SANDIA REPORT

SAND2014-4693
Unlimited Release
Printed May 2014

Reduced Order Modeling for Prediction and Control of Large-Scale Systems

Irina Kalashnikova, Srinivasan Arunajatesan, Matthew F. Barone, Bart G. van Bloemen Waanders, Jeffrey A. Fike

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-4693
Unlimited Release
Printed May 2014

Reduced Order Modeling for Prediction and Control of Large-Scale Systems

Irina Kalashnikova
Computational Mathematics
Sandia National Laboratories
P.O. Box 5800, MS 1320
Albuquerque, NM 87185-1320
ikalash@sandia.gov

Srinivasan Arunajatesan
Aerosciences Department
Sandia National Laboratories
P.O. Box 5800, MS 0825
Albuquerque, NM 87185–0825
sarunaj@sandia.gov

Matthew F. Barone
Aerosciences Department
Sandia National Laboratories
P.O. Box 5800, MS 0825
Albuquerque, NM 87185–0825
mbarone@sandia.gov

Bart G. van Bloemen Waanders
Uncertainty Quantification and Optimization Department
Sandia National Laboratories
P.O. Box 5800, MS 1319
Albuquerque, NM 87185–1319
bartv@sandia.gov

Jeffrey A. Fike
Component Science & Mechanics Department
Sandia National Laboratories
P.O. Box 5800, MS 1070
Albuquerque, NM 87185–1070
jafike@sandia.gov

Abstract

This report describes work performed from June 2012 through May 2014 as a part of a Sandia Early Career Laboratory Directed Research and Development (LDRD) project led by the first author. The objective of the project is to investigate methods for building stable and efficient proper orthogonal decomposition (POD)/Galerkin reduced order models (ROMs): models derived from a sequence of high-fidelity simulations but having a much lower computational cost. Since they are, by construction, small and fast, ROMs can enable real-time simulations of complex systems for on-the-spot analysis, control and decision-making in the presence of uncertainty. Of particular interest to Sandia is the use of ROMs for the quantification of the compressible captive-carry environment, simulated for the design and qualification of nuclear weapons systems. It is an unfortunate reality that many ROM techniques are computationally intractable or lack an *a priori* stability guarantee for compressible flows. For this reason, this LDRD project focuses on the development of techniques for building provably stable projection-based ROMs. Model reduction approaches based on continuous as well as discrete projection are considered.

In the first part of this report, an approach for building energy-stable Galerkin ROMs for linear hyperbolic or incompletely parabolic systems of partial differential equations (PDEs) using continuous projection is developed. The key idea is to apply a transformation induced by the Lyapunov function for the system, and to build the ROM in the transformed variables. It is shown that, for many PDE systems including the linearized compressible Euler and linearized compressible Navier-Stokes equations, the desired transformation is induced by a special inner product, termed the “symmetry inner product”. Attention is then turned to nonlinear conservation laws. A new transformation and corresponding energy-based inner product for the full nonlinear compressible Navier-Stokes equations is derived, and it is demonstrated that if a Galerkin ROM is constructed in this inner product, the ROM system energy will be bounded in a way that is consistent with the behavior of the exact solution to these PDEs, i.e., the ROM will be energy-stable. The viability of the linear as well as nonlinear continuous projection model reduction approaches developed as a part of this project is evaluated on several test cases, including the cavity configuration of interest in the targeted application area.

In the second part of this report, some POD/Galerkin approaches for building stable ROMs using discrete projection are explored. It is shown that, for generic linear time-invariant (LTI) systems, a discrete counterpart of the continuous symmetry inner product is a weighted L^2 inner product obtained by solving a Lyapunov equation. This inner product was first proposed by Rowley *et al.*, and

is termed herein the “Lyapunov inner product”. Comparisons between the symmetry inner product and the Lyapunov inner product are made, and the performance of ROMs constructed using these inner products is evaluated on several benchmark test cases. Also in the second part of this report, a new ROM stabilization approach, termed “ROM stabilization via optimization-based eigenvalue reassignment”, is developed for generic LTI systems. At the heart of this method is a constrained nonlinear least-squares optimization problem that is formulated and solved numerically to ensure accuracy of the stabilized ROM. Numerical studies reveal that the optimization problem is computationally inexpensive to solve, and that the new stabilization approach delivers ROMs that are stable as well as accurate.

Summaries of “lessons learned” and perspectives for future work motivated by this LDRD project are provided at the end of each of the two main chapters.

Acknowledgment

This research was funded by Sandia National Laboratories Laboratory Directed Research and Development (LDRD) program. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The authors would like to thank Prof. Lou Cattafesta at Florida State University and Prof. Karen Willcox at the Massachusetts Institute of Technology for useful discussions that lead to some of the ideas presented in Chapter 6 of this report.

Contents

List of Figures

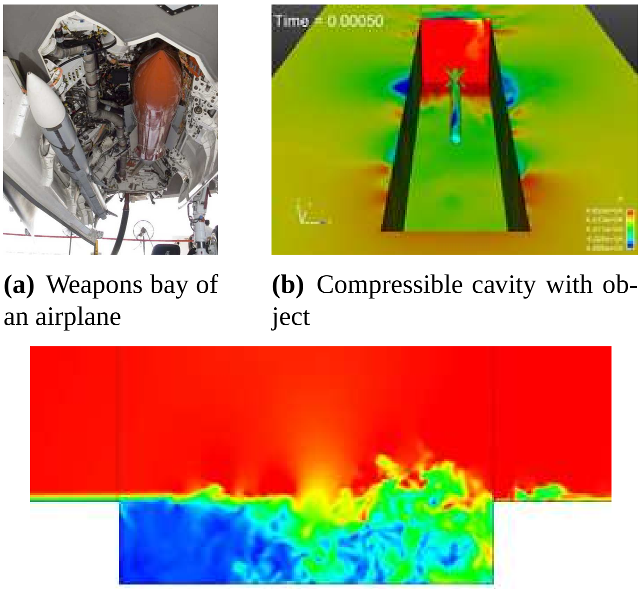
List of Tables

Chapter 1

Introduction

Numerous modern-day engineering problems require the simulation of complex systems with tens of millions or more unknowns. Despite improved algorithms and the availability of massively parallel computing resources, “high-fidelity” models are, in practice, often too computationally expensive for use in a design or analysis setting. An example of an application area of interest to Sandia in which this situation arises is the quantification of the captive-carry environment for the design of nuclear weapons systems (illustrated in Figure 1.1). Since a weapons bay and its contents experience large unsteady pressure loads when exposed to the grazing external flow field, special care must be taken to design these components such that they are able to withstand loads of this magnitude. Large Eddy Simulations (LES) with very fine meshes and long run times are required to predict accurately these dynamic loads. The simulations can take on the order of *weeks* to complete even when run in parallel on state-of-the-art supercomputers. The fact that they need to be repeated numerous times in a design, qualification and/or uncertainty quantification (UQ) setting presents an intractable computational burden. This situation has prompted researchers to develop reduced order models (ROMs): models constructed from high-fidelity simulations that retain the essential physics and dynamics of their corresponding full order models (FOMs), but have a much lower computational cost. Since ROMs are, by construction, small, they can enable uncertainty quantification as well as on-the-spot decision making and/or control.

In order to serve as a useful predictive tool, a ROM should possess the following properties: consistency (with respect to its corresponding high-fidelity model), stability, and convergence (to the solution of its corresponding high-fidelity model). The second of these properties, namely numerical stability, is particularly important, as it is a prerequisite for studying the convergence and accuracy of a ROM. It is well-known that popular model reduction approaches known as the proper orthogonal decomposition (POD) method [88, 12, 50] and the balanced proper orthogonal decomposition (BPOD) method [95, 79] lack, in general, an *a priori* stability guarantee. It is emphasized that instability of POD/BPOD ROMs is a *real* problem in some applications, notably in the field of fluid mechanics, where it is encountered in compressible and/or high Reynolds number flows (precisely the dynamics that are modeled in the targeted application area, the captive carry problem!). While there does exist a model reduction technique that has a rigorous stability guarantee, known as balanced truncation [73, 42], the computational cost of this method, which requires the computation and simultaneous diagonalization of infinite controllability and observability Gramians, makes this method computationally intractable for systems of very large dimensions (i.e., systems with more than 10,000 degrees of freedom [81]).



(a) Weapons bay of an airplane

(b) Compressible cavity with object

(c) 2D cross-section of compressible cavity

Figure 1.1. Compressible captive-carry problem

The objective of this Sandia National Laboratories' Laboratory Directed Research and Development (LDRD) project is to explore new approaches for developing stable and efficient reduced order models, and to study the viability of these models in predicting the dynamics of the compressible cavities modeled in the targeted captive-carry application. The project has theoretical as well as practical milestones:

- *Theoretical:* To develop novel stability-preserving model reduction approaches that could impact Sandia as well as the broader scientific community.

- *Practical:* To create software that enables the numerical study of the approaches developed as a part of this project.

The model reduction approaches identified as the most promising in this report will be implemented in the Sandia in-house LES flow solver, `SIGMA CFD`, currently used to simulate the captive-carry environment. Thus, the work described herein can be seen as a necessary first step towards providing a breakthrough capability for a mission-critical Sandia application.

The remainder of this report is organized as follows.

- Chapters 2–4 contain some preliminaries. Chapter 2 gives a brief literature review summarizing various approaches to building stable POD/Galerkin ROMs. Several notions of stability that are employed throughout this report are defined in Chapter 3. Projection-based model reduction, in particular, the POD/Galerkin method is overviewed in Chapter 4. Two projection approaches are detailed and compared/contrasted: continuous projection and discrete projection.

- Chapter 5 focuses on the development of energy-stable ROMs for linear as well as nonlinear conservation laws using continuous projection. First, the linear case is considered. The energy-stable model reduction approach developed specifically for the equations of linearized compressible inviscid flow in [20] is extended to generic linearized conservation laws. The key idea is to apply a transformation induced by the Lyapunov function for the system, and to build the ROM in the transformed variables. It is shown that, for many linearized PDE systems of the hyperbolic and incompletely parabolic type, the desired transformation is induced by a special weighted L^2 inner product, termed the "symmetry inner product". Next, a new methodology for building energy-stable ROMs for the full nonlinear compressible Navier-Stokes equations is developed. The approach is based on a carefully constructed transformation and energy inner product derived for the PDEs of interest. The proposed linear as well as nonlinear model reduction approaches are evaluated on several test cases, including the compressible cavity problem of interest, following a discussion of the ROM code (known as `Spirit`) in which the methods are implemented.

- Chapter 6 explores stability-preserving model reduction approaches based on discrete projection. It is demonstrated that a discrete weighted L^2 inner product first derived by Rowley *et al.* in [81] and termed herein the "Lyapunov inner product" is a discrete counterpart of the symmetry inner product (introduced in Chapter 5). The weighting matrix that defines the Lyapunov inner product can be computed in a black-box fashion for a stable linear time-invariant (LTI) system arising from the discretization of a linear system of PDEs in space. Some numerical studies of POD ROMs constructed in the Lyapunov inner product are performed, and comparisons are made to balanced truncation. Next, a new approach for stabilizing projection-based ROMs for LTI systems is developed, termed ROM stabilization via eigenvalue reassignment [61]. In this approach, a constrained nonlinear least-squares optimization problem that minimizes the error in the ROM output (thereby maximizing the accuracy of the ROM) is formulated. The said optimization problem is small, with at most as many degrees of freedom (dofs) as the number of dofs in the ROM, and therefore computationally inexpensive to solve. Numerical results reveal that the new stabilization approach delivers ROMs that are both stable and accurate.

- Some conclusions, a summary of "lessons learned", and perspectives for future work are discussed at the end of each of the two main chapters of this report, Chapter 5 and Chapter 6.

The reader is referred to the following SAND reports and articles written during the time of this LDRD project for more details on the topics described in this report: [55, 59, 60, 61, 35].

In addition to these publications, the ideas summarized herein were communicated to the broader scientific community in the following presentations given by the Principal Investigator (PI):

- I. Kalashnikova, S. Arunajatesan. "A Stable Galerkin Reduced Order Model (ROM) for Compressible Flow". *10th World Congress on Computational Mechanics (WCCM X)* (invited), Sao Paulo, Brazil, July 13, 2012.

- I. Kalashnikova, S. Arunajatesan. "Towards Feedback Control of Compressible Flows Using

Galerkin Reduced Order Models". *Second International Workshop on Model Reduction for Parameterized Systems (MoRePaS II)*, Schloss Reisensburg, Gunzburg, Germany, Oct. 2–5, 2012.

- I. Kalashnikova, S. Arunajatesan, B. van Bloemen Waanders. "Energy-Stable Galerkin Reduced Order Models for Prediction and Control of Fluid Systems". *SIAM Conference on Computational Science and Engineering (CSE13)* (invited), Boston, MA, Feb. 26, 2013.

- I. Kalashnikova , B. van Bloemen Waanders, S. Arunajatesan, M. Barone. "Stabilization of Galerkin Reduced Order Models (ROMs) for LTI Systems Using Controllers". *SIAM Conference on Control and Its Applications (CT13),* San Diego, CA, July 9, 2013.

- I. Kalashnikova, B. van Bloemen Waanders, S. Arunajatesan, M. Barone. "Stabilized Projection-Based Reduced Order Models for Uncertainty Quantification". *SIAM Conference on Uncertainty Quantification (SIAM UQ14)*, Savannah, GA, Mar. 31–Apr. 3, 2014.

- I. Kalashnikova, J. Fike, M. Barone, S. Arunajatesan. "Energy-stable Galerkin Reduced Order Models (ROMs) for Nonlinear Compressible Flow". *11th World Congress on Computational Mechanics (WCCM XI)*, Barcelona, Spain, July 20–25, 2014.

Chapter 2

Literature review

As mentioned in Chapter 1, reduced order models constructed using the POD/Galerkin method (Chapter 4) lack in general an *a priori* stability guarantee. The stability of these reduced models is commonly evaluated *a posteriori*: a ROM is constructed, used to predict some dynamical behavior, and subsequently deemed a success if the solutions generated by the ROM are numerically stable and accurately reproduce the expected behavior. There is some risk inherent in this sort of analysis: there is always the possibility that the ROM solution will exhibit non-physical unstable dynamics, which can lead to practical limitations of the ROM. Unfortunately, ROM instability is a *real* problem in some applications, including the targeted compressible cavity problem: as demonstrated in [25], a compressible fluid POD/Galerkin ROM might be stable for a given number of modes, but unstable for other choices of basis size.

Before beginning the discussion of the stability-preserving model reduction methods studied and developed as a part of this LDRD project, a concise review of existing approaches for building stable projection-based ROMs is given. It is noted that there *does* exist a projection-based model reduction technique possessing a rigorous stability guarantee, namely balanced truncation [73, 10], but this method is not considered here as it is not practical for the problem of interest: it is computationally intractable for systems having more than 10,000 dofs [81], and is, in general, limited to linear problems.

A literature search reveals that approaches for developing stability-preserving projection-based ROMs based on the POD fall into several categories.

The first category of methods derives (*a priori*) a stability-preserving model reduction framework, often specific to a particular equation set. In [81], Rowley *et al.* show that Galerkin projection preserves the stability of an equilibrium point at the origin if the ROM is constructed in an "energy-based" inner product. In [20, 57], Barone *et al.* demonstrate that a symmetry transformation leads to a stable formulation for a Galerkin ROM for the linearized compressible Euler equations [20, 57] and nonlinear compressible Navier-Stokes equations [55] with solid wall and far-field boundary conditions. In [85], Serre *et al.* propose applying the stabilizing projection developed by Barone *et al.* in [20, 57] to a skew-symmetric system constructed by augmenting a given linear system with its adjoint system. This approach yields a ROM that is stable at finite time even if the solution energy of the full-order model is growing. In [86, 87], Sirisup *et al.* develop a method for correcting long-term unstable behavior for POD models using a spectral viscosity (SV) diffusion convolution operator. The advantage of approaches such as these is they are physics-based, and guarantee *a priori* a stable ROM; the downside is that they can be difficult to implement, as access to the high-fidelity code and/or the governing partial differential equations (PDEs) is often required.

Another category of methods is aimed to remedy the so-called “mode truncation instability”. These methods [5, 76, 22, 15, 92], motivated by the observation that higher order modes can give rise to non-physical instabilities in the ROM system, are often physics-based and minimally intrusive to the ROM. In [15], a ROM stabilization methodology that achieves improved accuracy and stability through the use of a new set of basis functions representing the small, energy-dissipation scales of turbulent flows is derived by Balajewicz *et al.* The stabilization of ROMs using shift modes and residual modes was proposed in [76] and [22] by Noack *et al.* and Bergmann *et al.* respectively. Other authors, e.g., Terragni *et al.* [92], have demonstrated that the stability and performance of a ROM can be improved by adapting the POD manifold to the local dynamics.

In a third category of approaches, an unstable ROM is stabilized through a post-processing (*a posteriori*) stabilization step applied to an unstable algebraic ROM system. Ideally, the stabilization only minimally alters the ROM physics, so that the ROM’s accuracy is not sacrificed. In [7], Amsallem *et al.* propose a method for stabilizing projection-based linear ROMs through the solution of a small-scale convex optimization problem. In [24], a set of linear constraints for the left-projection matrix, given the right-projection matrix, are derived by Bond *et al.* to yield a projection framework that is guaranteed to generate a stable ROM. In [94], Zhu *et al.* derive some LES closure models for POD ROMs for the incompressible Navier-Stokes equations, and demonstrate numerically that the inclusion of these LES terms yields a ROM with increased numerical stability. In [32], Couplet *et al.* propose methods for correcting the behavior of a low-order POD-Galerkin system through a coefficient calibration/minimization. A nice feature of these and similar approaches is that they are easy to implement: often the stabilization step can be applied in a “black-box” fashion to an algebraic ROM system that has already been constructed. However, the approaches can give rise to inconsistencies between the ROM and FOM physics, thereby limiting the accuracy of the ROM.

Other ROM approaches, e.g., the Gauss–Newton with Approximated Tensors (GNAT) method of Carlberg *et al.* [27], circumvent the stability issue by formulating the ROM at the fully discrete level, that is, by projecting the ROM equations only after they have been discretized in space as well as in time. Various heuristics for obtaining stable ROMs have also been noted. For example, in [8], Amsallem *et al.* suggest that projection-based ROMs constructed for LTI systems in descriptor form tend to possess better numerical stability properties than those constructed for LTI systems in non-descriptor form.

The approaches described in this report fall into the first (Chapter 5 and Section 6.1) and third categories (Section 6.2) summarized above.

Chapter 3

Stability definitions

As stated in Chapter 1, this LDRD project focuses on the development of projection-based ROMs that possess an *a priori* stability guarantee. Before beginning this discussion, some general definitions of stability that will be used in the subsequent analysis are reviewed. Of particular interest is the concept of energy-stability, which appears in the stability proofs given in subsequent chapters. The term energy-stability is defined in Section 3.1, and related to more classical definitions of stability, e.g., Lyapunov stability, asymptotic stability, exponential stability and time-stability, in Sections 3.2–3.3.

3.1 Energy-stability

The concept of energy-stability originated in the literature involving the numerical analysis of spectral and finite difference discretizations to time-dependent PDEs [40, 47, 45]. It has also appeared in the Galerkin finite element method literature, e.g., [44, 67], where the energy-method was employed to derive stable Galerkin methods for hyperbolic conservation laws. It is well-known that physical systems admit a certain energy structure. The basic idea behind building energy-stable ROMs is that a ROM constructed for such systems should preserve this energy structure. Among the authors who have explored the concept of energy-stability in the context of model reduction are Rowley *et al.* [79] and Kwasniok [66]. In [79], Rowley *et al.* introduced a family of "energy-based" inner products for the purpose of constructing stable Galerkin ROMs for fluid problems. In [66], Kwasniok recognized the role of energy conservation in ROMs of nonlinear, incompressible fluid flow for atmospheric modeling applications, and proposed a Galerkin projection approach in which the ROM conserves turbulent kinetic energy or turbulent enstrophy.

The notion of energy-stability will be introduced in the context of a specific canonical model problem, then generalized. Consider, without loss of generality, the following scalar initial value problem, known as a Cauchy problem [64]:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathscr{L}(\mathbf{x}(t)), \quad t \geq 0 \\ \mathbf{x}(0) &= \mathbf{f}. \end{aligned} \tag{3.1}$$

Here, $\mathscr{L}$ denotes a linear differential operator with constant coefficients, $\mathbf{f} \in \mathbb{R}^n$ is the initial condition, and $\mathbf{x}(t) \in \mathbb{R}^n$ is the system state at time $t \geq 0$. The operator $\mathscr{L}$ is said to be semi-bounded with respect to an inner product $(\cdot,\cdot)$ if it satisfies the following inequality for all sufficiently smooth functions $\mathbf{w} \in L^2$:

$$(\mathbf{w}, \mathscr{L}\mathbf{w}) \leq \alpha(\mathbf{w}, \mathbf{w}), \tag{3.2}$$

where $\alpha \in \mathbb{R}$. The following theorem (quoted from [64]) states the conditions under which the Cauchy problem (3.1) is well-posed.

Theorem 3.1.1 [64]: The Cauchy problem (3.1) is well-posed if and only if the operator $\mathscr{L}$ is semi-bounded with respect to an inner product $(\cdot,\cdot)$ which corresponds to a norm equivalent to the L^2 norm.

Consider now a Galerkin approximation to (3.1), denoted here by $\mathbf{x}_N$, and satisfying

$$(\dot{\mathbf{x}}_N, \boldsymbol{\phi}) = (\mathscr{L}(\mathbf{x}_N), \boldsymbol{\phi}), \tag{3.3}$$

for all $\boldsymbol{\phi}$ sufficiently smooth, and suppose $\mathscr{L}$ is semi-bounded with respect to $(\cdot,\cdot)$. Setting $\boldsymbol{\phi} = \mathbf{x}_N$ in (3.3) leads to the following energy estimate for the Galerkin approximation:

$$\frac{dE_N}{dt} \leq 2\alpha E_N, \tag{3.4}$$

where $E_N \equiv \frac{1}{2}||\mathbf{x}_N||^2$ denotes the energy of the Galerkin approximation $\mathbf{x}_N$, and $||\cdot||$ is the norm induced by the inner product $(\cdot,\cdot)$. Applying Gronwall's lemma ((A.1) in Appendix A.1) to (3.4) gives the inequality

$$||\mathbf{x}_N(t)|| \leq e^{\frac{1}{2}\alpha t}||\mathbf{x}_N(0)||. \tag{3.5}$$

The result (3.5) says that the energy of the numerical solution to (3.3) is bounded in a way that is consistent with the behavior of the energy of the exact solution to the original differential equation (3.1), i.e., the numerical solution is energy-stable. This definition can be extended to a generic ROM system.

Definition 3.1.2 (Energy-Stability [45]): A ROM system is called energy-stable if

$$E_M(t) \leq e^{\alpha t} E_M(0), \tag{3.6}$$

for some constant $\alpha \in \mathbb{R}$, where

$$E_M \equiv \frac{1}{2}||\mathbf{x}_M||^2 \tag{3.7}$$

is the system energy of the ROM numerical solution $\mathbf{x}_M$, and $||\cdot||$ is a norm equivalent to the L^2 norm.

In general, a ROM system is not guaranteed to satisfy Definition 3.1.2 even if the underlying initial boundary value problem (IBVP) is well-posed and the full order system arising from the discretization of the governing PDEs in space is stable. However, it is often possible to ensure (3.6) holds for the ROM system through a careful selection of the reduced trial and test bases $\mathbf{\Phi}_M$ and $\mathbf{\Psi}_M$ and/or the inner product in which the projection step of the model reduction is performed (Chapters 5 and 6).

3.2 Lyapunov, asymptotic and exponential stability

The concept of energy-stability can be related to classical notions of stability, namely Lyapunov stability, asymptotic stability and exponential stability. Consider an autonomous nonlinear dynamical system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, \tag{3.8}$$

where $\mathbf{f} \in \mathbb{R}^n$ is a given function, subject to some initial condition $\mathbf{x}(0) = \mathbf{x}_0$. Let $\mathbf{x}_e$ be an equilibrium point of the system (3.8), meaning $\mathbf{f}(\mathbf{x}_e) = \mathbf{0}$ for all $t \geq 0$.

Definition 3.2.1 (Lyapunov, asymptotic and exponential stability) [11]: The equilibrium point $\mathbf{x}_e$ of (3.8) is said to be:

(a) *Lyapunov stable* if $\forall \varepsilon > 0$ there exists a $\delta(\varepsilon) > 0$ such that if $||\mathbf{x}(0) - \mathbf{x}_e|| < \delta$, then $||\mathbf{x}(t) - \mathbf{x}_e|| < \varepsilon \ \forall t \geq 0$.

(b) *Asymptotically stable* if there exists a $\delta > 0$ such that if $||\mathbf{x}(0) - \mathbf{x}_e|| < \delta$, then $\lim_{t\to\infty} ||\mathbf{x}(t) - \mathbf{x}_e|| = 0$.

(c) *Exponentially stable* if there exist $\alpha, \beta, \delta > 0$ such that if $||\mathbf{x}(0) - \mathbf{x}_e|| < \delta$, then $||\mathbf{x}(t) - \mathbf{x}_e|| \leq \alpha ||\mathbf{x}(0) - \mathbf{x}_e|| e^{-\beta t} \ \forall t \geq 0$.

In other words, if an equilibrium point of (3.8) is Lyapunov stable, solutions within a distance $\delta > 0$ from it will remain a distance $\varepsilon > 0$ from it for all time; if it is asymptotically stable, solutions within this distance will eventually converge to the equilibrium; if it is exponentially stable, the solutions will not only converge, but at an exponential rate. In general, exponential stability implies asymptotic stability, and asymptotic stability implies Lyapunov stability.

The following theorem, known as the Lyapunov Stability Theorem [11], can be used to characterize the stability of an equilibrium point $\mathbf{x}_e$ for (3.8).

Theorem 3.2.2 (Lyapunov Stability Theorem) [11]: Let V be a non-negative function on $\mathbb{R}^n$ and let $\dot{V}$ represent the time derivative of V along trajectories of the system dynamics (3.8), i.e., $\dot{V} = \frac{\partial V}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x})$. Let $B_r = B_r(\mathbf{x}_e)$ be a ball of radius r around an equilibrium point $\mathbf{x}_e$ of (3.8). If there exists an $r > 0$ such that V is positive definite and $\dot{V}$ is negative semi-definite for all $\mathbf{x} \in B_r$, then $\mathbf{x}_e$ is Lyapunov stable.

The function V defined in Theorem 3.2.2 above is known as the Lyapunov function for the system (3.8). Observe that the numerical energy E_N defined in (3.7) satisfies the definition of a Lyapunov function (Theorem 3.2.2) if $\frac{dE_N}{dt} \leq 0$. Thus, if a ROM is energy-stable with $\alpha = 0$ (Definition 3.1.2), then the ROM is Lyapunov stable. In Chapter 5 and Section 6.1, it is shown how Theorem 3.1.2 can be used to define a stability-preserving inner product for building stable ROMs.

The stability concepts introduced above simplify for the specific case of linear systems. It is straightforward to verify that for linear systems, asymptotic and exponential stability are equivalent. Moreover, the following result holds.

Theorem 3.2.3 (Asymptotic Stability Theorem for Linear Systems) [11]: A linear ROM system of the form $\frac{d\mathbf{x}_M}{dt} = \mathbf{A}_M\mathbf{x}_M$ is asymptotically (and exponentially) stable if and only if all the eigenvalues of the ROM system matrix $\mathbf{A}_M$ have strictly negative real parts.

Theorem 3.2.3 is commonly used to check numerically (*a posteriori*) the stability of a linear ROM constructed for a linear system (Sections 5.5, 6.1.1 and 6.2.4).

3.3 Time-stability

Another form of stability is what is referred to herein as "time-stability". Essentially, a system that is time-stable is one whose solution will not "blow up" (i.e., produce an unbounded output) given a finite input and/or non-zero initial condition. For a general nonlinear system, exponential stability implies time-stability, but time-stability is a stronger notion than asymptotic stability [48]. Since exponential and asymptotic stability are equivalent for linear systems, asymptotic stability *does* imply time-stability in this special case.

The concept of time-stability can also be defined in terms of the system energy.

Definition 3.3.1 (Time-Stability [45]): A ROM system is called time-stable if the numerical energy of the ROM solution is non-increasing in time for an arbitrary time step, i.e., if

$$\frac{dE_N}{dt} \leq 0. \tag{3.9}$$

It is straightforward to demonstrate that a time-stable scheme is also energy-stable. Suppose a ROM is time-stable, so the ROM solution satisfies the energy estimate (3.9). Applying Gronwall's lemma ((A.1) in Appendix A.1) to this inequality, $E_N(t) \leq E_N(0)$. Thus, (3.6) holds with $\alpha = 0$.

In general, the converse of the above statement does not hold: energy-stability does not necessarily imply time-stability. This is to be expected. The practical implication of a ROM possessing the energy-stability property is that its numerical solution is bounded in a way that is consistent with the behavior of the exact solutions of the governing equations. It is possible in general that the governing PDEs support instabilities. In this case, an energy-stable ROM may possess unbounded solutions as $t \longrightarrow \infty$, as (it can be argued) it should, if these unbounded solutions are physical.

Chapter 4

Proper Orthogonal Decomposition (POD)/Galerkin method to model reduction

In this chapter, the POD/Galerkin method to model reduction is overviewed. The POD/Galerkin method is a projection-based method. The projection step of the model reduction (Section 4.3) can be performed at either the discrete or continuous level of the governing equations. In the discrete projection approach, a semi-discrete (or fully discrete [27]) representation of the equations is projected onto the POD modes in a discrete inner product. In contrast, in the continuous projection approach, the Galerkin projection step is applied to the continuous system of PDEs in a continuous inner product onto a continuous representation of the POD basis. A comparison of these two projection methodologies is deferred until Section 4.3.

Consider a generic PDE (or system of PDEs) of the form

$$\dot{\mathbf{x}}(t) = \mathscr{L}(\mathbf{x}(t)) + \mathscr{N}(\mathbf{x}(t)), \tag{4.1}$$

in an open bounded domain Ω. In (4.1), $\mathscr{L} : \mathbb{R}^n \to \mathbb{R}^n$ is a linear spatial differential operator, and $\mathscr{N} : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear spatial differential operator. The symbol t denotes time, and $\mathbf{x} \in \mathbb{R}^n$ is called the state vector. The '$\cdot$' symbol denotes differentiation with respect to time, i.e., $\dot{\mathbf{x}} \equiv \frac{\partial \mathbf{x}}{\partial t}$.

Suppose the PDE system (4.1) has been discretized in space using some numerical scheme, e.g., the finite element method. The result will be a semi-discrete nonlinear system of the form:

$$\dot{\mathbf{x}}_N(t) = \mathbf{A}\mathbf{x}_N(t) + \mathbf{N}(\mathbf{x}_N(t)), \tag{4.2}$$

where $\mathbf{x}_N \in \mathbb{R}^N$ is the discretized state vector, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a temporally-constant matrix arising from the discretization of the linear operator $\mathscr{L}$ in (4.1), and $\mathbf{N} \in \mathbb{R}^N$ is a nonlinear function arising from the discretization of the nonlinear operator $\mathscr{N}$ in (4.1).

The general approach to projection-based model reduction consists of three steps, described below (Sections 4.1– 4.3).

Step 1: Calculation of reduced trial and test bases, denoted by $\mathbf{\Phi}_M = \begin{pmatrix} \boldsymbol{\phi}_1, & \cdots, & \boldsymbol{\phi}_M \end{pmatrix}$ and $\mathbf{\Psi}_M = \begin{pmatrix} \boldsymbol{\psi}_1, & \cdots, & \boldsymbol{\psi}_M \end{pmatrix}$ respectively, each of order M, with $M << N$.

Step 2: Approximation of the solution to (4.1) or (4.2) by

$$\mathbf{x}(t) \approx \sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i = \mathbf{\Phi}_M \mathbf{x}_M(t), \tag{4.3}$$

where $x_{M,i}(t)$ are the unknown ROM coefficients or modal amplitudes, to be determined in solving the ROM.

Step 3: Substitution of the approximation (4.3) into the governing system ((4.1) or (4.2)) and projection of this system onto the reduced test basis.

The result of this procedure is a "small" (size $M << N$) dynamical system that, for a suitable choice of reduced bases, accurately describes the dynamics of the full order system for some set of conditions. The reduced bases $\mathbf{\Phi}_M \in \mathbb{R}^{N\times M}$ and $\mathbf{\Psi}_M \in \mathbb{R}^{N\times M}$ are functions of space but not time, and are assumed to have full column rank. In the case that $\mathbf{\Psi}_M \neq \mathbf{\Phi}_M$, the projection is referred to as a Petrov-Galerkin projection. Otherwise, if $\mathbf{\Psi}_M = \mathbf{\Phi}_M$, the projection is referred to as a Galerkin projection. This terminology is introduced here as it will be shown later that the energy-stable model reduction approaches derived in this work are effectively Petrov-Galerkin methods.

4.1 Calculation of the reduced bases (*Step 1*)

There exist a number of approaches for calculating the reduced basis modes (*Step 1* of the model reduction), e.g., the POD method [88, 12, 50], the BPOD method [95, 79], the balanced truncation method [73, 42], the reduced basis method [93, 83]; also methods based on goal-oriented bases [25], generalized eigenmodes [17], and Koopman modes [82]. Attention is restricted here to the POD basis, but it is noted that the energy-stability results derived in this report hold for *any* choice of reduced basis. The reason for the choice of the POD reduced basis is two-fold. First, the POD is a widely used approach for computing efficient bases for dynamical systems. Moreover, ROMs constructed via the POD/Galerkin method lack in general an *a priori* stability guarantee (meaning POD/Galerkin ROMs would benefit from stability-preserving model reduction approaches such as those developed herein).

Discussed in detail in Lumley [71] and Holmes *et al.* [50], POD is a mathematical procedure that, given an ensemble of data and an inner product, denoted generically by $(\cdot,\cdot)$, constructs a basis for the ensemble. This basis is optimal in the sense that it describes more energy (on average) of the ensemble in the chosen inner product than any other linear basis of the same dimension M. The ensemble $\{\mathbf{x}^k : k = 1,\ldots,K\}$ is typically a set of K instantaneous snapshots of a numerical solution field, taken for K values of a parameter of interest, or at K different times. Mathematically, POD seeks an M-dimensional ($M << K$) subspace spanned by the set $\{\boldsymbol{\phi}_i\}$ such that the projection of the difference between the ensemble $\mathbf{x}^k$ and its projection onto the reduced subspace is minimized on average. It is a well-known result [20, 50, 65, 78] that the solution to the POD optimization problem reduces to the eigenvalue problem

$$\mathbf{R}\boldsymbol{\phi} = \lambda\boldsymbol{\phi}, \tag{4.4}$$

where $\mathbf{R}$ is a self-adjoint and positive semi-definite operator with its (i,j) entry given by $R_{ij} = \frac{1}{K}\left(\mathbf{x}^i,\mathbf{x}^j\right)$ for $1 \leq i,j \leq K$. If it is assumed that $\mathbf{R}$ is compact, then there exists a countable set of non-negative eigenvalues λ_i with associated eigenfunctions $\boldsymbol{\phi}_i$. It can be shown [50, 71] that the set of M eigenfunctions, or POD modes, $\{\boldsymbol{\phi}_i : i = 1, \ldots, M\}$ corresponding to the M largest eigenvalues of $\mathbf{R}$ is precisely the desired basis. This is the so-called "method of snapshots" for computing a POD basis [88].

4.2 Approximation of solution in reduced basis *(Step 2)*

Once the reduced basis is calculated, the solution $\mathbf{x}(t)$ is approximated as a linear combination of the reduced basis modes (4.3) (*Step 2*). Given this approximation, the following error formula can be shown for the POD [50, 65]:

$$\frac{1}{K}\sum_{i=1}^{K}\left|\left|\mathbf{x}^i - \sum_{j=1}^{M}\left(\mathbf{x}^i, \boldsymbol{\phi}_j\right)\boldsymbol{\phi}_j\right|\right|^2 = \sum_{k=M+1}^{J}\lambda_k, \tag{4.5}$$

where $J = \dim\left(\left\{\mathbf{x}^1, ..., \mathbf{x}^K\right\}\right)$, and where $\lambda_1 \geq ... \geq \lambda_J > 0$ are the positive eigenvalues of the operator $\mathbf{R}$ (4.4).

Typically, the size of the reduced basis is chosen based on an energy criterion. That is, M is selected to be the minimum integer such that

$$E_{POD}(M) \geq \text{tol}, \tag{4.6}$$

where $0 \leq \text{tol} \leq 1$ represents the snapshot energy represented by the POD basis, and

$$E_{POD}(M) \equiv \frac{\sum_{i=1}^{M}\lambda_i}{\sum_{i=1}^{K}\lambda_i}. \tag{4.7}$$

4.3 Projection (*Step 3*)

There are two approaches for performing *Step 3* of the model reduction: continuous and discrete projection. These approaches are described, as well as compared and contrasted, in the present subsection. Stability-preserving methods for constructing ROMs using these approaches will be detailed in Chapters 5 and 6.

4.3.1 Model reduction via continuous projection

In the continuous projection approach [20, 57], the continuous system of PDEs (4.1) is projected onto a continuous reduced test basis $\{\boldsymbol{\psi}_i\}_{i=1}^{M} \in \mathbb{R}^n$ in a continuous inner product $(\cdot, \cdot)$, for example, the usual L^2 inner product[1]

$$\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = \int_{\Omega}\mathbf{x}^{(1)T}\mathbf{x}^{(2)}d\Omega, \tag{4.8}$$

[1] Weighted variants of the L^2 inner product are considered later in this work.

where the $x_{M,i}(t)$ are the unknown ROM coefficients or modal amplitudes $\big($so that $\mathbf{x}_M^T \equiv \begin{pmatrix} x_{M,1}, & \cdots, & x_{M,M} \end{pmatrix}\big)$, to be determined in solving the ROM dynamical system (derived below).

Substituting (4.3) into (4.1), the following is obtained

$$\sum_{i=1}^{M} \dot{x}_{M,i}(t)\boldsymbol{\phi}_i = \mathscr{L}\left(\sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i\right) + \mathscr{N}\left(\sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i\right). \tag{4.9}$$

Next, a reduced test basis $\{\boldsymbol{\psi}_i\}_{i=1}^M \in \mathbb{R}^n$ is introduced, and the system of PDEs (4.9) is projected onto the reduced test basis modes $\boldsymbol{\psi}_j$ for $j = 1,2,...,M$ in the inner product $(\cdot,\cdot)$ to yield

$$\sum_{i=1}^{M} \dot{x}_{M,i}(t)\left(\boldsymbol{\psi}_j, \boldsymbol{\phi}_i\right) = \left(\boldsymbol{\psi}_j, \mathscr{L}\left(\sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i\right)\right) + \left(\boldsymbol{\psi}_j, \mathscr{N}\left(\sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i\right)\right), \tag{4.10}$$

for $j = 1,2,...,M$. Typically, the reduced trial and test bases $\boldsymbol{\phi}_i$ and $\boldsymbol{\psi}_i$ are chosen to be orthonormal in the inner product $(\cdot,\cdot)$, so that $(\boldsymbol{\psi}_j, \boldsymbol{\phi}_i) = \delta_{ij}$, where δ_{ij} denotes the Krönecker delta function. Invoking this property, as well as the linearity property of the operator $\mathscr{L}$, (4.10) simplifies to

$$\dot{x}_{M,j}(t) = \textstyle\sum_{i=1}^{M} x_{M,i}(t)\left(\boldsymbol{\psi}_j, \mathscr{L}(\boldsymbol{\phi}_i)\right) + \left(\boldsymbol{\psi}_j, \mathscr{N}\left(\sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i\right)\right), \tag{4.11}$$

for $j = 1,2,...,M$. The equations (4.11) define a set of M time-dependent nonlinear ordinary differential equations (ODEs) for the modal amplitudes $x_{M,j}(t)$ in (4.3).

4.3.2 Model reduction via discrete projection

In the discrete projection approach, the FOM ODE system (4.2) (the PDE system (4.1) discretized in space) is projected onto a discrete reduced test basis in a discrete inner product. Suppose this discrete inner product is the following weighted L^2 inner product:

$$\left(\mathbf{x}_N^{(1)}, \mathbf{x}_N^{(2)}\right)_{\mathbf{P}} = \mathbf{x}_N^{(1)T}\mathbf{P}\mathbf{x}_N^{(2)}, \tag{4.12}$$

where $\mathbf{P} \in \mathbb{R}^{N\times N}$ is a symmetric positive-definite matrix. Let $\mathbf{\Phi}_M \in \mathbb{R}^{N\times M}$ and $\mathbf{\Psi}_M \in \mathbb{R}^{N\times M}$ denote the reduced trial and reduced test bases for (4.2), respectively. Assume these matrices have full column rank, and are orthonormal in the inner product (4.12), so that $\mathbf{\Psi}_M^T\mathbf{P}\mathbf{\Phi}_M = \mathbf{I}_M$, where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. The first step in constructing a ROM for (4.2) using discrete projection is to approximate the solution $\mathbf{x}_N(t)$ by (4.3). Substituting (4.3) into (4.2), and projecting this system onto the reduced test basis, the following $M \times M$ dynamical system is obtained:

$$\dot{\mathbf{x}}_M(t) = \mathbf{A}_M\mathbf{x}_M(t) + \mathbf{N}_M\left(\mathbf{x}_M(t)\right), \tag{4.13}$$

where

$$\mathbf{A}_M = \mathbf{\Psi}_M^T\mathbf{P}\mathbf{A}\mathbf{\Phi}_M, \quad \mathbf{N}_M(\mathbf{x}_M) = \mathbf{\Psi}_M^T\mathbf{P}\mathbf{N}\left(\mathbf{\Phi}_M\mathbf{x}_M\right). \tag{4.14}$$

4.3.3 Continuous vs. discrete projection

In the majority of applications of reduced order modeling, the discrete projection approach is employed in constructing the ROM. This discrete approach has the advantage that boundary condition terms present in the discretized equation set are often (depending on the implementation) inherited by the ROM; that is, the ROM solution will satisfy the boundary conditions of the FOM. Certain properties of the numerical scheme used to solve the full equations may be inherited by the ROM as well. The discrete approach can be black-box, at least for linear systems ((4.2) with $\mathbf{N} = \emptyset$): it operates on the matrix **A** only, so that access to the high-fidelity code that was used to generate this matrix or the governing PDEs is not required provided **A** can be written out from the high-fidelity code. In contrast, the continuous projection approach is tied to the governing PDEs – the continuous problem (4.9) needs to be translated to the discrete setting, e.g., by interpolating the reduced basis modes and evaluating the continuous inner products in (4.11) using a numerical quadrature [20]. Although the continuous approach is inherently an embedded method, its similarity to spectral numerical approximation methods allows the use of analysis techniques employed by the spectral methods community [16, 57].

Which of the two projection approaches described above (continuous vs. discrete projection) is preferred depends on the application and the type of model reduction approach sought (e.g., embedded vs. black-box). The discussion in the remainder of this report is intended to aid the reader in selecting one of these approaches for his or her problem of interest.

Note that, regardless of which projection approach is used to build the ROM, the ROM dynamical system will have the form (4.13), as (4.11) has this form when written as a matrix problem. The solution to the ROM is obtained by advancing (4.13) forward in time using a time-integration scheme. This stage of the model reduction is known as the "online stage". The preceding steps (the collection of the snapshots used to build the reduced bases and *Steps 1–3* above) comprise the "offline stage" of the model reduction.

4.4 Efficiency considerations

It is straightforward to see that the projection of the linear operator in (4.1) or (4.2) can be computed just one time prior to the online time-integration step of the model reduction; in particular, it need not be recomputed in each time or Newton step of the model reduction. This is also the case when the nonlinear operator $\mathscr{N}$ has only polynomial nonlinearities. To see this, suppose $\mathscr{N}(\cdot)$ has a cubic nonlinearity:

$$\mathscr{N}(\mathbf{x}(t)) = \mathbf{x}^3(t). \tag{4.15}$$

The projection of (4.15) onto the reduced test basis modes is then:

$$\left(\boldsymbol{\psi}_j, \mathscr{N}\left(\sum_{i=1}^{M} x_{M,i}(t)\boldsymbol{\phi}_i\right)\right) = \sum_{l=1}^{M}\sum_{m=1}^{M}\sum_{n=1}^{M} x_{M,l}(t)x_{M,m}(t)x_{M,n}(t)\left(\boldsymbol{\psi}_j, \boldsymbol{\phi}_l\boldsymbol{\phi}_m\boldsymbol{\phi}_n\right), \tag{4.16}$$

for $j = 1, ..., M$. The inner product tensor $\left(\boldsymbol{\psi}_j, \boldsymbol{\phi}_l \boldsymbol{\phi}_m \boldsymbol{\phi}_n\right)$ needs to be computed just once in the offline stage of the model reduction.

In the case $\mathscr{N}$ (and hence **N**) has non-polynomial nonlinearities, the projection of this term *cannot* be pre-computed prior to the time-integration (online) stage of the model reduction. The cost of performing this inner product is $\mathscr{O}(N)$, where N is the number of degrees of freedom in the FOM and often very large ($N >> M$). Hence, "direct" treatment, or computation, of these inner products would invalidate the term *reduced* order model. A number of interpolation methods to overcome this difficulty have been proposed, e.g., the discrete empirical interpolation method (DEIM) [29], "best points" interpolation [74, 75], or gappy POD [33].

In the nonlinear systems considered herein (the compressible Navier-Stokes equations; Section 5.3), the equations and inner products are formulated such that the resulting systems have only polynomial nonlinearities, so that the projection of the nonlinear terms can be computed once in the offline stage of the model reduction and stored for use during the online stage. Hence, the nonlinear interpolation methods mentioned above are not used. Note, however, that the efficiency of the nonlinear ROMs may nonetheless be improved with interpolation of the nonlinear terms. This is because it requires $\mathscr{O}(M^{p+1})$ operations to evaluate the tensor arising from the projection of the nonlinear terms (e.g., the inner product tensor $\left(\boldsymbol{\psi}_j, \boldsymbol{\phi}_l \boldsymbol{\phi}_m \boldsymbol{\phi}_n\right)$ in (4.16) for $p = 3$), where p is the degree of nonlinearity in the function $\mathscr{N}$. This can amount to a cost of $\mathscr{O}(N)$ operations for $p >> 2$ and moderate M.

4.5 Special case: linear time-invariant (LTI) systems

The POD/Galerkin model reduction approach described above is now applied to a special kind of system that will be considered in Chapter 6, namely a linear time-invariant, or LTI, system. A system is called time-invariant if the output response for a given input does not depend on when that input is applied [11].

At the continuous level, an LTI system can be represented by a PDE (or system of PDEs) of the form

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathscr{L}(\mathbf{x}(t)) + \mathscr{L}_c(\mathbf{u}(t)), \\ \mathbf{y}(t) &= \mathscr{L}_o(\mathbf{x}(t)), \end{aligned} \tag{4.17}$$

in $\boldsymbol{\Omega}$. Here, $\mathbf{u} \in \mathbb{R}^p$ represents the vector of control variables, and $\mathbf{y} \in \mathbb{R}^q$ is the measured signal or output. The operator $\mathscr{L} : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth linear spatial-differential operator, like the linear operator that appears in (4.1). The operators $\mathscr{L}_c : \mathbb{R}^p \to \mathbb{R}^n$ and $\mathscr{L}_o : \mathbb{R}^n \to \mathbb{R}^q$ are smooth linear mappings. The abstract operators $\mathscr{L}$, $\mathscr{L}_c$ and $\mathscr{L}_o$ are introduced to keep the discussion as general as possible, and used in subsequent analysis.

If the PDE system (4.17) is discretized in space using some numerical scheme, e.g., the finite element method, the result is a semi-discrete LTI system of the form:

$$\begin{aligned} \dot{\mathbf{x}}_N(t) &= \mathbf{A}\mathbf{x}_N(t) + \mathbf{B}\mathbf{u}_P(t), \\ \mathbf{y}_{QN}(t) &= \mathbf{C}\mathbf{x}_N(t). \end{aligned} \tag{4.18}$$

Here, $\mathbf{u}_P \in \mathbb{R}^P$ is the discretized vector of control variables, and $\mathbf{y}_{QN} \in \mathbb{R}^Q$ is the discretized output; $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times P}$ and $\mathbf{C} \in \mathbb{R}^{Q\times N}$ are constant matrices (in particular, they are not functions of time t).

The first two steps of the model reduction approach for (4.17) or (4.18) are identical to the more general nonlinear case considered above. The step that yields a slightly different (simpler) result is the Galerkin projection step. Substituting (4.3) into (4.17), and projecting onto the reduced test basis in the continuous $(\cdot,\cdot)$ inner product, invoking the orthonormality of the reduced test and reduces trial basis functions, the following is obtained:

$$\begin{array}{rcl} \dot{x}_{M,j}(t) & = & \sum_{i=1}^{M} x_{M,i}(t)\left(\boldsymbol{\psi}_j, \mathscr{L}(\boldsymbol{\phi}_i)\right) + \left(\boldsymbol{\psi}_j, \mathscr{L}_c(\mathbf{u}(t))\right), \\ \mathbf{y}_{QM}(t) & = & \sum_{i=1}^{M} x_{M,i}(t)\mathscr{L}_o(\boldsymbol{\phi}_i), \end{array} \tag{4.19}$$

for $j = 1,2,...,M$. The equations (4.19) define a set of M time-dependent linear ODEs for the modal amplitudes $x_{M,i}(t)$ in (4.3), where $\mathbf{y}_{QM}$ is a reduced approximation of the output.

Similarly, substituting (4.3) into (4.18), and projecting this system onto the reduced test basis in the discrete inner product (4.12), the following $M \times M$ LTI system is obtained:

$$\begin{array}{rcl} \dot{\mathbf{x}}_M(t) & = & \mathbf{A}_M \mathbf{x}_M(t) + \mathbf{B}_M \mathbf{u}_P(t), \\ \mathbf{y}_{QM}(t) & = & \mathbf{C}_M \mathbf{x}_M(t), \end{array} \tag{4.20}$$

where

$$\mathbf{A}_M = \mathbf{\Psi}_M^T \mathbf{P}\mathbf{A}\mathbf{\Phi}_M, \quad \mathbf{B}_M = \mathbf{\Psi}_M^T \mathbf{P}\mathbf{B}, \quad \mathbf{C}_M = \mathbf{C}\mathbf{\Phi}_M. \tag{4.21}$$

The matrices $\mathbf{A}_M$, $\mathbf{B}_M$ and $\mathbf{C}_M$ can be pre-computed and stored in the offline stage of the model reduction – in particular, they need not be re-computed at each time step of the online time-integration stage of the ROM. Note that, when written in matrix form, the ROM constructed via continuous projection (4.19) has exactly the form (4.20).

Chapter 5

Stable ROMs via continuous projection

As stated in Chapter 1, the targeted application of the ROM approaches developed as a part of this LDRD project is the compressible captive-carry problem. For this problem, the governing PDEs are the nonlinear compressible Navier-Stokes equations in the high Reynolds number regime. These PDEs are given in Section 5.3. A literature search [18] reveals that the majority of POD/Galerkin model reduction approaches for fluid flow are based on the incompressible version of these equations. For the incompressible Navier-Stokes equations, the natural choice of inner product for the Galerkin projection step of the model reduction procedure is the L^2 inner product. This is because, in these models, the solution vector is taken to be the velocity vector $\mathbf{u}$, so that $||\mathbf{u}||_{L^2}$ is a measure of the global kinetic energy in the domain, and the POD modes optimally represent the kinetic energy present in the ensemble from which they were generated. The same is *not* true for the compressible Navier-Stokes equations. Hence, if a compressible fluid ROM is constructed in the L^2 inner product, the ROM solution may not satisfy the conservation relation implied by the governing equations, and may exhibit non-physical instabilities.

It is shown in this chapter that the ROM instability problem described above can be remedied by designing a special energy inner product in which to do the Galerkin projection step of the model reduction procedure for compressible flow problems.

In Section 5.1, an energy-stability preserving model reduction approach is developed for conservation laws of the hyperbolic and incompletely parabolic type. It is shown that a certain transformation applied to a generic hyperbolic or incompletely parabolic set of PDEs and induced by the Lyapunov function for these equations will yield a Galerkin ROM that is stable for any choice of reduced basis. It is shown in Section 5.2 that, for linear PDEs, the desired transformation is induced by a special weighted L^2 inner product, termed the “symmetry inner product”. The symmetry inner product is given for several systems of physical interest: the linearized compressible Euler equations, the linearized compressible Navier-Stokes equations, the wave equation, and the linearized shallow water equations (Sections 5.2.1–5.2.3).

In Section 5.3, attention is turned to developing energy-stable ROMs for the nonlinear compressible Navier-Stokes equations. Three forms of these equations are considered:

- The standard form of the compressible Navier-Stokes equations in the primitive variables, density, velocity and temperature (Section 5.3.1).
- The so-called ζ–form of the compressible Navier-Stokes equations, which uses specific vol-

ume instead of density and pressure instead of temperature, and has a lower degree nonlinearity than the standard form (Section 5.3.2).

- The nonlinear compressible isentropic Navier-Stokes equations in the velocity and enthalpy variables, valid for cold flows (Section 5.3.3).

An energy-based inner product for the third form of the equations of interest (the nonlinear compressible isentropic Navier-Stokes equations) was introduced by Rowley *et al.* in [81]. This inner product induces a meaningful physical quantity for these equations, namely the integrated stagnation energy or stagnation enthalpy, and is reviewed here as it motivates the formulation of a *new* transformation and corresponding energy inner product for the full compressible Navier-Stokes equations in the standard primitive variable form. The norm associated with this inner product induces a conserved quantity for a given flow: the total energy. If the projection step of the model reduction is performed in the total energy inner product, the resulting Galerkin ROM should preserve the stability of an equilibrium point at the origin for the compressible Navier-Stokes system.

Following a discussion of the compressible flow ROM code (known as `Spirit`) developed as a part of this project (Section 5.4), the performance of the various ROM approaches described in this chapter is evaluated on several test cases in Section 5.5, including a viscous laminar cavity problem. The chapter ends with a summary of some key observations and "lessons learned" from the various numerical experiments performed. Also included is a discussion of prospects for future work.

More detail on the content described in this chapter can be found in the following journal articles and SAND reports, written during the time of this LDRD project: [59, 55, 35, 18, 60, 61].

5.1 A stabilizing transformation for conservation laws

In this section, an approach for building energy-stable ROMs via continuous Galerkin projection is developed for PDE systems of the form:

$$\dot{\mathbf{q}} + \mathbf{A}_i \frac{\partial \mathbf{q}}{\partial x_i} - \mathbf{K}_{ij} \frac{\partial^2 \mathbf{q}}{\partial x_i \partial x_j} + \mathbf{G}\mathbf{q} = \mathbf{f}. \tag{5.1}$$

In (5.1), $\mathbf{q} \in \mathbb{R}^n$ denotes a vector of unknowns, $\mathbf{f} \in \mathbb{R}^n$ is a source term, $\mathbf{A}_i$, $\mathbf{K}_{ij}$ and $\mathbf{G}$ are $n \times n$ matrices, where $1 \leq i, j \leq d$, with d denoting the number of spatial dimensions, and $n \in \mathbb{N}$. The matrices $\mathbf{A}_i$, $\mathbf{K}_{ij}$ and $\mathbf{G}$ could be a function of space, but they are assumed to be steady (not a function of time t). The so-called Einstein notation (implied summation on repeated indices) has been employed in (5.1) and subsequent expressions. Most conservation laws, as well as many PDEs of physical interest, can be written in the form (5.1). If $\mathbf{K}_{ij} = \mathbf{0}\ \forall i, j$, (5.1) is known as a hyperbolic system [46]. Otherwise, if $\mathbf{K}_{ij} \neq \mathbf{0}$, (5.1) is known as an incompletely parabolic system [46].

Suppose there exists a transformation

$$\begin{array}{rccc} T: & \mathbb{R}^n & \to & \mathbb{R}^n, \\ & \mathbf{q} & \to & \mathbf{v}, \end{array} \tag{5.2}$$

such that in the new variables $\mathbf{v}$, the system (5.1) has the form

$$\dot{\mathbf{v}} + \mathbf{A}_i^S \frac{\partial \mathbf{v}}{\partial x_i} - \mathbf{K}_{ij}^S \frac{\partial^2 \mathbf{v}}{\partial x_i \partial x_j} + \mathbf{G}^S \mathbf{v} = \mathbf{f}^S, \tag{5.3}$$

where:

- *Property 1:* The matrices $\mathbf{A}_i^S$ are symmetric for all $1 \leq i \leq d$.
- *Property 2:* The matrices $\mathbf{K}_{ij}^S$ are symmetric for all $1 \leq i, j \leq d$.
- *Property 3:* The augmented viscosity matrix:

$$\mathbf{K}^S \equiv \begin{pmatrix} \mathbf{K}_{11}^S & \dots & \mathbf{K}_{1d}^S \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{d1}^S & \dots & \mathbf{K}_{dd}^S \end{pmatrix} \tag{5.4}$$

  is positive semi-definite.

Theorem 5.1.1: Suppose a ROM for (5.3) is constructed using continuous Galerkin projection in the $L^2(\Omega)$ inner product. Suppose the matrices in (5.3) satisfy *Properties 1–3* above. Suppose also that the reduced basis modes satisfy the boundary conditions of the full order system, or they are implemented weakly in the ROM in a stability-preserving way[1]. Let $\mathbf{v}_M$ denote the ROM solution to (5.3). Then the ROM is energy-stable with energy estimate

$$||\mathbf{v}_M(\cdot, T)||_2 \leq e^{\frac{1}{2}\beta_S T} ||\mathbf{v}_M(\cdot, 0)||_2, \tag{5.5}$$

where β_S is an upper bound on the eigenvalues of the matrix

$$\mathbf{B}^S \equiv \frac{\partial \mathbf{A}_i^S}{\partial x_i} + \frac{\partial^2 \mathbf{K}_{ij}^S}{\partial x_i \partial x_j} - 2\mathbf{G}^S. \tag{5.6}$$

Moreover, this energy-stability result holds for *any* choice of reduced basis.

Proof. To prove energy-stability of a ROM constructed for (5.3), it is necessary to bound the energy of the ROM solution to (5.3) with $\mathbf{f}^S = \mathbf{0}$:

$$\begin{array}{rl} \frac{dE_M}{dt} & = \frac{1}{2}\frac{d}{dt}||\mathbf{v}_M||_2^2 \\ & = \frac{1}{2}\frac{d}{dt}(\mathbf{v}_M, \mathbf{v}_M) \\ & = \left(\mathbf{v}_M, \frac{\partial \mathbf{v}_M}{\partial t}\right) \\ & = \left(\mathbf{v}_M, -\mathbf{A}_i^S \frac{\partial \mathbf{v}_M}{\partial x_i} + \mathbf{K}_{ij}^S \frac{\partial^2 \mathbf{v}_M}{\partial x_i \partial x_j} - \mathbf{G}^S \mathbf{v}_M\right) \\ & = -\int_\Omega \mathbf{v}_M^T \mathbf{A}_i^S \frac{\partial \mathbf{v}_M}{\partial x_i} \partial\Omega + \int_\Omega \mathbf{v}_M^T \mathbf{K}_{ij}^S \frac{\partial^2 \mathbf{v}_M}{\partial x_i \partial x_j} \partial\Omega - \int_\Omega \mathbf{v}_M^T \mathbf{G}^S \mathbf{v}_M \partial\Omega. \end{array} \tag{5.7}$$

[1] The reader is referred to [57] and Appendix A.5 for a discussion of stability-preserving weak implementations of boundary conditions for ROMs constructed using the continuous projection approach. In general, a weak implementation of boundary conditions will be stability-preserving provided the boundary conditions are well-posed.

Each of the terms in (5.7) will be bounded separately. First,

$$\begin{aligned} -\int_\Omega \mathbf{v}_M^T \mathbf{A}_i^S \frac{\partial \mathbf{v}_M}{\partial x_i}\partial\Omega &= -\frac{1}{2}\int_\Omega \frac{\partial}{\partial x_i}\left(\mathbf{v}_M^T\mathbf{A}_i^S\mathbf{v}_M\right)d\Omega + \frac{1}{2}\int_\Omega \mathbf{v}_M^T\frac{\partial \mathbf{A}_i^S}{\partial x_i}\mathbf{v}_M d\Omega \\ &= -\frac{1}{2}\int_{\partial\Omega} \mathbf{v}_M^T\mathbf{A}_i^S n_i \mathbf{v}_M d\Gamma + \frac{1}{2}\int_\Omega \mathbf{v}_M^T\frac{\partial \mathbf{A}_i^S}{\partial x_i}\mathbf{v}_M d\Omega. \end{aligned} \tag{5.8}$$

In (5.8), the property that each of the matrices $\mathbf{A}_i^S$ is symmetric has been employed (*Property 1*). The symbol Γ has been used to denote the boundary of Ω, $\partial\Omega$.

Next, note that:

$$\mathbf{K}_{ij}^S \frac{\partial^2 \mathbf{v}_M}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}^S \frac{\partial \mathbf{v}_M}{\partial x_j}\right) - \left(\frac{\partial \mathbf{K}_{ij}^S}{\partial x_i}\frac{\partial \mathbf{v}_M}{\partial x_j}\right). \tag{5.9}$$

Then,

$$\int_\Omega \mathbf{v}_M^T \mathbf{K}_{ij}^S \frac{\partial^2 \mathbf{v}_M}{\partial x_i \partial x_j}\partial\Omega = \int_\Omega \mathbf{v}_M^T \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}^S \frac{\partial \mathbf{v}_M}{\partial x_j}\right) d\Omega - \int_\Omega \mathbf{v}_M^T \frac{\partial \mathbf{K}_{ij}^S}{\partial x_i}\frac{\partial \mathbf{v}_M}{\partial x_j}\partial\Omega. \tag{5.10}$$

Again, each of the two terms in (5.10) will be bounded separately.

$$\begin{aligned} \int_\Omega \mathbf{v}_M^T \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}^S \frac{\partial \mathbf{v}_M}{\partial x_j}\right) d\Omega &= -\int_\Omega \frac{\partial \mathbf{v}_M}{\partial x_i}^T \mathbf{K}_{ij}^S \frac{\partial \mathbf{v}_M}{\partial x_j} d\Omega + \int_{\partial\Omega} \mathbf{v}_M^T \mathbf{K}_{ij}^S \frac{\partial \mathbf{v}_M}{\partial x_j} n_i d\Gamma \\ &\le \int_{\partial\Omega} \mathbf{v}_M^T \mathbf{K}_{ij}^S \frac{\partial \mathbf{v}_M}{\partial x_j} n_i d\Gamma, \end{aligned} \tag{5.11}$$

provided the matrix (5.4) is positive semi-definite (*Property 3*).

Now for the second term in (5.10):

$$\begin{aligned} -\int_\Omega \mathbf{v}_M^T \frac{\partial \mathbf{K}_{ij}^S}{\partial x_i}\frac{\partial \mathbf{v}_M}{\partial x_j}\partial\Omega &= -\frac{1}{2}\int_\Omega \frac{\partial}{\partial x_j}\left(\mathbf{v}_M^T \frac{\partial \mathbf{K}_{ij}^S}{\partial x_i}\mathbf{v}_M\right) d\Omega + \frac{1}{2}\int_\Omega \mathbf{v}_M^T \frac{\partial^2 \mathbf{K}_{ij}^S}{\partial x_i \partial x_j}\mathbf{v}_M d\Omega \\ &= -\frac{1}{2}\int_{\partial\Omega} \mathbf{v}_M^T \frac{\partial \mathbf{K}_{ij}^S}{\partial x_i} n_j \mathbf{v}_M d\Gamma + \frac{1}{2}\int_\Omega \mathbf{v}_M^T \frac{\partial^2 \mathbf{K}_{ij}^S}{\partial x_i \partial x_j}\mathbf{v}_M d\Omega. \end{aligned} \tag{5.12}$$

In (5.12), the property that the $\mathbf{K}_{ij}^S$ matrices and therefore their derivatives are symmetric has been employed (*Property 2*).

Finally, (5.8) and (5.10) are substituted into (5.7). The boundary integral terms may be neglected if the reduced basis modes satisfy the boundary conditions or the boundary conditions have been implemented in a stability-preserving way. The following bound is obtained:

$$\begin{aligned} \frac{1}{2}\frac{d}{dt}||\mathbf{v}_M||_2^2 &\le \frac{1}{2}\int_\Omega \mathbf{v}_M^T \left(\frac{\partial \mathbf{A}_i^S}{\partial x_i}\right)\mathbf{v}_M d\Omega + \frac{1}{2}\int_\Omega \mathbf{v}_M^T \frac{\partial^2 \mathbf{K}_{ij}^S}{\partial x_i \partial x_j}\mathbf{v}_M d\Omega - \int_\Omega \mathbf{v}_M^T \mathbf{G}^S \mathbf{v}_M \partial\Omega \\ &= \frac{1}{2}\int_\Omega \mathbf{v}_M^T \mathbf{B}^S \mathbf{v}_M d\Omega, \end{aligned} \tag{5.13}$$

where $\mathbf{B}^S$ is given by (5.6). Applying Gronwall's inequality ((A.1) in Appendix A.1) to (5.13), it is found that:

$$||\mathbf{v}_M(\cdot,T)||_2 \le e^{\frac{1}{2}\beta_S T}||\mathbf{v}_M(\cdot,0)||_2, \tag{5.14}$$

where β_S is an upper bound on the eigenvalues of the matrix $\mathbf{B}^S$ (5.6).

□

The proof of Theorem 5.1.1 is one of the new contributions of this report.

Note that, if $\mathbf{G} = \mathbf{0}$ in (5.1) and the $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ matrices are spatially-constant, it follows that $\beta_S = 0$ in (5.14). In this case, if the ROM for (5.1) is constructed in the variables $\mathbf{v}$, the ROM will be time-stable as well as stable in the sense of Lyapunov, in addition to being energy-stable. For linearized conservation laws (e.g., the linearized shallow water equations, the linearized compressible Euler equations, the linearized compressible Navier-Stokes equations), the property that $\mathbf{G} = \mathbf{0}$ and the $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ are spatially-constant will in general hold if the base flow is spatially uniform.

5.2 Stability-preserving "symmetry inner product" for linear conservation laws

A key property of systems of the form (5.1) is that they are symmetrizable [47, 20, 57]; that is, it is possible to derive a symmetric positive-definite matrix $\mathbf{H}$ such that:

- *Property 1*:* The matrices $\mathbf{HA}_i$ are symmetric for all $1 \leq i \leq d$.
- *Property 2*:* The matrices $\mathbf{HK}_{ij}$ are symmetric for all $1 \leq i, j \leq d$.
- *Property 3*:* The augmented viscosity matrix:

$$\mathbf{K}^H \equiv \begin{pmatrix} \mathbf{HK}_{11} & \dots & \mathbf{HK}_{1d} \\ \vdots & \ddots & \vdots \\ \mathbf{HK}_{d1} & \dots & \mathbf{HK}_{dd} \end{pmatrix} \tag{5.15}$$

  is positive semi-definite.

Since $\mathbf{H}$ is symmetric positive-definite, the following defines a valid inner product:

$$\left(\mathbf{q}^{(1)}, \mathbf{q}^{(2)}\right)_{(\mathbf{H},\Omega)} \equiv \int_{\Omega} \mathbf{q}^{(1)T} \mathbf{H} \mathbf{q}^{(2)} d\Omega. \tag{5.16}$$

Following the terminology introduced in [20, 57], the inner product (5.16) will be referred to as the "symmetry inner product". It is straightforward to see that the following corollary to Theorem 5.1.1 holds.

Corollary 5.2.1: Suppose a ROM for (5.1) is constructed using continuous Galerkin projection in the symmetry inner product (5.16). Suppose *Properties 1*-3** hold. Suppose also, as in Theorem 5.1.1, that the reduced basis modes satisfy the boundary conditions of the full order system, or they are implemented weakly in the ROM in a stability-preserving way. Let $\mathbf{q}_M$ denote the ROM solution to (5.1). Then the ROM is energy-stable with energy estimate

$$||\mathbf{q}_M(\cdot, T)||_{(\mathbf{H},\Omega)} \leq e^{\frac{1}{2}\beta_H T} ||\mathbf{q}_M(\cdot, 0)||_{(\mathbf{H},\Omega)}, \tag{5.17}$$

where β_H is an upper bound on the eigenvalues of the matrix

$$\mathbf{B}^H \equiv \frac{\partial(\mathbf{HA}_i)}{\partial x_i} + \frac{\partial^2(\mathbf{HK}_{ij})}{\partial x_i \partial x_j} - 2\mathbf{HG}. \tag{5.18}$$

Moreover, this energy-stability result holds for *any* choice of reduced basis.

Proof. Because of simple linear transformations, the proof is analogous to the proof of Theorem 5.1.1.

□

Again, in the case that $\mathbf{G} = \mathbf{0}$ and the $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ matrices are spatially-constant, it will follow from Corollary 5.2.1 that a ROM constructed in the symmetry inner product (5.16) will be time-stable and stable in the sense of Lyapunov, in addition to being energy-stable.

It is interesting to observe that a Galerkin projection of the governing (5.1) in the symmetry inner product (5.16) is equivalent to a Petrov-Galerkin projection. Let $\boldsymbol{\phi}_i$ for $i = 1,...,M$ denote the reduced trial basis vector for the solution $\mathbf{q}$. Performing a Galerkin projection of the equations (5.1) onto the modes $\boldsymbol{\phi}_k$ gives

$$\int_\Omega \boldsymbol{\phi}_k^T \mathbf{H} \left(\dot{\mathbf{q}} + \mathbf{A}_i \frac{\partial \mathbf{q}}{\partial x_i} + \mathbf{K}_{ij} \frac{\partial^2 \mathbf{q}}{\partial x_i \partial x_j} + \mathbf{G}\mathbf{q} \right) d\Omega = \int_\Omega \boldsymbol{\phi}_k^T \mathbf{H}\mathbf{f} d\Omega, \tag{5.19}$$

for $k = 1,...,M$. Equation (5.19) is equivalent to a Petrov-Galerkin projection of the equations (5.1) in the regular L^2 inner product

$$\int_\Omega \boldsymbol{\psi}_k^T \left(\dot{\mathbf{q}} + \mathbf{A}_i \frac{\partial \mathbf{q}}{\partial x_i} + \mathbf{K}_{ij} \frac{\partial^2 \mathbf{q}}{\partial x_i \partial x_j} + \mathbf{G}\mathbf{q} \right) d\Omega = \int_\Omega \boldsymbol{\psi}_k^T \mathbf{f} d\Omega, \tag{5.20}$$

where the reduced test basis functions are given by $\boldsymbol{\psi}_k = \mathbf{H}\boldsymbol{\phi}_k$, for all $k = 1,...,M$.

5.2.1 Application to linearized compressible Euler equations

Consider the linearized compressible Euler equations. These equations may be used if a compressible fluid system can be described by inviscid, small-amplitude perturbations about a steady-state mean flow. The equations are obtained from the full (nonlinear) compressible Euler equations by decomposing the fluid vector $\mathbf{q}(\mathbf{x},t)$ into a steady mean plus an unsteady fluctuation, i.e.,

$$\mathbf{q}(\mathbf{x},t) = \bar{\mathbf{q}}(\mathbf{x}) + \mathbf{q}'(\mathbf{x},t), \tag{5.21}$$

and linearizing these equations around the steady mean state $\bar{\mathbf{q}}$. If $\mathbf{q}^T = \begin{pmatrix} u_1, & u_2, & u_3, & \zeta, & p \end{pmatrix}$, where u_1, u_2 and u_3 are the three components of the velocity vector, ζ is the specific volume (the reciprocal of the density), and p is the pressure. The linearized compressible Euler equations take the form

$$\dot{\mathbf{q}} + \mathbf{A}_i \frac{\partial \mathbf{q}'}{\partial x_i} + \mathbf{G}\mathbf{q}' = \mathbf{0}. \tag{5.22}$$

In 3D, the convective flux matrices $\mathbf{A}_i$ in the linearized compressible Euler hyperbolic system (5.22) are given by:

$$\mathbf{A}_1 = \begin{pmatrix} \bar{u}_1 & 0 & 0 & 0 & \bar{\zeta} \\ 0 & \bar{u}_1 & 0 & 0 & 0 \\ 0 & 0 & \bar{u}_1 & 0 & 0 \\ -\bar{\zeta} & 0 & 0 & \bar{u}_1 & 0 \\ \gamma\bar{p} & 0 & 0 & 0 & \bar{u}_1 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} \bar{u}_2 & 0 & 0 & 0 & 0 \\ 0 & \bar{u}_2 & 0 & 0 & \bar{\zeta} \\ 0 & 0 & \bar{u}_2 & 0 & 0 \\ 0 & -\bar{\zeta} & 0 & \bar{u}_2 & 0 \\ 0 & \gamma\bar{p} & 0 & 0 & \bar{u}_2 \end{pmatrix}, \tag{5.23}$$
$$\mathbf{A}_3 = \begin{pmatrix} \bar{u}_3 & 0 & 0 & 0 & 0 \\ 0 & \bar{u}_3 & 0 & 0 & 0 \\ 0 & 0 & \bar{u}_3 & 0 & \bar{\zeta} \\ 0 & 0 & -\bar{\zeta} & \bar{u}_3 & 0 \\ 0 & 0 & \gamma\bar{p} & 0 & \bar{u}_3 \end{pmatrix}.$$

The matrix $\mathbf{G}$ in (5.22) is a function of the gradients of the base flow:

$$\mathbf{G} = \begin{pmatrix} \bar{u}_{1,1} & \bar{u}_{1,2} & \bar{u}_{1,3} & \bar{p}_{,1} & 0 \\ \bar{u}_{2,1} & \bar{u}_{2,2} & \bar{u}_{2,3} & \bar{p}_{,2} & 0 \\ \bar{u}_{3,1} & \bar{u}_{3,2} & \bar{u}_{3,3} & \bar{p}_{,3} & 0 \\ \bar{\zeta}_{,1} & \bar{\zeta}_{,2} & \bar{\zeta}_{,3} & -\bar{u}_{j,j} & 0 \\ \bar{p}_{,1} & \bar{p}_{,2} & \bar{p}_{,3} & \gamma\bar{u}_{j,j} & 0 \end{pmatrix}. \tag{5.24}$$

In the above matrices, $\gamma = C_P/C_V$ is the ratio of specific heats. The reader may verify that if the linearized compressible Euler system (5.22) is pre-multiplied by the following symmetric positive definite matrix:

$$\mathbf{H} = \begin{pmatrix} \bar{\rho} & 0 & 0 & 0 & 0 \\ 0 & \bar{\rho} & 0 & 0 & 0 \\ 0 & 0 & \bar{\rho} & 0 & 0 \\ 0 & 0 & 0 & \alpha^2\gamma\bar{\rho}^2\bar{p} & \bar{\rho}\alpha^2 \\ 0 & 0 & 0 & \bar{\rho}\alpha^2 & \frac{(1+\alpha^2)}{\gamma\bar{p}} \end{pmatrix}, \tag{5.25}$$

where α is a real, non-zero parameter, the convective flux matrices $\mathbf{HA}_i$ are all symmetric [20, 57].

5.2.2 Application to linearized compressible Navier-Stokes equations

Consider the 3D linearized compressible Navier-Stokes equations. These equations are appropriate when a compressible fluid system can be described by viscous, small-amplitude perturbations about a steady-state base flow. As with the linearized compressible Euler equations, to derive these equations from the full (nonlinear) compressible Navier-Stokes equations, the fluid vector $\mathbf{q}(\mathbf{x},t)$ is written as the sum of a steady mean plus an unsteady fluctuation (5.21), and a linearization around the steady mean is performed. If the viscous work terms are neglected from the equations[2] (appropriate, for example, in a low Mach number regime), the result is a linear incompletely parabolic

[2] To the authors' knowledge, the viscous work terms are invariably neglected from the linearized compressible Navier-Stokes equations by researchers studying energy-stability of these equations [47, 4]. The omission of these terms is justified only in the low Mach number regime, or in the case that the base flow is uniform. The extension of

system of the form

$$\dot{\mathbf{q}}' + \mathbf{A}_i \frac{\partial \mathbf{q}'}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}\frac{\partial \mathbf{q}}{\partial x_j}\right) + \mathbf{G}\mathbf{q}' = \mathbf{0}. \tag{5.26}$$

If the fluid vector is given by $\mathbf{q}^T = \begin{pmatrix} u_1, & u_2, & u_3, & T, & \rho \end{pmatrix}$, where T and ρ denote the fluid temperature and density respectively, the matrices that appear in (5.26) are given by the expressions found in [47], repeated below to keep this document self-contained

$$\mathbf{A}_1 \equiv \begin{pmatrix} \bar{u}_1 & 0 & 0 & R & \frac{R\bar{T}}{\rho} \\ 0 & \bar{u}_1 & 0 & 0 & 0 \\ 0 & 0 & \bar{u}_1 & 0 & 0 \\ \bar{T}(\gamma-1) & 0 & 0 & \bar{u}_1 & 0 \\ \bar{\rho} & 0 & 0 & 0 & \bar{u}_1 \end{pmatrix}, \quad \mathbf{A}_2 \equiv \begin{pmatrix} \bar{u}_2 & 0 & 0 & 0 & 0 \\ 0 & \bar{u}_2 & 0 & R & \frac{R\bar{T}}{\rho} \\ 0 & 0 & \bar{u}_2 & 0 & 0 \\ 0 & \bar{T}(\gamma-1) & 0 & \bar{u}_2 & 0 \\ 0 & \bar{\rho} & 0 & 0 & \bar{u}_2 \end{pmatrix},$$
$$\mathbf{A}_3 \equiv \begin{pmatrix} \bar{u}_3 & 0 & 0 & 0 & 0 \\ 0 & \bar{u}_3 & 0 & 0 & 0 \\ 0 & 0 & \bar{u}_3 & R & \frac{R\bar{T}}{\rho} \\ 0 & 0 & \bar{T}(\gamma-1) & \bar{u}_3 & 0 \\ 0 & 0 & \bar{\rho} & 0 & \bar{u}_3 \end{pmatrix}, \tag{5.27}$$

$$\mathbf{G} = \begin{pmatrix} \bar{u}_{1,1} & \bar{u}_{1,2} & \bar{u}_{1,3} & \frac{R}{\rho}\bar{\rho}_{,1} & \frac{1}{\rho}\left(\bar{u}_i\bar{u}_{1,i} + R\bar{T}_{,1}\right) \\ \bar{u}_{2,1} & \bar{u}_{2,2} & \bar{u}_{2,3} & \frac{R}{\rho}\bar{\rho}_{,2} & \frac{1}{\rho}\left(\bar{u}_i\bar{u}_{2,i} + R\bar{T}_{,2}\right) \\ \bar{u}_{3,1} & \bar{u}_{3,2} & \bar{u}_{3,3} & \frac{R}{\rho}\bar{\rho}_{,3} & \frac{1}{\rho}\left(\bar{u}_i\bar{u}_{3,i} + R\bar{T}_{,3}\right) \\ \bar{T}_{,1} & \bar{T}_{,2} & \bar{T}_{,3} & (\gamma-1)\bar{u}_{i,i} & \frac{1}{\rho}\left(\bar{u}_i\bar{T}_{,i} + (\gamma-1)\bar{T}\bar{u}_{i,i}\right) \\ \bar{\rho}_{,1} & \bar{\rho}_{,2} & \bar{\rho}_{,3} & 0 & \bar{u}_{i,i} \end{pmatrix}, \tag{5.28}$$

and

$$\mathbf{K}_{11} \equiv \frac{1}{\rho Re}\begin{pmatrix} 2\mu+\lambda & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \frac{\gamma\kappa}{Pr} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}_{12} \equiv \frac{1}{\rho Re}\begin{pmatrix} 0 & \lambda & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{5.29}$$

$$\mathbf{K}_{13} \equiv \frac{1}{\rho Re}\begin{pmatrix} 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}_{21} \equiv \frac{1}{\rho Re}\begin{pmatrix} 0 & \mu & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{5.30}$$

$$\mathbf{K}_{22} \equiv \frac{1}{\bar{\rho} Re}\begin{pmatrix} \mu & 0 & 0 & 0 & 0 \\ 0 & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \frac{\gamma\kappa}{Pr} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}_{23} \equiv \frac{1}{\bar{\rho} Re}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{5.31}$$

the energy-stability symmetrization approach presented here to the linearized compressible Navier-Stokes equations in which the viscous work terms are retained is the subject of present research. The linearized compressible Navier-Stokes equations with the viscous work terms retained are given in Appendix A.2.

$$\mathbf{K}_{31} \equiv \frac{1}{\bar{\rho}Re}\begin{pmatrix} 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}_{32} \equiv \frac{1}{\bar{\rho}Re}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{5.32}$$

$$\mathbf{K}_{33} \equiv \frac{1}{\bar{\rho}Re}\begin{pmatrix} \mu & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 2\mu+\lambda & 0 & 0 \\ 0 & 0 & 0 & \frac{\gamma\kappa}{Pr} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.33}$$

The parameters[3] appearing in the viscous stress matrices $\mathbf{K}_{ij}$ are: the Lamé viscosity coefficients λ and μ, the thermal diffusivity κ, the Prandtl number Pr, and the Reynolds number Re. The reader can verify that if the system (5.1) is pre-multiplied by the symmetric positive definite matrix given by

$$\mathbf{H} \equiv \begin{pmatrix} \bar{\rho} & 0 & 0 & 0 & 0 \\ 0 & \bar{\rho} & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & \frac{\bar{\rho}R}{\bar{T}(\gamma-1)} & 0 \\ 0 & 0 & 0 & 0 & \frac{R\bar{T}}{\bar{\rho}} \end{pmatrix}, \tag{5.34}$$

the "symmetrized" convective flux matrices $\mathbf{HA}_i$ and diffusive flux matrices $\mathbf{HK}_{ij}$ satisfy *Properties 1*–3** in Section 5.2. Here, R denotes the non-dimensional universal gas constant.

Note that the symmetry transformation exhibited above is not unique. For example, in [4], Abarbanel *et al.* exhibit a transformation of the form (5.3) in the linearized compressible Navier-Stokes equations written in the primitive variables $\mathbf{q}^T = \begin{pmatrix} \rho, & u_1, & u_2, & u_3, & p \end{pmatrix}$.

5.2.3 Application to other hyperbolic systems (e.g., shallow water equations, wave equation)

It is straightforward to derive the symmetrizer $\mathbf{H}$ for a number of other physical systems commonly of interest to the scientific community. The symmetrizers for two such systems are given in Table 5.1: the 1D wave equation and the 3D linearized shallow water equations. For the former equation, the wave equation, the original second order PDE $\ddot{u} = a^2 \frac{\partial^2 u}{\partial x^2}$ where $a \in \mathbb{R}$ denotes the wave speed, has been written as a first order system. The latter set of PDEs, namely the 3D linearized shallow water equations, are obtained from the full (nonlinear) shallow water equations by decomposing the fluid vector $\mathbf{q}(\mathbf{x},t)$ into a steady mean plus an unsteady fluctuation (5.21), and linearizing the full shallow water equations around the steady mean state $\bar{\mathbf{q}}$. The variable ϕ denotes the local

[3] Note that Section 5.2.2 give the dimensionless form of the linearized compressible Navier-Stokes equations. The details of the non-dimensionalization of these equations can be found in Appendix A.3. The gas constant R in the convective flux matrices (5.27) is the non-dimensional gas constant, given by (A.7). The dimensional gas constant has the value 8.31 $\frac{\text{J}}{\text{mol K}}$ for air.

height of the fluid above the equilibrium depth, and the variables u_1, u_2, and u_3 are the components of the fluid velocity vector [90].

Table 5.1. Symmetrizer **H** for several PDEs

Name	1D wave equation	3D linearized shallow water equations
Variables	$\mathbf{q}^T = \begin{pmatrix} \dot{u}, & \frac{\partial u}{\partial x} \end{pmatrix}$	$\mathbf{q}^T = \begin{pmatrix} u_1, & u_2, & u_3, & \phi \end{pmatrix}$
PDEs	$\dot{\mathbf{q}} = \mathbf{A}_1 \frac{\partial \mathbf{q}}{\partial x}$	$\dot{\mathbf{q}}' + \mathbf{A}_i \frac{\partial \mathbf{q}'}{\partial x_i} + \mathbf{G}\mathbf{q}' = \mathbf{0}$
$\mathbf{A}_1$	$\begin{pmatrix} 0 & a^2 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} \bar{u}_1 & 0 & 0 & 1 \\ 0 & \bar{u}_1 & 0 & 0 \\ 0 & 0 & \bar{u}_1 & 0 \\ \bar{\phi} & 0 & 0 & \bar{u}_1 \end{pmatrix}$
$\mathbf{A}_2$	–	$\begin{pmatrix} \bar{u}_2 & 0 & 0 & 0 \\ 0 & \bar{u}_2 & 0 & 1 \\ 0 & 0 & \bar{u}_2 & 0 \\ 0 & \bar{\phi} & 0 & \bar{u}_2 \end{pmatrix}$
$\mathbf{A}_3$	–	$\begin{pmatrix} \bar{u}_3 & 0 & 0 & 0 \\ 0 & \bar{u}_3 & 0 & 0 \\ 0 & 0 & \bar{u}_3 & 1 \\ 0 & 0 & \bar{\phi} & \bar{u}_3 \end{pmatrix}$
$\mathbf{H}$	$\begin{pmatrix} 1 & 0 \\ 0 & a^2 \end{pmatrix}$	$\begin{pmatrix} \bar{\phi} & 0 & 0 & 0 \\ 0 & \bar{\phi} & 0 & 0 \\ 0 & 0 & \bar{\phi} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

5.3 Nonlinear conservation laws

Attention is now turned to nonlinear conservation laws, namely the nonlinear compressible Navier-Stokes equations of interest in the targeted captive-carry application. Three forms of the equations are considered: the full compressible Navier-Stokes equations, the ζ–form of the full compressible Navier-Stokes equations, and the isentropic compressible Navier-Stokes equations. The notation '$_{,i}$' denotes differentiation with respect to the i^{th} spatial variable, i.e., $a_{,i} \equiv \frac{\partial a}{\partial x_i}$, and '$_{,t}$' denotes differentiation with respect to t, i.e., $a_{,t} \equiv \frac{\partial a}{\partial t}$, for a given variable $a \equiv a(\mathbf{x},t)$. As before, the Einstein convention (implied summation on repeated indices) has been employed.

5.3.1 Full compressible Navier-Stokes equations

The full compressible Navier-Stokes equations in dimensionless primitive variable form are given by the following system of PDEs [31]:

$$\begin{aligned} \rho_{,t} + \rho_{,j} u_j + \rho u_{j,j} &= 0, \\ \rho u_{i,t} + \rho u_{i,j} u_j + \tfrac{1}{\gamma M_{ref}^2} (\rho T)_{,i} - \tfrac{1}{Re} \tau_{ij,j} &= 0, \\ \rho T_{,t} + \rho u_j T_{,j} + (\gamma - 1) u_{j,j} \rho T - \left(\tfrac{\gamma}{PrRe}\right) \left(\kappa T_{,j}\right)_{,j} - \left(\frac{\gamma(\gamma-1)M_{ref}^2}{Re}\right) u_{i,j} \tau_{ij} &= 0, \end{aligned} \tag{5.35}$$

where $i, j = 1, 2, 3$ (in three spatial dimensions) and there is an implied summation on repeated indices. The first equation in (5.35) is the continuity equation; the second three equations are the momentum equations; the final equation is the energy equation. There are five unknowns: the density ρ, the three velocity components u_1, u_2 and u_3, and the temperature T. Re and Pr denote the Reynolds and Prandtl numbers respectively; γ denotes the ratio of specific heats; M_{ref} is the reference Mach number used in the non-dimensionalization[4]. The symbol τ_{ij} denotes the stress tensor, given by:

$$\tau_{ij} = \mu \left(u_{i,j} + u_{j,i}\right) + \lambda u_{k,k} \delta_{ij}, \tag{5.36}$$

for $i, j, k = 1, 2, 3$, where δ_{ik} denotes the Kronëcker delta, and μ and λ are the Lamé coefficients, typically assumed to satisfy the so-called Stokes' relation

$$\lambda + \frac{2}{3}\mu = 0. \tag{5.37}$$

The symbol κ denotes the thermal diffusivity, given by

$$\kappa = \frac{\mu c_p}{Pr}, \tag{5.38}$$

where c_p denotes the specific heat of the fluid at constant pressure. In deriving (5.35), the ideal gas law

$$p = \rho R T = \frac{\rho T}{\gamma M_{ref}^2}, \tag{5.39}$$

where p denotes the fluid pressure and R is the universal gas constant, has been employed. Assuming constant viscosities and diffusivities (see *Remark 5.3.1* below), the equations (5.35) have a cubic nonlinearity.

Remark 5.3.1: In general, the viscosity coefficients μ and λ, and the thermal diffusivity κ need not be spatially constant. A commonly used expression for μ is the Sutherland viscosity law [89], which is based on the kinetic theory of ideal gases and an idealized intermolecular-force potential. This law states that μ is related to the temperature T as follows:

$$\mu = \mu(T) = \frac{C_1 T^{3/2}}{T + C_2}, \tag{5.40}$$

[4] If u_{ref} denotes the reference value of the velocities, and c_{ref} is the reference speed of sound, the reference Mach number is given by $M_{ref} = u_{ref}/c_{ref}$. For a detailed discussion of the non-dimensionalization of the compressible Navier-Stokes equations, the reader is referred to Appendix A.3.

for some constants $C_1, C_2 \in \mathbb{R}$. The ROM code written as a part of this LDRD project, known as `Spirit` (Section 5.4), allows the specification of a linearized version of Sutherland's viscosity law (5.40) for the linearized compressible Euler and Navier-Stokes equations, but not for the nonlinear variants of these equations, for which a constant viscosity is assumed. The capability to use a viscosity of the form (5.40) may be added to the code in the future. The primary difficulty of using (5.40) comes from the fact that this expression contains a non-polynomial nonlinearity, meaning the projection of this term would need to be handled using some kind of interpolation to keep the ROM efficient, e.g., the discrete empirical interpolation method (DEIM) [29], "best points" interpolation [74, 75], or gappy POD [33].

5.3.2 Full compressible Navier-Stokes equations: ζ–form

The equations (5.35) can also be written in the so-called ζ–form, where $\zeta = 1/\rho$ denotes the fluid's specific volume:

$$\begin{array}{rcl} \zeta_{,t} + \zeta_{,j} u_j - \zeta u_{j,j} & = & 0, \\ u_{i,t} + u_{i,j} u_j + \zeta p_{,i} - \frac{1}{Re} \zeta \tau_{ij,j} & = & 0, \\ p_{,t} + u_j p_{,j} + \gamma u_{j,j} p - \left(\frac{\gamma}{PrRe}\right) \left(\kappa (p\zeta)_{,j}\right)_{,j} - \left(\frac{\gamma-1}{Re}\right) u_{i,j} \tau_{ij} & = & 0, \end{array} \tag{5.41}$$

for $i, j = 1, 2, 3$ (in three spatial dimensions). The first equation in (5.41) is the continuity equation, followed by the three momentum equations, and the energy equation. The symbol p denotes the fluid pressure. The stress tensor τ_{ij} is given by (5.36), and the symbols *Pr*, *Re*, μ, λ, κ and γ are the same as before.

The upshot of the ζ–form of the compressible Navier-Stokes equations (5.41) over the standard form (5.35) is (5.41) has only a quadratic nonlinearity, whereas (5.35) has a cubic nonlinearity. Hence, both the offline and online stages of a projection-based model reduction algorithm for (5.41) will be cheaper to evaluate than the offline and online stages of a projection-based model reduction algorithm for (5.35).

5.3.3 Isentropic compressible Navier-Stokes equations

The last version of the compressible Navier-Stokes equations considered is an approximate form of (5.35) and (5.41), namely the isentropic version of these equations. The isentropic compressible Navier-Stokes equations are valid for cold flows (flows for which $T_{wall} = T_\infty$ and the temperature gradients are small) at a moderate Mach number. The isentropic assumption is consistent with the neglect of the viscous dissipation and heat conduction terms in the energy equation, and constant viscosities in the momentum equations. Unlike the full equations (5.35) and (5.41), the fluid vector consists of only four unknowns in 3D: the enthalpy h, and the three components of the velocity vector, u_i for $i = 1, 2, 3$. In dimensionless form[5], the isentropic compressible Navier-

[5] The dimensional version of (5.42) can be found in [81]. The reference values used in the non-dimensionalization are given in Appendix A.3.

Stokes equations are as follows:

$$\begin{array}{rcl} h_{,t}+u_ih_{,i}+(\gamma-1)hu_{i,i} & = & 0, \\ u_{j,t}+u_iu_{j,i}+h_j-\frac{1}{Re}u_{j,ii} & = & 0, \end{array} \tag{5.42}$$

for $j=1,2,3$. Like the ζ–form of the full compressible Navier-Stokes equations (5.41) but unlike the original form (5.35), the isentropic Navier-Stokes equations (5.42) have only quadratic nonlinearities. The enthalpy h is related to the speed of sound according to the following relation:

$$c^2=(\gamma-1)h. \tag{5.43}$$

Since $c^2=\frac{\gamma p}{\rho}$, the following equation relates h to p and ρ:

$$h=\frac{\gamma p}{\rho(\gamma-1)}. \tag{5.44}$$

5.3.4 Energy inner products for the nonlinear compressible Navier-Stokes equations

Some energy inner products for the various forms of the nonlinear compressible Navier-Stokes equations given above are now derived. An essential property of the solution to these fluid equations is that the total system energy,

$$E_T=\int_\Omega\left(\rho e+\frac{1}{2}\rho u_iu_i\right)d\Omega, \tag{5.45}$$

is, in the absence of external sources, non-increasing in time[6], i.e.,

$$\frac{dE_T}{dt}\leq 0. \tag{5.46}$$

The first term in (5.45) represents the internal energy, and the second term represents the kinetic energy, with e denoting the internal energy per unit mass of the fluid. The quantity e is related to the pressure, density and temperature (assuming as before the ideal gas law (5.39)) via the relation:

$$p=(\gamma-1)\rho e, \quad T=\gamma(\gamma-1)M_{ref}^2e. \tag{5.47}$$

Since $E_T>0$ and $\frac{dE_T}{dt}\leq 0$, the energy (5.45) satisfies the definition of a Lyapunov function for its corresponding fluid system (Theorem 3.2.2 in Section 3.2). By the discussion in Section 3.2 as well as [81], if an inner product that induces the energy (5.45) is used to build the ROM, then the stability of an equilibrium point of the system at the origin should be preserved by the Galerkin projection step of the model reduction. This is a necessary, although not always sufficient, condition for time-stability of a nonlinear ROM.

[6] Or, equivalently, the entropy is non-decreasing in time, per the second law of thermodynamics, a.k.a. the Clausius-Duhem inequality [55].

It is noted that the use of an energy inner product to do the Galerkin projection step of the model reduction does not guarantee that the stability of an equilibrium point other than the origin is preserved, nor does it guarantee that the stability of limit cycles is preserved [81]. Moreover, an equilibrium point of a dynamical system is not necessarily an attractor of the system. Hence, a ROM constructed in an energy inner product may not preserve an attractor for the compressible Navier-Stokes equations. It is also noted that, in some situations, an energy-based inner product may not be appropriate, e.g., for jet noise problems in which the acoustic waves of interest contain a very small fraction of the energy [81].

Isentropic compressible Navier-Stokes equations

In [81], a physically meaningful energy inner product for the isentropic compressible Navier-Stokes equations (5.42) is introduced. This inner product is reviewed here, as it motivates the definition of an energy inner product for the full compressible Navier-Stokes equations in the primitive variables (5.35). The inner product requires first a transformation of the equations (5.42) into a new set of variables, and is based on either the stagnation enthalpy or the stagnation energy. The transformed variables are c, the speed of sound, and the three velocity components u_i for $i = 1,2,3$. In non-dimensional form, the nonlinear isentropic compressible Navier-Stokes equations in these variables are:

$$\begin{array}{rcl} c_{,t} + u_i c_{,i} + \frac{\gamma-1}{2} c u_{i,i} & = & 0, \\ u_{j,t} + u_i u_{j,i} + \frac{2}{\gamma-1} c c_{,j} - \frac{1}{Re} u_{j,ii} & = & 0, \end{array} \tag{5.48}$$

again for $j = 1,2,3$. In [81], Rowley *et al.* define the following inner product for (5.48):

$$\left(\mathbf{q}^{(1)}, \mathbf{q}^{(2)}\right)_{SE,\alpha} \equiv \int_\Omega \left(u_i^{(1)} u_i^{(2)} + \frac{2\alpha}{\gamma-1} c^{(1)} c^{(2)} \right) d\Omega, \tag{5.49}$$

where $\mathbf{q}^{(1)} \equiv \begin{pmatrix} c^{(1)}, & u_1^{(1)}, & u_2^{(1)}, & u_3^{(1)} \end{pmatrix}^T$ and similarly for $\mathbf{q}^{(2)}$. When $\alpha = 1$ in (5.49), the norm induced by (5.49) represents the stagnation enthalpy; when $\alpha = 1/\gamma$, the norm induced by (5.49) represents the stagnation energy[7].

Note that the nonlinearities in the transformed equations (5.48) are only quadratic. Let

$$\boldsymbol{\phi}_k \equiv \begin{pmatrix} \phi_k^c \\ \phi_k^{u_1} \\ \phi_k^{u_2} \\ \phi_k^{u_3} \end{pmatrix}, \tag{5.51}$$

[7] Note that, for the isentropic compressible Navier-Stokes equations, the stagnation energy

$$E_s = \int_\Omega \left(e + \frac{1}{2} u_i u_i \right) d\Omega, \tag{5.50}$$

and total energy (5.50) are equivalent (and similarly for stagnation enthalpy and total enthalpy). The second term in (5.50) is commonly referred to as the specific energy.

denote the k^{th} POD mode. Projecting (5.48) onto this mode in the inner product (5.49) and integrating the viscous term by parts[8] gives:

$$\int_\Omega \left(u_{j,t} + u_i u_{j,i} + \frac{2}{\gamma - 1} c c_{,j} \right) \phi_k^{u_j} d\Omega + \int_\Omega \frac{1}{Re} u_{j,i} \phi_{k,i}^{u_j} d\Omega + \int_\Omega \frac{2\alpha}{\gamma - 1} \left(c_{,t} + u_i c_{,i} + \frac{\gamma - 1}{2} c u_{i,i} \right) \phi_k^c d\Omega = 0, \tag{5.52}$$

for $k = 1, \ldots, M$. Upon discretization, (5.52) gives rise to a nonlinear matrix problem of the form:

$$\dot{x}_{M,i} + \sum_{k=1}^{M} A_{ik} x_{M,k} + \sum_{k=1}^{M} \sum_{q=1}^{M} A_{ikq}^{(3)} x_{M,k} x_{M,q} = 0, \tag{5.53}$$

for $i = 1, \ldots, M$, assuming the POD modes have been normalized with respect to the inner product (5.49). The cost of applying a time-integration scheme to (5.53) (the online stage of the model reduction) is $\mathscr{O}(M^3)$.

Full compressible Navier-Stokes equations

As mentioned at the beginning of this chapter, a ROM for (5.35) constructed in the L^2 inner product is not guaranteed to give rise to an energy-stable formulation. To remedy this problem, an inner product whose norm is the total energy of the fluid system (5.45) is formulated here. Before defining the said inner product, a transformation of the equations (5.35) is required. Let a be a new variable, given by

$$a^2 \equiv \rho. \tag{5.54}$$

Now, define the following transformed fluid vector:

$$\mathbf{q} \equiv \begin{pmatrix} a \\ b_1 \\ b_2 \\ b_3 \\ d \end{pmatrix} = \begin{pmatrix} a \\ au_1 \\ au_2 \\ au_3 \\ ae \end{pmatrix} \in \mathbb{R}^5. \tag{5.55}$$

In the transformed variables, the (non-dimensional) compressible Navier-Stokes equations (5.35) are as follows:

$$\begin{aligned} 2a\tfrac{\partial a}{\partial t} + \tfrac{\partial (ab_i)}{\partial x_i} &= 0, \\ \tfrac{\partial (ab_i)}{\partial t} + \tfrac{\partial (b_j b_i)}{\partial x_j} + (\gamma - 1)\tfrac{\partial (ad)}{\partial x_i} - \tfrac{1}{Re}\tfrac{\partial \tau_{ij}}{\partial x_j} &= 0, \\ \tfrac{\partial (ad)}{\partial t} + \tfrac{\partial (b_j d)}{\partial x_j} + (\gamma - 1)\tfrac{d}{a}\left(a\tfrac{\partial b_j}{\partial x_j} - b_j \tfrac{\partial a}{\partial x_j} \right) - \tfrac{\gamma}{PrRe}\tfrac{\partial}{\partial x_j}\left(\tfrac{\kappa}{a^2}\left(a\tfrac{\partial d}{\partial x_j} - d\tfrac{\partial a}{\partial x_j} \right)\right) - \tfrac{1}{Re}\tfrac{1}{a^2}\tau_{ij}\left(a\tfrac{\partial b_i}{\partial x_j} - b_i \tfrac{\partial a}{\partial x_j} \right) &= 0, \end{aligned} \tag{5.56}$$

where

$$\tau_{ij} = \frac{\mu}{a^2}\left[a\frac{\partial b_i}{\partial x_j} - b_i \frac{\partial a}{\partial x_j} + a\frac{\partial b_j}{\partial x_i} - b_j \frac{\partial a}{\partial x_i} \right] + \delta_{ij}\frac{\lambda}{a^2}\left(a\frac{\partial b_k}{\partial x_k} - b_k \frac{\partial a}{\partial x_k} \right), \tag{5.57}$$

[8] As before, the boundary condition terms are omitted from the ROM equations, which in general is justified if the POD modes satisfy the FOM boundary conditions; see [57] and Appendix A.5.

for $i,j=1,2,3$. (5.56) can be simplified as follows:

$$\begin{aligned}
2a\tfrac{\partial a}{\partial t}+a\tfrac{\partial b_i}{\partial x_i}+b_i\tfrac{\partial a}{\partial x_i} &= 0, \\
a\tfrac{\partial b_i}{\partial t}+\tfrac{1}{2}b_i\tfrac{\partial b_j}{\partial x_j}-\tfrac{1}{2a}b_ib_j\tfrac{\partial a}{\partial x_j}+b_j\tfrac{\partial b_i}{\partial x_j}+(\gamma-1)\left[a\tfrac{\partial d}{\partial x_i}+d\tfrac{\partial a}{\partial x_i}\right] & \\
-\tfrac{1}{Re}\tfrac{\partial \tau_{ij}}{\partial x_j} &= 0, \\
a\tfrac{\partial d}{\partial t}+b_j\tfrac{\partial d}{\partial x_j}-\tfrac{1}{a}\left(\gamma-\tfrac{1}{2}\right)db_j\tfrac{\partial a}{\partial x_j}+\left(\gamma-\tfrac{1}{2}\right)d\tfrac{\partial b_j}{\partial x_j}+\tfrac{\gamma}{PrRe}\tfrac{\partial}{\partial x_j}\left[\tfrac{\kappa}{a^2}\left(a\tfrac{\partial d}{\partial x_j}-d\tfrac{\partial a}{\partial x_j}\right)\right] & \\
-\tfrac{1}{Re}\tfrac{1}{a^2}\tau_{ij}\left(a\tfrac{\partial b_i}{\partial x_j}-b_i\tfrac{\partial a}{\partial x_j}\right) &= 0,
\end{aligned} \tag{5.58}$$

for $i,j=1,2,3$.

Consider the following inner product[9]:

$$\left(\mathbf{q}^{(1)},\mathbf{q}^{(2)}\right)_E \equiv \int_\Omega \frac{1}{2}\left(a^{(1)}d^{(2)}+a^{(2)}d^{(1)}+b_i^{(1)}b_i^{(2)}\right)d\Omega. \tag{5.59}$$

The norm induced by the inner product (5.59) is:

$$\begin{aligned}
||\mathbf{q}||_E^2 &\equiv (\mathbf{q},\mathbf{q})_E \\
&= \int_\Omega \left(ad+\tfrac{1}{2}b_ib_i\right)d\Omega \\
&= \int_\Omega \left(a^2e+\tfrac{1}{2}a^2u_iu_i\right)d\Omega \\
&= \int_\Omega \left(\rho e+\tfrac{1}{2}\rho u_iu_i\right)d\Omega \\
&= E_T.
\end{aligned} \tag{5.60}$$

That is, the norm induced by the inner product (5.59) is the total energy of the fluid system. If the Galerkin projection step of the model reduction procedure is done in the inner product (5.59), the resulting ROM should preserve the stability of an equilibrium point of the system at the origin.

The projection of the equations (5.58) onto the reduced basis modes in the total energy inner product (5.59) is now derived. Let

$$\boldsymbol{\phi}_k = \begin{pmatrix} \phi_k^a \\ \phi_k^{b_1} \\ \phi_k^{b_2} \\ \phi_k^{b_3} \\ \phi_k^d \end{pmatrix}, \tag{5.61}$$

denote the k^{th} POD mode.

Consider first the inviscid variant of (5.58) ($\mu=\lambda=\kappa=0$). The first step is to convert the non-polynomial nonlinearities in (5.58) into polynomial ones. To do this, the momentum and energy equations are each multiplied by a. Doing so, and projecting (5.58) onto the mode (5.61) in the total energy inner product (5.59) gives:

$$\begin{aligned}
&\int_\Omega \left(2a\tfrac{\partial a}{\partial t}+a\tfrac{\partial b_i}{\partial x_i}+b_i\tfrac{\partial a}{\partial x_i}\right)\phi_k^d d\Omega \\
&+\int_\Omega \left(a^2\tfrac{\partial b_i}{\partial t}+\tfrac{1}{2}b_ia\tfrac{\partial b_j}{\partial x_j}-\tfrac{1}{2}b_ib_j\tfrac{\partial a}{\partial x_j}+b_ja\tfrac{\partial b_i}{\partial x_j}+(\gamma-1)a\left[a\tfrac{\partial d}{\partial x_i}+d\tfrac{\partial a}{\partial x_i}\right]\right)\phi_k^{b_j}d\Omega \\
&\int_\Omega \left(a^2\tfrac{\partial d}{\partial t}+ab_j\tfrac{\partial d}{\partial x_j}-\left(\gamma-\tfrac{1}{2}\right)db_j\tfrac{\partial a}{\partial x_j}+\left(\gamma-\tfrac{1}{2}\right)da\tfrac{\partial b_j}{\partial x_j}\right)\phi_k^a d\Omega = 0.
\end{aligned} \tag{5.62}$$

[9] It is shown in Appendix A.4 that (5.59) is a valid inner product.

When discretized, (5.62) will yield a nonlinear system of the form:

$$\sum_{k=1}^{M}\sum_{q=1}^{M} M_{ikq}^{(3)} x_{M,q}\dot{x}_{M,k} + \sum_{k=1}^{M}\sum_{q=1}^{M} A_{ikq}^{(3)} x_{M,k}x_{M,q} + \sum_{k=1}^{M}\sum_{q=1}^{M}\sum_{r=1}^{M} A_{ikqr}^{(4)} x_{M,k}x_{M,q}x_{M,r} = 0, \tag{5.63}$$

for $i = 1,...,M$. The entries of the tensors $M_{ikq}^{(3)}$, $A_{ikq}^{(3)}$ and $A_{ikqr}^{(4)}$ appearing in (5.63), referred to as the third order mass tensor, the third order ROM coefficient tensor, and the fourth order ROM coefficient tensor, respectively, can be deduced from (5.62), and are not given here for the sake of brevity. All mass and ROM coefficient tensors in (5.63) can be pre-computed in the offline stage of the model reduction, and stored for use during the online time-integration stage of the ROM. Note that, in the online stage, it is necessary to evaluate tensor/vector products involving these mass and ROM coefficient tensors, which requires $\mathscr{O}(M^4)$ operations (Section 4.4). Ideas for reducing this computational cost are discussed at the end of this subsection.

For the more general viscous case, to remove the non-polynomial nonlinearities in (5.58), the momentum equation needs to be multiplied by a^3, and the energy equation by a^4. Doing so, and projecting these equations onto (5.61) in the inner product (5.59), following an integration by parts on the viscous terms[10], gives:

$$\begin{aligned}
&\int_\Omega \left(2a\tfrac{\partial a}{\partial t} + a\tfrac{\partial b_i}{\partial x_i} + b_i\tfrac{\partial a}{\partial x_i}\right)\phi_k^d d\Omega \\
&+ \int_\Omega \left(a^4\tfrac{\partial b_i}{\partial t} + \tfrac{1}{2}b_i a^3\tfrac{\partial b_j}{\partial x_j} - \tfrac{1}{2}a^2 b_i b_j\tfrac{\partial a}{\partial x_j} + b_j a^3\tfrac{\partial b_i}{\partial x_j} + (\gamma-1)a^3\left[a\tfrac{\partial d}{\partial x_i} + d\tfrac{\partial a}{\partial x_i}\right]\right)\phi_k^{b_j} d\Omega \\
&+ \int_\Omega \tfrac{1}{Re}\left(3\phi_k^{b_j}a^2\tfrac{\partial a}{\partial x_j} + a^3\tfrac{\partial \phi_k^{b_j}}{\partial x_j}\right)\tau_{ij} d\Omega \\
&+ \int_\Omega \left(a^5\tfrac{\partial d}{\partial t} + a^4 b_j\tfrac{\partial d}{\partial x_j} - \left(\gamma-\tfrac{1}{2}\right)a^3 d b_j\tfrac{\partial a}{\partial x_j} + \left(\gamma-\tfrac{1}{2}\right)d a^4\tfrac{\partial b_j}{\partial x_j}\right)\phi_k^a d\Omega \\
&+ \int_\Omega \kappa\left(a^2\tfrac{\partial \phi_k^d}{\partial x_j} + 4\phi_k^a a\tfrac{\partial a}{\partial x_j}\right)\left(a\tfrac{\partial d}{\partial x_j} - d\tfrac{\partial a}{\partial x_j}\right)d\Omega - \int_\Omega \tfrac{1}{Re}a^2\tau_{ij}\left(a\tfrac{\partial b_i}{\partial x_j} - b_i\tfrac{\partial a}{\partial x_j}\right)\phi_k^a d\Omega = 0.
\end{aligned} \tag{5.64}$$

When discretized, (5.64) will yield a nonlinear system of the form:

$$\begin{aligned}
&\textstyle\sum_{k=1}^{M}\sum_{q=1}^{M} M_{ikq}^{(3)} x_{M,q}\dot{x}_{M,k} + \sum_{k=1}^{M}\sum_{q=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{M}\sum_{t=1}^{M} M_{ikqrst}^{(6)} x_{M,q}x_{M,r}x_{M,s}x_{M,t}\dot{x}_{M,k} \\
&\textstyle+\sum_{k=1}^{M}\sum_{q=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{M}\sum_{t=1}^{M}\sum_{p=1}^{M} M_{ikqrstp}^{(7)} x_{M,q}x_{M,r}x_{M,s}x_{M,t}x_{M,p}\dot{x}_{M,k} \\
&\textstyle+\sum_{k=1}^{M}\sum_{q=1}^{M} A_{ikq}^{(3)} x_{M,k}x_{M,q} + \sum_{k=1}^{M}\sum_{q=1}^{M}\sum_{r=1}^{M} A_{ikqr}^{(4)} x_{M,k}x_{M,q}x_{M,r} \\
&\textstyle+\sum_{k=1}^{M}\sum_{q=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{M} A_{ikqrs}^{(5)} x_{M,k}x_{M,q}x_{M,r}x_{M,s} \\
&\textstyle+\sum_{k=1}^{M}\sum_{q=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{M}\sum_{t=1}^{M} A_{ikqrst}^{(6)} x_{M,k}x_{M,q}x_{M,r}x_{M,s}x_{M,t} \\
&\textstyle+\sum_{k=1}^{M}\sum_{q=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{M}\sum_{t=1}^{M}\sum_{p=1}^{M} A_{ikqrstp}^{(7)} x_{M,k}x_{M,q}x_{M,r}x_{M,s}x_{M,t}x_{M,p} = 0,
\end{aligned} \tag{5.65}$$

for $i = 1,...,M$. As before, the mass and ROM coefficient tensors $\mathbf{A}^{(k)}$ and $\mathbf{M}^{(k)}$ appearing in (5.65) are not given here explicitly for the sake of brevity, but can be inferred by comparing (5.65) with (5.64). The reader can observe that (5.65) has up to a 7^{th} order mass and ROM coefficient tensor. Like for the inviscid case (5.63), all mass and ROM coefficient tensors appearing in (5.65) can be pre-computed during the offline stage of the model reduction.

[10] Neglecting the resulting boundary integrals is justified if the POD modes satisfy the boundary conditions of the governing PDEs; see Appendix A.5 and [57].

Per the discussion in Section 4.4, the cost of evaluating the tensor/vector products arising when (5.65) is integrated forward in time is $\mathscr{O}(M^7)$. This cost is substantial even for small M, e.g., $M = \mathscr{O}(10)$. To restore efficiency of the ROM, the nonlinear terms in (5.64) can be interpolated using well-established methods such as DEIM [29], “best points” interpolation [74, 75], or gappy POD [33]. As the objective here is to evaluate the accuracy of ROMs constructed in the total energy inner product (proposed here for the first time), the nonlinear terms in (5.65) are handled directly, that is, without interpolation. The addition of interpolation capabilities to the `Spirit` code (Section 5.4) for the purpose of improving the efficiency of the nonlinear ROMs implemented there may be the subject of future work.

Note that no energy inner product is proposed for the ζ–form of the compressible Navier-Stokes equations (5.41). Deriving such an inner product seems *a priori* a promising path, as the ζ–form of the equations have only quadratic nonlinearities, instead of cubic nonlinearities (5.35). Unfortunately, the equations are not amenable to such a formulation if a projected system having only polynomial nonlinearities is desired.

5.4 Implementation/`Spirit` code

A parallel C++ code that reads in the snapshot data written by a high-fidelity code, assembles the necessary finite element representation of the snapshots and computes the numerical quadrature necessary for the evaluation of the inner products has been written by the authors. The code, known as `Spirit`, is kept in a `git` repository on the Sandia Restricted Network (SRN) Common Engineering Environment (CEE) space, and can be cloned from this repository using the following command:

```
    git clone user@ceerwsXXXXX:/projects/aerosciences/repo/git/spirit
```

To ensure software quality, the code is pulled from its repository, compiled and tested every night using a `cronjob` on the PI’s Linux workstation.

The stability analysis in this chapter has assumed that the integrals resulting from the projection of the governing equations onto the reduced basis modes are evaluated exactly in continuous form. This continuous result can be translated to the discrete setting through the use of high-precision numerical quadrature as follows. First, the snapshots and the POD basis modes are cast as a collection of continuous finite elements. It is then possible to construct a numerical quadrature operator that computes exactly (with respect to the finite element representation) all continuous inner products arising from the continuous Galerkin projection of the equations onto the POD modes.

For concreteness, the numerical quadrature is illustrated here on the simpler case of linearized equations, e.g., the linearized compressible Euler or Navier-Stokes equations. By the discussion in Section 5.2, it is necessary to compute numerically integrals of the form:

$$\left(\mathbf{q}^{(1)}, \mathbf{q}^{(2)}\right)_{(\mathbf{H},\Omega)} = \int_{\Omega} \mathbf{q}^{(1)T} \mathbf{H} \mathbf{q}^{(2)} d\Omega, \tag{5.66}$$

where $\mathbf{H}$ is the symmetrizer for the governing system. Suppose, without loss of generality, that the finite element shape functions are chosen to be bilinear. The discrete representations of the vectors $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$ are denoted by $\mathbf{q}^{h(1)}$ and $\mathbf{q}^{h(2)}$, respectively. The length of these vectors is equal to the number of mesh nodes N times the dimension of the vector, r. Let $\mathbf{H}_e^h$ be the $r \times r$ element inner product matrix, taken to be piecewise constant over each element. Then, the formula for numerical integration of (5.66) can be written as

$$\left(\mathbf{q}^{(1)}, \mathbf{q}^{(2)}\right)_{(\mathbf{H},\Omega)} = \mathbf{q}^{h(1)T}\mathbf{W}\mathbf{q}^{h(2)}, \tag{5.67}$$

where $\mathbf{W}$ is a sparse block matrix comprised of $N \times N$ blocks of dimension $r \times r$. The $(k,l)^{th}$ block of this matrix given by $w_{kl}\mathbf{I}$, where

$$w_{kl} = \sum_{e=1}^{n_{kl}^{el}} \mathbf{H}_e^h \sum_{j=1}^{4} N_{k_e}(\mathbf{x}_{j_e}) N_{l_e}(\mathbf{x}_{j_e}) \omega_{j_e}. \tag{5.68}$$

Here, the outer sum is over the elements connected to the $k-l$ nodal “edge”; the ω_{j_e} are the integration weights and the $\mathbf{x}_{j_e}$ are the integration points.

All calculations are performed in parallel using distributed matrix and vector data structures and parallel eigensolvers from the Trilinos project [49]. The parallelism[11] in `Spirit` allows for large data sets and a relatively large number of POD modes. The `libmesh` finite element library [63] is used to compute element quadratures. The online time-integration of the ROM system (4.18) (with the mass tensors and ROM coefficient tensors computed within `Spirit` and written to disk) is performed using a fourth-order Runge-Kutta scheme in MATLAB.

The `Spirit` code contains several linear as well as nonlinear compressible fluid PDE sets, which can be projected onto the POD reduced basis modes in several inner products. The physics and inner products available in `Spirit` at the time this report was written are summarized in Table 5.2. For the linearized equations, the base flow to linearize about, $\bar{\mathbf{q}}(\mathbf{x})$, can be set either to a spatially-constant value (for uniform base flow), or read in from file (for non-uniform base flow). Note that the linearized compressible Navier-Stokes equations in `Spirit` include the viscous work terms (A.2) (Appendix A.2). The base flow is not needed in the case of nonlinear equations, as the ROM (and hence the POD basis) is constructed for the full state vector $\mathbf{q}(\mathbf{x},t)$ in this case. The option to run dimensionally as well as non-dimensionally is available. In general, the latter is preferred, as the resulting systems are often better scaled. Additional capabilities available in `Spirit` include:

- The capability to specify various boundary conditions (e.g., no-penetration boundary condition, sponge far-field boundary conditions; Appendix A.5) in the ROM.
- The capability to add source terms (e.g., body force actuation; Appendix A.11) to the ROM equations.

It is straightforward to add new physics and boundary conditions to `Spirit` with an arbitrary number of dofs per node.

[11] Note that, at the present time, the input/output (I/O) in `Spirit` is serial: the snapshots and mesh data are read in on processor 0, then distributed to the other processors (and similarly for the output). All other operations, e.g., the POD basis calculation and Galerkin projection, are fully parallel.

Table 5.2. PDEs/inner products available in `Spirit`

PDEs	regimes	# dofs/node	unknowns	inner product
linearized compressible Euler	inviscid	5	$u_1', u_2', u_3', \zeta', p'$	L^2 symmetry
linearized compressible Navier-Stokes*	viscous inviscid	5	$u_1', u_2', u_3', T', \rho'$	L^2 symmetry
isentropic compressible Navier-Stokes	viscous inviscid	4	u_1, u_2, u_3, h	L^2
isentropic compressible Navier-Stokes	viscous inviscid	4	u_1, u_2, u_3, c	stagnation energy stagnation enthalpy
ζ–form compressible Navier-Stokes	viscous inviscid	5	u_1, u_2, u_3, ζ, p	L^2
compressible Navier-Stokes	viscous inviscid	5	u_1, u_2, u_3, T, ρ	L^2
compressible Navier-Stokes	viscous inviscid	5	$a = \sqrt{\rho}$, $\{b_i\}_{i=1}^3$, $d = ae$	total energy

* The linearized compressible Navier-Stokes equations in `Spirit` include the viscous work terms (A.2) (Appendix A.2).

One reason `Spirit` is designed as a stand-alone code is so that it can be synchronized with *any* high-fidelity CFD code that can write out a mesh and snapshot data stored at the nodes of this mesh. At the time of publication of this report, `Spirit` has been run with two high-fidelity flow solvers: `AERO-F` and `SIGMA CFD`. The `AERO-F` simulation code is an arbitrary Lagrangian-Eulerian finite volume code that can be used for high-fidelity aeroelastic analysis [69, 34]. `SIGMA CFD` is a Sandia in-house high-fidelity finite volume flow solver. This code is derived from `LESLIE3D`, an LES flow solver originally developed at the Computational Combustion Laboratory at the Georgia Institute of Technology, and has direct numerical simulation (DNS) as well as LES capabilities. For a detailed description of the schemes and models implemented within LESLIE3D, the reader is referred to [38, 37].

More information about the `Spirit` code can be found in [35, 19].

5.5 Numerical results

The linear as well as nonlinear model reduction methods detailed in the previous sections are now tested on three benchmarks, summarized in Table 5.3.

Table 5.3. `Spirit` test cases

Name	FOM code	FOM physics	Dynamics
2D inviscid pulse	`AERO-F`	Linearized compressible Euler	linear
2D viscous pulse	`SIGMA CFD`	Nonlinear compressible Navier-Stokes	linear
Viscous laminar cavity	`SIGMA CFD`	Nonlinear compressible Navier-Stokes	nonlinear

The first two benchmarks, the 2D inviscid and 2D viscous pulse test cases (Section 5.5.1 and 5.5.2), have effectively linear dynamics and are considered for code verification purposes. The third benchmark (Section 5.5.3), the viscous laminar cavity test case, is a problem whose solution exhibits inherently nonlinear dynamics, namely vortices that convect. The domain of interest for this problem is an open T–shaped cavity, precisely the geometry relevant for the targeted captive-carry application (Figure 1.1(c)). The problems are run in parallel on either the Red Sky cluster or a CEE Remove Graphics Workstation, both at Sandia.

In calculating the POD basis for the vector-valued fluid field $\mathbf{q}$ (or fluid fluctuation $\mathbf{q}'$, in the linear case), there are two approaches. The first approach is to calculate a separate scalar basis for each of the variables comprising $\mathbf{q}$ (or $\mathbf{q}'$). The second approach is to build a vector-valued basis $\{\boldsymbol{\phi}_k \in \mathbb{R}^n\}_{k=1}^{M}$for $\mathbf{q}$ (or $\mathbf{q}'$), where n is the number of dofs (PDEs) per node (in general $n = 5$ in 3D, except for the case of isentropic equations, where $n = 4$). The latter method involving the vector-valued basis is taken here. Previous work [81, 56] suggests that the scalar-valued method appears to offer no advantages over the vector-valued method and requires higher order models.

Among the results shown for the three test cases are time histories of the i^{th} ROM modal amplitude compared to the projection of the FOM CFD simulation onto the i^{th} POD mode. Mathematically, the figures shown compare as a function of time t:

$$x_{M,i}(t) \quad \text{vs.} \quad (\mathbf{q}_{\text{FOM}}, \boldsymbol{\phi}_i), \tag{5.69}$$

for $1 \leq i \leq M$, where $\mathbf{q}_{\text{FOM}}$ is the high-fidelity CFD solution from which the ROMs are constructed.

Also reported are the relative errors in the ROM solutions with respect to the snapshots, i.e.,

$$\mathscr{E}_{rel,\mathbf{q}} = \sqrt{\frac{\sum_{i=1}^{K_{max}} ||\mathbf{q}_{\text{FOM}}(t_i) - \mathbf{q}_M(t_i)||_2}{\sum_{i=1}^{K_{max}} ||\mathbf{q}_{\text{FOM}}(t_i)||_2}}, \tag{5.70}$$

where $\mathbf{q}_M(t) = \sum_{k=1}^{M} x_{M,k}\boldsymbol{\phi}_k(t)$ and $\mathbf{q}_{\text{FOM}}(t)$ denote the ROM and FOM solutions (respectively) at time t, and K_{max} denotes the integer such that $T_{max} = K_{max}\Delta t_{snap}$, where T_{max} is the maximum time until which the ROM is run, and Δt_{snap} is the time step between snapshots. The norm $||\cdot||_2$ in (5.70) is the discrete L^2 norm.

Note that, for all three test cases, emphasis is placed on reproducing a given CFD solution for a single set of flow conditions in a stable and accurate fashion. This is viewed as an essential prerequisite for applying the method to more complex situations, such as building ROMs valid across a parameter space or range of flow conditions, and for using the ROMs to do analysis (e.g., flow control, uncertainty quantification).

Note also that the ROMs tested do not include an implementation of the boundary conditions, although some boundary conditions are available in `Spirit`, as discussed earlier. For the viscous problems, the POD modes satisfy strongly the no-slip and adiabatic wall boundary conditions at the cavity walls, meaning the ROMs inherit automatically these boundary conditions from their underlying FOMs. It is conjectured that the inclusion of a far-field non-reflecting boundary condition in the ROMs may improve the models' stability and accuracy (Appendix A.5).

5.5.1 2D inviscid pulse in uniform base flow

The first benchmark is a linear inviscid problem. The test case involves the propagation and reflection of a cylindrical acoustic pulse from two parallel walls in a uniform base flow. The governing equations are the linearized compressible Euler equations (Section 5.2.1). The base flow velocity is taken to be uniform in the x_1-direction with free-stream Mach number $M_\infty \equiv \frac{\bar{u}_1}{c} = 0.25$. The initial condition at time $t = 0$ is

$$\frac{p'}{\bar{\rho}\bar{c}^2} = 0.1 M_\infty^2 \exp(-((x_1-10)^2+(x_2+1)^2)), \quad \frac{\zeta'}{\zeta} = -\frac{p'}{\bar{\rho}\bar{c}^2}, \quad u_1' = u_2' = u_3' = 0. \tag{5.71}$$

The exact solution for this IBVP can be found in [91].

The `AERO-F` node-centered finite volume code [34] is used to generate the snapshots from which the ROM POD bases are constructed. This high-fidelity solver is run in the linearized compressible Euler regime, as the problem is inviscid and the dynamics are linear. The numerical solution is performed on a 3D rectangular prism domain, with extent $0 \le x_1 \le 20$, $-5 \le x_2 \le 5$, $0 \le x_3 \le 1$, discretized by finite volumes that are interconnected to form a mesh of 1,052,551 unstructured tetrahedral elements (with a total of 197,226 nodes). Slip wall boundary conditions are applied on the constant x_2 and x_3 boundaries in the high-fidelity code. The high-fidelity simulation is run for a non-dimensional time of $T_{max} = 6.4$, using 624 time steps. Snapshots are saved every four time steps beginning at time $t = 0.57$ and ending at $t = T_{max}$, to yield a total of $K_{max} = 157$ snapshots. These snapshots are used to generate 12 mode POD bases in the various inner products evaluated. It is found that the 12 mode bases capture approximately 99.4% of the snapshot energy (4.6). The FOM as well as the ROMs are run non-dimensionally.

Eight of the ten ROM approaches[12] summarized in Table 5.2 are evaluated. As this is an inviscid problem, the viscosity coefficients in the compressible Navier-Stokes equations are all set to zero: $\mu = \lambda = \kappa = 0$. Since the high-fidelity code solves the *linearized* equations and the dynamics are linear, it is expected that the linearized ROMs will be able to accurately capture the solution to the problem. Note that, because the base flow is uniform, the $\mathbf{G}$ matrix in (5.22) vanishes, and the $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ matrices appearing in these equations are spatially constant. It follows from Corollary 5.2.1 that the linearized symmetry ROMs should be time-stable and stable in the sense of Lyapunov. The nonlinear ROMs, which should also be able to reproduce reasonably the solution dynamics, are tested as a verification of the implementation of the nonlinear physics in `Spirit`.

[12] The ROMs based on the linearized compressible Navier-Stokes equations are not considered; considered instead are ROMs based on the linearized compressible Euler equations, as the problem of interest is inviscid.

First, the stability of the two linearized inviscid ROMs[13] in Table 5.2 is evaluated using Theorem 3.2.3, that is, by checking the sign of the real parts of the eigenvalues of the ROM system matrices $\mathbf{A}_M$. The maximum real parts of the eigenvalues of $\mathbf{A}_M$ for the $M = 12$ linearized POD/Galerkin ROMs are given in Table 5.4. The reader can observe that the L^2 ROM is mildly unstable based on the eigenvalue criterion for stability. The symmetry ROM, in contrast, is stable.

Table 5.4. Maximum real part of eigenvalues of $\mathbf{A}_M$ for $M = 12$ mode linearized ROMs constructed in the L^2 and symmetry inner products (2D inviscid pulse problem)

Physics	Inner product	$\max\{Re[\lambda_i(\mathbf{A}_M)]\}$
Linearized compressible Euler	L^2	0.03
	symmetry	0.0

Next, the time history of the modal amplitudes $x_{M,i}$ is compared to the projection of the POD modes onto the snapshots $(\mathbf{q}_{CFD}, \boldsymbol{\phi}_i)$ for $i = 1, 2$ (5.69). Figure 5.1 shows these quantities as a function of time for the nonlinear compressible Navier-Stokes ROM constructed in the total energy inner product. The reader can observe that the agreement is excellent, which indicates that the ROM can reproduce well the snapshots from which it was constructed. Similar figures were produced using the remaining seven ROMs evaluated, and are not shown here for the sake of brevity.

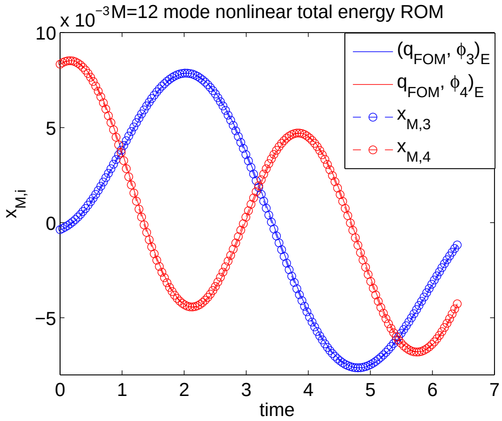
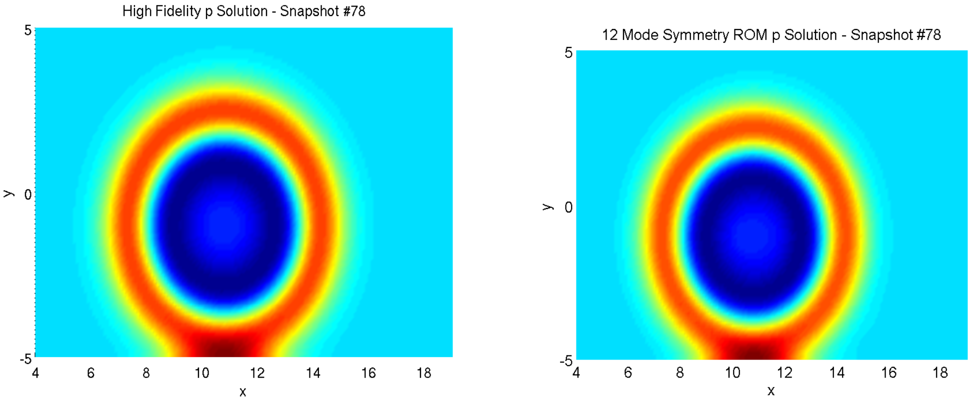


Figure 5.1. Time history of modal amplitudes for $M = 12$ mode nonlinear ROM constructed in the total energy inner product (2D inviscid pulse problem)

Figures 5.2 and 5.3 compare the high-fidelity pressure field (a) with the field reconstructed from the $M = 12$ mode compressible linearized Euler symmetry ROM (b) at the time of the 78^{th} and 157^{th} (final) snapshot respectively. There is a good qualitative agreement between the high-fidelity solution and the ROM solution.

Next, a quantitative evaluation of the ROMs is performed: the relative errors in the eight ROMs tested are calculated using the formula (5.70) as a function of the number of POD modes M (Table

[13] The stability of the nonlinear ROMs is not evaluated, as Theorem 3.2.3 is limited to linear systems.



(a) FOM

(b) 12 mode linearized compressible Euler symmetry ROM

Figure 5.2. Pressure fluctuation contours at time of the 78^{th} snapshot (2D inviscid pulse problem)

5.5). The size of the ROM basis is varied from six to eighteen. The errors in Table 5.5 reveal that all eight ROMs deliver accurate solutions, with the nonlinear isentropic L^2 ROM delivering in general the most accurate solution. Convergence with refinement in the basis size M is observed.

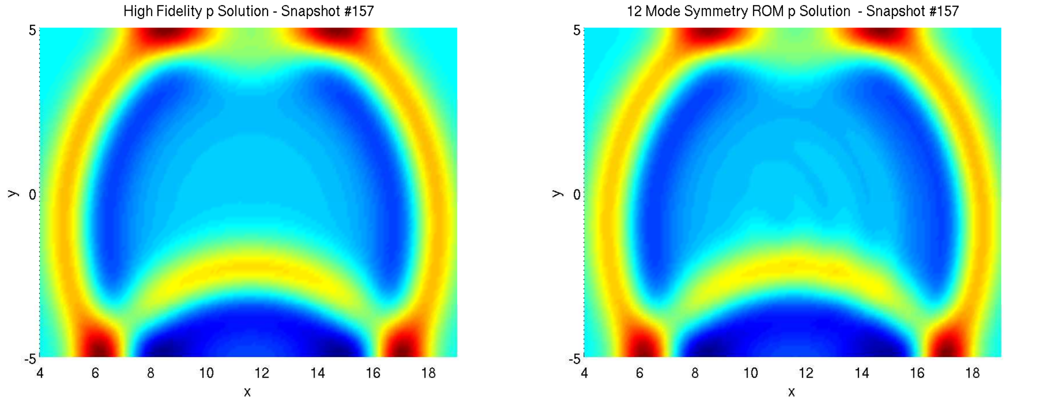
Table 5.5. ROM relative errors $\mathscr{E}_{rel,\mathbf{q}}$ (2D inviscid pulse problem)

Physics	Inner product	$M = 6$	$M = 12$	$M = 18$
Linearized compr Euler	L^2	1.38×10^{-4}	1.65×10^{-5}	1.64×10^{-5}
	symm	1.34×10^{-4}	1.66×10^{-5}	1.62×10^{-5}
Nonlinear compr isentropic N-S	L^2	5.72×10^{-5}	7.32×10^{-6}	7.16×10^{-6}
	stag ene	4.82×10^{-4}	1.26×10^{-5}	1.23×10^{-5}
	stag enth	2.81×10^{-4}	1.27×10^{-5}	1.23×10^{-5}
ζ–form nonlinear compr N-S	L^2	1.14×10^{-4}	1.68×10^{-5}	1.66×10^{-5}
Nonlinear compr N-S	L^2	1.08×10^{-4}	2.25×10^{-5}	2.19×10^{-5}
	tot ene	1.02×10^{-4}	1.18×10^{-5}	1.14×10^{-5}

The errors in Table 5.5 give an idea of how accurate are the ROM solutions. Also of interest is how well the ROMs can represent the total energy of the fluid. Figures 5.4 and 5.6 show as a function of time the total energy of the fluid calculated using the linearized and nonlinear ROMs (respectively), compared with the total energy of the fluid calculated using the FOM (5.45). Similarly, Figure 5.5 shows the stagnation energy (5.50) of the fluid calculated using the isentropic nonlinear ROMs compared with the stagnation energy of the fluid calculated using the FOM[14]. The reader can observe that the ROM total/stagnation energy converges to the FOM total/stagnation energy as the basis size M is increased. ROMs based on the nonlinear compressible Navier-Stokes equations[15]

[14] Recall that the total and stagnation energy is equivalent for the isentropic compressible Navier-Stokes equations.

[15] The total energy for the ROMs based on the ζ–form of the nonlinear compressible Navier-Stokes equations is not



(a) FOM

(b) 12 mode linearized compressible Euler symmetry ROM

Figure 5.3. Pressure fluctuation contours at time of the 157^{th} snapshot (2D inviscid pulse problem)

track better the total energy of the snapshots for small M (Figure 5.6).

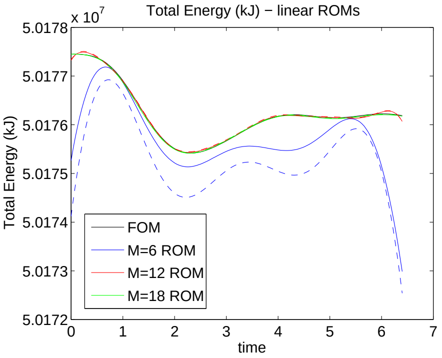


Figure 5.4. FOM vs. ROM total energy as a function of basis size M for linear compressible Euler ROMs: solid lines = L^2 ROMs, dashed lines = symmetry ROMs (2D inviscid pulse problem)

Having studied the accuracy of the ROMs, attention is now turned to efficiency. Table 5.6 gives the total CPU times (in seconds) of the offline (quadrature weight calculation and Galerkin projection) and online (time-integration using a fourth order Runge-Kutta method with a time increment of $\Delta t = \Delta t_{snap}/400$) stage of the model reduction for the various ROMs considered. The times reported under the “offline stage” header are from `Spirit` and are averages over a total of 16 processors of a CEE Remote Graphics Workstation. The times reported under the “online stage” header are calculated in MATLAB using the `tic/toc` command on a CEE Remote Graphics Workstation. The ROM time-integration is performed in serial. Not reported in Table 5.6 is the time required to calculate the POD bases. It takes approximately 100 seconds (on 16 processors of a

shown, as the result is virtually identical to the L^2 ROM for the nonlinear compressible Navier-Stokes equations in the primitive variables.

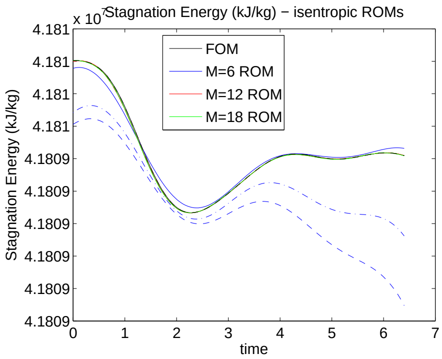


Figure 5.5. FOM vs. ROM stagnation energy as a function of basis size M for isentropic compressible Navier-Stokes ROMs: solid lines = L^2 ROMs, dashed lines = stagnation energy ROMs, dashed-dot lines = stagnation enthalpy ROMs (2D inviscid pulse problem)

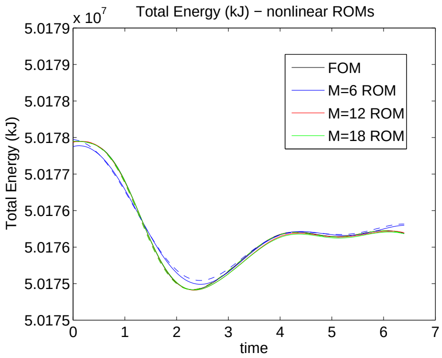


Figure 5.6. FOM vs. ROM total energy as a function of basis size M for nonlinear compressible Navier-Stokes ROMs: solid lines = L^2 ROMs, dashed lines = total energy ROMs (2D inviscid pulse problem)

CEE Remote Graphics Workstation) to calculate a POD basis of size $M = 18$ from $K_{max} = 157$ snapshots.

The CPU times reported in Table 5.6 reveal that, although the nonlinear ROMs are accurate, their offline as well as online efficiency can be improved, especially the efficiency of the nonlinear compressible Navier-Stokes ROMs constructed in the total energy inner product. The focus here is ROM accuracy and implementation verification; efficiency is to be addressed in future work. As discussed in Section 4.4, efficiency of the nonlinear ROMs can be recovered by using interpolation to handle the nonlinear terms appearing in these equations, e.g., DEIM [29], “best points” interpolation [74, 75], or gappy POD [33] (Section 4.4).

Table 5.6. CPU times (in seconds) for offline Galerkin projection and online time-integration stages of the model reduction for the ROMs considered (2D inviscid pulse problem)

POD basis size	ROM	Offline stage: Galerkin projection (`Spirit`, 16 procs)	Online stage: time-integration (MATLAB, 1 proc)
$M = 6$	Linear compr N-S	2.14×10^1	3.99
	Nonlinear isentropic compr N-S	5.87×10^1	2.35×10^1
	ζ–form nonlinear compr N-S	1.03×10^2	1.84×10^1
	Nonlinear L^2 compr N-S	9.45×10^2	5.30×10^1
	Nonlinear total energy compr N-S	2.80×10^3	7.18×10^1
$M = 12$	Linear compr N-S	5.82×10^1	3.27
	Nonlinear isentropic compr N-S	4.67×10^2	2.78×10^1
	ζ–form nonlinear compr N-S	7.63×10^2	2.37×10^1
	Nonlinear L^2 compr N-S	1.40×10^4	9.06×10^1
	Nonlinear total energy compr N-S	4.41×10^4	1.25×10^2
$M = 18$	Linear compr N-S	1.20×10^2	4.74
	Nonlinear isentropic compr N-S	1.28×10^3	4.49×10^1
	ζ–form nonlinear compr N-S	3.56×10^3	3.32×10^1
	Nonlinear L^2 compr N-S	6.64×10^4	1.81×10^2
	Nonlinear total energy compr N-S	2.06×10^5	2.79×10^2

5.5.2 2D viscous pulse in uniform base flow

The second test case is that of a 2D viscous acoustic pressure pulse in the following 2D prismatic domain: $\Omega = (-1,1)^2 \in \mathbb{R}^2$. The base flow is uniform, with the following values: $\bar{p} = 101,325$ Pa, $\bar{T} = 300$ K, $\bar{\rho} = \frac{\bar{p}}{R\bar{T}} = 1.17$ kg/m^3, $\bar{u}_1 = \bar{u}_2 = 0.0$ m/s, and $\bar{c} = 348.0$ m/s, where $\bar{c} \equiv \sqrt{\gamma R\bar{T}}$ is the mean speed of sound. The problem is initialized with a pressure pulse in the middle of the domain:

$$\begin{array}{c} p'(\mathbf{x};0) = 141.9e^{-10(x^2+y^2)}, \\ \rho'(\mathbf{x};0) = \frac{p'(\mathbf{x};0)}{R\bar{T}}, \\ T'(\mathbf{x};0) = 0, \\ u_1'(\mathbf{x};0) = u_2'(\mathbf{x};0) = 0. \end{array} \tag{5.72}$$

In terms of the mean values, the amplitude of the initial pressure pulse (5.72) is $0.001\bar{\rho}\bar{c}^2$. A constant viscosity μ is prescribed such that the Reynolds number for the problem is $Re = 28$, and a constant thermal diffusivity κ is prescribed such that the Prandtl number is $Pr = 0.72$. The Lamé coefficient λ is given by the Stokes' hypothesis (5.37).

The 2D viscous pulse problem differs from the 2D inviscid pulse problem considered earlier (Section 5.5.1) in two key ways: (1) it is viscous, and (2) the high-fidelity simulation is obtained using a nonlinear code, namely `SIGMA CFD`, in DNS mode.

As both `SIGMA CFD` and `Spirit` are 3D codes, a 2D mesh of the domain Ω is converted to a 3D mesh by extruding the 2D mesh in the x_3–direction by one element. The resulting 3D computational grid for this test case is composed of 3362 nodes, cast into 9600 tetrahedral finite elements within `Spirit`. To ensure the solution has no dynamics in the x_3–direction, the following values of the x_3–velocity component are specified: $\bar{u}_3 = 0$, $u_3'(\mathbf{x};0) = 0$. A no-slip and adiabatic wall boundary condition is imposed on the four sides of the domain in the x_1– and x_2–plane and symmetry boundary conditions are imposed for the $x_3 =$ constant boundary in the high-fidelity code.

The high-fidelity simulation from which the ROM is generated is performed until time $T_{max} = 0.01$ seconds. During this simulation, the initial pressure pulse (5.72) reflects from the walls of the domain a number of times and diffuses. Snapshots from the high-fidelity simulation are saved every $\Delta t_{snap} = 5 \times 10^{-5}$ seconds, to yield a total of $K_{max} = 200$ snapshots. These snapshots are used to construct 10 mode POD bases in the various inner products considered. Eight of the ten ROM approaches summarized in Table 5.2 are evaluated, namely those allowing the inclusion of viscosity. The size of the POD basis is determined using an energy criterion (4.6) (see Section 4.1): a basis of size 10 is selected since it is found that 10 modes capture effectively 100% of the snapshot energy (4.6).

Since the base flow for the viscous pulse example is uniform, in (5.1), $\mathbf{G} = \mathbf{0}$ and the $\mathbf{A}_i$ and $\mathbf{K}_{ij}$ matrices are spatially constant, meaning a symmetry ROM for the linearized compressible Navier-Stokes equations is expected to be time-stable and stable in the sense of Lyapunov. Since the dynamics for this example are effectively linear, the linearized ROMs are expected to capture well the solution. A more accurate ROM solution is expected from the nonlinear ROMs, however, as the high-fidelity solver is based on the full nonlinear equations.

As for the 2D inviscid pulse problem, the first property studied is stability of the ROMs considered. All eight ROMs remain stable during the time interval of interest. Both the L^2 and symmetry linearized ROMs are stable based on the eigenvalue criterion (Theorem 3.2.3). Note that this property is not guaranteed *a priori* for linearized L^2 ROMs. Indeed, it is found that an L^2 linearized ROM exhibits instabilities for a variant of this test case (discussed below under the heading "Dimensional vs. non-dimensional ROMs and stability").

Having checked stability, the time history of the modal amplitudes is plotted and compared to the projection of the snapshots onto the POD modes (5.69). The resulting curves for the $M = 10$ mode nonlinear ROM constructed in the L^2 inner product is shown here (Figure 5.7) as a representative of the overall results, which are comparable for all ROMs considered. Agreement is excellent.

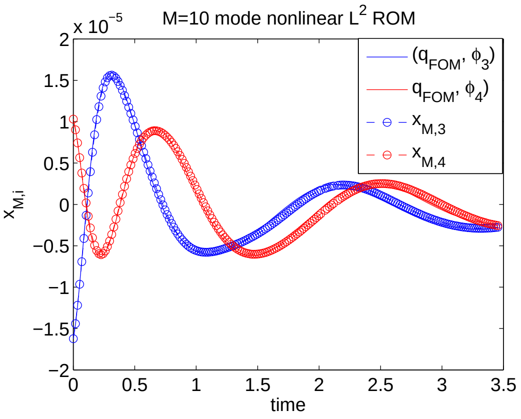


Figure 5.7. Time history of modal amplitudes for $M = 10$ mode nonlinear ROM constructed in the L^2 inner product (2D viscous pulse problem)

Next, relative errors in the ROM solutions with respect to the snapshots are calculated according to the formula (5.70) and reported (Table 5.7). It is found that all eight ROMs[16] evaluated deliver comparable, accurate solutions. Although it was expected for the nonlinear ROMs to deliver a more accurate solution than the linear ROMs, this does not appear to be the case.

As for the 2D inviscid pulse problem, it is interesting to compare the total fluid energy calculated using the ROMs with the total fluid energy calculated using the snapshots. Figure 5.8 shows the total fluid energy calculated using the linear ROMs compared with the total fluid energy calculated using the snapshots. The total fluid energy calculated using the ROMs converges to the total fluid energy calculated using the snapshots with M–refinement.

A somewhat surprising phenomenon is observed when examining the total fluid energy calculated using the nonlinear ROMs: whereas the kinetic energy calculated using the ROMs agrees well with the kinetic energy calculated using the FOM (Figure 5.9(a)), the same cannot be said of the internal energy calculated using the nonlinear compressible Navier-Stokes ROM in the primitive variables

[16] Results for ROMs based on the nonlinear compressible Navier-Stokes physics and the total energy inner product are not shown for the $M = 18$ case, due to unavailability of computational resources (Red Sky cluster downtime) at the time the data for this report were collected.

Table 5.7. ROM relative errors $\mathscr{E}_{rel,\mathbf{q}}$ (2D viscous pulse problem)

Physics	Inner product	$M = 5$	$M = 10$	$M = 15$
Linearized compr N-S	L^2	4.77×10^{-6}	1.42×10^{-6}	1.23×10^{-6}
	symm	5.01×10^{-6}	1.42×10^{-6}	1.24×10^{-6}
Nonlinear compr isentropic N-S	L^2	6.81×10^{-6}	7.18×10^{-6}	7.42×10^{-6}
	stag ene	1.50×10^{-5}	1.23×10^{-5}	1.24×10^{-5}
	stag enth	1.43×10^{-5}	1.22×10^{-5}	1.24×10^{-5}
ζ–form nonlinear compr N-S	L^2	6.14×10^{-6}	2.08×10^{-6}	1.65×10^{-6}
Nonlinear compr N-S	L^2	4.78×10^{-6}	1.42×10^{-6}	1.23×10^{-6}
	tot ene	4.33×10^{-6}	1.51×10^{-6}	–

(Figure 5.9(b)). Curiously enough, the internal energy calculated using ROMs based on the ζ–form of the compressible Navier-Stokes equations matches well the internal energy calculated using the FOM. The reason for the discrepancy is not clear at the present time, and should be investigated in future work.

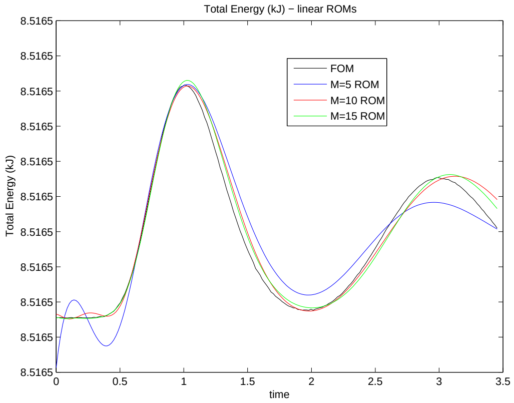
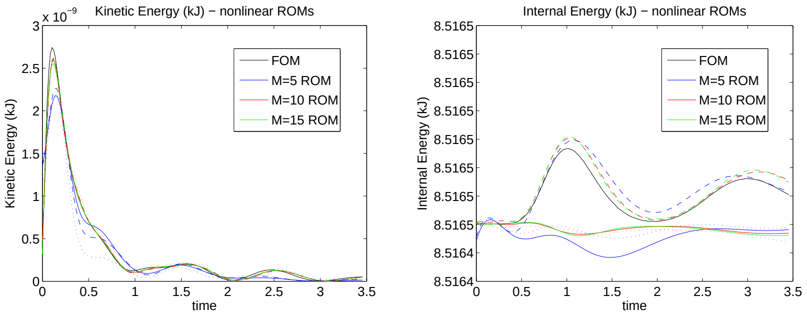


Figure 5.8. FOM vs. ROM total energy for linear ROMs as a function of basis size M: solid lines = L^2 ROMs, dashed lines = symmetry ROMs (2D viscous pulse problem)

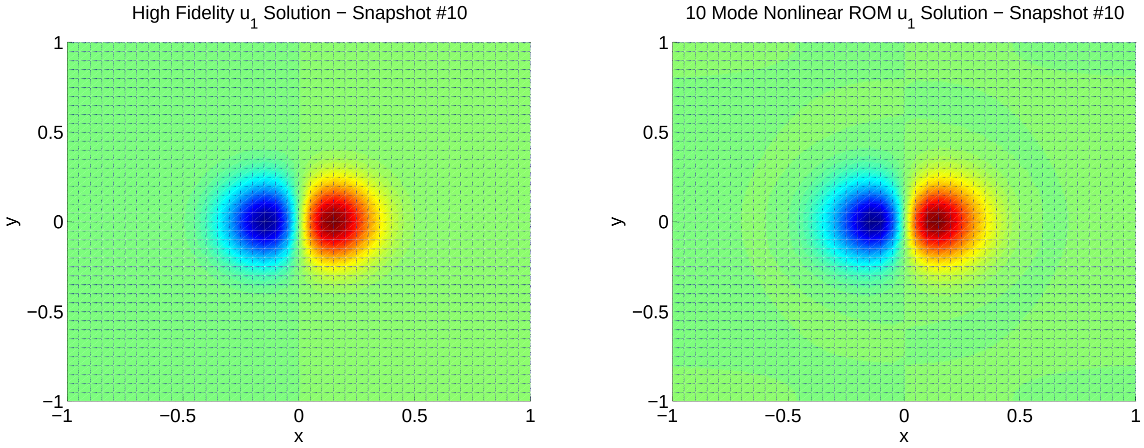
Finally, a visualization of the ROM solution[17] for the u_1 component of the velocity is compared to the u_1 snapshots at two times: the time of the 10^{th} snapshot and the time of the 150^{th} snapshot (Figures 5.10 and 5.11, respectively). Good agreement between the ROM solution and the snapshot is observed at both times.

[17] Without loss of generality, for the 10 mode nonlinear L^2 ROM; solutions computed using the other ROMs were indistinguishable from the solution shown.



(a) Kinetic energy **(b)** Internal energy

Figure 5.9. FOM vs. ROM kinetic and internal energies for nonlinear ROMs as a function of basis size M: solid lines = L^2 ROMs, dashed lines = ζ–form L^2 ROMs, dotted lines = total energy ROMs (2D viscous pulse problem)



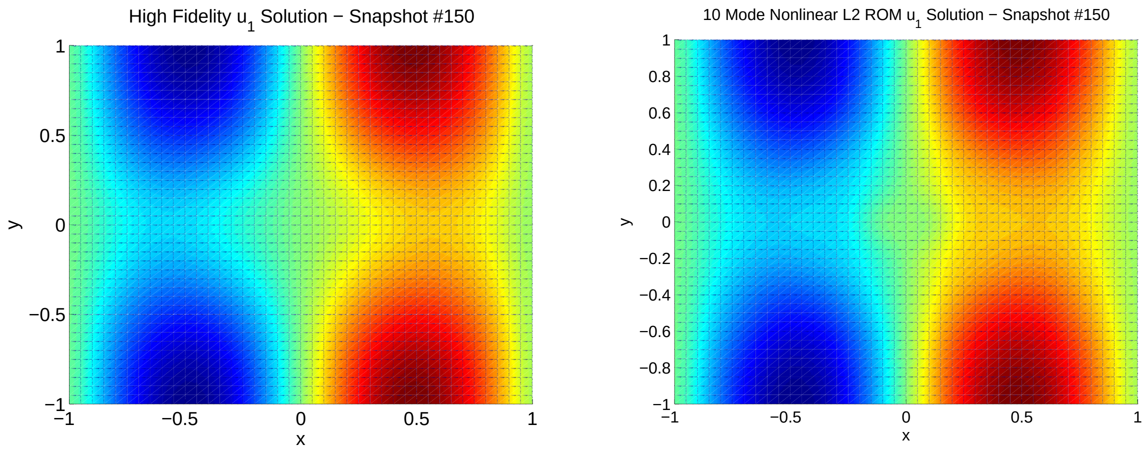
(a) FOM **(b)** 10 Mode Nonlinear L^2 ROM

Figure 5.10. u_1 contours at time of the 10^{th} snapshot (2D viscous pulse problem)

Dimensional vs. non-dimensional ROMs and stability

It is noted earlier that, although the L^2 ROMs for the 2D viscous pulse problem considered above are stable, this is not in general guaranteed *a priori*. Two interesting phenomena are observed while testing the various ROMs proposed here:

- L^2 ROMs based on viscous physics tend to be more stable than ROMs based on inviscid physics, suggesting viscosity has a stabilizing effect.
- L^2 ROMs constructed in non-dimensional variables tend to be more stable than ROMs constructed in dimensional variables, suggesting that bad scaling in the ROM equations can destabilize a ROM.

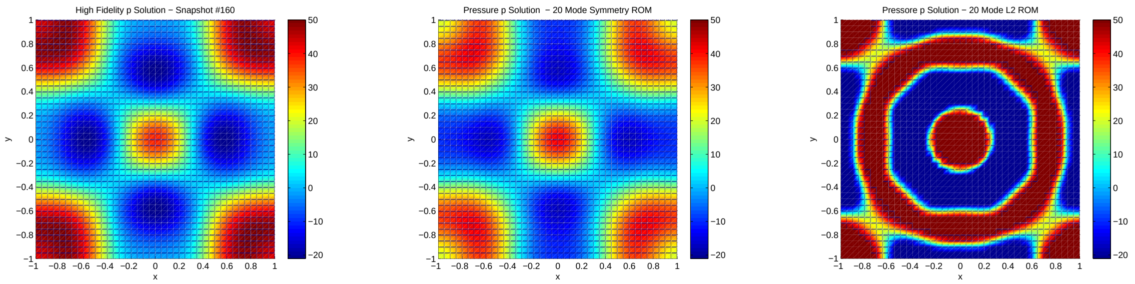


(a) FOM

(b) 10 mode nonlinear L^2 ROM

Figure 5.11. u_1 contours at time of the 150th snapshot (2D viscous pulse problem)

Both tendencies are apparent in an inviscid version of the 2D viscous pulse problem of Section 5.5.2[18]. This problem, detailed in [55], is identical to the 2D viscous pulse problem of Section 5.5.2 except it has no viscosity: $\mu = \lambda = \kappa = 0$. Two ROMs are constructed for this problem: an $M = 20$ mode L^2 ROM and an $M = 20$ mode symmetry ROM based on the linearized compressible Euler equations in dimensional variables. Whereas the latter (symmetry) ROM is stable based on the eigenvalue criterion (Theorem 3.3.3), the former (L^2) ROM exhibits a severe instability, which actually causes the L^2 ROM to blow up by the time of snapshot 160 (Figure 5.12(c)).



(a) FOM

(b) 20 mode linearized compressible Euler symmetry ROM

(c) 20 mode linearized compressible Euler L^2 ROM

Figure 5.12. Pressure field at time of the 160th snapshot (inviscid version of 2D viscous pulse problem)

5.5.3 Viscous laminar cavity

The third test case is that of a Mach 0.6 viscous laminar flow over a cavity in a T–shaped domain $\Omega = [(-6.4244, 10) \times (-1, 10) \times (0, 0.1)] \backslash [(-6.4244, 0) \times (-1, 0) \times (0, 0.1)] \backslash [(2, 10) \times (-1, 0) \times$

[18] Note that this is a different problem than the 2D inviscid pulse problem of Section 5.5.1.

$(0,0.1)]$. The flow conditions are similar to case L2 in [80]. The free stream pressure is 25 Pa, the free stream temperature is 300 K, and the free stream velocity is 208.8 m/s. The free stream pressure is kept low to keep the Reynolds number of the flow low. Two instances of the cavity problem, at two different Reynolds numbers, are considered:

- $Re = 1090.4$ (referred to as $Re \approx 1000$).
- $Re = 1453.9$ (referred to as $Re \approx 1500$).

The viscosity μ is spatially constant and calculated such that the above Reynolds numbers are achieved. The viscosity coefficient λ is calculated from μ using the Stokes' law relation (5.37). The thermal conductivity κ is also constant, calculated such that $Pr = 0.72$. The high-fidelity simulations are performed using the `SIGMA CFD` code in DNS mode. The following boundary conditions are specified in `SIGMA CFD`:

- At the inflow boundary (labeled 1 in Figure A.1), a value of the velocity and temperature that is above the free stream values is specified.
- The flow at the cavity walls (labeled 3 in Figure A.1) is assumed to be adiabatic and to satisfy a no-slip condition.
- The remaining outflow boundaries (labeled 2 in Figure A.1) are open, and a far-field boundary condition that suppresses the reflection of waves into the computational domain is implemented here (Section A.5.1).

The high-fidelity simulation is initialized by setting the flow in the cavity to have a zero velocity, free stream pressure, and temperature. The region above the cavity is initialized to free stream conditions and the flow is allowed to evolve. The discretized domain, illustrated in Figure 5.13, consists of 98,408 nodes, cast as 292,500 tetrahedral finite elements within `Spirit`. The reader can observe that the mesh is structured but non-uniform.

For each of the three Reynolds numbers considered, a total of $K_{max} = 100$ snapshots are collected from `SIGMA CFD`, taken every $\Delta t_{snap} = 1.0 \times 10^{-4}$ seconds, starting at time $t = 5.0 \times 10^{-2}$ seconds until time $T_{max} = 6.0 \times 10^{-2}$ seconds. The snapshot collection begins at time $t = 5.0 \times 10^{-2}$ seconds instead of at time $t = 0$ because it is around this time that a statistically stationary flow regime is reached[19]. The snapshots are used to construct POD bases of size 5, 10, 15 and 20 modes in the various inner products evaluated (for each Reynolds number considered). These bases are used to build POD/Galerkin ROMs using seven of the eight viscous ROM approaches summarized in Table 5.2. Results for ROMs based on the nonlinear compressible Navier-Stokes physics and the total energy inner product are not shown, due to unavailability of computational resources (Red Sky cluster downtime) at the time the data for this report were collected. ROMs based on the isentropic compressible Navier-Stokes equations (5.48) (Section 5.3.3) are also tested, as the flow conditions in the cavity are such that the isentropic assumption is reasonable [81].

[19] Statistically stationary conditions are determined by examining the pressure fluctuations at several locations on the cavity walls.

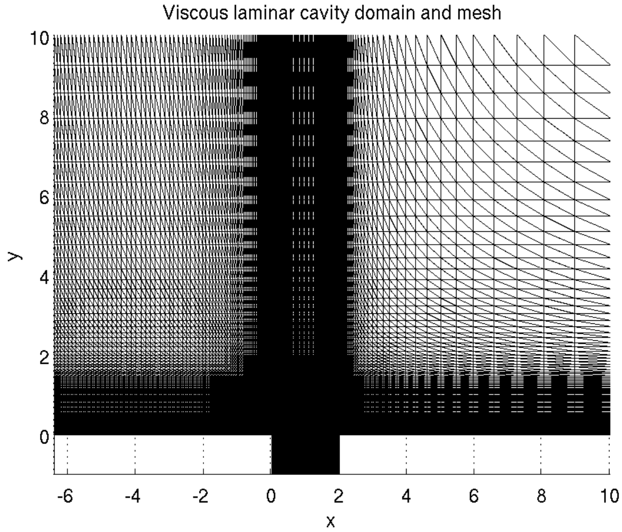


Figure 5.13. Domain and mesh for viscous laminar cavity problem

Linear vs. non-linear ROMs: expected performance

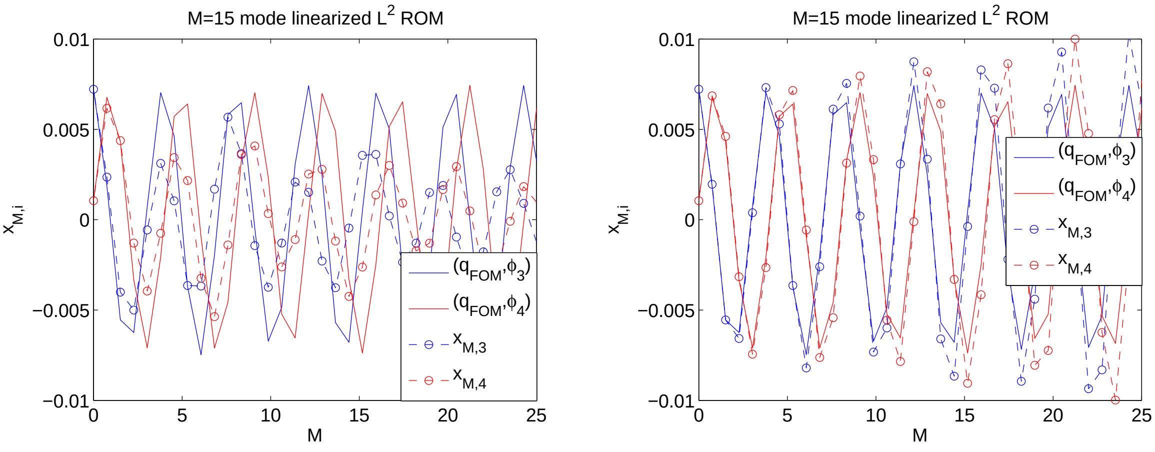
Unlike the pressure pulse problems considered in Sections 5.5.1 and 5.5.2, the viscous laminar cavity problem is inherently nonlinear, and does not possess a natural steady base flow component. It is therefore expected that a ROM based on the full nonlinear physics is needed to accurately represent the inherently nonlinear dynamics that form within the cavity, namely vortices which convect. To give a complete picture, ROMs based on the linearized compressible Navier-Stokes equations constructed using both the L^2 and the symmetry inner product are tested as well. In the linearized ROMs, the base flow is taken to be the average of the snapshots. The viscous work terms are included in the ROM equations (A.2) (see Appendix A.2).

In the local linearization approach outlined in the previous paragraph, the nonlinear dynamics of the flow are captured in the POD modes (Figure 5.15) but not in the equations projected onto these modes. Since a ROM based on linearized equations cannot be expected to reproduce all the nonlinear dynamics in the solution of an inherently nonlinear problem such as the viscous laminar cavity problem, some discussion of what features the linearized ROMs can and cannot be expected to capture is in order. The physics of the cavity flow are determined by the shear layer over the cavity. As the shear layer separates from the leading edge of the cavity, instabilities develop in the separated shear layer. These instabilities grow nonlinearly to form vortices convecting down the shear layer. This process is an inviscid instability growth process and arises due to the nonlinear terms in the convective part of the Navier-Stokes equations. ROMs built using a linearized form of the Navier-Stokes equations cannot be expected to capture accurately this phenomenon. Further downstream, the vortices impinge on the aft wall of the cavity giving rise to pressure waves that are propagated upstream through the free stream and the cavity. Depending upon the Reynolds number (and hence the free stream dynamic pressure), these waves can range from linear to nonlinear. Since the pressure fluctuations on the cavity walls are due to a combination of these waves and those generated by the shear layer vortices, it is expected that only the linear reflected waves should be captured by the linearized ROMs.

Numerical experiments involving the linearized ROMs reveal that, for problems with a non-uniform base flow (such as the viscous laminar cavity), it is critical to include the gradient of the base flow terms (the matrix **G** in (5.22) or (5.26)) in the equations projected onto the POD modes in the model reduction. Failure to include these terms will likely result in a ROM solution with incorrect dynamics (Figure 5.14(a)). In the case that the base flow is uniform (e.g., the 2D inviscid and viscous pulse problems), this matrix vanishes.

Unfortunately, the inclusion of the gradient of the base flow terms (the **G** matrix in (5.22) or (5.26)) may yield a ROM that is not time-stable. If a cavity problem has a feedback loop resonance, such as the viscous laminar cavity problem, it is the *nonlinear* saturation of the shear layer instabilities (i.e., vortex roll-up) that bounds the amplitude of the flow response. One might expect a linear model to be unstable, therefore: the **G** matrix could activate a shear layer instability. Viscosity may work to damp out this instability at low Reynolds numbers.

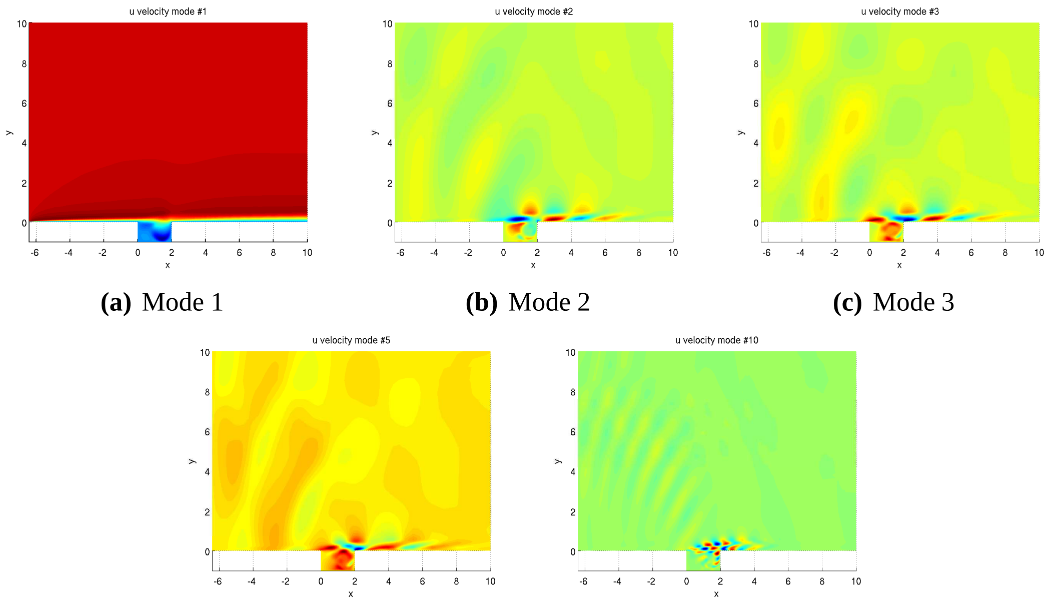
In the case of a viscous problem, note also that the energy-stability result of Corollary 5.2.1 is not valid if the viscous work terms are included in the ROM equations (A.2). A ROM based on equations having these terms may therefore exhibit a non-physical instability (Section A.2). The extension of the energy-stability symmetrization approach presented in Section 5.2 to the linearized compressible Navier-Stokes equations in which the viscous work terms are retained (A.2) is a worthwhile future research endeavor.



(a) **G** matrix excluded

(b) **G** matrix included

Figure 5.14. Effect of including gradient of base flow (**G** matrix in (5.22)) in linearized ROM: time history of modal amplitudes for $M = 15$ mode linearized ROM constructed in the L^2 inner product without (left) and with (right) the **G** matrix (viscous laminar cavity, $Re \approx 1500$)



(a) Mode 1 **(b)** Mode 2 **(c)** Mode 3

(d) Mode 5 **(e)** Mode 10

Figure 5.15. POD modes for $u_1(\mathbf{x},t)$ in L^2 inner product (viscous laminar cavity, $Re \approx 1500$)

Basis size selection

It is both interesting and useful to get an idea of the size of the POD basis that is required to capture the majority of the snapshot energy (4.6) before building and evaluating the various ROMs of interest. Table 5.8 gives the percent snapshot energy captured by the reduced basis modes as a function of M, the basis size, and Re, the Reynolds number[20]. The reader can observe that more modes are required to capture the same percentage of the snapshot energy as the Reynolds number is increased. Still, a relatively small number of modes capture most of the snapshot energy: just 15 modes capture approximately 99% of the snapshot energy for both Reynolds numbers considered. This is due to the fact that the Reynolds numbers considered here are not that high. For higher Reynolds numbers and turbulent flows, it is expected that many more modes ($M = \mathscr{O}(100)$) will be required to capture a large proportion of the snapshot energy.

Table 5.8. POD mode snapshot energy percentages as a function of basis size M and Reynolds number Re (viscous laminar cavity problem)

	$M = 5$	$M = 10$	$M = 15$	$M = 20$
$Re \approx 1000$	94.92%	98.36%	99.06%	99.33%
$Re \approx 1500$	91.01%	98.54%	99.44%	99.69%

[20] Table 5.8 gives the average snapshot fluctuation energy over all the inner products considered. For the nonlinear approaches, the energy in the first mode is omitted, as this represents effectively the energy in the mean flow; Figure 5.15(a).

Figure 5.16 plots the POD basis singular values (eigenvalues of the operator **R** in (4.4); see Section 4.1) on a semilog plot as a function of the basis size M. It is of interest for POD reduced order modeling how fast these singular values decay. The reader can observe a fairly rapid decay of the singular values for both Reynolds numbers considered.

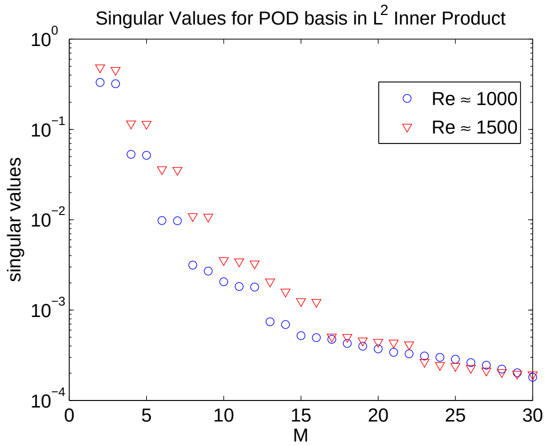


Figure 5.16. POD basis singular values (semilog plot) as a function of basis size M and Reynolds number Re (viscous laminar cavity problem)

ROM accuracy in representing the snapshots

First, the errors in the solutions produced by the ROMs studied are calculated using the formula (5.70) (Table 5.9 for $Re \approx 1000$ and Table 5.10 for $Re \approx 1500$). The reader can observe from Tables 5.9 and 5.10 that all ROMs considered achieve a relative error of between 1% and 10% with respect to the high-fidelity snapshots from which they are constructed. The $Re \approx 1000$ ROMs are in general more accurate than the $Re \approx 1500$ ROMs. Convergence of the ROMs with M–refinement is not observed, which suggests there is a limit to the accuracy the ROMs can achieve for the viscous laminar cavity problem. It is curious to observe that the ROMs based on the linearized physics are in some cases more accurate than the ROMs based on nonlinear physics. This result is unexpected, and leads one to prefer the linear ROMs over the nonlinear ROMs, as they are cheaper to evaluate. It is worthwhile to investigate why the nonlinear ROMs are not more accurate than the linear ROMs for this problem in future work.

Also interesting to note is that the ROMs based on the ζ–form of the compressible Navier-Stokes equations do not deliver a solution with the same accuracy as the ROMs based on the nonlinear compressible Navier-Stokes equations in the primitive variables. This is likely due to the fact that the different formulations, although mathematically equivalent, give rise to different POD modes.

As a sample illustration, the time history of the third and fourth ROM modal amplitudes compared to the projection of the snapshots onto the third and fourth POD modes for the $M = 20$ isentropic ROM constructed in the stagnation energy inner product for the $Re \approx 1500$ instance of the viscous laminar cavity are shown in Figure 5.17. Figure 5.18 shows contours of the u_1–velocity at the

Table 5.9. ROM relative errors $\mathscr{E}_{rel,\mathbf{q}}$ (viscous laminar cavity, $Re \approx 1000$)

Physics	Inner product	$M = 5$	$M = 10$	$M = 15$	$M = 20$
Linearized compr N-S	L^2	3.03×10^{-2}	3.04×10^{-2}	3.85×10^{-2}	4.57×10^{-2}
	symm	2.94×10^{-2}	2.98×10^{-2}	3.73×10^{-2}	5.17×10^{-2}
Nonlinear compr isentropic N-S	L^2	1.92×10^{-2}	2.12×10^{-2}	2.42×10^{-2}	2.21×10^{-2}
	stag ene	3.27×10^{-2}	3.81×10^{-2}	5.37×10^{-2}	3.38×10^{-2}
	stag enth	3.23×10^{-2}	3.53×10^{-2}	4.30×10^{-2}	3.11×10^{-2}
ζ–form nonlinear compr N-S	L^2	3.97×10^{-2}	4.20×10^{-2}	4.24×10^{-2}	4.92×10^{-2}
Nonlinear compr N-S	L^2	2.88×10^{-2}	2.93×10^{-2}	2.86×10^{-2}	2.94×10^{-2}

Table 5.10. ROM relative errors $\mathscr{E}_{rel,\mathbf{q}}$ (viscous laminar cavity, $Re \approx 1500$)

Physics	Inner product	$M = 5$	$M = 10$	$M = 15$	$M = 20$
Linearized compr N-S	L^2	3.03×10^{-2}	3.04×10^{-2}	3.85×10^{-2}	4.57×10^{-2}
	symm	2.96×10^{-2}	2.98×10^{-2}	3.73×10^{-2}	5.17×10^{-2}
Nonlinear compr isentropic N-S	L^2	1.98×10^{-2}	2.79×10^{-2}	2.73×10^{-2}	2.76×10^{-2}
	stag ene	4.18×10^{-2}	4.59×10^{-2}	4.63×10^{-2}	2.31×10^{-2}
	stag enth	4.10×10^{-2}	4.53×10^{-2}	4.33×10^{-2}	2.69×10^{-2}
ζ–form nonlinear compr N-S	L^2	6.27×10^{-2}	9.13×10^{-2}	1.01×10^{-1}	6.45×10^{-2}
Nonlinear compr N-S	L^2	4.53×10^{-2}	5.11×10^{-2}	4.82×10^{-2}	4.21×10^{-2}

times of the 1^{st}, 50^{th} and 100^{th} snapshot for this ROM solution compared with the FOM solution. The figures reveal that the ROM is able to capture accurately the solution dynamics.

ROM accuracy in representing snapshot kinetic and internal energies

Next, the fluid kinetic and internal energies calculated using the linear (Figures 5.19 and 5.22) and nonlinear[21] (Figures 5.21 and 5.24) ROMs are compared with the fluid kinetic and internal energies calculated using the FOM. The fluid specific and stagnation energies (Figures 5.20 and 5.23) calculated using the isentropic ROMs are also compared with the fluid specific and stagnation energies calculated using the FOM. Only the specific energies for the $Re \approx 1500$ isentropic ROMs agree well with that of the snapshots (Figure 5.23(a)). It is not known at the present time why this

[21] Results for the ζ–form ROMs are not shown, as they are indistinguishable from the nonlinear ROM results.

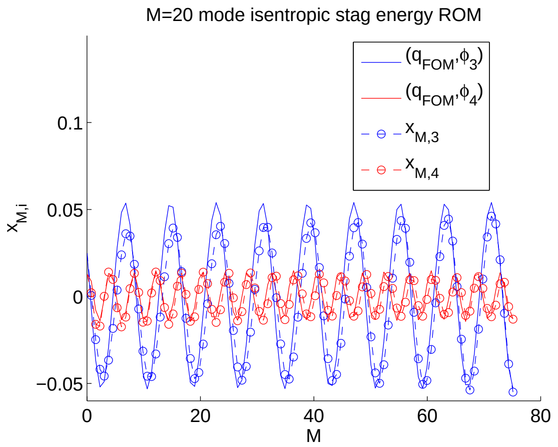


Figure 5.17. Time history of modal amplitudes for $M = 20$ mode nonlinear isentropic ROM constructed in the stagnation energy inner product (viscous laminar cavity, $Re \approx 1500$)

is the case.

ROM efficiency

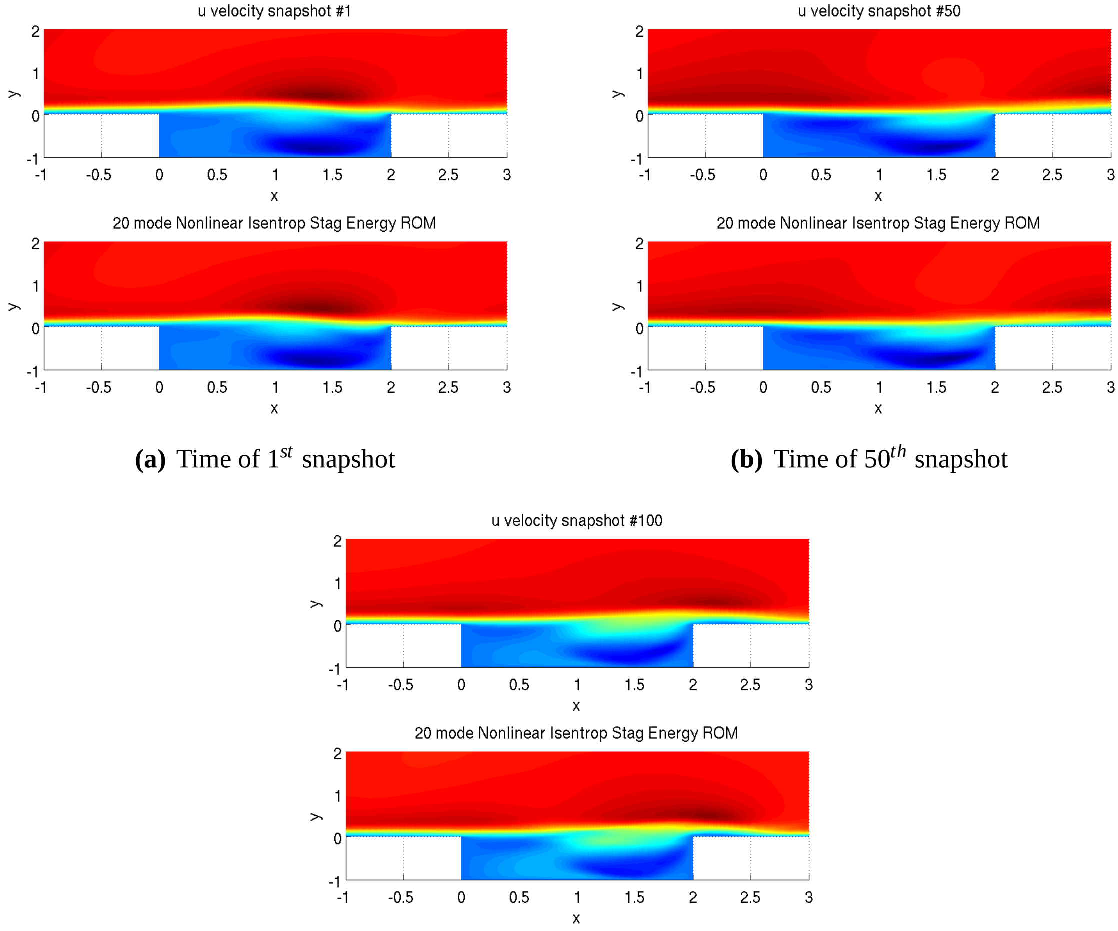
Lastly, efficiency of the various ROMs evaluated is examined. Table 5.11 gives the total CPU times for the offline (quadrature weight calculation and Galerkin projection) and online (time-integration using a fourth order Runge-Kutta scheme with a time increment of $\Delta t = 1.0 \times 10^{-5}$ seconds) stage of the model reduction for the various ROMs. All times are in seconds. The times reported under the "offline stage" header are from `Spirit` and are averages over a total of 96 processors of the Red Sky cluster. The times reported under the "online stage" header are calculated in MATLAB using the `tic/toc` command on a CEE Remote Graphics Workstation, and are for serial (one processor) runs. It takes between 20–30 seconds to calculate a POD basis of size $M = 25$ from $K_{max} = 100$ snapshots in parallel on 96 processors on the Red Sky cluster for all the ROMs.

The reader can observe that both the offline and online stages of the model reduction take significantly more time for the nonlinear compressible Navier-Stokes ROMs than the offline and online stages for ROMs based on the linear, nonlinear isentropic, and nonlinear ζ–form of these equations. The relative expense of the nonlinear compressible Navier-Stokes ROM increases at an exponential rate with M. This is a result of the fact that, for the compressible Navier-Stokes equations in the primitive variables, the computation of higher-order ROM coefficient tensors (in the offline stage) and tensor/vector products involving these higher-order tensors (in the online stage) is required. Efficiency of the nonlinear compressible Navier-Stokes ROMs can be recovered by using interpolation to handle the nonlinear terms appearing in the ROM equations, e.g., DEIM [29], "best points" interpolation [74, 75], or gappy POD [33] (Section 4.4).

Additional numerical results for the viscous laminar cavity, in which the ROM is constructed from a total of $K_{max} = 500$ snapshots and run until time $T_{max} = 0.1$ seconds, can be found in Appendix

Table 5.11. CPU times (in seconds) for offline Galerkin projection and online time-integration stages of the model reduction for the ROMs considered (viscous laminar cavity problem)

POD basis size	ROM	Offline stage: Galerkin projection (`Spirit`, 96 procs)	Online stage: time-integration (MATLAB, 1 proc)
$M = 5$	Linear compr N-S	1.46	2.09
	Nonlinear isentropic compr N-S	1.31	2.38
	ζ–form nonlinear compr N-S	1.74	2.93
	Nonlinear L^2 compr N-S	7.17	1.01×10^1
$M = 10$	Linear compr N-S	1.32	2.76
	Nonlinear isentropic compr N-S	8.81	3.48
	ζ–form nonlinear compr N-S	1.13×10^1	3.67
	Nonlinear L^2 compr N-S	8.31×10^1	8.31
$M = 15$	Linear compr N-S	2.80	2.96
	Nonlinear isentropic compr N-S	1.92×10^1	5.01
	ζ–form nonlinear compr N-S	2.98×10^1	4.51
	Nonlinear L^2 compr N-S	4.01×10^2	1.65×10^1
$M = 20$	Linear compr N-S	4.22	3.14
	Nonlinear isentropic compr N-S	4.82×10^1	5.33
	ζ–form nonlinear compr N-S	7.37×10^1	5.56
	Nonlinear L^2 compr N-S	2.15×10^3	2.42×10^1



(a) Time of 1^{st} snapshot

(b) Time of 50^{th} snapshot

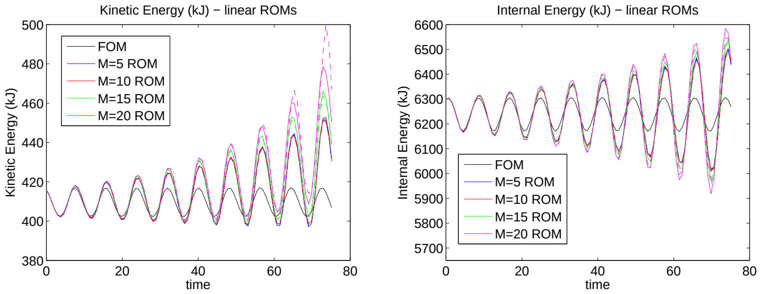
(c) Time of 100^{th} snapshot

Figure 5.18. u_1–velocity contours at several times for FOM compared to $M = 20$ nonlinear isentropic ROM constructed in the stagnation energy inner product (viscous laminar cavity, $Re \approx 1500$)

A.6.

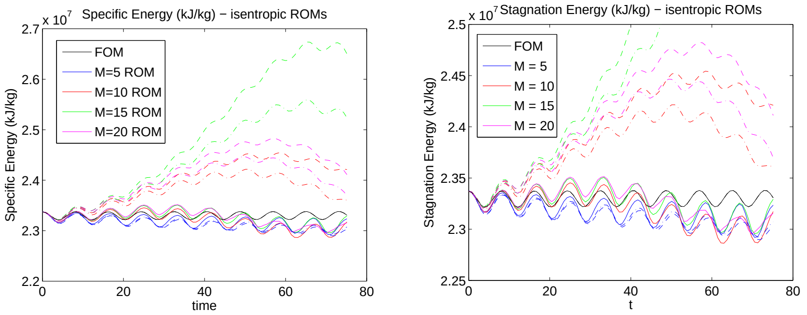
5.6 Summary

In this chapter, the energy-stability preserving model reduction approach developed specifically for the equations of linearized compressible inviscid flow in [20, 57] is generalized: for ROMs constructed using the continuous projection approach, it is shown that a transformation of a generic PDE system of the hyperbolic or incompletely parabolic type leads to a stable formulation of the Galerkin ROM for this system. It is then shown that, for many linear PDE systems, the said transformation is induced by a special inner product, referred to as the “symmetry inner product”. If the Galerkin projection step of the model reduction procedure is performed in this inner product, the



(a) Kinetic energy **(b)** Internal energy

Figure 5.19. FOM vs. ROM kinetic and internal energy for linear ROMs as a function of basis size M: solid lines = L^2 ROMs, dashed lines = symmetry ROMs (viscous laminar cavity, $Re \approx 1000$)

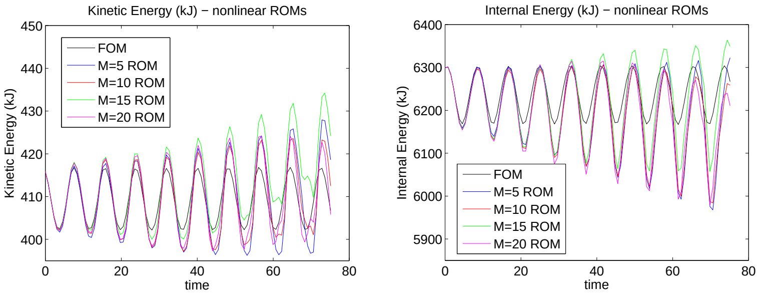


(a) Specific energy **(b)** Stagnation energy

Figure 5.20. FOM vs. ROM specific and stagnation energy for isentropic ROMs as a function of basis size M: solid lines = L^2 ROMs, dashed lines = stagnation energy ROMs, dashed-dot lines = stagnation enthalpy ROMs (viscous laminar cavity, $Re \approx 1000$)

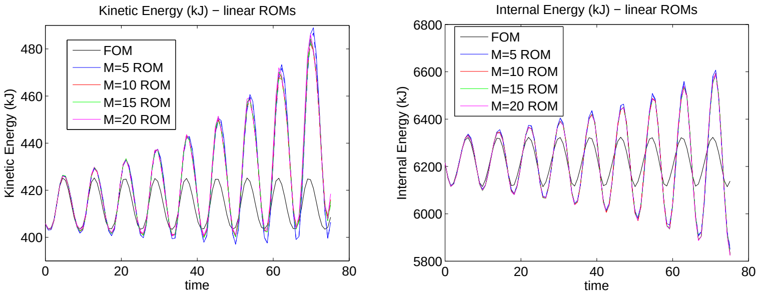
resulting ROM is guaranteed to satisfy certain stability bounds regardless of the reduced basis employed. Examples of the symmetry inner product for linear systems of PDEs that commonly arise in modeling applications (e.g., the linearized compressible Euler and Navier-Stokes equations; the wave equation; the linearized shallow water equations) are given.

Next, approaches for building energy-stable ROMs for the equations of nonlinear compressible flow, the PDEs of interest in the targeted compressible captive-carry application, are explored. Three forms of the nonlinear compressible Navier-Stokes equations are considered: the full compressible Navier-Stokes equations in the primitive variables, the ζ–form of the full compressible Navier-Stokes equations, and the isentropic compressible Navier-Stokes equations. An inner product that induces the integrated stagnation energy or stagnation enthalpy of the flow, proposed originally by Rowley *et al.* in [81] for the isentropic compressible Navier-Stokes equations, is reviewed. This inner product motivates the derivation of a *new* transformation and corresponding energy inner product for the full compressible Navier-Stokes equations, presented for the first time in this report. The proposed new inner product induces the total energy of the fluid system, a physically meaningful quantity that is conserved for problems in which no external forcing is applied. If the



(a) Kinetic energy **(b)** Internal energy

Figure 5.21. FOM vs. ROM kinetic and internal energy for nonlinear ROMs as a function of basis size M: solid lines = L^2 ROMs (viscous laminar cavity, $Re \approx 1000$)



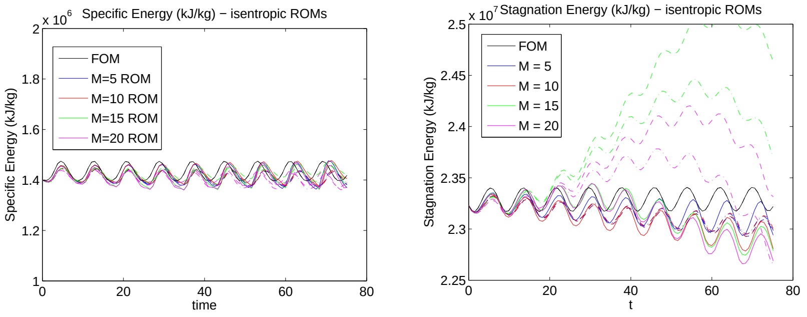
(a) Kinetic energy **(b)** Internal energy

Figure 5.22. FOM vs. ROM kinetic and internal energy for linear ROMs as a function of basis size M: solid lines = L^2 ROMs, dashed lines = symmetry ROMs (viscous laminar cavity, $Re \approx 1500$)

projection step of the model reduction is performed in the total energy inner product, the resulting Galerkin ROM should preserve the stability of an equilibrium point at the origin for the governing equations.

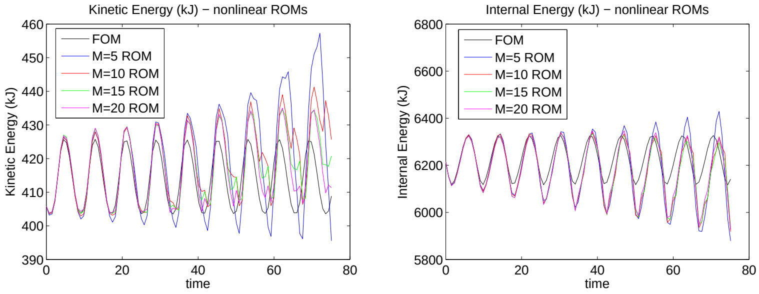
A stability-preserving discrete implementation of the continuous Galerkin projection method is developed. A parallel C++ code, `Spirit`, that was developed as a part of this LDRD project and builds ROMs for various linearized as well as nonlinear compressible flow physics using continuous Galerkin projection in several inner products is described. The code uses vector data structures and parallel eigensolvers from the Trilinos project [49] and the quadrature routines from the `libmesh` finite element library [63]. As a stand-alone code, `Spirit` can be synchronized with any high-fidelity solver that can output a mesh and snapshot data stored at the nodes of this mesh, including the Sandia in-house LES flow solver used in the captive-carry application, `SIGMA CFD`.

The performance of the various ROMs described in this chapter is evaluated on three test cases: a 2D inviscid pulse problem, a 2D viscous pulse problem, and a viscous laminar cavity problem at several Reynolds numbers. For all three tests, emphasis is placed on reproducing a given CFD solution for a single set of flow conditions in a stable and accurate fashion, as this is a prerequisite to using the ROM in a predictive setting. The first two test cases have effectively linear dynamics,



(a) Specific energy **(b)** Stagnation energy

Figure 5.23. FOM vs. ROM specific and stagnation energy for isentropic ROMs as a function of basis size M: solid lines = L^2 ROMs, dashed lines = stagnation energy ROMs, dashed–dot lines = stagnation enthalpy ROMs (viscous laminar cavity, $Re \approx 1500$)



(a) Kinetic energy **(b)** Internal energy

Figure 5.24. FOM vs. ROM kinetic and internal energy for nonlinear ROMs as a function of basis size M: solid lines = L^2 ROMs (viscous laminar cavity, $Re \approx 1500$)

whereas the third case has inherently nonlinear dynamics, namely vortices that convect, and is posed on the geometry of interest in the targeted application (an open T–shaped cavity).

For the first two test cases, all linear as well as nonlinear ROMs considered perform comparably, and capture well the snapshots from which they are constructed. For these problems, the linearized ROMs are sufficient and recommended due to their lower computational cost. The proposed energy-stable ROMs do indeed possess better numerical stability properties than their L^2 ROM counterparts, which lack in general an *a priori* stability guarantee. There is, however, an unexpected discrepancy between the internal energy calculated using the nonlinear ROMs for the 2D viscous pulse test case and the internal energy calculated using the snapshots.

For the third test case, the viscous laminar cavity problem, all linear and nonlinear ROM approaches deliver a solution with approximately the same accuracy. The ROM solutions are reasonable but convergence of the ROM solution with M–refinement is not in general observed, and the ROMs do not accurately reproduce the fluid total energy for the entire time interval considered. Further numerical experiments performed for the viscous laminar cavity (Appendix A.6) demonstrate that the POD/Galerkin approach may yield ROMs that are unreliable for longer time cavity simulations, even when an energy inner product is employed to do the Galerkin projection. The method seems highly dependent on the set of snapshots employed in calculating the POD basis: when more snapshots are included in the basis calculation, many of the ROMs exhibit unexpected instabilities (Appendix A.6).

5.6.1 Prospects for future work

Further code verification is recommended to ensure a bug-free implementation of the proposed energy inner products in `Spirit`. It may be possible to improve the accuracy of the proposed nonlinear ROMs for compressible cavity problems by applying some recently proposed ideas, e.g., through the incorporation of fine-scales into the ROM basis [5, 76, 22, 15, 92], through the addition of LES turbulence closure terms to the ROM equations [94], through the incorporation of boundary condition terms in the ROM equations [39, 57] (Appendix A.5), and/or through an adaptive *h*–refinement of the ROM basis [26]. It may also be worthwhile to see if the situation can be improved by devising specialized snapshot collection/sampling methods (e.g., methods based on “optimal” sampling strategies [70]; methods in which low-energy modes are included in the POD basis [82, 15]). It is conjectured that using a set of snapshots spaced closer together in time (i.e., with a smaller Δt_{snap}) to construct the POD basis may yield a more accurate and stable ROM for the viscous laminar cavity problem [14]. For problems with a periodic limit-cycle solution like the viscous laminar cavity problem, using a snapshot set representing just one period of the solution to calculate the POD basis may also give rise to a better ROM [14]. Lastly, it has been argued that POD modes corresponding to complex conjugate pair singular values of the snapshot matrix (eigenvalues of the operator $\mathbf{R}$ in (4.4); see Section 4.1) should be retained in pairs in the POD basis truncation [82, 14]. This strategy is not employed in all the experiments summarized in this report, and may be considered in the future.

An additional concern worth addressing in future work involves the efficiency of ROMs constructed using the proposed total energy inner product for the nonlinear compressible Navier-Stokes equations. As discussed earlier in this chapter, the formulation requires the projection of high-order polynomial terms in the offline stage of the model reduction, and the evaluation of tensor/vector products involving large ROM coefficient tensors in the online stage of the model reduction, which can invalidate the term *reduced* order model except for very small M. This difficulty can be overcome through the incorporation of interpolation methods, e.g., the discrete empirical interpolation method (DEIM) [29], “best points” interpolation [74, 75], or gappy POD [33], into the ROM approach. A theoretical and numerical study of the effect of such interpolation methods on ROM accuracy and stability would be required.

Per the discussion in Chapter 1, Large Eddy Simulations of the captive-carry scenario performed using the `SIGMA CFD` code can take on the order of weeks to complete, as long run-times and fine meshes are required. The ultimate goal is to use in place of these high-fidelity simulations a ROM constructed from some set of high-fidelity snapshots. Achieving this goal requires the implementation of model reduction capabilities in `SIGMA CFD`, a task planned for the near future. In order to be useful, the ROM approach selected for this implementation must:

- *Be capable of predicting the cavity dynamics beyond the time-horizon up to which the snapshots were collected and for a different set of flow conditions.* A model reduction method that delivers ROMs which are robust with respect to parameter changes, e.g., different flow conditions, is required. A survey of the literature suggests that a general fast and robust method for adapting pre-computed ROMs to a new set of physical or modeling parameters is still lacking at the present time, although some promising ideas have been proposed [6, 26].

- *Be reliable and fast enough to be used for on-the-spot analysis (e.g., uncertainty quantification, flow control).* The ROM-based analysis of particular interest to Sandia is in the area of uncertainty quantification (UQ). Here, the idea would be to replace the high-fidelity model in a sampling-based Markov Chain Monte Carlo (MCMC) algorithm with a ROM, which, unlike the high-fidelity simulation, can be queried many times in real or near-real time at a low computational complexity. For a survey of ROM-based UQ methods for the captive carriage application, the reader is referred to [18]. Also of interest for the captive carriage application is ROM-based flow control, with is discussed briefly in Appendix A.11 and may be pursued further in future research.

The numerical tests performed as a part of this LDRD project and summarized in this chapter suggest the POD/Galerkin approach may not be the best choice of model reduction method for the compressible cavity problems simulated using `SIGMA CFD`. Other approaches, e.g., the GNAT method [27], have yielded promising results for problems such as these, and may be a better option.

Chapter 6

Stable ROMs via discrete projection

Chapter 5 focuses on the construction of energy-stable projection-based ROMs for linear as well as nonlinear conservation laws using continuous projection. Attention is now turned to developing stable projection-based ROMs using the discrete projection approach. Only linear time-invariant (LTI) systems (Section 4.5) are considered, as a first step towards the more general nonlinear case. One advantage of the discrete projection approaches described in this chapter over the continuous projection approaches of Chapter 5 is that the discrete approaches can be implemented in a "black-box" fashion, that is, without requiring access to the governing PDEs or high-fidelity code discretizing these PDEs. A consequence of this property is that the approaches are not limited to a particular physics set (e.g., the compressible captive-carry problem); they can be applied to *any* problem in *any* application.

In Section 6.1, a discrete counterpart of the symmetry inner product (developed in Section 5.2 for linearized conservation laws) is derived. This inner product is termed the "Lyapunov inner product", and was first proposed in the context of model reduction by Rowley *et al.* in [81], but has not been tested extensively at the present time. A numerical study of the performance of ROMs constructed in this inner product is undertaken here.

In Section 6.2, a *new* approach for building stable projection-based ROMs for LTI systems is developed. The approach, termed "ROM stabilization via eigenvalue reassignment", stabilizes unstable ROMs through an *a posteriori* post-processing step applied to the algebraic ROM system. This stabilization step consists of a reassignment of the eigenvalues of the ROM system matrix and is motivated by ideas from control theory [11, 96]. Accuracy of the stabilized ROM is ensured through the formulation and numerical solution of a constrained nonlinear least-squares optimization problem in which the error in the ROM output is minimized.

For more detail on the methods described in this chapter, the reader is referred to the following journal articles and SAND reports, written as a part of this LDRD project: [59, 60, 61].

In each of the main section of this chapter, Section 6.1 and Sections 6.2, the methods described therein are tested on the same two test cases, referred to as the international space station (ISS) benchmark and the electrostatically actuated beam benchmark. The avoid unnecessary repetition, the test cases are summarized here. For both examples, the error in the ROM output relative to the full order model output is computed and reported. This value is calculated using the formula

$$\mathscr{E}_{rel,\mathbf{y}} = \sqrt{\frac{\sum_{i=1}^{K_{max}} |\mathbf{y}_{QN}(t_i) - \mathbf{y}_{QM}(t_i)|}{\sum_{i=1}^{K_{max}} |\mathbf{y}_{QN}(t_i)|}}. \tag{6.1}$$

Here the symbol K_{max} denotes the integer such that $T_{max} = K_{max}\Delta t_{snap}$, where T_{max} is the maximum time until which the ROM is run. The notation $|\cdot|$ in (6.1) denotes the absolute value, which evaluates to a scalar for the numerical examples considered herein, as they both have one output ($Q = 1$ in (4.18)).

Benchmark #1: International space station (ISS)

The first numerical example involves a structural model of the Russian service module component of the international space station (ISS) [10]. This service module is a large flexible structure whose dynamics can be described using a linearized form of the equations of motion (a second order PDE system). Written in first order LTI form, the model consists of a system of the form (4.18) with $N = 270$. The matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ defining (4.18) are downloaded from the ROM benchmark repository [28]. The matrix $\mathbf{A}$ is sparse, as it comes from a finite element discretization. A single output is considered, corresponding to the first row of the matrix $\mathbf{C}$. It is verified that the FOM system is stable: the maximum real part of the eigenvalues of $\mathbf{A}$ is -0.0031.

Benchmark #2: Electrostatically actuated beam

The second numerical example is that of an electrostatically actuated beam. Applications for this model include microelectromechanical systems (MEMS) devices such as electromechanical radio frequency (RF) filters [68]. Given a simple enough shape, these devices can be modeled as one-dimensional beams embedded in two or three dimensional space. The beam considered here is supported on both sides, and has two dofs: the deflection perpendicular to the beam (the flexural displacement), and the rotation in the deformation plane (the flexural rotation). The equations of motion are determined from a Lagrangian formulation. It is assumed that the beam deflection is small, so that geometric nonlinearities can be neglected. The resulting linear PDEs are discretized using the finite element method following the approach presented in [54, 68]. The result of this discretization is a second order linear semi-discrete system of the form:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{E}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) &= \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned} \tag{6.2}$$

where $\ddot{\mathbf{x}} \equiv \frac{\partial^2 \mathbf{x}}{\partial t^2}$. The input matrix $\mathbf{B}$ corresponds to a loading of the middle node of the domain, and $\mathbf{y}(t)$ is the flexural displacement at the middle node of the domain. The damping matrix $\mathbf{E}$ is taken to be a linear combination of the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{K}$:

$$\mathbf{E} = c_M\mathbf{M} + c_K\mathbf{K}, \tag{6.3}$$

with $c_M = 10^2$ and $c_K = 10^{-2}$. Letting $\dot{\hat{\mathbf{x}}}(t) \equiv \mathbf{x}(t)$, the second order system (6.2) can be written as the following first order system:

$$\begin{aligned} \begin{pmatrix} \mathbf{E} & \mathbf{M} \\ \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{x}}(t) \\ \dot{\tilde{\mathbf{x}}}(t) \end{pmatrix} + \begin{pmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \tilde{\mathbf{x}}(t) \end{pmatrix} &= \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) &= \begin{pmatrix} \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \tilde{\mathbf{x}}(t) \end{pmatrix}, \end{aligned} \tag{6.4}$$

or

$$\begin{aligned} \dot{\mathbf{z}}(t) &= \mathbf{A}\mathbf{z}(t) + \tilde{\mathbf{B}}\mathbf{u}(t) \\ \mathbf{y}(t) &= \tilde{\mathbf{C}}\mathbf{z}(t), \end{aligned} \tag{6.5}$$

where $\mathbf{z}(t) \equiv \begin{pmatrix} \mathbf{x}(t) \\ \tilde{\mathbf{x}}(t) \end{pmatrix} \in \mathbb{R}^{2N}$ and

$$\mathbf{A} \equiv \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{E} \end{pmatrix}, \quad \tilde{\mathbf{B}} \equiv \begin{pmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{B} \end{pmatrix}, \quad \tilde{\mathbf{C}} \equiv \begin{pmatrix} \mathbf{C} & \mathbf{0} \end{pmatrix}. \tag{6.6}$$

The matrices **M** and **K** in (6.2), which are sparse, as they come from a finite element discretization, are downloaded from the Oberwolfach model reduction benchmark collection [3]. These global matrices are then disassembled into their local counterparts, and reassembled to yield a discretization of any desired size. In the full order model for which results are reported here, $N = 5000$, so (6.5) has 10,000 dofs. It is verified that the full order system is stable: the maximum real part of the eigenvalues of **A** is -0.0016. It is worthwhile to note that, unlike for the ISS example, the matrix **A** that defines the system (6.5) for the electrostatically actuated beam test case is *not* sparse. In particular, it is straightforward to see from (6.5) that this matrix is of the form $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1, & \mathbf{A}_2 \end{pmatrix}^T$ where $\mathbf{A}_1 \in \mathbb{R}^{N\times N}$ is sparse, but $\mathbf{A}_2 \in \mathbb{R}^{N\times N}$ is dense.

6.1 Stability-preserving Lyapunov inner product

In Chapter 5, a method for constructing energy-stable ROMs via continuous projection of a system of PDEs was presented. The discussion in Chapter 5 motivates the following question: can the energy inner product be determined in a black-box fashion for any given full order model system? It is shown in the present section that there is a discrete counterpart of the symmetry inner product (recall that the symmetry inner product is the energy inner product for linear conservation laws; Section 5.2), first derived by Rowley *et al.* [81] and termed the "Lyapunov inner product" herein. Although the Lyapunov inner product has appeared in several publications [81, 85, 8], to the authors' knowledge, a numerical study of the properties and performance of POD ROMs constructed in the Lyapunov inner product is lacking from the literature at the present time, and one of the contributions of this work.

Consider an LTI system of the form (4.18). Suppose the system is stable in the sense of Lyapunov, i.e., all eigenvalues of the matrix **A** have non-positive real parts (Theorem 3.2.3). Since **A** is stable, there exists a Lyapunov function for

$$\dot{\mathbf{x}}_N(t) = \mathbf{A}\mathbf{x}_N(t). \tag{6.7}$$

In particular,

$$V(\mathbf{x}_N) = \mathbf{x}_N^T \mathbf{P} \mathbf{x}_N, \tag{6.8}$$

is a Lyapunov function for (6.7), where **P** is the solution of the following Lyapunov equation:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{Q}. \tag{6.9}$$

Here, **Q** is some positive-definite matrix [11]. A positive definite solution **P** to (6.9) exists provided **A** is stable. Moreover, if **Q** is symmetric, **P** is symmetric as well. Given **A** and **Q**, a solution to the Lyapunov equation (6.9) can be obtained, for instance, using the `lyap` function in the MATLAB control toolbox [2]:

```
P = lyap(A′, Q, [], speye(N, N)).
```

Assume the system (6.7) is stable and a positive-definite symmetric **P** has been computed from (6.9). Since **P** is symmetric positive-definite, the following

$$\left(\mathbf{x}_N^{(1)}, \mathbf{x}_N^{(2)}\right)_{\mathbf{P}} \equiv \mathbf{x}_N^{(1)T} \mathbf{P} \mathbf{x}_N^{(2)}, \tag{6.10}$$

defines an inner product. Let $\mathbf{\Phi}_M$ be a reduced basis of size M, so that

$$\mathbf{x}_N(t) \approx \mathbf{\Phi}_M \mathbf{x}_M(t), \tag{6.11}$$

where $\mathbf{x}_M(t)$ denotes the ROM solution. Theorem 6.1.1 (summarized here from [81] to keep this report self-contained) shows that (6.10) is the energy inner product for this system.

Theorem 6.1.1 (from [81]): Assume the linear full order system (6.7) is stable. Suppose a ROM for (6.7) is constructed via a Galerkin projection in the $(\cdot,\cdot)_{\mathbf{P}}$ inner product (6.10), to yield the following reduced linear system:

$$\dot{\mathbf{x}}_M = \mathbf{\Phi}_M^T \mathbf{P} \mathbf{A} \mathbf{\Phi}_M \mathbf{x}_M, \tag{6.12}$$

where it has been assumed that the basis $\mathbf{\Phi}_M$ has been constructed to be orthonormal in the $(\cdot,\cdot)_{\mathbf{P}}$ inner product, i.e., $\mathbf{\Phi}_M^T \mathbf{P} \mathbf{\Phi}_M = \mathbf{I}_M$ where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. Then, the ROM (6.12) is energy-stable, time-stable and stable in the sense of Lyapunov.

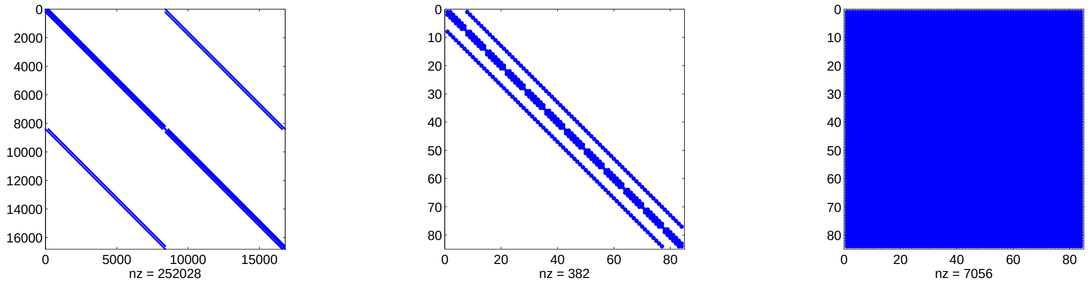
Proof. It is shown that the energy $E_M \equiv \frac{1}{2}||\mathbf{x}_M||_2^2$ of the ROM system (6.12) is non-increasing:

$$\begin{aligned}
\frac{dE_M}{dt} &= \frac{1}{2}\frac{d}{dt}(\mathbf{x}_M, \mathbf{x}_M)_2 \\
&= \mathbf{x}_M^T \dot{\mathbf{x}}_M \\
&= \mathbf{x}_M^T \mathbf{\Phi}_M^T \mathbf{P} \mathbf{A} \mathbf{\Phi}_M \mathbf{x}_M \\
&= \mathbf{x}_M^T \mathbf{\Phi}_M^T \left(\tfrac{1}{2}\mathbf{P}\mathbf{A} + \tfrac{1}{2}\mathbf{P}^T \mathbf{A}\right) \mathbf{\Phi}_M \mathbf{x}_M \\
&= \mathbf{x}_M^T \mathbf{\Phi}_M^T \left(\tfrac{1}{2}\mathbf{P}\mathbf{A} + \tfrac{1}{2}\mathbf{A}^T \mathbf{P}\right) \mathbf{\Phi}_M \mathbf{x}_M \\
&= -\tfrac{1}{2}\mathbf{x}_M^T \mathbf{\Phi}_M^T \mathbf{Q} \mathbf{\Phi}_M \mathbf{x}_M \\
&< 0,
\end{aligned} \tag{6.13}$$

since $\mathbf{Q} > \mathbf{0}$. It follows that (6.12) is time-stable, stable in the sense of Lyapunov and energy-stable (Section 3).

□

The Lyapunov inner product (6.10) is a discrete counterpart of the continuous symmetry inner product (5.16). This inner product can be employed to construct stable Galerkin ROMs for LTI systems of the form (4.18) using discrete projection. An interesting question that arises is whether the matrix **P** defining the Lyapunov inner product (6.10) is related in some way to the matrix **W** (5.67) that is used to perform the continuous projection in the symmetry inner product. In general, the answer is no. In particular, **W** is by construction a sparse matrix (Figure 6.1(a)), whereas **P** may be dense even if **A** is sparse. This is clear from Figures 6.1(b) and (c), which show (respectively) the sparsity pattern of a sample **A** matrix[1], and its corresponding **P** matrix.



(a) W **(b) A** **(c) P**

Figure 6.1. Sparsity structure of representative **P** matrix for a given sparse **A** matrix compared to sparsity structure of representative **W** matrix

One downside of the Lyapunov inner product is that the matrix **P** which defines this inner product is admittedly expensive to compute: the cost of solving the Lyapunov equation (6.9) requires $\mathscr{O}(N^3)$ operations. As a consequence, the Lyapunov inner product has the same downside as another model reduction approach with an *a priori* stability guarantee, namely balanced truncation [73, 42]: it may not be practical to compute the matrix **P** defining the Lyapunov inner product for very large systems.

It is worthwhile to note that computing **P** (6.9) is less computationally intensive than reducing a system using balanced truncation, which requires the solution of *two* Lyapunov equations for the so-called observability and reachability Gramians *and* the factorizations of these Gramians [73, 42] (see Appendix A.7). The computational cost of calculating the weighting matrix that defines the Lyapunov inner product relative to the computational cost of reducing a system using balanced truncation is studied numerically in Section 6.1.1. Note that it can be shown that the balanced truncation algorithm may be viewed as a projection algorithm in a special Lyapunov inner product [81]. A proof uncovering this connection is given in Appendix A.8.

As observed earlier for the symmetry inner product, it is clear from (6.12) that the Galerkin projection of the system (6.7) in the Lyapunov inner product (6.10) can be viewed as a Petrov-Galerkin projection of this system in the regular L^2 inner product, with the reduced test basis given by

[1] The **A** matrix whose sparsity pattern is shown in Figure 6.1(b) is the “PDE example” in the SLICOT model reduction benchmark repository [28].

$\mathbf{\Psi}_M = \mathbf{P}\mathbf{\Phi}_M$, where $\mathbf{\Phi}_M$ is the reduced trial basis.

6.1.1 Numerical experiments

The performance of POD/Galerkin ROMs constructed in the Lyapunov inner product is now evaluated on two examples: the international space station problem, and the electrostatically actuated beam problem, introduced at the beginning of this chapter.

International space station (ISS) test case

To generate the snapshots from which the POD bases are constructed, the full order model is solved using a backward Euler time integration scheme with an initial condition of $\mathbf{x}_N(0) = \mathbf{0}$ and $\mathbf{u}_P(t) = (1 \times 10^4)\delta_{t=0}$. That is, at time $t = 0$, an impulse of magnitude 1×10^4 is applied. A total of $K_{max} = 2000$ snapshots are collected, every $\Delta t_{snap} = 5 \times 10^{-5}$ seconds, until time $T_{max} = 0.1$ seconds. These snapshots are used to construct POD bases of sizes $M = 5$, 10, 20, 30, and 40. For each M, a POD basis is computed using the L^2 inner product, as well as the Lyapunov inner product (6.10). The matrix $\mathbf{P}$ defining the inner product (6.10) is obtained using the `lyap` function in MATLAB's control toolbox with $\mathbf{Q} = \mathbf{I}_N$, the $N \times N$ identity matrix (Section 6.1). The POD ROM solutions are compared with solutions obtained by reducing the system using balanced truncation [73, 42]. First, the eigenvalues of the ROM matrix $\mathbf{A}_M$ for each M are computed to determine stability using Theorem 3.2.3. The maximum real part of the eigenvalues of these ROM system matrices is plotted in Figure 6.2 as a function of M. The reader can observe that the Lyapunov inner product POD ROMs and balanced truncation ROMs are stable for all M considered – all the real parts of the eigenvalues of these systems' matrices are ≤ 0. In contrast, the L^2 POD ROMs are unstable for all M.

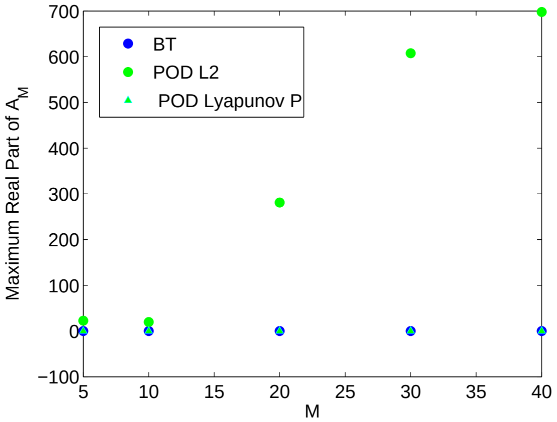


Figure 6.2. Maximum real part of eigenvalues of ROM system matrix $\mathbf{A}_M$ for ISS problem

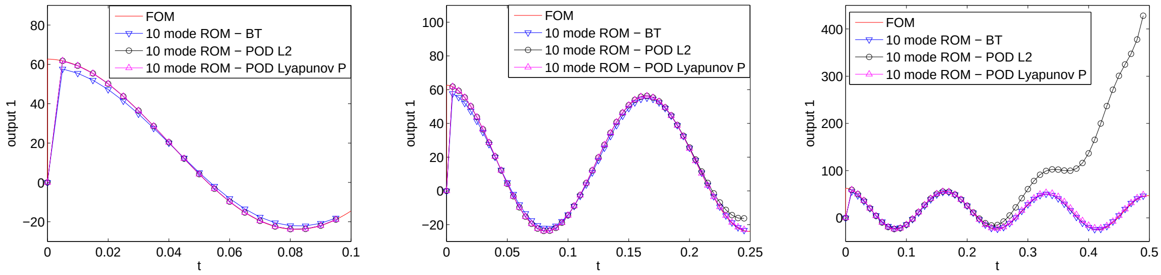
Having checked stability, each ROM is run until a specified time T_{max}, and the average error in the output relative to the full order model (6.1) is computed. The relative errors (6.1) in the output for ROMs of different sizes run up to different values of T_{max} are summarized in Table 6.1. In the case a ROM went unstable and (6.1) overflowed, the table contains an entry of '$-$'.

The objective of the $K_{max} = 2000$ ($T_{max} = 0.1$ seconds) run is to test how well the POD bases can reproduce the snapshots from which they were constructed, as exactly $K_{max} = 2000$ snapshots (taken up to $T_{max} = 0.1$ seconds) were used to generate these bases. Although the L^2 POD ROM is unstable for all values of M considered (Figure 6.2), this ROM still produces a reasonable solution for $M = 5$ and $M = 10$ (Figure 6.3(a) and Table 6.1). The instability manifests itself if a larger basis size is used, however. The Lyapunov ROM remains stable and accurate – orders of magnitude more accurate than the balanced truncation ROM for each M considered (Table 6.1).

The objective of the $K_{max} = 5000$ ($T_{max} = 0.25$ seconds) and $K_{max} = 10,000$ ($T_{max} = 0.5$ seconds) runs is to test the predictive capabilities of the POD ROMs relative to the balanced truncation ROMs for long-time simulations. The reduced order models are run for a much longer time horizon than the run used to generate the POD bases employed in building the ROMs. For $K_{max} = 5000$, The L^2 POD ROM exhibits an instability for all M considered except $M = 10$. For this value of M, the balanced truncation ROM and Lyapunov POD ROM are more accurate than the L^2 POD ROM, however (Figure 6.3(b) and Table 6.1). For $K_{max} = 10,000$, the L^2 POD ROM is unstable for all M considered. This instability is apparent in Figure 6.3(c). Hence, the instability identified in the earlier eigenvalue analysis (Figure 6.2) manifests itself if the L^2 POD ROM is run for a long enough time. For $K_{max} = 5000$ and $K_{max} = 10,000$, the Lyapunov POD ROM is more accurate than the balanced truncation ROM for small M. However, its accuracy is limited, as there does not appear to be a convergence with M-refinement.

Table 6.1. Relative errors (6.1) $\mathscr{E}_{rel,\mathbf{y}}$ in ROM output for ISS problem

		M				
K_{max}	Method	5	10	20	30	40
2000	Balanced truncation	9.80×10^{-2}	6.39×10^{-2}	9.56×10^{-3}	2.34×10^{-3}	8.34×10^{-4}
	POD L^2	1.09×10^{-4}	3.14×10^{-7}	$-$	$-$	$-$
	POD Lyapunov $\mathbf{P}$	8.69×10^{-6}	4.05×10^{-7}	1.13×10^{-6}	8.44×10^{-7}	9.22×10^{-7}
5000	Balanced truncation	7.64×10^{-2}	4.68×10^{-2}	8.14×10^{-3}	1.87×10^{-3}	5.58×10^{-4}
	POD L^2	2.41	4.73×10^{-2}	$-$	$-$	$-$
	POD Lyapunov $\mathbf{P}$	2.88×10^{-2}	5.24×10^{-3}	1.31×10^{-2}	1.21×10^{-2}	2.86×10^{-2}
10,000	Balanced truncation	6.87×10^{-2}	4.47×10^{-2}	7.08×10^{-3}	1.78×10^{-3}	5.76×10^{-4}
	POD L^2	165	3.24	$-$	$-$	$-$
	POD Lyapunov $\mathbf{P}$	5.25×10^{-2}	6.46×10^{-2}	9.92×10^{-2}	1.08×10^{-1}	9.92×10^{-2}



(a) $K_{max} = 2000$ **(b)** $K_{max} = 5000$ **(c)** $K_{max} = 10,000$

Figure 6.3. $\mathbf{y}_{QM}(t)$ for $M = 10$ ROMs (FOM = full order model) for ISS problem

Electrostatically actuated beam test case

To generate the snapshots from which several POD bases are constructed for the electrostatically actuated beam example, the full order model (4.18) is solved using a backward Euler time integration scheme with an initial condition of $\mathbf{x}_N(0) = \mathbf{0}$ and an input corresponding to a periodic on/off switching, i.e.,

$$\mathbf{u}_P(t) = \begin{cases} & 0.005 < t < 0.01, 0.015 < t < 0.02, \\ 1, & 0.03 < t < 0.035, \\ 0, & \text{otherwise} \end{cases} \qquad (6.14)$$

A total of $K_{max} = 1000$ snapshots are collected, every $\Delta t_{snap} = 5 \times 10^{-5}$ seconds, until time $T_{max} = 0.05$ seconds. From these snapshots, 5, 10, 20 and 30 mode ROMs are constructed using POD in the L^2 inner product, and POD in the Lyapunov inner product. In solving the Lyapunov equation (6.9) for the Lyapunov inner product weighting matrix $\mathbf{P}$, the matrix $\mathbf{Q}$ is taken to be the $N \times N$ identity matrix. The system (4.18) is reduced also using balanced truncation.

As for the ISS example, the first step is to study the stability of each ROM. Figure 6.4 shows the maximum real part of the ROM system matrices $\mathbf{A}_M$ for each M considered. It is found that the L^2 ROM is unstable for each M, and becomes more unstable with increasing M. In contrast, the balanced truncation and POD Lyapunov inner product ROMs are stable for all M considered, as expected.

Next, the accuracy of each ROM is examined. Table 6.2 summarizes the errors (6.1) in the ROM solutions relative to the full order model solution for three runs of different lengths. As before, an entry of '$-$' in the table indicates that the error overflowed due to an instability in the ROM.

The objective of the first run ($K_{max} = 1000$) is to study how well the POD ROMs can reproduce the snapshots from which they were constructed, and to compare these ROMs' performance with the performance of ROMs constructed using balanced truncation. The reader can observe that the POD ROM constructed in the Lyapunov inner product is the most accurate. The POD L^2 ROM is both unstable as well as inaccurate (Figure 6.5(a)).

The second two runs ($K_{max} = 2000$ and $K_{max} = 5000$) are aimed to study the predictive capabilities of the ROMs for long-time simulations. The full order model is run until times 0.1 and 2.5 seconds respectively. As before, only snapshots up to time $t = 0.05$ seconds are used to construct the POD bases for the ROMs. In addition to the signal (6.35), the following inputs are applied in both the full order model and the ROM:

$$\mathbf{u}_P(t) = \begin{cases} & 0.055 < t < 0.06, 0.065 < t < 0.07, \\ & 0.08 < t < 0.085, 0.105 < t < 0.11, \\ & 0.115 < t < 0.12, 0.13 < t < 0.135, \\ & 0.205 < t < 0.21, 0.215 < t < 0.22, \\ 1, & 0.23 < t < 0.235, \\ 0, & \text{otherwise}. \end{cases} \tag{6.15}$$

The reader may observe by examining Table 6.2 and Figure 6.5 that the balanced truncation ROMs are in general the most accurate. The POD ROMs constructed in the Lyapunov inner product nonetheless produce reasonable results (Figures 6.5(b)-(c)) and appear to be converging to the full order model solution with M-refinement (Table 6.2). The POD L^2 ROM result is not shown in Figures 6.5(b)-(c), as the solution produced by this ROM blows up around time $t = 0.02$ seconds.

Table 6.2. Relative errors (6.1) $\mathscr{E}_{rel,\mathbf{y}}$ in ROM output for electrostatically actuated beam problem

		M			
K_{max}	Method	5	10	20	30
1000	Balanced truncation	6.29×10^{-2}	4.51×10^{-3}	6.93×10^{-5}	3.60×10^{-6}
	POD L^2	8.56×10^{-1}	6.62	–	–
	POD Lyapunov $\mathbf{P}$	2.05×10^{-3}	6.23×10^{-5}	2.09×10^{-8}	1.35×10^{-8}
2000	Balanced truncation	5.84×10^{-2}	4.47×10^{-3}	6.29×10^{-5}	3.17×10^{-6}
	POD L^2	7.76	4.26×10^{3}	–	–
	POD Lyapunov $\mathbf{P}$	3.62×10^{-2}	1.12×10^{-2}	3.47×10^{-4}	4.13×10^{-5}
5000	Balanced truncation	7.36×10^{-2}	4.77×10^{-3}	5.48×10^{-5}	2.77×10^{-6}
	POD L^2	4.40×10^{3}	–	–	–
	POD Lyapunov $\mathbf{P}$	1.80×10^{-1}	1.09×10^{-1}	2.03×10^{-2}	6.09×10^{-3}

Lastly, the level of computational resources required for computing the Lyapunov inner product and the level of computational resources required for performing model reduction via balanced truncation [73, 42] are compared. Table 6.3 gives the CPU times for the sum of the following operations in the balanced truncation [73, 42] algorithm as a function of N, the problem size: calculation of the observability Gramian, calculation of the controllability Gramian, and calculation of the balancing transformation (Appendix A.8). All computations are performed in serial using MATLAB's linear algebra capabilities and MATLAB's control toolbox [2], on a Linux workstation with 6 Intel Xeon 2.93 GHz CPUs. Both methods exhibit $O(N^3)$ scaling. Although the Lyapunov inner product computation is costly, as it requires the solution of a Lyapunov equation, it completes in 2-3 times less CPU time than the balanced truncation algorithm. This is because balanced truncation requires the solution of *two* Lyapunov equations for the observability and reachability Gramians, as well as the Cholesky and eigenvalue factorizations of these Gramians.

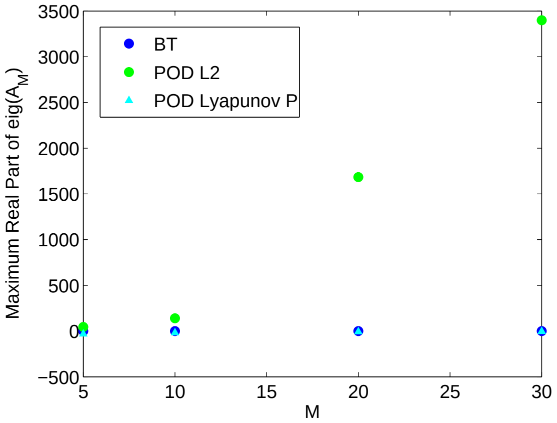


Figure 6.4. Maximum real part of eigenvalues of ROM system matrix $\mathbf{A}_M$ for electrostatically actuated beam problem

Table 6.3. CPU times (in seconds) for balanced truncation vs. Lyapunov inner product computations (electrostatically actuated beam problem)

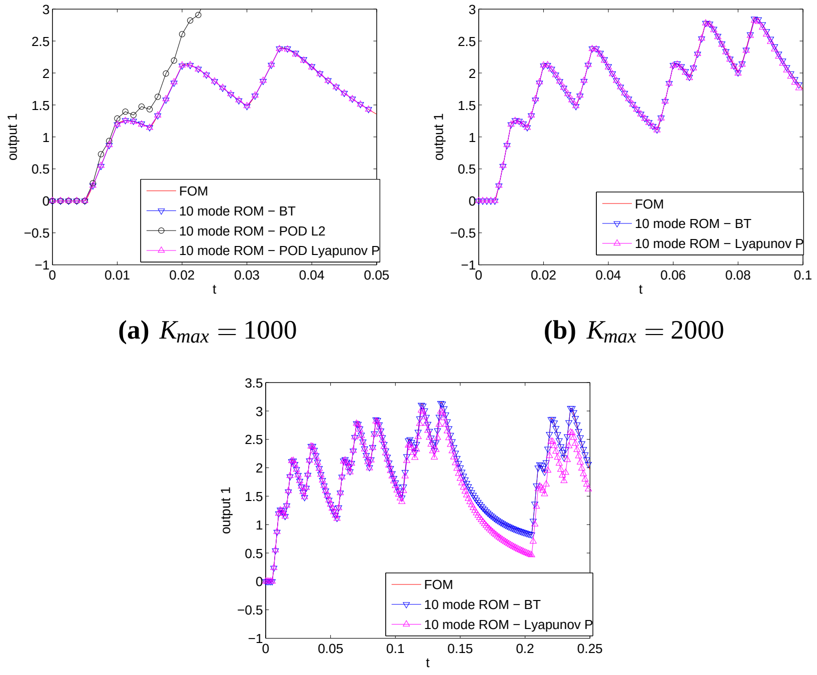
	N			
Method	1250	2500	5000	10,000
Lyapunov Inner Product	5.08×10^1	4.60×10^2	4.02×10^3	6.09×10^4
Balanced Truncation	1.09×10^2	1.10×10^3	1.04×10^4	1.24×10^5

6.2 ROM stabilization via optimization-based eigenvalue reassignment

In this section, two *new* algorithms for stabilizing LTI systems of the form (4.18) by modifying the unstable eigenvalues of $\mathbf{A}_M$ through a "black-box" post-processing step applied to the given (unstable) ROM system are proposed. These algorithms were developed as a part of the LDRD project summarized in this report. It will be assumed from this point onward that the matrix $\mathbf{A}$ defining the FOM system (4.18) is stable. Algorithm 2 is the primary contribution of this report. Algorithm 1 is provided, as it served as a strategic foundation for the final development (Algorithm 2). It is given here not only for the sake of completeness, but also because it is shown in Section 6.2.3 that Algorithm 2 can be seen as a variant of Algorithm 1.

6.2.1 Algorithm 1: ROM stabilization via full state feedback (a.k.a. pole placement)

The first ROM stabilization algorithm is motivated by the observation that (4.20) is an LTI system, and, as such, can be stabilized using full state feedback, or pole placement, methods from control theory [11, 96]. The general approach of stabilizing an LTI system using full state feedback is



(a) $K_{max} = 1000$

(b) $K_{max} = 2000$

(c) $K_{max} = 5000$

Figure 6.5. $\mathbf{y}_{QM}(t)$ for $M = 10$ ROMs (FOM = full order model) for electrostatically actuated beam problem

reviewed below.

Consider the open loop ROM LTI system (4.20), where it is assumed $\mathbf{u}(t)$ is given, so that $\mathbf{B}_M\mathbf{u}(t)$ represents, for instance, a given source for the equations. The objective of full state feedback (pole placement) is to redesign the dynamics of the system (4.20) through feedback of the state. If $\mathbf{A}_M$ is unstable, it is desired to redesign the system such that it is stable. Towards this end, the open-loop system (4.20) is transformed into a closed-loop system, and a feedback controller that positions the closed loop eigenvalues of the system is developed. The first step is to select a control matrix $\mathbf{B}_C \in \mathbb{R}^{M\times J}$ for some integer J, and modify the system (4.20) by adding to it the control $\mathbf{B}_C\mathbf{u}_C(t)$:

$$\begin{aligned} \dot{\mathbf{x}}_M(t) &= \mathbf{A}_M\mathbf{x}_M(t) + \mathbf{B}_M\mathbf{u}(t) + \mathbf{B}_C\mathbf{u}_C(t) \\ \mathbf{y}_M(t) &= \mathbf{C}_M\mathbf{x}_M(t). \end{aligned} \tag{6.16}$$

Here, $\mathbf{u}_C(t) \in \mathbb{R}^J$ is a control that will be designed to modify the dynamics of the original system (4.20) such that it is stable. For an LTI system representing some physical dynamics, $\mathbf{B}_C$ is typically selected to represent a physical control that can be imposed on the system, e.g., actuation applied to a boundary of a fluid domain. Next, a linear control law of the form $\mathbf{u}_C(t) = -\mathbf{K}_C\mathbf{x}_M(t)$ is assumed, where $\mathbf{K}_C \in \mathbb{R}^{J\times M}$ is the control matrix, to be determined. Substituting this law into (6.16) and rearranging, the following is obtained:

$$\begin{aligned} \dot{\mathbf{x}}_M(t) &= (\mathbf{A}_M - \mathbf{B}_C\mathbf{K}_C)\mathbf{x}_M(t) + \mathbf{B}_M\mathbf{u}(t) \\ \mathbf{y}_M(t) &= \mathbf{C}_M\mathbf{x}_M(t). \end{aligned} \tag{6.17}$$

The system (6.17) is a system of the form (4.20) but with $\mathbf{A}_M$ replaced by $\tilde{\mathbf{A}}_M$, where

$$\tilde{\mathbf{A}}_M \equiv \mathbf{A}_M - \mathbf{B}_C\mathbf{K}_C. \tag{6.18}$$

The reader can observe that if it is possible to compute the control matrix $\mathbf{K}_C$ such that $\tilde{\mathbf{A}}_M$ is stable, the ROM system (6.16) will be stable.

In order to formulate a well-posed ROM stabilization algorithm based on the approach outlined above, a number of questions need to be addressed:

(i) How should the control matrix $\mathbf{B}_C$ be selected? Typically, when applying pole placement algorithms, a *physical* system is stabilized using a *physical* controller. In this case, the controller matrix $\mathbf{B}_C$ is added at the level of the algebraic system (6.16). In this context, what does $\mathbf{B}_C$ mean? What should it mean?

(ii) What eigenvalues should the stabilized ROM matrix $\tilde{\mathbf{A}}_M$ (6.18) be prescribed to have? It is clear that the eigenvalues should lie in the stable half of the complex plane, but what physical values should they have?

(iii) Does the solution $\mathbf{K}_C$ to the pole placement problem exist?

(iv) How has the stabilization affected the accuracy of the ROM? By modifying the ROM system (4.20), inconsistencies between the FOM and ROM physics have been introduced.

In this subsection, only question 3, the existence question, will be addressed. Answering this question gives rise to a preliminary ROM stabilization algorithm, referred to as “Algorithm 1”. The remaining questions are addressed through the formulation of “Algorithm 2”, described in Section 6.2.2.

Before formulating an algorithm which guarantees the existence of the solution to the pole placement problem described above, it is useful to recall the following theorem.

Theorem 6.2.1 (quoted from [11]): If the pair $(\mathbf{A}_M, \mathbf{B}_C)$ is controllable[2], there exists a feedback $\mathbf{u}_C(t) = -\mathbf{K}_C\mathbf{x}_M$ such that the eigenvalues of $\tilde{\mathbf{A}}_M$ (6.18) can be arbitrarily assigned.

In general, the pair $(\mathbf{A}_M, \mathbf{B}_C)$ may not be controllable. However, it is possible to apply Theorem 6.2.1 by working in the controllable and observable[3] subspaces of $\mathbf{A}_M$ and $\mathbf{B}_C$, which can be isolated through the Kalman decomposition. A detailed discussion of the Kalman decomposition

[2] An LTI system (4.18) is controllable (a.k.a. reachable) if for any $\mathbf{x}_0, \mathbf{x}_f \in \mathbb{R}^N$, there exists a $T > 0$ and $\mathbf{u} : [0,T] \to \mathbb{R}$ such that the corresponding solution satisfies $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathbf{x}(T) = \mathbf{x}_f$ [11, 96]. To test for controllability of a linear system, it is sufficient to check the rank of the controllability matrix

$$\mathbf{W}_c \equiv \begin{pmatrix} \mathbf{B}, & \mathbf{AB}, & \cdots, & \mathbf{A}^{N-1}\mathbf{B} \end{pmatrix}. \tag{6.19}$$

The LTI system (4.18) is controllable if and only if the controllability matrix (6.19) is invertible [96, 11].

[3] An LTI system (4.18) is observable if for any $T > 0$ it is possible to determine the state of the system $\mathbf{x}(T)$ through measurements of $\mathbf{y}(t)$ and $\mathbf{u}(t)$ on the interval $[0,T]$ [11, 96]. To test for observability of a linear system, it is sufficient

can be found in classical control theory texts, e.g., [11, 96]. The key result of the Kalman theorem is that the state space can be decomposed into four parts: a part that is reachable and observable, a part that is reachable but not observable, a part that is not reachable but observable and a part that is neither reachable nor observable. The procedure is summarized in Algorithm 1.

Algorithm 1

- Pick a control matrix $\mathbf{B}_C$, e.g., $\mathbf{B}_C = \mathbf{1}_M$.
- Given $\mathbf{B}_C$, use the Kalman decomposition to isolate the controllable and observable parts of $\mathbf{A}_M$ and $\mathbf{B}_C$, call them $\mathbf{A}_M^{co} = \mathbf{U}\mathbf{A}_M\mathbf{U}^T$ and $\mathbf{B}_C^{co} = \mathbf{U}\mathbf{B}_C$ respectively.
- Compute the eigenvalues $\lambda_1^{co}, \dots, \lambda_{M^{co}}^{co}$ of $\mathbf{A}_M^{co}$.
- Reassign the unstable eigenvalues of $\mathbf{A}_M^{co}$ to make them stable, e.g., for $k = 1$ to M^{co}, set

$$\lambda_k = \min\{Re(\lambda_k^{co}), -Re(\lambda_k^{co})\} + i \cdot Im(\lambda_k^{co}), \tag{6.21}$$

  where $Re(z)$ and $Im(z)$ denote respectively the real and imaginary parts of a complex number $z \in \mathbb{C}$, and $i \equiv \sqrt{-1}$.
- Compute $\mathbf{K}_C$ such that $\mathbf{A}_M^{co} - \mathbf{K}_C\mathbf{B}_C^{co}$ has these eigenvalues using full state feedback (a.k.a pole placement) algorithms from control theory.
- Set $\mathbf{A}_M = \mathbf{U}^T(\mathbf{A}_M^{co} - \mathbf{K}_C\mathbf{B}_C^{co})\mathbf{U}$.

Typically in full state feedback, the matrix $\mathbf{B}_C$ represents a physical control that would be applied to a physical system of the form (4.20) so as to stabilize this system. The situation of interest here is not entirely comparable, as it has been assumed that the *physical* system underlying (4.20) *is* stable (and hence does not need stabilization via full state feedback); it is the *algebraic* ROM system (4.20) that is unstable, and hence the matrix $\mathbf{B}_C$ is added to the system at the algebraic level. This scenario complicates the interpretation of (and therefore the choice of) $\mathbf{B}_C$. In general, it can be argued that the choice of $\mathbf{B}_C$ does not matter provided the unstable eigenvalues of $\mathbf{A}_M$ are controllable and observable given the choice of $\mathbf{B}_C$. In the numerical example studied below, $\mathbf{B}_C$ is selected to be a vector of all ones.

It remains to provide some discussion of approaches for selecting the eigenvalues of the stabilized matrix $\tilde{\mathbf{A}}_M$. One possible choice is to replace the real parts of the unstable eigenvalues of $\mathbf{A}_M$ with their negatives (6.21), or some negative scaled multiple of these values. Another option is to try to match the eigenvalues of the stabilized ROM matrix $\tilde{\mathbf{A}}_M$ with the eigenvalues of the FOM matrix $\mathbf{A}$ (provided the computational resources to compute the FOM eigenvalues are available, which may not be the case for very large systems). In general, the eigenvalues of a stable ROM will lie on or near the manifold of the eigenvalues of the FOM from which the ROM was constructed. This is illustrated in Figure 6.6, which shows the eigenvalue manifold of the FOM matrix $\mathbf{A}$ and a ROM matrix $\mathbf{A}_M$ for an $M = 20$ mode ROM constructed via balanced truncation [73, 42] for a variant of the international space station benchmark. In fact, if $M = N$ in a ROM, that is, a ROM

to check the rank of the observability matrix

$$\mathbf{W}_o^T \equiv \begin{pmatrix} \mathbf{C}, & \mathbf{CA}, & \cdots, & \mathbf{CA}^{N-1} \end{pmatrix}. \tag{6.20}$$

The LTI system (4.18) is observable if and only if the observability matrix (6.20) is full rank [11, 96].

is constructed with a full basis of the space $\mathbb{R}^N$, $\mathbf{A}_M \sim \mathbf{A}$ (as can be seen from (4.21)), so that $\mathbf{A}_M$ will have the same eigenvalues as $\mathbf{A}$.

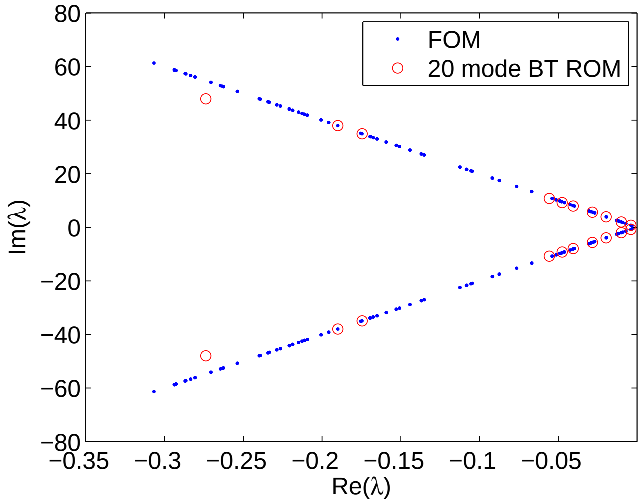


Figure 6.6. Eigenvalue manifold of FOM matrix $\mathbf{A}$ and ROM matrix $\mathbf{A}_M$ for an $M = 20$ mode ROM constructed via balanced truncation for a variant of the ISS benchmark

6.2.2 Algorithm 2: ROM stabilization through solution of constrained nonlinear least squares optimization problem

The primary downside of Algorithm 1 (Section 6.2.1) is it is unclear *a priori* how a particular choice of the control matrix $\mathbf{B}_C$ and stabilized eigenvalues will affect the accuracy of the resulting stabilized ROM. This problem is remedied in the present section through the development of a new algorithm, “Algorithm 2”. In this algorithm, the eigenvalues of the stabilized matrix $\tilde{\mathbf{A}}_M$ are determined such that the ROM output solution deviates minimally from the FOM output solution. Hence, questions 2 and 4 in Section 6.2.1 are addressed explicitly. As will be clear shortly, Algorithm 2 does not require the selection of a control matrix $\mathbf{B}_C$ (question 1).

Consider the ROM LTI system (4.20). Note that it is possible to work out analytically in closed form the exact solution to this system. The reader may verify that the solution to this system is given by

$$\mathbf{x}_M(t) = \exp(t\mathbf{A}_M)\mathbf{x}_M(0) + \int_0^t \exp\{(t-\tau)\mathbf{A}_M\}\mathbf{B}_M u(\tau)d\tau. \tag{6.22}$$

In equation (6.22), $\exp(\cdot)$ denotes the matrix exponential. It is worthwhile to note that this quantity is not an issue to compute, as the ROM system matrix $\mathbf{A}_M$ is small. Given the solution for the ROM state vector (6.22), the ROM output is given by

$$\mathbf{y}_M(t) = \mathbf{C}_M \left[\exp(t\mathbf{A}_M)\mathbf{x}_M(0) + \int_0^t \exp\{(t-\tau)\mathbf{A}_M\}\mathbf{B}_M u(\tau)d\tau\right]. \tag{6.23}$$

The existence of an analytical solution to the ROM LTI system (4.20) motivates the formulation of the following optimization problem, to be solved for the eigenvalues of the stabilized ROM system:

$$\begin{aligned} \min_{\lambda_i^u} \quad & \sum_{k=1}^{K} ||\mathbf{y}^k - \mathbf{y}_M^k||_2^2. \\ \text{s.t.} \quad & Re(\lambda_i^u) < 0, i = 1, ..., L \end{aligned} \tag{6.24}$$

The optimization is over the unstable eigenvalues of the original ROM system matrix $\mathbf{A}_M$, denoted by λ_i^u, for $i = 1, .., L$ where $L \leq M$ is the number of unstable eigenvalues of $\mathbf{A}_M$. The shorthand $\mathbf{y}^k$ denotes the FOM output at time t_k, i.e., $\mathbf{y}^k \equiv \mathbf{y}(t_k)$. In a model reduction approach based on an empirical basis computed from a set of snapshots of the high-fidelity solution, e.g., the POD or BPOD method, these values are available at the snapshot times. The shorthand $\mathbf{y}_M^k$ denotes the ROM output at time t_k, i.e., $\mathbf{y}_M^K \equiv \mathbf{y}_M(t_k)$. It is given by the formula (6.23). The constraint in (6.24) ensures that the stabilized ROM eigenvalues are in the stable half of the complex plane Here $Re(z)$ denotes the real part of a complex number $z \in \mathbb{C}$. Equation (6.24) is a constrained nonlinear least-squares optimization problem with inequality constraints.

Remark that the optimization problem (6.24) is small: there are at most M dofs, and solving the problem does not require operating on any matrices that are of size $\mathscr{O}(N)$. This optimization problem can be solved using standard algorithms for constrained optimization, e.g., an SQP algorithm with line search globalization, BFGS for Hessian approximations, and an interior point method to handle the inequality constraints [77].

An interesting question that arises is whether the solution to the optimization problem (6.24) is unique. A sufficient condition for a minimization problem of the form

$$\min_x f(x), \tag{6.25}$$

where $x \in \mathbb{R}^n$ is a real vector and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth function, to have a unique solution is for f to be convex [77]. In this case, any stationary point of f is a global minimizer of f, and hence a local minimizer of f will be the global minimizer of f. It is straightforward to show that the objective function in (6.24) is not necessarily convex. Since convexity is a sufficient but not a necessary condition for uniqueness of the solution to (6.24), the optimization problem could have a unique solution, but this scenario is not guaranteed. The numerical tests performed in Section 6.2.4 suggest that the optimization problem (6.24) has in general multiple solutions.

It turns out that it is convenient to implement and solve the optimization problem (6.24) in the "characteristic variables", defined by $\mathbf{z}_M(t) = \mathbf{S}_M^{-1}\mathbf{x}_M(t)$, where $\mathbf{S}_M^{-1}$ is the matrix that diagonalizes $\mathbf{A}_M$, i.e., $\mathbf{A}_M = \mathbf{S}_M\mathbf{D}_M\mathbf{S}_M^{-1}$. The steps of the stabilization are detailed in Algorithm 2. Note that, although it is assumed here $\mathbf{A}_M$ is diagonalizable, the extension to non-diagonalizable $\mathbf{A}_M$ is straightforward. In this case, the eigenvalue decomposition in Algorithm 2 (6.26) is replaced with the Jordan decomposition.

Algorithm 2

```
• Diagonalize the ROM matrix $\mathbf{A}_M$:

      $\mathbf{A}_M = \mathbf{S}_M \mathbf{D}_M \mathbf{S}_M^{-1}.$                (6.26)

• Initialize a diagonal $M \times M$ matrix $\tilde{\mathbf{D}}_M$.
• Set $j = 1$.
• for $i = 1$ to $M$
    if $Re(D_M(i,i)) < 0$
      Set $\tilde{D}_M(i,i) = D_M(i,i)$.
    else
      Set $\tilde{D}_M(i,i) = \lambda_j^u$.
    endif
  endfor
• Increment $j \leftarrow j+1$.
• Solve the optimization problem (6.24) for the eigenvalues $\{\lambda_j^u\}$ with $\mathbf{y}_M(t)$ given by

      $\mathbf{y}_M(t) = \mathbf{C}_M \left[ \mathbf{S}_M \exp(t\tilde{\mathbf{D}}_M)\mathbf{S}_M^{-1}\mathbf{x}_M(0) + \int_0^t \mathbf{S}_M \exp\{(t-\tau)\tilde{\mathbf{D}}_M\}\mathbf{S}_M^{-1}\mathbf{B}_M u(\tau) d\tau \right],$   (6.27)

  using an optimization algorithm.
• Evaluate $\tilde{\mathbf{D}}_M$ at the solution of the optimization problem (6.24).
• The stabilized LTI ROM system is now given by

      $\dot{\mathbf{x}}_M(t) = \tilde{\mathbf{A}}_M \mathbf{x}_M(t) + \mathbf{B}_M \mathbf{u}(t)$
      $\mathbf{y}_M(t) = \mathbf{C}_M \mathbf{x}_M(t),$                (6.28)

  where $\tilde{\mathbf{A}}_M = \mathbf{S}_M \tilde{\mathbf{D}}_M \mathbf{S}_M^{-1}$.
```

6.2.3 Connection between Algorithm 1 and Algorithm 2

One notable difference between Algorithms 1 and 2 is that, unlike the former algorithm, the latter algorithm does not employ directly full state feedback (a.k.a. pole placement) routines from control theory to solve for the stabilized ROM matrix $\tilde{\mathbf{A}}_M$. However, it turns out that it is possible to show that Algorithm 2 is equivalent to Algorithm 1 for a specific choice of control matrices $\mathbf{B}_C$ and $\mathbf{K}_C$.

Suppose $\mathbf{A}_M$ has $L \leq M$ unstable eigenvalues λ_k^u, each with corresponding eigenvector $\mathbf{s}_k^u$. Let $\tilde{\lambda}_k^u$ denote the stabilized value of λ_k^u, obtained by solving the optimization problem (6.24). The reader can verify that $\tilde{\mathbf{A}}_M$ in (6.28) is equivalent to

$$\tilde{\mathbf{A}}_M = \mathbf{A}_M - \mathbf{B}_C \mathbf{K}_C, \tag{6.29}$$

where

$$\mathbf{B}_C = \begin{pmatrix} \mathbf{s}_1^u, & \cdots, & \mathbf{s}_L^u \end{pmatrix} \in \mathbb{R}^{M \times L} \tag{6.30}$$

$$\mathbf{K}_C = \begin{pmatrix} \lambda_1^u - \tilde{\lambda}_1^u & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2^u - \tilde{\lambda}_2^u & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \lambda_L^u - \tilde{\lambda}_L^u & 0 \end{pmatrix} \mathbf{S}_M^{-1} \in \mathbb{R}^{L\times M}. \tag{6.31}$$

6.2.4 Numerical experiments

The performance of the ROM stabilization algorithms described in this section is now assessed on the two benchmarks considered in this chapter: the ISS benchmark, and the electrostatically actuated beam benchmark, both introduced earlier.

Typically, the size of a reduced POD basis, namely M, is calculated using the energy criterion (4.6). That is, M is selected such that the reduced basis $\mathbf{\Phi}_M$ captures some fixed percentage of the snapshot energy, e.g., 95% or 99% (see [71, 50]). For the problems considered here, M is chosen to be the smallest integer such that: (1) the basis $\mathbf{\Phi}_M$ captures at least 99% of the snapshot energy, (2) the resulting POD/Galerkin ROM has at least one unstable eigenvalue, and (3) the POD/Galerkin ROM goes unstable during the time horizon considered. This strategy of choosing M is a natural one given the objective of this chapter: to evaluate the ROM stabilization algorithms developed in this section.

For the ISS example the performance of Algorithm 1 and the performance of Algorithm 2 are evaluated. This comparison is intended to highlight the superiority of Algorithm 2 over Algorithm 1. For the sake of brevity, results for only Algorithm 2 (established in the context of the ISS example as the superior algorithm) are shown for the electrostatically actuated beam example.

To solve the constrained nonlinear least squares optimization at the heart of Algorithm 2 (6.24), the `fmincon` function in the MATLAB optimization toolbox [1, 77] is employed. The `Algorithm` option required by this function is set to `interior-point` with exact (analytic) Jacobians. An analytic expression for the Jacobian of the objective function for the specific case of $\mathbf{u}(t) = \mathbf{0}$ and one output of interest in (6.24) can be found in Appendix A.9. Deriving and implementing an analytic Jacobian is recommended over using finite difference Jacobians calculated within the MATLAB optimization toolbox. Since analytic Jacobians are exact, they are accurate. In contrast, finite difference Jacobians can be inaccurate for some problems as a result of an arbitrary selection of the finite difference increment. Moreover, the solution of the optimization problem (6.24) is much faster with exact Jacobian due to fewer required function evaluations. With exact Jacobians, the number of function evaluations per optimization step is constant. In particular, it does not grow with L, the number of eigenvalues reassigned by the optimization algorithm. The default `fmincon` settings for this method are used, which can be found in [1].

Note that the `fmincon` function will compute only real solutions to an optimization problem. In general the eigenvalues of the matrix $\mathbf{A}_M$ may be complex, however. To allow the `fmincon` algorithm to compute complex eigenvalue solutions of the ROM stabilization optimization problem (6.24), a complex-valued functional form for λ_j^u may be assumed. In this approach, λ_j^u in line 9 of

Algorithm 2 is replaced with

$$\lambda_j^u \leftarrow \lambda_j^{ur} + i \cdot \lambda_j^{uc} \in \mathbb{C}, \quad \lambda_j^{ur}, \lambda_j^{uc} \in \mathbb{R}, \tag{6.32}$$

(where $i \equiv \sqrt{-1}$) and (6.24) is solved for $\lambda_j^{ur}, \lambda_j^{uc} \in \mathbb{R}$ subject to the constraint that $\lambda_j^{ur} < 0$. Since complex eigenvalues of $\mathbf{A}_M$ occur in complex-conjugate pairs, if λ_j^u has the form (6.32), then λ_{j+1} in Algorithm 2 must have the form

$$\lambda_{j+1}^u = \lambda_{j+1}^{ur} - i \cdot \lambda_{j+1}^{uc} \in \mathbb{C}, \quad \lambda_{j+1}^{ur}, \lambda_{j+1}^{uc} \in \mathbb{R}. \tag{6.33}$$

It follows that the approach of assuming complex-conjugate pair solutions to (6.24) does not give rise to more dofs than the default approach of solving for real solutions to this problem. In fact, the former approach has fewer constraints.

The numerical results section includes comparisons of the following CPU times for both problems considered:

- The CPU time required for the time-integration of the FOM.
- The CPU time required for the offline (snapshot collection, loading of system matrices/snaptions, calculation of the POD basis, Galerkin projection, and numerical solution of the optimization problem (6.24)) stage of the POD/Galerkin ROMs.
- The CPU time required for the online (time-integration) stage of the POD/Galerkin ROMs.

All computations are performed in serial using MATLAB's linear algebra capabilities on a Linux workstation with 6 Intel Xeon 2.93 GHz CPUs. Note that the FOM CPU times do not include the time to discretize the relevant PDEs using the finite element method and assemble the global system matrix. This is due to the fact that the matrices defining the FOM were downloaded from a model reduction benchmark repository, and access to the high-fidelity code that generated these matrices is not available to the authors.

In general, ROMs are employed for many-query and/or real-time analysis. In these contexts, it is critical that the online time-integration stage of the ROM has a low computational cost and fast run-time. Although the offline construction of the reduced order model, which includes the collection of snapshots, the construction of the POD basis, the Galerkin projection, and the solution of the optimization problem (6.24), can be computationally intensive, this step is done only *one* time when the ROM is constructed. The cost of this computation does not affect the run-time of the online step of the model reduction, the step relevant to analysis using the ROM. Nonetheless, it may be of interest how many times the ROM would need to be run (online) to compensate the cost of the (offline) pre-processing step. For this reason, estimates of the number of online ROM runs that would be required to offset the offline ROM cost are given for each example considered following the CPU time data (Tables 6.8 and 6.12).

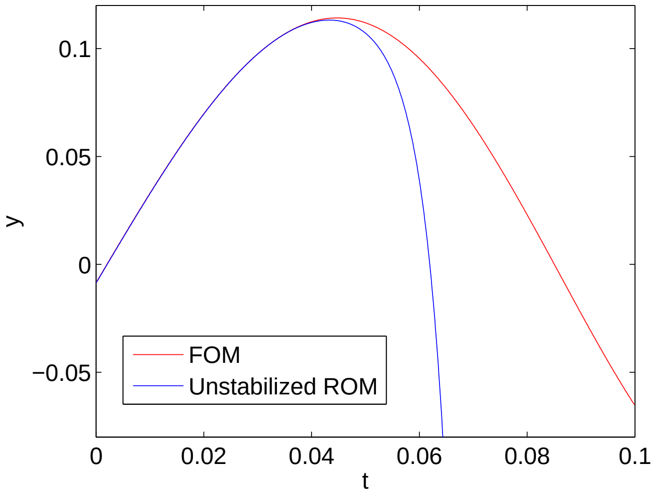


Figure 6.7. Outputs for $M = 20$ unstabilized POD/Galerkin ROM vs. FOM output for ISS problem

International space station (ISS) test case

As before (Section 6.1.1), to generate the snapshots from which the POD bases are constructed, the full order model (4.18) is solved using a backward Euler time integration scheme with an initial condition of $\mathbf{x}_N(0) = \mathbf{1}_N$ ($N \times 1$ vector of all ones) and no input $(\mathbf{u}(t) = \mathbf{0})$. A total of $K_{max} = 2000$ snapshots are collected, every $\Delta t_{snap} = 5 \times 10^{-5}$, until time $T_{max} = 0.1$ seconds. These snapshots are used to compute a POD basis of size $M = 20$, and a POD/Galerkin ROM of size $M = 20$ is constructed using this basis. For this problem, the $M = 20$ mode POD/Galerkin ROM is found to be unstable with four unstable eigenvalues. This basis captures essentially 100% of the snapshot energy, and the value $M = 20$ is the smallest basis size such that the ROM exhibits an instability. The numerical values of the unstable eigenvalues are: $\lambda_1^u = 242.5$, $\lambda_2^u = 32.90 + 26.99i$, $\lambda_3^u = 32.90 - 26.99i$, $\lambda_4^u = 2.712$. Figure 6.7 shows the FOM output $\mathbf{y}(t)$ (in red) compared to the unstabilized ROM output (in blue). The unstabilized ROM output diverges from the FOM output around time $t = 0.05$ and approaches $-\infty$ as $t \to \infty$ due to the ROM instability. The relative error $\mathscr{E}_{rel,\mathbf{y}}$ in the unstabilized ROM output (6.1) is 1737.9.

The $M = 20$ mode POD/Galerkin ROM for the ISS problem is stabilized first by Algorithm 1, then by Algorithm 2. These results illustrate the superiority of Algorithm 2 over Algorithm 1.

Stabilization via Algorithm 1

First, the $M = 20$ mode unstable POD/Galerkin ROM is stabilized using Algorithm 1 with the control matrix $\mathbf{B}_C$ selected to be an $M \times 1$ vector of all ones: $\mathbf{B}_C = \mathbf{1}_M$. The next step in the stabilization is to select the desired eigenvalues of the stabilized ROM matrix $\tilde{\mathbf{A}}_M$. Let λ_k^u for $k = 1, \ldots, 4$ denote the unstable eigenvalues for $\mathbf{A}_M$, and let $\tilde{\lambda}_k^u$ denotes the corresponding eigenvalues of $\tilde{\mathbf{A}}_M$ (that is, the values λ_k^u will be replaced within the stabilization algorithm). Here, the following functional form for $\tilde{\lambda}_k^u$ will be considered:

$$\tilde{\lambda}_k^u = -\alpha \cdot Re(\lambda_k^u) + i \cdot Im(\lambda_k^u), \quad \alpha > 0, \tag{6.34}$$

for $k = 1, ..., 4$, where $Re(z)$ and $Im(z)$ denote respectively the real and imaginary parts of a complex number $z \in \mathbb{C}$ and $i \equiv \sqrt{-1}$. The transformation (6.34) flips the sign of the real part of an unstable eigenvalue of $\mathbf{A}_M$ (thereby making it stable), and scales this value by a positive constant α. Three choices of the parameter α in (6.34) will be tested here:

- $\alpha = 0.1$.
- $\alpha = 1$.
- $\alpha = 10$.

The objective is to study the error in the stabilized ROM for several choices of $\tilde{\lambda}_i^u$. The choices are admittedly ad hoc, as there is no clear guideline for what the eigenvalues of $\tilde{\mathbf{A}}_M$ should be. Note that as α is increased, the eigenvalues $\tilde{\lambda}_k^u$ are pushed further into the left (stable) half of the complex plane.

Figure 6.8 shows the outputs computed by the three stabilized ROMs obtained using Algorithm 1. The relative errors in the stabilized ROM outputs are given in Table 6.4. All three ROMs are stable (by construction). The ROM stabilized by Algorithm 1 with $\alpha = 1$ is slightly more accurate than the ROM stabilized by Algorithm 1 with $\alpha = 0.1$. This may lead the reader to conjecture that the accuracy of the stabilized ROM will improve as the eigenvalues are pushed further and further into the left half of the complex plane. However, the ROM stabilized by Algorithm 1 with $\alpha = 10$ results demonstrate that this is not the case: the ROM with its eigenvalues pushed the most into the left half of the complex plane is the least accurate.

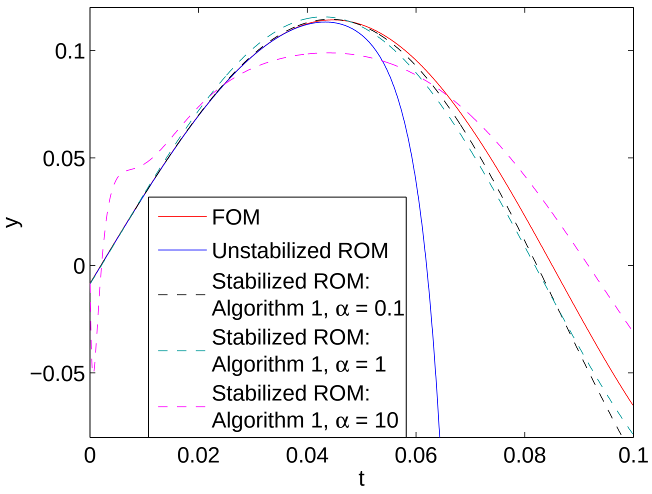


Figure 6.8. Outputs for $M = 20$ POD/Galerkin ROMs stabilized via Algorithm 1 vs. FOM output for ISS problem

Table 6.4. Relative errors in $M = 20$ POD/Galerkin ROM for ISS problem stabilized via Algorithm 1

ROM	$\mathscr{E}_{rel,\mathbf{y}}$
Unstabilized	1737.8
ROM stabilized via Algorithm 1 with $\alpha = 0.1$	1.51×10^{-2}
ROM stabilized via Algorithm 1 with $\alpha = 1$	1.16×10^{-2}
ROM stabilized via Algorithm 1 with $\alpha = 10$	2.26×10^{-2}

The numerical results presented here show that Algorithm 1 works in the sense that it will stabilize an unstable ROM. Unfortunately, the accuracy of a ROM stabilized using this algorithm is in general unknown before the ROM is stabilized and the ROM output is computed. Moreover, for some choices of $\tilde{\lambda}_i^u$ the accuracy may be unacceptable.

Stabilization via Algorithm 2

The $M = 20$ POD/Galerkin ROM for the ISS benchmark is now stabilized using Algorithm 2. Let λ_k^u for $k = 1,...,4$ denote the four unstable eigenvalues of $\mathbf{A}_M$. Two options for the eigenvalue solutions to the optimization problem (6.24) are considered:

- *Option 1:* Solve for $\lambda_i^u \in \mathbb{R}$ subject to the constraint that $\lambda_i^u < 0$ for $i = 1,...,4$.
- *Option 2:* Solve for $\lambda_1, \lambda_2^{ur}, \lambda_2^{uc}, \lambda_4 \in \mathbb{R}$ subject to the constraint that $\lambda_1, \lambda_2^{ur}, \lambda_4 < 0$ and set $\lambda_2^u = \lambda_2^{ur} + i\lambda_2^{uc}$, $\lambda_3^u = \lambda_2^{ur} - i\lambda_2^{uc}$ (that is, λ_3^u is set to be the complex-conjugate of λ_2^u: $\lambda_3^u = \bar{\lambda}_2^u$).

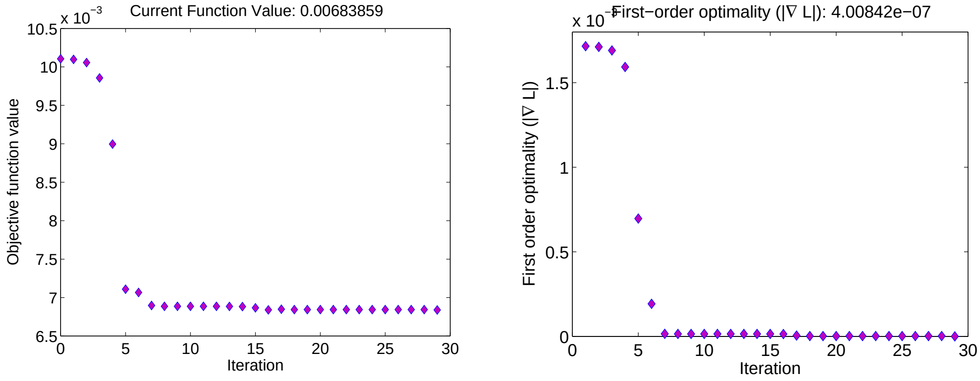
Per the discussion at the beginning of Section 6.2.4, Option 2 is more general than Option 1 and has fewer inequality constraints. The optimization problem (6.24) at the heart of Algorithm 2 is solved using the `fmincon` function in MATLAB's optimization toolbox. The `Algorithm` option required by this function is set to `interior-point`, and an initial guess of -1 for all the variables is used. For functional forms of the eigenvalues given by both Option 1 and Option 2, the optimization algorithm converges to a local minimum solution in less than 30 optimization iterations and 30 function evaluations. Table 6.5 shows some key information about the convergence of the optimization algorithm. The reader may observe that fewer iterations and function evaluations are required with Option 2 than with Option 1, which has more constraints. Figures 6.9 and 6.10 illustrate further the performance of the optimization algorithm for Option 1 and Option 2 respectively. For both options, the optimality conditions are satisfied to the specified tolerance at the value of the optimal solution[4].

An interesting question that arises is how the numbers in Table 6.5 change with M, the reduced basis size. Numerical experiments reveal that it is not necessarily the case that as M increases,

[4] For a constrained optimization problem such as (6.24), the first order optimality conditions require that the gradient of Lagrangian of the objective function $L(\lambda_1^u, ..., \lambda_L^u)$ be equal to zero, i.e., $\frac{\partial L}{\partial \lambda_k^u} = 0$ for all $k = 1,...,L$ where $L < M$ is the number of eigenvalues of $\mathbf{A}_M$ stabilized by Algorithm 2. A detailed discussion of this and other optimality conditions for the problem (6.24) can be found in [1, 77].

Table 6.5. Performance of `fmincon` interior point method for Algorithm 2 applied to ISS problem

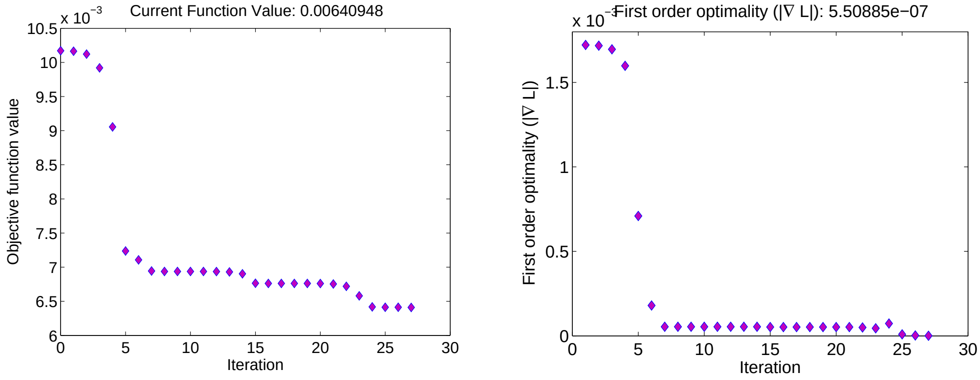
	Algorithm 2 with Option 1 (real eigenvalues)	Algorithm 2 with Option 2 (complex-conjugate eigenvalues)
# upper bound constraints	4	3
# optimization iterations	29	27
# function evaluations	30	30
first order optimality at convergence ($\|\nabla L\|$)	4.00×10^{-7}	5.51×10^{-7}



(a) Function value

(b) First order optimality ($|\nabla L|$)

Figure 6.9. Performance of interior point algorithm for Algorithm 2 with Option 1 (real eigenvalues) as a function of iteration number (ISS problem)



(a) Function value

(b) First order optimality ($|\nabla L|$)

Figure 6.10. Performance of interior point algorithm for Algorithm 2 with Option 2 (complex-conjugate eigenvalues) as a function of iteration number (ISS problem)

more optimization iterations and function evaluations are required to obtain the solution to the optimization problem (6.24). The performance of the interior point method depends on a number of factors, including: (1) the number of optimization dofs (i.e., the number of unstable eigenvalues of a ROM), (2) the number of upper bound constraints, (3) the character of the objective function, (4) the proximity of the initial guess to the optimal solution, and (5) the tolerances used in the optimization algorithm; not M, the reduced basis size, directly. Some additional performance results of the `fmincon` interior point method for Algorithm 2 applied to the ISS problem for different (larger) values of M are given in Appendix A.10 (Tables A.3–A.4). For the ISS problem, the ROM does in general become more unstable with increasing M, but more optimization iterations are not always required (Table A.3).

The eigenvalue solutions to the optimization problem (6.24) with both Option 1 and Option 2 are given in Table 6.6, compared with the values of the original unstable eigenvalues of $\mathbf{A}_M$. It is interesting to observe that the eigenvalues computed by the optimization algorithm with Option 1 are very different in their numerical values than those computed by the optimization algorithm with Option 2. Both are local minimizers of the optimization function (6.24). As discussed in Section 6.2.2, the optimization value is not guaranteed to be unique.

Table 6.6. Original (unstable) eigenvalues of $\mathbf{A}_M$ for $M = 20$ mode POD/Galerkin ROM and new stable eigenvalues computed using Algorithm 2 (ISS problem)

	Original Unstable $\mathbf{A}_M$	Algorithm 2 with Option 1 (real eigenvalues)	Algorithm 2 with Option 2 (complex-conjugate eigenvalues)
λ_1^u	2.42×10^2	-1.32	-1.98
λ_2^u	$3.29 \times 10^1 + 2.70 \times 10^1 i$	-2.12×10^{-2}	$-6.47 \times 10^{-3} + 1.42 \times 10^1 i$
λ_3^u	$3.29 \times 10^1 - 2.70 \times 10^1 i$	-2.13×10^{-2}	$-6.47 \times 10^{-3} - 1.42 \times 10^1 i$
λ_4^u	2.71	-1.33×10^{-4}	-1.38×10^{-4}

Table 6.7 gives the error in the ROM algorithm relative to the FOM output for an $M = 20$ POD/Galerkin ROM stabilized via Algorithm 2 with Option 1 and Option 2 for the ISS problem. Both options give a ROM with a relative error between 2.5% and 2.6%. This is a significant improvement in accuracy compared to the same ROM stabilized via Algorithm 1 (Table 6.4). Most importantly, in contrast to Algorithm 1, Algorithm 2 guarantees some level of accuracy in the stabilized ROM, as it minimizes the error in the ROM output by construction. Recall that the accuracy of a ROM stabilized via Algorithm 1 is unknown *a priori*, and it may require some trial and error to obtain a stabilized ROM with an acceptable error (Section 6.2.4).

Figure 6.11 shows the output computed from ROMs stabilized using Algorithm 2. The reader may observe that the stabilized ROM outputs are in much better agreement with the FOM output than the ROMs stabilized using Algorithm 1 (Figure 6.8).

Table 6.8 summarizes the CPU times for the time-integration step of the FOM, in addition to the CPU times for the offline and online stages of the $M = 20$ POD/Galerkin ISS ROM. The reader can observe by examining Table 6.8 that the $M = 20$ online stage of the POD/Galerkin ROM requires approximately 45 times less CPU time than the time-integration stage of the FOM. To offset the total preprocess time of the ROM (the time required to run the FOM to collect snapshots, calculate

Table 6.7. Relative errors in $M = 20$ POD/Galerkin ROM for ISS problem stabilized via Algorithm 2

ROM	$\mathscr{E}_{rel,\mathbf{y}}$
Unstabilized	1.74×10^{3}
ROM stabilized via Algorithm 2 with Option 1 (real eigenvalues)	2.59×10^{-2}
ROM stabilized via Algorithm 2 with Option 2 (complex-conjugate eigenvalues)	2.52×10^{-2}

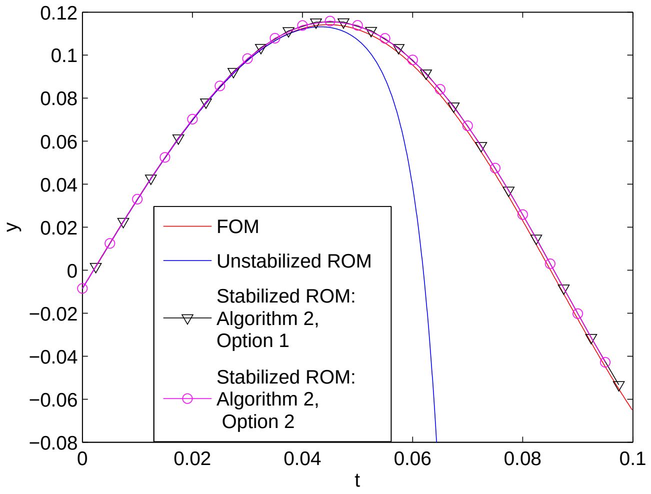


Figure 6.11. Outputs for $M = 20$ POD/Galerkin ROMs stabilized via Algorithm 2 vs. FOM output for ISS problem

the POD basis, perform the Galerkin projection, and solve the optimization problem (6.24)), the ROM would need to be run approximately 53 times. It is worthwhile to note that the optimization step of the model reduction, which consists of the solution of the optimization problem (6.24) is very fast: it takes less than a minute to complete.

Electrostatically actuated beam test case

The second numerical example is that of the electrostatically actuated beam, detailed at the beginning of this chapter. This example tests the performance of Algorithm 2 on a problem defined by a dense matrix $\mathbf{A}$ and from a different application than the ISS example. The example also demonstrates the methodology presented in this section on a larger-scale problem with has a forcing term $(\mathbf{B}_M\mathbf{u}(t) \neq \mathbf{0})$.

To generate the snapshots from which POD bases are constructed, the full order model (6.5) is solved using a backward Euler time integration scheme with an initial condition of $\mathbf{z}(0) = \mathbf{0}$ and

Table 6.8. Time-integration CPU times for ISS problem: FOM vs. $M = 20$ POD/Galerkin ROM stabilized via Algorithm 2

Model	Operations	CPU time (sec)
FOM	Time-Integration	1.71×10^2
ROM – offline stage	Snapshot collection (FOM time-integration)	1.71×10^2
	Loading of matrices/snapshots	6.99×10^{-2}
	POD	6.20
	Projection	8.18×10^{-3}
	Optimization*	2.28×10^1
ROM – online stage	Time-Integration	3.77

*Optimization times reported are means of the time required to solve (6.24) with real eigenvalues and the time required to solve (6.24) with complex-conjugate eigenvalues.

an input corresponding to a periodic on/off switching, i.e.,

$$\mathbf{u}(t) = \begin{cases} 1, & 0.005 < t < 0.01, 0.015 < t < 0.02, 0.03 < t < 0.035 \\ 0, & \text{otherwise.} \end{cases} \tag{6.35}$$

A total of $K_{max} = 1000$ snapshots are collected, every $\Delta t_{snap} = 5 \times 10^{-5}$ seconds, until time $T_{max} = 0.05$ seconds. From these snapshots, an $M = 17$ mode POD/Galerkin ROM is constructed. The ROM is found to be unstable, with four unstable eigenvalues. These eigenvalues have the following numerical values: $\lambda_1^u = 16,053$, $\lambda_2^u = 48.985$, $\lambda_3^u = 12.650$, $\lambda_4^u = 0.05202$. The basis size $M = 17$ is selected since this is the smallest integer for which the ROM exhibits an instability. It captures effectively 100% of the snapshot energy. Figure 6.12 shows the FOM output $\mathbf{y}(t)$ (in red) compared to the unstabilized ROM output (in blue). The relative error in the unstabilized ROM output (6.1) evaluates to `NaN` ("not a number") on a finite precision arithmetic machine due to overflow caused by the ROM instability. The $M = 17$ mode POD/Galerkin ROM is stabilized by Algorithm 2. Algorithm 1 is not considered for the sake of brevity, and since the superiority of Algorithm 2 has been established already.

Stabilization via Algorithm 2

The $M = 17$ POD/Galerkin ROM for the electrostatically actuated beam benchmark is stabilized using Algorithm 2. The four unstable eigenvalues of $\mathbf{A}_M$ will be denoted by λ_k^u for $k = 1, \dots, 4$. Similarly to the ISS test case, two options for the eigenvalue solutions to the optimization problem (6.24) will be considered:

- *Option 1:* Solve for $\lambda_i^u \in \mathbb{R}$ subject to the constraint that $\lambda_i^u < 0$ for $i = 1, \dots, 4$.
- *Option 2:* Solve for $\lambda_1^{ur}, \lambda_1^{uc}, \lambda_2^{ur}, \lambda_2^{uc} \in \mathbb{R}$ subject to the constraint that $\lambda_1^{ur}, \lambda_2^{ur}, < 0$ and set $\lambda_1^u = \lambda_1^{ur} + i\lambda_1^{uc}$, $\lambda_2^u = \lambda_1^{ur} - i\lambda_1^{uc}$, $\lambda_3^u = \lambda_2^{ur} + i\lambda_2^{uc}$, $\lambda_4^u = \lambda_3^{ur} - i\lambda_3^{uc}$ (that is, λ_3^u is taken to be the complex-conjugate of λ_2^u: $\lambda_3^u = \overline{\lambda_2^u}$).

Option 2 is more general than Option 1 and has fewer inequality constraints; however, Option 1 may be more consistent with the system dynamics, as the unstable eigenvalues of $\mathbf{A}$ are all real. As

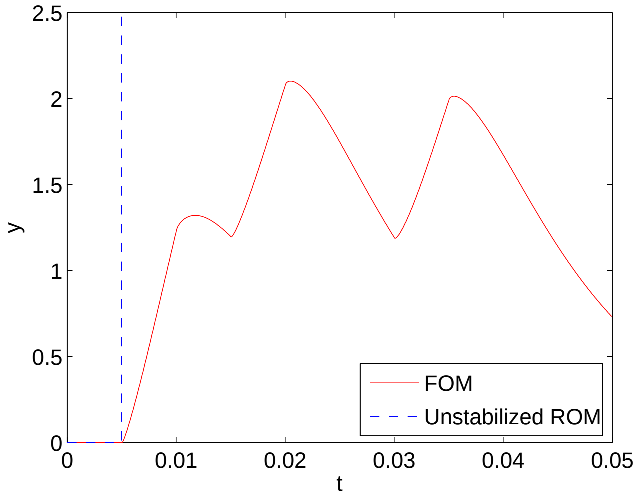


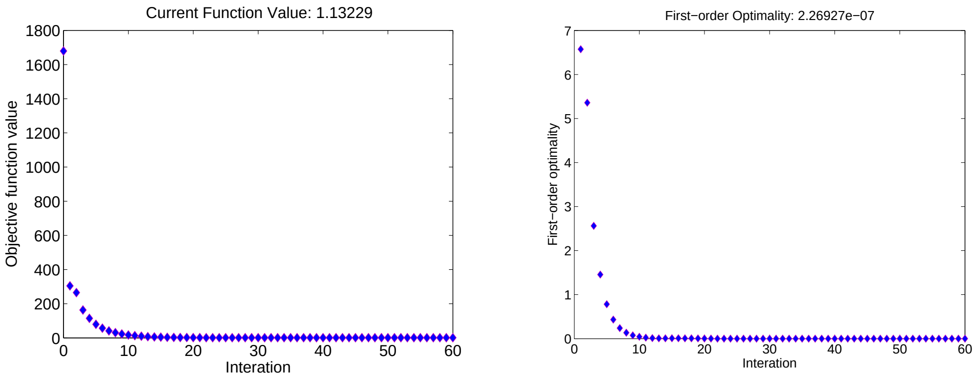
Figure 6.12. Outputs for $M = 17$ unstabilized POD/Galerkin ROM vs. FOM output for electrostatically actuated beam problem

before, the `fmincon` function in the MATLAB optimization toolbox will be used to solve the optimization problem (6.24), with the `Algorithm` option set to `interior-point` and an initial guess of -1 for all four variables optimized over in (6.24). For the functional form of the eigenvalues assumed in Option 1, the algorithm converges in 60 optimization iterations, and requires 64 function evaluations. For the functional form of the eigenvalues assumed in Option 2, which has less constraints than Option 1, fewer optimization iterations and function evaluations are required to achieve convergence: 31 optimization iterations, and 32 function evaluations. Some key information about the convergence of the optimization algorithm for both of these options is summarized in Table 6.9, and Figures 6.13 and 6.14. For both options, the optimality conditions are satisfied to the specified tolerance at the value of the optimal solution.

Table 6.9. Performance of `fmincon` interior point method for Algorithm 2 applied to electrostatically actuated beam problem

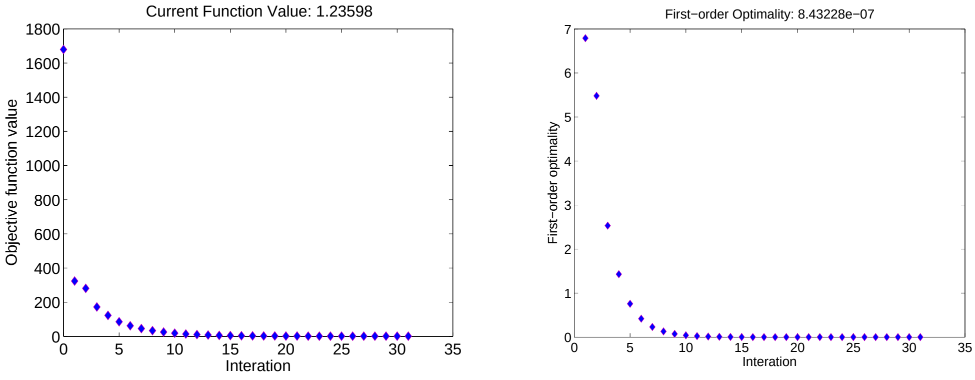
	Algorithm 2 with Option 1 (real eigenvalues)	Algorithm 2 with Option 2 (complex-conjugate eigenvalues)
# upper bound constraints	4	2
# optimization iterations	60	31
# function evaluations	64	32
first-order optimality at convergence ($\|\nabla L\|$)	2.27×10^{-7}	8.43×10^{-7}

Similarly to the ISS problem, Appendix A.10 (Tables A.5–A.6) gives some additional performance results of the `fmincon` interior point method for Algorithm 2 for different (larger) values of M. ROMs with larger basis sizes possess in general more unstable eigenvalues, and more optimization



(a) Function value **(b)** First order optimality ($|\nabla L|$)

Figure 6.13. Performance of interior point algorithm for Algorithm 2 with Option 1 (real eigenvalues) as a function of iteration number (electrostatically actuated beam problem)



(a) Function value **(b)** First order optimality ($|\nabla L|$)

Figure 6.14. Performance of interior point algorithm for Algorithm 2 with Option 2 (complex-conjugate eigenvalues) as a function of iteration number (electrostatically actuated beam problem)

iterations are required to obtain the solution of the optimization problem (6.24) using the interior point method.

The solutions obtained by Algorithm 2 with both Option 1 and Option 2 are given in Table 6.10, compared with the values of the original unstable eigenvalues of $\mathbf{A}_M$. As for the ISS benchmark, the eigenvalues computed by the optimization algorithm with Option 1 are different in their numerical values from those computed by the optimization algorithm with Option 2. This suggests that the optimization function (6.24) for this problem has multiple local minimizers/minima.

Table 6.11 gives the error in the ROM algorithm relative to the FOM output for an $M = 20$ POD/Galerkin ROM stabilized via Algorithm 2 with Option 1 and Option 2. For both options, the relative error in the stabilized ROM output is approximately 2%.

Finally, Figure 6.15 shows the output computed from ROMs stabilized using Algorithm 2. There

Table 6.10. Original (unstable) eigenvalues of $\mathbf{A}_M$ for $M = 17$ mode POD/Galerkin ROM and new stable eigenvalues computed using Algorithm 2 (electrostatically actuated beam problem)

	Original Unstable $\mathbf{A}_M$	Algorithm 2 with Option 1	Algorithm 2 with Option 2
λ_1^u	1.61×10^4	-6.88×10^5	$-1.16 \times 10^5 - 2.25 \times 10^4 i$
λ_2^u	4.90×10^1	-3.54×10^2	$-1.16 \times 10^5 + 2.25 \times 10^4 i$
λ_3^u	1.27×10^1	-1.97×10^4	$-3.32 \times 10^3 - 1.81 \times 10^2 i$
λ_4^u	5.20×10^{-2}	-1.40×10^4	$-3.32 \times 10^2 + 1.81 \times 10^2 i$

Table 6.11. Relative errors in $M = 17$ POD/Galerkin ROM for electrostatically actuated beam problem stabilized via Algorithm 2

ROM	$\mathscr{E}_{rel,\mathbf{y}}$
Unstabilized	`NaN`
ROM stabilized via Algorithm 2 with Option 1 (real eigenvalues)	1.94×10^{-2}
ROM stabilized via Algorithm 2 with Option 2 (complex-conjugate eigenvalues)	2.02×10^{-2}

is good agreement between the FOM output and stabilized ROM outputs.

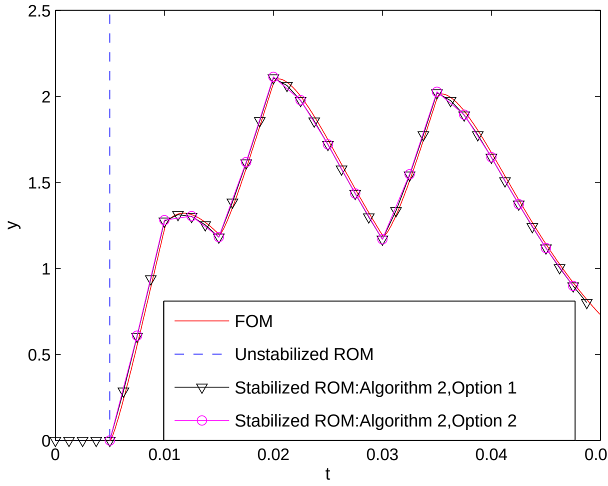


Figure 6.15. Outputs for $M = 17$ POD/Galerkin ROMs stabilized via Algorithm 2 vs. FOM output for electrostatically actuated beam problem

Table 6.12 summarizes some CPU times for the electrostatically-actuated beam problem: the CPU times for the FOM, as well as the CPU times for the offline and online stages of the $M = 17$ POD/Galerkin electrostatically-actuated beam ROM. The results in this table reveal that the online stage of the model reduction, the stage relevant to real-time calculations involving the ROM, took only 6.78 seconds, compared to 7.10×10^4 seconds for the time-integration stage of the FOM. To offset the total preprocess time of the ROM (the time required to run the FOM to collect snapshots, calculate the POD basis, perform the Galerkin projection, and solve the optimization problem (6.24)), the ROM would need to be run approximately 1×10^4 times. This large number of online ROM runs required to offset the offline ROM cost is due to the large CPU time associated with the FOM run for this large dense problem. As for the ISS problem, the optimization step of the model reduction does not contribute significantly to the CPU time of the offline stage of the ROM, taking just 1.5 minutes.

Table 6.12. Time-integration CPU times for electrostatically actuated beam problem: FOM vs. $M = 17$ POD/Galerkin ROM stabilized via Algorithm 2

Model	Operations	CPU time (sec)
FOM	Time-Integration	7.10×10^4
ROM – offline stage	Snapshot collection (FOM time-integration)	7.10×10^4
	Loading of matrices/snapshots	5.17
	POD	1.09×10^1
	Projection	2.55×10^1
	Optimization*	8.79×10^1
ROM – online stage	Time-Integration	6.78

*Optimization times reported are means of the time required to solve (6.24) with real eigenvalues and the time required to solve (6.24) with complex-conjugate eigenvalues.

6.3 Summary

It is demonstrated in this chapter that a discrete counterpart of the symmetry inner product developed in Chapter 5 is a weighted L^2 inner product obtained by solving a Lyapunov equation, first derived in [81] by Rowley *et al.* For completeness, this inner product, referred to as the "Lyapunov inner product", is re-derived here, and it is shown using the energy method that this inner product gives rise to stable ROMs constructed via discrete projection. The performance of POD ROMs constructed using the symmetry and Lyapunov inner products is assessed on several numerical examples for which POD ROMs constructed in the L^2 inner product manifest instabilities.

The key properties of the symmetry inner product and Lyapunov inner product are summarized in Table 6.13. Both inner products are weighted L^2 inner products and have the same origin: they are induced by the Lyapunov function for the governing system of equations. The symmetry inner product is a continuous inner product derived for a specific PDE system of the form (5.1). Projection in this inner product requires access to the governing PDEs, which gives rise to a projection algorithm that is embedded. The Lyapunov inner product is discrete, on the other hand, and operates on an LTI system of the form (4.18) arising from the discretization of a PDE of the form

Table 6.13. Comparison of symmetry inner product and Lyapunov inner product

Symmetry Inner Product (5.16)	**Lyapunov Inner Product** (6.10)
Continuous	Discrete
For linear PDE system of the form $\dot{\mathbf{q}}+\mathbf{A}_i\frac{\partial \mathbf{q}}{\partial x_i}+\mathbf{K}_{ij}\frac{\partial^2 \mathbf{q}}{\partial x_i \partial x_j}+\mathbf{G}\mathbf{q}=\mathbf{f}$	For linear ODE system of the form $\dot{\mathbf{x}}_N=\mathbf{A}\mathbf{x}_N$
Defined for unstable systems but time-stability of ROM is not guaranteed	Undefined for unstable systems
Induced by Lyapunov function for the system	Induced by Lyapunov function for the system
Equation specific	Black-box
Derived analytically in closed form	Computed numerically by solving a Lyapunov equation
Sparse	Dense

(4.17) in space using some numerical scheme, e.g., the finite element method. Projection in the Lyapunov inner product is therefore a black-box algorithm, as only the **A**, **B** and **C** matrices in (4.18) are needed; in particular, access to the governing equations is *not* required. The symmetric positive definite matrix that defines the Lyapunov inner product can also be computed numerically in a black-box fashion by solving a Lyapunov equation. The existence of a solution to this Lyapunov equation is certain only if the full order system (4.18) is stable; hence the Lyapunov inner product is not defined for unstable systems. In contrast, the symmetry inner product *is* defined for unstable systems. In this case, a ROM constructed in this inner product will be energy-stable, by construction. However, it will not be time-stable, i.e., it may produce (physical) solutions that are unbounded as $t \to \infty$. The discussion above may lead the reader to prefer the Lyapunov inner product to the symmetry inner product, as the former inner product can be computed in a black-box fashion for any stable linear system, and can be used to build a ROM for this system without accessing the PDEs. One of the biggest drawbacks of the Lyapunov inner product projection approach involves its large computational cost. To solve numerically the Lyapunov equation that defines this inner product, $\mathscr{O}(N^3)$ operations are required. Moreover, since the matrix that defines the Lyapunov inner product is typically dense (in contrast to the matrix defining the symmetry inner product, which is sparse), at least $\mathscr{O}(N^2)$ storage is required [43]. As a result, creating ROMs using the Lyapunov inner product may not be practical for systems of very large size. The Lyapunov inner product may nonetheless be preferable to balanced truncation, which requires the solution of two Lyapunov equations, and the storage of two Gramians, in addition to Cholesky and eigenvalue factorization of these Gramians. For large-scale unsteady problems, the symmetry inner product combined with the continuous projection approach is recommended by the authors, despite its more involved implementation.

In the second part of this chapter, a new approach for stabilizing unstable reduced order models for LTI systems through an *a posteriori* post-processing step applied to the algebraic ROM system is developed. This stabilization step consists of a reassignment of the eigenvalues of the ROM system matrix. First, it is shown how the system's eigenvalues can be modified by adding to the system a linear control term, and solving for the control matrix using full state feedback (a.k.a. pole placement) algorithms from control theory. This approach will yield a stable ROM provided the ROM system's unstable eigenvalues are controllable and observable; however, although the stabilized ROM will be stable, it may not be accurate. To ensure accuracy in the stabilized ROM, a second algorithm is developed, in which the eigenvalues of the stabilized ROM system are computed by solving a constrained nonlinear least-squares optimization problem in which the error in the ROM output is minimized. This problem is small ($< \mathscr{O}(M)$, where M is the number of dofs in the ROM), and therefore computationally inexpensive to solve using standard optimization algorithms. The second stabilization algorithm is the primary contribution of this chapter, but both algorithms are presented and evaluated, as the first algorithm led to the formulation of the second. The proposed ROM stabilization approaches are applicable to ROMs constructed using *any* choice of reduced basis for *any* application. The two algorithms are evaluated on two benchmarks: the ISS problem and the electrostatically actuated beam problem. Numerical tests reveal that the second algorithm effectively stabilizes an unstable ROM, delivering a modified ROM that is both stable as well as accurate.

6.4 Prospects for future work

An interesting and useful future research endeavor is the extension of the "ROM stabilization via eigenvalue reassignment" method (Algorithm 2) described in Section 6.2 to general nonlinear problems and predictive applications.

For nonlinear problems with stable fixed points and/or limit cycle solutions (e.g., the classical fluid mechanics problem involving flow around a cylinder), a natural extension of the algorithm would involve: (1) determining the stable fixed points of the system, (2) linearizing the system around these points, and (3) using the algorithms developed in Section 6.2 of this report to stabilize the linearized system. Extensions of Algorithm 2 to generic nonlinear problems would require a precise definition of stability, and perhaps a reformulation of the optimization problem (6.24), as it is likely not possible to work out analytically the expression for $\mathbf{y}_M(t)$ (6.23) for the general nonlinear case.

In order to use Algorithm 2 in predictive applications, the stabilized ROM must be constructed such that it is robust with respect to parameter changes. For this, an error indicator that would estimate the ROM error in a regime for which high-fidelity snapshots are not available is required.

Appendix A

A.1 Gronwall’s lemma

Gronwall’s lemma (also known as Gronwall’s inequality) allows one to bound a function that is known to satisfy a certain differential or integral inequality by the solution of the corresponding differential or integral equation [41]. The differential form of this inequality is used herein:

$$\dot{\mathbf{x}}(t) \leq \beta(t)\mathbf{x}(t) \;\;\Rightarrow\;\; \mathbf{x}(T) \leq \mathbf{x}(0)e^{\int_0^T \beta(s)ds}, \tag{A.1}$$

for $\beta \in L^2$, $t, T \geq 0$.

A.2 Linearized compressible Navier-Stokes equations with the viscous work terms included

In Section 5.2.2, the linearized compressible Navier-Stokes equations with the viscous work terms omitted were given (5.26). The full version of these equations, namely the equations with the viscous work terms included, is now given. In this case, the governing system (in non-dimensional form; Appendix A.3) is as follows:

$$\dot{\mathbf{q}}' + (\mathbf{A}_i - \tilde{\mathbf{A}}_i)\mathbf{q}'_{,i} - [\mathbf{K}_{ij}\mathbf{q}'_{,j}]_{,i} + \mathbf{G}\mathbf{q}' = \mathbf{0}, \tag{A.2}$$

instead of (5.26) where the $\tilde{\mathbf{A}}_i$ for $i = 1, 2, 3$ matrices are given by:

$$\begin{gathered}
\tilde{\mathbf{A}}_1 \equiv \frac{2}{ReT}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bar{\tau}_{11} & \bar{\tau}_{12}\bar{\tau}_{13} & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{\mathbf{A}}_2 \equiv \frac{2}{ReT}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bar{\tau}_{21} & \bar{\tau}_{22} & \bar{\tau}_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \\
\tilde{\mathbf{A}}_3 \equiv \frac{2}{ReT}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \bar{\tau}_{31} & \bar{\tau}_{32} & \bar{\tau}_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
\end{gathered} \tag{A.3}$$

Here, R denotes the dimensionless gas constant (Appendix A.3). The remaining matrices in (A.2), as well as the symbols appearing in (A.3), are defined in Section 5.2.2.

Note that the stability result of Corollary 5.2.1 breaks down in the case $\tilde{\mathbf{A}}_i \neq \emptyset$, i.e., in the presence of viscous work, which can offset the energy balance of the ROM system. To the authors' knowledge, the viscous work terms are invariably neglected from the linearized compressible Navier-Stokes equations by researchers studying energy-stability of these equations [47, 4]. The extension of the energy-stability symmetrization approach presented in Section 5.2 to the linearized compressible Navier-Stokes equations in which the viscous work terms are retained (A.2) would be a worthwhile future research endeavor.

A.3 Non-dimensionalization of the compressible Navier-Stokes equations

In this section, the non-dimensionalization of the compressible Navier-Stokes equations is detailed. Let u_{ref}, ρ_{ref}, T_{ref} and L_{ref} denote the reference (non-dimensionalization) values for the fluid velocities, density, temperature and length scales respectively. Given these values, the reference pressure, speed of sound and time scales are given by:

$$\begin{aligned} p_{ref} &= \rho_{ref} u_{ref}^2, \\ c_{ref} &= \sqrt{\gamma R T_{ref}}, \\ t_{ref} &= \tfrac{L_{ref}}{u_{ref}}, \end{aligned} \tag{A.4}$$

where R is the universal gas constant and γ is the ratio of specific heats. For viscous problems, let μ_{ref} denote reference value for the viscosity μ. Then, the reference value for the viscosity λ is

$$\lambda_{ref} = -\frac{2}{3}\mu_{ref}, \tag{A.5}$$

from Stokes' hypothesis. Typically the reference value for the thermal diffusivity κ_{ref} is given.

If T denotes the dimensional temperature and T^* denotes the dimensionless temperature, the relationship between the two is:

$$T^* = \frac{p^*}{\rho^* R^*}, \tag{A.6}$$

where p^* and ρ^* denote the dimensionless pressure and density respectively, and R^* denotes the dimensionless gas constant:

$$R^* = \frac{R T_{ref}}{c_{ref}^2}, \tag{A.7}$$

with R denoting the dimensional gas constant.

The Reynolds number, *Re*, is given by

$$Re = \frac{\rho_{ref} c_{ref} L_{ref}}{\mu_{ref}}, \tag{A.8}$$

and the Prandtl number, Pr, is given by

$$Pr = \frac{c_p \mu_{ref}}{\kappa_{ref}}, \tag{A.9}$$

with c_p denoting the specific heat at constant pressure. The reference Mach number is:

$$M_{ref} = \frac{u_{ref}}{c_{ref}}. \tag{A.10}$$

For the isentropic compressible Navier-Stokes equations (Section 5.3.3), the reference value for the enthalpy h, denoted by h_{ref}, is needed. It is straightforward to work out that h_{ref} is given by:

$$h_{ref} = \frac{p_{ref}}{\rho_{ref}} = u_{ref}^2. \tag{A.11}$$

(A.11) is also the reference value for the internal energy, e:

$$e_{ref} = u_{ref}^2. \tag{A.12}$$

A.4 Proof that the total energy inner product (5.59) for the compressible Navier-Stokes is a valid inner product

Here, it is verified that (5.59) is a valid inner product, by checking that the four inner product axioms hold.

(i) *Bilinearity:* $\left(\mathbf{q}^1+\mathbf{q}^2,\mathbf{q}^3\right)_E = \left(\mathbf{q}^1,\mathbf{q}^3\right)_E + \left(\mathbf{q}^2,\mathbf{q}^3\right)_E$.

$$\begin{aligned}\left(\mathbf{q}^1+\mathbf{q}^2,\mathbf{q}^3\right)_E &= \textstyle\int_\Omega \frac{1}{2}\Big((a^{(1)}+a^{(2)})b^{(3)} + a^{(3)}(b^{(1)}+b^{(2)}) + (a_1^{(1)}+a_1^{(2)})a_1^{(3)} + (a_2^{(1)}+a_2^{(2)})a_2^{(3)} \\ &\quad +(a_3^{(1)}+a_3^{(2)})a_3^{(3)}\Big)\, d\Omega \\ &= \textstyle\int_\Omega \frac{1}{2}\Big(a^{(1)}b^{(3)} + a^{(2)}b^{(3)} + a^{(3)}b^{(1)} + a^{(3)}b^{(2)} + a_1^{(1)}a_1^{(3)} + a_1^{(2)}a_1^{(3)} + a_2^{(1)}a_2^{(3)} \\ &\quad +a_2^{(2)}a_2^{(3)} + a_3^{(1)}a_3^{(3)} + a_3^{(2)}a_3^{(3)}\Big)\, d\Omega \\ &= \textstyle\int_\Omega \frac{1}{2}\Big(a^{(1)}b^{(3)} + a^{(3)}b^{(1)} + a_1^{(1)}a_1^{(3)} + a_2^{(1)}a_2^{(3)} + a_3^{(1)}a_3^{(3)}\Big)\, d\Omega \\ &\quad + \textstyle\int_\Omega \frac{1}{2}\Big(a^{(2)}b^{(3)} + a^{(3)}b^{(2)} + a_1^{(2)}a_1^{(3)} + a_2^{(2)}a_2^{(3)} + a_3^{(2)}a_3^{(3)}\Big)\, d\Omega \\ &= \left(\mathbf{q}^1,\mathbf{q}^3\right)_E + \left(\mathbf{q}^2,\mathbf{q}^3\right)_E.\end{aligned} \tag{A.13}$$

(ii) *Linearity:* $\left(\alpha\mathbf{q}^1,\mathbf{q}^2\right)_E = \alpha\left(\mathbf{q}^1,\mathbf{q}^2\right)_E$*, for* $\alpha \in \mathbb{R}$.

$$\begin{aligned}\left(\alpha\mathbf{q}^1,\mathbf{q}^1\right)_E &= \textstyle\int_\Omega \frac{1}{2}\Big(\alpha a^{(1)}b^{(2)} + \alpha a^{(2)}b^{(1)} + \alpha a_1^{(1)}a_1^{(2)} + \alpha a_2^{(1)}a_2^{(2)} + \alpha a_3^{(1)}a_3^{(2)}\Big)\, d\Omega \\ &= \alpha\left(\mathbf{q}^1,\mathbf{q}^2\right)_E.\end{aligned} \tag{A.14}$$

(iii) *Symmetry:* $\left(\mathbf{q}^1,\mathbf{q}^2\right)_E = \left(\mathbf{q}^2,\mathbf{q}^1\right)_E$.

$$\begin{aligned}\left(\mathbf{q}^1,\mathbf{q}^2\right)_E &= \int_\Omega \tfrac{1}{2}\left(a^{(1)}b^{(2)} + a^{(2)}b^{(1)} + a_1^{(1)}a_1^{(2)} + a_2^{(1)}a_2^{(2)} + a_3^{(1)}a_3^{(2)}\right) d\Omega \\ &= \left(\mathbf{q}^2,\mathbf{q}^1\right)_E .\end{aligned} \tag{A.15}$$

(iv) *Positive definiteness:* $(\mathbf{q},\mathbf{q})_E \geq 0$ *and* $(\mathbf{q},\mathbf{q})_E = 0$ *if and only if* $\mathbf{q} = \mathbf{0}$.

$$(\mathbf{q},\mathbf{q})_E = E_T \geq 0 \tag{A.16}$$

where E_T is defined in (5.60), since $\rho, e \geq 0$ (to be physical).

Suppose $(\mathbf{q},\mathbf{q})_E = 0$. Then, from (5.60):

$$\int_\Omega \left(ab + \tfrac{1}{2}[a_1^2 + a_2^2 + a_3^2]\right) d\Omega \quad = 0, \tag{A.17}$$

or

$$\tfrac{1}{2}\int_\Omega [a_1^2 + a_2^2 + a_3^2] d\Omega \quad = -\int_\Omega ab d\Omega. \tag{A.18}$$

The left-hand-side of (A.18) is necessarily ≥ 0. The right-hand-side is necessarily ≤ 0 since $ab = a^2 e = \rho e$ and $\rho, e \geq 0$. The only way for this to be true is if $\mathbf{q} = \mathbf{0}$.

□

A.5 Boundary conditions for compressible fluid ROMs constructed via continuous projection

Per the discussion in Section 4.3.3, ROMs constructed using the continuous projection approach may not automatically inherit the boundary conditions of the underlying FOM. Let $\Omega \in \mathbb{R}^3$ denote the domain on which a problem of interest is posed, and let $\partial\Omega$ denote the boundary of this domain. It is useful to partition $\partial\Omega$ as follows:

$$\partial\Omega = \partial\Omega_F \cup \partial\Omega_W \cup \partial\Omega_I, \tag{A.19}$$

with $\partial\Omega_F \cap \partial\Omega_W \cap \partial\Omega_I = \emptyset$. In (A.19), $\partial\Omega_F$, $\partial\Omega_W$ and $\partial\Omega_I$ denote the far-field, solid wall and inflow boundaries, respectively. Figure A.1 illustrates the partition of $\partial\Omega$ (A.19) for a cavity geometry. In this figure, boundary 1 represents $\partial\Omega_I$, boundary 2 represents $\partial\Omega_F$, and boundary 3 represents $\partial\Omega_W$.

For viscous fluid problems, the relevant boundary condition on $\partial\Omega_W$ is typically a no-slip condition on the velocities and adiabatic wall condition on the temperature [55]; for inviscid fluid problems, a slip wall condition on the velocities is the physically relevant condition on $\partial\Omega_W$ [57].

The far-field boundary $\partial\Omega_F$ is an artificial boundary introduced due to the fact that the boundary Ω used in a fluid simulation is by construction finite, in contrast to the infinite physical space

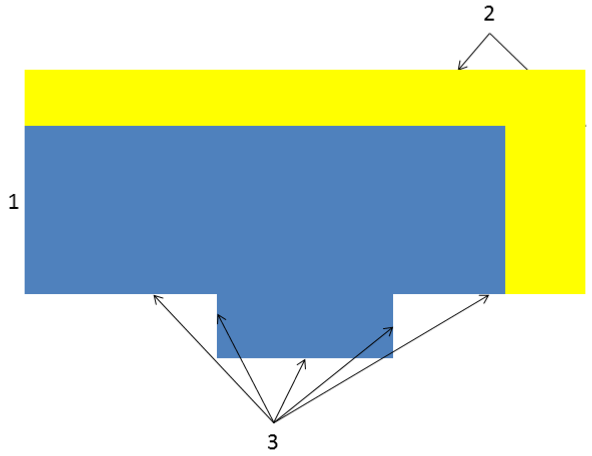


Figure A.1. Partition of boundary $\partial\Omega$ for the cavity configuration (1 = $\partial\Omega_I$ = inflow boundary, 2 = $\partial\Omega_F$ = far-field boundary, 3 = $\partial\Omega_W$ = solid wall boundary)

on which the physical problem is posed. Without far-field boundary conditions, non-physical reflections of unsteady waves may be observed in the far-field. These unwanted reflections can affect the accuracy of the simulation and possibly lead to numerical instability.

At the inflow boundary $\partial\Omega_I$, an inflow solution profile is typically prescribed using characteristic variables [57].

One way to implement boundary conditions in a ROM constructed using the continuous projection method is through a weak formulation. Consider a system of conservation laws of the form (5.1). Let $\boldsymbol{\phi}_k(\mathbf{x})$ denote a reduced basis mode. Projecting (5.1) onto $\boldsymbol{\phi}_k$ and integrating the viscous terms by parts, the following is obtained:

$$\begin{aligned} \int_\Omega \dot{\mathbf{q}}\boldsymbol{\phi}_k d\Omega + \int_\Omega \mathbf{A}_i \frac{\partial \mathbf{q}}{\partial x_i}\boldsymbol{\phi}_k d\Omega + \int_\Omega \frac{\partial \mathbf{K}_{ij}}{\partial x_j}\frac{\partial \mathbf{q}}{\partial x_i}\boldsymbol{\phi}_k d\Omega + \int_\Omega \frac{\partial \boldsymbol{\phi}_k}{\partial x_j}\mathbf{K}_{ij}\frac{\partial \mathbf{q}}{\partial x_i} d\Omega \\ - \int_{\partial\Omega} \mathbf{K}_{ij}\boldsymbol{\phi}_k \frac{\partial \mathbf{q}}{\partial x_i} n_j d\Gamma + \int_\Omega \mathbf{G}\mathbf{q}\boldsymbol{\phi}_k d\Omega = \int_\Omega \mathbf{f}\boldsymbol{\phi}_k d\Omega, \end{aligned} \tag{A.20}$$

for $k = 1, ..., M$, where n_j is the j^{th} component of the outward facing normal to $\partial\Omega$. The reader can observe from (A.20) that if a homogeneous Dirichlet boundary condition is desired on $\partial\Omega$ (e.g., the no-slip boundary condition) and the reduced bases modes $\boldsymbol{\phi}_k$ satisfy this boundary condition, the boundary integral in (A.20) will vanish, and no special implementation of the boundary condition in the ROM is needed. Otherwise, the boundary condition is implemented by substituting the value of $\boldsymbol{\phi}_k$ and/or its derivatives into the boundary integral in (A.20). In the case of an inhomogeneous

Dirichlet boundary condition, it may be necessary to integrate the inviscid terms in (A.20) by parts as well in order to apply the boundary condition [57].

For a discussion of the implementation of solid wall boundary conditions in a compressible fluid ROM constructed using continuous projection, the reader is referred to [57, 19]. Here, some discussion of the far-field boundary condition and its implementation is provided. A numerical study of this boundary condition on ROM stability and accuracy was begun during this project, and it would be worthwhile to continue this work in the future.

Note that an additional way to ensure that a ROM solution respects the FOM boundary conditions in the case of the nonlinear compressible Navier-Stokes equations (one that may be studied in future work) is to factor out the base flow (represented by the first POD mode; Figure 5.15(a)) from the solution vector: $\mathbf{q}(\mathbf{x},t) - \bar{\mathbf{q}}(\mathbf{x})$. In the approach taken in Chapter 5, the nonlinear ROMs are constructed for the full state vector $\mathbf{q}(\mathbf{x},t)$. Suppose the POD modes satisfy some steady non-zero Dirichlet boundary conditions consistent with those in the FOM. If the base flow $\bar{\mathbf{q}}$ is not subtracted from the ROM solution (a linear combination of these modes), its value at the boundaries could evolve in time, leading to incorrect dynamics predicted by the ROM [14].

A.5.1 Non-reflecting far-field boundary conditions implemented via the sponge region method

As discussed above, an appropriate far-field boundary condition is one that will suppress the reflection of waves from the outer computational boundaries (boundary 2 in Figure A.1). This boundary condition, known as the non-reflecting boundary condition, can be implemented using a sponge region method. Let $\Omega_{sponge} \subset \Omega$ denote the sponge region, a region near the outflow boundary (shaded in yellow in Figure A.1). Suppose, without loss of generality, that the sponge region is given by:

$$\Omega_{sponge} \equiv \left\{x_2 : x_{2,s} \leq x_2 \leq x_{2,max}, x_1, x_3 \in \Omega\right\}, \tag{A.21}$$

where Ω is a domain whose boundaries are aligned with the x_1, x_2 and x_3 coordinate axes, with $\max_{x_2}\{\Omega\} = x_{2,max} \in \mathbb{R}$ and $x_{2,s} \in \mathbb{R}$, with $x_{2,s} < x_{2,max}$. Suppose also that the governing PDEs have the form:

$$\frac{\partial \mathbf{q}}{\partial t} + \mathscr{L}(\mathbf{q}) + \mathscr{N}(\mathbf{q}) = \mathbf{0}, \tag{A.22}$$

where $\mathscr{N}$ is the nonlinear operator, $\mathscr{L}$ is the linear operator, and $\mathbf{q}$ is the vector of unknowns. The sponge region implementation of the far-field non-reflecting boundary condition amounts to adding a source term $\mathbf{f}_{sponge}$ to (A.22) to yield a system:

$$\frac{\partial \mathbf{q}}{\partial t} + \mathscr{L}(\mathbf{q}) + \mathscr{N}(\mathbf{q}) = \mathbf{f}_{sponge}(\mathbf{q}), \tag{A.23}$$

where

$$\mathbf{f}_{sponge}(\mathbf{q}) = \begin{cases} \sigma_{sponge}\left(\frac{x_2 - x_{2,s}}{x_{2,max} - x_{2,s}}\right)\left(\frac{\mathbf{q}_{target} - \mathbf{q}}{\Delta t}\right), & \text{in } \Omega_{sponge} \\ 0, & \text{otherwise.} \end{cases} \tag{A.24}$$

In (A.24), $\mathbf{q}_{target}$ is a vector of "target" values for the variables in the sponge region Ω_{sponge}, and $\sigma_{sponge} \in \mathbb{R}$ is a parameter controlling the "strength" of the sponge region. The reader can observe by examining (A.24) that the sponge region implementation of the far-field boundary condition is effectively a penalty formulation, with σ_{sponge} representing a penalty parameter that penalizes the deviation of $\mathbf{q}$ from $\mathbf{q}_{target}$ in the far-field. The symbol Δt in (A.24) is the time step used in advancing numerically the simulation forward in time.

Plugging (A.24) into (A.23) and rearranging, the following is obtained:

$$\frac{\partial \mathbf{q}}{\partial t} + \mathscr{L}(\mathbf{q}) + \mathscr{L}_{sponge}(\mathbf{q}) + \mathscr{N}(\mathbf{q}) = \mathbf{s}_{sponge}, \tag{A.25}$$

where

$$\mathscr{L}_{sponge}(\mathbf{q}) = \frac{\sigma_{sponge}}{\Delta t}\left(\frac{\max\{x_2 - x_{2,s}, 0\}}{x_{2,max} - x_{2,s}}\right)\mathbf{q}, \tag{A.26}$$

and

$$\mathbf{s}_{sponge} = \frac{\sigma_{sponge}}{\Delta t}\left(\frac{\max\{x_2 - x_{2,s}, 0\}}{x_{2,max} - x_{2,s}}\right)\mathbf{q}_{target}. \tag{A.27}$$

The operator defined by $\mathscr{L}_{sponge}$ (A.26) gives rise to a mass-like matrix upon discretization. The vector (A.27) is a forcing (load vector) term.

In the special case of the linearized PDEs, the system with the sponge layer source term has the form:

$$\frac{\partial \mathbf{q}'}{\partial t} + \mathscr{L}(\mathbf{q}') = \mathbf{f}_{sponge}(\mathbf{q}'), \tag{A.28}$$

where $\mathbf{q}' = \mathbf{q} - \bar{\mathbf{q}}$ is the fluctuation vector, with $\bar{\mathbf{q}}$ denoting the (steady) mean flow. In this case,

$$\mathbf{f}_{sponge}(\mathbf{q}') = \begin{cases} \sigma_{sponge}\left(\frac{x_2 - x_{2,s}}{x_{2,max} - x_{2,s}}\right)\left(\frac{\mathbf{q}'_{target} - \mathbf{q}'}{\Delta t}\right), & \text{in } \Omega_{sponge} \\ 0, & \text{otherwise} \end{cases} \tag{A.29}$$

where $\mathbf{q}'_{Target} = \mathbf{q}_{Target} - \bar{\mathbf{q}}$.

The capability to specify a non-reflecting boundary condition via the sponge region method has been added to `Spirit` as a part of this LDRD project. The implementation of this boundary condition has been verified on a simple test case involving inviscid flow inside a duct (below). The example demonstrates that a ROM with no boundary conditions constructed from a FOM in which the non-reflecting far-field boundary condition is specified may not reproduce accurately the FOM solution dynamics. This issue can be remedied by implementing the far-field boundary condition in the ROM using the sponge region method. It would be worthwhile to examine the effect of the far-field non-reflecting boundary condition on the compressible cavity simulations of interest in future work. It is conjectured that including the far-field boundary condition can improve the stability properties of a ROM by damping out any reflections that propagate back into the domain through the outflow boundary, which could destabilize the ROM. It is also conjectured, as suggested in [39], that if the domain is large enough, the implementation of the far-field boundary condition may not be necessary.

Proof of concept: non-reflecting far-field sponge region boundary condition applied to duct pressure pulse test case

The implementation of the non-reflecting far-field boundary condition via the sponge region method in `Spirit` is tested on a 1D test case involving an inviscid pressure pulse in a duct. The domain $\Omega = (0,50) \in \mathbb{R}$ is discretized using 501 points, and extruded in the x_2 and x_3 directions by one element to yield a 3D mesh for the `Spirit` run. The free-stream and reference conditions are as follows:

$$\begin{gathered} p_{ref} = 101,325 \text{ Pa}, \\ T_{ref} = 300 \text{ K}, \\ u_{ref} = 104.4 \text{ m/s}, \\ \rho_{ref} = 1.172 \text{ kg/m}^3. \end{gathered} \tag{A.30}$$

The problem is inviscid. The pressure pulse is triggered by a sinusoidal forcing for the x_1– momentum equation of the form

$$f_{u_1}(t) = 0.005 u_{ref} \cos(200\pi t), \tag{A.31}$$

in the region for which $x_1 \in (1, 1.5)$. In the high-fidelity fluid code `SIGMA CFD`, a sponge layer far-field boundary condition is prescribed near the outflow boundary

$$\Omega_{sponge} = \{x_1 : 40 \leq x_1 \leq 50\}. \tag{A.32}$$

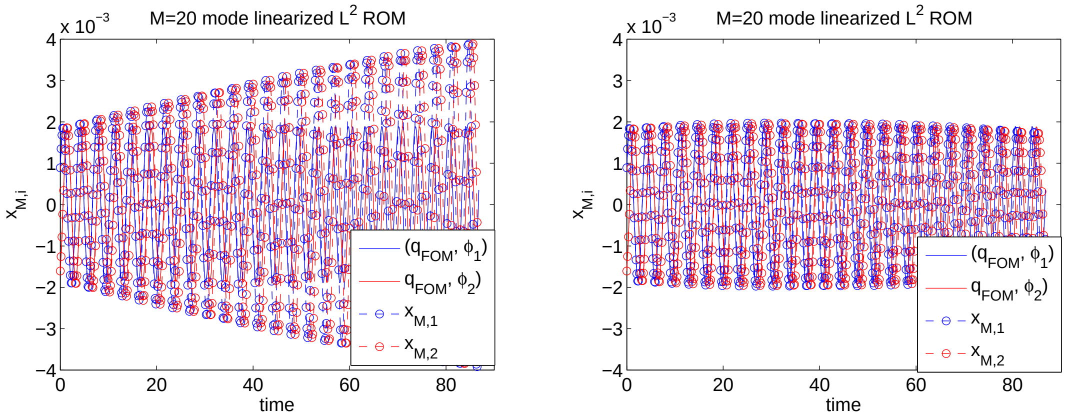
The boundary condition is of the form (A.24) but with x_1 replacing x_2. The parameters σ_{sponge} and $\mathbf{q}_{Target}$ are as follows:

$$\sigma_{sponge} = 0.01, \tag{A.33}$$

$$\mathbf{q}_{target} = \begin{pmatrix} u_{1,target} \\ u_{2,target} \\ u_{3,target} \\ \rho_{target} \\ T_{target} \end{pmatrix} = \begin{pmatrix} 104.4 \text{ m/s} \\ 0 \\ 0 \\ 1.172 \text{ kg/m}^3 \\ 300 \text{ K} \end{pmatrix}, \tag{A.34}$$

(in dimensional variables). The high-fidelity simulation of the duct pressure pulse test case is run in `SIGMA CFD` until time $T_{max} = 0.5$ seconds. A total of $K_{max} = 1000$ snapshots are collected (taken every $\Delta t_{snap} = 5 \times 10^{-4}$ seconds), from which a 20 mode POD basis is computed. A 20 mode POD/Galerkin ROM is then constructed for the linearized compressible Euler equations[1] using the L^2 inner product, both with ($\sigma_{sponge} = 0.01$) and without ($\sigma_{sponge} = 0$) the sponge region boundary condition applied in the far-field. Figure A.2 shows a time history of the ROM coefficients $a_1(t)$ and $a_2(t)$ compared with the projection of modes one and two onto the snapshots (5.69) for both cases. The results are as expected: the ROM with the sponge layer enforcement of the far-field boundary condition (Figure A.2(b)) represents much better the snapshot dynamics than the ROM without the sponge layer enforcement of the far-field boundary condition (Figure A.2(a)). Since the POD modes do not satisfy strongly the far-field boundary condition, the latter ROM is not fully consistent with the FOM.

[1] The linearized equations are appropriate here as the problem dynamics are effectively linear.



(a) No sponge ($\sigma_{sponge} = 0$) **(b)** Sponge ($\sigma_{sponge} = 0.01$)

Figure A.2. Time history of modal amplitudes for $M = 20$ mode L^2 linear ROMs (duct pressure test case)

A.6 Additional results for viscous laminar cavity problem with $K_{max} = 500$ snapshots

In this section, some additional results for the viscous laminar cavity problem (considered in Section 5.5.3) are given. The same Reynolds numbers, $Re \approx 1000$ and $Re \approx 1500$, as before are considered. Now, however, a total of $K_{max} = 500$ snapshots are collected from `SIGMA CFD`, taken every $\Delta t_{snap} = 1.0 \times 10^{-4}$ seconds, starting at time $t = 5.0 \times 10^{-2}$ seconds until time $T_{max} = 0.1$ seconds. The snapshots are used to construct POD bases of size 5, 10, 15 and 20 modes in the various inner products evaluated (for each Reynolds number considered), as before, but now the ROMs are run up to time $T_{max} = 0.1$ seconds. The snapshot energies captured by the POD bases are approximately the same as before (Table 5.8), and the decay of the singular values of the snapshot matrix is also comparable (Figure 5.16).

Tables A.1–A.2 reveal that the ROMs evaluated (Table 5.2) do not perform very well for this instance of the problem. Most of the ROMs go unstable. The most stable and accurate ROMs are those constructed for the isentropic compressible Navier-Stokes equations. It is interesting that the isentropic ROMs constructed in the stagnation energy and stagnation enthalpy inner products (5.49) exhibit an instability. The cause of this instability should be investigated in future work. It is worthwhile to comment that the Reynolds numbers considered here are much higher than the Reynolds number considered in [81], an earlier work in which the effectiveness of ROMs constructed using the stagnation energy and stagnation enthalpy inner products (5.49) for the isentropic compressible Navier-Stokes equations was evaluated. As noted earlier, the use of an energy inner product to do the Galerkin projection step of the model reduction does not guarantee that the stability of an equilibrium point other than the origin is preserved, nor does it guarantee that the stability of limit cycles is preserved [81]. Moreover, an equilibrium point of a dynamical system is not necessarily an attractor of the system. Hence, a ROM constructed in an energy inner product may not preserve a Navier-Stokes attractor. This may be what is happening here. It is also possible that boundary effects at the far-field boundary of the cavity are destabilizing the ROM (Appendix

A.5). This can be remedied by implementing a non-reflecting boundary condition at the far-field boundary in the ROM or solving the problem on a larger domain so that the boundary effects are negligible [39].

It is noted that a ROM with much better stability properties was obtained when the snapshot set employed to compute the POD basis includes the initial transient present in the high-fidelity solution between time $t = 0$ and time $t = 5.0 \times 10^{-2}$ seconds (results not shown here). Given that the quality of the ROM seems to be highly dependent on which snapshots are employed to construct the POD basis, it would be worthwhile to examine the effect of various snapshot collection strategies (e.g., [70, 82, 15]) on ROM stability and accuracy in future work.

Table A.1. ROM relative errors $\mathscr{E}_{rel,\mathbf{q}}$ (viscous laminar cavity, $Re \approx 1000$, $K = 500$ snapshots)

Physics	Inner product	$M = 5$	$M = 10$	$M = 15$	$M = 20$
Linearized compr N-S	L^2	5.56	5.63	34.0	33.3
	symm	5.05	5.11	27.5	27.1
Nonlinear compr isentropic N-S	L^2	7.99×10^{-2}	5.96×10^{-2}	9.39×10^{-2}	8.34×10^{-2}
	stag ene	3.94×10^{-2}	4.06×10^{-2}	NaN	NaN
	stag enth	4.05×10^{-2}	4.02×10^{-2}	NaN	NaN
ζ–form nonlinear compr N-S	L^2	7.34×10^{-1}	2.54×10^{-1}	1.58×10^{-1}	NaN
Nonlinear compr N-S	L^2	NaN	1.27×10^{-1}	NaN	7.06×10^{-2}

Table A.2. ROM relative errors $\mathscr{E}_{rel,\mathbf{q}}$ (viscous laminar cavity, $Re \approx 1500$, $K = 500$ snapshots)

Physics	Inner product	$M = 5$	$M = 10$	$M = 15$	$M = 20$
Linearized compr N-S	L^2	9.13	9.26	9.08	13.2
	symm	8.35	8.55	8.54	11.2
Nonlinear compr isentropic N-S	L^2	9.34×10^{-2}	1.26×10^{-1}	4.43×10^{-2}	8.44×10^{-2}
	stag ene	5.28×10^{-2}	5.54×10^{-2}	6.58×10^{-2}	NaN
	stag enth	5.59×10^{-2}	5.81×10^{-2}	6.58×10^{-2}	NaN
ζ–form nonlinear compr N-S	L^2	7.98×10^{-1}	NaN	NaN	NaN
Nonlinear compr N-S	L^2	NaN	2.41×10^{-1}	NaN	1.88×10^{-1}

A.7 Balanced truncation algorithm for model reduction

The balanced truncation algorithm, first introduced by Moore [73], assumes a semi-discrete full order model of the form (4.18). The linear system (4.18) is first transformed into a balanced form that isolates observable and reachable (or controllable) modes. This is achieved by simultaneously diagonalizing the reachability (or controllability) and observability Gramians. The reachability (or controllability) Gramian (Chapter 30 of [23])

$$\mathbf{P} \equiv \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt, \tag{A.35}$$

is the unique symmetric (at least) positive semi-definite solution of the Lyapunov equation

$$\mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{0}. \tag{A.36}$$

The observability Gramian (Chapter 30 of [23])

$$\mathbf{Q} \equiv \int_0^\infty e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C} e^{\mathbf{A}t} dt, \tag{A.37}$$

is the unique symmetric (at least) positive semi-definite solution of the Lyapunov equation

$$\mathbf{A}^T\mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{0}. \tag{A.38}$$

It will be assumed herein that the matrix $\mathbf{A}$ defining the full order system (4.18) is stable, i.e., it has no eigenvalues with a positive real part. It will also be assumed $(\mathbf{A},\mathbf{C})$ is observable and $(\mathbf{A},\mathbf{B})$ is reachable (controllable). If this is true, the Lyapunov equations (A.36) and (A.38) will have positive definite solutions $\mathbf{P}$ and $\mathbf{Q}$ respectively (Chapter 6 of [62]). For a discussion of balanced truncation applied to unstable systems, the reader is referred to [21].

The balanced truncation algorithm is summarized below for the specific case of real system matrices[2] $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. First, the reachability Gramian $\mathbf{P}$ is obtained by solving the Lyapunov equation (A.36). Next, the observability Gramian $\mathbf{Q}$ is obtained by solving the Lyapunov equation (A.38). The Cholesky factorization of $\mathbf{P}$ is computed,

$$\mathbf{P} = \mathbf{U}\mathbf{U}^T. \tag{A.39}$$

followed by an eigenvalue decomposition of $\mathbf{U}^T\mathbf{Q}\mathbf{U}$:

$$\mathbf{U}^T\mathbf{Q}\mathbf{U} = \mathbf{K}\mathbf{\Sigma}^2\mathbf{K}^T. \tag{A.40}$$

The balancing transformation matrices:

$$\mathbf{T}_{bal} = \mathbf{\Sigma}^{1/2}\mathbf{K}^T\mathbf{U}^{-1}, \quad \mathbf{T}_{bal}^{-1} = \mathbf{U}\mathbf{K}\mathbf{\Sigma}^{-1/2}, \tag{A.41}$$

can now be computed[3], where the entries of $\mathbf{\Sigma}$ are in decreasing order. The change of variables $\tilde{\mathbf{x}}_N(t) = \mathbf{T}_{bal}\mathbf{x}_N(t)$ is applied to the full-order LTI system (4.18) to yield:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}_N(t) &= \mathbf{T}_{bal}\mathbf{A}\mathbf{T}_{bal}^{-1}\tilde{\mathbf{x}}_N(t) + \mathbf{T}_{bal}\mathbf{B}\mathbf{u}_P(t), \\ \mathbf{y}_{QN}(t) &= \mathbf{C}\mathbf{T}_{bal}^{-1}\tilde{\mathbf{x}}_N(t). \end{aligned} \tag{A.42}$$

[2] In the case these matrices are complex, the transpose operation ^{T} in the algorithm (and all analysis of this algorithm) should be replaced with a Hermitian transpose ^{H}.

[3] In practice, the transformation matrices (A.41) are typically computed as $\mathbf{T}_{bal} = \mathbf{V}^T\mathbf{Z}^T$, and $\mathbf{T}_{bal}^{-1} = \mathbf{U}\mathbf{W}$, where $\mathbf{Z}$ is the Cholesky factor of the observability Gramian ($\mathbf{Q} = \mathbf{Z}\mathbf{Z}^T$), and $\mathbf{W}$ is the left singular vector of $\mathbf{U}^T\mathbf{Z}$ ($\mathbf{U}^T\mathbf{Z} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T$). This is due to numerical stability issues that could arise in computing $\mathbf{\Sigma}^{-1/2}$ in (A.41).

Next, the matrices $\tilde{\mathbf{A}} \equiv \mathbf{T}_{bal}\mathbf{A}\mathbf{T}_{bal}^{-1}$, $\tilde{\mathbf{B}} \equiv \mathbf{T}_{bal}\mathbf{B}$, $\tilde{\mathbf{C}} \equiv \mathbf{C}\mathbf{T}_{bal}^{-1}$ are partitioned as follows:

$$\begin{gathered}\tilde{\mathbf{A}} = \left(\begin{array}{c|c} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \hline \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{array}\right), \quad \tilde{\mathbf{B}} = \left(\begin{array}{c} \tilde{\mathbf{B}}_1 \\ \hline \tilde{\mathbf{B}}_2 \end{array}\right), \\ \tilde{\mathbf{C}} = \left(\begin{array}{c|c} \tilde{\mathbf{C}}_1 & \tilde{\mathbf{C}}_2 \end{array}\right). \end{gathered} \tag{A.43}$$

Here, the blocks with subscript 1 correspond to the most observable and reachable states, and blocks with subscript 2 correspond to the least observable and reachable states. Finally, the reduced system for a ROM of size M is given by:

$$\begin{aligned} \dot{\mathbf{x}}_M(t) &= \mathbf{A}_M\mathbf{x}_M(t) + \mathbf{B}_M\mathbf{u}_P(t), \\ \mathbf{y}_{QM}(t) &= \mathbf{C}_M\mathbf{x}_M(t), \end{aligned} \tag{A.44}$$

where $\mathbf{A}_M = \tilde{\mathbf{A}}_{11}$, $\mathbf{B}_M = \tilde{\mathbf{B}}_1$, $\mathbf{C}_M = \tilde{\mathbf{C}}_1$. The left and right reduced bases are given respectively by:

$$\mathbf{\Psi}_M = \mathbf{T}_{bal}^T(:,1:M), \quad \mathbf{\Phi}_M = \mathbf{S}_{bal}(:,1:M), \tag{A.45}$$

where $\mathbf{S}_{bal} \equiv \mathbf{T}_{bal}^{-1}$.

In effect, balanced truncation is a method for computing the test and trial bases $\mathbf{\Psi}_M$ and $\mathbf{\Phi}_M$ in (4.20). Given the test and trial bases defined in (A.45), the ROM system matrices (A.44) can be obtained from the formulas (4.21). The entries of the diagonal matrix $\mathbf{\Sigma}$ in (A.41) are known as the Hankel singular values of the system (4.18). Assuming a ROM of size M has been constructed using balanced truncation, the following error bound on the output can be shown [95]:

$$||\mathbf{y}_{QN}(t) - \mathbf{y}_{QM}(t)||_2 \leq 2\sum_{i=M+1}^{N} \sigma_i ||\mathbf{u}_P(t)||_2. \tag{A.46}$$

Generally, balanced truncation is viewed as the “gold standard” in model reduction. Although it is not optimal in the sense that there may be other ROMs with smaller error norms, the approach has *a priori* error bounds that are close to the lowest bounds achievable by any reduced order model [79]. Unfortunately, balanced truncation becomes computationally intractable for systems of very large dimension (e.g., of size $N \geq 10,000$), and hence is not practical for many systems of physical interest [81]. This is due to the high computational cost of solving the Lyapunov equations (A.36) and (A.38) for the reachability and observability Gramians ($\mathscr{O}(N^3)$ operations). The storage requirements of balanced truncation can be prohibitive as well. Even efficient iterative schemes developed for large sparse Lyapunov equations compute the solution to (A.36) and (A.38) in dense form, and hence require $\mathscr{O}(N^2)$ storage [43]. Unlike POD, balanced truncation delivers ROMs that preserve stability of a stable system (4.18) [73], however.

A.8 Lyapunov inner product associated with balanced truncation

In comparing the steps of the balanced truncation algorithm with the discussion in Section 6.1, the reader may observe some similarities. In particular, both algorithms require the solution of a

Lyapunov equation for a Gramian used to transform and reduce the system. Here, this connection is investigated further. In particular, it is shown that the balanced truncation algorithm (Appendix A.7) may be viewed as a projection algorithm in a special Lyapunov inner product.

Suppose the stable LTI system (4.18) has been reduced using the balanced truncation model reduction algorithm summarized in Appendix A.7. In order to uncover the inner product associated with balanced truncation, several transformations are required.

The first step is to substitute (A.41) into (A.45). Then, the following expressions for the left and right bases are obtained:

$$\mathbf{\Psi}_M^T = \mathbf{T}_{bal}(1:M,:) = \mathbf{\Sigma}^{1/2}(1:M,:)\mathbf{K}^T\mathbf{U}^{-1}, \tag{A.47}$$

$$\mathbf{\Phi}_M = \mathbf{S}_{bal}(:,1:M) = \mathbf{UK\Sigma}^{-1/2}(:,1:M). \tag{A.48}$$

Remark that (A.47) and (A.48) satisfy the following identity:

$$\mathbf{\Sigma}^{-1}(1:M,1:M)\mathbf{\Psi}_M^T\mathbf{P} = \mathbf{\Phi}_M^T, \tag{A.49}$$

where **P** is the reachability Gramian (A.39). It follows that the ROM system matrices in (A.44) are:

$$\mathbf{A}_M = \mathbf{\Psi}_M^T\mathbf{A\Phi} = \mathbf{\Psi}_M^T\mathbf{AP}^T\mathbf{\Psi}_M\mathbf{\Sigma}^{-1}(1:M,1:M), \tag{A.50}$$

$$\mathbf{B}_M = \mathbf{\Psi}_M^T\mathbf{B}, \tag{A.51}$$

$$\mathbf{C}_M = \mathbf{C\Phi} = \mathbf{CP}^T\mathbf{\Psi}_M\mathbf{\Sigma}^{-1}(1:M,1:M). \tag{A.52}$$

Defining

$$\mathbf{z}_M(t) \equiv \mathbf{\Sigma}^{-1/2}(1:M,1:M)\mathbf{x}_M(t), \tag{A.53}$$

and employing the symmetry property of the reachability Gramian ($\mathbf{P} = \mathbf{P}^T$), (A.44) becomes:

$$\begin{array}{ll} \dot{\mathbf{z}}_M(t) & = \hat{\mathbf{\Psi}}_M^T\mathbf{AP}\hat{\mathbf{\Psi}}_M\mathbf{z}_M(t) + \hat{\mathbf{\Psi}}_M^T\mathbf{Bu}_P(t), \\ \mathbf{y}_{QM}(t) & = \mathbf{CP}\hat{\mathbf{\Psi}}_M\mathbf{z}_M(t), \end{array} \tag{A.54}$$

where

$$\hat{\mathbf{\Psi}}_M \equiv \mathbf{\Psi}_M\mathbf{\Sigma}^{-1/2}(1:M,1:M). \tag{A.55}$$

It is clear that (A.54) defines a projection of the original LTI system (4.18) in an L^2 inner product weighted by the reachability Gramian matrix **P**. This matrix defines a true inner product in the case when **P** is symmetric positive-definite, which will hold if $(\mathbf{A},\mathbf{B})$ is reachable (controllable)[4].

A property of balanced truncation is that it preserves stability when applied to stable systems [42] (Appendix A.7). This result can be proven using the energy method. The proof is analogous to the proof of Theorem 6.1.1.

[4] Reachability (a.k.a. controllability) is a standard concept in control theory. The author is referred to [11] for a detailed discussion of reachability (controllability).

A.9 Jacobian of objective function in ROM stabilization optimization problem (6.24)

In this section, an analytic expression for the Jacobian of the objective function in the optimization problem (6.24) for the specific case when $\mathbf{u}(t) = \mathbf{0}$, $\mathbf{y} \in \mathbb{R}$ (there is a single output of interest), and $\lambda_i^u \in \mathbb{R}$ is derived. In many cases, it is possible to derive analytically the Jacobian of the objective function in (6.24) without these simplified assumptions, but this derivation will be problem-dependent (i.e., it will depend on the specific forcing $\mathbf{u}(t)$). Let $y^k \equiv \mathbf{y}^k \in \mathbb{R}$ and $y_M^k \equiv \mathbf{y}_M^k \in \mathbb{R}$. If $\mathbf{u}(t) = \mathbf{0}$, the objective function in (6.24) evaluates to:

$$f = ||\mathbf{F}||_2^2, \tag{A.56}$$

where

$$\mathbf{F} \equiv \begin{pmatrix} \mathbf{CS}\exp(\mathbf{D}t_1)\mathbf{S}^{-1}\mathbf{x}(0) - y^1 \\ \mathbf{CS}\exp(\mathbf{D}t_2)\mathbf{S}^{-1}\mathbf{x}(0) - y^2 \\ \vdots \\ \mathbf{CS}\exp(\mathbf{D}t_K)\mathbf{S}^{-1}\mathbf{x}(0) - y^K \end{pmatrix} \in \mathbb{R}^K. \tag{A.57}$$

Let $\mathbf{J}$ denote the Jacobian of f (A.56). The reader can verify that

$$\mathbf{J} = 2\mathbf{J}_F^T\mathbf{F} \in \mathbb{R}^L \tag{A.58}$$

where the $(k,l)^{th}$ entry of $\mathbf{J}_F$ is given by

$$J_F(k,l) = t_k \mathbf{CS}\exp(\hat{\mathbf{D}}_l t_k)\mathbf{S}^{-1}\mathbf{x}(0), \tag{A.59}$$

for $k = 1,..,K$ and $l = 1,...,L$. In equation (A.59),

$$\hat{\mathbf{D}}_l \equiv \begin{pmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & \lambda_l^u & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix} \in \mathbb{R}^{M\times M}, \tag{A.60}$$

that is, $\hat{\mathbf{D}}_l$ is a matrix with a single entry of λ_l^u in the position $(\hat{l},\hat{l})$, where $\hat{l}$ is the position of the l^{th} reassigned eigenvalue in the original matrix $\mathbf{D}$.

A.10 Additional performance results for ROM stabilization via eigenvalue reassignment Algorithm 2

The following tables give some additional performance results (the number of unstable eigenvalues, the number of upper bound constraints, the number of optimization iterations, the number of function evaluations, and the first order optimality at convergence) for Algorithm 2 applied to the ISS and electrostatically actuated beam problems considered in Section (**??**). These results enable one to study how these quantities change as M, the reduced basis size, is increased. The performance of the interior point method depends more on the number of dofs in the optimization problem (6.24), rather than the basis size M directly. For the problems considered herein, as M is increased, in general so does the number of unstable eigenvalues of the ROM.

Table A.3. Performance of `fmincon` interior point method for Algorithm 2 applied to ISS problem as a function of M (real eigenvalues)

M	20	40	60
# unstable eigenvalues	4	5	6
# upper bound constraints	4	5	6
# optimization iterations	29	58	45
# function evaluations	30	59	46
first-order optimality at convergence $(\lvert\nabla L\rvert)$	4.00×10^{-7}	9.88×10^{-7}	2.46×10^{-7}

Table A.4. Performance of `fmincon` interior point method for Algorithm 2 applied to ISS problem as a function of M (complex-conjugate eigenvalues)

M	20	40	60
# unstable eigenvalues	4	5	6
# upper bound constraints	3	3	3
# optimization iterations	27	50	62
# function evaluations	30	52	64
first-order optimality at convergence $(\lvert\nabla L\rvert)$	5.51×10^{-7}	2.46×10^{-7}	3.94×10^{-7}

Table A.5. Performance of `fmincon` interior point method for Algorithm 2 applied to electrostatically actuated beam problem as a function of M (real eigenvalues)

M	17	34	51
# unstable eigenvalues	4	10	14
# upper bound constraints	4	10	14
# optimization iterations	60	78	96
# function evaluations	64	82	100
first-order optimality at convergence $(\lvert\nabla L\rvert)$	2.27×10^{-7}	4.61×10^{-7}	2.13×10^{-7}

Table A.6. Performance of `fmincon` interior point method for Algorithm 2 applied to electrostatically actuated beam problem as a function of *M* (complex-conjugate eigenvalues)

M	17	34	51
# unstable eigenvalues	4	10	14
# upper bound constraints	2	5	7
# optimization iterations	31	35	78
# function evaluations	32	36	79
first-order optimality at convergence $(\lvert\nabla L\rvert)$	8.43×10^{-7}	6.20×10^{-6}	1.08×10^{-7}

A.11 Flow control using ROMs

Flow control refers to the ability to manipulate a fluid flow to change its behavior in a desirable way, e.g., to reduce the noise in the flow, to increase the efficiency of a combustion process, to modify the stability of a laminar flow, or to reduce structural vibration caused by a flow passing over a structure. There exist a variety of schemes for open-loop[5] as well as closed-loop[6] control of complex physical systems [11, 96]. Unfortunately, for high-fidelity fluid models, the discretized systems are often too large to be able to apply classical flow control methods. This is especially the case if an optimal controller is sought in real or near-real time. Because reduced order models are small and inexpensive by construction, they have a great potential for making the flow control problem feasible.

In the targeted compressible captive-carry problem, a flow control strategy that minimizes cavity oscillations (i.e., resonances) is sought. Suppose a high speed flow passes over the cavity (Figure A.3). A shear layer will form at the cavity's upstream corner. This shear layer will amplify disturbances in the flow and convect them downstream, scattering them into pressure fluctuations on the cavity's downstream wall. These pressure fluctuations will propagate back upstream, and excite further disturbances in the shear layer near the upstream corner. The result is a feedback loop of self-sustained cavity oscillations [53, 51]. The pressure fluctuations on the downstream wall of the cavity translate to large pressure loads within the cavity, which are undesirable as they can lead to damage of the cavity and/or components within the cavity. Hence, an optimal feedback controller is one that minimizes the pressure fluctuations on the downstream wall, and therefore the oscillations within the cavity.

In designing a closed-loop controller for the system described above, it is necessary to have a sensor (or set of sensors) and an actuator (or set of actuators). The former provides an output to be controlled, whereas the latter represents an input that can be tuned to achieve the desired output. Typically in flow control, the controller consists of actuation, either in the form of a body

[5] An open-loop controller is a non-feedback controller, i.e., a controller that does not use feedback to determine if its output has achieved the desired goal of the input.

[6] A closed-loop controller is a feedback controller, i.e., a controller consisting of a set of sensors for the measurement of some system parameter that can communicate with a set of actuators, which can subsequently alter the dynamics of the underlying system.

force or boundary actuator (sucking or blowing), in an upstream section of the domain. For the compressible cavity, a reasonable location to apply the actuation is in a region near the upstream corner of the cavity where the shear layer originates. A reasonable output of interest is the root-mean-square pressure fluctuation (p'_{rms}) halfway up the downstream wall of the cavity. This value correlates well with the pressure loads within the cavity and on the cavity walls. The flow control configuration is illustrated in Figure A.3.

Following standard control theory terminology [11, 96], let the “plant” refer to the physical system to be controlled, and let the “estimator” be the model used to estimate the state from which the controller will be calculated. Here, the plant is the high-fidelity model of the compressible cavity based on the nonlinear compressible Navier-Stokes equations. Because of its large computational cost, it is not feasible to use the high-fidelity simulation as the estimator. An alternative is to use as the estimator a ROM for the high-fidelity model, compute the controller based on the ROM, and then apply the ROM-based controller to the high-fidelity model (see Figure A.4 for an illustration of the general approach). This idea has been studied by a number of authors, e.g., Illingworth *et al.* [53], Rowley *et al.* [82], Barbagello *et al.* [17], Bagheri *et al.* [13], Ilak [52], mostly in the context of incompressible flow. These works have promoted the use linear control theory, i.e., using a *linear* low-dimensional model of the fluid flow to calculate a feedback controller for the original *nonlinear* system. Here, the idea is applied to the compressible Navier-Stokes equations. The ROM-based control approach explored as a part of this project is summarized in the following steps (illustrated in Figure A.4).

Step 1: Collect snapshots from a nonlinear high-fidelity CFD simulation for some set of inputs $\mathbf{u}(t)$, and construct a POD basis from t his snapshot set.

Step 2: Build a ROM for a linearized version of the governing fluid system.

Step 3: Compute the optimal controller, denoted by $u_{opt}(t)$, using the ROM.

Step 4: Apply the optimal ROM-based controller at the high-fidelity model level.

A particular kind of linear controller is amenable to the flow control problem of interest, namely a linear quadratic regulator (LQR) controller, described in detail in [11, 96]. The optimal LQR controller $u_{opt}(t)$ is one that minimizes (for the configuration of interest) the following cost functional:

$$J = \frac{1}{T}\int_0^T [p'^2_{wall} + \tau u^2_{opt}]dt, \tag{A.61}$$

where p'_{wall} is the pressure fluctuation at the downstream wall of the cavity and $\tau > 0$ is a parameter that controls the cost of the control u_{opt}. It can be shown [11, 96] that the solution to the LQR problem is obtained by solving a Lyapunov equation. The numerical solution of this equation is, in general, tractable only if it is computed for a relatively low-dimensional model.

The capability to apply body force actuators has been added to the `Spirit` code as a part of this LDRD project to enable flow control. As a preliminary step, the proposed ROM-based flow control approach outlined above is tested on a proof of concept example: a driven inviscid pulse test case. The design of LQR ROM-based controllers for the compressible cavity may be the subject of future work.

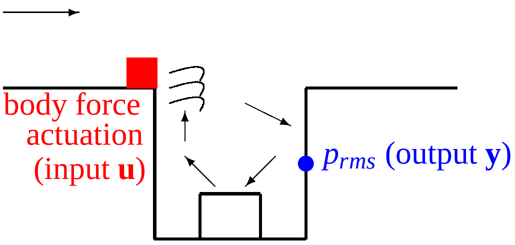


Figure A.3. Target cavity flow control problem

A.11.1 Proof of concept: control of a driven inviscid pulse

The ROM-based flow control algorithm outlined above is tested on a simpler problem than the targeted compressible cavity configuration, namely the problem of a driven inviscid pulse in a uniform base flow. Consider the compressible Euler equations (5.22) on the 2D domain $\Omega = (-1,1)^2$ in a uniform base flow having the following values:

$$\begin{aligned} \bar{p} &= 10.1325 \text{ Pa}, \\ \bar{T} &= 300 \text{ K}, \\ \bar{\rho} = \frac{\bar{p}}{R\bar{T}} &= 1.17 \times 10^{-4} \text{ kg/m}^3, \\ \bar{u}_1 = \bar{u}_2 = \bar{u}_3 &= 0.0 \text{ m/s}, \\ \bar{c} &= 347.9693 \text{ m/s}. \end{aligned} \tag{A.62}$$

Driving the flow is a force for the x_2–momentum equation. The force is given by the following oscillatory function

$$f_{u_2}(\mathbf{x},t) = (1 \times 10^{-4})\cos(2000\pi t), \tag{A.63}$$

and is applied in four elements near the center of the domain, for $\mathbf{x} \in (-0.1, 0)^2$ (Figure A.5). The high-fidelity solution is obtained in the `SIGMA CFD` code using a mesh with 3362 nodes. The high-fidelity simulation is run until time $T_{max} = 5 \times 10^{-2}$ seconds. A total of $K_{max} = 2500$ snapshots (saved every $\Delta t_{snap} = 2 \times 10^{-5}$ seconds) are collected and used to construct a 20 mode POD basis using the symmetry inner product (with **H** given by (5.25)).

The following flow control problem for the driven pulse example is formulated:

Compute the body force actuation $u_{opt}(t)$ in the region $\mathbf{x} \in (-0.1, 0)^2$ such that the root-mean-square pressure fluctuation p'_{rms} at the point $\mathbf{x} = (1,0)$ is minimized (black dot in Figure A.5).

The controller is computed using the LQR approach and a symmetry ROM estimator based on the linearized compressible Euler equations (Section 5.2.1). Once the optimal input is calculated, it is applied to the high-fidelity simulation for the purpose of testing its efficacy. Effectively, this is a verification problem, as it is known *a priori* what the controller should be: $u_{opt}(t) = -f_{u_2}(\mathbf{x},t)$.

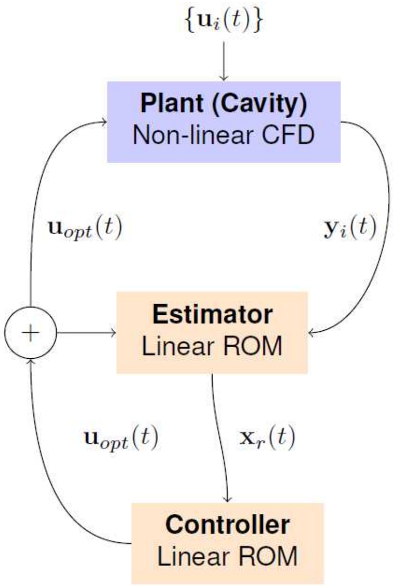


Figure A.4. POD-Based cavity flow control road map

First, the 20 mode symmetry ROM is run in the uncontrolled regime, to verify that it can reproduce well the dynamics of the FOM. Figure A.6 shows the time history of the modal amplitudes $x_{M,i}$ compared to the projection of the POD modes onto the snapshots $(\mathbf{q}_{CFD}, \boldsymbol{\phi}_i)$ for $i = 1, 2$ (5.69). One can see good agreement between the ROM coefficients and the projection of the snapshots onto the POD modes for all times considered. Figure A.7 shows a comparison of the snapshots with the ROM solution at the time of the 100^{th} snapshot. Again, good agreement is observed.

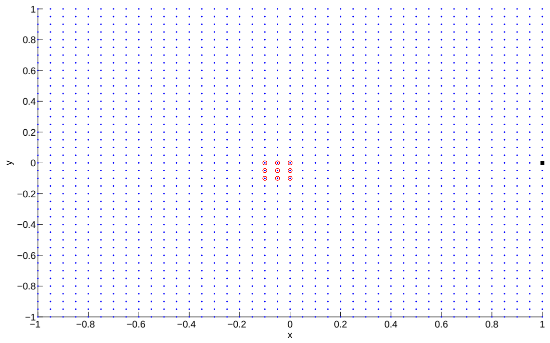


Figure A.5. Domain with forced/actuated nodes (red) and response node (black) (driven inviscid pulse problem)

Having verified the uncontrolled ROM's ability to reproduce the snapshots from which it is constructed, the sought-after LQR ROM-based controller is calculated using the `lqr` function in MATLAB's control toolbox [2]. Figure A.8(a) shows the optimal controller u_{opt} obtained by the LQR algorithm. As expected, the optimal controller is precisely the function $u_{opt}(t) = -f_{u_2}(\mathbf{x}, t)$ to machine precision. Figure A.8 (b) shows $p'(0, 1; t)$ for the uncontrolled FOM, compared with this

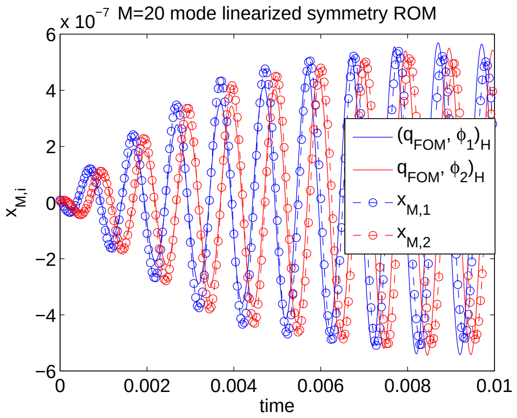
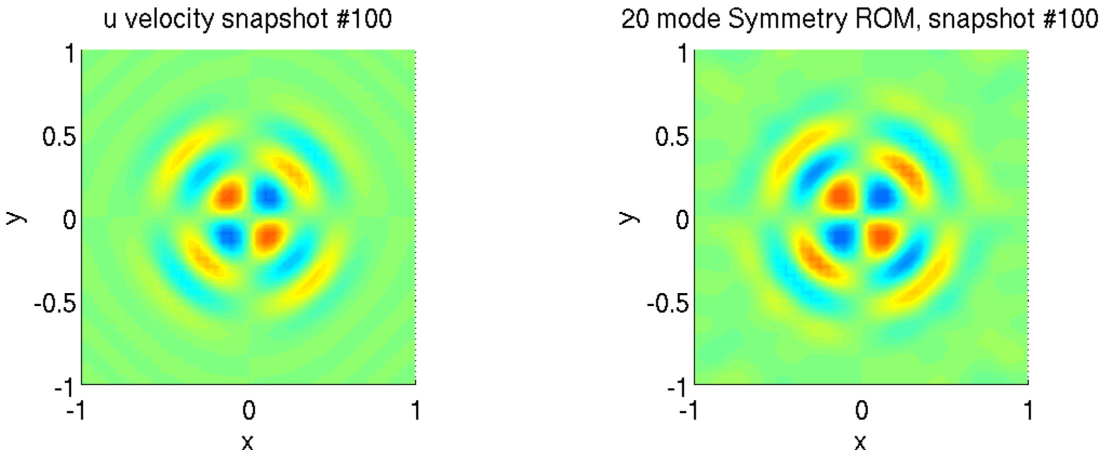


Figure A.6. Time history of ROM coefficients a_1 and a_2 (driven inviscid pulse problem, uncontrolled)

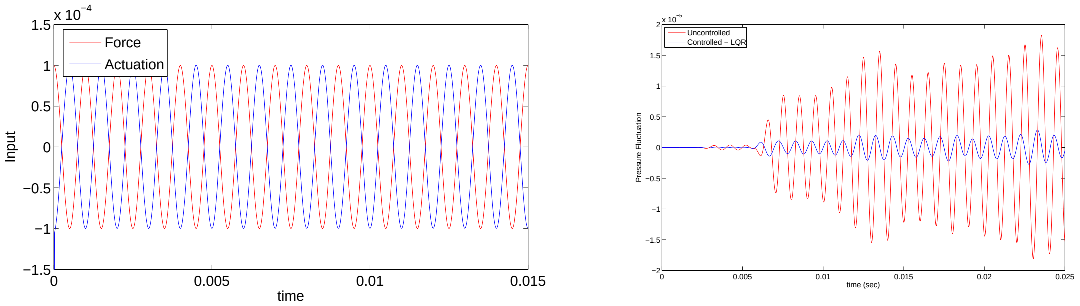


(a) FOM

(b) 20 Mode Symmetry ROM

Figure A.7. Pressure fluctuation contours at time of 100^{th} snapshot (driven inviscid pulse problem, uncontrolled)

value when the controller is applied. The reader can observe that the controller effectively wipes out the pressure fluctuation at the point where it is minimized.



(a) Force (A.63) (red) and LQR controller (blue)

(b) $p'(0,1;t)$ for uncontrolled (red) and controlled (blue) FOM

Figure A.8. ROM-computed LQR controller solution (driven inviscid pulse problem)

References

[1] The Mathworks, Inc. Optimization Toolbox User's Guide, 1990–2008.

[2] The Mathworks, Inc. Control Systems Toolbox User's Guide, 1992–1998.

[3] Oberwolfach benchmark collection. Oberwolfach benchmark collection, `http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark`, 2005.

[4] S. Abarbanel and D. Gottlieb. Optimal time splitting for two- and three-dimensional Navier-Stokes equations with mixed derivatives. *J. Comput. Phys.*, 41:1–33, 1981.

[5] I. Akhtar, A.H. Nayfeh, and C.J. Ribbens. On the stability and extension of reduced-order Galerkin models in incompressible flows: a numerical study of vortex shedding. *Theoret. Comput. Fluid Dyn.*, 23(3):213–237, 2009.

[6] D. Amsallem and C. Farhat. Interpolation method for the adaptation of reduced-order models to parameter changed and its application to aeroelasticity. *AIAA J.*, 46:1803–1813, 2008.

[7] D. Amsallem and C. Farhat. Stabilization of projection-based reduced order models. *Int. J. Numer. Meth. Engng.*, 91(4):358–377, 2012.

[8] D. Amsallem and C. Farhat. On the stability of projection-based linear reduced-order models: descriptor vs. non-descriptor forms. *Springer MS&A Series (accepted for publication)*, 8, 2013.

[9] D. Amsallem, M. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Numer. Meth. Engng.*, 92:891–916, 2012.

[10] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2001.

[11] K.J. Astrom and R.M. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton University Press, 2008.

[12] N. Aubry, P. Holmes, J. Lumley, and E. Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Psychology Today and Tomorrow*, 192:115–173, 1988.

[13] S. Bagheri, L. Brandt, and D.S. Henningson. Input-output analysis, model reduction and control of the flat-plate boundary layer. *J. Fluid Mech.*, 620:263–298, 2009.

[14] M. Balajewicz. Personal communication, May 2014.

[15] M.J. Balajewicz, E.H. Dowell, and B.R. Noack. Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation. *J. Fluid Mech.*, 729:285–308, 2013.

[16] A. Barbagallo, D. Sipp, and P.J. Schmid. Convergence results for pseudospectral approximations of hyperbolic systems by a penalty-type boundary treatment. *Math. Comput.*, 196(57):585–596, 1991.

[17] A. Barbagallo, D. Sipp, and P.J. Schmid. Closed-loop control of an open cavity flow using reduced-order models. *J. Fluid Mech.*, 641:1–50, 2009.

[18] M.F. Barone. Survey of model reduction methods for compressible fluid problems. Sandia National Laboratories Report (in preparation), Sandia National Laboratories, Albuquerque, NM, 2014.

[19] M.F. Barone, I. Kalashnikova, M.R. Brake, and D.J. Segalman. Reduced order modeling of fluid/structure interaction. Sandia National Laboratories Report, SAND No. 2009-7189, Sandia National Laboratories, Albuquerque, NM, 2009.

[20] M.F. Barone, I. Kalashnikova, D.J. Segalman, and H. Thornquist. Stable Galerkin reduced order models for linearized compressible flow. *J. Comput. Phys.*, 288:1932–1946, 2009.

[21] P. Benner, M. Castillo, E.S. Quintana-Orti, and G. Quintana-Orti. Parallel model reduction of large-scale unstable systems. *Advances in Parallel Computing*, 13:251–258, 2004.

[22] M. Bergmann, C.-H. Bruneau, and A. Iollo. Enablers for robust POD models. *J. Comput. Phys.*, 228(2):516–538, 2009.

[23] R.H. Bishop. *The mechatronics handbook.* CRS Press LLC, 2002.

[24] B.N. Bond and L. Daniel. Guaranteed stable projection-based model reduction for indefinite and unstable linear systems. *"Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design"*, 2008.

[25] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model constrained optimization for reduction of large-scale systems. *J. Comput. Phys.*, 224:880–896, 2007.

[26] K. Carlberg. Adaptive *h*–refinement for reduced-order models. *arXiv e-Print*, 2014.

[27] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.*, 24(2):623–647, 2013.

[28] Y. Chahlaoui and P. Van Dooren. Benchmark examples for model reduction of linear time invariant systems. `http://www.icm.tu-bs.de/NICONET/benchmodred.html`, 2013.

[29] S. Chaturantabut and D.C. Sorensen. Nonlinear model reduction via discrete empirical inter-

polation. *SIAM J. Sci. Comput.*, 32:2737–2764, 2010.

[30] G. Chen. *Stability of Nonlinear Systems*. Encyclopedia of RF and Microwave Engineering, Wiley, NY, 2004.

[31] S.S. Collis. A computational investigation of receptivity in high-speed flow near a swept leading edge. Ph.D. Thesis, Stanford University, Stanford, CA, 1997.

[32] M. Couplet, C. Basdevant, and P. Sagaut. Calibrated reduced-order POD-Galerkin systems for fluid modelling. *J. Comput. Phys.*, 207:192–220, 2005.

[33] R. Everson and L. Sirovich. Karhunen-Loeve procedure for gappy data. *J. Optical Society of America A*, pages 1657–1664, 1995.

[34] C. Farhat, P. Geuzaine, and G. Brown. Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers & Fluids*, 32:3–29, 2003.

[35] J.A. Fike. Construction of reduced order models for the non-linear Navier-Stokes equations using the proper orthogonal decomposition (POD)/Galerkin method. Sandia National Laboratories Report, SAND No. 2013-7051, Sandia National Laboratories, Albuquerque, NM, 2013.

[36] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Nonlinear model reduction for uncertainty quantification in large-scale inverse problems. *Int. J. Num. Meth. Engng.*, 81(12):1581–1608, 2010.

[37] F. Genin and S. Menon. Dynamics of sonic jet injection into supersonic crossflow. *J. Turbulence*, 11(4):1–30, 2010.

[38] F. Genin and S. Menon. Studies of shock/turbulent shear layer interaction using large-eddy simulation. *Computers and Fluids*, 39:800–819, 2010.

[39] X. Gloerfelt. Compressible proper orthogonal decomposition/Galerkin reduced-order model of self-sustained oscillations in a cavity. *Phys. Fluids*, 20:115105, 2008.

[40] D. Gottlieb and S.A. Orszag. *Numerical Analysis of Spectral Methods*. SIAM, 1977.

[41] T.H. Gronwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Ann. of Math.*, 20(2):292–296, 1919.

[42] S. Gugercin and A.C. Antoulas. A survey of model reduction by balanced truncation and some new results. *Int. J. Control*, 77(8):748–766, 2004.

[43] S. Gugercin and J.-R. Li. Smith-type methods for balanced truncations of large spare systems. *Lecture notes in computational science and engineering*, 45:49–82, 2005.

[44] M.D. Gunzburger. On the stability of Galerkin methods for initial-boundary value problems

for hyperbolic systems. *Math. Comp.*, 31(139):661–675, 1977.

[45] B. Gustafsson. *High order difference methods for time dependent PDE.* Springer-Verlag, 2008.

[46] B. Gustafsson, H.-O. Kreiss, and J. Oliger. *Time Dependent Problems and Difference Methods.* Wiley-Interscience, 1995.

[47] B. Gustafsson and A. Sundstrom. Incompletely parabolic problems in fluid dynamics. *SIAM J. Appl. Math.*, 35(2):343–357, 1978.

[48] W.M. Haddad and S.G. Nersesov. *Stability and control of large-scale dynamical systems: A Vector Dissipative Systems Approach.* Princeton University Press, 2011.

[49] M.A. Heroux, R.A. Bartlett, V.E. Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, R.B. Lehoucq, K.R. Long, R.P. Pawlowski, E.T. Phipps, A.G. Salinger, H.K. Thornquist, R.S. Tuminaro, J.M. Willenbring, A. Williams, and K.S. Stanley. An overview of the Trilinos project. *ACM Trans. Math. Softw.*, 31(3), 2005.

[50] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge University Press, 1996.

[51] L.N. Cattafesta III, Q. Song, D.R. Williams, C.W. Rowley, and F.S. Alvi. Active control of flow-induced cavity oscillations. *Prog. Aerosp. Sci.*, 44:479–502, 2008.

[52] M. Ilak. Model reduction and feedback control of transitional channel flow. Ph.D. Thesis, Princeton University, Princeton, NJ, 2009.

[53] S.J. Illingworth, A.S. Morgans, and C.W. Rowley. Feedback control of flow resonances using balanced reduced-order models. *J. Sound and Vibration*, 330(8):1567–1581, 2011.

[54] W. Weaver Jr., S.P. Timoshenko, and D.H. Young. *Vibration problems in engineering.* Wiley, 5th Ed., 1990.

[55] I. Kalashnikova and S. Arunajatesan. A stable Galerkin reduced order model for compressible flow. WCCM-2012-19407, 10th World Congress on Computational Mechanics (WCCM), Sao Paulo, Brazil, 2012.

[56] I. Kalashnikova and M. Barone. Efficient non-linear proper orthogonal decomposition (POD)/Galerkin reduced order models with stable penalty enforcement of boundary conditions. *Int. J. Numer. Meth. Engng.*, 90(11):1337–1362, 2012.

[57] I. Kalashnikova and M.F. Barone. On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment. *J. Comput. Phys.Int. J. Numer. Meth. Engng.*, 83:1345–1375, 2010.

[58] I. Kalashnikova and M.F. Barone. Stable and efficient Galerkin reduced order models for non-linear fluid flow. AIAA-2011-3110, 6th AIAA Theoretical Fluid Mechanics Conference,

Honolulu, HI, 2011.

[59] I. Kalashnikova, M.F. Barone, S. Arunajatesan, and B.G. van Bloemen Waanders. Construction of energy-stable Galerkin reduced order models. Sandia National Laboratories Report, SAND No. 2013-4063, Sandia National Laboratories, Albuquerque, NM, 2013.

[60] I. Kalashnikova, M.F. Barone, S. Arunajatesan, and B.G. van Bloemen Waanders. Construction of energy-stable projection-based reduced order models. *Appl. Math. Computat. (under review)*, 2014.

[61] I. Kalashnikova, B.G. van Bloemen Waanders, S. Arunajatesan, and M.F. Barone. Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment. *Comp. Meth. Appl. Mech. Engng.*, 272:251–270, 2014.

[62] H. Kimura. *Chain-scattering approach to H-infinity control*. Sprinter, 1997.

[63] B. Kirk, J.W. Peterson, R.H. Stogner, and G.F. Carey. `libMesh`: A C++ library for parallel adaptive mesh refinement/coarsening simulations. *Eng. Comput.*, 22(3–4):237–254, 2006.

[64] H.O. Kreiss and J. Lorenz. *Initial-Boundary Value Problems and the Navier-Stokes Equations*. Academic Press, Inc., 1989.

[65] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition for a general equation in fluid dynamics. *SIAM J. Num. Anal.*, 40(2):492–515, 2002.

[66] F. Kwasniok. Empirical low-order models of barotropic flow. *J. Atmos. Sci.*, 61(2):235–245, 2004.

[67] W.J. Layton. Stable Galerkin methods for hyperbolic systems. *SIAM. J. Numer. Anal.*, 20(3):221–233, 1983.

[68] J. Lienemann, E.B. Rudnyi, and J.G. Korvink. MST MEMS model order reduction: Requirements and benchmarks. *Linear Algebra Appl.*, 415(2–3):469–498, 2006.

[69] T. Lieu, C. Farhat, and M. Lesoinne. Reduced-order fluid/structure modeling of a complete aircraft configuration. *Comp. Methods Appl. Mech. Engng.*, 195:5730–5742, 2006.

[70] E. Lombardi, M. Bergmann, S. Camarri, and A. Iollo. Low-order models: optimal sampling and linearized control strategies. INRIA Technical Report, No. 7092, 2009.

[71] J.L. Lumley. *Stochastic tools in turbulence*. Academic Press: New York, 1971.

[72] A. Mani. Analysis and optimization of numericla sponge layers as nonreflective boundary treatment. *J. Comput. Phys.*, 231(2):704–716, 2012.

[73] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1), 1981.

[74] N.C. Nguyen, A.T. Patera, and J. Peraire. A ‘best points’ interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Meth. Engng.*, 73:521–543, 2008.

[75] N.C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *Int. J. Numer. Meth. Engng.*, 76:27–55, 2008.

[76] B.R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models of the transient and post-transient cylinder wake. *J. Fluid Mech.*, 497:335–363, 2003.

[77] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[78] M. Rathinam and L.R. Petzold. A new look at proper orthogonal decomposition. *SIAM J. Num. Anal.*, 41(5):1893–1925, 2003.

[79] C.W. Rowley. Model reduction for fluids using balanced proper orthogonal decomposition. *Int. J. Bif. Chaos*, 15(3):997–1013, 2005.

[80] C.W. Rowley, T. Colonius, and A.J. Basu. On self-sustained oscillations in two-dimensional compressible flow over rectangular cavities. *J. Fluid Mech.*, 455:315–346, 2002.

[81] C.W. Rowley, T. Colonius, and R.M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D*, 189:115–129, 2004.

[82] C.W. Rowley, I. Mezic, S. Baheri, P. Schlatter, and D.S. Henningson. Reduced-order models for flow control: balanced models and Koopman modes. *Seventh IUTAM Symposium on Laminar-Turbulent Transition*, 2009.

[83] G. Rozza. Reduced basis approximation and error bounds for potential flows in parametrized geometries. *Commun. Comput. Phys.*, 9(1):1–48, 2011.

[84] V. Sankaran and S. Menon. LES of scalar mixing in supersonic shear layers. *Proceedings of the Combustion Institute*, 30(2):2835–2842, 2004.

[85] G. Serre, P. Lafon, X. Gloerfelt, and C. Bailly. Reliable reduced-order models for time-dependent linearized Euler equations. *J. Comput. Phys.*, 231(15):5176–5194, 2012.

[86] S. Sirisup and G.E. Karniadakis. A spectral viscosity method for correcting the long-term behavior of POD models. *J. Comput. Phys.*, 194:92–116, 2004.

[87] S. Sirisup, D. Xiu, G. Karniadakis, and X. Kevrekidis. Equation-free/Galerkin-free POD-assisted computation of incompressible flows. *J. Comput. Phys.*, 207:568–587, 2005.

[88] L. Sirovich. Turbulence and the dynamics of coherent structures, part iii: dynamics and scaling. *Q. Appl. Math.*, 45(3):583–590, 1987.

[89] W. Sutherland. The viscosity of gases and molecular force. *Philosophical Magazine, S. 5*, 36:507–531, 1893.

[90] E. Tadmor. Spectral methods for hyperbolic problems. Lecture Notes Delivered at Ecole des Ondes, *"Méthodes numériques d'order élevé pour les ondes en régime transitoire"*, 1994.

[91] C.K.W. Tam and J.C. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *J. Comput. Phys.*, 107:262–281, 1993.

[92] F. Terragni, E. Valero, and J.M. Vega. Local POD plus galerkin projection in the unsteady lid-driven cavity problem. *SIAM. J. Sci. Comput.*, 33(6):3538–3561, 2011.

[93] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-bases *a posteriori* error bounds. *J. Num. Meth. Fluids*, 47:773–788, 2005.

[94] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Comput. Methods Appl. Mech. Engng.*, pages 237–240, 2012.

[95] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.

[96] K. Zhou. *Robust and optimal control*. Prentice Hall, 1996.

DISTRIBUTION:

1	MS	0825	Srinivasan Arunajatesan, 01515
1	MS	0825	Matthew F. Barone, 01515
1	MS	0825	Jeffrey L. Payne, 01515
1	MS	1070	Jeffrey A. Fike, 01526
1	MS	1070	Matthew R. Brake, 01526
1	MS	1319	Andrew G. Salinger, 01442
1	MS	1319	Bart G. van Bloemen Waanders, 01441
1	MS	1320	Michael L. Parks, 01442
1	MS	1320	S. Scott Collis, 01440
1	MS	9159	Martin Drohmann, 08954
1	MS	9159	Kevin T. Carlberg, 08954
1	MS	0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories