquad (7.44)$$

The solution to this problem is established in the following lemma, leading to the proposed iterative algorithm presented in Algorithm 6.

Lemma. Let the singular value decomposition of $\mathbf{M}_s(\mathbf{Q}_i)$ be

$$\mathbf{M}_s(\mathbf{Q}_i) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{7.45}$$

with $\mathbf{U} \in \mathrm{O}(k)$, $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ a diagonal matrix and $\mathbf{V} \in \mathrm{O}(k)$. Then

$$\max_{\mathbf{Q} \in \mathrm{O}(k)} \langle \mathbf{Q}, \mathbf{M}_s(\mathbf{Q}_i)\rangle = \sum_{j=1}^{k} \sigma_j(\mathbf{M}_s(\mathbf{Q}_i)), \tag{7.46}$$

where $\{\sigma_j(\mathbf{M}_s(\mathbf{Q}_i))\}_{j=1}^{k}$ denotes the set of singular values of $\mathbf{M}_s(\mathbf{Q}_i)$. The solution to the maximization problem is unique and equal to $\mathbf{U}\mathbf{V}^T$.

Proof. This lemma is a consequence of Property 4. A proof is offered in [113] in a more general setting. $\square$

Algorithm 6 Iterative algorithm

1: Define a parameter $s > s_{\min}$.
2: Compute $\mathbf{G} = \beta \mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma \mathbf{C}_{rl}^T \mathbf{C}_{rl_0}$.
3: Choose an initial matrix $\mathbf{Q}_0 \in \mathrm{O}(k)$.
4: **for** $i = 0, \cdots$ **do**
5: $\quad \mathbf{U}_{i+1}\mathbf{\Sigma}_{i+1}\mathbf{V}_{i+1}^T = \alpha(\mathbf{A}_{rl}\mathbf{Q}_i\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}_i\mathbf{A}_{rl_0}) + s\mathbf{Q}_i + \mathbf{G}$ {Compute an SVD}
6: $\quad \mathbf{Q}_{i+1} = \mathbf{U}_{i+1}\mathbf{V}_{i+1}^T$
7: **end for**

In order to show that the fixed points of the recursive algorithm are the critical points of F_2, one needs to characterize these critical points. This is done in the following theorem.

Theorem 1. The critical points $\mathbf{Q}^\star$ of F_2 are orthogonal matrices satisfying the identity

$$\mathbf{Q}^\star\mathbf{S} = \alpha(\mathbf{A}_{rl}\mathbf{Q}^\star\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}^\star\mathbf{A}_{rl_0}) + \beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}, \tag{7.47}$$

where $\mathbf{S}$ is a symmetric matrix.

Proof. A proof is offered in Appendix K.3. $\qquad\qquad\square$

Theorem 2. The set of the fixed points of the recursive algorithm is exactly the set of the critical points of F_2.

Proof. A proof of the theorem is presented in Appendix K.4. $\qquad\qquad\square$

7.5 Extension to Second-Order Systems

7.5.1 Structural Systems

Second-order parametric Ordinary Differential Equations arising from the discretiza-
tion of parameterized linear structural mechanics systems can be written under the
form (see section 5.2)

$$\mathbf{M}(\boldsymbol{\mu})\frac{d^2\mathbf{x}}{dt^2}(t) + \mathbf{D}(\boldsymbol{\mu})\frac{d\mathbf{x}}{dt}(t) + \mathbf{K}(\boldsymbol{\mu})\mathbf{x}(t) \quad = \quad \mathbf{B}(\boldsymbol{\mu})\mathbf{u}(t) \qquad (7.48)$$

$$\mathbf{y}(t) \quad = \quad \mathbf{C}(\boldsymbol{\mu})\mathbf{x}(t), \qquad (7.49)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the time-depend vector of structural displacements, $\mathbf{M}$ is the
mass matrix, $\mathbf{D}$ the linear damping matrix and $\mathbf{K}$ the stiffness matrix. These $n \times n$
real matrices are assumed to be symmetric positive definite and as such belong to
the corresponding matrix manifold $\mathrm{SPD}(n)$. The matrices $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$, $\mathbf{B}$ and $\mathbf{C}$ are
parameterized by $\boldsymbol{\mu} \in \mathbb{R}^{N_p}$.

Equations (7.48)–(7.49) are here subsequently orthogonally projected by the Galerkin
method onto a subspace $\mathcal{S}(\boldsymbol{\mu}) \in \mathcal{G}(k, n)$ spanned by the columns of a full-column rank
Reduced-Order Basis $\mathbf{V}(\boldsymbol{\mu}) \in \mathcal{ST}(k, n)$. This is a special case of the Petrov-Galerkin

method presented in section 2.2.2 with $\mathbf{W}(\boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})$. The extension to projection using different test and trial bases for second-order systems is straightforward. $\mathbf{x}(t)$ is then approximated as

$$\mathbf{x}(t) \approx \mathbf{V}(\boldsymbol{\mu})\mathbf{x}_r(t), \ \forall t \in [T_0, T_f], \tag{7.50}$$

and the reduced dynamical system reads

$$\mathbf{M}_r(\boldsymbol{\mu})\frac{d^2\mathbf{x}_r}{dt^2}(t) + \mathbf{D}_r(\boldsymbol{\mu})\frac{d\mathbf{x}_r}{dt}(t) + \mathbf{K}_r(\boldsymbol{\mu})\mathbf{x}_r(t) = \mathbf{B}_r(\boldsymbol{\mu})\mathbf{u}(t) \tag{7.51}$$

$$\mathbf{y}_r(t) = \mathbf{C}_r(\boldsymbol{\mu})\mathbf{x}_r(t), \tag{7.52}$$

where the reduced operators are defined as in section 5.2. $\mathbf{M}_r(\boldsymbol{\mu})$, $\mathbf{D}_r(\boldsymbol{\mu})$ and $\mathbf{K}_r(\boldsymbol{\mu})$ belong to the manifold $\mathrm{SPD}(k)$ of symmetric positive matrices of size k. In the sequel, $\mathcal{P}(p, k, q)$ is defined as the set of quintuplets defining second-order ROMs with an input-space, a state-space and an output-space of respective dimensions p, k and q:

$$\mathcal{P}(p, k, q) = \{(\mathbf{M}_r, \mathbf{D}_r, \mathbf{K}_r, \mathbf{B}_r, \mathbf{C}_r), \ |$$
$$\mathbf{M}_r \in \mathrm{SPD}(k), \ \mathbf{D}_r \in \mathrm{SPD}(k), \ \mathbf{K}_r \in \mathrm{SPD}(k), \ \mathbf{B}_r \in \mathbb{R}^{k \times p}, \ \mathbf{C}_r \in \mathbb{R}^{q \times k}\} \tag{7.53}$$

Similarly to the first-order systems case, a group action α_2 is considered and to each element $(\mathbf{M}_r, \mathbf{D}_r, \mathbf{K}_r, \mathbf{B}_r, \mathbf{C}_r) \in \mathcal{P}(p, k, q)$ is associated an orbit

$$\mathcal{F}_2\left(\bar{\mathbf{M}}_r, \bar{\mathbf{D}}_r, \bar{\mathbf{K}}_r, \bar{\mathbf{B}}_r, \bar{\mathbf{C}}_r\right) = \left\{\alpha_2\left(\mathbf{Q}, (\bar{\mathbf{M}}_r, \bar{\mathbf{D}}_r, \bar{\mathbf{K}}_r, \bar{\mathbf{B}}_r, \bar{\mathbf{C}}_r)\right) \mid \mathbf{Q} \in \mathrm{O}(k)\right\}$$
$$= \left\{(\mathbf{Q}^T\bar{\mathbf{M}}_r\mathbf{Q}, \mathbf{Q}^T\bar{\mathbf{D}}_r\mathbf{Q}, \mathbf{Q}^T\bar{\mathbf{K}}_r\mathbf{Q}, \mathbf{Q}^T\bar{\mathbf{B}}_r, \bar{\mathbf{C}}_r\mathbf{Q}) \mid \mathbf{Q} \in \mathrm{O}(k)\right\}. \tag{7.54}$$

Problem P1 and P2 defined in section 7.2 can then be naturally extended to second-order systems, replacing the triplets $\left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl}\right)$ belonging to $\mathcal{R}(p, k, q)$ by quintuplets $(\mathbf{M}_{rl}, \mathbf{D}_{rl}, \mathbf{K}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl})$, $l = 1, \cdots, N_R + 1$, belonging to $\mathcal{P}(p, k, q)$. Similarly, the algorithms for interpolation of ROMs are extended as follows.

7.5.2 Step A for Problem P1

Choosing an element $l_0 \in 1, \cdots, N_R$, the following sequence of problems is solved

$$\min_{R \in \mathcal{F}_2(\mathbf{M}_{rl}, \mathbf{D}_{rl}, \mathbf{K}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl})} \|R - R_{l_0}\|^2_{R_{l_0}}, \quad l = 1, \cdots, N_R, \tag{7.55}$$

where, for any $R_0 \in \mathcal{P}(p, k, q)$ having non-null elements, $\| \ \|_{R_0}$ is now defined as $\forall R = \left(\mathbf{M}_r, \mathbf{D}_r, \mathbf{K}_r, \mathbf{B}_r, \mathbf{C}_r\right) \in \mathcal{P}(p, k, q)$

$$\|R\|^2_{R_0} = \alpha_M \|\mathbf{M}_r\|^2_F + \alpha_D \|\mathbf{D}_r\|^2_F + \alpha_K \|\mathbf{K}_r\|^2_F + \beta \|\mathbf{B}_r\|^2_F + \gamma \|\mathbf{C}_r\|^2_F, \tag{7.56}$$

with

$$\alpha_M = \frac{1}{\|\mathbf{M}_{rl_0}\|^2_F}, \ \alpha_D = \frac{1}{\|\mathbf{D}_{rl_0}\|^2_F}, \ \alpha_K = \frac{1}{\|\mathbf{K}_{rl_0}\|^2_F}, \ \beta = \frac{1}{\|\mathbf{B}_{rl_0}\|^2_F}, \ \gamma = \frac{1}{\|\mathbf{C}_{rl_0}\|^2_F}. \tag{7.57}$$

If any element of the quintuplet is not specified —for instance $\mathbf{D}_r$ for undamped systems— the corresponding term is dropped from problem (7.55).

7.5.3 Step A for Problem P2

The problem to be solved is the same as in the case of first-order systems.

$$\min_{\mathbf{Q} \in O(k)} \|\mathbf{V}(\mu_l)\mathbf{Q} - \mathbf{V}(\mu_{l_0})\|^2_F, \tag{7.58}$$

Hence, the method developed in section 7.3.1 can be applied directly as is.

7.5.4 Step B

The interpolation algorithm developed in chapter 3 is applied after identifying the matrix manifold each element of the set $\tilde{\mathcal{R}}(\boldsymbol{\mu}_l)$, $l = 1, \cdots, N_R$, respectively belongs to. In the present case,

- $\tilde{\mathbf{M}}_{rl}$, $\tilde{\mathbf{D}}_{rl}$ and $\tilde{\mathbf{K}}_{rl}$ belong to the manifold $\mathrm{SPD}(k)$ of symmetric positive matrices of size k.

- $\tilde{\mathbf{B}}_{rl}$ belongs to the manifold $\mathbb{R}^{k \times p}$ of $k \times p$ real matrices.

- $\tilde{\mathbf{C}}_{rl}$ belongs to the manifold $\mathbb{R}^{q \times k}$ of $q \times k$ real matrices.

7.5.5 Case of Interpolation of Reduced Stiffness Matrices

The following simple case is here considered.

- The structural systems are undamped. Hence the term involving $\mathbf{D}_r$ is dropped from the minimization problem (7.55).

- The mass matrices are lumped, and the mass distribution is uniform, that is

$$\mathbf{M}(\boldsymbol{\mu}) = m(\boldsymbol{\mu})\mathbf{I}_n, \ l = 1, \cdots, N_R. \tag{7.59}$$

Hence the reduced mass matrices are

$$\mathbf{M}_r(\boldsymbol{\mu}) = \mathbf{X}(\boldsymbol{\mu})^T \mathbf{M}(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu}) = m(\boldsymbol{\mu})\mathbf{X}(\boldsymbol{\mu})^T \mathbf{X}(\boldsymbol{\mu}) = m(\boldsymbol{\mu})\mathbf{I}_k, \ l = 1, \cdots, N_R. \tag{7.60}$$

- There are no specific input or output matrices specified. As such, the terms involving $\mathbf{B}_r$ and $\mathbf{C}_r$ are dropped from the minimization problem (7.55) as well.

In this case, there is a very simple analytic solution to Problem P1, developed as follows.

1. Step A.

   Choosing an integer $l_0 \in 1, \cdots, N_R$, for a given a fixed $l = 1, \cdots, N_R$, Problem (7.56) becomes

$$\min_{\mathbf{Q} \in O(n)} \alpha_K \|\mathbf{Q}^T \mathbf{K}_{rl} \mathbf{Q} - \mathbf{K}_{rl_0}\|_F^2 + \alpha_M \|\mathbf{Q}^T \mathbf{M}_{rl} \mathbf{Q} - \mathbf{M}_{rl_0}\|_F^2, \qquad (7.61)$$

   that is

$$\min_{\mathbf{Q} \in O(n)} \alpha_K \|\mathbf{Q}^T \mathbf{K}_{rl} \mathbf{Q} - \mathbf{K}_{rl_0}\|_F^2 + \alpha_M \|(m(\boldsymbol{\mu}_l) - m(\boldsymbol{\mu}_{l_0}))\mathbf{I}_k\|_F^2. \qquad (7.62)$$

   This is equivalent to the maximization problem

$$\max_{\mathbf{Q} \in O(n)} \mathrm{tr}(\mathbf{Q}^T \mathbf{K}_{rl} \mathbf{Q} \mathbf{K}_{rl_0}). \qquad (7.63)$$

   Since the operators $\mathbf{K}_{rl}$ and $\mathbf{K}_{rl_0}$ are symmetric, there exists a closed form for the optimal solution to this problem [119], subsequently reported for completeness purposes.

   Theorem 3. Let $\mathbf{K}_{rl} = \mathbf{U}_l \boldsymbol{\Lambda}_l \mathbf{U}_l^T$ and $\mathbf{K}_{rl_0} = \mathbf{U}_{l_0} \boldsymbol{\Lambda}_{l_0} \mathbf{U}_{l_0}^T$ be two eigenvalue decompositions with the eigenvalues ordered decreasingly. Then[3]

$$\mathbf{Q}_l = \mathbf{U}_l \mathbf{U}_{l_0}^T$$

---

[3]When $l = l_0$, $\mathbf{Q}_{l_0} = \mathbf{I}_k$ as expected.

is the optimal value of (7.63).

The operators $\tilde{\mathbf{K}}_{rl}$, $l = 1, \cdots, N_R$, are consequently obtained as

$$\tilde{\mathbf{K}}_{rl} = \mathbf{Q}_l^T \mathbf{K}_{rl} \mathbf{Q}_l = \mathbf{U}_{l_0} \mathbf{U}_l^T \mathbf{K}_{rl} \mathbf{U}_l \mathbf{U}_{l_0}^T = \mathbf{U}_{l_0} \mathbf{\Lambda}_l \mathbf{U}_{l_0}^T, \tag{7.64}$$

and $\tilde{\mathbf{M}}_{rl}$, $l = 1, \cdots, N_R$, as

$$\tilde{\mathbf{M}}_{rl} = \mathbf{Q}_l^T \mathbf{M}_{rl} \mathbf{Q}_l = m(\boldsymbol{\mu}_l) \mathbf{Q}_l^T \mathbf{I}_k \mathbf{Q}_l = m(\boldsymbol{\mu}_l) \mathbf{I}_k. \tag{7.65}$$

2. Step B.

The interpolation algorithm on the tangent space to the manifold of SPD matrices can be followed step-by-step for the sets $\{\tilde{\mathbf{M}}_{rl}\}_{l=1}^{N_R}$ and $\{\tilde{\mathbf{K}}_{rl}\}_{l=1}^{N_R}$. The interpolation of the elements of the set $\{\tilde{\mathbf{M}}_{rl}\}_{l=1}^{N_R} = \{m(\boldsymbol{\mu}_l) \mathbf{I}_k\}_{l=1}^{N_R}$ is straightforward and is not developed here. The elements of the set $\{\tilde{\mathbf{K}}_{rl}\}_{l=1}^{N_R} = \{\mathbf{U}_{l_0} \mathbf{\Lambda}_l \mathbf{U}_{l_0}^T\}_{l=1}^{N_R}$ are interpolated and the following operator $\mathbf{K}_{rN_R+1}$ is obtained. Details of the computations are provided in Appendix K.5.

$$\mathbf{K}_{rN_R+1} = \mathbf{U}_{l_0} \mathbf{\Lambda}_{N_R+1} \mathbf{U}_{l_0}^T, \tag{7.66}$$

where the interpolated eigenvalues operator is

$$\mathbf{\Lambda}_{N_R+1} = \prod_{l=1}^{N_R} \mathbf{\Lambda}_l^{\omega_l}. \tag{7.67}$$

The following three important conclusions can be drawn from this result.

1. The final interpolated matrix does not depend on the choice of the reference point l_0' for the interpolation.

2. Equation (7.67) is equivalent to

$$\log(\mathbf{\Lambda}_{N_R+1}) = \sum_{l=1}^{N_R} \omega_l \log(\mathbf{\Lambda}_l).$$

(7.68)

This shows that the algorithm has interpolated the logarithms of the diagonal matrices of eigenvalues of the operators ordered decreasingly: denoting each diagonal matrix

$$\mathbf{\Lambda}_l = \begin{pmatrix} \lambda_{1,l} & & (0) \\ & \ddots & \\ (0) & & \lambda_{k,l} \end{pmatrix}, l = 1, \cdots, N_R + 1,$$

(7.69)

one obtains

$$\ln(\lambda_{i,N_R+1}) = \sum_{l=1}^{N_R} \omega_l \ln(\lambda_{i,l}), \ i = 1, \cdots, k.$$

(7.70)

Hence, the largest eigenvalues have been interpolated, then the second largest ones, and so on.

3. The choice of the reference element l_0 for Step A has affected the basis $\mathbf{U}_{l_0}$ in which the matrix $\mathbf{K}_{rN_\mu+1}$ is written but not the final interpolated value of its eigenvalues $\mathbf{\Lambda}_{N_\mu+1}$ as shown by Eq. (7.68).

7.6 Applications

The proposed methods are here applied to first-order dynamical systems in section 7.6.1 and to second-order systems in section 7.6.2.

7.6.1   Application to First-Order Systems

Test Example

In this first example, an artificial single-input single-output system depending on a single parameter $\mu \in \mathbb{R}$ is considered. Its objective is to present some of the principal concepts attached to the problem of interest. This example is defined by

$$\mathbf{A}(\mu) = \mathbf{A}_0 + \mu \mathbf{A}_1 \in \mathbb{R}^{3\times 3}, \quad \mathbf{B}(\mu) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{C}(\mu) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}. \tag{7.71}$$

$\mathbf{A}_0$ and $\mathbf{A}_1$ are randomly generated and equal to

$$\mathbf{A}_0 = \begin{bmatrix} 0.54 & 0.86 & -0.43 \\ 1.83 & 0.32 & 0.34 \\ -2.26 & -1.31 & 3.58 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 2.77 & 0.73 & -0.21 \\ -1.35 & -0.06 & -0.12 \\ 3.03 & 0.71 & 1.49 \end{bmatrix}. \tag{7.72}$$

The subspaces spanned by the reduced-order bases are the same ones considered in section 7.1.2, namely

$$\begin{aligned} \mathcal{S}_{\mathbf{W}}(\mu) &= \operatorname{span}\left\{ \mathbf{e}_1, \cos(\mu)\mathbf{e}_2 + \sin(\mu)\mathbf{e}_3 \right\}, \\ \mathcal{S}_{\mathbf{V}}(\mu) &= \operatorname{span}\left\{ \mathbf{e}_1, \cos\left(\mu + \frac{\pi}{4}\right)\mathbf{e}_2 + \sin\left(\mu + \frac{\pi}{4}\right)\mathbf{e}_3 \right\}. \end{aligned} \tag{7.73}$$

Three pre-computed ROMs $\left\{ \left(\mathbf{A}_{rl}, \mathbf{B}_{rl}, \mathbf{C}_{rl} \right) \right\}_{l=1}^{3}$ are considered with $\mu_1 = 0$, $\mu_2 = 0.5$ and $\mu_3 = 1$. These ROMs are built by Petrov-Galerkin projection of their full-order counterparts using the reduced-order bases $\mathbf{W}(\mu_l)$ and $\mathbf{V}(\mu_l)$, $l = 1, 2, 3$, respectively spanning $\mathcal{S}_{\mathbf{W}}(\mu_l)$ and $\mathcal{S}_{\mathbf{V}}(\mu_l)$. These reduced-order bases take the

following form

$$\mathbf{W}(\mu_l) = \begin{bmatrix} \mathbf{e}_1 & \cos(\mu)\mathbf{e}_2 + \sin(\mu)\mathbf{e}_3 \end{bmatrix} \mathbf{P}(\mu_l)$$

$$\mathbf{V}(\mu_l) = \begin{bmatrix} \mathbf{e}_1 & \cos\left(\mu + \frac{\pi}{4}\right)\mathbf{e}_2 + \sin\left(\mu + \frac{\pi}{4}\right)\mathbf{e}_3 \end{bmatrix} \mathbf{Q}(\mu_l),$$

(7.74)

where $\mathbf{P}(\mu_l), \mathbf{Q}(\mu_l) \in \mathrm{O}(2)$. For the sake of simplicity, it is here assumed that $\mathbf{P}(\mu_2) = \mathbf{Q}(\mu_2) = \mathbf{I}_2$. As such the bases of reference are

$$\begin{bmatrix} \mathbf{e}_1 & \cos(\mu)\mathbf{e}_2 + \sin(\mu)\mathbf{e}_3 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{e}_1 & \cos\left(\mu + \frac{\pi}{4}\right)\mathbf{e}_2 + \sin\left(\mu + \frac{\pi}{4}\right)\mathbf{e}_3 \end{bmatrix}, \quad \mu \in \mathbb{R}.$$

(7.75)

$\mathbf{P}(\mu_l), \mathbf{Q}(\mu_l)$, $l = 1, 3$ are randomly generated and in the present case equal to

$$\mathbf{P}(\mu_1) = \begin{bmatrix} -0.6672 & -0.7449 \\ -0.7449 & 0.6672 \end{bmatrix}, \quad \mathbf{Q}(\mu_1) = \begin{bmatrix} -0.9675 & -0.2527 \\ -0.2527 & 0.9675 \end{bmatrix}, \quad (7.76)$$

and

$$\mathbf{P}(\mu_3) = \begin{bmatrix} -0.9864 & -0.1645 \\ -0.1645 & 0.9864 \end{bmatrix}, \quad \mathbf{Q}(\mu_3) = \begin{bmatrix} -0.9097 & -0.4152 \\ -0.4152 & 0.9097 \end{bmatrix}. \quad (7.77)$$

The goal of the following numerical experiment is to show that the proposed approaches are able to recover the expressions of the ROMs in the bases of reference defined in Eq. (7.75), and hence lead to a consistent interpolation procedure. The entries of the matrices constituting the pre-computed ROMs at parameters μ_l, $l = 1, 2, 3$, are reported as black squares in Fig. 7.1. Interpolating in between these quantities would lead to inconsistent approximation as shown with the black dashed lines that do not match the expression of the parametric ROMs in the references bases (blue stars). However, applying Step A of the proposed algorithms enables recovering the

Table 7.1: Comparison of the values of F_1 at the returned iterates

	$\mu_1 = 0$	$\mu_3 = 1$
$F_1(\mathbf{Q}(\mu_l)^T)$	0.77521	0.96854
$F_1(\mathbf{Q}_1^\star(\mu_l))$	0.76454	0.96704
$F_1(\mathbf{Q}_2^\star(\mu_l))$	0.76454	0.96704

expressions of the ROMs in the bases of reference as shown by the green triangles (Approach 1, that is the optimization method) and red dots (Approach 2, that is the fixed point method). Thus, interpolating in between these quantities (green and red lines) lead to very accurate approximations of the reference ROMs denoted by the blue stars. The convergence histories of both approaches are reported in Fig. 7.2.

One can also analyze the solutions obtained using the two proposed approaches. Ideally, one would like to recover the isometry $\mathbf{Q}(\mu_l)^T$, $l = 1, 3$ as solutions of the proposed algorithms. Recalling that in section 7.3.1, an heuristic was developed in order to define the minimization problem, the reader should note that $\mathbf{Q}(\mu_l)^T$ will not be the exact minimizer of that problem but is hopefully close to it. Table 7.1 compares the values of the cost function F_1 at $\mathbf{Q}(\mu_l)^T$ and the solutions $\mathbf{Q}_1^\star(\mu_l)$ and $\mathbf{Q}_2^\star(\mu_l)$ returned by Approaches 1 and 2 respectively. Both approaches return iterates that correspond to the same value of the function F_1. The function F_1 evaluated at $\mathbf{Q}(\mu_l)^T$, $l = 1, 3$ is within respectively 1.4% and 0.2% of the value returned by both algorithms.

Structural System

An example mass-damper-spring system previously studied in Reference [110] is considered. This system is schematically represented in Figure 7.3. Each operating point

(a) $a_{r11}(\mu)$

(b) $a_{r12}(\mu)$

(c) $b_{r1}(\mu)$

(d) $a_{r21}(\mu)$

(e) $a_{r22}(\mu)$

(f) $b_{r2}(\mu)$

(g) $c_{r1}(\mu)$

(h) $c_{r2}(\mu)$
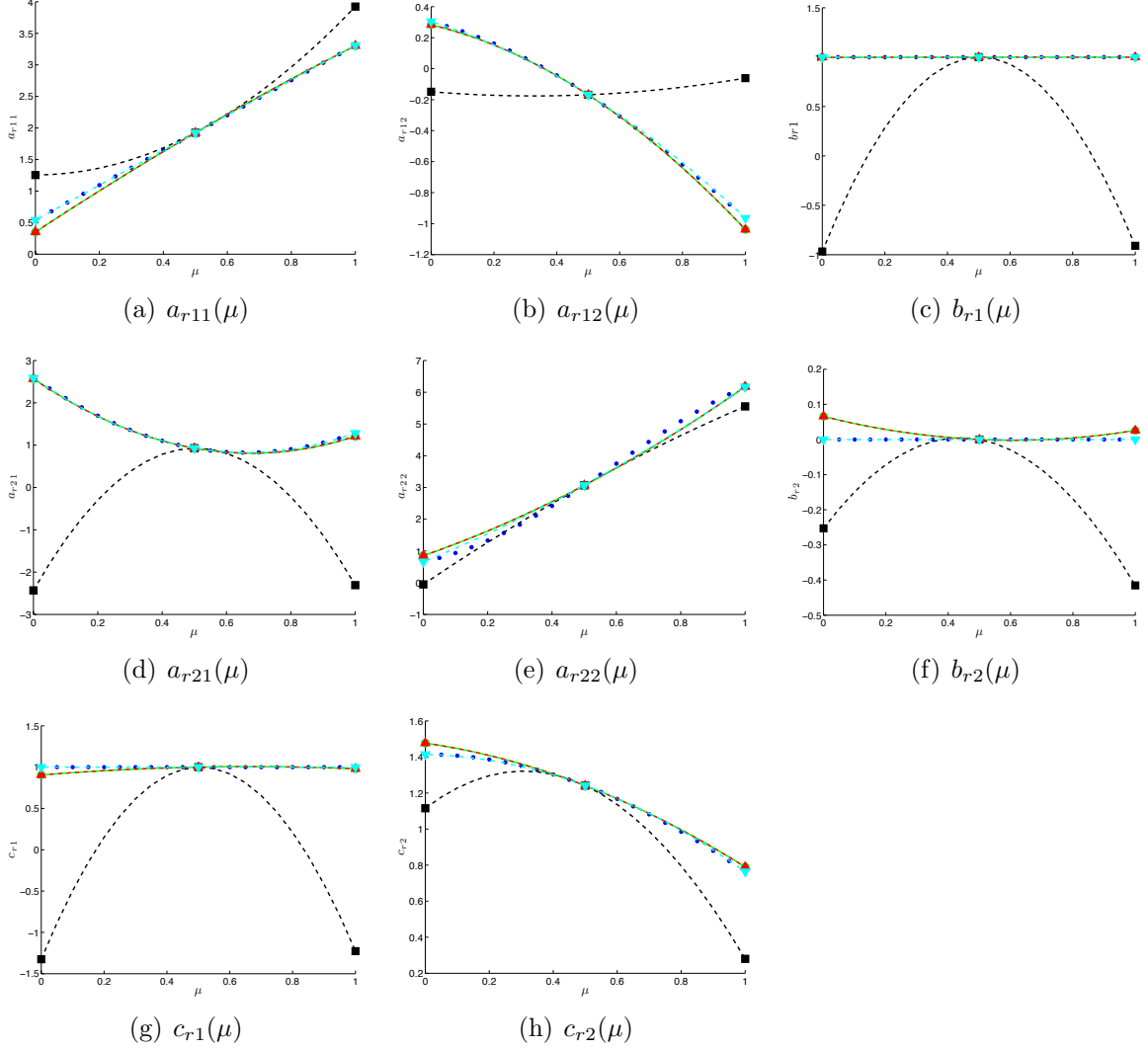
Figure 7.1: Entries of the parametric ROM for the proposed example: (1) in the reference bases (blue star), (2) Interpolating the pre-computed ROMs (black squares) leads to the black dashed lines, (3) Interpolating the pre-computed ROMs after applying Step A of the proposed algorithm with Approach 1 (green triangles) leads to the dashed green lines. Similarly, after Step A using Approach 2, ROMs having the red dots as entries are obtained, leading to the dashed red lines after interpolation. Solving Problem P2 with the proposed method leads to transformed ROMs having the light blue triangles as entries, resulting in the light blue dashed lines after interpolation.
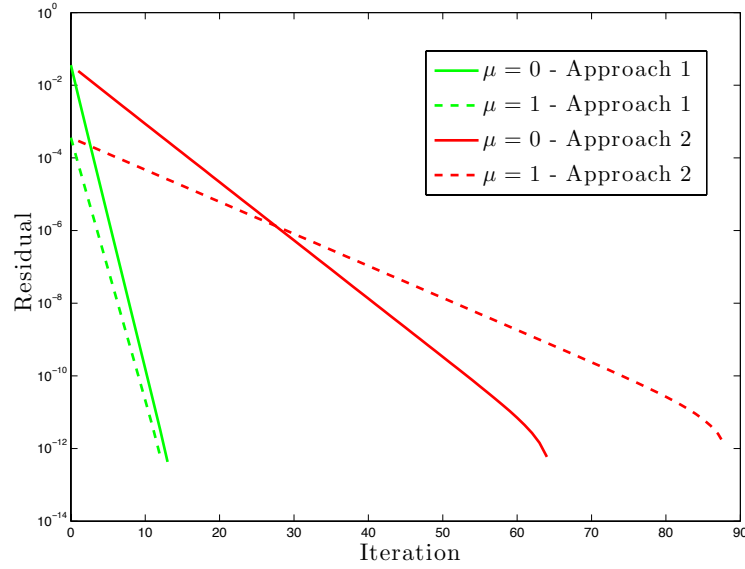
Figure 7.2: Comparison of the residual history for Step A using Approaches 1 and 2.

of this system is defined by a single parameter μ, resulting in the definition of the values of the masses $\{m_j\}_{j=1}^4$, springs $\{k_j\}_{j=1}^6$ and dampers $\{c_j\}_{j=1}^4$ as reported in Table 7.6.1.

Two different predictions of the behavior of the system are first considered for the value $\mu = 0.25$ of the parameter: (a) the full-order dynamic model of dimension $n = 8$, (b) a ROM counterpart of dimension $k = 4$ directly built using a Two-Sided Krylov moment matching technique [36]. Very good agreements for the frequency responses of the two systems appear from the comparison of their respective Bode plots in Figure 7.4 as well as the frequency response of the corresponding error system in Figure 7.5.

Next, the accuracy of the proposed interpolation procedures is assessed by computing the frequency response at $\mu = 0.25$ using preduced-order information precomputed at the values $\mu_1 = 0$, $\mu_2 = 0.5$ and $\mu_3 = 1$. More specifically, (c) a ROM computed

using a Global POD (see section 2.8.3) approach is computed by building a left and a right Reduced-Order Basis of size $k = 4$ and then projecting the directly computed full-order model built at $\mu = 0.25$ onto those bases. This method cannot processed in real-time for two reasons: first, the full model needs to be recomputed at $\mu = 0.25$. Also, the computation of the ROB requires the singular value decomposition (SVD) of a matrix which size scales with the full model. (d) the method proposed in [110] is then considered. While the method does not require the computation of the full model, it however requires an SVD of a matrix scaling with the full model. The two methods proposed in this paper are subsequently considered: (e) the methods solving Problem P1 and (f) the principal angles-based method solving Problem P2. Both methods operate on matrices scaling only with the size of the ROM, and as such, can be processed in real-time.

Figure 7.4 shows that the methods compare favorably with their direct ROM counterpart. The ROM built using the global technique, predicts however a position of the peak of the frequency response that differs from the one predicted by the other models. This is confirmed by inspection of the Bode plot of the errors systems with respect to the full system in Fig. 7.5. A comparison of the eigenvalues of the dynamical systems in Fig. 7.6 shows a very good agreement between the eigenvalues of the direct ROM, the interpolated ROMs built with methods (d) (e) and (f), and the largest eigenvalues of the full model. However, there is a discrepancy with the eigenvalues computed using the global basis method. The two algorithms considered for solving Problem P1 provide identical results. Their respective convergence histories are reported in Fig. 7.7.

The ROM techniques are then compared to their full-order counterparts over the full parameter range $\mu \in [0, 1]$. For that purpose, the relative $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of
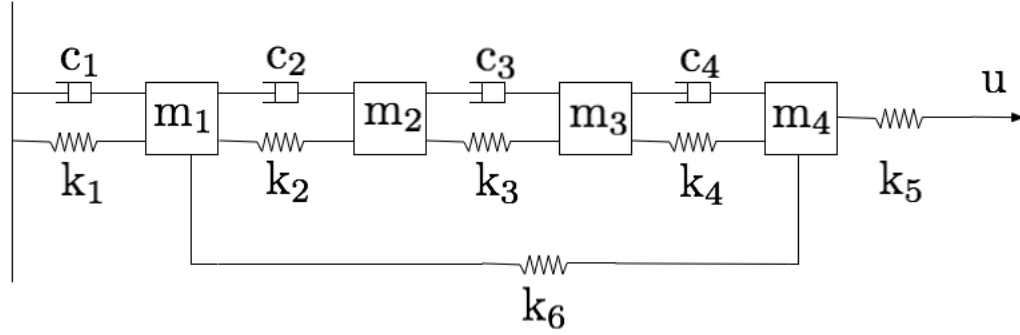
Figure 7.3: Schematic representation of the mass-damper-spring system example.

the error systems between each ROM and the full model are computed and reported in Figures 7.8 and 7.9. Again, the largest error is obtained with the global method. The three other interpolation procedures lead to errors that compare favorably with the one obtained with the directly computed ROM. The reader should note that the two methods proposed in this paper enable real-time computations.

Masses		Dampers		Springs	
(in kg)		(in Ns/m)		(in N/m)	
m_1	125	c_1	μ	k_1	$2 + 2\mu$
m_2	25	c_2	1.6	k_2	1
m_3	5	c_3	0.4	k_3	3
m_4	1	c_4	0.1	k_4	9
				k_5	27
				k_6	$1 + 2\mu$

Table 7.2: Parameterized mechanical system properties.
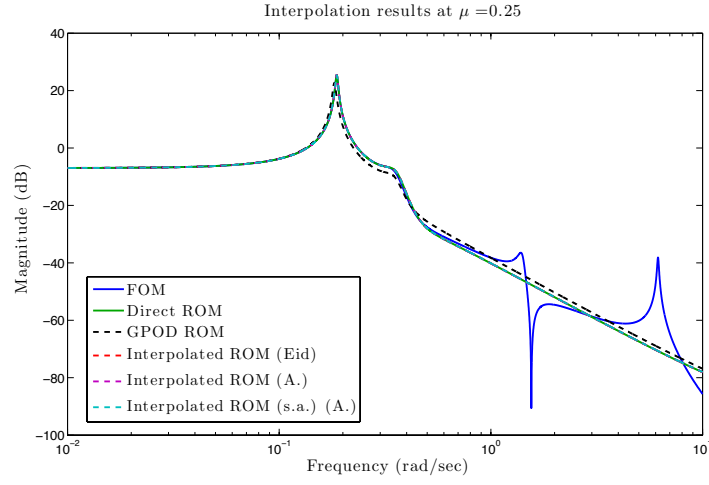
Figure 7.4: Bode plot of the FOM and ROMs for $\mu = 0.25$.
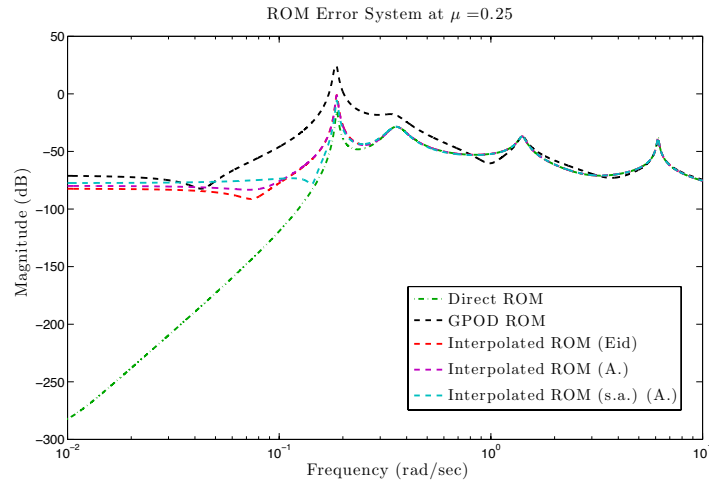
Figure 7.5: Bode plot of the error systems between the FOM and ROMs for $\mu = 0.25$.
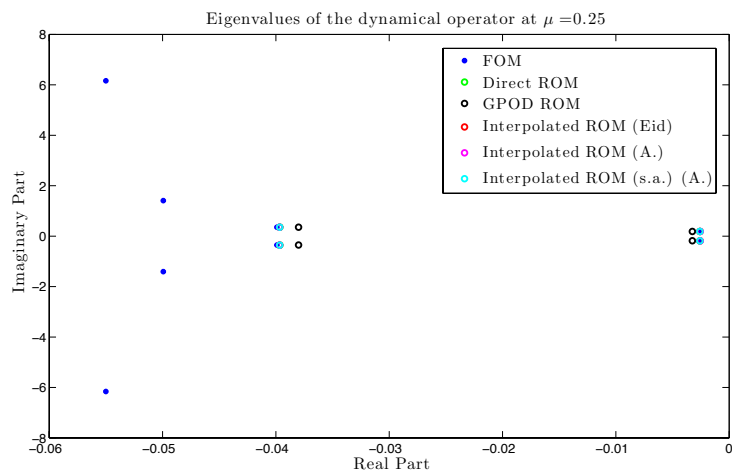
Figure 7.6: Comparison of the eigenvalues of the dynamical operators for $\mu = 0.25$.
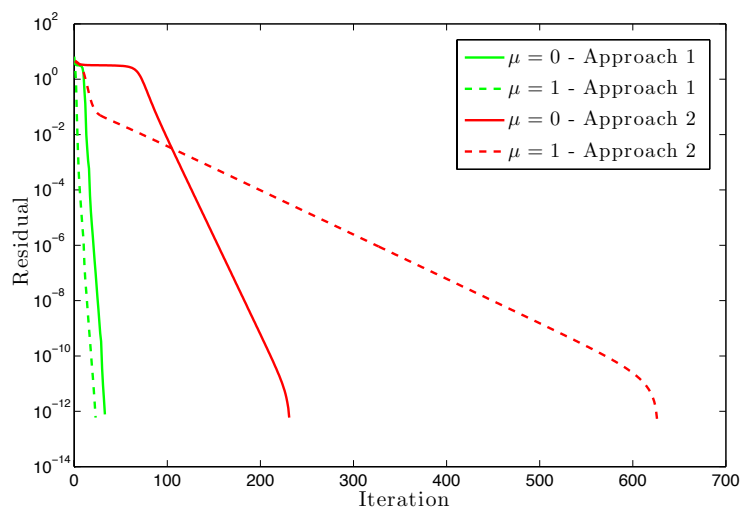
Figure 7.7: Comparison of the residual history for Step A using Approaches 1 and 2.
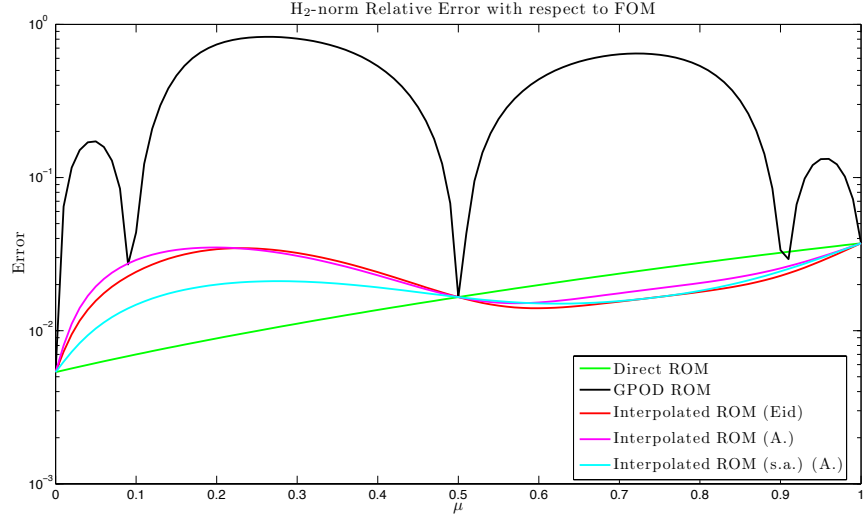
Figure 7.8: $\mathcal{H}_2$-norm relative error between the various reduced-order models and the full-order model.
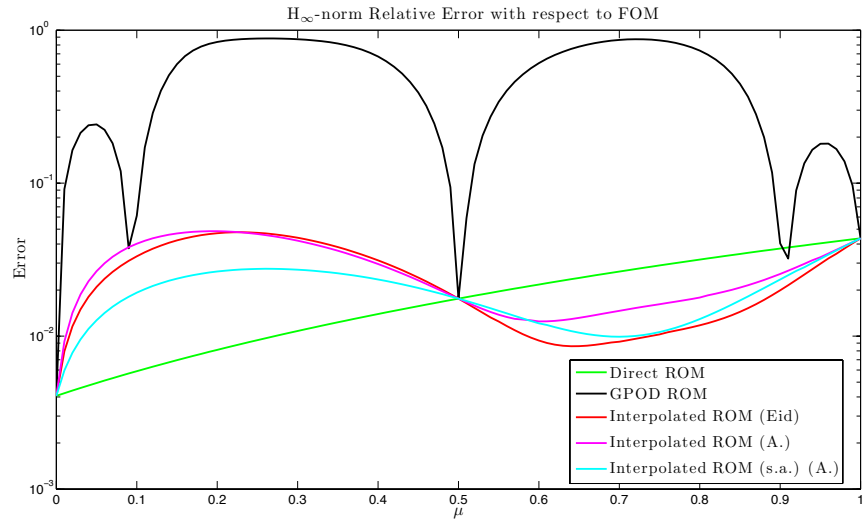
Figure 7.9: $\mathcal{H}_\infty$-norm relative error between the various reduced-order models and the full-order model.

Fluid System

The AGARD Wing 445.6 weakened model 3 is considered. This wing was first stud-ied experimentally by Yates in Ref. [105]. The present model contains 800 shell elements defined by 441 nodes, which corresponds to 2646 degrees of freedom before the boundary conditions are applied.

The structural ROM is built using Modal Truncation of the FEM structural model. Four principal modes are extracted ($N_m = 4$). The eigenfrequencies of these modes are compared to the eigenfrequencies of the first four modes of the experimental model from [105] in Table 7.3. A good agreement between these frequencies can be observed.

Mode number	Frequency (Hz) (Yates experimental model)	Frequency (Hz) (present FEM model)	Description
1	9.60	9.76	First bending
2	38.17	39.52	First torsion
3	48.35	50.39	Second bending
4	91.54	96.99	Second torsion

Table 7.3: AGARD Wing 445.6 retained modes for the structure

The fluid model is based on an inviscid mesh containing 22,000 nodes, that is 110,000 degrees of freedom. A three dimensional Euler code based on a finite volume approximation is used to solve the fluid equations. This code is based on a second-order Galerkin fluxes reconstruction using Roe's method with a Van Albada limiter. The Euler equations are linearized around a steady-state for the fluid sub-system, as developed in Chapter 4.

For a given Mach number-angle of attack (M_∞, α) configuration, complex-valued snapshots are built in the frequency domain by exciting the fluid sub-system at a given frequency and with a given structural mode. Six frequencies (including zero) are considered, resulting in $N_{snap} = 44$ real-valued snapshots. A Reduced-Order

Basis of size $N_\Phi = 40$ is then built using the Proper Orthogonal Decomposition method and a fluid ROM is obtained by Galerkin projection of the full-order model. The procedure for building the ROM has been preciously validated by Lieu in his thesis [84]: comparison with nonlinear and linear full-order models were shown to produce very good agreements with the ROM.

In a numerical experiment, a database of four pre-computed fluid ROMs is considered. These ROMs correspond to the flight conditions $(M_{\infty 1},\ \alpha_1) = (0.901, 0°)$, $(M_{\infty 2},\ \alpha_2) = (0.925, 0°)$, $(M_{\infty 3},\ \alpha_3) = (0.950, 0°)$ and $(M_{\infty 4},\ \alpha_4) = (0.975, 0°)$. A target operating point is then considered at $(M_{\infty 5},\ \alpha_5) = (0.930, 0°)$. The method developed in this chapter to solve Problem P2 is applied to interpolate the reduced fluid operators $\hat{\mathbf{H}}_r$, $\mathbf{R}_r$, $\hat{\mathbf{G}}_r$ and $\hat{\mathbf{P}}'_m$ defined in section 4.4.5:

- $\hat{\mathbf{H}}_{rl}$ belongs to the manifold $\mathbb{R}^{N_\Phi \times N_\Phi}$.

- $\mathbf{R}_{rl}$ and $\hat{\mathbf{G}}_{rl}$ belong to the manifold $\mathbb{R}^{N_\Phi \times N_m}$.

- $\hat{\mathbf{P}}'_{ml}$ belongs to the manifold $\mathbb{R}^{N_m \times N_\Phi}$.

The time history of the coupled aeroelastic system at the target configuration, at a pressure $p_\infty = 14$ lbf/in^2 is considered: an initial displacement of the bending mode is imposed and the time histories of the general displacements of the structure are compared for: (1) a ROM directly computed at the target configuration $(M_{\infty 5},\ \alpha_5) = (0.930, 0°)$, (2) a ROM interpolated by the method developed to solve Problem P2. The time-histories are reported in Fig. 7.10 and very good agreements between the two can be observed.
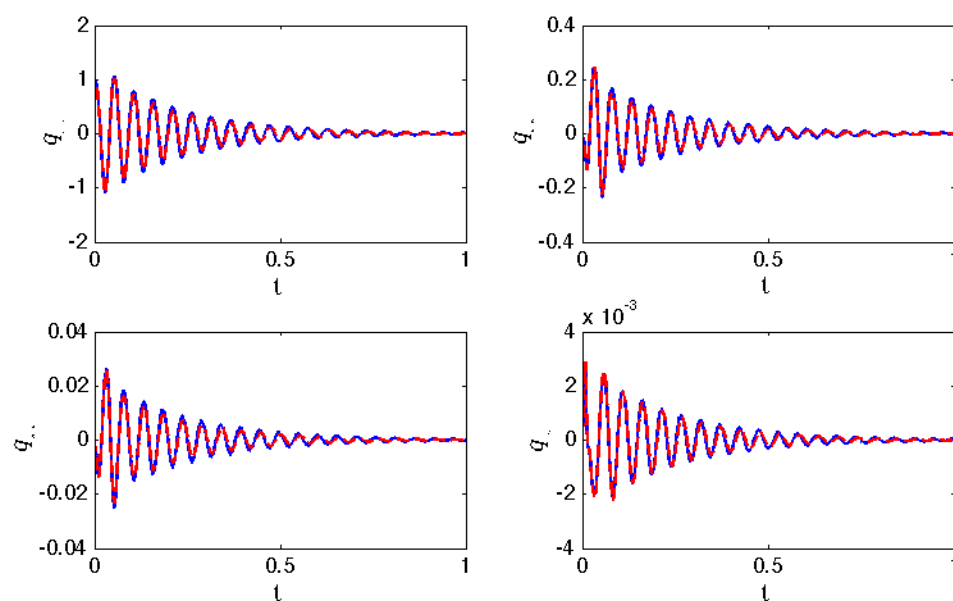
Figure 7.10: Comparison of the time-history of the general structural displacements for the coupled fluid/structure system obtained building a fluid ROM (1) directly (in blue), (2) by interpolation (in red).

7.6.2 Application to Second-Order Systems

System with Mode Veering

The framework proposed in this chapter is illustrated with a simple system presenting a curve veering phenomenon. This phenomenon is presented in [117]. A simple mechanical system presenting a mode veering phenomenon is considered. This system, first studied in [118], is constituted of two masses coupled by three springs as shown in Fig. 7.11. The system is parameterized by the spring stiffness k_1 and all the other parameters are fixed: $m_1 = m_2 = k_2 = 1$ and $\tilde{k} = 0.05$. The matrices of the system are

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} k_1 + \tilde{k} & -\tilde{k} \\ -\tilde{k} & k_2 + \tilde{k} \end{bmatrix}. \tag{7.78}$$

A reference configuration is chosen as $k_{1,1} = 0.1$ and, hence the varying parameter is defined as

$$\mu = k_1 - k_{1,1}. \tag{7.79}$$

Here, the domain of variation for μ will be chosen as $\mathcal{D} = [0, k_{1,2} - k_{1,1}]$, where $k_{1,2} = 3$. The analytic expression for the eigenvalues of this system is

$$\lambda_{1,2} = \frac{k_1 m_2 + k_2 m_1 + \tilde{k}(m_1 + m_2) \mp \sqrt{((k_1 + \tilde{k})m_2 - (k_2 + \tilde{k})m_1)^2 + 4m_1 m_2 \tilde{k}^2}}{2m_1 m_2}. \tag{7.80}$$

With the values of the parameters that have been chosen, one obtains

$$\lambda_{1,2}(\mu) = \frac{\mu + 1.2 \mp \sqrt{((\mu - 0.9)^2 + 0.01}}{2}. \tag{7.81}$$

Figure 7.12 presents the behavior of this eigenvalues as μ varies in the interval $\mathcal{D} = [0, 2.9]$. The mode veering phenomenon occurs at the critical value $k_1 = 1$, that is $\mu = 0.9$: the corresponding directions of the eigenmodes are reported in Figure 7.13. The first eigendirection is depicted as a blue line, and the second direction as a red line. It can be observed that the modes undergo a rotation of angle $\frac{\pi}{2}$ between the values $\mu = 0$ and $\mu = 2.9$, at $\mu = 0.9$ the modes have undergone a rotation of $\frac{\pi}{4}$.
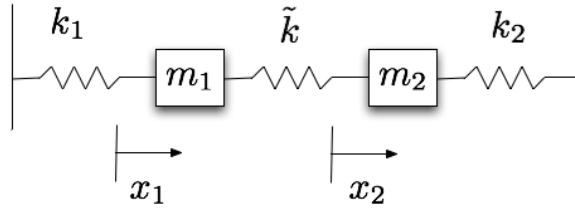
Figure 7.11: Mechanical System of Interest
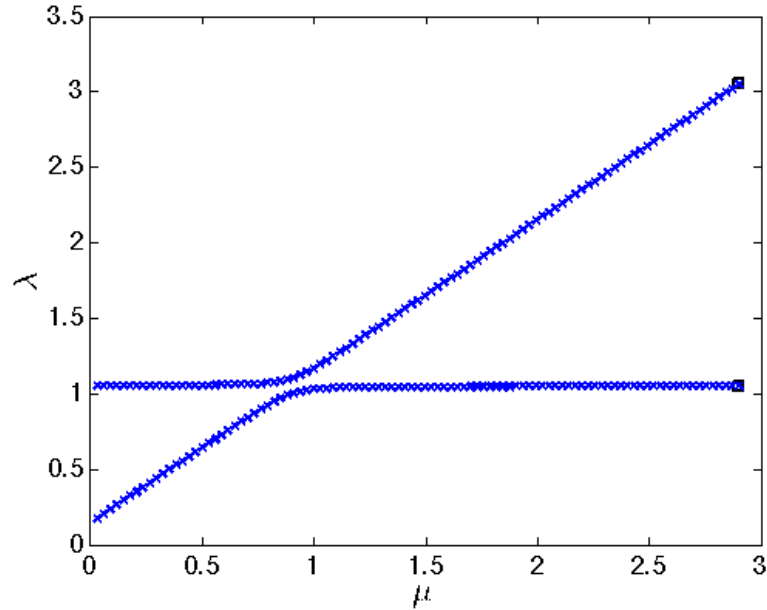
Figure 7.12: Eigenvalues loci for the system of interest
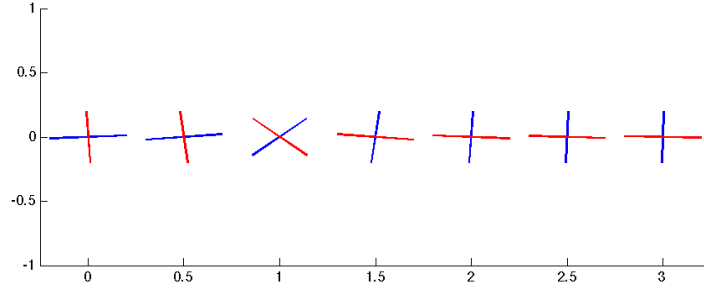
Figure 7.13: Eigenmodes for the system of interest

The goal of this section is to compare the behavior of the two methods proposed in this paper on the mode veering problem. Two precomputed configurations are considered:

$$\mu_1 = 0, \quad \mu_2 = 2.9. \tag{7.82}$$

The pre-computed quantities of interest for each problem are reported as follows.

- Problem P1:

$$\mathbf{M}_r(\mu_1) = \mathbf{I}_2, \quad \mathbf{K}_r(\mu_1) = \mathbf{\Lambda}(\mu_1) = \begin{bmatrix} 0.1472 & 0 \\ 0 & 1.0528 \end{bmatrix}, \tag{7.83}$$

$$\mathbf{M}_r(\mu_2) = \mathbf{I}_2, \quad \mathbf{K}_2(\mu_2) = \mathbf{\Lambda}(\mu_2) = \begin{bmatrix} 1.0488 & 0 \\ 0 & 3.0512 \end{bmatrix}. \tag{7.84}$$

- Problem P2: In addition to the quantities precomputed for Problem P1, the inner product between the local modes is also provided.

$$\mathbf{P}_{12} = \mathbf{X}(\mu_1)^T \mathbf{X}(\mu_2) = \begin{bmatrix} 0.0802 & 0.9968 \\ -0.9968 & 0.0802 \end{bmatrix}. \tag{7.85}$$

Problem P1 is expected to be more challenging to solve than Problem P2, since no information about the behavior of the eigenvectors is available. On the contrary, the rotation of the eigenvectors due to the mode veering can be detected from inspection of $\mathbf{P}_{12}$ and the proposed method developed for Problem P2 is expected to give more accurate interpolation results. The interpolation results are reported in Figures 7.14. For the critical value of the parameter $\mu_{cr} = 0.9$, the reduced-order matrices that are returned by each method are reported as follows:

- Problem P1:

$$\hat{\mathbf{M}}_r(\mu_{cr}) = \mathbf{I}_2, \quad \hat{\mathbf{K}}_r(\mu_{cr}) = \begin{bmatrix} 0.427 & 0 \\ 0 & 1.6730 \end{bmatrix}, \qquad (7.86)$$

In the present case, Step A.2 has returned $\mathbf{Q}_2 = \mathbf{I}_2$ in accordance with Theorem 3. Hence no transformation of the pre-computed ROMs is done: the rotation of the eigenvectors due to the mode veering cannot be predicted by this method.

- Problem P2:

$$\hat{\mathbf{M}}_r(\mu_{cr}) = \mathbf{I}_2, \quad \hat{\mathbf{K}}_r(\mu_{cr}) = \begin{bmatrix} 1.0438 & 0.0243 \\ 0.0243 & 1.0735 \end{bmatrix}. \qquad (7.87)$$

The algorithm has proceeded by first expressing all the pre-computed ROMs in the same coordinate basis $\mathbf{X}(\mu_1)$ and then interpolating these ROMs. In the present case, Step A.2 has returned $\mathbf{Q}_2 = \begin{bmatrix} 0.0802 & -0.9968 \\ 0.9968 & 0.0802 \end{bmatrix}$: the first eigenmode of the second configuration has been matched with the second eigenmode of the first configuration and the second eigenmode of the second configuration

with the first eigenmode of the first configuration. This results in ROM expressed in consistent bases (the basis $\mathbf{X}(\mu_1)$) and one can check that $\hat{\mathbf{K}}_r(\mu_{cr})$ is a good approximation of $\mathbf{X}(\mu_1)^T\mathbf{K}(\mu_{cr})\mathbf{X}(\mu_1) = \begin{bmatrix} 1.0445 & 0.0497 \\ 0.0497 & 1.0555 \end{bmatrix}$. As such the mode veering was accurately captured by the method.

As expected by the analysis in section 7.5.5, the algorithm solving Problem P1 interpolates the highest and lowest eigenvalues having no information about the eigenmodes behavior: it assumes that the highest eigenvalues correspond to the same mode in both configurations. This is, however, not the case as shown by the behavior of the eigenmodes in a mode veering configuration. The algorithm for solving Problem P2 takes into account the eigenvectors rotation and, as such, is able to predict the mode veering phenomenon quite accurately. Eigenmode crossing is a similar phenomenon to mode veering and it is also expected that the second algorithm would better perform in such a case.
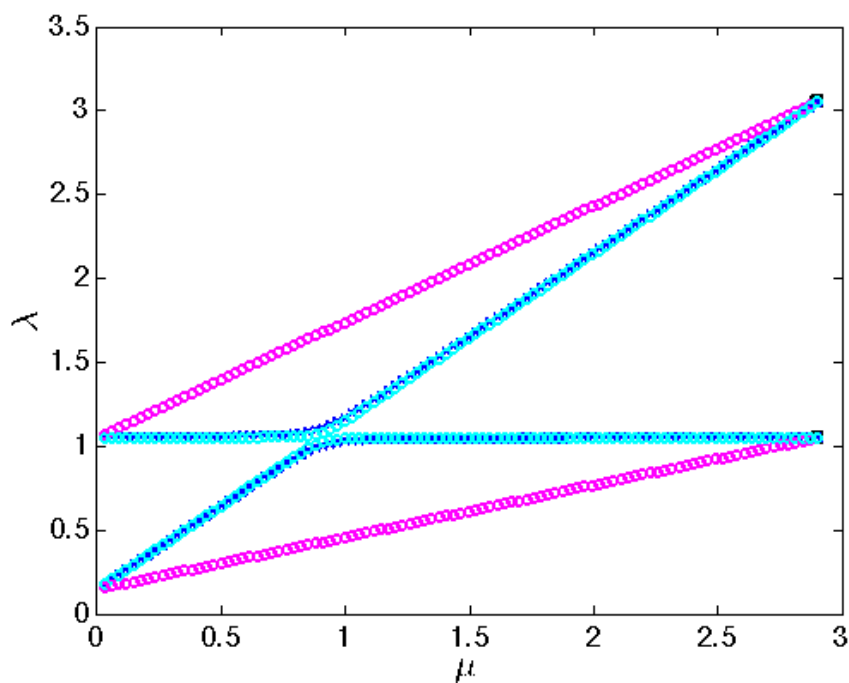
Figure 7.14: Eigenvalues loci for the interpolated systems. Black squares: pre-computed points, blue stars: exact eigenvalues, magenta dots: eigenvalues of the system interpolated using the algorithm for solving Problem P1, light blue dots: eigenvalues of the system interpolated using the algorithm for solving Problem P2.

Chapter 8

Conclusions

This chapter summarizes the work conducted in the present dissertation and provides perspectives for future investigation.

8.1 Summary

In this thesis, novel interpolation-based procedures for adapting reduced-order models to parameter changes were developed. These methods were tested on fluid and structural systems and academic and industrial problems. Such procedures were enabled by the development of a general, multivariate and high-order interpolation algorithm on matrix manifolds. This algorithm was applied to the adaptation of fluid and structural reduced-order bases and models for aeroelastic systems. Finally, a machine learning-based procedure was shown to be effective for exploiting a database of reduced-order bases and models, leading to real-time aeroelastic predictions.

A careful analysis of Petrov-Galerkin-based model reduction techniques first highlighted that subspaces and not reduced bases are the meaningful quantities characterizing a reduced-order model. As such, the Grassmann manifold was considered as

the manifold of interest for reduced-order bases interpolation across the parameter space. The Grassmann manifold being a nonlinear matrix manifold, classic interpolation techniques fail to produce interpolated quantities belonging to that manifold. Hence, a general interpolation algorithm based on concepts from the field of differential geometry was designed to enforce the constraint of belonging to a matrix manifold. High-order, multivariate compact schemes were shown to produce accurate reduced-order models for basic, academic test cases.

The technique of interpolation on the Grassmann manifold was then applied to the aeroelastic prediction of the behavior of two full aircraft configurations (F-18/A and F-16 Block 40) at subsonic, transonic and supersonic speeds. Reduced-order models obtained using the novel interpolation technique were shown to reproduce well their counterparts obtained directly at the very same flight conditions as well as their high-fidelity, full-order counterparts. This method was then extended to routine analysis applications, where databases of reduced-order bases and models can be obtained. A machine learning-based procedure allowing the local selection of an optimal interpolation scheme in the parameter space was developed. Not only were the resulting reduced-order models accurate for the F-16 model, but impressive computational speed gains were achieved, paving the way for the embedding of this method into on-line, real-time flutter analysis procedures.

It was finally shown that the use of high-fidelity, parallel codes can be bypassed in the interpolation procedure by focusing solely on the interpolation of the reduced-order linear operators. Applications to the adaptation of structural and fluid reduced-order models with respect to geometric parameter changes were provided, showing the ability of the method to generate reduced-order models in real-time for new, "last minute" configurations.

8.2   Perspectives for Future Work

In this thesis, CFD-based aeroelastic predictions are computed in near-real time: for instance, a transient aeroelastic response of a full F-16 Block 40 aircraft configuration can be obtained in 8 minutes on a 64-CPU computational cluster. Such a response is computed with a ROM based on an interpolated reduced-order basis. The generation of the reduced-order basis can be done in real-time using the interpolation algorithm on the Grassmann manifold: it is generated in one minute. The seven remaining minutes correspond to the generation of a steady-state solution at the new configuration. Such as steady-state is computed using the high-fidelity CFD code and as such, its computation cannot be processed in real-time, nor on a single processor. As such, a procedure that would generate an aeroelastic response in real-time has to by-pass this steady-state generation. This is the case of a method based on the interpolation of the underlying reduced-order operators. In this thesis, such an interpolation has been successfully presented for an AGARD wing system at one configuration in the transonic flight regime ($M_\infty = 0.930$). The cost associated with interpolating such a ROM to new parameter values is decreased to a few seconds. Further investigation is required to validate the proposed method on a larger set of operating points, as well as on full-aircraft configurations.

The proposed interpolation method has shown great promises for the adaptation of linear reduced-order bases and models with respect to parameter variations. The development of similar method for the adaptation of nonlinear reduced-order models would allow the alleviation of important computational burdens for the analysis of intrinsically nonlinear physical phenomena, such as limit-cycle oscillation in aeroelasticity for instance.

The application of the interpolation procedure to routine analysis has shown the

ability of the method to produce new reduced-order models in real time. In the fields of optimization and design, there is also a great interest in the capability of generating cheaply a large number of accurate models, as many-queries techniques are of the essence. As such, embedding adaptation techniques inside such frameworks could lead to faster analyses for large-scale physical models.

Appendix A

Sensitivity of the Singular Value Decomposition

In this section, a general matrix $\mathbf{A} \in \mathbb{R}^{n \times k}$ ($n \geq k$), and its thin singular value decomposition are considered.

$$\mathbf{A} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T, \tag{A.1}$$

where $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_k] \in \mathcal{ST}(k, n)$, $\boldsymbol{\Sigma} = \mathbf{diag}(\sigma_1, \cdots, \sigma_k) \in \mathbb{R}^{k \times k}$, and $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_k] \in \mathrm{O}(k)$. The k first left singular vectors are $\{\mathbf{u}_i \in \mathbb{R}^n\}_{i=1}^k$ and the right singular vectors are $\{\mathbf{v}_i \in \mathbb{R}^k\}_{i=1}^k$. The singular values $\{\sigma_i\}_{i=1}^k$ are here assumed to be strictly positive (otherwise the compact SVD can be considered instead).

$\mathbf{A}$ is assumed to smoothly depend on a unique parameter μ. The multivariate case where μ is a vector of parameters can be derived easily from the present analysis.

Property: The sensitivities of the singular values σ_i and their associated left and

right singular vectors are

$$
\frac{d\sigma_i}{d\mu} = \mathbf{u}_i^T \frac{d\mathbf{A}}{d\mu} \mathbf{v}_i, \ \ i = 1, \cdots, k, \tag{A.2}
$$

$$
\frac{d\mathbf{U}}{d\mu} = \left(\frac{d\mathbf{A}}{d\mu} - \mathbf{U}\frac{d\mathbf{\Sigma}}{d\mu}\mathbf{V}^T - \mathbf{U}\mathbf{\Sigma}\frac{d\mathbf{V}^T}{d\mu} \right) \mathbf{V}\mathbf{\Sigma}^{-1} \tag{A.3}
$$

$$
\frac{d\mathbf{v}_i}{d\mu} = \sum_{j=1}^{k} c_{ij}\mathbf{v}_j, \ \ i = 1, \cdots, k, \tag{A.4}
$$

with $\dfrac{d\mathbf{U}}{d\mu} = \begin{bmatrix} \dfrac{d\mathbf{u}_1}{d\mu} & \cdots & \dfrac{d\mathbf{u}_k}{d\mu} \end{bmatrix}$ and

$$
\begin{cases}
c_{ij} = \dfrac{1}{\sigma_i^2 - \sigma_j^2}\mathbf{v}_j^T \left(\dfrac{d\mathbf{A}^T}{d\mu}\mathbf{A} + \mathbf{A}^T\dfrac{d\mathbf{A}}{d\mu} \right)\mathbf{v}_i, & \text{if } i \neq j \\[4mm]
c_{jj} = 0.
\end{cases}
\tag{A.5}
$$

The derivation of these formulae is presented next.

A.1 Singular Values

To calculate the sensitivity of singular values $\left\{ \dfrac{d\sigma_i}{d\mu} \right\}_{i=1}^{k}$, the following relation is considered:

$$
\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i. \tag{A.6}
$$

Taking the derivative and premultiplying by $\mathbf{u}_i^T$ leads to (as $\mathbf{u}_i^T\mathbf{u}_i = 1$)

$$
\mathbf{u}_i^T\frac{d\mathbf{A}}{d\mu}\mathbf{v}_i + \mathbf{u}_i^T\mathbf{A}\frac{d\mathbf{v}_i}{d\mu} = \frac{d\sigma_i}{d\mu} + \sigma_i\mathbf{u}_i^T\frac{d\mathbf{u}_i}{d\mu}, \tag{A.7}
$$

The orthogonality of $\mathbf{u}_i$ implies that

$$\mathbf{u}_i^T \frac{d\mathbf{u}_i}{d\mu} = 0. \tag{A.8}$$

and Eq. (A.7) becomes

$$\mathbf{u}_i^T \frac{d\mathbf{A}}{d\mu} \mathbf{v}_i + \mathbf{u}_i^T \mathbf{A} \frac{d\mathbf{v}_i}{d\mu} = \frac{d\sigma_i}{d\mu}. \tag{A.9}$$

Moreover, using the orthogonality of $\mathbf{v}_i$,

$$\mathbf{u}_i^T \mathbf{A} \frac{d\mathbf{v}_i}{d\mu} = \mathbf{u}_i^T \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \frac{d\mathbf{v}_i}{d\mu} = \mathbf{\Sigma}_i \mathbf{V}^T \frac{d\mathbf{v}_i}{d\mu} = \sigma_i \mathbf{v}_i^T \frac{d\mathbf{v}_i}{d\mu} = 0 \tag{A.10}$$

where $\mathbf{\Sigma}_i$ denotes the i-th column vector of $\mathbf{\Sigma}$.

This leads to the desired singular value sensitivity

$$\frac{d\sigma_i}{d\mu} = \mathbf{u}_i^T \frac{d\mathbf{A}}{d\mu} \mathbf{v}_i. \tag{A.11}$$

This result shows that, in order to calculate the sensitivity of the singular values, all is needed is the sensitivity of the matrix $\mathbf{A}$ and the corresponding left and right singular vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ respectively.

A.2 Right Singular Vectors

To calculate the sensitivity of the right singular vectors $\left\{ \frac{d\mathbf{v}_i}{d\mu} \right\}_{i=1}^{k}$, the fact that the vectors $\{\mathbf{v}_i\}_{i=1}^{k}$ are eigenvectors of the matrix $\mathbf{A}^T\mathbf{A}$ is used. This is quite convenient, since the problem is now turned into one of calculating the sensitivity of the eigenvectors of a symmetric matrix which form a complete basis. Using the results presented in Reference [87], one can calculate the sensitivity of these eigenvectors according to

the following equations:

$$\frac{d\mathbf{v}_i}{d\mu} = \sum_{j=1}^{k} c_{ij} \mathbf{v}_j, \tag{A.12}$$

with

$$\begin{cases} c_{ij} &= \dfrac{1}{\sigma_i^2 - \sigma_j^2} \mathbf{v}_j^T \left(\dfrac{d\mathbf{A}^T}{d\mu} \mathbf{A} + \mathbf{A}^T \dfrac{d\mathbf{A}}{d\mu} \right) \mathbf{v}_i, & \text{if } i \neq j \\ c_{jj} &= 0. \end{cases} \tag{A.13}$$

This method does not strictly work for repeated singular values, as this results in unbounded coefficients.

A.3 Left Singular Vectors

The derivative of the singular value decomposition Eq. (A.1) is first taken.

$$\frac{d\mathbf{A}}{d\mu} = \frac{d\mathbf{U}}{d\mu} \mathbf{\Sigma} \mathbf{V}^T + \mathbf{U} \frac{d\mathbf{\Sigma}}{d\mu} \mathbf{V}^T + \mathbf{U} \mathbf{\Sigma} \frac{d\mathbf{V}^T}{d\mu}. \tag{A.14}$$

$\dfrac{d\mathbf{A}}{d\mu}$ is already known. Moreover, from the previous derivations, $\dfrac{d\mathbf{\Sigma}}{d\mu}$ and $\dfrac{d\mathbf{V}}{d\mu}$ are known as well. Since $\mathbf{\Sigma}$ is invertible and $\mathbf{V}$ is orthogonal, one can solve for $\dfrac{d\mathbf{U}}{d\mu}$ with

$$\frac{d\mathbf{U}}{d\mu} = \left(\frac{d\mathbf{A}}{d\mu} - \mathbf{U} \frac{d\mathbf{\Sigma}}{d\mu} \mathbf{V}^T - \mathbf{U} \mathbf{\Sigma} \frac{d\mathbf{V}^T}{d\mu} \right) \mathbf{V} \mathbf{\Sigma}^{-1}. \tag{A.15}$$

Appendix B

Computation of $\nabla_{\mathbf{x}}\tilde{F}$

Goal. Compute $\nabla_{\mathbf{x}}\tilde{F}$ where

$$\tilde{F}(\mathbf{A}, \mathbf{B}) = \|\mathrm{Exp}_{\mathbf{Y}_0}(\mathbf{Y}_0\mathbf{A} + \mathbf{Z}\mathbf{B}) - \mathbf{Y}_1\|_F^2, \tag{B.1}$$

and $\mathrm{Exp}_{\mathbf{Y}_0}(\mathbf{Y}_0\mathbf{A} + \mathbf{Z}\mathbf{B})$ is defined as

$$\mathbf{Q}\mathbf{R} = (\mathbf{I}_n - \mathbf{Y}_0\mathbf{Y}_0^T)\boldsymbol{\Gamma} \quad \text{(compact QR decomposition)} \tag{B.2}$$

$$\begin{bmatrix} \mathbf{M} \\ \mathbf{N} \end{bmatrix} = \mathtt{expm} \begin{bmatrix} \mathbf{Y}_0^T\boldsymbol{\Gamma} & -\mathbf{R}^T \\ \mathbf{R} & \mathbf{0}_{k,k} \end{bmatrix} \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0}_{k,k} \end{bmatrix} \tag{B.3}$$

$$\mathrm{Exp}_{\mathbf{Y}_0}(\boldsymbol{\Gamma}) = \mathbf{Y}\mathbf{M} + \mathbf{Q}\mathbf{N}, \tag{B.4}$$

with $\boldsymbol{\Gamma} = \mathbf{Y}_0\mathbf{A} + \mathbf{Z}\mathbf{B}$.

Computation. The vector $\mathbf{x} \in \mathbb{R}^{n_{DOF}}$ contains the independent entries in $\mathbf{A}$ and $\mathbf{B}$, namely, $\{\mathbf{A}_{i,j},\ i \in \{1, \cdots, k\},\ j \in \{i+1, \cdots, k\}\}$ and $\{\mathbf{B}_{i,j},\ i \in \{1, \cdots, n-k\},\ j \in \{1, \cdots, k\}\}$.

It can be shown that:

$$
\begin{aligned}
\tilde{F}(\mathbf{A}, \mathbf{B}) &= \operatorname{tr}\left[\mathbf{G}(\mathbf{M}(\mathbf{A},\mathbf{B}), \mathbf{Q}(\mathbf{B}), \mathbf{N}(\mathbf{A},\mathbf{B}))\right] \\
&= \operatorname{tr}\left[(\mathbf{Y}_0\mathbf{M} + \mathbf{Q}\mathbf{N} - \mathbf{Y}_1)^T(\mathbf{Y}_0\mathbf{M} + \mathbf{Q}\mathbf{N} - \mathbf{Y}_1)\right] \\
&= \operatorname{tr}(\mathbf{M}^T\mathbf{M}) + \operatorname{tr}(\mathbf{N}^T\mathbf{N}) - 2\operatorname{tr}(\mathbf{Y}_1^T\mathbf{Y}_0\mathbf{M}) - 2\operatorname{tr}(\mathbf{Y}_1^T\mathbf{Q}\mathbf{N}) + k,
\end{aligned}
\tag{B.5}
$$

from which the following sensitivities can be obtained

$$
\frac{\partial \mathbf{G}}{\partial \mathbf{M}} = 2(\mathbf{M} - \mathbf{Y}_0^T\mathbf{Y}_1) \tag{B.6}
$$

$$
\frac{\partial \mathbf{G}}{\partial \mathbf{Q}} = 2(\mathbf{Y}_0\mathbf{M} - \mathbf{Y}_1)\mathbf{N}^T \tag{B.7}
$$

$$
\frac{\partial \mathbf{G}}{\partial \mathbf{N}} = 2(\mathbf{N} - \mathbf{Q}^T\mathbf{Y}_1). \tag{B.8}
$$

From $\mathbf{Q}\mathbf{R} = \mathbf{Z}\mathbf{B} = \mathbf{K}$ and $\mathbf{B} = \mathbf{Z}^T\mathbf{K}$, the following sensitivities can be obtained by differentiating every step in the modified Gram-Schmidt algorithm:

$$
\frac{\partial \mathbf{Q}_{i,j}}{\partial \mathbf{B}},\ \frac{\partial \mathbf{R}_{l,s}}{\partial \mathbf{B}} \in \mathbb{R}^{r \times k},\quad i \in \{1, \cdots, n\},\ j, l \in \{1, \cdots, r\},\ s \in \{1, \cdots, k\}, \tag{B.9}
$$

as

$$
\frac{\partial \mathbf{Q}_{i,j}}{\partial \mathbf{B}} = \mathbf{Z}^T \frac{\partial \mathbf{Q}_{i,j}}{\partial \mathbf{K}}\quad i \in \{1, \cdots, n\},\ j \in \{1, \cdots, r\}, \tag{B.10}
$$

and

$$
\frac{\partial \mathbf{R}_{l,s}}{\partial \mathbf{B}} = \mathbf{Z}^T \frac{\partial \mathbf{R}_{l,s}}{\partial \mathbf{K}}\quad l \in \{1, \cdots, r\},\ s \in \{1, \cdots, k\}, \tag{B.11}
$$

where r is the number of column vectors in the $\mathbf{Q}$ matrix, that is the rank of the matrix $\mathbf{K} = \mathbf{Z}\mathbf{B}$.

Due to the fact that the sensitivities of a matrix exponential to entries in the matrix are extremely difficult to obtain analytically, the following sensitivities are obtained

by finite differences:

$$\frac{\partial \mathbf{M}_{i,j}}{\partial \mathbf{A}}, \ \frac{\partial \mathbf{N}_{l,s}}{\partial \mathbf{A}} \ \in \mathbb{R}^{\frac{k(k-1)}{2}}, \ \frac{\partial \mathbf{M}_{i,j}}{\partial \mathbf{R}}, \ \frac{\partial \mathbf{N}_{l,s}}{\partial \mathbf{R}} \ \in \mathbb{R}^{\frac{k(k+1)}{2}}, \ \ i,j,s \in \{1,\cdots,k\}, \ l \in \{1,\cdots,r\}. \tag{B.12}$$

Finally, the sensitivities of $\tilde{F}$ with respect to the independent entries in $\mathbf{A}$ and $\mathbf{B}$ are computed as

$$
\begin{aligned}
\frac{\partial \tilde{F}}{\partial \mathbf{A}_{i,j}} &= \text{tr}\left(\left(\frac{\partial \mathbf{G}}{\partial \mathbf{M}}\right)^T \frac{\partial \mathbf{M}}{\partial \mathbf{A}_{i,j}}\right) + \text{tr}\left(\left(\frac{\partial \mathbf{G}}{\partial \mathbf{N}}\right)^T \frac{\partial \mathbf{N}}{\partial \mathbf{A}_{i,j}}\right) \\
&= 2\left(\text{tr}\left((\mathbf{M}-\mathbf{Y}_0^T\mathbf{Y}_1)^T \frac{\partial \mathbf{M}}{\partial \mathbf{A}_{i,j}}\right) + \text{tr}\left((\mathbf{N}-\mathbf{Q}^T\mathbf{Y}_1)^T \frac{\partial \mathbf{N}}{\partial \mathbf{A}_{i,j}}\right)\right),
\end{aligned}
\tag{B.13}
$$

for $i \in \{1,\cdots,k\}$, $j \in \{i+1,\cdots,k\}$, and

$$
\begin{aligned}
\frac{\partial \tilde{F}}{\partial \mathbf{B}_{i,j}} &= 2\sum_{l=1}^{k}\sum_{s=l}^{k}\left(\text{tr}\left((\mathbf{M}-\mathbf{Y}_0^T\mathbf{Y}_1)^T \frac{\partial \mathbf{M}}{\partial \mathbf{R}_{l,s}}\right) + \text{tr}\left((\mathbf{N}-\mathbf{Q}^T\mathbf{Y}_1)^T \frac{\partial \mathbf{N}}{\partial \mathbf{R}_{l,s}}\right)\right)\left(\mathbf{Z}^T\frac{\partial \mathbf{R}_{l,s}}{\partial \mathbf{K}}\right)_{i,j} \\
&\quad + 2\sum_{a=1}^{k}\sum_{b=1}^{r}((\mathbf{Y}_0\mathbf{M}-\mathbf{Y}_1)\mathbf{N}^T)_{a,b}\left(\mathbf{Z}^T\frac{\partial \mathbf{Q}_{a,b}}{\partial \mathbf{K}}\right)_{i,j}.
\end{aligned}
\tag{B.14}
$$

Appendix C

Proofs for Preservation Properties of Interpolation on SPD(n)

Here, two important propositions characterizing the proposed generalized interpolation method in a tangent space to a matrix manifold are proven. For this purpose, it is assumed that the generalized interpolation method is equipped with a standard multivariate interpolation algorithm that approximates constant functions exactly (for example, see [76]).

Proposition 2. Let $S = \left(\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \cdots, \boldsymbol{\mu}^{(N_R)} \right)$ denote a set of N_R operating points and let $\{ \mathbf{Q}_i = \mathbf{Q} \}_{i=1}^{N_R}$ denote a corresponding set of N_R identical symmetric positive definite matrices of dimension n. Given a new operating point $\boldsymbol{\mu}^{(N_R+1)} \notin S$, the interpolation method described in section 3.6.2 delivers a matrix $\mathbf{Q}_{N_R+1}$ which is identical to $\mathbf{Q}$ as well.

Proof. In this case, the step-by-step application of the generalized interpolation method

presented in section 3.6.2 to the set of N_R SPD matrices $\mathbf{Q}(\boldsymbol{\mu}^{(i)}) = \mathbf{Q}$ gives the following results.

- Step 0. $\mathbf{Q}(\boldsymbol{\mu}^{(i_0)}) = \mathbf{Q}$ is chosen as a reference element of $\text{SPD}(n)$.

- Step 1. Each element $\mathbf{Q}(\boldsymbol{\mu}^{(i)}) = \mathbf{Q}$ is obviously close to itself and therefore is mapped to the matrix $\boldsymbol{\Gamma}_i$ of $T_{\mathbf{Q}}\text{SPD}(n)$ as follows

$$\boldsymbol{\Gamma}_i = \text{Log}_{\mathbf{Q}}(\mathbf{Q}) = \texttt{logm}\left(\mathbf{Q}^{-1/2}\mathbf{Q}\mathbf{Q}^{-1/2}\right) = \texttt{logm}\left(\mathbf{I}_n\right) = \mathbf{0}_{n,n} \qquad \text{(C.1)}$$

where $\mathbf{0}_{n,n}$ denotes the zero square matrix of size n.

- Step 2. Each entry of the matrix $\boldsymbol{\Gamma}_{N_R+1}$ associated with the target operating point $\boldsymbol{\mu}^{(N_R+1)}$ is computed by interpolating the corresponding entries of the matrices $\{\boldsymbol{\Gamma}_i = \mathbf{0}_{n,n}\}_{i=1}^{N_R}$ associated with the operating points $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^{N_R}$. Since a standard interpolation algorithm approximates exactly a constant function,

$$\boldsymbol{\Gamma}_{N_R+1} = \mathbf{0}_{n,n}. \qquad \text{(C.2)}$$

- Step 3. The matrix $\boldsymbol{\Gamma}_{N_R+1} \in T_{\mathbf{Q}}\text{SPD}(n)$ is mapped to a matrix $\mathbf{Q}_{N_R+1} \in \text{SPD}(n)$ as follows

$$\begin{aligned}\mathbf{Q}_{N_R+1} &= \text{Exp}_{\mathbf{Q}}(\boldsymbol{\Gamma}_{N_R+1}) = \mathbf{Q}^{1/2}\texttt{expm}(\boldsymbol{\Gamma}_{N_R+1})\mathbf{Q}^{1/2} = \mathbf{Q}^{1/2}\texttt{expm}(\mathbf{0}_{n,n})\mathbf{Q}^{1/2}\\ &= \mathbf{Q}^{1/2}\mathbf{I}_n\mathbf{Q}^{1/2} = \mathbf{Q}, \qquad \text{(C.3)}\end{aligned}$$

which proves the above proposition.

$\square$

Proposition 3. Let $S = \left(\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \cdots, \boldsymbol{\mu}^{(N_R)}\right)$ denote a set of N_R operating points and let $\left\{\boldsymbol{\Omega}_{n,i}^2\right\}_{i=1}^{N_R}$ denote a corresponding set of N_R diagonal matrices with n positive entries

$$
\boldsymbol{\Omega}_{n,i}^2 =
\begin{bmatrix}
\omega_{1,i}^2 = \left(\omega_1\left(\boldsymbol{\mu}^{(i)}\right)\right)^2 & & & & (0) \\
& \omega_{2,i}^2 = \left(\omega_2\left(\boldsymbol{\mu}^{(i)}\right)\right)^2 & & & \\
& & \ddots & & \\
(0) & & & & \omega_{n,i}^2 = \left(\omega_n\left(\boldsymbol{\mu}^{(i)}\right)\right)^2
\end{bmatrix}.
\quad \text{(C.4)}
$$

Given a new operating point $\boldsymbol{\mu}^{(N_R+1)} \notin S$, the interpolation method described in section 3.6.2 delivers a matrix $\boldsymbol{\Omega}_{n,N_R+1}^2$ which is a diagonal matrix with n positive entries. This means that the set of positive diagonal matrices is stable by interpolation.

Proof. Similarly to the proof of Proposition 1, the step-by-step application of the generalized interpolation method presented in section 3.6.2 to the set of N_R SPD matrices $\boldsymbol{\Omega}_{n,i}^2$ (C.4) gives the following results.

- Step 0. $\boldsymbol{\Omega}_{n,i_0}^2$ is chosen as a reference element of SPD(n).

- Step 1. Each matrix $\boldsymbol{\Omega}_{n,i}^2$ that is sufficiently close to $\boldsymbol{\Omega}_{n,i_0}^2$ is mapped to a matrix

$\mathbf{\Gamma}_i$ of $T_{\mathbf{\Omega}^2_{n,i_0}}$ SPD(n) as follows

$$
\begin{aligned}
\mathbf{\Gamma}_i &= \mathrm{Log}_{\mathbf{\Omega}^2_{n,i_0}}(\mathbf{\Omega}^2_{n,i}) \\
&= \mathrm{logm}\left(\mathbf{\Omega}^2_{n,i_0}{}^{-1/2}\mathbf{\Omega}^2_{n,i}\mathbf{\Omega}^2_{n,i_0}{}^{-1/2}\right) \\
&= \mathrm{logm}\left(\begin{bmatrix} \omega^{-1}_{1,i_0} & & & (0) \\ & \omega^{-1}_{2,i_0} & & \\ & & \ddots & \\ (0) & & & \omega^{-1}_{n,i_0} \end{bmatrix}\begin{bmatrix} \omega^2_{1,i} & & & (0) \\ & \omega^2_{2,i} & & \\ & & \ddots & \\ (0) & & & \omega^2_{n,i} \end{bmatrix}\right. \\
&\qquad\qquad \left.\begin{bmatrix} \omega^{-1}_{1,i_0} & & & (0) \\ & \omega^{-1}_{2,i_0} & & \\ & & \ddots & \\ (0) & & & \omega^{-1}_{n,i_0} \end{bmatrix}\right) \\
&= \mathrm{logm}\left(\begin{bmatrix} \left(\dfrac{\omega_{1,i}}{\omega_{1,i_0}}\right)^2 & & & (0) \\ & \left(\dfrac{\omega_{2,i}}{\omega_{2,i_0}}\right)^2 & & \\ & & \ddots & \\ (0) & & & \left(\dfrac{\omega_{n,i}}{\omega_{n,i_0}}\right)^2 \end{bmatrix}\right) \\
&= \begin{bmatrix} 2\log\left(\dfrac{\omega_{1,i}}{\omega_{1,i_0}}\right) & & & (0) \\ & 2\log\left(\dfrac{\omega_{2,i}}{\omega_{2,i_0}}\right) & & \\ & & \ddots & \\ (0) & & & 2\log\left(\dfrac{\omega_{n,i}}{\omega_{n,i_0}}\right) \end{bmatrix}.
\end{aligned}
\tag{C.5}
$$

- Step 2. Each entry of the matrix $\mathbf{\Gamma}_{N_R+1}$ associated with the target operating point $\boldsymbol{\mu}^{(N_R+1)}$ is computed by interpolating the corresponding entries of the

matrices $\{\mathbf{\Gamma}_i\} \in \mathbb{R}^{n \times n}$ associated with the operating points $\{\boldsymbol{\mu}^{(i)}\}$. Since each matrix $\mathbf{\Gamma}_i$ (C.6) is in this case diagonal, $\mathbf{\Gamma}_{N_R+1}$ is also diagonal and can be written as

$$
\mathbf{\Gamma}_{N_R+1} = \begin{bmatrix} \alpha_1 & & & (0) \\ & \alpha_2 & & \\ & & \ddots & \\ (0) & & & \alpha_n \end{bmatrix}.
\tag{C.6}
$$

- Step 3. The matrix $\mathbf{\Gamma}_{N_R+1} \in T_{\boldsymbol{\Omega}^2_{n,i_0}} \mathrm{SPD}(n)$ is mapped to a matrix $\mathbf{K}_{rN_R+1}$[1] on

---

[1]This notation is used in chapter 5 to denote a reduced stiffness matrix.

SPD(n) as follows

$$
\begin{aligned}
\mathbf{K}_{rN_R+1} &= \mathrm{Exp}_{\mathbf{\Omega}^2_{n,i_0}}(\mathbf{\Gamma}_{N_R+1}) \\
&= \mathbf{\Omega}^2_{n,i_0}{}^{1/2}\mathrm{expm}(\mathbf{\Gamma}_{N_R+1})\mathbf{\Omega}^2_{n,i_0}{}^{1/2}
\end{aligned}
$$

$$
= \begin{bmatrix} \omega_{1,i_0} & & & (0) \\ & \omega_{2,i_0} & & \\ & & \ddots & \\ (0) & & & \omega_{n,i_0} \end{bmatrix} \begin{bmatrix} \exp(\alpha_1) & & & (0) \\ & \exp(\alpha_2) & & \\ & & \ddots & \\ (0) & & & \exp(\alpha_n) \end{bmatrix}
$$

$$
\begin{bmatrix} \omega_{1,i_0} & & & (0) \\ & \omega_{2,i_0} & & \\ & & \ddots & \\ (0) & & & \omega_{n,i_0} \end{bmatrix}
$$

$$
= \begin{bmatrix} \omega^2_{1,i_0}\exp(\alpha_1) & & & (0) \\ & \omega^2_{2,i_0}\exp(\alpha_2) & & \\ & & \ddots & \\ (0) & & & \omega^2_{n,i_0}\exp(\alpha_n) \end{bmatrix}
$$

$$
= \begin{bmatrix} \omega^2_{1,N_R+1} & & & (0) \\ & \omega^2_{2,N_R+1} & & \\ & & \ddots & \\ (0) & & & \omega^2_{n,N_R+1} \end{bmatrix} = \mathbf{\Omega}^2_{n,N_R+1}, \tag{C.7}
$$

which proves the proposition. $\square$

Appendix D

Logarithm Mapping for the Grassmann Manifold Represented by Non-Compact Bases

Proposition. Let $\mathcal{Y}, \mathcal{Z} \in \mathcal{G}(k,n)$ be subspaces respectively represented by matrices $\mathbf{Y}$ and $\mathbf{Z}$ belonging to the non-compact Stiefel manifold $\mathbb{R}_*^{n \times k}$. Then $\mathrm{Log}_{\mathcal{Y}}(\mathcal{Z})$ is represented by $\mathbf{\Gamma}$ satisying

$$(\mathbf{I}_n - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T)\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T \quad \text{(thin SVD)} \qquad \text{(D.1)}$$

$$\mathbf{\Gamma} = \tilde{\mathbf{U}}\mathbf{tan}^{-1}(\tilde{\mathbf{\Sigma}})\tilde{\mathbf{V}}^T. \qquad \text{(D.2)}$$

Proof. Showing that $\mathrm{Exp}_{\mathcal{Y}}(\mathcal{V}) = \mathcal{Z} = \mathrm{span}\{\mathbf{Z}\}$, where $\mathcal{V} \in T_{\mathcal{Y}}\mathcal{G}(k,n)$ is represented

by $\mathbf{\Gamma}$ will prove the proposition.

$$\text{Exp}_{\mathcal{Y}}(\mathcal{V}) = \text{span}\left\{\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\tilde{\mathbf{V}}\mathbf{cos}(\mathbf{tan}^{-1}\tilde{\mathbf{\Sigma}}) + \tilde{\mathbf{U}}\mathbf{sin}(\mathbf{tan}^{-1}\tilde{\mathbf{\Sigma}})\right\}$$

$$= \text{span}\left\{\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\tilde{\mathbf{V}}\mathbf{cos}(\mathbf{tan}^{-1}\tilde{\mathbf{\Sigma}})\right. \tag{D.3}$$

$$\left. + \mathbf{\Pi}_{\mathbf{Y},\mathbf{Y}}\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}}\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{sin}(\mathbf{tan}^{-1}\tilde{\mathbf{\Sigma}})\right\}.$$

Since

$$\mathbf{cos}(\mathbf{tan}^{-1}\tilde{\mathbf{\Sigma}}) = \tilde{\mathbf{\Sigma}}^{-1}\mathbf{sin}(\mathbf{tan}^{-1}\tilde{\mathbf{\Sigma}}) = \mathbf{diag}\left(\frac{1}{\sqrt{1+\tilde{\sigma}_i^2}}\right) = \left(\mathbf{I}_n + \tilde{\mathbf{\Sigma}}^2\right)^{-\frac{1}{2}} \tag{D.4}$$

is non-singular,

$$\text{Exp}_{\mathcal{Y}}(\mathcal{V}) = \text{span}\left\{\left(\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} + \mathbf{\Pi}_{\mathbf{Y},\mathbf{Y}}\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}}\right)\tilde{\mathbf{V}}\left(\mathbf{I}_n + \tilde{\mathbf{\Sigma}}^2\right)^{-\frac{1}{2}}\right\}$$

$$= \text{span}\left\{\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} + \mathbf{\Pi}_{\mathbf{Y},\mathbf{Y}}\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}}\right\}$$

$$= \text{span}\left\{\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} + \mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1} - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}}\right\}$$

$$= \text{span}\left\{\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} + \mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1} - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\right\}$$

$$= \text{span}\left\{\mathbf{Z}(\mathbf{Y}^T\mathbf{Z})^{-1}\right\}$$

$$= \text{span}\left\{\mathbf{Z}\right\}$$

$$= \mathcal{Z}.$$

$$\tag{D.5}$$

$\square$

Appendix E

Rank-Preserving Orthogonal Basis Perturbations

An important proposition characterizing the sensitivity of full-rank orthogonal bases to perturbations is here proven. It is shown that any orthogonal basis can be perturbed by a non-zero perturbation without modification of its rank. This property will be of great importance for the computation of the gradient of the logarithm mapping at the origin, since this shows that this computation is amenable by a finite difference approximation.

Proposition: Let $\boldsymbol{\mu}^{(0)} \in \mathbb{R}^{N_p}$ and $\mathbf{Y} = \mathbf{Y}(\boldsymbol{\mu}^{(0)}) \in \mathcal{ST}(k,n)$ be an orthogonal basis continuously differentiable with respect of the variables $\{\mu_j\}_{j=1}^{N_p}$. Defining

$$\mathbf{E} = \nabla\mathbf{Y}(\boldsymbol{\mu}^{(0)}) = \left[\frac{\partial\mathbf{Y}}{\partial\mu_1}(\boldsymbol{\mu}^{(0)}) \quad \dots \quad \frac{\partial\mathbf{Y}}{\partial\mu_{N_p}}(\boldsymbol{\mu}^{(0)})\right] \in \mathbb{R}^{n\times(kN_p)1}, \qquad \text{(E.1)}$$

---

[1]$\mathbf{E} \in \mathbb{R}^{n\times(kN_p)}$ means that $\mathbf{E}$ is the gradient of an n-by-k matrix with respect to N_p independent variables. $\mathbf{E}$ is here represented by an n-by-$k \times N_p$ matrix.

there exists a strictly positive real number $R > 0$ such that $\forall \delta\boldsymbol{\mu} \in \mathcal{C}_{N_p}(0, R)$, $\text{rank}(\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu}) = \text{rank}(\mathbf{Y}) = k$. $\mathcal{C}_{N_p}(0, R)$ denotes here the circle of radius R centered in zero in $\mathbb{R}^{N_p}$.

Proof. Let $\delta\boldsymbol{\mu} = (\delta\mu_1, \cdots, \delta\mu_{N_p})$. $\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu}$ can be expanded in terms of the columns of $\mathbf{E}$ as

$$\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu} = \mathbf{Y} + \sum_{j=1}^{N_p} \delta\mu_j \mathbf{E}_j = \mathbf{Y} + \sum_{j=1}^{N_p} \delta\mu_j \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}). \tag{E.2}$$

As $\text{rank}[\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu}] = \text{rank}[(\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu})^T(\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu})]$ and this matrix is square and of small size $k \leq n$, this suggests considering

$$(\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu})^T(\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu}) = \mathbf{Y}^T\mathbf{Y} + \sum_{j=1}^{N_p} \delta\mu_j(\mathbf{Y}^T\mathbf{E}_j + \mathbf{E}_j^T\mathbf{Y}) + \sum_{i=1}^{N_p}\sum_{j=1}^{N_p} \delta\mu_j\delta\mu_i \mathbf{E}_j^T\mathbf{E}_i.$$

$$\tag{E.3}$$

The matrices $\mathbf{Y}$, $\mathbf{E}$ and thus $\{\mathbf{E}_j\}_{j=1}^{N_p} = \left\{\dfrac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)})\right\}_{j=1}^{N_p}$ present the following properties by construction

$$\mathbf{Y}^T\mathbf{Y} = \mathbf{I}_k \tag{E.4}$$

$$\mathbf{Y}^T\mathbf{E}_j + \mathbf{E}_j^T\mathbf{Y} = \mathbf{0}_{k,k}, \quad j = 1, \cdots, N_p. \tag{E.5}$$

The expression in Eq. (E.3) can thus be simplified as

$$(\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu})^T(\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu}) = \mathbf{I}_k + \sum_{i=1}^{N_p}\sum_{j=1}^{N_p} \delta\mu_j\delta\mu_i \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)})^T \frac{\partial \mathbf{Y}}{\partial \mu_i}(\boldsymbol{\mu}^{(0)}). \tag{E.6}$$

The identity matrix $\mathbf{I}_k$ is here perturbed by a second-order term. This suggests that if $\delta\boldsymbol{\mu}$ is chosen small enough, the identity matrix will not be too perturbed and its rank will be unchanged. To this effect, the Gershgorin Circle Theorem [35] is used. This theorem presents how well the diagonal elements of a matrix approximate its eigenvalues. For completeness purposes, it is reported here as follows.

Theorem: Let $\mathbf{A}$ be a square matrix of size k that can be written as $\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \mathbf{D} + \mathbf{F}$ where $\mathbf{D} = \mathbf{diag}(d_1, \cdots, d_k)$ and $\mathbf{F}$ is a matrix with zero diagonal entries. Then the spectrum of $\mathbf{A}$ denoted by $\lambda(\mathbf{A})$ satisfies

$$\lambda(\mathbf{A}) \subseteq \bigcup_{r=1}^{k} \mathcal{D}_r, \tag{E.7}$$

where $\mathcal{D}_r = \{z \in \mathbb{C} \;:\; |z - d_r| \leq \sum_{t=1}^{k} |\mathbf{F}_{rt}|, \; \forall t = 1, \cdots, k\}$.

Using the same notations as in the theorem, in the present case

$$\mathbf{A} = (\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu})^T (\mathbf{Y} + \mathbf{E} \cdot \delta\boldsymbol{\mu})$$

$$\mathbf{X} = \mathbf{I}_k$$

$$\mathbf{D} = \mathbf{I}_k + \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} \delta\mu_l \delta\mu_m \mathbf{diag} \left(\frac{\partial \mathbf{Y}}{\partial \mu_l}(\boldsymbol{\mu}^{(0)})^T \frac{\partial \mathbf{Y}}{\partial \mu_m}(\boldsymbol{\mu}^{(0)}) \right)$$

$$\mathbf{F} = \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} \delta\mu_l \delta\mu_m \left(\frac{\partial \mathbf{Y}}{\partial \mu_l}(\boldsymbol{\mu}^{(0)})^T \frac{\partial \mathbf{Y}}{\partial \mu_m}(\boldsymbol{\mu}^{(0)}) - \mathbf{diag} \left(\frac{\partial \mathbf{Y}}{\partial \mu_l}(\boldsymbol{\mu}^{(0)})^T \frac{\partial \mathbf{Y}}{\partial \mu_m}(\boldsymbol{\mu}^{(0)}) \right) \right),$$

$$\tag{E.8}$$

$\mathbf{diag}(\mathbf{M})$ denoting the matrix containing the diagonal elements of a square matrix $\mathbf{M}$.

Let $\mathbf{B}^{(l,m)} \in \mathbb{R}^{k \times k}$ be defined as

$$\mathbf{B}^{(l,m)} = \left(\frac{\partial \mathbf{Y}}{\partial \mu_l} (\boldsymbol{\mu}^{(0)})^T \frac{\partial \mathbf{Y}}{\partial \mu_m} (\boldsymbol{\mu}^{(0)}) \right), \quad l = 1, \cdots, N_p, \quad m = 1, \cdots, N_p. \tag{E.9}$$

The (r, t)-th entry of this square matrix of size k is denoted by $\mathbf{B}_{r,t}^{(l,m)}$ in the remainder of this section. In order to conserve the rank of $\mathbf{Y}(\boldsymbol{\mu}^{(0)})$, the perturbed matrix $\mathbf{A}$ in Eq. (E.8) must have full rank as well. A necessary condition for this, given by the Gershgorin Theorem, is that zero is not a possible eigenvalue for $\mathbf{A}$.

Mathematically, a sufficient condition is

$$|d_r| > \sum_{t=1}^{k} |\mathbf{F}_{rt}|, \quad \forall r = 1, \cdots, k, \tag{E.10}$$

that is

$$\forall r = 1, \cdots, k, \quad \left| 1 + \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} \delta\mu_l \delta\mu_m \mathbf{B}_{r,r}^{(l,m)} \right| - \sum_{\substack{t \neq r \\ 1 \leq t \leq k}} \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} |\delta\mu_l \delta\mu_m \mathbf{B}_{r,t}^{(l,m)}| > 0. \tag{E.11}$$

For a given $r = 1, \cdots, k$, the following series of inequalities hold

$$\left| 1 + \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} \delta\mu_l \delta\mu_m \mathbf{B}_{r,r}^{(l,m)} \right| - \sum_{\substack{t \neq r \\ 1 \leq t \leq k}} \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} \left| \delta\mu_l \delta\mu_m \mathbf{B}_{r,t}^{(l,m)} \right|$$

$$> 1 - \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} |\delta\mu_l| |\delta\mu_m| \left| \mathbf{B}_{r,r}^{(l,m)} \right| - \sum_{\substack{t \neq r \\ 1 \leq t \leq k}} \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} |\delta\mu_l| |\delta\mu_m| \left| \mathbf{B}_{r,t}^{(l,m)} \right|$$

$$> 1 - \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} |\delta\mu_l| |\delta\mu_m| \mathbf{C}_{l,m}^{(r)}$$

$$> 1 - \left| \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} |\delta\mu_l| |\delta\mu_m| \mathbf{C}_{l,m}^{(r)} \right|, \tag{E.12}$$

where the $\mathbf{C}^{(r)}$s are defined as the square matrices of size N_p having positive (l,m)-th entries equal to

$$\mathbf{C}_{l,m}^{(r)} = \sum_{t=1}^{k} \left| \mathbf{B}_{r,t}^{(l,m)} \right|. \tag{E.13}$$

Using the Cauchy-Schwartz inequality

$$\forall \mathbf{w} = (w_1, \cdots, w_k) \in \mathbb{R}^{N_p}, \quad \left| \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} w_l w_m \mathbf{C}_{l,m}^{(r)} \right| = |\mathbf{w}^T \mathbf{C}^{(r)} \mathbf{w}| \leq \|\mathbf{w}\|^2 \left\| \mathbf{C}^{(r)} \right\|_2, \tag{E.14}$$

the following series of k inequalities hold

$$\left| 1 + \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} \delta\mu_l \delta\mu_m \mathbf{B}_{r,r}^{(l,m)} \right| - \sum_{\substack{t \neq r \\ 1 \leq t \leq k}} \sum_{l=1}^{N_p} \sum_{m=1}^{N_p} |\delta\mu_l \delta\mu_m \mathbf{B}_{r,t}^{(l,m)}|$$

$$> 1 - \|\|\delta\boldsymbol{\mu}\|\|_2^2 \left\| \mathbf{C}^{(r)} \right\|_2$$

$$> 1 - \|\delta\boldsymbol{\mu}\|_2^2 \left\| \mathbf{C}^{(r)} \right\|_2. \tag{E.15}$$

Choosing the most constraining one, the sufficient condition for rank preservation becomes a condition on $\delta\boldsymbol{\mu}$

$$\forall r = 1, \cdots, k, \ \|\delta\boldsymbol{\mu}\|_2 < \frac{1}{\left\| \mathbf{C}^{(r)} \right\|_2^{1/2}} \Leftrightarrow \|\delta\boldsymbol{\mu}\|_2 < \frac{1}{\max_{1 \leq r \leq k} \left\| \mathbf{C}^{(r)} \right\|_2^{1/2}}. \tag{E.16}$$

which proves the proposition. An algorithm that enforces this condition can then be constructed step-by-step as follows

- Step 1. Define the k matrices $\mathbf{C}^{(r)}$, $r = 1, \cdots, k$, such that

$$\forall l, m = 1, \cdots, N_p, \quad \mathbf{C}_{l,m}^{(r)} = \sum_{t=1}^{k} \left| \mathbf{B}_{r,t}^{(l,m)} \right|. \tag{E.17}$$

- Step 2. Find the index $r_0 \in 1, \cdots, k$ such that

$$\left\| \mathbf{C}^{(r_0)} \right\|_2 = \max_{1 \leq r \leq k} \left\| \mathbf{C}^{(r)} \right\|_2 . \tag{E.18}$$

- Step 3. Choose

$$\delta \boldsymbol{\mu} = (\delta \mu_1, \cdots, \delta \mu_{N_p}), \tag{E.19}$$

where

$$\forall l = 1, \cdots, N_p, \quad \delta \mu_l = \frac{\alpha_l}{\left\| \mathbf{C}^{(r_0)} \right\|_2^{1/2}} \tag{E.20}$$

and where the α_l are real numbers satisfying the constraint $\sum_{l=1}^{N_p} \alpha_l^2 < 1$.

In short, $\delta \boldsymbol{\mu} \in \mathcal{C}_{N_p}(0, R)$, the circle centered at the origin and of radius $R = \dfrac{1}{\left\| \mathbf{C}^{(r_0)} \right\|_2^{1/2}}$.

$\square$

Appendix F

Derivability at the Origin of the Logarithm Mapping on the Grassmann Manifold

An algorithm for computing the logarithm map on the Grassmann manifold at the origin point is here presented. This method relies on a finite difference approximation scheme. The continuity and derivability of this mapping at the origin are proven and it is shown that second-order accuracy can be obtained using a centered finite-difference approximation at no additional cost.

F.1 Computation

The same notations used in Appendix E apply here. As in section 3.7.4, the logarithm mapping for the Grassmann manifold is denoted by $\mathbf{\Gamma}$. A reference value $\boldsymbol{\mu}^{(0)}$ for the parameters is considered as well as the associated subspace $\mathcal{Y}_0 = \mathrm{span}\left\{\mathbf{Y}(\boldsymbol{\mu}^{(0)})\right\}$.

Let $j = 1, \cdots, N_p$ and $\delta\boldsymbol{\mu}_j = \delta\mu_j \mathbf{e}_j$, where $\mathbf{e}_j$ denotes the j-th canonical unit vector in $\mathbb{R}^{N_p}$. Assuming that $\delta\boldsymbol{\mu}_j$ is chosen accordingly to the procedure presented in Appendix E,

$$\text{rank}\left[\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j \frac{\partial\mathbf{Y}}{\partial\mu_j}(\boldsymbol{\mu}^{(0)}))\right] = \text{rank}[\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \nabla\mathbf{Y}(\boldsymbol{\mu}^{(0)}).\delta\boldsymbol{\mu}_j] = \text{rank}[\mathbf{Y}(\boldsymbol{\mu}^{(0)})].$$
(F.1)

In general $\left(\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j \frac{\partial\mathbf{Y}}{\partial\mu_j}(\boldsymbol{\mu}^{(0)})\right)$ is not an orthonormal basis but since it has full column rank, a QR factorization can be computed

$$\left(\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j \frac{\partial\mathbf{Y}}{\partial\mu_j}(\boldsymbol{\mu}^{(0)})\right) = \mathbf{Q}_j\mathbf{R}_j.$$
(F.2)

$\mathbf{R}_j$ is an upper-triangular invertible square matrix of size k and $\mathbf{Q}_j \in \mathcal{ST}(k,n)$ is a perturbed orthogonal basis.

The logarithm mapping of the subspace $\mathcal{S}_j$ spanned by $\mathbf{Q}_j$ to the tangent space of

the Grassmann manifold at $\mathcal{Y}_0$ proceeds first by computing the thin SVD

$$
\begin{aligned}
\tilde{\mathbf{U}}_j \tilde{\mathbf{\Sigma}}_j \tilde{\mathbf{V}}_j^T &= (\mathbf{I}_n - \mathbf{Y}(\boldsymbol{\mu}^{(0)})\mathbf{Y}(\boldsymbol{\mu}^{(0)})^T)\mathbf{Q}_j(\mathbf{Y}(\boldsymbol{\mu}^{(0)})^T\mathbf{Q}_j)^{-1} \\
&= \mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp, \mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp} \left(\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \right) \mathbf{R}_j^{-1} \\
&\quad \left(\mathbf{Y}(\boldsymbol{\mu}^{(0)})^T \left(\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \right) \mathbf{R}_j^{-1} \right)^{-1} \\
&= \mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp, \mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp} \left(\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \right) \left(\mathbf{I}_k + \delta\mu_j \mathbf{Y}(\boldsymbol{\mu}^{(0)})^T \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \right)^{-1} \\
&= \mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp, \mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp} \left(\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \right) \left(\mathbf{I}_k - \delta\mu_j \mathbf{Y}(\boldsymbol{\mu}^{(0)})^T \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \right) \\
&\quad + \mathcal{O}(\delta\mu_j^2) \\
&= \mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp, \mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp} \left(\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \delta\mu_j (\mathbf{I}_n - \mathbf{Y}(\boldsymbol{\mu}^{(0)})\mathbf{Y}(\boldsymbol{\mu}^{(0)})^T) \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \right) + \mathcal{O}(\delta\mu_j^2) \\
&= \delta\mu_j \mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp, \mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp} \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) + \mathcal{O}(\delta\mu_j^2)
\end{aligned}
$$

(F.3)

and then by letting $\mathbf{\Gamma}_j = \tilde{\mathbf{U}}_j \tan^{-1}\tilde{\mathbf{\Sigma}}_j \tilde{\mathbf{V}}_j^T$. This shows that the matrix $\mathbf{\Gamma}_j$ is a first order matrix with respect to the perturbation quantity $\delta\mu_j$, and the second-order terms in Eq. (F.3) can be neglected as such.

Thus, defining the N_p matrices $\mathbf{H}^j(\boldsymbol{\mu}^{(0)})$, $j = 1, \cdots, N_p$ as

$$
\mathbf{H}^j(\boldsymbol{\mu}^{(0)}) = \mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp, \mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp} \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) = (\mathbf{I}_n - \mathbf{Y}(\boldsymbol{\mu}^{(0)})\mathbf{Y}(\boldsymbol{\mu}^{(0)})^T) \frac{\partial \mathbf{Y}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}), \quad \text{(F.4)}
$$

the algorithm for building the derivatives at the origin for each $j = 1, \cdots, N_p$, proceeds in 4 steps as follows:

- Step 1. Build $\mathbf{H}^j(\boldsymbol{\mu}^{(0)})$.

- Step 2. Compute the thin SVD for $\mathbf{H}^j(\boldsymbol{\mu}^{(0)})$ as

$$\mathbf{H}^j(\boldsymbol{\mu}^{(0)}) = \mathbf{U}_j \boldsymbol{\Sigma}_j \mathbf{V}_j^T. \tag{F.5}$$

- Step 3. Build the logarithmic mappings of the perturbed subspace $\mathcal{S}_j$

$$\mathrm{Log}_{\mathcal{Y}_0}(\mathcal{S}_j) \leftrightarrow \boldsymbol{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu}_j) = \mathbf{U}_j \mathbf{tan}^{-1}(\delta\mu_j \boldsymbol{\Sigma}_j) \mathbf{V}_j^T. \tag{F.6}$$

- Step 4. Compute the derivatives for the logarithmic mapping at the origin by first order finite differences

$$\frac{\partial \boldsymbol{\Gamma}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) \approx \frac{\boldsymbol{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu}_j) - \boldsymbol{\Gamma}(\boldsymbol{\mu}^{(0)})}{\delta\mu_j} = \frac{\mathbf{U}_j \mathbf{tan}^{-1}(\delta\mu_j \boldsymbol{\Sigma}_j) \mathbf{V}_j^T}{\delta\mu_j}, \quad j = 1, \cdots, N_p, \tag{F.7}$$

using the property $\boldsymbol{\Gamma}(\boldsymbol{\mu}^{(0)}) = \mathbf{0}_{n,k}$ that holds since the tangent space to the Grassmann manifold at $\mathcal{Y}_0$ is considered.

F.2 Continuity and Derivability

Let $\delta\boldsymbol{\mu} = (\delta\mu_1, \cdots, \delta\mu_{N_p})$ be a vector of parameters satisfying the condition derived in Appendix E . Necessarily,

$$\mathrm{rank}[\mathbf{Y}(\boldsymbol{\mu}^{(0)}) + \nabla\mathbf{Y}(\boldsymbol{\mu}^{(0)}) \cdot \delta\boldsymbol{\mu}] = \mathrm{rank}[\mathbf{Y}(\boldsymbol{\mu}^{(0)})]. \tag{F.8}$$

A QR factorization and a calculation similar as the one in Eq. (F.3) leads to

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = (\mathbf{I}_n - \mathbf{Y}(\boldsymbol{\mu}^{(0)})\mathbf{Y}(\boldsymbol{\mu}^{(0)})^T)\nabla\mathbf{Y}(\boldsymbol{\mu}^{(0)}) \cdot \delta\boldsymbol{\mu} + \mathcal{O}(\|\delta\boldsymbol{\mu}\|_2^2) = \tilde{\mathbf{G}} \tag{F.9}$$

and

$$\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu}) = \mathbf{U}\mathbf{tan}^{-1}(\mathbf{\Sigma})\mathbf{V}^T. \tag{F.10}$$

Let $\mathbf{G}$ be the matrix defined as

$$\mathbf{G} = \left(\mathbf{I}_n - \mathbf{Y}(\boldsymbol{\mu}^{(0)})\mathbf{Y}(\boldsymbol{\mu}^{(0)})^T\right)\nabla\mathbf{Y}(\boldsymbol{\mu}^{(0)})\cdot\delta\boldsymbol{\mu} = \mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp,\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp}\nabla\mathbf{Y}(\boldsymbol{\mu}^{(0)})\cdot\delta\boldsymbol{\mu} \tag{F.11}$$

and let $\sigma_1(\mathbf{M}) = \|\mathbf{M}\|_2$ denote the largest singular value for a given matrix $\mathbf{M}$. Since $\tilde{\mathbf{G}} = \mathbf{G} + \mathbf{E}$ with $\|\mathbf{E}\|_2 \le \|\mathbf{G}\|_2$ for $\|\delta\boldsymbol{\mu}\|_2$ sufficiently small, $\|\tilde{\mathbf{G}}\|_2 \le 2\|\mathbf{G}\|_2$. As a consequence, the largest singular value for $\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu})$ satisfies

$$\begin{aligned}
\|\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu})\|_2 &= \sigma_1(\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu})) \\
&= \tan^{-1}(\sigma_1(\tilde{\mathbf{G}})) \\
&\le \sigma_1(\tilde{\mathbf{G}}) = \|\tilde{\mathbf{G}}\|_2 \\
&\le 2\|\mathbf{G}\|_2 \\
&\le 2\|\mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp,\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp}\nabla\mathbf{Y}(\boldsymbol{\mu}^{(0)}).\delta\boldsymbol{\mu}\|_2 \\
&\le 2\left\|\mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp,\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp}\sum_{j=1}^{N_p}\delta\mu_j\frac{\partial\mathbf{Y}}{\partial\mu_j}(\boldsymbol{\mu}^{(0)})\right\|_2 \\
&\le 2\left\|\mathbf{\Pi}_{\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp,\mathbf{Y}(\boldsymbol{\mu}^{(0)})^\perp}\sum_{j=1}^{N_p}\frac{\partial\mathbf{Y}}{\partial\mu_j}(\boldsymbol{\mu}^{(0)})\right\|_2\|\delta\boldsymbol{\mu}\|_2 \\
&\le C(\boldsymbol{\mu}^{(0)})\|\delta\boldsymbol{\mu}\|_2.
\end{aligned} \tag{F.12}$$

It is essential to note that the constant $C(\boldsymbol{\mu}^{(0)})$ only depends on the values of $\mathbf{Y}(\cdot)$ and its gradient at the point $\boldsymbol{\mu}^{(0)}$. Since $\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)}) = \mathbf{0}_{n,k}$,

$$\left\|\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu}) - \mathbf{\Gamma}(\boldsymbol{\mu}^{(0)})\right\|_2 \le C(\boldsymbol{\mu}^{(0)})\|\delta\boldsymbol{\mu}\|_2. \tag{F.13}$$

This is a very important inequality which shows that $\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\boldsymbol{\mu})$ is Lipschitz-continuous in the neighborhood of $\boldsymbol{\mu}^{(0)}$, and in particular that $\mathbf{\Gamma}(\boldsymbol{\mu})$ is continuous at $\boldsymbol{\mu}^{(0)}$. The differentiability can be established by considering an Analytic Singular Value Decomposition (ASVD) [120, 121] in Eq. (F.9) in place of the classic SVD. The derivability of the logarithm mapping is then immediate.

Furthermore, the expression for the derivative in Eq. (F.7) is second-order in $\delta\boldsymbol{\mu}$, as for a centered finite-difference derivative,

$$
\begin{aligned}
\frac{\partial \mathbf{\Gamma}}{\partial \mu_j}(\boldsymbol{\mu}^{(0)}) &= \frac{\mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} + \delta\mu_j \mathbf{e}_j) - \mathbf{\Gamma}(\boldsymbol{\mu}^{(0)} - \delta\mu_j \mathbf{e}_j)}{2\delta\mu_j} + \mathcal{O}(|\delta\mu_j|^2) \\
&= \frac{\mathbf{U}_j \tan^{-1}(\delta\mu_j \mathbf{\Sigma}_j)\mathbf{V}_j^T - \mathbf{U}_j \tan^{-1}(-\delta\mu_j \mathbf{\Sigma}_j)\mathbf{V}_j^T}{2\delta\mu_j} + \mathcal{O}(|\delta\mu_j|^2) \\
&= \frac{2\mathbf{U}_j \tan^{-1}(\delta\mu_j \mathbf{\Sigma}_j)\mathbf{V}_j^T}{2\delta\mu_j} + \mathcal{O}(|\delta\mu_j|^2) \\
&= \frac{\mathbf{U}_j \tan^{-1}(\delta\mu_j \mathbf{\Sigma}_j)\mathbf{V}_j^T}{\delta\mu_j} + \mathcal{O}(|\delta\mu_j|^2).
\end{aligned}
\tag{F.14}
$$

This shows that the second-order approximation of the derivative of the logarithmic map at the origin can be obtained at half the cost.

Appendix G

Interpolation of Lines on the Grassmann Manifold in $\mathbb{R}^3$

This special case corresponds to the interpolation of one dimensional linear spaces in $\mathbb{R}^3$. The bases $\{\mathbf{y}_i\}_{i=1}^{N_R} = \{\mathbf{y}(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R}$ representing these subspaces belong to $\mathcal{ST}(1,3)$ which is also the unit sphere $\mathbb{S}^3$ in $\mathbb{R}^3$. Hence, in the spherical set of coordinates, one can write

$$
\mathbf{y}(\boldsymbol{\mu}) = \begin{bmatrix} \cos\phi(\boldsymbol{\mu})\sin\theta(\boldsymbol{\mu}) \\ \sin\phi(\boldsymbol{\mu})\sin\theta(\boldsymbol{\mu}) \\ \cos\theta(\boldsymbol{\mu}) \end{bmatrix}. \tag{G.1}
$$

The interpolation algorithm is then followed step-by-step

- Step 0. A subspace $\mathcal{Y}(\boldsymbol{\mu}^{(i_0)})$ represented by $\mathbf{y}_{i_0}$ is chosen as a reference point. Without loss of generality, it is here assumed that $\phi(\boldsymbol{\mu}^{(i_0)}) = \theta(\boldsymbol{\mu}^{(i_0)}) = 0$, that

is

$$
\mathbf{y}_{i_0} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{G.2}
$$

- Step 1. Map each element $\mathbf{y}_i$ that lie in the neighborhood of $\mathbf{y}_{i_0}$ (the vectors whose directions that are not orthogonal to $\mathcal{Y}(\boldsymbol{\mu}^{(i_0)})$) to $T_{\mathcal{Y}_{i_0}}\mathcal{G}(1,3)$ as

$$
(\mathbf{I}_3 - \mathbf{y}_{i_0}\mathbf{y}_{i_0}^T)\mathbf{y}_i(\mathbf{y}_{i_0}^T\mathbf{y}_i)^{-1} = \tilde{\mathbf{u}}\tilde{\sigma}\tilde{v} \quad \text{(thin SVD)} \tag{G.3}
$$

$$
\boldsymbol{\gamma}_i = \tilde{\mathbf{u}}\tan^{-1}(\tilde{\sigma})\tilde{v}. \tag{G.4}
$$

In the present case $\tilde{\sigma}$, $\tilde{v} \in \mathbb{R}$, and without any loss of generality, one can assume that $\tilde{v} = 1$, since its 2-norm is one. Furthermore,

$$
\mathbf{y}_{i_0}^T\mathbf{y}_i = \cos\theta(\boldsymbol{\mu}^{(i)}), \tag{G.5}
$$

and the mappings (G.3)–(G.4) can be written as

$$
\begin{bmatrix} \cos\phi(\boldsymbol{\mu}^{(i)})\tan\theta(\boldsymbol{\mu}^{(i)}) \\ \sin\phi(\boldsymbol{\mu}^{(i)})\tan\theta(\boldsymbol{\mu}^{(i)}) \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\phi(\boldsymbol{\mu}^{(i)}) \\ \sin\phi(\boldsymbol{\mu}^{(i)}) \\ 0 \end{bmatrix} \tan\theta(\boldsymbol{\mu}^{(i)}) \text{ (thin SVD)} \tag{G.6}
$$

$$
\boldsymbol{\gamma}_i = \begin{bmatrix} \theta(\boldsymbol{\mu}^{(i)})\cos\phi(\boldsymbol{\mu}^{(i)}) \\ \theta(\boldsymbol{\mu}^{(i)})\sin\phi(\boldsymbol{\mu}^{(i)}) \\ 0 \end{bmatrix}, \tag{G.7}
$$

if $0 \leq \theta(\boldsymbol{\mu}^{(i)}) < \frac{\pi}{2}$ and, by a thin SVD,

$$
\begin{bmatrix} \cos \phi(\boldsymbol{\mu}^{(i)}) \tan \theta(\boldsymbol{\mu}^{(i)}) \\ \sin \phi(\boldsymbol{\mu}^{(i)}) \tan \theta(\boldsymbol{\mu}^{(i)}) \\ 0 \end{bmatrix} = \begin{bmatrix} -\cos \phi(\boldsymbol{\mu}^{(i)}) \\ -\sin \phi(\boldsymbol{\mu}^{(i)}) \\ 0 \end{bmatrix} \tan(\pi - \theta(\boldsymbol{\mu}^{(i)})) \quad \text{(G.8)}
$$

$$
\boldsymbol{\gamma}_i = \begin{bmatrix} (\theta(\boldsymbol{\mu}^{(i)}) - \pi) \cos \phi(\boldsymbol{\mu}^{(i)}) \\ (\theta(\boldsymbol{\mu}^{(i)}) - \pi) \sin \phi(\boldsymbol{\mu}^{(i)}) \\ 0 \end{bmatrix}, \quad \text{(G.9)}
$$

if $\frac{\pi}{2} < \theta(\boldsymbol{\mu}^{(i)}) \leq \pi$.

One should note at this point that each one-dimensional subspace $\mathcal{Y}_i$, $i = 1, \cdots, N_R$, $i \neq i_0$ can be represented by one of two bases, namely

$$
\mathbf{y}_a(\boldsymbol{\mu}^{(i)}) = \begin{bmatrix} \cos \phi_i \sin \theta_i \\ \sin \phi_i \sin \theta_i \\ \cos \theta_i \end{bmatrix}, \quad \mathbf{y}_b(\boldsymbol{\mu}^{(i)}) = \begin{bmatrix} \cos(\phi_i + \pi) \sin(\pi - \theta_i) \\ \sin(\phi_i + \pi) \sin(\pi - \theta_i) \\ \cos(\pi - \theta_i) \end{bmatrix}, \quad \text{(G.10)}
$$

where $\phi_i = \phi(\boldsymbol{\mu}^{(i)}) \in [0, \ 2\pi]$ and $\theta_i = \theta(\boldsymbol{\mu}^{(i)}) \in [0, \ \frac{\pi}{2})$. The representant of $\text{Log}_{\mathcal{Y}_{i_0}}(\mathcal{Y}_i)$, $i = 1, \cdots, N_R$, $i \neq i_0$ using $\mathbf{y}_a(\boldsymbol{\mu}^{(i)})$ is given by Eq. (G.7)

$$
\boldsymbol{\gamma}_i = \begin{bmatrix} \theta_i \cos \phi_i \\ \theta_i \sin \phi_i \\ 0 \end{bmatrix}. \quad \text{(G.11)}
$$

If $\mathbf{y}_b(\boldsymbol{\mu}^{(i)})$ is used instead, $\text{Log}_{\mathcal{Y}_{i_0}}(\mathcal{Y}_i)$, $i = 1, \cdots, N_R$, $i \neq i_0$ is represented by

Eq. (G.9):

$$\boldsymbol{\gamma}_i = \begin{bmatrix} (\pi - \theta_i - \pi)\cos(\phi_i + \pi) \\ (\pi - \theta_i - \pi)\sin(\phi_i + \pi) \\ 0 \end{bmatrix} = \begin{bmatrix} \theta_i \cos\phi_i \\ \theta_i \sin\phi_i \\ 0 \end{bmatrix}. \qquad (G.12)$$

In conclusion, $\mathrm{Log}_{\mathcal{Y}_{i_0}}$ maps each $\mathcal{Y}_i$, $i = 1, \cdots, N_R$ to a vector γ_i which does not depend on its representative basis $\mathbf{y}_i$ and which is of the form

$$\boldsymbol{\gamma}_i = \begin{bmatrix} \theta_i \cos\phi_i \\ \theta_i \sin\phi_i \\ 0 \end{bmatrix}, \qquad (G.13)$$

with $\phi_i \in [0,\ 2\pi]$ and $\theta_i \in [0,\ \frac{\pi}{2})$. One can hence write

$$\mathrm{Log}_{\mathcal{Y}_{i_0}} : \mathcal{U}_{\mathcal{Y}_{i_0}} \in \mathcal{G}(1,3) \longrightarrow \mathbb{S}^2\left(\frac{\pi}{2}\right). \qquad (G.14)$$

$\mathrm{Log}_{\mathcal{Y}_{i_0}}$ maps the set $\mathcal{U}_{\mathcal{Y}_{i_0}}$ of one-dimensional subspaces in $\mathbb{R}^3$ that are not orthogonal to $\mathcal{Y}_{i_0}$, to the two-dimensional sphere $\mathbb{S}^2\left(\frac{\pi}{2}\right)$ of radius $\frac{\pi}{2}$ (see Fig. G.1). Furthermore, one should note that although $\boldsymbol{\gamma}_i \in \mathbb{R}^3$, its last coordinate entry is zero. This is a direct result of the fact that the dimension of the tangent space to a Riemannian manifold has the same dimension as the manifold itself. In the present case dim $\mathcal{G}(1,3) = 1(3-1) = 2$, which is consistent with the form of $\boldsymbol{\gamma}_i$.

- Step 2. Compute $\boldsymbol{\gamma}_{N_R+1}$ associated with $\mu^{(N_R+1)}$ by interpolating the entries of the vectors $\boldsymbol{\gamma}_i$, $i = 1, \cdots, N_R$. Necessarily, the particular form of the vectors
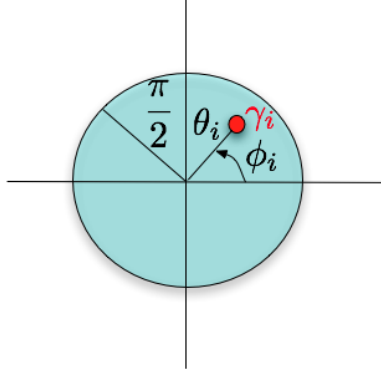
Figure G.1: Logarithmic mapping on the tangent space to $\mathcal{G}(1,3)$.

$\boldsymbol{\gamma}_i$ implies that $\boldsymbol{\gamma}_{N_R+1}$ will be of the form

$$
\boldsymbol{\gamma}_{N_R+1} = \begin{bmatrix} x_{N_R+1} \\ y_{N_R+1} \\ 0 \end{bmatrix},
\tag{G.15}
$$

where x_{N_R+1} is interpolated from the quantities $\{\theta_i \cos \phi_i\}_{i=1}^{N_R}$ and y_{N_R+1} from the quantities $\{\theta_i \sin \phi_i\}_{i=1}^{N_R}$. Hence, if $\boldsymbol{\gamma}_{N_R+1} \neq \mathbf{0}$, one can denote

$$
\theta_{N_R+1} = \sqrt{x_{N_R+1}^2 + y_{N_R+1}^2},
\tag{G.16}
$$

and

$$
\cos \phi_{N_R+1} = \frac{x_{N_R+1}}{\sqrt{x_{N_R+1}^2 + y_{N_R+1}^2}}, \quad \sin \phi_{N_R+1} = \frac{y_{N_R+1}}{\sqrt{x_{N_R+1}^2 + y_{N_R+1}^2}}.
\tag{G.17}
$$

- Step 3. $\mathcal{V}_{N_R+1}$ represented by $\boldsymbol{\gamma}_{N_R+1}$ is mapped to

$$
\mathrm{Exp}_{\mathbf{y}_{i_0}}(\mathcal{V}_{N_R+1}) = \mathrm{span}\left\{ \mathbf{y}_{i_0} v \cos(\sigma) + \mathbf{u} \sin(\sigma) \right\}
\tag{G.18}
$$

with $\boldsymbol{\gamma}_{N_R+1} = \mathbf{u}\sigma v$, a thin SVD. In the present case, $v = 1$, $\sigma = \theta_{N_R+1}$ and

$$\mathbf{u} = \begin{bmatrix} \cos \phi_{N_R+1} \\ \sin \phi_{N_R+1} \\ 0 \end{bmatrix}. \tag{G.19}$$

Hence

$$\mathrm{Exp}_{\mathcal{Y}_{i_0}}(\mathcal{V}_{N_R+1}) = \mathrm{span} \left\{ \begin{bmatrix} \cos \phi_{N_R+1} \sin \theta_{N_R+1} \\ \sin \phi_{N_R+1} \sin \theta_{N_R+1} \\ \cos \theta_{N_R+1} \end{bmatrix} \right\}, \tag{G.20}$$

and the algorithm has proceeded by naturally interpolating the angles (in the spherical set of coordinates) between the pre-computed lines $\{\mathcal{Y}(\boldsymbol{\mu}^{(i)})\}_{i=1}^{N_R}$ and their reference counterpart $\mathcal{Y}(\boldsymbol{\mu}^{(i_0)})$.

Appendix H

Interpolation with Radial Basis Functions

H.1   Interpolation Problem

The bivariate case (two parameters $N_p = 2$) is considered in this section without any loss of generality. The extension of the method to higher values of N_p is straightforward.

Let $\mathcal{D}$ be a continuous subdomain of $\mathbb{R}^2$ and let N arbitrarily distributed points $P_i = (x_i, y_i) \in \mathcal{D}, \ i = 1, \cdots, N$ be considered.

Problem 1. Let F be a continuous function defined on $\mathcal{D}$. Find a smooth interpolating function $H(x, y)$ satisfying

$$H(x_i, y_i) = F(x_i, y_i) \ \ i = 1, \cdots, N. \tag{H.1}$$

Problem 2. Let F be a continuously differentiable function defined on $\mathcal{D}$. Find

a smooth interpolating function $H_d(x, y)$ satisfying

$$
\begin{aligned}
H_d(x_i, y_i) &= F(x_i, y_i) \quad i = 1, \cdots, N & \text{(H.2)} \\
\frac{\partial H_d}{\partial x}(x_i, y_i) &= \frac{\partial F}{\partial x}(x_i, y_i) \quad i = 1, \cdots, N, & \text{(H.3)} \\
\frac{\partial H_d}{\partial y}(x_i, y_i) &= \frac{\partial F}{\partial y}(x_i, y_i) \quad i = 1, \cdots, N, . & \text{(H.4)}
\end{aligned}
$$

H.2 Radial Basis Functions

Definition 1. Let $\beta > 0$. The Gaussian radial basis function is defined as

$$
B(r) = \exp\left(-\beta r^2\right). \tag{H.5}
$$

Definition 2. Let $R > 0$, $q > 0$. Hardy's multiquadrics radial basis function is defined as

$$
B(r) = \left(r^2 + R^2\right)^q. \tag{H.6}
$$

Definition 3. Let $k \in \mathbb{N}$. The polyharmonic radial basis function is defined as

$$
B(r) = \begin{cases} r^k \log r & \text{if } k \text{ is even} \\ r^k & \text{if } k \text{ is odd} \end{cases} \tag{H.7}
$$

H.3 Non-Compact Interpolation Method

An interpolating function of the following form is considered [76]

$$H(x, y) = \sum_{k=1}^{N} a_k B_k(x, y) \tag{H.8}$$

where the B_k are basis functions and the a_k are determined by a series of interpolation constraints

$$\sum_{k=1}^{N} a_k B_k(x_i, y_i) = F(x_i, y_i), \quad i = 1, \cdots, N. \tag{H.9}$$

The basis functions B_k are chosen so that H is smooth. One of the possibilities is to choose radial basis functions

$$B_k(x, y) = B(r_k), \quad r_k = \sqrt{(x - x_k)^2 + (y - y_k)^2}, \tag{H.10}$$

where B is one of the functions defined in section H.2.

If Hardy's multiquadrics are used, it is recommended in Ref. [76] to choose $q = \dfrac{1}{4}$ and

$$R = \sqrt{\frac{1}{10} \max \left(\max_{1 \leq i,k \leq N} |x_i - x_k|, \max_{1 \leq i,k \leq N} |y_i - y_k| \right)}. \tag{H.11}$$

From Eq. (H.9) it can be observed that, in order to determine the coefficients a_k in the interpolating function, an N-by-N linear system needs to be solved.

H.4 Compact Interpolation Method

In this section, information about the value of the derivatives of the function F is taken into account when the interpolation function is determined. For that purpose,

an interpolating function of the following form is considered —the method in Ref. [122] is extended here to any continuously differentiable radial basis function—

$$H_d(x, y) = \sum_{k=1}^{N} a_k B_k(x, y) + \sum_{k=1}^{N} b_k \frac{\partial B_k}{\partial x}(x, y) + \sum_{k=1}^{N} c_k \frac{\partial B_k}{\partial y}(x, y). \tag{H.12}$$

The different sets of coefficients a_k, b_k and c_k are determined by considering the interpolation constraints (H.2)–(H.4):

$$\sum_{k=1}^{N} a_k B_k(x_i, y_i) + \sum_{k=1}^{N} b_k \frac{\partial B_k}{\partial x}(x_i, y_i) + \sum_{k=1}^{N} c_k \frac{\partial B_k}{\partial y}(x_i, y_i) = F(x_i, y_i) \quad i = 1, \cdots, N,$$

$$\sum_{k=1}^{N} a_k \frac{\partial B_k}{\partial x}(x_i, y_i) + \sum_{k=1}^{N} b_k \frac{\partial^2 B_k}{\partial x^2}(x_i, y_i) + \sum_{k=1}^{N} c_k \frac{\partial^2 B_k}{\partial x \partial y}(x_i, y_i) = \frac{\partial F}{\partial x}(x_i, y_i) \quad i = 1, \cdots, N,$$

$$\sum_{k=1}^{N} a_k \frac{\partial B_k}{\partial y}(x_i, y_i) + \sum_{k=1}^{N} b_k \frac{\partial^2 B_k}{\partial x \partial y}(x_i, y_i) + \sum_{k=1}^{N} c_k \frac{\partial^2 B_k}{\partial y^2}(x_i, y_i) = \frac{\partial F}{\partial y}(x_i, y_i) \quad i = 1, \cdots, N.$$

$$\tag{H.13}$$

Computing a_k, b_k, c_k, $k = 1, \cdots, N$ involves solving a linear system of size $3N$. More generally, for a multivariate interpolation in dimension N_p, the linear system to be solved is of size $(N_p + 1)N$.

Appendix I

Iterative Methods for the Resolution of Large Sparse Linear Systems

I.1   Introduction

A left-hand side matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$, a right-hand side vector $\mathbf{b} \in \mathbb{R}^n$ and an initial guess $\mathbf{x}_0 \in \mathbb{R}^n$ are given. $\mathbf{A}$ is here assumed to be non-symmetric. The Krylov subspace iterative methods that are considered here solve the following linear problem

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{I.1}$$

by computing at each iteration m an approximation $\mathbf{x}_m$ to $\mathbf{x}$ such that $\mathbf{x}_m$ belongs to the affine subspace $\mathbf{x}_0 + \mathcal{K}_m$ where

$$\mathcal{K}_m = \mathrm{span}\left\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \cdots, \mathbf{A}^{m-1}\mathbf{r}_0\right\} \tag{I.2}$$

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is referred to as a Krylov subspace.

The two methods that are presented in this Appendix compute $\mathbf{x}_m$ such that it is solution to the following minimization problem

$$\min_{\mathbf{y} \in \mathbf{x}_0 + \mathcal{K}_m} \|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2. \tag{I.3}$$

As such, both methods produce identical vectors $\mathbf{x}_m$ at the m-th iteration. While it can be shown that the GMRES method leads to half the computational cost of the GCR method —both in terms of operations and memory storage,— the GCR method can be embedded in a Krylov subspace recycling procedure for multiple right-hand sides problems, as presented in section I.4.

I.2 Generalized Minimal Residual Algorithm

The GMRES procedure is reported in Algorithm 7.

I.3 Generalized Conjugate Residual Algorithm

The GCR procedure is described in Algorithm 8.

I.4 A Method for Solving Linear Systems with Multiple Right-Hand Sides

This method was first proposed in Ref. [124] as an extension to non-symmetric systems of the method originally developed in [125] for the Conjugate Gradient Algorithm.

Algorithm 7 GMRES

Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$.
$\beta \leftarrow \|\mathbf{r}_0\|_2$
$\mathbf{v}_1 \leftarrow \mathbf{r}_0/\beta$
$j \leftarrow 1$
repeat
  Compute $\mathbf{w}_j = -\mathbf{A}\mathbf{v}_j$.
  for $i = 1, \cdots, j$ **do**
    $h_{ij} = \mathbf{w}_j^T \mathbf{v}_i$
    $\mathbf{w}_j = \mathbf{w}_j - h_{ij}\mathbf{v}_i$
  end for
  $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|_2$
  if $h_{j+1,j} = 0$ **then**
    $m \leftarrow j$
    Exit the loop
  end if{Rank-one update for the approximate Jacobian}
  $j \leftarrow j + 1$
until $h_{j+1,j} > 0$
$\mathbf{H} = \left[h_{i,j}\right]_{1 \leq i \leq m+1, 1 \leq j \leq m}$
Solve $\mathbf{y}_m = \mathrm{argmin}\|\beta\mathbf{e}_1 - \mathbf{H}\mathbf{y}\|$.
$\mathbf{x} \leftarrow \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m$.

The following set of linear systems is considered

$$\mathbf{A}\mathbf{x}^{(i)} = \mathbf{b}^{(i)}, \quad i = 1, \cdots, N_{RHS}. \tag{I.4}$$

The main idea is to reuse the previously computed Krylov subspaces in order to accelerate the convergence of the Krylov iterations for the next systems.

The GCR algorithm computes, for a given system, search directions $\mathbf{p}_1, \cdots, \mathbf{p}_l$, which are $\mathbf{A}^T\mathbf{A}$-conjugate, that is

$$\mathbf{p}_i^T \mathbf{A}^T \mathbf{A} \mathbf{p}_j = 0, \quad \text{if } i \neq j. \tag{I.5}$$

The method is here illustrated in the case $N_{RHS} = 2$ but can be extended to a larger

---

Algorithm 8 GCR

Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$.

$\mathbf{p}_0 \leftarrow \mathbf{r}_0$

Choose $\eta > 0$.

repeat

$$\alpha_j = \frac{\mathbf{r}_j^T(\mathbf{A}\mathbf{p}_j)}{(\mathbf{A}\mathbf{p}_j)^T(\mathbf{A}\mathbf{p}_j)}$$

$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j$

$\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{A}\mathbf{p}_j$

for $i = 0, \cdots, j$ **do**

$$\beta_{ij} = -\frac{(\mathbf{A}\mathbf{r}_{j+1})^T(\mathbf{A}\mathbf{p}_i)}{(\mathbf{A}\mathbf{p}_i)^T(\mathbf{A}\mathbf{p}_i)}.$$

end for

$\mathbf{p}_{j+1} = \mathbf{r}_{j+1} + \sum_{i=0}^{j} \beta_{ij}\mathbf{p}_i$

until $\|\mathbf{r}_j\|_2 < \eta$ {Metric for convergence}

$\mathbf{x} \leftarrow \mathbf{x}_{j+1}$

---

number of systems in a straightforward fashion. The algorithm proceeds as follows

1. Solve the first linear system

$$\mathbf{A}\mathbf{x}^{(1)} = \mathbf{b}^{(1)} \tag{I.6}$$

using the GCR method. A Krylov subspace $\mathcal{K}_m^{(1)} = \mathrm{span}\left\{\mathbf{p}_1, \cdots, \mathbf{p}_m\right\}$ is hence generated. The search directions are are $\mathbf{A}^T\mathbf{A}$-conjugate.

2. The solution $\mathbf{x}^{(2)}$ to

$$\mathbf{A}\mathbf{x}^{(2)} = \mathbf{b}^{(2)} \tag{I.7}$$

is looked for as

$$\mathbf{x}^{(2)} = \mathbf{x}_0^{(2)} + \mathbf{y}^{(2)}, \quad \mathbf{x}_0^{(2)} \in \mathcal{K}_m^{(1)}, \ \mathbf{y}^{(2)} \in \mathcal{L}, \tag{I.8}$$

where the subspaces $\mathcal{K}_m$ and $\mathcal{L}$ are orthogonal in the $\mathbf{A}^T\mathbf{A}$-norm, that is,

$$\mathbf{p}^T\mathbf{A}^T\mathbf{A}\mathbf{q} = 0, \quad \forall \mathbf{p} \in \mathcal{K}_m, \ \forall \mathbf{q} \in \mathcal{L}. \tag{I.9}$$

Letting $\mathbf{P}^{(1)} = \begin{bmatrix} \mathbf{p}_1 & \cdots & \mathbf{p}_m \end{bmatrix}$, $\mathbf{x}_0^{(2)} = \mathbf{P}^{(1)}\mathbf{q}_0^{(2)}$ can be readily computed as the solution to

$$\mathbf{P}^{(1)T}\mathbf{A}^T\mathbf{A}\mathbf{P}^{(1)}\mathbf{q}_0^{(2)} = \mathbf{P}^{(1)T}\mathbf{A}^T\mathbf{b}^{(2)}, \tag{I.10}$$

which is a diagonal system because of the $\mathbf{A}^T\mathbf{A}$-conjugacy property of the search directions. This system can be solved at a very low cost.

3. $\mathbf{y}^{(2)}$ is then computed with the GCR method by linear combinations of search directions $\mathbf{p}_{m+1}, \mathbf{p}_{m+2}, \cdots$, that are $\mathbf{A}^T\mathbf{A}$-conjugate to $\mathbf{p}_1, \cdots, \mathbf{p}_m$.

Appendix J

Staggered Time Integration Scheme for the Aeroelastic System

As presented in section 4.4.7, the general form of the BDF scheme for the fluid equation in the aeroelastic coupled system is

$$\alpha_2 \bar{\mathbf{A}} \mathbf{w}^{n+\frac{5}{2}} + \alpha_1 \bar{\mathbf{A}} \mathbf{w}^{n+\frac{3}{2}} + \alpha_0 \bar{\mathbf{A}} \mathbf{w}^{n+\frac{1}{2}} + \Delta t_n \left(\bar{\mathbf{H}} \mathbf{w}^{n+\frac{5}{2}} + (\bar{\mathbf{E}} + \bar{\mathbf{C}}) \bar{\mathbf{K}} \dot{\mathbf{u}}^{n+\frac{5}{2}^P} + \bar{\mathbf{G}} \bar{\mathbf{K}} \mathbf{u}^{n+\frac{5}{2}^P} \right) = \mathbf{0},$$

$$(\text{J.1})$$

This appendix presents the associated staggered scheme for time-integrating the coupled fluid-structure system.

J.1   Time Iteration 1

$\mathbf{w}^0$, $\mathbf{u}^0$ and $\dot{\mathbf{u}}^0$ are known initially. $\tilde{\Delta} t = \Delta t^2$. The Explicit Euler scheme for the first iteration goes as follows:

1. Predict $\tilde{\mathbf{u}}$ and $\dot{\tilde{\mathbf{u}}}$

$$\tilde{\mathbf{u}}^P = \mathbf{u}^0 + \frac{\tilde{\Delta}t}{2}\dot{\mathbf{u}}^0, \tag{J.2}$$

$$\dot{\tilde{\mathbf{u}}}^P = \dot{\mathbf{u}}^0. \tag{J.3}$$

2. Obtain $\tilde{\mathbf{w}}$ by solving

$$\frac{\bar{\mathbf{A}}}{\tilde{\Delta}t}\tilde{\mathbf{w}} = \left[\frac{\bar{\mathbf{A}}}{\tilde{\Delta}t} - \bar{\mathbf{H}}\right]\mathbf{w}^0 - \bar{\mathbf{G}}\bar{\mathbf{K}}\tilde{\mathbf{u}}^P - (\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}}\dot{\tilde{\mathbf{u}}}^P. \tag{J.4}$$

3. Obtain $\dot{\tilde{\mathbf{u}}}$ by solving

$$\left[\frac{\mathbf{M}}{\tilde{\Delta}t} + \frac{\tilde{\Delta}t}{2}\mathbf{K}_s\right]\dot{\tilde{\mathbf{u}}} = \mathbf{P}\tilde{\mathbf{w}} - \mathbf{K}_s\mathbf{u}^0 + \left[\frac{\mathbf{M}}{\tilde{\Delta}t} - \frac{\tilde{\Delta}t}{2}\mathbf{K}_s\right]\dot{\mathbf{u}}^0 \tag{J.5}$$

4. Obtain $\tilde{\mathbf{u}}$ as

$$\tilde{\mathbf{u}} = \frac{\tilde{\Delta}t}{2}(\dot{\tilde{\mathbf{u}}} + \dot{\mathbf{u}}^0) + \mathbf{u}^0. \tag{J.6}$$

It can be shown that this first iteration is first-order accurate in time.

J.2 Time Iteration 2

A three-step BDF scheme is used for the second time-step as follows :

1. Predict $\mathbf{u}^{\frac{1}{2}}$ and $\dot{\mathbf{u}}^{\frac{1}{2}}$

$$\mathbf{u}^{\frac{1}{2}P} = \tilde{\mathbf{u}} + \frac{\Delta t}{2}\dot{\tilde{\mathbf{u}}} \tag{J.7}$$

$$\dot{\mathbf{u}}^{\frac{1}{2}P} = (1 + \tau_1)\dot{\tilde{\mathbf{u}}} - \tau_1\dot{\mathbf{u}}^0 \tag{J.8}$$

with $\Delta t_1 = \dfrac{\Delta t}{2}$ and thus $\tau_1 = \dfrac{\Delta t}{2\tilde{\Delta} t}$.

2. Define $\alpha_2 = \dfrac{1 + 2\tau_1}{\tau_1(1 + \tau_1)}$, $\alpha_1 = -1 - \dfrac{1}{\tau_1}$ and $\alpha_0 = \dfrac{\tau_1}{1 + \tau_1}$ and obtain $\mathbf{w}^{\frac{1}{2}}$ by solving

$$\left[\frac{1 + 2\tau_1}{\tau_1(1 + \tau_1)}\frac{\bar{\mathbf{A}}}{\tilde{\Delta} t} + \bar{\mathbf{H}}\right]\mathbf{w}^{\frac{1}{2}} = \left(1 + \frac{1}{\tau_1}\right)\frac{\bar{\mathbf{A}}}{\tilde{\Delta} t}\tilde{\mathbf{w}} - \frac{\tau_1}{1 + \tau_1}\frac{\bar{\mathbf{A}}}{\tilde{\Delta} t}\mathbf{w}^0 - \bar{\mathbf{G}}\bar{\mathbf{K}}\mathbf{u}^{\frac{1}{2}P} - (\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}}\dot{\mathbf{u}}^{\frac{1}{2}P}. \tag{J.9}$$

3. Obtain $\dot{\mathbf{u}}^1$ by solving

$$\left[\frac{\mathbf{M}}{\Delta t} + \frac{\Delta t}{4}\mathbf{K}_s\right]\dot{\mathbf{u}}^1 = \mathbf{P}\mathbf{w}^{\frac{1}{2}} - \mathbf{K}_s\tilde{\mathbf{u}} + \left[\frac{\mathbf{M}}{\Delta t} - \frac{\Delta t}{4}\mathbf{K}_s\right]\dot{\tilde{\mathbf{u}}}. \tag{J.10}$$

4. Obtain $\mathbf{u}^1$ as

$$\mathbf{u}^1 = \frac{\Delta t}{2}(\dot{\mathbf{u}}^1 + \dot{\tilde{\mathbf{u}}}) + \tilde{\mathbf{u}}. \tag{J.11}$$

It can be shown that this scheme second-order accurate in time.

J.3 Time Iteration 3

The three-step BDF scheme used for the third time-step goes as follows:

1. Predict $\mathbf{u}^{\frac{3}{2}}$ and $\dot{\mathbf{u}}^{\frac{3}{2}}$

$$\mathbf{u}^{\frac{3}{2}P} = \mathbf{u}^1 + \frac{\Delta t}{2}\dot{\mathbf{u}}^1 \tag{J.12}$$

$$\dot{\mathbf{u}}^{\frac{3}{2}P} = \frac{3}{2}\dot{\mathbf{u}}^1 - \frac{1}{2}\dot{\tilde{\mathbf{u}}}, \tag{J.13}$$

as $\Delta t_2 = \Delta t$ and $\Delta t_1 = \dfrac{\Delta t}{2}$, thus $\tau_2 = \dfrac{1}{2}$.

2. Defining $\alpha_2 = \dfrac{4}{3}$, $\alpha_1 = -\dfrac{3}{2}$ and $\alpha_0 = \dfrac{1}{6}$, obtain $\mathbf{w}^{\frac{3}{2}}$ by solving

$$\left[\frac{4}{3}\frac{\bar{\mathbf{A}}}{\Delta t} + \bar{\mathbf{H}}\right]\mathbf{w}^{\frac{3}{2}} = \frac{3}{2}\frac{\bar{\mathbf{A}}}{\Delta t}\mathbf{w}^{\frac{1}{2}} - \frac{1}{6}\frac{\bar{\mathbf{A}}}{\Delta t}\tilde{\mathbf{w}} - \bar{\mathbf{G}}\bar{\mathbf{K}}\mathbf{u}^{\frac{3}{2}\,P} - (\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}}\dot{\mathbf{u}}^{\frac{3}{2}\,P}. \tag{J.14}$$

3. Obtain $\dot{\mathbf{u}}^2$ by solving

$$\left[\frac{\mathbf{M}}{\Delta t} + \frac{\Delta t}{4}\mathbf{K}_s\right]\dot{\mathbf{u}}^2 = \mathbf{P}\mathbf{w}^{\frac{3}{2}} - \mathbf{K}_s\mathbf{u}^1 + \left[\frac{\mathbf{M}}{\Delta t} - \frac{\Delta t}{4}\mathbf{K}_s\right]\dot{\mathbf{u}}^1. \tag{J.15}$$

4. Obtain $\mathbf{u}^2$ as

$$\mathbf{u}^2 = \frac{\Delta t}{2}(\dot{\mathbf{u}}^2 + \dot{\mathbf{u}}^1) + \mathbf{u}^1. \tag{J.16}$$

This scheme second-order accurate in time.

J.4 Successive Time Iterations

A three-step BDF scheme with uniform time-stepping is now used.

1. Predict $\mathbf{u}^{n+\frac{1}{2}}$, $\dot{\mathbf{u}}^{n+\frac{1}{2}}$

$$\mathbf{u}^{n+\frac{1}{2}\,P} = \mathbf{u}^n + \frac{\Delta t}{2}\dot{\mathbf{u}}^n \tag{J.17}$$

$$\dot{\mathbf{u}}^{n+\frac{1}{2}\,P} = \frac{3}{2}\dot{\mathbf{u}}^n - \frac{1}{2}\dot{\mathbf{u}}^{n-1}, \tag{J.18}$$

since the mesh is now equispaced in time, $\tau_{n+1} = 1$.

2. Defining $\alpha_2 = \dfrac{3}{2}$, $\alpha_1 = -2$ and $\alpha_0 = \dfrac{1}{2}$, obtain $\mathbf{w}^{n+\frac{1}{2}}$ by solving

$$\left[\frac{3\bar{\mathbf{A}}}{2\Delta t} + \bar{\mathbf{H}}\right] \mathbf{w}^{n+\frac{1}{2}} = \frac{2\bar{\mathbf{A}}}{\Delta t}\mathbf{w}^{n-\frac{1}{2}} - \frac{\bar{\mathbf{A}}}{2\Delta t}\mathbf{w}^{n-\frac{3}{2}} - \bar{\mathbf{G}}\bar{\mathbf{K}}\mathbf{u}^{n+\frac{1}{2}P} - (\bar{\mathbf{E}} + \bar{\mathbf{C}})\bar{\mathbf{K}}\dot{\mathbf{u}}^{n+\frac{1}{2}P}.$$

(J.19)

3. Obtain $\dot{\mathbf{u}}^{n+1}$ by solving

$$\left[\frac{\mathbf{M}}{\Delta t} + \frac{\Delta t}{4}\mathbf{K}_s\right] \dot{\mathbf{u}}^{n+1} = \mathbf{P}\mathbf{w}^{n+\frac{1}{2}} - \mathbf{K}_s\mathbf{u}^n + \left[\frac{\mathbf{M}}{\Delta t} - \frac{\Delta t}{4}\mathbf{K}_s\right] \dot{\mathbf{u}}^n$$

(J.20)

4. Obtain $\mathbf{u}^{n+1}$ as

$$\mathbf{u}^{n+1} = \frac{\Delta t}{2}(\dot{\mathbf{u}}^{n+1} + \dot{\mathbf{u}}^n) + \mathbf{u}^n.$$

(J.21)

This scheme is also second-order accurate in time.

Appendix K

Proofs of Propositions and Theorems in Chapter 7

K.1 Proof of Proposition 1

Let $\left(\mathbf{V}(\boldsymbol{\mu}), \mathbf{W}(\boldsymbol{\mu})\right)$ be two representative bases of the trial and test subspaces

$$\left(\mathcal{S}_{\mathbf{V}}(\boldsymbol{\mu}), \mathcal{S}_{\mathbf{W}}(\boldsymbol{\mu})\right). \tag{K.1}$$

The transfer function $\mathbf{H}_r(s; \boldsymbol{\mu})$ in Eq. (2.18) uniquely defines the input-ouput behavior of the reduced system. In terms of the bases $\mathbf{V}(\boldsymbol{\mu})$ and $\mathbf{W}(\boldsymbol{\mu})$, $\mathbf{H}_r$ can be written as

$$\mathbf{H}_r(s; \boldsymbol{\mu}) =$$
$$\mathbf{C}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\left(s\mathbf{I}_k - \left(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{V}(\boldsymbol{\mu})\right)^{-1}\mathbf{W}(\boldsymbol{\mu})^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}(\boldsymbol{\mu})\right)^{-1}\left(\mathbf{W}(\boldsymbol{\mu})^T\mathbf{V}(\boldsymbol{\mu})\right)^{-1}\mathbf{W}(\boldsymbol{\mu})^T\mathbf{B}(\boldsymbol{\mu}). \tag{K.2}$$

Let $\left(\tilde{\mathbf{V}}(\boldsymbol{\mu}), \tilde{\mathbf{W}}(\boldsymbol{\mu})\right)$ be two other representative bases of the same subspaces. One can then define two non-singular matrices $\mathbf{P}(\boldsymbol{\mu})$, $\mathbf{Q}(\boldsymbol{\mu})$ such that

$$\tilde{\mathbf{V}}(\boldsymbol{\mu}) = \mathbf{V}(\boldsymbol{\mu})\mathbf{P}(\boldsymbol{\mu}), \quad \tilde{\mathbf{W}}(\boldsymbol{\mu}) = \mathbf{W}(\boldsymbol{\mu})\mathbf{Q}(\boldsymbol{\mu}). \tag{K.3}$$

The dependence of the matrices in terms of the vector of parameters $\boldsymbol{\mu}$ is subsequently dropped in order to alleviate the notations. The transfer function $\tilde{\mathbf{H}}_r$ of the resulting ROM is

$$\tilde{\mathbf{H}}_r(s) = \mathbf{C}\tilde{\mathbf{V}}\left(s\mathbf{I}_k - \left(\tilde{\mathbf{W}}^T\tilde{\mathbf{V}}\right)^{-1}\tilde{\mathbf{W}}^T\mathbf{A}\tilde{\mathbf{V}}\right)^{-1}\left(\tilde{\mathbf{W}}^T\tilde{\mathbf{V}}\right)^{-1}\tilde{\mathbf{W}}^T\mathbf{B}. \tag{K.4}$$

In terms of the bases $\mathbf{V}$ and $\mathbf{W}$,

$$
\begin{aligned}
\tilde{\mathbf{H}}_r(s) &= \mathbf{C}\mathbf{V}\mathbf{P}\left(s\mathbf{I}_k - \left(\mathbf{Q}^T\mathbf{W}^T\mathbf{V}\mathbf{P}\right)^{-1}\mathbf{Q}^T\mathbf{W}^T\mathbf{A}\mathbf{V}\mathbf{P}\right)^{-1}\left(\mathbf{Q}^T\mathbf{W}^T\mathbf{V}\mathbf{P}\right)^{-1}\mathbf{Q}^T\mathbf{W}^T\mathbf{B} \\
&= \mathbf{C}\mathbf{V}\mathbf{P}\left(s\mathbf{I}_k - \left(\mathbf{W}^T\mathbf{V}\mathbf{P}\right)^{-1}\mathbf{Q}^{-T}\mathbf{Q}^T\mathbf{W}^T\mathbf{A}\mathbf{V}\mathbf{P}\right)^{-1}\left(\mathbf{W}^T\mathbf{V}\mathbf{P}\right)^{-1}\mathbf{Q}^{-T}\mathbf{Q}^T\mathbf{W}^T\mathbf{B} \\
&= \mathbf{C}\mathbf{V}\mathbf{P}\left(s\mathbf{P}^{-1}\mathbf{P} - \mathbf{P}^{-1}\left(\mathbf{W}^T\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{A}\mathbf{V}\mathbf{P}\right)^{-1}\mathbf{P}^{-1}\left(\mathbf{W}^T\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{B} \\
&= \mathbf{C}\mathbf{V}\mathbf{P}\mathbf{P}^{-1}\left(s\mathbf{I}_k - \left(\mathbf{W}^T\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{A}\mathbf{V}\right)^{-1}\mathbf{P}\mathbf{P}^{-1}\left(\mathbf{W}^T\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{B} \\
&= \mathbf{C}\mathbf{V}\left(s\mathbf{I}_k - \left(\mathbf{W}^T\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{A}\mathbf{V}\right)^{-1}\left(\mathbf{W}^T\mathbf{V}\right)^{-1}\mathbf{W}^T\mathbf{B} \\
&= \mathbf{H}_r(s).
\end{aligned}
\tag{K.5}
$$

The fact that the input-output behavior of a dynamical system is uniquely determined by its transfer function terminates the proof.

K.2   Proof of Proposition 3

Let $\mathbf{Q} \in \mathrm{O}(k)$ and $F_1(\mathbf{Q})$ denote

$$
\begin{aligned}
F_1(\mathbf{Q}) &= \alpha\|\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q} - \mathbf{A}_{rl_0}\|_F^2 + \beta\|\mathbf{Q}^T\mathbf{B}_{rl} - \mathbf{B}_{rl_0}\|_F^2 + \gamma\|\mathbf{C}_{rl}\mathbf{Q} - \mathbf{C}_{rl_0}\|_F^2 \\
&= \alpha\mathrm{tr}\Big([\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q} - \mathbf{A}_{rl_0}]^T[\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q} - \mathbf{A}_{rl_0}]\Big) + \beta\mathrm{tr}\Big([\mathbf{Q}^T\mathbf{B}_{rl} - \mathbf{B}_{rl_0}]^T[\mathbf{Q}^T\mathbf{B}_{rl} - \mathbf{B}_{rl_0}]\Big) \\
&\quad + \gamma\mathrm{tr}\Big([\mathbf{C}_{rl}\mathbf{Q} - \mathbf{C}_{rl_0}]^T[\mathbf{C}_{rl}\mathbf{Q} - \mathbf{C}_{rl_0}]\Big) \\
&= \alpha\left(\mathrm{tr}(\mathbf{A}_{rl}^T\mathbf{A}_{rl}) + \mathrm{tr}(\mathbf{A}_{rl_0}^T\mathbf{A}_{rl_0}) - 2\mathrm{tr}(\mathbf{A}_{rl_0}^T\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q})\right) + \beta\left(\mathrm{tr}(\mathbf{B}_{rl}^T\mathbf{B}_{rl}) + \mathrm{tr}(\mathbf{B}_{rl_0}^T\mathbf{B}_{rl_0})\right. \\
&\quad \left. - 2\mathrm{tr}(\mathbf{B}_{rl_0}^T\mathbf{Q}^T\mathbf{B}_{rl})\right) + \gamma\left(\mathrm{tr}(\mathbf{C}_{rl}^T\mathbf{C}_{rl}) + \mathrm{tr}(\mathbf{C}_{rl_0}^T\mathbf{C}_{rl_0}) - 2\mathrm{tr}(\mathbf{C}_{rl_0}^T\mathbf{C}_{rl}\mathbf{Q})\right).
\end{aligned}
$$

$$(\text{K.6})$$

Since several terms do not depend on $\mathbf{Q}$,

$$
\min_{\mathbf{Q}\in\mathrm{O}(k)} F_1(\mathbf{Q}) = \max_{\mathbf{Q}\in\mathrm{O}(k)} F_2(\mathbf{Q}), \tag{K.7}
$$

with

$$
\begin{aligned}
F_2(\mathbf{Q}) &= \alpha\mathrm{tr}(\mathbf{A}_{rl_0}^T\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}) + \beta\mathrm{tr}(\mathbf{B}_{rl_0}^T\mathbf{Q}^T\mathbf{B}_{rl}) + \gamma\mathrm{tr}(\mathbf{C}_{rl_0}^T\mathbf{C}_{rl}\mathbf{Q}) \\
&= \alpha\mathrm{tr}(\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T) + \beta\mathrm{tr}(\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T\mathbf{Q}^T) + \gamma\mathrm{tr}(\mathbf{C}_{rl_0}^T\mathbf{C}_{rl}\mathbf{Q}) \\
&= \alpha\mathrm{tr}(\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T) + \beta\mathrm{tr}(\mathbf{Q}\mathbf{B}_{rl_0}\mathbf{B}_{rl}^T) + \gamma\mathrm{tr}(\mathbf{C}_{rl_0}^T\mathbf{C}_{rl}\mathbf{Q}) \\
&= \alpha\mathrm{tr}(\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T) + \beta\mathrm{tr}(\mathbf{B}_{rl_0}\mathbf{B}_{rl}^T\mathbf{Q}) + \gamma\mathrm{tr}(\mathbf{C}_{rl_0}^T\mathbf{C}_{rl}\mathbf{Q}) \\
&= \alpha\mathrm{tr}(\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T) + \mathrm{tr}\Big([\beta\mathbf{B}_{rl_0}\mathbf{B}_{rl}^T + \gamma\mathbf{C}_{rl_0}^T\mathbf{C}_{rl}]\mathbf{Q}\Big) \\
&= \alpha\langle\mathbf{Q}^T\mathbf{A}_{rl}^T\mathbf{Q}, \mathbf{A}_{rl_0}^T\rangle + \langle\beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}, \mathbf{Q}\rangle \\
&= \alpha\langle\mathbf{Q}^T\mathbf{A}_{rl}\mathbf{Q}, \mathbf{A}_{rl_0}\rangle + \langle\beta\mathbf{B}_{rl}\mathbf{B}_{rl_0}^T + \gamma\mathbf{C}_{rl}^T\mathbf{C}_{rl_0}, \mathbf{Q}\rangle
\end{aligned}
$$

$$(\text{K.8})$$

K.3 Proof of Theorem 1

The goal of this section is to characterize the critical points of the function

$$
\begin{aligned}
F_2 : \ &O(k) \longrightarrow \mathbb{R} \\
&\mathbf{Q} \longmapsto \alpha \mathrm{tr}(\mathbf{Q}^T \mathbf{A}_{rl} \mathbf{Q} \mathbf{A}_{rl_0}^T) + \mathrm{tr}\Big([\beta \mathbf{B}_{rl_0} \mathbf{B}_{rl}^T + \gamma \mathbf{C}_{rl_0}^T \mathbf{C}_{rl}] \mathbf{Q}\Big).
\end{aligned}
\tag{K.9}
$$

The following analysis is an extension of the one proposed in [113] to the present form of the objective function presenting the additional second term.

K.3.1 Lagrangian

The objective function can be written as

$$
F_2(\mathbf{Q}) = \alpha \langle \mathbf{Q}^T \mathbf{A}_{rl}^T \mathbf{Q}, \mathbf{A}_{rl_0}^T \rangle + \langle \mathbf{G}, \mathbf{Q} \rangle = \alpha \langle \mathbf{Q}^T \mathbf{A}_{rl} \mathbf{Q}, \mathbf{A}_{rl_0} \rangle + \langle \mathbf{G}, \mathbf{Q} \rangle,
\tag{K.10}
$$

$\mathbf{G}$ being here defined as $\mathbf{G} = \beta \mathbf{B}_{rl} \mathbf{B}_{rl_0}^T + \gamma \mathbf{C}_{rl}^T \mathbf{C}_{rl_0}$.

The Lagrangian is then

$$
\begin{aligned}
\mathcal{L}(\mathbf{Q}, \mathbf{S}) &= F_2(\mathbf{Q}) + \left\langle \frac{1}{2} \mathbf{S}, \mathbf{I}_k - \mathbf{Q}^T \mathbf{Q} \right\rangle \\
&= \alpha \langle \mathbf{Q}^T \mathbf{A}_{rl} \mathbf{Q}, \mathbf{A}_{rl_0} \rangle + \langle \mathbf{G}, \mathbf{Q} \rangle + \left\langle \frac{1}{2} \mathbf{S}, \mathbf{I}_k - \mathbf{Q}^T \mathbf{Q} \right\rangle,
\end{aligned}
\tag{K.11}
$$

where $\frac{1}{2}\mathbf{S}$ is a symmetric matrix of Lagrangian multipliers.

K.3.2 First-Order Optimality Condition

Using the following identities [113]

$$\nabla_{\mathbf{Q}}\langle\mathbf{M}, \mathbf{Q}\rangle = \mathbf{M} \tag{K.12}$$

$$\nabla_{\mathbf{Q}}\langle\mathbf{M}, \mathbf{Q}^T\mathbf{Q}\rangle = \mathbf{Q}(\mathbf{M} + \mathbf{M}^T) \tag{K.13}$$

$$\nabla_{\mathbf{Q}}\langle\mathbf{Q}^T\mathbf{M}\mathbf{Q}, \mathbf{N}\rangle = \mathbf{M}\mathbf{Q}\mathbf{N}^T + \mathbf{M}^T\mathbf{Q}\mathbf{N}, \tag{K.14}$$

the gradient of the Lagrangian with respect to $\mathbf{Q}$ is obtained as

$$\nabla_{\mathbf{Q}}\mathcal{L}(\mathbf{Q}, \mathbf{S}) = \alpha(\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}\mathbf{A}_{rl_0}) + \mathbf{G} - \mathbf{Q}\mathbf{S} \tag{K.15}$$

which leads to the first-order optimality condition

$$\mathbf{Q}\mathbf{S} = \alpha(\mathbf{A}_{rl}\mathbf{Q}\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}\mathbf{A}_{rl_0}) + \mathbf{G}, \tag{K.16}$$

with the constraint $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k$ and the property that $\mathbf{S}$ is symmetric.

K.4   Proof of Theorem 2

The goal of this section is to prove that the set of fixed points of the proposed recursive algorithm is equal to the set of critical points of the objective function F_2.

Let $\mathbf{G} = \beta\mathbf{B}_{rl_0}\mathbf{B}_{rl}^T + \gamma\mathbf{C}_{rl_0}^T\mathbf{C}_{rl}$ and $\hat{\mathbf{Q}}$ be a fixed point of the recursive method defined in Algorithm 2. Then $\hat{\mathbf{Q}}$ satisfies $\hat{\mathbf{Q}} = \hat{\mathbf{U}}\hat{\mathbf{V}}^T$ where

$$\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^T = \alpha(\mathbf{A}_{rl}\hat{\mathbf{Q}}\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\hat{\mathbf{Q}}\mathbf{A}_{rl_0}) + s\hat{\mathbf{Q}}\mathbf{G} \tag{K.17}$$

is a singular value decomposition. Since $\hat{\mathbf{V}}$ is an orthogonal matrix,

$$\hat{\mathbf{U}}\hat{\mathbf{V}}^T\hat{\mathbf{V}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}^T = \alpha(\mathbf{A}_{rl}\hat{\mathbf{Q}}\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\hat{\mathbf{Q}}\mathbf{A}_{rl_0}) + s\hat{\mathbf{Q}} + \mathbf{G}, \qquad (\text{K.18})$$

that is

$$\hat{\mathbf{Q}}\mathbf{S} = \alpha(\mathbf{A}_{rl}\hat{\mathbf{Q}}\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\hat{\mathbf{Q}}\mathbf{A}_{rl_0}) + \mathbf{G}, \qquad (\text{K.19})$$

where $\mathbf{S} = \hat{\mathbf{V}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}^T - s\mathbf{I}_k$ is a symmetric matrix. $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ being orthogonal, $\hat{\mathbf{Q}}$ is orthogonal as well and therefore meets the requirements of Theorem 2. The set of fixed points of Algorithm 2 is included in the set of critical point of F_2.

Conversely, let $\mathbf{Q}^\star$ be a critical point of F_2. Then, there exists a symmetric matrix $\mathbf{S}$ such that Eq. (7.47) holds with $\mathbf{Q}^\star$ orthogonal. Since, $\mathbf{S}$ is real and symmetric, it is diagonalizable as

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T, \qquad (\text{K.20})$$

the eigenvalues in $\boldsymbol{\Lambda}$ being real, ordered in a decreasing fashion, and $\mathbf{U}$ orthogonal. Then,

$$\mathbf{Q}^\star\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T = \alpha(\mathbf{A}_{rl}\mathbf{Q}^\star\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}^\star\mathbf{A}_{rl_0}) + \mathbf{G} \qquad (\text{K.21})$$

and

$$\mathbf{Q}^\star\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T + s\mathbf{Q}^\star = \alpha(\mathbf{A}_{rl}\mathbf{Q}^\star\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}^\star\mathbf{A}_{rl_0}) + s\mathbf{Q}^\star + \mathbf{G}, \qquad (\text{K.22})$$

which can also be written as

$$(\mathbf{Q}^\star\mathbf{U})(\boldsymbol{\Lambda} + s\mathbf{I})\mathbf{U}^T = \alpha(\mathbf{A}_{rl}\mathbf{Q}^\star\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}^\star\mathbf{A}_{rl_0}) + s\mathbf{Q}^\star + \mathbf{G}. \qquad (\text{K.23})$$

One needs now to show, in order to conclude the proof, that $(\mathbf{Q}^\star\mathbf{U})(\boldsymbol{\Lambda} + s\mathbf{I})\mathbf{U}^T$ is a singular value decomposition. $\mathbf{Q}^\star\mathbf{U}$ and $\mathbf{U}$ being orthogonal matrices, and $\boldsymbol{\Lambda} + s\mathbf{I}$

being a diagonal matrix with diagonal entries ordered decreasingly, it is sufficient to show that $\mathbf{\Lambda} + s\mathbf{I}$ has positive entries.

From Eq. (K.21),

$$\|\mathbf{\Lambda}\|_2 = \|\mathbf{Q}^\star \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T\|_2 = \|\alpha(\mathbf{A}_{rl}\mathbf{Q}^\star\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}^\star\mathbf{A}_{rl_0}) + \mathbf{G}\|_2, \tag{K.24}$$

and

$$
\begin{aligned}
\|\mathbf{\Lambda}\|_2 &\leq \alpha\|\mathbf{A}_{rl}\mathbf{Q}^\star\mathbf{A}_{rl_0}^T + \mathbf{A}_{rl}^T\mathbf{Q}^\star\mathbf{A}_{rl_0}\|_2 + \|\mathbf{G}\|_2 \\
&\leq \alpha(\|\mathbf{A}_{rl}\|_2\|\mathbf{Q}^\star\|_2\|\mathbf{A}_{rl_0}^T\|_2 + \|\mathbf{A}_{rl}^T\|_2\|\mathbf{Q}^\star\|_2\|\mathbf{A}_{rl_0}\|_2) + \|\mathbf{G}\|_2 \\
&\leq 2\alpha\|\mathbf{A}_{rl}\|_2\|\mathbf{A}_{rl_0}\|_2 + \|\mathbf{G}\|_2 \\
&\leq s_{\min}
\end{aligned}
\tag{K.25}
$$

by definition of $s_{\min}$ in Eq. (7.43). Denoting by $\lambda_i \; i = 1, \cdots, k$ the diagonal entries in $\mathbf{\Lambda}$, this implies that

$$-s_{\min} \leq \lambda_i \leq s_{min}, \; i = 1, \cdots, k, \tag{K.26}$$

and, since $s > s_{\min}$,

$$\lambda_i + s > 0, \; i = 1, \cdots, k, \tag{K.27}$$

concluding the proof. The set of critical points of F_2 is included in the set of fixed points of Algorithm 2 and hence, these two sets are equal.

K.5 Interpolation of Stiffness Matrices in the Tangent Space to the Manifold of Symmetric Positive Definite Matrices

The interpolation algorithm on the tangent space to the manifold of SPD matrices is followed step-by-step.

1. Step B.0. A point $l'_0 \in 1, \cdots, N_R$ is chosen as reference. Note that l'_0 may differ from l_0.

2. Step B.1. Each point $\tilde{\mathbf{K}}_{rl}$ that is sufficiently close to $\tilde{\mathbf{K}}_{rl'_0}$ is mapped to a matrix $\boldsymbol{\Gamma}_l$ of $T_{\tilde{\mathbf{K}}_{rl'_0}} \mathrm{SPD}(k)$ using the logarithm map $\mathrm{Log}_{\tilde{\mathbf{K}}_{rl'_0}}$. This can be written as

$$
\begin{aligned}
\boldsymbol{\Gamma}_l &= \mathrm{Log}_{\tilde{\mathbf{K}}_{rl'_0}}(\tilde{\mathbf{K}}_{rl}) \\
&= \mathrm{logm}\left(\tilde{\mathbf{K}}_{rl'_0}^{-1/2}\tilde{\mathbf{K}}_{rl}\tilde{\mathbf{K}}_{rl'_0}^{-1/2}\right) \\
&= \mathrm{logm}\left(\mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{-1/2}\mathbf{U}_{l_0}^T\mathbf{U}_{l_0}\boldsymbol{\Lambda}_l\mathbf{U}_{l_0}^T\mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{-1/2}\mathbf{U}_{l_0}^T\right) \\
&= \mathrm{logm}\left(\mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{-1/2}\boldsymbol{\Lambda}_l\boldsymbol{\Lambda}_{l'_0}^{-1/2}\mathbf{U}_{l_0}^T\right) \\
&= \mathbf{U}_{l_0}\mathrm{logm}\left(\boldsymbol{\Lambda}_{l'_0}^{-1}\boldsymbol{\Lambda}_l\right)\mathbf{U}_{l_0}^T,
\end{aligned}
$$

   log denoting here the matrix logarithm.

3. Step B.2. Each entry of the matrix $\boldsymbol{\Gamma}_{N_R+1}$ associated with the target operating point $\boldsymbol{\mu}_{N_R+1}$ is computed by interpolating the corresponding entries of the matrices $\{\boldsymbol{\Gamma}_l\} \in \mathbb{R}^{k \times k}$ associated with the operating points $\{\boldsymbol{\mu}_l\}$.

Many interpolation schemes can be put under the following generic form:

$$\mathbf{\Gamma}_{N_R+1} = \mathbf{\Gamma}(\boldsymbol{\mu}_{N_R+1}) = \sum_{l=1}^{N_R} \omega_l(\boldsymbol{\mu}_{N_R+1})\mathbf{\Gamma}(\boldsymbol{\mu}_l) = \sum_{l=1}^{N_R} \omega_l(\boldsymbol{\mu}_{N_R+1})\mathbf{\Gamma}_l, \qquad (\text{K}.28)$$

where the weights $\omega_l(\boldsymbol{\mu})$ satisfy

$$\sum_{l=1}^{N_R} \omega_l(\boldsymbol{\mu}) = 1, \ \forall \boldsymbol{\mu}, \qquad (\text{K}.29)$$

in the present case, denoting $\omega_l = \omega_l(\boldsymbol{\mu}_{N_R+1})$ to alleviate the notations, one obtains for $\mathbf{\Gamma}_{N_R+1}$,

$$
\begin{aligned}
\mathbf{\Gamma}_{N_R+1} &= \sum_{l=1}^{N_R} \omega_l \mathbf{\Gamma}_l \\
&= \sum_{l=1}^{N_R} \omega_l \mathbf{U}_{l_0} \texttt{logm}\left(\mathbf{\Lambda}_{l_0'}^{-1}\mathbf{\Lambda}_l\right)\mathbf{U}_{l_0}^T \\
&= \mathbf{U}_{l_0} \sum_{l=1}^{N_R} \texttt{logm}\left(\mathbf{\Lambda}_{l_0'}^{-1}\mathbf{\Lambda}_l\right)^{\omega_l}\mathbf{U}_{l_0}^T \\
&= \mathbf{U}_{l_0} \texttt{logm}\left(\mathbf{\Lambda}_{l_0'}^{-\sum_{i=1}^{N_R}\omega_i}\prod_{l=1}^{N_R}\mathbf{\Lambda}_l^{\omega_l}\right)\mathbf{U}_{l_0}^T \\
&= \mathbf{U}_{l_0} \texttt{logm}\left(\mathbf{\Lambda}_{l_0'}^{-1}\prod_{l=1}^{N_R}\mathbf{\Lambda}_l^{\omega_l}\right)\mathbf{U}_{l_0}^T
\end{aligned}
$$

4. Step B.3. The matrix $\mathbf{\Gamma}_{N_R+1} \in T_{\tilde{\mathbf{K}}_{rl_0'}}\text{SPD}(k)$ is mapped to a matrix $\mathbf{K}_{rN_R+1}$

belonging to the manifold SPD(lk) using the exponential map $\mathrm{Exp}_{\mathbf{K}_{rl'_0}}$.

$$
\begin{aligned}
\mathbf{K}_{rN_R+1} &= \mathrm{Exp}_{\tilde{\mathbf{K}}_{rl'_0}}(\boldsymbol{\Gamma}_{N_R+1}) \\
&= \tilde{\mathbf{K}}_{rl'_0}^{1/2}\,\mathrm{expm}(\boldsymbol{\Gamma}_{N_R+1})\tilde{\mathbf{K}}_{rl'_0}^{1/2} \\
&= \mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{1/2}\mathbf{U}_{l_0}^T\,\mathrm{expm}\!\left(\mathbf{U}_{l_0}\,\mathrm{logm}\left(\boldsymbol{\Lambda}_{l'_0}^{-1}\Pi_{l=1}^{N_R}\boldsymbol{\Lambda}_l^{\omega_l}\right)\mathbf{U}_{l_0}^T\right)\mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{1/2}\mathbf{U}_{l_0}^T \\
&= \mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{1/2}\mathbf{U}_{l_0}^T\mathbf{U}_{l_0}\,\mathrm{expm}\!\left(\mathrm{logm}\left(\boldsymbol{\Lambda}_{l'_0}^{-1}\Pi_{l=1}^{N_R}\boldsymbol{\Lambda}_l^{\omega_l}\right)\right)\mathbf{U}_{l_0}^T\mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{1/2}\mathbf{U}_{l_0}^T \\
&= \mathbf{U}_{l_0}\boldsymbol{\Lambda}_{l'_0}^{1/2}\boldsymbol{\Lambda}_{l'_0}^{-1}\prod_{l=1}^{N_R}\boldsymbol{\Lambda}_l^{\omega_l}\boldsymbol{\Lambda}_{l'_0}^{1/2}\mathbf{U}_{l_0}^T \\
&= \mathbf{U}_{l_0}\prod_{l=1}^{N_R}\boldsymbol{\Lambda}_l^{\omega_l}\mathbf{U}_{l_0}^T
\end{aligned}
$$

Bibliography

[1] Ashley, H. and Zartarian, G., "Piston Theory - A New Aerodynamic Tool for the Aeroelastician," *Journal of Aerospace Sciences*, Vol. 23, 1956, pp. 1109–1118.

[2] Geuzaine, P., Brown, G., Harris, C. and Farhat, C., "Aeroelastic Dynamic Analysis of a Full F-16 Configuration for Various Flight Conditions," *AIAA Journal*, Vol. 41, 2003, pp. 363–371.

[3] Woon, S. K. and Marshall, S. A., "Design of Multivariable Control Systems Using Reduced-Order Models," *Electronics Letters*, Vol. 11, No. 15, 1975, pp. 341–342.

[4] LeGresley, P. A. and Alonso, J., "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition," *AIAA Paper 00-25450*, 2000.

[5] Lall, S., Mardsen, J. E. and Glavaski, S., "A Subspace Approach to Balanced Truncation for Model Reduction of Nonlinear Control Systems," *International Journal of Robust and Nonlinear Control*, Vol. 12, 2002, pp. 519–535.

[6] Michopoulos, J., Tsompanopoulou, P., Houstis, E., Rice, J., Farhat, C., Lesoinne, M. and Lechenault, F., "DDEMA: a Data-Driven Environment for Multiphysics Applications," *Lecture Notes in Computer Science, 2660, Part IV, ed. P. M. A.*

Sloot, D. Abramson, A. Bogdanov, J. J. Dongarra, A. Zomaya and Y. Gorbache, Springer-Verlag, Haidelberg, 2003, pp. 309–318.

[7] Cortial, J., Farhat, C., Rajashekhar, M. and Guibas, L., "Compressed Sensing and Time-Parallel Reduced-Order Modeling for Structural Health Monitoring using a DDDAS," *Lecture Notes in Computer Science, ed. Y. Shi et al., Springer-Verlag*, Vol. 11, No. 4487, 2007, pp. 1171–1179.

[8] Romanowski, M. C., "Reduced Order Unsteady Aerodynamic and Aeroelastic Models Using Karhunen-Loeve Eigenmodes," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1996, AIAA-1996-3981, pp. 7–13.

[9] Kim, T., "Frequency-Domain Karhunen-Loeve Method and Its Application to Linear Dynamic Systems," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 2117–2123.

[10] Hall, K. C., Thomas, J. P. and Dowell, E. H., "Reduced-Order Modelling of Unsteady Small-Disturbance Flows Using a Frequency Domain Proper Orthogonal Decomposition Technique," *AIAA Paper 99-16520*, 1999.

[11] Hall, K. C., Thomas, J. P. and Dowell, E. H., "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows," *AIAA Journal*, Vol. 38, No. 10, October 2000, pp. 1853–1862.

[12] Willcox, K. and Peraire, J., "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 40, No. 11, November 2002, pp. 2323–2330.

[13] Kim, T. and Bussoletti, J. E., "An Optimal Reduced-Order Aeroelastic Modeling Based on a Response-Based Modal Analysis of Unsteady CFD Models," *AIAA Paper 2001-1525*, 2001.

[14] Kim, T., Hong M., Bhatia, K. B. and SenGupta G., "Aeroelastic Model Reduction for Affordable Computational Fluid Dynamics-Based Flutter Analysis," *AIAA Journal*, Vol. 43, No. 12, December 2005, pp. 2487-2495.

[15] Epureanu, B. I., "A Parametric Analysis of Reduced Order Models of Viscous Flows in Turbomachinery," *Journal of Fluids and Structures*, Vol. 17, 2003, pp. 971–982.

[16] Thomas, J. P., Dowell, E. H. and Hall, K. C., "Three-Dimensional Transonic Aeroelasticity Using Proper Orthogonal Decomposition-Based Reduced Order Models," *Journal of Aircraft*, Vol. 40, No. 3, 2003, pp. 544–551.

[17] Lieu, T. and Lesoinne, M., "Parameter Adaptation of Reduced Order Models for Three-Dimensional Flutter Analysis," *AIAA Paper 2004-0888*, 2004.

[18] Cizmas, P. G. A. and Palacios, A., "Proper Orthogonal Decomposition of Turbine Rotor-Stator Interaction," *Journal of Propulsion and Power*, Vol. 19, No. 2, March 2003, pp. 268–281.

[19] Silva, W., "Identification of Nonlinear Aeroelastic Systems Based on the Volterra Theory: Progress and Opportunities," *Nonlinear Dynamics*, Vol. 39, No. 1-2, January 2005, pp. 25–62.

[20] Silva, W. A. and Bartels, R. E., "Development of Reduced-Order Models for Aeroelastics Analysis and Flutter Prediction Using the CFL3D v6.0 Code," *Journal of Fluids and Structures*, Vol. 19, No. 6, July 2005, pp. 729–745.

[21] Spitler, J. E., "Initial Studies of Low-Order Turbulence Modeling of the Wind Turbine In-Flow Environment," *AIAA Paper 2004-1004*, 2004.

[22] Karhunen, K., "Zur Spektraltheorie Stochastischer Prozesse," *Ann. Acad. Sci. Fennicae*, Vol. 34, 1946.

[23] Loeve, M., "Fonctions Aléatoires de Second Ordre," *Comptes Rendus Acad. Sci. Paris*, Vol. 220, 1945.

[24] Hong, M., Bhatia, K., SenGupta, G., Kim, T., Kuruvila, G., Silva, W., Bartels, R. and Biedron, R., "Simulations of a Twin-Engine Transport Flutter Model in the Transonic Dynamics Tunnel," *International Forum On Aeroelastic and Structural Dynamics Paper IFAS D 2003-US-44*, 2003.

[25] Lieu, T., Farhat, C. and Lesoinne, M., "POD-based Aeroelastic Analysis of a Complete F-16 Configuration: ROM Adaptation and Demonstration," *AIAA Paper 2005-2295*, 2005.

[26] Lieu, T. and Farhat, C., "Adaptation of POD-Based Aeroelastic ROMs for Varying Mach Number and Angle of Attack: Application to a Complete F-16 Configuration," *AIAA Paper 2005-7666*, 2005.

[27] Lieu, T., Farhat, C. and Lesoinne, M., "Reduced-Order Fluid/Structure Modeling of a Complete Aircraft Configuration," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, 2006, pp. 5730–5742.

[28] Lieu, T. and Farhat, C., "Adaptation of Aeroelastic Reduced-Order Models and Application to an F-16 Configuration," *AIAA Journal*, Vol. 45, 2007, pp. 1244–1257.

[29] Amsallem, D., Farhat, C. and Lieu, T., "High-Order Interpolation of Reduced-Order Models for Near Real-Time Aeroelastic Predictions," *Paper IF-081, International Forum on Aeroelasticity and Structural Dynamics, Stockholm, Sweden, June 18-20*, 2007.

[30] Amsallem, D., Farhat, C. and Lieu, T., "Aeroelastic Analysis of F-16 and F-18/A Configurations Using Adapted CFD-Based Reduced-Order Models," *AIAA Paper 2007-2364, 48th Structures, Structural Dynamics, and Materials Conference, Honolulu, Hawaii, April 23-26*, 2007.

[31] Amsallem, D. and Farhat, C., "Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity," *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1803–1813.

[32] Dowell, E. H. and Hall, K., "Modeling of Fluid-Structure Interaction," *Annual Review Fluid Mechanics*, Vol. 33, 2001, pp. 445–490.

[33] Rathinam, M. and Petzold, L. R., "A New Look at Proper Orthogonal Decompositionm" *SIAM Journal of Numerical Analysis*, Vol. 41, No. 5, 2003, pp. 1893–1925.

[34] Gustafsson, B., "High Order Difference Methods for Time Dependent PDE," *Springer*, 2008.

[35] Golub, G. H. and Van Loan, C. F., "Matrix Computations, 3rd Ed.," *The Johns Hopkins Univ. Press*, Baltimore, MD, 1996.

[36] Antoulas, A. C., "Approximation of Large-scale Dynamical Systems," *Advances in Design and Control, SIAM*, 2005.

[37] Bui-Thanh, T., and Willcox, K., "Model Reduction for Large-Scale CFD Applications Using the Balanced Proper Orthogonal Decomposition," *17th AIAA Computational Dynamics Conference*, 2005.

[38] Moore, B. C., "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Transactions on Automatic Control*. Vol. 26, No. 1, 1981, pp. 17–32.

[39] Enns, D., "Model Reduction for Control Model Design," *Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University*, 1984.

[40] Holmes, P., Lumley, J. and Berkooz, G.,"Turbulence, Coherent Structures, Dynamical Systems and Symmetry," *Cambridge University Press*, 1996.

[41] Kirby, M., Boris, J. P. and Sirovich, L., "A Proper Orthogonal Decomposition of a Simulated Supersonic Shear Layer," *International Journal for Numerical Methods in Fluids*, Vol. 10, 1990, pp. 411–428.

[42] Rowley, C. W., "Model Reduction for Fluids Using Balanced Proper Orthogonal Decomposition," *International Journal on Bifurcation and Chaos*, Vol. 15, No. 3, 2005, pp. 997–1013.

[43] Antoulas, A. C., Sorensen, D. C. and Gugercin, S., "A Survey of Model Reduction Methods for Large-Scale Systems," *Rice University*, 2006.

[44] Serban, R., Homescu, C. and Petzold, L. R., "The Effect of Problem Perturbations on Nonlinear Dynamical Systems and Their Reduced-Order Models," *SIAM Journal on Scientific Computing*, Vol. 29, No. 6, 2007, pp. 2621–2643.

[45] Hay, A., Borggaard, J. and Pelletier, D., "Reduced-Order Models for Parameter Dependent Geometries Based on Shape Sensitivity Analysis of the POD," *AIAA Paper 2008-5962*, 2008.

[46] Hay, A., Borggaard, J., Akhtar, I. and Pelletier, D., "Reduced-Order Models for Parameter Dependent Geometries Based on Shape Sensitivity Analysis," *Journal of Computational Physics*, Vol. 229, 2010, pp. 1327–1352.

[47] Taylor, J., "Dynamics of Large Scale Structures in Turbulent Shear Layers," *Ph.D. Thesis, Clarkson University*, 2001.

[48] Taylor, J. and Glauser, M., "Towards Practical Flow Sensing and Control via POD and LSE Based Low-Dimensional Tools," *2002 ASME Fluids Engineering Division Summer Meeting*, 2002, pp. 1–9.

[49] Schmidt, R. and Glauser, M., "Improvements in Low Dimensional Tools for Flow-Structure Interaction Problems: Using Global POD," *AIAA Paper 2004-0889*, 2004.

[50] Keane, A. J. and Nair, P. B, "Computational Approaches for Aerospace Design," *Wiley*, 2005.

[51] Ito, K. and Ravindran, S. S., "A Reduced-Order Method for Simulation and Control of Fluid Flows," *Journal of Computational Physics*, Vol. 143, 1998, pp. 403–425.

[52] Maute, K., Weickum, G., and Eldred M., "A Reduced-Order Stochastic Finite Element Approach for Design Optimization Under Uncertainty," *Structural Safety*, Vol. 31, 2009, pp. 450–459.

[53] Weickum, G., Eldred, M. S. and Maute K., "A Multi-Point Reduced-Order Modeling Approach of Transient Structural Dynamics with Application to Robust Design Optimization," *Structural Multidisciplinary Optimization*, Vol. 38, 2009, pp. 599–611.

[54] Carlberg, K. and Farhat, C., "A Compact Proper Orthogonal Decomposition Basis for Optimization-Oriented Reduced-Order Model," *AIAA Paper 2008-5964, 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, Canada*, 2008.

[55] Peterson, J., "The Reduced-Basis Method for Incompressible Viscous Flow Calculations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, No. 4, 1989, pp. 777–786.

[56] Allen, M., Weickum, G. and Maute K., "Application of Reduced-Order Models for the Stochastic Design Optimization of Dynamic Systems," *AIAA Paper 2004-4614*, 2004.

[57] Weickum, G., Eldred, M. S. and Maute, K., "Multi-Point Extended Reduced Order Modeling For Design Optimization and Uncertainty Analysis," *AIAA Paper 2006-2145*, 2006.

[58] Degroote, J., Vierendeels, J. and Willcox, K., "Interpolation Among Reduced-Order Matrices to Obtain Parameterized Models for Design, Optimization and Probabilistic Analysis," *International Journal for Numerical Methods in Fluids*, Vol. 63, 2010, pp. 207–230.

[59] Bechtold, T., Hohlfeld, D., Rudnyi, E. B. and Günther, M., "Efficient Extraction of Thin-Film Thermal Parameters From Numerical Models Via Parametric Model Order Reduction," *Journal of Micromechanics and Microengineering*, Vol. 20, 2010, 045030.

[60] Bui-Thanh, T., Willcox, K. and Ghattas, O., "Parametric Reduced-Order Models for Probabilistic Analysis of Unsteady Aerodynamic Applications," *AIAA Journal*, Vol. 46, No. 10, 2008, pp. 2520–2529.

[61] Haasdonk, B., Ohlberger, M. and Rozza, G., "A Reduced Basis Method for Evolution Schemes With Parameter-Dependent Explicit Operators," *Electronic Transactions on Numerical Analysis*, Vol. 32, 2008, pp. 145–161.

[62] Haasdonk, B. and Ohlberger, M., "Reduced Basis Method for Finite Volume Approximations of Parameterized Linear Evolution Equations," *Mathematical Modelling and Numerical Analysis*, Vol. 42, 2008, 277–302.

[63] Audouze, C., De Vuyst F. and Nair, P.B., "Reduced-Order Modeling of Parameterized PDEs Using Time-Space-Parameter Principal Component Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 80, 2009, pp. 1025–1057.

[64] Gu, C. and Roychowdhury, J., "Model Reduction via Projection onto Nonlinear Manifolds, with Applications to Analog Circuits and Biochemical Systems," *International Conference on Computer Aided Design*, 2008.

[65] Chern, S. S., Chen, W. H. and Lam, K. S., "Lectures on Differential Geometry," *World Scientific*, 2000.

[66] Absil, P.-A., Mahony, R. and Sepulchre, R., "Optimization Algorithms on Matrix Manifolds," *Princeton*, 2008.

[67] Perdigao Do Carmo, M., "Riemannian Geometry," *Birkhauser*, 1992.

[68] Helgason, S., "Differential Geometry, Lie Groups and Symmetric Spaces," *American Mathematical Society, Providence*, 2001.

[69] Boothby, W. M., "An Introduction to Differentiable Manifolds and Riemannian Geometry," *Academic Press*, 2003.

[70] Absil, P.-A., Mahony, R. and Sepulchre, R., "Riemannian Geometry of Grassmann Manifolds with a View on Algorithmic Computation," *Acta Applicandae Mathematicae*, Vol. 80, No. 2, 2004, pp. 199–220.

[71] Nocedal, J. and Wright, S., "Numerical Optimization," *Springer Series in Operations Research and Financial Engineering*, 2006.

[72] Rahman, I., Drori, I., Stodden, V., Donoho, D. and Schröder, P., "Multiscale Representations for Manifold-Valued Data," *Multiscale Model. Simul.*, Vol. 4, 2005, pp. 1201–1232.

[73] Hüper, K. and Silva Leite F., "On the Geometry of Rolling and Interpolation Curves on $\mathcal{S}^n$, $\mathcal{SO}_n$ and Graßmann Manifolds," *Journal of Dynamical and Control Systems*, Vol. 13, No. 4, 2007, pp. 467-502.

[74] Pennec, X., Fillard., P. and Ayache, N., "A Riemannian Framework for Tensor Computation," *International Journal of Computer Vision*, Vol. 66, No. 1, 2006, pp. 41–66.

[75] Bendsøe, M. P. and Kikuchi, N., "Generating Optimal Topologies in Structural Design Using A Homogeneization Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, 1988, pp. 197–224.

[76] Späth, H., "Two Dimensional Spline Interpolation Algorithms," *K Peters, Ltd.*, 1995.

[77] Amsallem, D., Cortial, J., Carlberg, K. and Farhat, C., "A Method for Interpolating on Manifolds Structural Dynamics Reduced-Order Models," *International Journal for Numerical Methods in Engineering*, Vol. 80, 2009, pp. 1241–1258.

[78] Edelman, E., Arias, T. A. and Smith, S. T., "The Geometry of Algorithms with Orthogonality Constraints," *SIAM J. Matrix Anal. Appl.*, Vol. 20, No. 2, 1999, pp. 303–353.

[79] Begelfor, E. and Werman, M., "Affine Invariance Revisited," *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, 2006, pp. 2087 – 2094.

[80] Björck, Å. and Golub, G. H., "Numerical Methods for Computing Angles Between Linear Subspaces," *Mathematics of Computation*, Vol. 27, No. 123, 1973, pp. 579–594.

[81] Beckte, T. and Trefethen, L. N., "Reviving the Method of Particular Solutions," *SIAM Review*, Vol. 47, No. 3, 2005, pp. 469–491.

[82] Dhillon, I. S. and Modha, D. S., "Concept Decompositions for Large Sparse Text Data Using Clustering," *Machine Learning*, Vol. 42, No. 1, 2001, pp. 143–176.

[83] Van Overschee, P. and De Moor, B., "Subspace Algorithms for the Stochastic Identication Problem," *Automatica*, Vol. 29, No. 3, 1993, pp. 649–660.

[84] Lieu, T., "Adaptation of Reduced Order Models for Applications in Aeroelasticity," *Ph.D. thesis, University of Colorado at Boulder*, 2004.

[85] Knyazev, A. V. and Argentati, M. E., "Principal Angles Between Subspaces in an A-Based Scalar Product: Algorithms and Perturbation Estimates," *SIAM J. Sci. Comput.*, Vol. 23, No. 6, 2002, pp. 2009–2041.

[86] Gugercin, S. and Antoulas, A. C., "A Survey of Model Reduction by Balanced Truncation and Some New Results," *International Journal of Control*, Vol. 77. No. 8, 2004, pp. 748–766.

[87] Fox, R. L. and Kapoor, M. P., "Rates of Change of Eigenvalues and Eigenvectors," *AIAA Journal*, Vol. 6, No. 12, 1968, pp. 2426–2429.

[88] Amsallem, D., Cortial, J. and Farhat, C., "On-Demand CFD-Based Aeroelastic Predictions Using a Database of Reduced-Order Bases and Models," *AIAA Paper 2009-800, 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, Orlando, Florida, January 5-8*, 2009.

[89] Amsallem, D., Cortial, J. and Farhat, C., "Fast CFD-based Aeroelastic Predictions Using a Database of Reduced-Order Models," *Paper IFASD-2009-006, International Forum on Aeroelasticity and Structural Dynamics, Seattle, Washington, June 22-25*, 2009.

[90] Romanowski, M. C. and Dowell, E. H., "Aeroelastic Analysis of an Airfoil Using Eigenmode Based Reduced-Order Unsteady Aerodynamics," *AIAA Paper 1995-1380*, 1995.

[91] Farhat, C., Lesoinne, M. and Maman, N., "Mixed Explicit/Implicit Time Integration of Coupled Aeroelastic Problems: Three-Field Formulation, Geometric Conservation and Distributed Solution," *International Journal of Numerical Methods in Engineering*, Vol. 21, 1995, pp. 807–835.

[92] Lesoinne, M., Sarkis, M., Hetmaniuk, U. and Farhat, C., "A Linearized Method for the Frequency Analysis of Three-Dimensional Fluid/Structure Interaction Problems in All Flow Regimes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, 2001, pp. 3121–3146.

[93] Hetmaniuk, U., "Linearized Aeroelasticity," *Center for Aerospace Structures, University of Colorado, Boulder*, 1998.

[94] Baker Jr., G. and Graves-Morris, P., "Padé Approximants," *Encyclopedia of Mathematics and its Applications*.

[95] Avery, P., Farhat, C. and Reese, G., "Fast Frequency Sweep Computations Using a Multi-Point Padé-Based Reconstruction Method and an Efficient Iterative Solver," *International Journal of Numerical Methods in Engineering*, Vol. 69, No. 13, 2006, pp. 2848–2875.

[96] Hu, P., Bodson M. and Brenner, M., "Towards Real-Time Simulation of Aeroservoelastic Dynamics for a Flight Vehicle from Subsonic to Hypersonic Regime," *AIAA Paper 2008-6375, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Honolulu, Hawaii, August 18-21*, 2008.

[97] Danowsky, B. P., Thompson, P. M., Farhat, C., Lieu, T., Harris, C. and Lechniak, J., "A Complete Aeroservoelastic Model: Incorporation of Oscillation-Reduction-Control into a High-Order CFD/FEM Fighter Aircraft Model," *AIAA*

Paper 2009-5708, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Chicago, Illinois, August 10-13, 2009.

[98] Peterson, L. D. and Crawley, E. F., "Improved Exponential Time Series Approximation of Unsteady Aerodynamic Operators," *Journal of Aircraft*, Vol. 25, 1988, pp. 121–127.

[99] Guyan, R. J., "Reduction of Stiffness and Mass Matrices," *AIAA Journal*, Vol. 3, No. 2, 1965, pp. 380.

[100] Flanigan, C. C., "Development of the IRS Component Dynamic Reduction Method for Substructure Analysis," *AIAA Paper 1991–1056*, 1991.

[101] Kerschen, G., Golinval, J. C., Vakakis, A. F. and Bergman, L. A., "The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: an Overview," *Nonlinear Dynamics*, Vol. 41, 2005, pp. 147–169.

[102] Amabili, M., Sarkar, A. and Paidoussis, M. P., "Reduced-Order Models for Nonlinear Vibrations of Cylindrical Shells via the Proper Orthogonal Decomposition Method," *Journal of Fluids and Structures*, Vol. 18, No. 2, 2003, pp. 227–250.

[103] Han, S. and Feeny, B. F., "Enhanced Proper Orthogonal Decomposition for the Modal Analysis of Homogeneous Structures," *Journal of Vibration and Control*, Vol. 8, No. 1, 2002, pp. 19–40.

[104] Shapiro, B., "Creating Compact Models of Complex Electronic Systems: An Overview and Suggested Use of Existing Model Reduction and Experimental System Identification Tools," *IEEE Transactions on Components and Packaging Technologies*, Vol. 26, No. 1, 2003, pp. 165–172.

[105] Yates, E. C., "Agard Standard Aeroelastic Configurations for Dynamic Response, Candidate Configuration I, -Wing 445.6.," *NASA TM-100462*, 1987.

[106] Ewins, D. J., "Modal Testing, Theory, Practice and Application (2nd edn)," *Research Study Press LTD*, 2000.

[107] Mitchell, T., "Machine Learning," *McGraw Hill,* 1997.

[108] Hastie, T., Tibshirani, R. and Friedman, J., "The Elements of Statistical Learning," *Springer*, 2009.

[109] Bungartz, H. J. and Griebel M., "Sparse Grids," *Acta Numerica,* Vol. 13, 2004, pp. 147–269.

[110] Lohmann, B. and Eid, R., "Efficient Order Reduction of Parametric and Nonlinear Models by Superposition of Locally Reduced Models," *Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2007 und 2008, Shaker Verlag, Aachen*, 2009.

[111] Eid, R., Lohmann, B. and Panzer, H., "Interpolation-Based Parametric Model Reduction Using Krylov Subspaces," *Autumn School on Future Developments in MOR, Terschelling, 22 September 2009*.

[112] Helmke, U. and Moore, J. B. , "Optimization and Dynamical Systems," *Springer-Verlag*, 1994.

[113] Fraikin, C., Nesterov, Y. and Van Dooren, P., "Optimizing the Coupling Between Two Isometric Projections of Matrices," *SIAM J. Matrix Anal. Appl.*, Vol. 30, 2008, pp. 324–345.

[114] Helmke, U., Hüper, K., Moore, J. B. and Schulte-Herbrüggen, Th., "Gradient Flows Computing the C-numerical Range with Applications in NMR Spectroscopy," *J. Glob. Optim.*, Vol. 23, 2002, pp. 283–308.

[115] Kim, T. S. and Kim, Y. Y., " Mac-Based Mode Tracking in Structural Topology Optimization," *Computers and Structures,* Vol. 74, 2000, pp. 375–383.

[116] Brüls, O., Duysinx, P. and Golinval, J. C., "The Global Modal Parameterization for Non-Linear Model-Order Reduction in Flexible Multibody Dynamics," *International Journal of Numerical Methods in Engineering*, Vol. 69, 2007, pp. 958–977.

[117] Morand, H. J.-P. and Ohayon, R., "Fluid-Structure Interaction," *Wiley*, 1995.

[118] Stephen, N. G., "On Veering of Eigenvalue Loci," *Journal of Vibrational Acoustics*, Vol. 131, 054501, 2009.

[119] Rendl, F. and Wolkowicz, H., "Application of Parametric Programming and Eigenvalue Maximization to the Quadratic Assignment Problem," *Math. Program.*, Vol. 53, 1992, pp. 63–78.

[120] Bunse-Gerstner, A., Byers, R., Mehrmann, V., and Nichols, N. K., "Numerical Computation of an Analytic Singular Value Decomposition of a Matrix Valued Function," *Numer. Math.*, Vol. 60, 1991, pp. 1–40.

[121] Chern, J.-L., and Dieci, L., "Smoothness and Periodicity of Some Matrix Decompositions", *SIAM J. Matrix Anal. Appl.*, Vol. 22, No. 3, 2000, pp. 772–792.

[122] Hardy, R.L., "Theory and Applications of the Multiquadric-Biharmonic Method", *Computers Math. Applic.*, Vol. 19, No. 8–9, 1990, pp. 163–208.

[123] Saad, Y., and Schutz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systens, "*SIAM J Sci. Stat. Comput.*, Vol. 7, No. 3, 1986, pp. 856–869.

[124] Tezaur, R., Macedo, A., and Farhat, C., "Iterative Solution of Large-Scale Acoustic Scattering Problems with Multiple Right-Hand Sides by a Domain Decomposition Method with Lagrange Multipliers," *International Journal of Numerical Methods in Engineering*, Vol. 51, 2001, pp. 1175–1193.

[125] Farhat, C., and Chern, P. S., "Tayloring Domain Decomposition Methods for Efficient Parallel Coarse-Grid Solution and for Systems with Many Right-Hand Sides," *Contemporary Mathematics*, Vol. 180, 1994, pp. 401–406.