

Finite Horizon Optimality and Operator Splitting in Model Reduction of Large-Scale Dynamical Systems

Klajdi Sinani

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mathematics

Serkan Gugercin, Chair

Christopher A. Beattie

Jeffrey T. Borggaard

Mark Embree

June 18, 2020

Blacksburg, Virginia

Keywords: Model Reduction, Dynamical Systems, IRKA, Unstable Systems, Finite
Horizon, $\mathcal{H}_2(t_f)$ Optimal, Operator Splitting, POD

Copyright 2020, Klajdi Sinani

Finite Horizon Optimality and Operator Splitting in Model Reduction of Large-Scale Dynamical Systems

Klajdi Sinani

(ABSTRACT)

Simulation, design, and control of dynamical systems play an important role in numerous scientific and industrial tasks. The need for detailed models leads to large-scale dynamical systems, posing tremendous computational difficulties when employed in numerical simulations. In order to overcome these challenges, we perform model reduction, replacing the large-scale dynamics with high-fidelity reduced representations. There exist a plethora of methods for reduced order modeling of linear systems, including the Iterative Rational Krylov Algorithm (IRKA), Balanced Truncation (BT), and Hankel Norm Approximation. However, these methods generally target stable systems and the approximation is performed over an infinite time horizon. If we are interested in a finite horizon reduced model, we utilize techniques such as Time-limited Balanced Truncation (TLBT) and Proper Orthogonal Decomposition (POD). In this dissertation we establish interpolation-based optimality conditions over a finite horizon and develop an algorithm, Finite Horizon IRKA (FHIRKA), that produces a locally optimal reduced model on a specified time-interval. Nonetheless, the quantities being interpolated and the interpolant are not the same as in the infinite horizon case. Numerical experiments comparing FHIRKA to other algorithms further support our theoretical results. Next, we discuss model reduction for nonlinear dynamical systems. For models with unstructured nonlinearities, POD is the method of choice. However, POD is input dependent and not optimal with respect to the output. Thus, we use operator splitting to integrate the best features of system theoretic approaches with trajectory based methods such as POD in order to mitigate the effect of the control inputs for the approximation of

nonlinear dynamical systems. We reduce the linear terms with system theoretic methods and the nonlinear terms via POD. Evolving the linear and nonlinear terms separately yields the reduced operator splitting solution. We present an error analysis for this method, as well as numerical results that illustrate the effectiveness of our approach. While in this dissertation we only pursue the splitting of linear and nonlinear terms, this approach can be implemented with Quadratic Bilinear IRKA or Balanced Truncation for Quadratic Bilinear systems to further diminish the input dependence of the reduced order modeling.

Finite Horizon Optimality and Operator Splitting in Model Reduction of Large-Scale Dynamical Systems

Klajdi Sinani

(GENERAL AUDIENCE ABSTRACT)

Simulation, design, and control of dynamical systems play an important role in numerous scientific and industrial tasks such as signal propagation in the nervous system, heat dissipation, electrical circuits and semiconductor devices, synthesis of interconnects, prediction of major weather events, spread of fires, fluid dynamics, machine learning, and many other applications. The need for detailed models leads to large-scale dynamical systems, posing tremendous computational difficulties when applied in numerical simulations. In order to overcome these challenges, we perform model reduction, replacing the large-scale dynamics with high-fidelity reduced representations. Reduced order modeling helps us to avoid the outstanding burden on computational resources. Numerous model reduction techniques exist for linear models over an infinite horizon. However, in practice we usually are interested in reducing a model over a specific time interval. In this dissertation, given a reduced order, we present a method that finds the best local approximation of a dynamical system over a finite horizon. We present both theoretical and numerical evidence that supports the proposed method. We also develop an algorithm that integrates operator splitting with model reduction to solve nonlinear models more efficiently while preserving a high level of accuracy.

Dedication

To Jesus Christ

“Whatever you do, do it all for the glory of God.” 1 Corinthians 10:31.

To my lovely and beautiful wife, Elise

To my wonderful parents, Kudret and Naxhie

Acknowledgments

First and foremost I want to thank my advisor, Dr. Serkan Gugercin, without whom this dissertation would not have been possible. His extensive knowledge, diligence, and patience have been crucial to the success of this research. In addition to being extremely helpful from an academic perspective, Dr. Gugercin's passion and love for model reduction have been a source of inspiration and encouragement throughout my graduate studies. I would also like to thank the members of my committee, Dr. Christopher Beattie, Dr. Jeff Borggaard, and Dr. Mark Embree for all their help and willingness to talk to me anytime I asked. Their advice, suggestions, and insights during seminars, talks, and conversations have been invaluable. I want to thank all of my fellow grad students for making my graduate school experience at Virginia Tech very enjoyable and I cherish all of the memories we have made together. Special thanks for their friendships to Hrayr Aprahamian, Mehdi Bouhafara, and Haroun Meghaichi.

Contents

List of Figures	x
List of Tables	xii
1 Introduction	1
1.1 Introduction	1
2 Model Reduction of Linear Dynamical Systems	7
2.1 Model Reduction of Linear Dynamical Systems	7
2.2 Projection Based Model Reduction	10
2.3 Error Measures	11
2.4 Interpolatory Model Reduction	17
2.4.1 $\mathcal{H}_2$ Optimal Interpolation Methods	20
2.5 Balanced Truncation	23
2.6 Model Reduction for Unstable Systems	29
2.6.1 Optimal $\mathcal{L}_2$ Model Reduction	30
2.6.2 Balanced Truncation for Unstable Systems	34
3 Finite Horizon Model Reduction	37
3.1 Reduced Order Modeling on a Finite Horizon	37
3.2 Error Measures on a Finite Horizon	39

3.3	Time-limited Balanced Truncation	44
3.4	Gramian based $\mathcal{H}_2(t_f)$ optimality conditions	46
3.5	$\mathcal{H}_2(t_f)$ Optimal Model Reduction: MIMO Case	50
3.5.1	Implication of the interpolatory $\mathcal{H}_2(t_f)$ optimality conditions	61
4	Algorithmic Developments for $\mathcal{H}_2(t_f)$ Model Reduction	64
4.1	$\mathcal{H}_2(t_f)$ Optimality Conditions: SISO Case	64
4.2	A Descent-type Algorithm for the SISO Case	75
4.3	Numerical Comparisons	76
4.4	Matrix Exponential Approximation	88
4.5	Summary of Finite Horizon MOR	91
5	Operator Splitting with Model Reduction	94
5.1	Nonlinear Model Reduction	95
5.1.1	Quadratic Bilinear Systems	95
5.1.2	Proper Orthogonal Decomposition	97
5.1.3	Discrete Empirical Interpolation Method	100
5.2	Linear Operator Splitting	101
5.3	Nonlinear Operator Splitting	103
5.4	Operator Splitting and MOR for General Nonlinearities	104
5.5	Error Analysis for IPS	109
5.6	Numerical Results	117

5.6.1	Nonlinear RC Ladder	118
5.6.2	Chafee-Infante Model	126
5.6.3	Warning: Tubular Reactor	132
6	Conclusions and Outlook	135
	Bibliography	137

List of Figures

4.1	FHIRKA and other algorithms for the heat model	79
4.2	FHIRKA and other algorithms for the ISS model	80
4.3	FHIRKA and other algorithms for the unstable model	81
4.4	FHIRKA and POD for the ISS model	82
4.5	FHIRKA and POD for the heat model	82
4.6	FHIRKA and POD for the unstable model	83
4.7	Output Plots for Convection Diffusion Model	92
4.8	Error Plots for Convection Diffusion Model	92
5.1	Operator Splitting: Step 1	105
5.2	Operator Splitting: Step 2	105
5.3	Operator Splitting: Step 3	106
5.4	RC Ladder Circuit [177]	118
5.5	Jacobian of the Nonlinearity	119
5.6	Output Error: IPS vs POD	120
5.7	State Error: IPS vs POD	121
5.8	ROM Error on the Linear Terms	121
5.9	ROM Error on the Nonlinear Terms	122
5.10	IPS vs Backward Euler; $r = 8$, $h = 0.0025$	123

5.11 IPS vs Backward Euler; $h = 0.01$	123
5.12 IPS Errors; $h = 0.01$	124
5.13 IPS vs Backward Euler; $r = 6$	124
5.14 IPS Errors; $r = 6$	125
5.15 Error vs r ; $h = 0.01$	125
5.16 Error vs h ; $r = 6$	126
5.17 Jacobian of the Nonlinearity	127
5.18 IPS vs Backward Euler; $r = 16$, $h = 0.0001$	128
5.19 IPS vs Backward Euler; $h = 0.0001$	129
5.20 IPS Errors; $h = 0.0001$	129
5.21 IPS vs Backward Euler; $r = 16$	130
5.22 IPS Errors; $r = 16$	130
5.23 Error vs r ; $h = 0.0001$	131
5.24 Error vs h ; $r = 16$	131
5.25 Operator Splitting vs Backward Euler	133
5.26 Condition Number of the Jacobian of the Nonlinearity	134

List of Tables

4.1	$\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for a heat model ($n = 197$)	84
4.2	$\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for an ISS model ($n = 270$)	84
4.3	$\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for an unstable system ($n = 402$)	85
4.4	$\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for an unstable system ($n = 4002$) . . .	85
4.5	$\mathcal{H}_2(t_f)$ errors for TLBT and FHIRKA for a heat model ($n = 197$)	86
4.6	$\mathcal{H}_2(t_f)$ errors for TLBT and FHIRKA for an ISS model ($n = 270$)	86
4.7	$\mathcal{H}_2(t_f)$ errors for IRKA and FHIRKA for a heat model ($n = 197$)	87
4.8	$\mathcal{H}_2(t_f)$ errors for IRKA and FHIRKA for an ISS model ($n = 270$)	87
4.9	$\mathcal{H}_2(t_f)$ errors for IRKA and FHIRKA for an unstable model ($n = 402$)	88
4.10	Matrix Exponential Computation	91

Chapter 1

Introduction

1.1 Introduction

The ever-increasing demand for greater resolution when simulating, designing, and controlling dynamical systems in numerous scientific applications such as signal propagation in the nervous system [115, 183], heat dissipation [44], electrical circuits and semiconductor devices, [96], synthesis of interconnects [41], prediction of major weather events [8], spread of fires [128], fluid dynamics [39, 182], large scale inverse problems [55, 59, 144], machine learning [151, 174, 175] etc. and the ensuing large-scale mathematical models present enormous computational difficulties when applied in numerical simulations. In order to overcome these challenges we use *reduced order modeling* (ROM).

For instance, consider a stable linear dynamical system that could result from the discretization of a linear PDE:

$$\begin{aligned}\mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{1.1.1}$$

where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times n}$ are constant matrices. The variable $\mathbf{x}(t) \in \mathbb{R}^n$ denotes an internal variable, $\mathbf{u}(t) \in \mathbb{R}^m$ denotes the control inputs, and $\mathbf{y}(t) \in \mathbb{R}^p$ denotes the outputs. For a large scale system n could be huge, i.e., n could take values in the hundreds of thousands, maybe even millions. Even for linear systems, working with such large orders still constitutes a big challenge. For this reason, we aim to replace the original model with

a lower dimension model:

$$\begin{aligned}\mathbf{E}_r \dot{\mathbf{x}}_r(t) &= \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{B}_r \mathbf{u}(t), \\ \mathbf{y}_r(t) &= \mathbf{C}_r \mathbf{x}_r(t),\end{aligned}\tag{1.1.2}$$

where $\mathbf{A}_r, \mathbf{E}_r \in \mathbb{R}^{r \times r}$, $\mathbf{B}_r \in \mathbb{R}^{r \times m}$, and $\mathbf{C}_r \in \mathbb{R}^{p \times r}$ with $r \ll n$. Our goal is to approximate the true outputs of the original dynamical system with the outputs of the reduced system for the same input in an appropriate norm, i.e., $\mathbf{y}_r(t) \approx \mathbf{y}(t)$ for a wide range of inputs.

Numerous model reduction techniques exist for linear dynamical systems as in (1.1.1). Methods such as Hankel Norm Approximation [77, 78, 110, 188], Balanced Truncation (BT) [139, 140] and its various extensions such as Frequency Weighted Balanced Truncation, Balanced Truncation for Quadratic Bilinear Systems, and Time-limited Balanced Truncation (TLBT) depend on the system gramians; see [8, 33, 88, 126]. An alternative to gramian based methods is interpolatory model reduction [13, 14, 66, 149, 152, 178]. Methods like the *Iterative Rational Krylov Algorithm* (IRKA) [9, 11, 23, 25, 43, 67, 86], and its variants TFIRKA [20, 24], Bilinear IRKA (BIRKA) [27, 29, 68], and Quadratic Bilinear IRKA [2, 3, 34], yield accurate reduced order models under appropriate norms. For more details on interpolatory model order reduction see the recent book [11]. In this thesis, we focus on nonparametric systems; for model reduction of parametric dynamical systems we refer the reader to [19, 36, 95, 97, 142, 143, 150].

For general nonlinear systems Proper Orthogonal Decomposition (POD) [52, 108, 121, 131, 156] is the method of choice. However, POD is input dependent. For techniques that help us make better choices for the training input we refer the reader to [101].

In addition to the system based intrusive frameworks, there exist many data driven approaches including Loewner [12, 81, 135], Vector Fitting [56, 58], AAA [141], and Dynamic Mode Decomposition (DMD) [124, 125, 163].

For linear model reduction we focus primarily on IRKA and BT. IRKA is a fixed point iteration that takes advantage of the pole residue expansion of the transfer function of

the dynamical system at hand. With Balanced Truncation, as the name suggests, first we balance the observability and reachability gramians, and then perform truncation in the balanced basis. Upon convergence IRKA guarantees a locally optimal reduced model for single input/single output (SISO) systems, and BT provides an upper bound for the $\mathcal{H}_\infty$ error between the full and the reduced model.

Often, using techniques like balanced truncation and IRKA, we approximate the large system with a smaller system over an infinite time horizon. However, in industrial and scientific applications we never run simulations over an infinite time horizon. Usually, we are interested only in the behavior of the dynamical system over a finite time interval. Techniques like time-limited balanced truncation [75, 87, 126, 153, 154] and Proper Orthogonal Decomposition(POD) [102, 108] yield high-fidelity reduced models on a finite horizon.

One of the major contributions of this thesis is the derivation of an $\mathcal{H}_2(t_f)$ optimality condition for model reduction over a finite time horizon. Before discussing our new results we investigate some related work. Melchior et al. [137] propose a method that constructs an optimal reduced model by minimizing the Frobenius norm of the error for linear time-varying systems on a finite horizon. Gugercin et al. [86] work within a frequency based framework to establish the $\mathcal{H}_2$ optimality conditions for a full horizon, while we take advantage of the time-domain representation of the dynamical system to derive the conditions for a finite time. Following a similar approach to Wilson [181], Goyal and Redmann [82] exploit the Lyapunov equations associated with the dynamical system in a finite horizon and the relationship between the $\mathcal{H}_2(t_f)$ norm and the finite-time Gramians to obtain the $\mathcal{H}_2(t_f)$ optimality conditions and propose an IRKA-type algorithm with the aid of the system gramians that solve a set of Sylvester equations. However, this IRKA type algorithm only produces a nearly optimal reduced order model. We follow a different approach by

representing the impulse response of the reduced dynamical system as follows:

$$\mathbf{h}_r(t) = \sum_{k=1}^r \phi_k e^{\lambda_k t}, \quad (1.1.3)$$

and treat ϕ_k 's and λ_k 's as the optimization parameters. In addition to establishing $\mathcal{H}_2(t_f)$ optimality conditions over a finite time horizon, we also introduce an algorithm which yields better approximations of the large-scale dynamical systems compared to other model reduction methods. Establishing $\mathcal{H}_2(t_f)$ optimality conditions for model reduction over a finite horizon enables us to reduce unstable systems optimally under the $\mathcal{H}_2(t_f)$ measure. Computational experiments on a heat model, an International Space Station (ISS) model, and an unstable model, further support our results.

Another major contribution of this dissertation is the development of an operator splitting algorithm that solves unstructured reduced nonlinear systems accurately. Consider the general nonlinear dynamical system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t)). \end{aligned} \quad (1.1.4)$$

If the nonlinearity has the form

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \sum_{k=1}^m \mathbf{N}_k \mathbf{x}(t) \mathbf{u}_k(t) + \mathbf{B}\mathbf{u}(t),$$

then we have a bilinear system, and this nonlinear model can be approximated by BIRKA [27, 29] or BT for bilinear systems [31, 62] without input dependence. If

$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{H} \cdot \mathbf{x}(t) \otimes \mathbf{x}(t) + \sum_{k=1}^m \mathbf{N}_k \mathbf{x}(t) \mathbf{u}_k(t) + \mathbf{B}\mathbf{u}(t),$$

i.e., if we have a quadratic bilinear system, the system in (1.1.4) can be approximated via

QB-IRKA [34] or BT for quadratic bilinear systems [33]. However, as mentioned earlier, for nonlinear systems without any structure POD remains the method of choice. For POD we exploit the singular value decomposition of a snapshot matrix obtained by simulating the dynamical system. Since POD relies on a specific trajectory, it is input dependent; therefore, different inputs yield different reduced order models. To mitigate the input dependence of POD, we integrate the best features of a data-driven method such as POD, and a system theoretic method like IRKA or Balanced Truncation using operator splitting. Operator splitting is a very effective technique that is used to split terms in ODEs and PDEs based on the criteria of interest [132]. For example, we may split terms with different physics, terms with different spatial characteristics, or, as in our case, linear terms from nonlinear ones. We split the linear and nonlinear parts because we want to perform model reduction separately. Initially, we reduce the order of the linear terms via IRKA or BT, and the nonlinear terms via POD. Once we have obtained the reduced order terms we evolve the linear and nonlinear parts separately over each subinterval.

Structure of the Dissertation

- Chapter 2 reviews linear model reduction. In the first part we discuss reduced order modeling for linear asymptotically stable systems using system theoretic methods like IRKA and Balanced Truncation. Then, we describe a few existing approaches for model order reduction of unstable systems like $\mathcal{L}_2$ -IRKA [134] and Balanced Truncation for unstable systems [189].
- Chapter 3 discusses our contributions for finite horizon model order reduction. First, we review existing techniques like time-limited balanced truncation and a gramian based approach. Then, we establish the optimality conditions for the multi input/multi output case.

- Chapter 4 delineates the computational framework arising from the established finite horizon optimality conditions. We construct a descent algorithm, perform numerical experiments to test the newly constructed algorithm, and present the results.
- Chapter 5 presents another major contribution of this dissertation. We start by reviewing nonlinear reduced order modeling, specifically methods like QB-IRKA, POD, and DEIM. Then, we construct an algorithm that combines IRKA and POD via operator splitting. We perform an error analysis on this algorithm and test it numerically.
- Chapter 6 concludes the dissertation with a summary of our work and the outlook for these research topics.

Chapter 2

Model Reduction of Linear Dynamical Systems

In this chapter we review existing model reduction approaches for linear dynamical systems, specifically, the Iterative Rational Krylov Algorithm (IRKA) and Balanced Truncation (BT). We also discuss error measures and define concepts like asymptotically stable systems, i.e., systems whose poles lie on the open left plane. Lastly, we cover extensions of IRKA and BT for systems with poles on the right half-plane.

2.1 Model Reduction of Linear Dynamical Systems

Consider the linear dynamical system:

$$\begin{aligned}\mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \quad \text{with } \mathbf{x}(0) = \mathbf{0},\end{aligned}\tag{2.1.1}$$

where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times n}$ are constant matrices. In this system $\mathbf{x}(t) \in \mathbb{R}^n$ is the internal variable, also called the state variable if the matrix $\mathbf{E}$ is invertible. The dimension of the system is n , the input is $\mathbf{u}(t) \in \mathbb{R}^m$, and the output is $\mathbf{y}(t) \in \mathbb{R}^p$. If $m = p = 1$, then the dynamical system is called single-input/single-output (SISO). If $m > 1$ and $p > 1$, the system is called multi-input/multi-output (MIMO).

Definition 2.1. A dynamical system is called asymptotically stable if $Re(\lambda_k) < 0$ for $k = 1, \dots, n$, where the λ_k 's denote the poles of the dynamical system and $Re(\cdot)$ denotes the real part of a complex number.

Definition 2.2. A dynamical system is called stable if $Re(\lambda_k) \leq 0$ for $k = 1, \dots, n$, where the λ_k 's denote the poles of the dynamical system and $Re(\cdot)$ denotes the real part of a complex number, provided that the poles on the imaginary axis are not defective, i.e., the geometric multiplicity is the same as the algebraic multiplicity.

When n is very large, e.g., $n > 10^6$, the simulations for design, control, and other applications are very costly, computationally speaking. The purpose of model reduction is to replace the original model with a lower dimension model that has the form

$$\begin{aligned} \mathbf{E}_r \dot{\mathbf{x}}_r(t) &= \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{B}_r \mathbf{u}(t), \\ \mathbf{y}_r(t) &= \mathbf{C}_r \mathbf{x}_r(t), \quad \text{with } \mathbf{x}_r(0) = \mathbf{0}, \end{aligned} \tag{2.1.2}$$

where $\mathbf{A}_r, \mathbf{E}_r \in \mathbb{R}^{r \times r}$, $\mathbf{B}_r \in \mathbb{R}^{r \times m}$, $\mathbf{C}_r \in \mathbb{R}^{p \times r}$, and $\mathbf{D}_r \in \mathbb{R}^{p \times m}$ with $r \ll n$, and such that the outputs of the reduced system are good approximations of the corresponding true outputs over a wide range of inputs, i.e, $\mathbf{y}_r(t) \approx \mathbf{y}(t)$. Later in the chapter we clarify how we measure closeness.

For the sake of clarity and simplicity of presentation, we assume $\mathbf{E} = \mathbf{I}$, and have the following dynamical system:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t). \end{aligned} \tag{2.1.3}$$

We aim to produce a reduced order model of the form

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{B}_r \mathbf{u}(t) \\ \mathbf{y}_r(t) &= \mathbf{C}_r \mathbf{x}_r(t).\end{aligned}\tag{2.1.4}$$

Nonetheless, all of the methods discussed in this dissertation extend to the case where $\mathbf{E} \neq \mathbf{I}$, but nonsingular. If $\mathbf{E} \neq \mathbf{I}$, we can multiply both sides of the first equation in (2.1.1) by $\mathbf{E}^{-1}$ to retrieve an equivalent dynamical system of the same form as in (2.1.3) since $\mathbf{E}$ is assumed to be nonsingular. However, in practice, we do not multiply by $\mathbf{E}^{-1}$ to simplify the system, because that might destroy the sparsity of $\mathbf{A}$. Similarly to (2.1.3), we have $\mathbf{E}_r = \mathbf{I}_r$ for the reduced model (2.1.4).

Remark 2.3. In this dissertation we focus on the case where $\mathbf{E}$ is nonsingular, i.e, we work with ordinary differential equations (ODEs). If $\mathbf{E}$ is singular, the model is a system of differential algebraic equations (DAEs). For extensions of the methods like Balanced Truncation (BT) and the Iterative Rational Krylov Algorithm (IRKA) to DAEs, see [89, 172].

As is the case with any approximation, we need to evaluate how good the approximation is. Thus, we need to define the error measures that are necessary for the quantification of the model reduction error. For model order reduction of asymptotically stable linear systems we mainly use the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms. For the definition of these norms we transform (2.1.3) and (2.1.4) to the frequency domain. We give further details on these error measures in Section 2.3. To obtain the frequency domain representation of the full order dynamical systems we can compute either Laplace or Fourier transforms of the quantities of interest. There are robust methods that work with these frequency domain measures such as the Iterative Rational Krylov Algorithm (IRKA) [21, 86], Balanced Truncation [8, 139, 140], Hankel Norm Approximation [77] etc., which we can use to approximate these large scale systems by a reduced order model. The main methods that we examine in this thesis are Balanced Truncation (BT) and the Iterative Rational Krylov Algorithm (IRKA).

2.2 Projection Based Model Reduction

In order to achieve model reduction, most methods use projection, more specifically, some variant of a Petrov-Galerkin or Galerkin projection. The original state is approximated by $\mathbf{x}(t) \approx \mathbf{V}\mathbf{x}_r(t)$, where $\mathbf{x}_r(t) \in \mathbb{R}^r$, and $\mathbf{V} \in \mathbb{R}^{n \times r}$ is a basis for an r -dimensional subspace. Thus, we plug in the approximation $\mathbf{V}\mathbf{x}_r(t)$ into (2.1.3) and obtain:

$$\mathbf{V}\dot{\mathbf{x}}_r(t) - \mathbf{R}(\mathbf{x}_r(t)) = \mathbf{A}\mathbf{V}\mathbf{x}_r(t) + \mathbf{B}\mathbf{u}(t), \quad (2.2.1)$$

where $\mathbf{R}(\mathbf{x}_r(t))$ is a residual. We rewrite (2.2.1) as

$$\mathbf{R}(\mathbf{x}_r(t)) = \mathbf{V}\dot{\mathbf{x}}_r(t) - \mathbf{A}\mathbf{V}\mathbf{x}_r(t) - \mathbf{B}\mathbf{u}(t). \quad (2.2.2)$$

In order to determine the trajectory of the reduced internal variable $\mathbf{x}_r$, we enforce a Petrov-Galerkin projection by multiplying both sides of (2.2.2) with $\mathbf{W}^T$ to obtain

$$\mathbf{W}^T\mathbf{R}(\mathbf{x}_r(t)) = \mathbf{W}^T\mathbf{V}\dot{\mathbf{x}}_r(t) - \mathbf{W}^T\mathbf{A}\mathbf{V}\mathbf{x}_r(t) - \mathbf{W}^T\mathbf{B}\mathbf{u}(t) = \mathbf{0} \quad (2.2.3)$$

If we obtain the bases $\mathbf{W}, \mathbf{V}$ such that $\mathbf{W}$ is bi-orthogonal to $\mathbf{V}$, i.e., $\mathbf{W}^T\mathbf{V} = \mathbf{I}_r$ then we have

$$\dot{\mathbf{x}}_r(t) = \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{B}_r\mathbf{u}(t),$$

$$\mathbf{y}_r(t) = \mathbf{C}_r\mathbf{x}_r(t)$$

such that

$$\begin{aligned}\mathbf{A}_r &= \mathbf{W}^T \mathbf{A} \mathbf{V}, \\ \mathbf{B}_r &= \mathbf{W}^T \mathbf{B}, \text{ and} \\ \mathbf{C}_r &= \mathbf{C} \mathbf{V},\end{aligned}\tag{2.2.4}$$

where $\mathbf{A}_r \in \mathbb{R}^{r \times r}$, $\mathbf{B}_r \in \mathbb{R}^{r \times m}$, and $\mathbf{C}_r \in \mathbb{R}^{p \times r}$.

2.3 Error Measures

When we approximate a large scale dynamical system by a reduced order model, we need to compute the approximation error. Thus, we need appropriate error measures. The error analysis for linear dynamical systems will be conducted in the frequency domain initially, since the difference between the outputs $\mathbf{Y}(s)$ and $\mathbf{Y}_r(s)$ is directly linked to the difference between the full and reduced transfer functions. Let $\mathbf{Y}(s)$, $\mathbf{Y}_r(s)$, and $\mathbf{U}(s)$ be the Laplace transforms of $\mathbf{y}(t)$, $\mathbf{y}_r(t)$ and $\mathbf{u}(t)$. After taking the Laplace transforms of the original model (2.1.1) and the reduced model (2.1.2) we obtain

$$\mathbf{Y}(s) = (\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B})\mathbf{U}(s),$$

and

$$\mathbf{Y}_r(s) = (\mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r)\mathbf{U}(s).$$

The mappings $\mathbf{U} \mapsto \mathbf{Y}$ and $\mathbf{U} \mapsto \mathbf{Y}_r$ are the *transfer functions* associated with the full and reduced model respectively, and they are denoted by:

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B},$$

and

$$\mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r,$$

where both $\mathbf{H}(s)$ and $\mathbf{H}_r(s)$ are $p \times m$ matrix valued rational functions.

We have

$$\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{U}(s), \quad \text{and} \quad \mathbf{Y}_r(s) = \mathbf{H}_r(s)\mathbf{U}(s). \quad (2.3.1)$$

In the frequency domain we have

$$\mathbf{Y}(s) - \mathbf{Y}_r(s) = (\mathbf{H}(s) - \mathbf{H}_r(s))\mathbf{U}(s).$$

Equivalently, the output $\mathbf{y}(t)$ in the time domain is given by

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) = \int_0^t \mathbf{h}(t - \tau)\mathbf{u}(\tau)d\tau, \quad (2.3.2)$$

where

$$\mathbf{h}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B}, \quad t > 0 \quad (2.3.3)$$

is the impulse response of the full model. Note that the transfer function $\mathbf{H}(s)$ is the Laplace transform of the impulse response $\mathbf{h}(t)$.

The error analysis in the time and frequency domains are equivalent. Thus, we measure how close $\mathbf{H}_r(s)$ is to $\mathbf{H}(s)$ using the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms.

Definition 2.4. Suppose $\mathbf{G}(s)$ and $\mathbf{H}(s)$ are transfer functions corresponding to linear stable dynamical systems with the same input and output dimensions. The $\mathcal{H}_2$ inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_2}$ in the frequency domain is defined as

$$\langle \mathbf{G}(s), \mathbf{H}(s) \rangle_{\mathcal{H}_2} := \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}(\overline{\mathbf{G}}(-i\omega)\mathbf{H}(i\omega)^T) d\omega,$$

and as a result, the $\mathcal{H}_2$ norm is given by

$$\|\mathbf{H}(s)\|_{\mathcal{H}_2} := \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{H}(i\omega)\|_F^2 d\omega \right)^{1/2},$$

where $\|\cdot\|_F$ represents the Frobenius norm. Recall that the impulse response $\mathbf{h}(t)$ is the inverse Laplace transform of the transfer function $\mathbf{H}(s)$. In the time domain we have

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2}^2 = \int_0^{\infty} \|\mathbf{h}\|_F^2 dt.$$

We revisit the error analysis in the time domain with more details in Chapter 3. Computation of the $\mathcal{H}_2$ norm relies on the reachability and observability gramians of the system (2.1.3). Let us define the concepts of reachability and observability first.

Definition 2.5. [8] Let $\psi(\mathbf{u}; \mathbf{x}_0; t)$ be the solution to the state equations $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ with $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathbb{X}$ be the subspace of all the states of the system. A state $\bar{\mathbf{x}}$ is *reachable* from the zero state if there exists an input function $\bar{\mathbf{u}}$, of finite energy, and a time $\bar{t} < \infty$ such that

$$\bar{\mathbf{x}} = \psi(\bar{\mathbf{u}}; \mathbf{0}; \bar{t}).$$

We say that the system is completely reachable if $\mathbb{X} = \mathbb{R}^n$. A state $\bar{\mathbf{x}} \in \mathbb{X}$ is *unobservable* if $\mathbf{y}(t) = \mathbf{C}\psi(\mathbf{0}; \bar{\mathbf{x}}; t) = \mathbf{0}$ for all $t \geq 0$. Let $\mathbb{X}^{un}$ be the unobservable subspace of $\mathbb{X}$ for the dynamical system (2.1.3). Then, (2.1.3) is fully *observable* if $\mathbb{X}^{un} = \mathbf{0}$.

Recall the impulse response of the system (2.1.3) is given by (2.3.3). Define

$$\mathbf{h}_{re}(t) = e^{\mathbf{A}t}\mathbf{B}, \quad t > 0 \tag{2.3.4}$$

to be the input-to-state response, and

$$\mathbf{h}_{ob}(t) = \mathbf{C}e^{\mathbf{A}t}, \quad t > 0 \quad (2.3.5)$$

to be the state-to-output response [8]. The response definitions in (2.3.3), (2.3.4), and (2.3.5) enable us to define the reachability and observability gramians.

Definition 2.6. Suppose the dynamical system (2.1.3) is asymptotically stable. Then, the reachability gramian of the system is defined as

$$\mathcal{P} := \int_0^\infty \mathbf{h}_{re}(t)\mathbf{h}_{re}^T(t)dt = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t}dt, \quad (2.3.6)$$

and its observability gramian is defined as

$$\mathcal{Q} := \int_0^\infty \mathbf{h}_{ob}^T(t)\mathbf{h}_{ob}(t)dt = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t}dt. \quad (2.3.7)$$

Note that the output matrix $\mathbf{C}$ is irrelevant in terms of reachability, while the input matrix $\mathbf{B}$ plays no role in terms of observability. Therefore, instead of describing reachability and observability in terms of the dynamical system, we may depict these concepts as the reachability of the pair $(\mathbf{A}, \mathbf{B})$, and the observability of the pair $(\mathbf{C}, \mathbf{A})$.

We can compute these gramians by solving the following Lyapunov equations for $\mathcal{P}$ and $\mathcal{Q}$:

$$\mathbf{A}\mathcal{P} + \mathcal{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = 0 \text{ and } \mathbf{A}^T\mathcal{Q} + \mathcal{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} = 0. \quad (2.3.8)$$

If the dynamical system is asymptotically stable the solutions of (2.3.8), i.e., the matrices $\mathcal{P}$, $\mathcal{Q}$ are unique and symmetric positive semi-definite. Furthermore, if the pair $(\mathbf{A}, \mathbf{B})$ is reachable, and the pair $(\mathbf{C}, \mathbf{A})$ is observable, then $\mathcal{P}$ and $\mathcal{Q}$ are positive definite matrices. For more details, we refer the reader to [8]. Once we have the solutions of the Lyapunov

equations (2.3.8), using the definition of the $\mathcal{H}_2$ norm of the system (2.1.1), it follows:

$$\|\mathbf{H}\|_{\mathcal{H}_2} = \sqrt{\text{tr}(\mathbf{C}\mathcal{P}\mathbf{C}^T)} = \sqrt{\text{tr}(\mathbf{B}^T\mathcal{Q}\mathbf{B})}. \quad (2.3.9)$$

Alternatively, if the matrix $\mathbf{A}$ is diagonalizable, [8] provides a method based on the poles and residues of the transfer function to calculate the $\mathcal{H}_2$ norm:

$$\|\mathbf{H}(s)\|_{\mathcal{H}_2} = \sum_{k=1}^n = \text{tr}(\text{res}[\mathbf{H}(-s)\mathbf{H}^T(s), \lambda_k]), \quad (2.3.10)$$

where λ_k denotes the eigenvalues of $\mathbf{A}$ and

$$\text{res}[\mathbf{H}(-s)\mathbf{H}^T(s), \lambda_k] = \lim_{s \rightarrow \lambda_k} \mathbf{H}(-s)\mathbf{H}^T(s)(s - \lambda_k).$$

In order to further explore the aforementioned equivalence between the error analysis in the time and frequency domains, we define the L_2 norm in the time domain.

Definition 2.7. Let $\mathbf{f}(t)$ and $\mathbf{g}(t)$ be vector valued real functions defined on the interval $\mathcal{I} = [0, \infty)$. Then, the L_2 inner product is defined as

$$\langle \mathbf{f}(t), \mathbf{g}(t) \rangle = \left(\int_0^\infty \mathbf{f}(t)^T \mathbf{g}(t) dt \right)^{\frac{1}{2}}, \quad (2.3.11)$$

and

$$\|\mathbf{f}(t)\|_{L_2} = \left(\int_0^\infty \|\mathbf{f}(t)\|_2^2 dt \right)^{\frac{1}{2}}. \quad (2.3.12)$$

The $\mathcal{H}_2$ norm relates to the L_∞ norm of the output $\mathbf{y}(t)$ in the time domain as

$$\|\mathbf{y}\|_{L_\infty} = \sup_{t>0} \|\mathbf{y}(t)\|_\infty \leq \|\mathbf{H}\|_{\mathcal{H}_2} \|\mathbf{u}\|_{L_2}.$$

In model reduction, we aim to minimize the error between the full and the reduced problem,

so we are interested in the following:

$$\|\mathbf{y} - \mathbf{y}_r\|_{L_\infty} \leq \|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_2} \|\mathbf{u}\|_{L_2}.$$

In future sections, we discuss how these definitions of the $\mathcal{H}_2$ norm and the aforementioned methods for its computation facilitate establishing $\mathcal{H}_2$ optimality conditions. As mentioned above, in addition to the $\mathcal{H}_2$ norm, we use the $\mathcal{H}_\infty$ norm to measure the error. The $\mathcal{H}_\infty$ norm is defined as

$$\|\mathbf{H}\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \|\mathbf{H}(i\omega)\|_2$$

where $\|\cdot\|_2$ denotes the 2-norm of a matrix. The $\mathcal{H}_\infty$ norm of a dynamical system is directly related to the L_2 induced operator norm of the operator which maps $\mathbf{u}$ into $\mathbf{y}$. We have

$$\|\mathbf{H}\|_{\mathcal{H}_\infty} = \sup_{\mathbf{u} \in L_2} \|\mathbf{y}\|_{L_2} = \sup_{\mathbf{u} \in L_2} \left(\int_0^\infty \|\mathbf{y}(t)\|_2^2 dt \right)^{1/2} \text{ for all } \mathbf{u} \text{ such that } \|\mathbf{u}\|_{L_2} = 1.$$

Hence, for the model reduction problem we have

$$\|\mathbf{y} - \mathbf{y}_r\|_{L_2} \leq \|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_\infty} \|\mathbf{u}\|_{L_2}.$$

Therefore, if we want to minimize the output error in the L_∞ norm, an $\mathcal{H}_2$ -based model reduction technique should be used. On the other hand, if we aim to have the minimal output error in the L_2 norm, we should use an $\mathcal{H}_\infty$ -based model reduction technique. Depending on the norm of interest, we may choose which technique to use. If we want an upper bound for the $\mathcal{H}_\infty$ error, we use balanced truncation. If we want to find a local minimum for the $\mathcal{H}_2$ error we use IRKA.

It is important to note that we are bounding the L_2 and L_∞ error measures, which are defined on the time domain, respectively by the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms, which are defined on

the frequency domain. For more details on the equivalence between the time and frequency domain error measures, we refer the reader to the recent book [11].

2.4 Interpolatory Model Reduction

When we reduce the order of a dynamical system we are essentially approximating the full order system with a reduced order model. Recall that

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

is a rational function of degree n , which we want to approximate via *interpolation* with a rational function of degree r , i.e.,

$$\mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r.$$

Enforcing interpolation conditions for a scalar-valued transfer function is straightforward. In the single-input/single-output (SISO) case, where

$$\mathbf{H}(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$$

is a scalar rational function of degree n and

$$\mathbf{H}_r(s) = \mathbf{c}_r^T(s\mathbf{I}_r - \mathbf{A}_r)^{-1}\mathbf{b}_r$$

is a scalar rational function of degree r , the interpolation is straightforward. In order to interpolate, we need a set of interpolation points $\{\sigma_i\}_i^r \subset \mathbb{C}$ and construct $\mathbf{H}_r$ such that

$$\mathbf{H}(\sigma_k) = \mathbf{H}_r(\sigma_k) \text{ for } i = 1, 2, 3, \dots, r. \quad (2.4.1)$$

We construct $\mathbf{H}_r$ by projection as shown in section 2.2.

However, if $\mathbf{H}$ is a transfer function corresponding to a multi-input/multi-output (MIMO) system, then $\mathbf{H}$ is a $p \times m$ matrix-valued function. In this case interpolation becomes slightly more complicated. One option is to enforce the interpolation conditions pointwise; however, this would require a large reduced order r since for each interpolation point, $p \times m$ interpolation conditions would be necessary. Even reasonable input and output dimensions would lead to a large number of interpolation conditions and, as a result, defeat the purpose of model reduction. Tangential interpolation is an alternative approach. This means that the matrix-valued approximant $\mathbf{H}_r$ interpolates the original transfer function $\mathbf{H}$ along certain tangential directions. Thus, we need to select left and right tangential directions as well as left and right interpolation points. $\mathbf{H}_r$ is a right-tangential interpolant to $\mathbf{H}$ if

$$\mathbf{H}(\sigma_k)\mathbf{r}_k = \mathbf{H}_r(\sigma_k)\mathbf{r}_k, \quad (2.4.2)$$

where $\sigma_k \in \mathbb{C}$ is a right interpolation point and $\mathbf{r}_k \in \mathbb{C}^m$ is a right tangential direction. Analogously, $\mathbf{H}_r$ is a left-tangential interpolant to $\mathbf{H}$ if

$$\mathbf{l}_k^T \mathbf{H}(\mu_k) = \mathbf{l}_k^T \mathbf{H}_r(\mu_k) \quad (2.4.3)$$

where $\mu_k \in \mathbb{C}$ is a left interpolation point and $\mathbf{l}_k \in \mathbb{C}^p$ is a left tangential direction. Note that the interpolation points σ_k and μ_k cannot be poles of either the full system or the reduced model. Next we address the question of how to obtain the projection bases $\mathbf{V}$ and $\mathbf{W}$.

For the MIMO case, given the transfer function $\mathbf{H}$, right interpolation points $\{\sigma_i\}_{i=1}^r \subset \mathbb{C}$, left interpolation points $\{\mu_i\}_{i=1}^r \subset \mathbb{C}$, right directions $\{\mathbf{r}_i\}_{i=1}^r \subset \mathbb{C}^m$, left directions $\{\mathbf{l}_i\}_{i=1}^r \subset \mathbb{C}^p$, we construct $\mathbf{W} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$ in the following manner:

$$\begin{aligned} \mathbf{V} &= [(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_1 \ \cdots \ (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_r], \\ \mathbf{W}^T &= \begin{bmatrix} \mathbf{l}_1^T \mathbf{C} (\mu_1 \mathbf{I} - \mathbf{A})^{-1} \\ \vdots \\ \mathbf{l}_r^T \mathbf{C} (\mu_r \mathbf{I} - \mathbf{A})^{-1} \end{bmatrix}. \end{aligned} \quad (2.4.4)$$

Computing the projection bases $\mathbf{V}$ and $\mathbf{W}$ as in (2.4.4) and changing these bases so that $\mathbf{W}^T \mathbf{V} = \mathbf{I}$, enables us to project down the state space matrices $\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{C}_r = \mathbf{C} \mathbf{V}$, and $\mathbf{B}_r = \mathbf{W}^T \mathbf{B}$ and satisfy the Lagrange tangential interpolation conditions in (2.4.2) and (2.4.3). Furthermore, if (2.4.4) holds, and $\sigma_k = \mu_k$ for all k , we have tangential Hermite interpolation, i.e.,

$$\mathbf{l}_k^T \mathbf{H}'(\mu_k) \mathbf{r}_k = \mathbf{l}_k^T \mathbf{H}'_r(\mu_k) \mathbf{r}_k. \quad (2.4.5)$$

For the SISO case, obviously, there is no need for tangential directions or left/right categorization of the interpolation points. Given a set of interpolation points $\{\sigma_i\}_{i=1}^r \subset \mathbb{C}$ we construct the model reduction bases similarly to the MIMO case where the tangential directions are equal to one:

$$\begin{aligned} \mathbf{V} &= [(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{b} \ \cdots \ (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{b}] \\ \mathbf{W}^T &= \begin{bmatrix} \mathbf{c}^T (\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \\ \vdots \\ \mathbf{c}^T (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \end{bmatrix}, \end{aligned} \quad (2.4.6)$$

and $\mathbf{W}^T \mathbf{V} = \mathbf{I}$. Obtaining the reduced matrices $\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{c}_r = \mathbf{c} \mathbf{V}$, and $\mathbf{b}_r = \mathbf{W}^T \mathbf{b}$, we compute a reduced order model $\mathbf{H}_r$ that satisfies the Lagrange and Hermite interpolation

conditions, i.e.,

$$\begin{aligned}\mathbf{H}(\sigma_k) &= \mathbf{H}_r(\sigma_k) \text{ and} \\ \mathbf{H}'(\sigma_k) &= \mathbf{H}'_r(\sigma_k)\end{aligned}\tag{2.4.7}$$

for $i = 1, 2, 3, \dots, r$. Interpolatory model reduction and its applications have been addressed also in [13, 14, 66, 72, 73, 83, 149, 152, 179, 186, 187]. For a comprehensive review of interpolatory model reduction we refer the reader to the recent book [11]. Since projection based interpolatory model reduction is, after all, an approximation via interpolation, the quality of such an approximation depends on the choice of the interpolation points. Our goal is to construct an optimal reduced model with respect to some norm. We are particularly interested in optimality in the $\mathcal{H}_2$ norm. In other words, if we have a full-order dynamical system $\mathbf{H}(s)$, we want to construct a reduced-order model $\mathbf{H}_r(s)$ such that

$$\|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_2} \leq \left\| \mathbf{H} - \hat{\mathbf{H}}_r \right\|_{\mathcal{H}_2},$$

where $\hat{\mathbf{H}}_r$ is any dynamical system of dimension r . $\mathcal{H}_2$ optimality informs our choice of interpolation points and helps us avoid ad hoc selections. We describe how to achieve local optimality next in Section 2.4.1.

2.4.1 $\mathcal{H}_2$ Optimal Interpolation Methods

The previous section showed how to construct a reduced model that satisfies the interpolation conditions given a set of initial shifts. However, we have no information whether the obtained reduced-order model is optimal. In this section we will describe how to obtain optimality, at least locally, in the $\mathcal{H}_2$ norm. Model reduction with respect to the $\mathcal{H}_2$ norm has been studied extensively; see, for example, [7, 16, 43, 45, 47, 49, 70, 86, 91, 106, 111, 130, 136, 145, 168, 178, 180, 181, 184] and the references therein.

Recall the optimization problem we are considering: If $\mathbf{H}(s)$ is the transfer function for a

large dynamical system, find a new reduced model $\mathbf{H}_r(s)$ which minimizes the $\mathcal{H}_2$ error. In other words, find $\mathbf{H}_r$ such that

$$\|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_2} = \min_{\dim(\hat{\mathbf{H}}_r)=r} \|\mathbf{H} - \hat{\mathbf{H}}_r\|_{\mathcal{H}_2}.$$

Assuming $\mathbf{A}_r$ is diagonalizable, we write the pole-residue expansion of $\mathbf{H}_r(s)$ as

$$\mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r = \sum_{k=1}^r \frac{\mathbf{l}_k \mathbf{r}_k^T}{s - \lambda_k},$$

where λ_k are the poles of $\mathbf{H}_r$, while $\mathbf{l}_k$ and $\mathbf{r}_k$ are residue directions, and $\mathbf{l}_k \mathbf{r}_k^T$ are rank-1 residues. Taking advantage of the pole-residue expansion [86] establishes necessary local optimality conditions, as stated in Theorem 2.8.

Theorem 2.8. [86] *Let $\mathbf{H}_r(s)$ be the best r^{th} order rational approximation of a stable linear model $\mathbf{H}$ with respect to the $\mathcal{H}_2$ norm. Then*

$$\begin{aligned} \mathbf{l}_k^T \mathbf{H}(-\lambda_k) &= \mathbf{l}_k^T \mathbf{H}_r(-\lambda_k), \\ \mathbf{H}(-\lambda_k) \mathbf{r}_k &= \mathbf{H}_r(-\lambda_k) \mathbf{r}_k, \text{ and} \\ \mathbf{l}_k^T \mathbf{H}'(-\lambda_k) \mathbf{r}_k &= \mathbf{l}_k^T \mathbf{H}_r'(-\lambda_k) \mathbf{r}_k, \end{aligned}$$

for $k = 1, 2, \dots, r$, where λ_k denotes the poles of the reduced system and $\mathbf{l}_k, \mathbf{r}_k$ are the residue directions.

For the SISO case, the pole residue expansion of the reduced transfer function is

$$\mathbf{H}_r(s) = \sum_{k=1}^r \frac{\phi_k}{s - \lambda_k}$$

where λ_k are the poles of the reduced system and ϕ_k are the corresponding residues. This pole-residue expansion can be computed easily through the eigenvalue decomposition for

$s\mathbf{I}_r - \mathbf{A}_r$. The eigenvalue decomposition is relatively cheap, since the size of the reduced system is relatively small. The next corollary follows directly from Theorem 2.8, however it first appeared in [136].

Corollary 2.9. [136] *Let $\mathbf{H}_r(s)$ be the best r^{th} order rational approximation of $\mathbf{H}(s)$ with respect to the $\mathcal{H}_2$ norm. Then*

$$\mathbf{H}(-\lambda_k) = \mathbf{H}_r(-\lambda_k), \text{ and}$$

$$\mathbf{H}'(-\lambda_k) = \mathbf{H}'_r(-\lambda_k)$$

for $k = 1, 2, \dots, r$ where the λ_k 's denote the poles of the reduced system.

Thus, in order to satisfy the first-order conditions for optimality in the $\mathcal{H}_2$ norm, we need to interpolate at the mirror images of the poles of the reduced model [86, 136]. However, since we do not have any knowledge of the reduced system poles, the model reduction algorithm must be iterative. Building upon Theorem 2.8 and Corollary 2.9, the Iterative Rational Krylov Algorithm (IRKA) was developed [86]. IRKA produces a reduced model that satisfies the first-order optimality conditions in the $\mathcal{H}_2$ norm. Indeed, in the SISO case, it is guaranteed to yield at least a locally optimal reduced order model [67]. The optimality conditions in Theorem 2.8 and Corollary 2.9 can equivalently be derived using the norm expressions in (2.3.9) via differentiation with respect to the state space matrices [27, 181].

By picking a set of initial interpolation points, we can compute $\mathbf{V}$ and $\mathbf{W}$ as described in (2.4.4) or (2.4.6), and obtain a reduced order model. Then we compute the pole residue expansion of the reduced model. After computing the pole residue expansion, we use the mirror images of the poles of the reduced system as interpolation points and repeat the process to obtain another reduced model. We continue until the convergence condition is satisfied. Algorithm 1 presents the pseudocode for IRKA for SISO systems. For more details about IRKA, see [86]. While IRKA it is a great tool for reducing a linear asymptotically

stable system, it is quite limited when we deal with unstable systems, i.e., systems that have poles in the right half plane. We can see from the definition of the $\mathcal{H}_2$ norm that the error is unbounded. Of course we can attempt to reduce an unstable system via IRKA as in [165], but we have no guarantees of convergence or accuracy, even though the reduced model captures some of the unstable poles of the full order system. Note that in some projection based model reduction, unstable poles can appear even if the original system is asymptotically stable and we refer to the recent paper [64] for some details on this aspect. In Section 2.6.1 we discuss an $\mathcal{L}_2$ approach for unstable systems [134].

2.5 Balanced Truncation

So far we have discussed interpolatory projection based model order reduction and methods like IRKA. Balanced Truncation is another effective method for model order reduction of linear asymptotically stable systems [8, 88, 139, 140]. Unlike IRKA, which produces a locally optimal reduced order model in the $\mathcal{H}_2$ norm, balanced truncation is not optimal in any norm, but it allows us to bound the model reduction error with respect to the $\mathcal{H}_\infty$ norm. As we have observed so far, reducing the order of a dynamical system requires the elimination of some of the state variables. Since balanced truncation generates a reduced model that is potentially different from IRKA, it makes decisions about which states to eliminate in a different manner. We explore balanced truncation in this section. Write the dynamical system (2.1.3) as

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

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times n}$. Approximation by balanced truncation is achieved by eliminating the states that are hard to reach and hard to observe. We say that a state is hard to reach if the minimal *energy* required to transition a system from the zero

Algorithm 1 IRKA Pseudocode

Input: Original state space matrices and initial shift selection

Output: Reduced state space matrices

- Pick an r -fold initial shift set selection that is closed under conjugation
 - $\mathbf{V} = [(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{b} \dots (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{b}]$
 - $\mathbf{W} = [(\sigma_1 \mathbf{I} - \mathbf{A})^{-T} \mathbf{c}^T \dots (\sigma_r \mathbf{I} - \mathbf{A})^{-T} \mathbf{c}^T]$
 - Change the bases so that $\mathbf{W}^T \mathbf{V} = \mathbf{I}_r$.
 - while (not converged)
 - $\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{b}_r = \mathbf{W}^T \mathbf{b}$, and $\mathbf{c}_r = \mathbf{c} \mathbf{V}$
 - Compute a pole-residue expansion of $\mathbf{H}_r(s)$:

$$\mathbf{H}_r(s) = \mathbf{c}_r^T (s \mathbf{I}_r - \mathbf{A}_r)^{-1} \mathbf{b}_r = \sum_{i=1}^r \frac{\phi_i}{s - \lambda_i}$$
 - $\sigma_i \leftarrow -\lambda_i$, for $i = 1, \dots, r$
 - $\mathbf{V} = [(\sigma_1 \mathbf{I} - \mathbf{A})^{-1} \mathbf{b} \dots (\sigma_r \mathbf{I} - \mathbf{A})^{-1} \mathbf{b}]$
 - $\mathbf{W} = [(\sigma_1 \mathbf{I} - \mathbf{A})^{-T} \mathbf{c}^T \dots (\sigma_r \mathbf{I} - \mathbf{A})^{-T} \mathbf{c}^T]$
 - Change the bases so that $\mathbf{W}^T \mathbf{V} = \mathbf{I}_r$.
 - $\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{b}_r = \mathbf{W}^T \mathbf{b}$, and $\mathbf{c}_r = \mathbf{c} \mathbf{V}$
-

state at $t = -\infty$ to state $\mathbf{x}_{re}$ at time $t = 0$ is high. This energy is quantified by the norm of the control input. A state is difficult to observe if it yields low energy when we observe the output of the state $\mathbf{x}_{ob}$ with no input. The observation energy is quantified by the norm of the output. Furthermore, assuming the system is reachable and asymptotically stable, the minimal energy required to reach a state $\mathbf{x}_{re}$ is given by

$$E_{re} = \mathbf{x}_{re}^T \mathcal{P}^{-1} \mathbf{x}_{re}, \quad (2.5.1)$$

and the maximal observation energy yielded by the state $\mathbf{x}_{ob}$ is

$$E_{ob} = \mathbf{x}_{ob}^T \mathcal{Q} \mathbf{x}_{ob}. \quad (2.5.2)$$

Thus, in order to classify which states are hard to reach and to observe we use the reachability and observability gramians defined in (2.3.6) and (2.3.7). It follows from (2.5.1) and (2.5.2) that the states which are hard to reach are in the span of the eigenvectors of $\mathcal{P}$ corresponding to small eigenvalues, and the states which are hard to observe are in the span of the eigenvectors of $\mathcal{Q}$ corresponding to small eigenvalues. For more details on the concepts of reachability and observability, we refer the reader to [8].

If $\mathbf{A}$ is asymptotically stable, the solutions $\mathcal{P}, \mathcal{Q}$ to the Lyapunov equations are unique symmetric positive semi-definite matrices. One popular and effective method for solving the Lyapunov equations (2.3.8) is the Bartels-Stewart algorithm [18, 92]. However, the Bartels-Stewart algorithm computes a Schur decomposition, hence the number of arithmetic operations is $\mathcal{O}(n^3)$ and the storage required is $\mathcal{O}(n^2)$. Thus, balanced truncation is too expensive for large-scale systems if we solve the Lyapunov equations (2.3.8) with the Bartels-Stewart algorithm. We can reduce the cost of balanced truncation if we solve the Lyapunov equations using less expensive, yet accurate, techniques such as Alternating Direction Implicit (ADI) [37, 118, 127] and Krylov methods [38, 113, 164, 173]. Since balanced truncation requires

the solution of Lyapunov equations, which is quite taxing, the cost of the approximation is closely connected to the method that we use to solve these equations.

As we have already mentioned, balanced truncation eliminates the states which are hard to reach and hard to observe. If the states which are difficult to reach are easy to observe or vice-versa, we need to find a basis where the states that are hard to reach, are also hard to observe. Since the quantity of interest is the output, we consider transforming the state variable $\mathbf{x}$. In other words, we find a linear state transformation $\mathbf{T}$, where $\mathbf{T}$ is nonsingular, such that

$$\tilde{\mathbf{x}} = \mathbf{T}\mathbf{x}. \quad (2.5.3)$$

Plugging (2.5.3) into (2.1.3) we obtain the equivalent system

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}(t) &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}\mathbf{u}(t) \\ \mathbf{y}(t) &= \tilde{\mathbf{C}}\tilde{\mathbf{x}}(t), \end{aligned} \quad (2.5.4)$$

where

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1}, \\ \tilde{\mathbf{B}} &= \mathbf{T}\mathbf{B}, \text{ and} \\ \tilde{\mathbf{C}} &= \mathbf{C}\mathbf{T}^{-1}. \end{aligned} \quad (2.5.5)$$

Given a nonsingular matrix $\mathbf{T}$, we transform the gramians as

$$\hat{\mathcal{P}} = \mathbf{T}\mathcal{P}\mathbf{T}^T \text{ and } \hat{\mathcal{Q}} = \mathbf{T}^{-T}\mathcal{Q}\mathbf{T}^{-1}.$$

Therefore, in order to guarantee the states that are hard to reach are simultaneously hard to observe we need to find an invertible state transformation $\mathbf{T}$ that yields $\hat{\mathcal{P}} = \hat{\mathcal{Q}}$ in the transformed basis. If $\hat{\mathcal{P}} = \hat{\mathcal{Q}}$, we say that the reachable, observable and stable system Σ is balanced. Σ is principal axis balanced if $\hat{\mathcal{P}} = \hat{\mathcal{Q}} = \text{diag}(\sigma_1, \dots, \sigma_n)$ for $\sigma_i = \sqrt{\lambda_i(\mathcal{P}\mathcal{Q})}$ where these λ_i 's denote the eigenvalues of $\mathcal{P}\mathcal{Q}$. The values σ_i are known as the Hankel

singular values of the system. Before computing the balancing transformation $\mathbf{T}$, we need the Cholesky factor $\mathbf{U}$ of $\mathcal{P}$ and the eigendecomposition of $\mathbf{U}^T \mathcal{Q} \mathbf{U}$:

$$\mathcal{P} = \mathbf{U} \mathbf{U}^T \text{ and } \mathbf{U}^T \mathcal{Q} \mathbf{U} = \mathbf{K} \mathbf{G}^2 \mathbf{K}^T.$$

Lemma 2.10. [8] *Given the reachable, observable and stable system*

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{0} \end{array} \right]$$

and the corresponding gramians $\mathcal{P}$ and $\mathcal{Q}$, a principal axis balancing transformation is given as follows:

$$\mathbf{T} = \mathbf{G}^{1/2} \mathbf{K}^T \mathbf{U}^{-1} \text{ and } \mathbf{T}^{-1} = \mathbf{U} \mathbf{K} \mathbf{G}^{-1/2}.$$

In the balanced basis the states which are hard to observe are also hard to reach and they correspond to small Hankel singular values. Let's assume the system

$$\left[\begin{array}{c|c} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hline \hat{\mathbf{C}} & \mathbf{0} \end{array} \right]$$

is balanced. Consider the following matrix partitions:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}, \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}, \hat{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & 0 \\ 0 & \mathbf{G}_{22} \end{bmatrix}$$

for

$$\begin{aligned}\mathbf{G} &= \text{diag}(\sigma_1, \dots, \sigma_n) = \mathcal{P} = \mathcal{Q}, \\ \mathbf{G}_1 &= (\sigma_1, \dots, \sigma_r), \text{ and} \\ \mathbf{G}_2 &= (\sigma_{r+1}, \dots, \sigma_n),\end{aligned}$$

where σ_i denotes the i -th Hankel singular values of the system for $i = 1, 2, \dots, n$. Note $\mathbf{A}_{11} \in \mathbb{R}^{r \times r}$, $\mathbf{G}_{11} \in \mathbb{R}^{r \times r}$, $\mathbf{B}_1 \in \mathbb{R}^{r \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times r}$. Then the system

$$\Sigma_r := \left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{B}_1 \\ \hline \mathbf{C}_1 & \mathbf{0} \end{array} \right] \quad (2.5.6)$$

is a reduced order model obtained by balanced truncation. Balanced truncation preserves asymptotic stability [148], and provides an upper bound for the error in the $\mathcal{H}_\infty$ norm [65]. Thus, the reduced model (2.5.6) is asymptotically stable and satisfies

$$\|\Sigma - \Sigma_r\|_{\mathcal{H}_\infty} \leq 2(\sigma_{r+1} + \dots + \sigma_n), \quad (2.5.7)$$

where $\sigma_{r+1}, \dots, \sigma_n$ are the non-repeated $n - r$ smallest Hankel singular values of the system. If a Hankel singular value is repeated, then it is counted only once. The equality is achieved if $\mathbf{G}_{22} = \sigma_n$ [8]. The balancing method described above is numerically inefficient and ill-conditioned. For this reason, when implementing balanced truncation, square-root balancing is preferred [8]. Next, we provide a description of the square-root balancing method. First, we compute the Cholesky factorizations of the gramians without computing the gramians themselves:

$$\mathcal{P} = \mathbf{U}\mathbf{U}^T \text{ and } \mathcal{Q} = \mathbf{L}\mathbf{L}^T.$$

In other words, we solve the Lyapunov equations for the Cholesky factors $\mathbf{U}$ and $\mathbf{L}$ of the

gramians, rather than the gramians. Then, we compute the singular value decomposition

$$\mathbf{U}^T \mathbf{L} = \mathbf{Y} \mathbf{\Sigma} \mathbf{Z}^T.$$

Let $\mathbf{Y}_r$ and $\mathbf{Z}_r$ be the matrices consisting of the first r columns of $\mathbf{Y}$ and $\mathbf{Z}$, respectively, and $\mathbf{\Sigma}_r$ be the diagonal matrix with the largest r singular values on the diagonal. If we define $\mathbf{V} = \mathbf{U} \mathbf{Y}_r \mathbf{\Sigma}_r^{-1/2}$ and $\mathbf{W} = \mathbf{L} \mathbf{Z}_r \mathbf{\Sigma}_r^{-1/2}$, we can compute $\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}$, $\mathbf{B}_r = \mathbf{W}^T \mathbf{B}$, and $\mathbf{C}_r = \mathbf{C} \mathbf{V}$; hence, we obtain the reduced model. Although balanced truncation produces a high fidelity, asymptotically stable reduced order model, solving Lyapunov equations can be very expensive. Thus, any improvement with respect to the efficiency of balanced truncation requires methods that solve the aforementioned Lyapunov equations fast; see [30, 37, 38, 113, 118, 127, 164, 173].

2.6 Model Reduction for Unstable Systems

While BT and IRKA are very reliable for reducing stable linear systems, this is not the case for unstable systems. Before continuing to discuss model reduction for unstable system, let us formally define unstable systems.

Definition 2.11. A dynamical system is called unstable if at least one of the following is true:

- $Re(\lambda_k) > 0$ for at least one k or where λ_k -s denote the poles of the dynamical system and $Re(\cdot)$ denotes the real part of a complex number.
- The algebraic multiplicity of the poles located on the imaginary axis is greater than their geometric multiplicity.

As we have seen already in this thesis, IRKA produces a locally optimal approximation

under the $\mathcal{H}_2$ -norm. However, if the poles of the system have a non-negative real part, the $\mathcal{H}_2$ -norm of the error system cannot be bounded. The system gramians as described in Definition 2.6 cannot be extended to dynamical systems with poles in the right half plane, since the integrals in (2.3.6) and (2.3.7) are unbounded. Nonetheless, the solutions to the Lyapunov equations (2.3.8) may still exist. Furthermore, these solutions are unique if and only if the eigenvalues of $\mathbf{A}$ do not overlap with the eigenvalues of $-\bar{\mathbf{A}}$. Therefore, we need a different framework for reducing unstable systems via balanced truncation.

2.6.1 Optimal $\mathcal{L}_2$ Model Reduction

Even though we cannot guarantee a high-fidelity reduced order model while reducing an unstable system, we can ensure there exists a bounded related norm so that we can modify IRKA to approximate unstable systems. Before describing an iteratively corrected rational Krylov algorithm, we discuss briefly $\mathcal{L}_2$ systems and the $\mathcal{L}_2$ norm. Let $\mathcal{L}_2^n(\mathbb{R})$ be the set of vector-valued functions with finite “energy” on $\mathbb{R}$:

$$\mathcal{L}_2^n(\mathbb{R}) = \{\mathbf{x}(t) \in \mathbb{R}^n : \int_{-\infty}^{\infty} \|\mathbf{x}(t)\|^2 dt < \infty\}.$$

Define $\mathcal{A} : \mathcal{L}_2^n(\mathbb{R}) \mapsto \mathcal{L}_2^n(\mathbb{R})$ as $\mathcal{A}\mathbf{x} = \dot{\mathbf{x}} - \mathbf{A}\mathbf{x}$ on all vector-valued functions $\mathbf{x}(t) \in \mathcal{L}_2^n(\mathbb{R})$ with absolutely continuous components and derivative $\dot{\mathbf{x}} \in \mathcal{L}_2^n(\mathbb{R})$. If $\mathbf{A}$ has eigenvalues that lie on the imaginary axis, then there is $\mathbf{f} \in \mathcal{L}_2^n(\mathbb{R})$ such that $\mathcal{A}\mathbf{x} = \mathbf{f}$ does not have a solution in $\mathcal{L}_2^n(\mathbb{R})$. On the other hand, if $\mathbf{A}$ has no purely imaginary eigenvalues, $\mathcal{A}\mathbf{x} = \mathbf{f}$ has a unique solution in $\mathcal{L}_2^n(\mathbb{R})$. Using the definition of $\mathcal{A}$, we obtain the following input-output mapping

$$\mathbf{y}(t) = [\mathbf{C}\mathcal{A}^{-1}\mathbf{B}]\mathbf{u}(t) = \int_{-\infty}^{\infty} h(t - \tau)\mathbf{u}(\tau)d\tau, \quad (2.6.1)$$

which is also a convolution operator. Following the discussion in [134], if the eigenvalues of $\mathbf{A}$ lie to the left of the imaginary axis, we have

$$\begin{aligned} [\mathcal{A}^{-1}\mathbf{f}](t) &= \mathbf{x}(t) = \int_{-\infty}^t e^{\mathbf{A}(t-\tau)}\mathbf{f}(\tau)d\tau, \\ \mathbf{y}(t) &= [\mathbf{C}\mathcal{A}^{-1}\mathbf{B}\mathbf{u}](t) = \int_{-\infty}^{\infty} h(t-\tau)\mathbf{u}(\tau)d\tau, \end{aligned}$$

and

$$h(t) = \begin{cases} \mathbf{C}e^{\mathbf{A}t}\mathbf{B}, & \text{for } t \geq 0; \\ 0, & \text{for } t < 0. \end{cases} \quad (2.6.2)$$

On the other hand, if the eigenvalues of $\mathbf{A}$ have positive real parts, then

$$\begin{aligned} [\mathcal{A}^{-1}\mathbf{f}](t) &= \mathbf{x}(t) = - \int_t^{-\infty} e^{\mathbf{A}(t-\tau)}\mathbf{f}(\tau)d\tau, \\ \mathbf{y}(t) &= [\mathbf{C}\mathcal{A}^{-1}\mathbf{B}\mathbf{u}](t) = \int_{-\infty}^{\infty} h(t-\tau)\mathbf{u}(\tau)d\tau, \end{aligned}$$

and

$$h(t) = \begin{cases} 0, & \text{for } t \geq 0; \\ -\mathbf{C}e^{\mathbf{A}t}\mathbf{B}, & \text{for } t < 0. \end{cases} \quad (2.6.3)$$

If the system is unstable, i.e., the eigenvalues of $\mathbf{A}$ lie both to the left and to right of the imaginary axis, we separate the system into two parts, where one is stable and the other is antistable. Let $\mathbf{X}^+$ be a basis for $\mathcal{U}^+$ and $\mathbf{X}^-$ for $\mathcal{U}^-$, where $\mathcal{U}^+$ and $\mathcal{U}^-$ are invariant subspaces of $\mathbf{A}$ corresponding to stable and antistable eigenvalues, respectively. In other words, $\mathcal{U}^+ = \text{Ran}(\mathbf{X}^+)$ and $\mathcal{U}^- = \text{Ran}(\mathbf{X}^-)$. Since $\dim(\mathcal{U}^+) + \dim(\mathcal{U}^-) = n$, the matrix $\mathbf{X} = [\mathbf{X}^+ \ \mathbf{X}^-]$ has rank n , hence it's nonsingular. Thus, we can write

$$\mathbf{A}[\mathbf{X}^+ \ \mathbf{X}^-] = [\mathbf{X}^+ \ \mathbf{X}^-] \begin{bmatrix} \mathbf{M}^+ & 0 \\ 0 & \mathbf{M}^- \end{bmatrix},$$

where $\mathbf{M}^+$ is stable and $\mathbf{M}^-$ is antistable. If we let $\mathbf{Y} = (\mathbf{X}^{-1})^* = [\mathbf{Y}^+ \ \mathbf{Y}^-]$, then,

$$\mathbf{\Pi}^+ = \mathbf{X}^+(\mathbf{Y}^+)^* \text{ and } \mathbf{\Pi}^- = \mathbf{X}^-(\mathbf{Y}^-)^*$$

are the stable and antistable projectors for $\mathbf{A}$. These spectral projectors enable us to separate a linear unstable system into its stable and antistable components. In this case we have

$$\mathbf{y}(t) = [\mathbf{CA}^{-1}\mathbf{Bu}](t) = \int_{-\infty}^{\infty} \mathbf{h}(t - \tau)\mathbf{u}(\tau)d\tau,$$

where

$$\mathbf{h}(t) = \begin{cases} \mathbf{C}e^{\mathbf{A}t}\mathbf{\Pi}^+\mathbf{B}, & \text{for } t \geq 0; \\ -\mathbf{C}e^{\mathbf{A}t}\mathbf{\Pi}^-\mathbf{B}, & \text{for } t < 0; \end{cases}. \quad (2.6.4)$$

As we can see, it is possible to separate an unstable system into its stable and antistable components. Since IRKA enables us to reduce stable systems with high accuracy, the $\mathcal{L}_2$ optimality conditions in [134] requires the reduced order stable subsystem to be an $\mathcal{H}_2$ optimal approximation of the full order stable subsystem, and the negative of the reduced order antistable component of the reduced unstable system is also an $\mathcal{H}_2$ optimal approximation of the negative full order antistable subsystem. After reducing each component separately, we negate the component which corresponds to the antistable part of the system; then, we combine the reduced components together. Hence, we obtain a reduced unstable system. Now, let us explore the $\mathcal{L}_2$ error in the frequency domain. Recall in Section 2.3 we obtained the frequency domain representation of the dynamical systems by taking the Laplace transform of the time representations (1.1.1) and (1.1.2). However, the existence of the Laplace transform of $\mathbf{u}$ is not guaranteed if $\mathbf{u} \in \mathcal{L}_2^n(\mathbb{R})$. Thus, we apply a Fourier transform to (2.6.4) and obtain

$$\hat{\mathbf{y}}(\omega) = \mathbf{C}(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{\Pi}^+\mathbf{B}\hat{\mathbf{u}}(\omega) + \mathbf{C}(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{\Pi}^-\mathbf{B}\hat{\mathbf{u}}(\omega) = \mathbf{C}(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\hat{\mathbf{u}}(\omega),$$

where $\hat{\mathbf{y}}(\omega)$ and $\hat{\mathbf{u}}(\omega)$ are the Fourier transforms of $\mathbf{y}(t)$ and $\mathbf{u}(t)$, respectively. Then,

$$\mathbf{H}^+(i\omega)\hat{\mathbf{u}}(\omega) + \mathbf{H}^-(i\omega)\hat{\mathbf{u}}(\omega) = \mathbf{H}(i\omega)\hat{\mathbf{u}}(\omega),$$

where $\mathbf{H}$ is the total transfer function, $\mathbf{H}^+$ the transfer function of the stable part and $\mathbf{H}^-$ the transfer function of the antistable part. Let $\mathcal{L}_2(i\mathbb{R})$ be the Hilbert space whose elements are the meromorphic functions $\mathbf{G}(s)$ such that $\int_{-\infty}^{\infty} |\mathbf{G}(i\omega)|^2 d\omega$ is finite. Note that the transfer functions of the stable and the antistable parts of the dynamical systems are contained in $\mathcal{L}_2(i\mathbb{R})$. Then, for any two functions $\mathbf{G}, \mathbf{H}$ in $\mathcal{L}_2(i\mathbb{R})$ that represent real dynamical systems, the inner product is defined as

$$\langle \mathbf{G}, \mathbf{H} \rangle = \int_{-\infty}^{\infty} \mathbf{G}(-i\omega) \mathbf{H}(i\omega) d\omega.$$

As a result, the $\mathcal{L}_2(i\mathbb{R})$ norm of $\mathbf{H}$ is

$$\|\mathbf{H}\|_{\mathcal{L}_2} = \left(\int_{-\infty}^{\infty} |\mathbf{H}(i\omega)|^2 d\omega \right)^{1/2}. \quad (2.6.5)$$

We know $\mathcal{L}_2(i\mathbb{R})$ can be written as a direct sum : $\mathcal{L}_2(i\mathbb{R}) = \mathcal{H}_2(\mathbb{C}^-) \oplus \mathcal{H}_2(\mathbb{C}^+)$ where $\mathcal{H}_2(\mathbb{C}^-)$ denotes the set of functions that are analytic in the open left half plane $\mathbb{C}^-$ and $\mathcal{H}_2(\mathbb{C}^+)$ denotes the set of functions that are analytic in the open right half plane $\mathbb{C}^+$. If $\mathbf{H}_r^+$ denotes the stable reduced subsystem, $\mathbf{H}_r^-$ denotes the antistable reduced subsystem, and $\mathbf{H}_r$ denotes the total reduced unstable system, then we have

$$\|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{L}_2}^2 = \|\mathbf{H}^+ - \mathbf{H}_r^+\|_{\mathcal{H}_2(\mathbb{C}^+)}^2 + \|\mathbf{H}^- - \mathbf{H}_r^-\|_{\mathcal{H}_2(\mathbb{C}^-)}^2.$$

The following theorem [134] presents the interpolatory $\mathcal{L}_2$ optimality conditions for unstable systems.

Theorem 2.12. [134] *Let $\mathbf{H}_r(s)$ be the best r^{th} order approximation of an $\mathcal{L}_2$ system $\mathbf{H}(s)$. Moreover, suppose $\mathbf{H}_r(s)$ has simple poles $\{\lambda_k\}_{k=1}^r$ such that the first j poles are stable and the last $r - j$ poles are antistable. Then*

$$\begin{aligned}\mathbf{H}^+(-\lambda_k) &= \mathbf{H}_r^+(-\lambda_k), \text{ and} \\ (\mathbf{H}^+)'(-\lambda_k) &= (\mathbf{H}_r^+)'(-\lambda_k)\end{aligned}\tag{2.6.6}$$

for $k = 1, \dots, j$; and

$$\begin{aligned}\mathbf{H}^-(-\lambda_k) &= \mathbf{H}_r^-(-\lambda_k), \text{ and} \\ (\mathbf{H}^-)'(-\lambda_k) &= (\mathbf{H}_r^-)'(-\lambda_k)\end{aligned}\tag{2.6.7}$$

for $k = j + 1, \dots, r$.

Thus, in order to compute the $\mathcal{L}_2$ error in optimal $\mathcal{L}_2$ model reduction, we need to compute the $\mathcal{H}_2$ error obtained during the reduction of the stable and antistable components. For a detailed description of $\mathcal{L}_2$ IRKA and a proof of Theorem 2.12, we refer the reader to [134]. Algorithm 2 presents a sketch of $\mathcal{L}_2$ IRKA.

2.6.2 Balanced Truncation for Unstable Systems

The concepts of gramians and model reduction by balanced truncation can be extended to unstable dynamical systems with no poles on the imaginary axis [189]. Let's write (1.1.1) as

$$\Sigma := \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & 0 \end{array} \right].\tag{2.6.8}$$

In order to get a balanced realization of the system, we need to compute the reachability and observability gramians $\mathcal{P}$ and $\mathcal{Q}$. For stable systems we obtain the gramians $\mathcal{P}$ and $\mathcal{Q}$ by solving the Lyapunov equations (2.3.8). However, if $\mathbf{A}$ has eigenvalues whose real parts are positive, the gramians of the system cannot be defined in the same way as in Definition

Algorithm 2 $\mathcal{L}_2$ IRKA Pseudocode

Input: Original state space matrices and initial shift selection

Output: Reduced state space matrices

- Decompose $\mathbf{H}$ into minimal stable and antistable systems.
 - Make an initial shift selection closed under conjugation and ordered as follows:
 $\{\sigma_1, \dots, \sigma_k\} \subset \mathbb{C}^+$ and $\{\sigma_{k+1}, \dots, \sigma_r\} \subset \mathbb{C}^-$.
 - Negate the antistable subsystem.
 - Reduce each subsystem via IRKA.
 - Negate the reduced subsystem corresponding to the antistable subsystem.
 - Add the reduced stable and antistable systems.
-

2.6 since the infinite integral in the time domain would not make sense. Hence, we need to redefine the reachability and observability gramians. Suppose $\mathbf{T}$ is a transformation such that

$$\left[\begin{array}{c|c} \mathbf{TAT}^{-1} & \mathbf{TB} \\ \hline \mathbf{CT}^{-1} & 0 \end{array} \right] = \left[\begin{array}{cc|c} \mathbf{A}_1 & 0 & \mathbf{B}_1 \\ 0 & \mathbf{A}_2 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{C}_2 & 0 \end{array} \right],$$

where $\mathbf{A}_1$ is stable and $\mathbf{A}_2$ is antistable. Let $\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2 \geq 0$ be solutions to the Lyapunov equations:

$$\begin{aligned} \mathbf{A}_1 \mathcal{P}_1 + \mathcal{P}_1 \mathbf{A}_1^T + \mathbf{B}_1 \mathbf{B}_1^T &= 0, \\ \mathbf{A}_1^T \mathcal{Q}_1 + \mathcal{Q}_1 \mathbf{A}_1 + \mathbf{C}_1^T \mathbf{C}_1 &= 0, \\ (-\mathbf{A}_2) \mathcal{P}_2 + \mathcal{P}_2 (-\mathbf{A}_2)^T + \mathbf{B}_2 \mathbf{B}_2^T &= 0, \text{ and} \\ (-\mathbf{A}_2)^T \mathcal{Q}_2 + \mathcal{Q}_2 (-\mathbf{A}_2) + \mathbf{C}_2^T \mathbf{C}_2 &= 0. \end{aligned}$$

Additionally, [189] showed $\mathcal{P}$ and $\mathcal{Q}$ can be computed as

$$\mathcal{P} = \mathbf{T}^{-1} \begin{bmatrix} \mathcal{P}_1 & 0 \\ 0 & \mathcal{P}_2 \end{bmatrix} \mathbf{T}^{-T},$$

and

$$\mathcal{Q} = \mathbf{T}^{-1} \begin{bmatrix} \mathcal{Q}_1 & 0 \\ 0 & \mathcal{Q}_2 \end{bmatrix} \mathbf{T}^{-T}.$$

The reachability and observability gramians, $\mathcal{P}$ and $\mathcal{Q}$ do indeed satisfy the frequency domain definition of the gramians [189].

Recall the generalized Hankel singular values are defined as $\sigma_i = \sqrt{\lambda_i(\mathcal{P}\mathcal{Q})}$ such that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$. In other words, the generalized Hankel singular values of an unstable system can be computed via the Hankel singular values of its stable and antistable components. After this point, we follow the same steps as in balanced truncation for stable systems. This method eliminates the states associated with the smallest Hankel singular values without making a distinction between the values associated with the stable subsystem and those associated with the antistable part. The only criterion for the elimination is the magnitude of the Hankel singular values. If the dynamical system (2.6.8) has no poles on the imaginary axis, then neither truncated system has any poles on the imaginary axis, and the same upper bound as in (2.5.7) holds [189].

Chapter 3

Finite Horizon Model Reduction

Generally, for asymptotically stable linear models, techniques such as balanced truncation (BT) [139, 140] and the Iterative Rational Krylov Algorithm (IRKA) [86] produce high fidelity reduced order models. However, balanced truncation and IRKA approximate the large-scale system with a smaller system over the time interval $[0, \infty)$. In this chapter and throughout this dissertation we refer to the interval $[0, \infty)$ as an infinite time horizon. To differentiate from an infinite time horizon, we refer to the interval $[0, t_f]$, where $t_f > 0$, as a finite horizon. As we observed in Section 2.6, reducing the order of a model over an infinite horizon introduces additional challenges for unstable systems. Furthermore, in industrial and scientific applications we run simulations over a finite time horizon. In this chapter we discuss model reduction over a finite horizon in general, and how we can take advantage of the methods developed here to reduce unstable systems with high fidelity.

3.1 Reduced Order Modeling on a Finite Horizon

Recall the linear dynamical system:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{3.1.1}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times n}$ are constant matrices. The variable $\mathbf{x}(t) \in \mathbb{R}^n$ denotes an internal variable, $\mathbf{u}(t) \in \mathbb{R}^m$ denotes the control inputs, and $\mathbf{y}(t) \in \mathbb{R}^p$ denotes the outputs. We want to replace the original model with a lower dimension model:

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{B}_r \mathbf{u}(t), \\ \mathbf{y}_r(t) &= \mathbf{C}_r \mathbf{x}_r(t),\end{aligned}\tag{3.1.2}$$

where $\mathbf{A}_r \in \mathbb{R}^{r \times r}$, $\mathbf{B}_r \in \mathbb{R}^{r \times m}$, and $\mathbf{C}_r \in \mathbb{R}^{p \times r}$ with $r \ll n$. Our goal is to approximate the true outputs of the original system with the outputs of the reduced system for the same input in an appropriate norm. In this chapter, we only explore the behavior of the dynamical system over a finite time interval. Existing techniques like time-limited balanced truncation (TLBT) [75, 87, 126] and Proper Orthogonal Decomposition (POD) [108] yield high-fidelity reduced models, however, none of these methods produce any locally optimal model with respect to the output. For TLBT error upper bounds, see [87, 153, 154]. Proper Orthogonal Decomposition is optimal in projecting the observed data. For a more thorough discussion of TLBT, see Section 3.3, and for POD, see Section 5.1.2.

In this chapter, we establish $\mathcal{H}_2(t_f)$ optimality conditions over a finite time horizon and set the stage for an algorithm that yields better approximations of the large-scale dynamical systems compared to other existing model reduction methods. With this goal in mind, we explore initially the gramian based framework [82, 91, 181], the Lyapunov equations associated with the dynamical system on a finite horizon, and the relationship between the $\mathcal{H}_2(t_f)$ norm and the finite-time Gramians to obtain finite-time optimality conditions. However, in the Gramian-based framework, the reduced quantities satisfy the optimality conditions only approximately [82]. Inspired by the frequency based framework and the $\mathcal{H}_2$ optimality conditions for a full horizon in [86], we take advantage of the time-domain representation of the dynamical system to derive the conditions for a finite time horizon. We

represent the impulse response of the reduced dynamical system as follows:

$$\mathbf{h}_r(t) = \mathbf{C}_r e^{\mathbf{A}_r t} \mathbf{B}_r = \sum_{i=1}^r e^{\lambda_i t} \boldsymbol{\ell}_i \mathbf{r}_i^T, \quad (3.1.3)$$

where λ_i 's are the eigenvalues of $\mathbf{A}_r$, and $\boldsymbol{\ell}_i \in \mathbb{C}^{p \times 1}$, $\mathbf{r}_i \in \mathbb{C}^{m \times 1}$. In other words, the impulse response is expressed as a sum of r rank-1 $p \times m$ matrices. To simplify the presentation, we assume that λ_i 's, the reduced order poles, are simple. The representation (3.1.3) is nothing but a state-space transformation on $\mathbf{h}_r(t) = \mathbf{C}_r e^{\mathbf{A}_r t} \mathbf{B}_r$ using the eigenvectors of $\mathbf{A}_r$. Using the parametrization of the reduced model in (3.1.3), we derive interpolatory optimality conditions in the $\mathcal{H}_2(t_f)$ norm and implement a model reduction algorithm that satisfies these optimality conditions.

This approach allows us to implement a more efficient model reduction algorithm. Establishing $\mathcal{H}_2(t_f)$ optimality conditions for model reduction over a finite horizon also enables us to reduce unstable systems optimally under the $\mathcal{H}_2(t_f)$ measure [166].

3.2 Error Measures on a Finite Horizon

In this section we introduce an appropriate error measure for the approximation of dynamical systems on a finite horizon. In Chapter 2 we explored error measures in the frequency domain. Nonetheless, the error analysis for linear dynamical systems in the frequency and time domain are equivalent. Recall from Chapter 2 that

$$\begin{aligned} \mathbf{Y}(s) &= \mathbf{H}(s) \mathbf{U}(s) \quad \text{with} \quad \mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B}, \text{ and} \\ \mathbf{Y}_r(s) &= \mathbf{H}_r(s) \mathbf{U}(s) \quad \text{with} \quad \mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1} \mathbf{B}_r, \end{aligned}$$

where $\mathbf{H}(s)$ and $\mathbf{H}_r(s)$ are the transfer functions associated with the full and reduced models in (3.1.1) and (3.1.2), respectively. Also, $\mathbf{Y}(s)$, $\mathbf{Y}_r(s)$, and $\mathbf{U}(s)$ are the Fourier transforms of the outputs of the full and reduced models, $\mathbf{y}(t)$, $\mathbf{y}_r(t)$ and of the control input $\mathbf{u}(t)$. Therefore,

$$\mathbf{Y}(s) - \mathbf{Y}_r(s) = (\mathbf{H}(s) - \mathbf{H}_r(s))\mathbf{U}(s).$$

From Definition 2.4 we have

$$\langle \mathbf{G}(s), \mathbf{H}(s) \rangle_{\mathcal{H}_2} := \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}(\overline{\mathbf{G}}(-i\omega) \mathbf{H}(i\omega)^T) d\omega,$$

and

$$\|\mathbf{H}(s)\|_{\mathcal{H}_2} := \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{H}(i\omega)\|_F^2 d\omega \right)^{1/2}.$$

Recall that the transfer function of a dynamical system is the Fourier transform of the impulse response. The $\mathcal{H}_2$ norm in the time domain is defined in terms of the impulse response. The formal definition follows.

Definition 3.1. If $\mathbf{g}(t)$ and $\mathbf{h}(t)$ are the impulse responses of asymptotically stable linear dynamical systems, the $\mathcal{H}_2$ inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_2}$ in the time domain is

$$\langle \mathbf{h}(t), \mathbf{g}(t) \rangle_{\mathcal{H}_2} = \int_0^{\infty} \text{tr}(\mathbf{h}(t)^T \mathbf{g}(t)) dt,$$

and, as a result,

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2}^2 = \int_0^{\infty} \|\mathbf{h}\|_F^2 dt,$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Recall that the transfer function error is an upper bound for the output error between the full and the reduced models [10], as shown in the following inequality:

$$\|\mathbf{y} - \mathbf{y}_r\|_{L_\infty} \leq \|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_2} \|\mathbf{u}\|_{L_2}. \quad (3.2.1)$$

Therefore, approximating the transfer function allows us to obtain a reduced order model whose output is very close to the original output. Similarly, if we approximate the impulse response, we bound the output error. However, in scientific and industrial applications we do not simulate models for infinite time horizons. Minimizing the $\mathcal{H}_2$ norm of the error produces a good approximation over the infinite horizon; nevertheless, we could get better results if we focus our efforts over a finite horizon of interest and allow for larger errors outside of the interval of interest. Our analysis for model reduction over a finite time horizon is naturally conducted in the time domain. Inspired by the infinite time horizon $\mathcal{H}_2$ inner product and norm, we define the $\mathcal{H}_2(t_f)$ inner product and norm for a finite horizon as follows.

Definition 3.2. If $\mathbf{g}(t)$ and $\mathbf{h}(t)$ are the impulse responses of linear dynamical systems, the $\mathcal{H}_2(t_f)$ inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_2(t_f)}$ in the time domain is

$$\langle \mathbf{h}(t), \mathbf{g}(t) \rangle_{\mathcal{H}_2(t_f)} = \int_0^{t_f} \text{tr}(\mathbf{h}(t)\mathbf{g}(t)^T) dt,$$

and, as a result,

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 = \int_0^{t_f} \|\mathbf{h}(t)\|_F^2 dt.$$

In addition to enabling us to obtain better approximations over a finite horizon, the time-limited $\mathcal{H}_2(t_f)$ norm provides an error measure for model reduction of unstable systems.

Similar to the $\mathcal{H}_2$ norm, the $\mathcal{H}_2(t_f)$ norm can be computed by the time-limited gramians associated with the dynamical system (3.1.1).

Let $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ be the state space matrices of the dynamical system (3.1.1) and $t \in (0, t_f]$. Then the time-limited reachability gramian is

$$\mathcal{P}(t_f) := \int_0^{t_f} e^{\mathbf{A}t} \mathbf{B} \mathbf{B}^T e^{\mathbf{A}^T t} dt, \quad (3.2.2)$$

and the time-limited observability gramian is

$$\mathcal{Q}(t_f) := \int_0^{t_f} e^{\mathbf{A}^T t} \mathbf{C}^T \mathbf{C} e^{\mathbf{A}t} dt. \quad (3.2.3)$$

The time-limited gramians solve the following Lyapunov equations for $\mathcal{P}(t_f)$ and $\mathcal{Q}(t_f)$:

$$\begin{aligned} \mathbf{A} \mathcal{P}(t_f) + \mathcal{P}(t_f) \mathbf{A}^T + \mathbf{B} \mathbf{B}^T - e^{\mathbf{A}t_f} \mathbf{B} \mathbf{B}^T e^{\mathbf{A}^T t_f} &= 0 \\ \mathbf{A}^T \mathcal{Q}(t_f) + \mathcal{Q}(t_f) \mathbf{A} + \mathbf{C}^T \mathbf{C} - e^{\mathbf{A}^T t_f} \mathbf{C}^T \mathbf{C} e^{\mathbf{A}t_f} &= 0. \end{aligned} \quad (3.2.4)$$

The following corollary describes the relation between the $\mathcal{H}_2(t_f)$ norm and the finite time gramians depicted in (3.2.2) and (3.2.3).

Corollary 3.3. *Given the time-limited reachability and observability gramians in (3.2.2) and (3.2.3), the $\mathcal{H}_2(t_f)$ norm is computed as follows:*

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)} = \sqrt{\text{tr}(\mathbf{C} \mathcal{P}(t_f) \mathbf{C}^T)} = \sqrt{\text{tr}(\mathbf{B}^T \mathcal{Q}(t_f) \mathbf{B})}. \quad (3.2.5)$$

Proof. From Definition 3.2 we have

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 = \int_0^{t_f} \|\mathbf{h}(t)\|_F^2 dt.$$

Therefore,

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 = \int_0^{t_f} \text{tr}(\mathbf{h}(t)(\mathbf{h}(t)^T) dt. \quad (3.2.6)$$

Plugging $\mathbf{h}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B}$ into (3.2.6), we obtain

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 = \int_0^{t_f} \text{tr}(\mathbf{C}e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t}\mathbf{C}^T) dt. \quad (3.2.7)$$

Since $\mathbf{C}$ is a constant matrix, we can rewrite Equation (3.2.7) as

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 = \text{tr} \left(\mathbf{C} \left(\int_0^{t_f} e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt \right) \mathbf{C}^T \right) dt.$$

Since

$$\mathcal{P}(t_f) := \int_0^{t_f} e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt,$$

we have

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 = \text{tr}(\mathbf{C}\mathcal{P}(t_f)\mathbf{C}^T).$$

As a result,

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)} = \sqrt{\text{tr}(\mathbf{C}\mathcal{P}(t_f)\mathbf{C}^T)}.$$

Similarly, we can write

$$\begin{aligned} \|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 &= \int_0^{t_f} \text{tr}((\mathbf{h}(t))^T \mathbf{h}(t)) dt \\ &= \int_0^{t_f} \text{tr}(\mathbf{B}^T e^{\mathbf{A}^T t} \mathbf{C}^T \mathbf{C} e^{\mathbf{A}t} \mathbf{B}) dt \\ &= \text{tr} \left(\mathbf{B}^T \left(\int_0^{t_f} e^{\mathbf{A}^T t} \mathbf{C}^T \mathbf{C} e^{\mathbf{A}t} dt \right) \mathbf{B} \right) dt. \end{aligned} \quad (3.2.8)$$

Thus,

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)}^2 = \text{tr}(\mathbf{B}^T \mathcal{Q}(t_f) \mathbf{B}),$$

and

$$\|\mathbf{h}(t)\|_{\mathcal{H}_2(t_f)} = \sqrt{\text{tr}(\mathbf{B}^T \mathcal{Q}(t_f) \mathbf{B})}.$$

□

We use the definitions of the finite time norms and the computation methods presented in this section to derive $\mathcal{H}_2(t_f)$ optimality conditions in later sections.

3.3 Time-limited Balanced Truncation

Consistent with the theme of this chapter, we explore model reduction via balanced truncation for a finite horizon. Balanced truncation reduces a model by taking advantage of the observability and reachability gramians which are defined on an infinite horizon. However, as we have already noted, no simulation continues infinitely over time, and practically we are interested in approximating large scale models only over a limited time interval. Therefore, we restrict the limits of integration and consider time-limited gramians on finite intervals [75, 87, 126].

Definition 3.4. Let $\{\lambda_i\}_{i=1}^n$ be the set of the eigenvalues of the product $\mathcal{P}(t_f)\mathcal{Q}(t_f)$. Define $\sigma_i = \sqrt{\lambda_i}$ for $i = 1, 2, \dots, n$. Then, $\{\sigma_i\}_{i=1}^n$ is the set of the time-limited Hankel singular values of the dynamical system for final time t_f .

Note that the Lyapunov operators in (3.2.4) are the same as in (2.3.8), where we deal with infinite-time Gramians, but the inhomogenities are different. The bulk of the cost for balanced truncation consists of the solutions to the corresponding Lyapunov equations, hence, dealing with these inhomogenities is essential [126]. One approach approximates the exponential terms $e^{\mathbf{A}t_f}\mathbf{B}$ and $e^{\mathbf{A}^T t_f}\mathbf{C}^T$ via rational Krylov subspaces. After approximating

the exponential terms, we can compute the low rank factors of the time-limited gramians by solving the corresponding Lyapunov equations. Once we have computed the low-rank factors, we perform square root balancing similar to balanced truncation in the infinite horizon case in Chapter 2. The success of time-limited balanced truncation hinges on our capability to accurately approximate the finite time gramians $\mathcal{P}(t_f)$ and $\mathcal{Q}(t_f)$. For an in-depth discussion of how the time-limited gramians compare to the infinite gramians and how to approximate the action of the matrix exponentials we refer the reader to [126].

Depending on the interval of interest, time-limited balanced truncation does not necessarily preserve stability when reducing stable systems. Therefore, it does not guarantee an $\mathcal{H}_\infty$ upper bound similar to (2.5.7). Nonetheless, upper bounds for time-limited balanced truncation do exist. For instance, [154] establishes the following bound for the output error.

Theorem 3.5. [154] *Let $\mathbf{A}_r$ be a real matrix. Suppose the eigenvalues of $\mathbf{A}_r$ and $-\mathbf{A}_r$ do not overlap. Assume further that $\mathbf{A}$ and $-\mathbf{A}_r$ do not have any eigenvalues in common. Let $\mathcal{P}(t_f)$ and $\mathcal{P}_r(t_f)$ denote the reachability gramians of the full and reduced systems, respectively. Let $\mathcal{P}_{12}(t_f)$ be the solution to the following Sylvester equation:*

$$\mathbf{A}\mathcal{P}_{12}(t_f) + \mathcal{P}_{12}(t_f)\mathbf{A}_r + \mathbf{B}\mathbf{B}_r^T - e^{\mathbf{A}t_f}\mathbf{B}\mathbf{B}_r^T e^{\mathbf{A}_r t_f} = 0.$$

Then, for a reduced system generated by TLBT we have

$$\max_{t \in [0, t_f]} \|\mathbf{y}(t) - \mathbf{y}_r(t)\|_2 \leq \epsilon \|\mathbf{u}(t)\|_{L_{t_f}^2}, \quad (3.3.1)$$

where

$$\epsilon := \sqrt{\text{tr}(\mathbf{C}\mathcal{P}(t_f)\mathbf{C}^T) + \text{tr}(\mathbf{C}_r\mathcal{P}_r(t_f)\mathbf{C}_r^T) - 2\text{tr}(\mathbf{C}\mathcal{P}_{12}(t_f)\mathbf{C}_r^T)},$$

and

$$\|\mathbf{u}(t)\|_{L^2_{t_f}} = \int_0^{t_f} (\mathbf{u}(t))^T \mathbf{u}(t) dt.$$

For an L^2_T error bound, see [153]. Furthermore, [87] presents a modified version of the method presented in [75] that provides a simple $\mathcal{H}_\infty$ upper bound for the error.

Provided the eigenvalues of $\mathbf{A}$ and $-\mathbf{A}$ do not overlap with each other, the time-limited gramians exist for unstable dynamical systems. In such cases, time-limited balanced truncation is a good candidate for model reduction of unstable systems.

3.4 Gramian based $\mathcal{H}_2(t_f)$ optimality conditions

Wilson [181] established the $\mathcal{H}_2$ optimality conditions over an infinite horizon by taking advantage of the infinite gramians associated with the dynamical system. Halevi [91] followed a gramian-based approach to obtain the $\mathcal{H}_2$ optimality conditions for weighted model reduction. In this section we briefly review the Wilson framework for the infinite horizon case. Subsequently, we discuss the gramian based optimality conditions derived in [82]. Using the $\mathcal{H}_2(t_f)$ norm definition and similar techniques as in [181] and [91], [82] obtains an expression for the error system. The necessary $\mathcal{H}_2(t_f)$ optimality conditions for model reduction are attained by differentiating the $\mathcal{H}_2(t_f)$ error expression with respect to the reduced matrices.

Let us start by discussing the infinite horizon case. Recall $\mathbf{H} = \mathbf{C}(s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}$ is the transfer function of the full system of order n and $\mathbf{H}_r = \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r$ the transfer function of the reduced system of order r . Then,

$$\mathbf{H}_e = \mathbf{H} - \mathbf{H}_r = \mathcal{C}(s\mathcal{J}_{n+r} - \mathcal{A})^{-1}\mathcal{B} \quad (3.4.1)$$

is the error system, where

$$\mathcal{C} = [\mathbf{C} \quad -\mathbf{C}_r], \quad \mathcal{A} = \begin{bmatrix} \mathbf{A} & \\ & \mathbf{A}_r \end{bmatrix}, \quad \text{and} \quad \mathcal{B} = \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_r \end{bmatrix}.$$

Let $\mathcal{P}_e$ be the infinite horizon reachability gramian and $\mathcal{Q}_e$ the infinite horizon observability gramian for the error system (3.4.1). Based on Definition 2.6, we have

$$\begin{aligned} \mathcal{P}_e &= \int_0^\infty e^{\mathcal{A}t} \mathcal{B} \mathcal{B}^T e^{\mathcal{A}^T t} dt, \quad \text{and} \\ \mathcal{Q}_e &= \int_0^\infty e^{\mathcal{A}^T t} \mathcal{C}^T \mathcal{C} e^{\mathcal{A}t} dt. \end{aligned} \tag{3.4.2}$$

Therefore, $\mathcal{P}_e$ and $\mathcal{Q}_e$ are the solutions to the following Lyapunov equations:

$$\begin{aligned} \mathcal{A} \mathcal{P}_e + \mathcal{P}_e \mathcal{A}^T + \mathcal{B} \mathcal{B}^T &= 0, \quad \text{and} \\ \mathcal{A}^T \mathcal{Q}_e + \mathcal{Q}_e \mathcal{A} + \mathcal{C}^T \mathcal{C} &= 0. \end{aligned} \tag{3.4.3}$$

Then, the $\mathcal{H}_2$ norm of the error system is

$$\|\mathcal{H}\|_{\mathcal{H}_2} = \sqrt{\text{tr}(\mathcal{C} \mathcal{P}_e \mathcal{C}^*)} = \sqrt{\text{tr}(\mathcal{B}^T \mathcal{Q}_e \mathcal{B})}. \tag{3.4.4}$$

Differentiating the $\mathcal{H}_2$ error expression in (3.4.4) subject to the Lyapunov equations (3.4.3), [181] establishes the following necessary optimality conditions.

Theorem 3.6. *Let $\mathbf{H}_r$ be the best r^{th} order rational approximation of $\mathbf{H}$ with respect to the $\mathcal{H}_2$ norm, i.e.,*

$$\|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_2} = \min_{\mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r} \|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_2}.$$

Then,

$$\mathbf{A}_r = -\mathcal{Q}_{22}\mathcal{Q}_{12}^T\mathbf{A}\mathcal{P}_{12}\mathcal{P}_{22}^{-1},$$

$$\mathbf{B}_r = -\mathcal{Q}_{22}\mathcal{Q}_{12}^T\mathbf{B}, \text{ and}$$

$$\mathbf{C}_r = \mathbf{C}\mathcal{P}_{12}\mathcal{P}_{22}^{-1},$$

where

$$\mathcal{P}_e = \begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} \\ \mathcal{P}_{12}^T & \mathcal{P}_{22} \end{bmatrix}$$

is the reachability gramian of the error system,

$$\mathcal{Q}_e = \begin{bmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} \\ \mathcal{Q}_{12}^T & \mathcal{Q}_{22} \end{bmatrix}$$

is the observability gramian, and $\mathcal{P}_{11}, \mathcal{Q}_{11} \in \mathbb{R}^{n \times n}$, $\mathcal{P}_{12}, \mathcal{Q}_{12} \in \mathbb{R}^{n \times r}$, and $\mathcal{P}_{22}, \mathcal{Q}_{22} \in \mathbb{R}^{r \times r}$.

The Wilson optimality conditions [181] are equivalent to the interpolation conditions in [86, 136]; see [27] for more details. Using similar tools to Wilson, [82] extends the gramian based optimality conditions to the time-limited case. From Equations (3.2.2) and (3.2.3) the time limited reachability and observability gramians of the error system are

$$\begin{aligned} \mathcal{P}_e(t_f) &= \int_0^{t_f} e^{\mathcal{A}t} \mathcal{B} \mathcal{B}^T e^{\mathcal{A}^T t} dt, \text{ and} \\ \mathcal{Q}_e(t_f) &= \int_0^{t_f} e^{\mathcal{A}^T t} \mathcal{C}^T \mathcal{C} e^{\mathcal{A}t} dt. \end{aligned} \tag{3.4.5}$$

From (3.2.4) follows that the gramians $\mathcal{P}_e(t_f)$ and $\mathcal{Q}_e(t_f)$ solve

$$\begin{aligned} \mathcal{A}\mathcal{P}(t_f) + \mathcal{P}(t_f)\mathcal{A}^T + \mathcal{B}\mathcal{B}^T - e^{\mathcal{A}t_f}\mathcal{B}\mathcal{B}^Te^{\mathcal{A}^T t_f} &= 0, \text{ and} \\ \mathcal{A}^T\mathcal{Q}(t_f) + \mathcal{Q}(t_f)\mathcal{A} + \mathcal{C}^T\mathcal{C} - e^{\mathcal{A}^T t_f}\mathcal{C}^T\mathcal{C}e^{\mathcal{A}t_f} &= 0. \end{aligned} \quad (3.4.6)$$

Consequently,

$$\|\mathcal{H}\|_{\mathcal{H}_2(t_f)} = \sqrt{\text{tr}(\mathcal{C}\mathcal{P}_e(t_f)\mathcal{C}^*)} = \sqrt{\text{tr}(\mathcal{B}^T\mathcal{Q}_e(t_f)\mathcal{B})}. \quad (3.4.7)$$

The Wilson optimality conditions have been extended to the finite horizon case in [82].

Theorem 3.7. *Let $\mathbf{A}_r$, $\mathbf{B}_r$, and $\mathbf{C}_r$ be the state-space matrices of a locally optimal reduced order approximation of the full order system (3.1.1) under the $\mathcal{H}_2(t_f)$ norm. Suppose*

$$\begin{aligned} \mathbf{A}_r &= \mathbf{S}^{-1}\mathbf{D}\mathbf{S}, \\ \tilde{\mathbf{B}}_r &= \mathbf{S}\mathbf{B}_r, \text{ and} \\ \tilde{\mathbf{C}}_r &= \mathbf{C}_r\mathbf{S}^{-1}, \end{aligned} \quad (3.4.8)$$

where $\mathbf{D}$ is a diagonal matrix. Furthermore, the gramians $\mathcal{P}_{12}(t_f)$, $\mathcal{P}_r(t_f)$, $\tilde{\mathcal{Q}}_{12}(t_f)$, $\tilde{\mathcal{Q}}_r(t_f)$, $\tilde{\mathcal{Q}}_r$ and $\tilde{\mathcal{Q}}_{12}$ solve the following Lyapunov equations respectively:

$$\begin{aligned} \mathbf{A}\tilde{\mathcal{P}}_{12}(t_f) + \tilde{\mathcal{P}}_{12}(t_f)\mathbf{D} + \mathbf{B}\tilde{\mathbf{B}}_r^T - e^{\mathbf{A}t_f}\mathbf{B}\tilde{\mathbf{B}}_r^Te^{\mathbf{D}t_f} &= 0, \\ \mathbf{D}\tilde{\mathcal{P}}_r(t_f) + \tilde{\mathcal{P}}_r(t_f)\mathbf{D} + \tilde{\mathbf{B}}_r\tilde{\mathbf{B}}_r^T - e^{\mathbf{D}t_f}\tilde{\mathbf{B}}_r\tilde{\mathbf{B}}_r^Te^{\mathbf{D}t_f} &= 0, \\ \mathbf{D}\tilde{\mathcal{Q}}_{12}(t_f) + \tilde{\mathcal{Q}}_{12}(t_f)\mathbf{A} + \tilde{\mathbf{C}}_r^T\mathbf{C} - e^{\mathbf{D}t_f}\tilde{\mathbf{C}}_r^T\mathbf{C}e^{\mathbf{D}t_f} &= 0, \\ \mathbf{D}\tilde{\mathcal{Q}}_r(t_f) + \tilde{\mathcal{Q}}_r(t_f)\mathbf{D} + \tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_r - e^{\mathbf{D}t_f}\tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_re^{\mathbf{D}t_f} &= 0, \\ \mathbf{D}\tilde{\mathcal{Q}}_r + \tilde{\mathcal{Q}}_r\mathbf{D} + \tilde{\mathbf{C}}_r^T\tilde{\mathbf{C}}_r &= 0, \text{ and} \\ \mathbf{D}\tilde{\mathcal{Q}}_{12} + \tilde{\mathcal{Q}}_{12}\mathbf{A}^T + \tilde{\mathbf{C}}_r^T\mathbf{C} &= 0. \end{aligned}$$

Then, it holds that

$$\begin{aligned}\tilde{\mathbf{C}}_r &= \mathbf{C}\tilde{\mathcal{P}}_{12}(t_f)\tilde{\mathcal{P}}_r(t_f), \\ \tilde{\mathbf{B}}_r &= \tilde{\mathcal{Q}}_r(t_f)\tilde{\mathcal{Q}}_{12}(t_f)\mathbf{B}, \text{ and} \\ \mathbf{e}_i^T \tilde{\mathcal{Q}}_{12}[\tilde{\mathcal{P}}_{12}(t_f) - t_f e^{\mathbf{A}t_f} \mathbf{B}\tilde{\mathbf{B}}^T e^{\mathbf{D}t_f}] \mathbf{e}_i &= \mathbf{e}_i^T \tilde{\mathcal{Q}}_r[\tilde{\mathcal{P}}_r(t_f) - t_f e^{\mathbf{D}t_f} \tilde{\mathbf{B}}_r \tilde{\mathbf{B}}_r^T e^{\mathbf{D}t_f}] \mathbf{e}_i\end{aligned}\tag{3.4.9}$$

for all $i \in \{1, \dots, r\}$ where $\mathbf{e}_i$ is the i -th unit vector.

In addition to deriving the optimality conditions in Theorem 3.7, [82] also proposes a time-limited IRKA type algorithm that approximately satisfies the established optimality conditions.

We revisit Theorem 3.7 and Algorithm 3 in the next sections after we discuss the interpolation based optimality conditions for the $\mathcal{H}_2(t_f)$ norm.

3.5 $\mathcal{H}_2(t_f)$ Optimal Model Reduction: MIMO Case

In this section we explore the following problem: Consider a dynamical system with impulse response

$$\mathbf{h}(t) = \sum_{i=1}^n e^{\rho_i t} \mathbf{c}_i \mathbf{b}_i^T, \tag{3.5.1}$$

or equivalently, with transfer function,

$$\mathbf{H}(s) = \sum_{i=1}^n \frac{\mathbf{c}_i \mathbf{b}_i^T}{s - \rho_i}, \tag{3.5.2}$$

where $\mathbf{c}_i \in \mathbb{R}^p$ and $\mathbf{b}_i \in \mathbb{R}^m$, for $i = 1, \dots, n$ are residue directions. This is called the pole-residue form, where the ρ_i 's are the poles of the (rational) transfer function $\mathbf{H}(s)$ with the corresponding rank-1 residues $\mathbf{c}_i \mathbf{b}_i^T$.

Algorithm 3 Time-limited IRKA-type Algorithm Pseudocode

Input: Original state space matrices

Output: Reduced state space matrices

- Make an initial guess for the reduced matrices
- while (not converged)
 - Compute an eigendecompositon of $\mathbf{A}_r$ and define $\tilde{\mathbf{B}}_r$, $\tilde{\mathbf{C}}_r$ as follows:

$$\mathbf{A}_r = \mathbf{S}^{-1}\mathbf{D}\mathbf{S},$$

$$\tilde{\mathbf{B}}_r = \mathbf{S}\mathbf{B}_r, \text{ and}$$

$$\tilde{\mathbf{C}}_r = \mathbf{C}_r\mathbf{S}^{-1}.$$

- Solve for $\mathbf{V}$ and $\mathbf{W}$:

$$-\mathbf{A}\mathbf{V} - \mathbf{V}\mathbf{D} = \mathbf{B}\tilde{\mathbf{B}}_r^T - e^{\mathbf{A}t_f}\mathbf{B}\tilde{\mathbf{B}}_r^T e^{\mathbf{D}t_f}$$

$$-\mathbf{W}\mathbf{D} - \mathbf{A}^T\mathbf{W} = \mathbf{C}^T\tilde{\mathbf{C}}_r - e^{\mathbf{A}^T t_f}\mathbf{C}^T\tilde{\mathbf{C}}_r e^{\mathbf{D}t_f}.$$

- Perform a change of basis so that $\mathbf{W}^T\mathbf{V} = \mathbf{I}$.
 - $\mathbf{A}_r = \mathbf{W}^T\mathbf{A}\mathbf{V}$, $\mathbf{B}_r = \mathbf{W}^T\mathbf{B}$, and $\mathbf{C}_r = \mathbf{C}\mathbf{V}$.
-

Given a reduced order r , the problem is to find the reduced model with the impulse response

$$\mathbf{h}_r(t) = \sum_{i=1}^r e^{\lambda_i t} \boldsymbol{\ell}_i \mathbf{r}_i^T \quad (3.5.3)$$

and transfer function

$$\mathbf{H}_r(s) = \sum_{i=1}^r \frac{\boldsymbol{\ell}_i \mathbf{r}_i^T}{s - \lambda_i}, \quad (3.5.4)$$

where $\boldsymbol{\ell}_i \in \mathbb{C}^p$ and $\mathbf{r}_i \in \mathbb{C}^m$, for $i = 1, \dots, r$ are the residue directions and the λ_i 's are the poles of the transfer function $\mathbf{H}_r(s)$ with the corresponding rank-1 residues $\boldsymbol{\ell}_i \mathbf{r}_i^T$ such that $\|\mathbf{h} - \mathbf{h}_r\|_{\mathcal{H}_2(t_f)}$ is minimized.

As in the regular $\mathcal{H}_2$ case, this is a non-convex optimization problem and we focus on local minimizers. Using the parametrization (3.5.3), we derive interpolation-based necessary conditions for optimality. The main result is given by Theorem 3.10. However, we need many supplementary results, Lemmas 3.8 and 3.9, to reach this final conclusion.

It is immediately clear that the $\mathcal{H}_2(t_f)$ -error, denoted by $\mathbf{J}$, satisfies

$$\begin{aligned} \mathbf{J} &= \|\mathbf{h} - \mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2 \\ &= \|\mathbf{h}\|_{\mathcal{H}_2(t_f)}^2 - 2\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} + \|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2, \end{aligned} \quad (3.5.5)$$

where the inner product $\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)}$ is real since $\mathbf{h}(t)$ and $\mathbf{h}_r(t)$ are real. Finding the first-order necessary conditions for optimal $\mathcal{H}_2(t_f)$ model reduction requires computing the gradient of the error expression (3.5.5) with respect to the optimization variables. Since the reduced model, as described by the impulse response in $\mathbf{h}_r(t)$, is parametrized by the reduced order poles $\{\lambda_i\}$, and the residue directions $\{\boldsymbol{\ell}_i\}$ and $\{\mathbf{r}_i\}$, we will compute the gradient of the error with respect to these variables. Since the first term in the error (3.5.5), i.e., $\|\mathbf{h}\|_{\mathcal{H}_2(t_f)}$, is a constant, we will be focusing on the remaining two terms only. First, in the next lemma, we will formulate these two last terms with regard to $\{\lambda_i\}$, $\{\boldsymbol{\ell}_i\}$ and $\{\mathbf{r}_i\}$.

Lemma 3.8. *Let $\mathbf{h}(t) = \sum_{j=1}^n e^{\rho_j t} \mathbf{c}_j \mathbf{b}_j^T$ and $\mathbf{h}_r(t) = \sum_{i=1}^r e^{\lambda_i t} \boldsymbol{\ell}_i \mathbf{r}_i^T$ be, respectively, the im-*

pulse responses of the full and reduced models as described in (3.5.1) and (3.5.3). Then,

$$\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} = \sum_{j=1}^n \sum_{i=1}^r \ell_i^T \mathbf{c}_j \mathbf{b}_j^T \mathbf{r}_i \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j} \quad (3.5.6)$$

and

$$\|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2 = \sum_{i=1}^r \sum_{j=1}^r \ell_i^T \ell_j \mathbf{r}_j^T \mathbf{r}_i \frac{e^{(\lambda_i + \lambda_j)t_f} - 1}{\lambda_i + \lambda_j}. \quad (3.5.7)$$

Proof. The both results follow from the definition of the $\mathcal{H}_2(t_f)$ inner product. First consider

$$\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} = \text{tr} \left(\int_0^{t_f} \mathbf{h}_r(t)^T \mathbf{h}(t) dt \right).$$

Plug $\mathbf{h}(t) = \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T e^{\rho_j t}$ and $\mathbf{h}_r(t) = \sum_{i=1}^r \ell_i \mathbf{r}_i^T e^{\lambda_i t}$ into this formula to obtain

$$\begin{aligned} \langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} &= \text{tr} \left(\int_0^{t_f} \sum_{i=1}^r (\ell_i \mathbf{r}_i^T e^{\lambda_i t})^T \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T e^{\rho_j t} dt \right) \\ &= \text{tr} \left(\sum_{i=1}^r \sum_{j=1}^n \mathbf{r}_i \ell_i^T \mathbf{c}_j \mathbf{b}_j^T \int_0^{t_f} e^{(\lambda_i + \rho_j)t} dt \right). \end{aligned}$$

Computing the integral and using the fact that $\text{tr}(\mathbf{A}_1 \mathbf{A}_2) = \text{tr}(\mathbf{A}_2 \mathbf{A}_1)$ for two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ of appropriate sizes, we obtain

$$\begin{aligned} \langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} &= \text{tr} \left(\sum_{j=1}^n \sum_{i=1}^r \mathbf{r}_i \ell_i^T \mathbf{c}_j \mathbf{b}_j^T \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j} \right) \\ &= \sum_{j=1}^n \sum_{i=1}^r \ell_i^T \mathbf{c}_j \mathbf{b}_j^T \mathbf{r}_i \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j}, \end{aligned}$$

which proves (3.5.6). Then, (3.5.7) follows directly by replacing $\mathbf{h}(t)$ with $\mathbf{h}_r(t)$ in this derivation. $\square$

For the infinite time horizon, Theorem 2.8 tells us that a locally $\mathcal{H}_2$ optimal reduced transfer

function is a tangential Hermite interpolant of the original transfer function at the mirror images of the reduced poles. We show that in the finite horizon case, even though Hermite tangential interpolation is still the necessary condition for optimality, the quantity being interpolated and the interpolant are different. Lemma 3.9 defines the interpolated function and the interpolant.

Lemma 3.9. *Let*

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, \quad (3.5.8)$$

with $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times n}$, be the transfer function of the full-order model with the pole-residue representation

$$\mathbf{H}(s) = \sum_{i=1}^n \frac{\mathbf{c}_i \mathbf{b}_i^T}{s - \rho_i}, \quad (3.5.9)$$

where $\mathbf{c}_i \in \mathbb{C}^p$, $\mathbf{b}_i \in \mathbb{C}^m$, and $\rho_i \in \mathbb{C}$ for $i = 1, \dots, n$. For a finite-time horizon $[0, t_f]$, define

$$\mathbf{G}(s) = -e^{-st_f} \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{B} + \mathbf{H}(s). \quad (3.5.10)$$

Let

$$\mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1} \mathbf{B}_r = \sum_{i=1}^r \frac{\boldsymbol{\ell}_i \mathbf{r}_i^T}{s - \lambda_i} \quad (3.5.11)$$

be the transfer function of an r^{th} order approximation of $\mathbf{H}(s)$ where $\mathbf{A}_r \in \mathbb{R}^{r \times r}$, $\mathbf{B}_r \in \mathbb{R}^{r \times m}$, $\mathbf{C}_r \in \mathbb{R}^{p \times r}$, $\boldsymbol{\ell}_i \in \mathbb{C}^p$, $\mathbf{r}_i \in \mathbb{C}^m$, and $\lambda_i \in \mathbb{C}$ for $i = 1, \dots, r$. Define

$$\mathbf{G}_r(s) = -e^{-st_f} \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1} e^{\mathbf{A}_r t_f} \mathbf{B}_r + \mathbf{H}_r(s). \quad (3.5.12)$$

Then,

$$\mathbf{G}(-\lambda_k) = \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j}, \quad (3.5.13)$$

$$\mathbf{G}'(-\lambda_k) = - \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{(t_f(\lambda_k + \rho_j) - 1)e^{(\lambda_k + \rho_j)t_f} + 1}{(\lambda_k + \rho_j)^2}, \quad (3.5.14)$$

$$\mathbf{G}_r(-\lambda_k) = \sum_{j=1}^r \boldsymbol{\ell}_j \mathbf{r}_j^T \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j}, \quad \text{and} \quad (3.5.15)$$

$$\mathbf{G}'_r(-\lambda_k) = - \sum_{j=1}^r \boldsymbol{\ell}_j \mathbf{r}_j^T \frac{(t_f(\lambda_k + \lambda_j) - 1)e^{(\lambda_k + \lambda_j)t_f} + 1}{(\lambda_k + \lambda_j)^2}. \quad (3.5.16)$$

Proof. Note the definition of $\mathbf{G}(s) = -e^{-st_f} \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{B} + \mathbf{H}(s)$ and recall our assumption that the eigenvalues of $\mathbf{A}$ are simple. Therefore, $e^{\mathbf{A}t_f}$ is also diagonalizable by the eigenvectors of $\mathbf{A}$. Using this fact and the pole-zero residue decomposition of $\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} = \sum_{j=1}^n \frac{\mathbf{c}_j \mathbf{b}_j^T}{s - \rho_j}$, we obtain

$$\begin{aligned} \mathbf{G}(s) &= -e^{-st_f} \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{B} + \mathbf{H}(s) \\ &= -e^{-st_f} \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{e^{\rho_j t_f}}{s - \rho_j} + \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{1}{s - \rho_j} \end{aligned} \quad (3.5.17)$$

$$\begin{aligned} &= \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{e^{(-s + \rho_j)t_f}}{s - \rho_j} + \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{1}{s - \rho_j} \\ &= \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{e^{(-s + \rho_j)t_f} - 1}{-s + \rho_j}. \end{aligned} \quad (3.5.18)$$

Thus, $\mathbf{G}(-\lambda_k) = \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j}$, which proves (3.5.13). To prove (3.5.14), we first differentiate (3.5.18) with respect to s to obtain

$$\mathbf{G}'(s) = \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{t_f(s - \rho_j)e^{(-s + \rho_j)t_f} + e^{(-s + \rho_j)t_f} - 1}{(s - \rho_j)^2}.$$

Plugging in $s = -\rho_k$ in this last expression yields the desired result (3.5.14). The proofs of (3.5.14) and (3.5.16) follow analogously. Recall $\mathbf{G}_r(s) = -e^{-st_f} \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1} e^{\mathbf{A}_r t_f} \mathbf{B}_r +$

$\mathbf{H}_r(s)$. Rewrite

$$\mathbf{G}_r(s) = -e^{-st_f} \sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{e^{\lambda_j t_f}}{s - \lambda_j} + \sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{1}{s - \lambda_j} \quad (3.5.19)$$

$$\begin{aligned} &= \sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{e^{(-s+\lambda_j)t_f}}{s - \lambda_j} + \sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{1}{s - \lambda_j} \\ &= \sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{e^{(-s+\lambda_j)t_f} - 1}{-s + \lambda_j}. \end{aligned} \quad (3.5.20)$$

Thus, we have proven (3.5.15). Differentiating (3.5.20) with respect to s we get

$$\mathbf{G}'_r(s) = \sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{t_f(s - \lambda_j)e^{(-s+\lambda_j)t_f} + e^{(-s+\lambda_j)t_f} - 1}{(s - \lambda_j)^2}. \quad (3.5.21)$$

Substituting λ_k for s in (3.5.21), we get (3.5.16) and prove the lemma. $\square$

Theorem 3.10. *Let $\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ be the transfer function of the full-order model with the pole-residue representation as defined in (3.5.9). Let $\mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{I}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r$ with pole residue representation as defined in (3.5.11) be the transfer function of an r^{th} order optimal approximation of $\mathbf{H}(s)$ with respect to the $\mathcal{H}_2(t_f)$ norm. For a finite-time horizon $[0, t_f]$, define $\mathbf{G}(s)$ as in (3.5.10) and $\mathbf{G}_r(s)$ as in (3.5.12). Then, for $k = 1, 2, \dots, r$,*

$$\ell_k^T \mathbf{G}(-\lambda_k) = \ell_k^T \mathbf{G}_r(-\lambda_k), \quad (3.5.22)$$

$$\mathbf{G}(-\lambda_k) \mathbf{r}_k = \mathbf{G}_r(-\lambda_k) \mathbf{r}_k, \quad \text{and} \quad (3.5.23)$$

$$\ell_k^T \mathbf{G}'(-\lambda_k) \mathbf{r}_k = \ell_k^T \mathbf{G}'_r(-\lambda_k) \mathbf{r}_k. \quad (3.5.24)$$

Proof. As mentioned above, let $\mathbf{J}$ denote the $\mathcal{H}_2(t_f)$ error norm square, i.e.,

$$\begin{aligned} \mathbf{J} &= \|\mathbf{h} - \mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2 \\ &= \|\mathbf{h}\|_{\mathcal{H}_2(t_f)}^2 - 2\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} + \|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2. \end{aligned}$$

The expressions for $\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)}$ and $\|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2$ in terms of the optimization variables $\{\lambda_k\}$, $\{\mathbf{r}_k\}$, and $\{\boldsymbol{\ell}_k\}$, for $k = 1, 2, \dots, r$, were already derived in Lemma (3.8). To make the gradient computations with respect to the k th parameter more clear, we separate the k -th term from these expressions. For example, we write $\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)}$ in (3.5.6) as

$$\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} = \sum_{j=1}^n \boldsymbol{\ell}_k^T \mathbf{c}_j \mathbf{b}_j^T \mathbf{r}_k \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} + \sum_{j=1}^n \sum_{\substack{i=1 \\ i \neq k}}^r \boldsymbol{\ell}_i^T \mathbf{c}_j \mathbf{b}_j^T \mathbf{r}_i \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j}.$$

Following the same procedure for $\|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2$, we obtain

$$\begin{aligned} \mathbf{J} = & \int_0^{t_f} \mathbf{h}(t)^T \mathbf{h}(t) dt - 2 \left(\sum_{j=1}^n \boldsymbol{\ell}_k^T \mathbf{c}_j \mathbf{b}_j^T \mathbf{r}_k \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} + \sum_{j=1}^n \sum_{\substack{i=1 \\ i \neq k}}^r \boldsymbol{\ell}_i^T \mathbf{c}_j \mathbf{b}_j^T \mathbf{r}_i \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j} \right) \\ & + \boldsymbol{\ell}_k^T \mathbf{r}_k \mathbf{r}_k^T \frac{e^{(2\lambda_k)t_f} - 1}{2\lambda_k} + \sum_{\substack{i=1 \\ i \neq k}}^r \boldsymbol{\ell}_i^T \mathbf{r}_k \mathbf{r}_i^T \frac{e^{(\lambda_i + \lambda_k)t_f} - 1}{\lambda_i + \lambda_k} \\ & + \sum_{\substack{j=1 \\ j \neq k}}^r \boldsymbol{\ell}_k^T \mathbf{r}_j \mathbf{r}_j^T \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} + \sum_{\substack{j=1 \\ j \neq k}}^r \sum_{\substack{i=1 \\ i \neq k}}^r \boldsymbol{\ell}_i^T \mathbf{r}_j \mathbf{r}_i^T \frac{e^{(\lambda_i + \lambda_j)t_f} - 1}{\lambda_i + \lambda_j}. \end{aligned} \quad (3.5.25)$$

To compute the gradient of the cost function $\mathbf{J}$ we perturb the cost functional with respect to the residue directions, i.e., $\boldsymbol{\ell}_k \rightarrow \boldsymbol{\ell}_k + \Delta \boldsymbol{\ell}_k$ and $\mathbf{r}_k \rightarrow \mathbf{r}_k + \Delta \mathbf{r}_k$:

$$\begin{aligned} \tilde{\mathbf{J}}_k = & \int_0^{t_f} \mathbf{h}(t)^T \mathbf{h}(t) dt - 2 \left(\sum_{j=1}^n (\boldsymbol{\ell}_k + \Delta \boldsymbol{\ell}_k)^T \mathbf{c}_j \mathbf{b}_j^T (\mathbf{r}_k + \Delta \mathbf{r}_k) \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} \right. \\ & + \sum_{j=1}^n \sum_{\substack{i=1 \\ i \neq k}}^r \boldsymbol{\ell}_i^T \mathbf{c}_j \mathbf{b}_j^T \mathbf{r}_i \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j} \Big) \\ & + (\boldsymbol{\ell}_k + \Delta \boldsymbol{\ell}_k)^T (\boldsymbol{\ell}_k + \Delta \boldsymbol{\ell}_k) (\mathbf{r}_k + \Delta \mathbf{r}_k)^T (\mathbf{r}_k + \Delta \mathbf{r}_k) \frac{e^{(2\lambda_k)t_f} - 1}{2\lambda_k} \\ & + \sum_{\substack{i=1 \\ i \neq k}}^r \boldsymbol{\ell}_i^T (\boldsymbol{\ell}_k + \Delta \boldsymbol{\ell}_k) (\mathbf{r}_k + \Delta \mathbf{r}_k)^T \mathbf{r}_i \frac{e^{(\lambda_i + \lambda_k)t_f} - 1}{\lambda_i + \lambda_k} \\ & + \sum_{\substack{j=1 \\ j \neq k}}^r (\boldsymbol{\ell}_k + \Delta \boldsymbol{\ell}_k)^T \mathbf{r}_j \mathbf{r}_j^T (\mathbf{r}_k + \Delta \mathbf{r}_k) \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} + \sum_{\substack{j=1 \\ j \neq k}}^r \sum_{\substack{i=1 \\ i \neq k}}^r \boldsymbol{\ell}_i^T \mathbf{r}_j \mathbf{r}_i^T \frac{e^{(\lambda_i + \lambda_j)t_f} - 1}{\lambda_i + \lambda_j}. \end{aligned}$$

Then, collecting the terms that are multiplied by $\Delta \ell_k$ and $\Delta \mathbf{r}_k$, we obtain

$$\begin{aligned}\nabla_{\mathbf{r}_k} \mathbf{J} &= -2\ell_k^T \sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} + 2\ell_k^T \sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} \text{ and} \\ \nabla_{\ell_k} \mathbf{J} &= -2 \left(\sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} \right) \mathbf{r}_k + 2 \left(\sum_{j=1}^r \ell_j \mathbf{r}_j^T \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} \right) \mathbf{r}_k.\end{aligned}$$

Setting $\nabla_{\mathbf{r}_k} \mathbf{J} = 0$ and $\nabla_{\ell_k} \mathbf{J} = 0$, and using Lemma 3.9, mainly (3.5.13) and (3.5.15), yield

$$\begin{aligned}\ell_k^T \mathbf{G}(-\lambda_k) &= \ell_k^T \mathbf{G}_r(-\lambda_k), \text{ and} \\ \mathbf{G}(-\lambda_k) \mathbf{r}_k &= \mathbf{G}_r(-\lambda_k) \mathbf{r}_k,\end{aligned}$$

which proves (3.5.22) and (3.5.23). To prove (3.5.24), we differentiate $\mathbf{J}$ in (3.5.25) with respect to the k -th pole λ_k . Note that we have written $\mathbf{J}$ in such a way to isolate the terms that depend on λ_k from the ones that do not. Thus, many of the terms in (3.5.25) have a zero derivative and we obtain:

$$\begin{aligned}\frac{\partial \mathbf{J}}{\partial \lambda_k} &= -2\ell_k^T \left(\sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{t_f(\lambda_k + \rho_j)e^{(\rho_j + \lambda_k)t_f}}{(\lambda_k + \rho_j)^2} - \frac{e^{(\rho_j + \lambda_k)t_f} - 1}{(\lambda_k + \rho_j)^2} \right) \mathbf{r}_k \\ &\quad + 2\ell_k^T \left(\sum_{i=1}^r \ell_i \mathbf{r}_i^T \frac{t_f(\lambda_i + \lambda_k)e^{(\lambda_i + \lambda_k)t_f}}{(\lambda_i + \lambda_k)^2} - \frac{e^{(\lambda_i + \lambda_k)t_f} - 1}{(\lambda_i + \lambda_k)^2} \right) \mathbf{r}_k.\end{aligned}\tag{3.5.26}$$

Note that the first term in (3.5.26) corresponds to the derivative of the second term in (3.5.25) and the second term in (3.5.26) corresponds to the derivative of the last four terms in (3.5.25). We rewrite (3.5.26) to obtain

$$\frac{\partial \mathbf{J}}{\partial \lambda_k} = -2K_1 + 2K_2,\tag{3.5.27}$$

where

$$K_1 = \ell_k^T \left(\sum_{j=1}^n \mathbf{c}_j \mathbf{b}_j^T \frac{(t_f(\lambda_k + \rho_j) - 1)e^{(\rho_j + \lambda_k)t_f} + 1}{(\lambda_k + \rho_j)^2} \right) \mathbf{r}_k\tag{3.5.28}$$

and

$$K_2 = \ell_k^T \left(\sum_{i=1}^r \ell_i \mathbf{r}_i^T \frac{(t_f(\lambda_i + \lambda_k) - 1)e^{(\lambda_i + \lambda_k)t_f} + 1}{(\lambda_i + \lambda_k)^2} \right) \mathbf{r}_k. \quad (3.5.29)$$

Lemma 3.9, specifically (3.5.14) and (3.5.16), shows that the expressions in the parentheses in (3.5.28) and (3.5.29) are, respectively, $-\mathbf{G}'(-\lambda_k)$ and $-\mathbf{G}'_r(-\lambda_k)$. If $\frac{\partial \mathbf{J}}{\partial \lambda_k} = 0$, then

$$\ell_k^T \mathbf{G}'(-\lambda_k) \mathbf{r}_k = \ell_k^T \mathbf{G}'_r(-\lambda_k) \mathbf{r}_k,$$

which completes the proof. $\square$

Remark 3.11. We note that the interval of interest is problem dependent and the choice of the interval, i.e., the choice of t_f , may depend on the model. In some cases, we pick the final simulation time arbitrarily. In other cases, the final time can be chosen based on how fast the system is decaying. If we wait too long before ending the simulation, i.e., we pick a relatively large value for t_f , FHIRKA essentially reduces to IRKA, since the exponential term virtually vanishes. However, these optimality conditions hold for any choice of $t_f > 0$. If we let $t_f \rightarrow \infty$, we recover the optimality conditions in Theorem 2.8.

Remark 3.12. In the infinite-horizon case, if $\mathbf{H}_r(s)$ is the best $\mathcal{H}_2$ approximation to $\mathbf{H}(s)$, then $\mathbf{H}_r(s)$ interpolates $\mathbf{H}(s)$. However, in the finite-horizon case, the interpolant is $\mathbf{G}_r(s)$, and the interpolated function is $\mathbf{G}(s)$; thus, $\mathbf{H}_r(s)$ does not interpolate $\mathbf{H}(s)$. To give more intuition about these resulting interpolation conditions, consider the time-limited function $\mathbf{g}(t)$ such that

$$\mathbf{g}(t) = \begin{cases} \mathbf{h}(t), & \text{for } t < t_f; \\ 0, & \text{for } t \geq t_f. \end{cases}$$

A direct calculation shows that $\mathbf{G}(s)$ is the Laplace transform of $\mathbf{g}(t)$:

$$\begin{aligned}
\mathcal{L}^{-1}\{\mathbf{G}(s)\} &= \mathbf{h}(t) + \sum_{i=1}^n u_{t_f}(t) \mathbf{c}_i \mathbf{b}_i^T e^{\lambda_i t_f} e^{\lambda_i(t-t_f)} \\
&= \mathbf{h}(t) + \sum_{i=1}^n u_{t_f}(t) \mathbf{c}_i \mathbf{b}_i^T e^{\lambda_i t},
\end{aligned}$$

where $u_{t_f}(t)$ is the unit step function

$$u_{t_f}(t) = \begin{cases} 0, & t < t_f; \\ 1, & t \geq t_f. \end{cases}$$

Similarly let $\mathbf{g}_r(t)$ denote the time-limited version of $\mathbf{h}_r(t)$. Then its Laplace transform is $\mathbf{G}_r(s)$. Therefore, the optimality conditions (3.5.22)–(3.5.24) correspond to optimal interpolation of $\mathbf{G}(s)$ (Laplace transform of the time-limited function $\mathbf{g}(t)$) by $\mathbf{G}_r(s)$ (Laplace transform of the time limited function $\mathbf{g}_r(t)$). The fact that $\mathbf{g}(t)$ and $\mathbf{g}_r(t)$ are both time-limited is the precise reason why we cannot simply apply $\mathcal{H}_2$ optimal reduction to $\mathbf{G}(s)$. The method of [24], called TF-IRKA, does not require the original function to be a rational function. Thus, in principle we can use TF-IRKA to reduce $\mathbf{G}(s)$. However, the resulting reduced model is a rational function without any structure. In our case, the reduced model $\mathbf{G}_r(s)$ needs to retain the same structure as $\mathbf{G}(s)$ so that we can extract an $\mathbf{H}_r(s)$. In other words, if we simply apply an $\mathcal{H}_2$ optimal algorithm to $\mathbf{G}(s)$, we would be approximating a finite horizon model by an infinite horizon one and we cannot extract $\mathbf{H}_r(s)$. Therefore, a new algorithmic framework is needed, as we discuss in more detail in Chapter 4.

Remark 3.13. For an unstable dynamical system without any purely imaginary eigenvalues, one can work with the $\mathcal{L}_2$ norm by decomposing it into a stable and anti-stable system, and then obtain an interpolatory reduced model based on this measure, as we discussed in Subsection 2.6.1. However, this solution requires destroying the causality of the underlying dynamics [134]. Working with a finite-time interval allows us to reduce unstable models

while preserving the causality of the system.

3.5.1 Implication of the interpolatory $\mathcal{H}_2(t_f)$ optimality conditions

Theorem 3.10 extends the interpolatory infinite-horizon $\mathcal{H}_2$ optimality conditions in Theorem 2.8 to the finite-horizon case. Note that in the case of asymptotically stable dynamical systems, if we let $t_f \rightarrow \infty$, we recover the infinite-horizon conditions in Theorem 2.8. The major difference from the regular $\mathcal{H}_2$ problem is that optimality no longer requires that the reduced model $\mathbf{H}_r(s)$ tangentially interpolates the full model $\mathbf{H}(s)$. Instead, the auxiliary reduced-order function $\mathbf{G}_r(s)$ should be a tangential Hermite interpolant to the auxiliary full-order function $\mathbf{G}(s)$. However, the optimal interpolation points and the optimal tangential directions still result from the pole-residue representation of the reduced-order transfer function $\mathbf{H}_r(s)$. This situation is similar to the interpolatory optimality conditions for the frequency-weighted $\mathcal{H}_2$ -optimal model reduction problem in which one tries to minimize a weighted $\mathcal{H}_2$ norm in the frequency domain, i.e., find $\mathbf{H}_r(s)$ that minimizes $\|\mathbf{W}(\mathbf{H} - \mathbf{H}_r)\|_{\mathcal{H}_2}$ where $\mathbf{W}(s)$ represents a weighting function in the frequency domain. As [43] showed, the optimality in the frequency-weighted $\mathcal{H}_2$ -norm requires that a function of $\mathbf{H}_r(s)$ tangentially interpolates a function of $\mathbf{H}(s)$. Despite this conceptual similarity, the resulting interpolation conditions are drastically different from what we obtained here, as one would expect due to the different measures. For details, we refer the reader to [43].

As we pointed out in Section 3.4, in addition to the interpolatory framework, one can represent the $\mathcal{H}_2$ optimality conditions in terms of Sylvester equations, leading to a projection framework for the reduced model. This means that given the full-model $\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, one constructs two bases $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times r}$ with $\mathbf{V}^T \mathbf{W} = \mathbf{I}_r$ such that the reduced-order quantities are obtained via projection, i.e.,

$$\mathbf{A}_r = \mathbf{W}^T \mathbf{A} \mathbf{V}, \quad \mathbf{B}_r = \mathbf{W}^T \mathbf{B}, \quad \text{and} \quad \mathbf{C}_r = \mathbf{C} \mathbf{V}. \quad (3.5.30)$$

In the infinite-horizon case, [181] showed that the optimal $\mathcal{H}_2$ reduced model is indeed guaranteed to be obtained via projection. As discussed in Section 3.4, recently, [82] established the Sylvester-equation based optimality conditions for the time-limited $\mathcal{H}_2$ model reduction problem; i.e., they extended the Wilson framework to the time-limited (finite-horizon) $\mathcal{H}_2$ problem. Furthermore, [82] developed a projection-based IRKA-type numerical algorithm to construct the reduced models. However, as the authors point out in [82], even though their algorithm yields high-fidelity reduced models in terms of the $\mathcal{H}_2(t_f)$ measure, the resulting reduced model satisfies the optimality conditions only approximately. This is not surprising in light of the optimality conditions we derived here. Since the optimality requires that $\mathbf{G}_r(s)$ should interpolate $\mathbf{G}(s)$ (as opposed to $\mathbf{H}_r(s)$ interpolating $\mathbf{H}(s)$), unlike in the infinite-horizon case, the reduced model in the finite-horizon case is *not* necessarily given by a projection as in (3.5.30). Therefore, a projection-based approach would satisfy the optimality conditions only approximately. This was also the case in [43] where even though a projection-based IRKA-like algorithm produced high-fidelity reduced models in the weighted norm, it satisfied the optimality conditions approximately.

The advantage of the interpolation framework and the parametrization (3.5.3) we consider here is that we do not require the reduced-model to be obtained via projection. By treating the poles and residues in (3.5.3) as the parameters and directly working with them, we can obtain a reduced model to satisfy the optimality conditions exactly.

Remark 3.14. The finite-horizon approximation problem for *discrete-time dynamical systems* has been considered in [137]. The derivation in [137], however, allows the reduced-model quantities to vary at every time-step, thus using a time-varying reduced model, as opposed to the time-invariant formulation considered here and in [82]. Allowing time-varying quantities drastically simplifies the gradient computations, leading to a recurrence relations for optimality. Even though at first glance discrete finite horizon linear time-varying (LTV) systems may appear more complicated, the derivation of the optimality conditions is straightforward due to the aforementioned gradient computations. Even though $\mathcal{H}_2(t_f)$

optimal model reduction for continuous finite horizon linear time invariant (LTI) models appears less complex, it requires dealing with matrix exponentials which arise due to the finite horizon integrals. The model reduction problem for finite-horizon $\mathcal{H}_2$ approximation for *time-invariant* discrete-time dynamical systems is still an open question.

Chapter 4

Algorithmic Developments for $\mathcal{H}_2(t_f)$

Model Reduction

In this chapter we discuss pole- residue based $\mathcal{H}_2(t_f)$ optimality conditions for single-input/single-output (SISO) systems. We also derive a descent algorithm (FHIRKA) which generates a reduced model that satisfies the $\mathcal{H}_2(t_f)$ optimality conditions. The numerical results obtained further support our theoretical results.

4.1 $\mathcal{H}_2(t_f)$ Optimality Conditions: SISO Case

Remark 4.1. The optimality conditions for SISO systems follow directly from the conditions established in Chapter 3 for the multi-input/multi-output case. However, we rederive the optimality conditions for the reader who is not interested in the theoretical framework for the MIMO case. Furthermore, the SISO derivations are easier to follow than the MIMO ones, and consequently, they are helpful in understanding the essence of these results for the reader who is not familiar with concepts such as tangential interpolation. The reader who is comfortable with the interpolation conditions in the MIMO case, may skip the proofs of the Lemmas 4.2, 4.3, 4.5 and of Theorem 4.6.

Consider the SISO linear dynamical system:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{c}^T \mathbf{x}(t),\end{aligned}\tag{4.1.1}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, and $\mathbf{c} \in \mathbb{R}^n$. We also have the state $\mathbf{x}(t) : \mathbb{R} \mapsto \mathbb{R}^n$, the input $\mathbf{u}(t) : \mathbb{R} \mapsto \mathbb{R}$, and the output $\mathbf{y}(t) : \mathbb{R} \mapsto \mathbb{R}$. The reduced order model is

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{b}_r \mathbf{u}(t) \\ \mathbf{y}_r(t) &= \mathbf{c}_r^T \mathbf{x}_r(t),\end{aligned}\tag{4.1.2}$$

where $\mathbf{A}_r \in \mathbb{R}^{r \times r}$, $\mathbf{b}_r \in \mathbb{R}^r$, and $\mathbf{c}_r \in \mathbb{R}^r$ with $r \ll n$. The state $\mathbf{x}_r(t) : \mathbb{R} \mapsto \mathbb{R}^r$ and the dimensions of the input $\mathbf{u}(t)$ and output $\mathbf{y}_r(t)$ remain unchanged. While SISO systems can be considered a special case of MIMO systems, analyzing the SISO case on its own may reveal some nuance that is specific to the SISO systems as well as shed light on the choices we make when constructing a finite horizon algorithm. Let

$$\mathbf{h}(t) = \mathbf{c}^T e^{\mathbf{A}t} \mathbf{b} = \sum_{k=1}^n \psi_k e^{\rho_k t}\tag{4.1.3}$$

be the impulse response of a dynamical system of order n where $\psi_k \in \mathbb{R}$ and $\rho_k \in \mathbb{C}$ are the residues and poles of the transfer function, respectively. We aim to produce a locally optimal approximant under the $\mathcal{H}_2(t_f)$ norm that has the following impulse response:

$$\mathbf{h}_r(t) = \mathbf{c}_r^T e^{\mathbf{A}_r t} \mathbf{b}_r = \sum_{i=1}^r \phi_i e^{\lambda_i t},\tag{4.1.4}$$

where $\mathbf{b}_r, \mathbf{c}_r \in \mathbb{R}^r$, $\phi \in \mathbb{R}$, and $\lambda \in \mathbb{C}$. The residues ϕ are scalars since they are obtained by the inner product $\mathbf{c}_r^T \mathbf{b}_r$.

Using definition 3.2 of the $\mathcal{H}_2(t_f)$ norm, we can measure the error between the full and

reduced models. It follows that

$$\|\mathbf{h} - \mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2 = \|\mathbf{h}\|_{\mathcal{H}_2(t_f)}^2 - \langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} + \|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2. \quad (4.1.5)$$

In order to minimize the expression in (4.1.5), we need to differentiate with respect to $\mathbf{h}_r$. Since $\mathbf{h}_r$ is determined by the poles and the residues of the reduced model, we write (4.1.5) in terms of these poles and residues. The impulse response of the full system $\mathbf{h}$ is clearly constant with respect to the poles and residues of the reduced system; hence, it does not affect the minimum of the error. For this reason, in the following lemmas, we deal only with the last two terms in (4.1.5).

Lemma 4.2. *Let*

$$\mathbf{h}(t) = \mathbf{c}^T e^{\mathbf{A}t} \mathbf{b} = \sum_{j=1}^n \psi_j e^{\rho_j t}$$

and

$$\mathbf{h}_r(t) = \mathbf{c}_r^T e^{\mathbf{A}_r t} \mathbf{b}_r = \sum_{i=1}^r \phi_i e^{\lambda_i t},$$

where the ρ_j 's and the ψ_j 's are respectively the poles and residues of the full system, and λ_i 's and ϕ_i 's are respectively the poles and residues of the reduced system. Then,

$$\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} = \sum_{j=1}^n \phi_k \psi_j \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} + \sum_{j=1}^n \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \psi_j \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j}. \quad (4.1.6)$$

Proof. From Definition 3.2, we have

$$\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} = \int_0^{t_f} \mathbf{h}(t) \cdot \mathbf{h}_r(t) dt.$$

As a result,

$$\begin{aligned}
\langle \mathbf{h}, \mathbf{h}_r \rangle_{\mathcal{H}_2(t_f)} &= \int_0^{t_f} \mathbf{h}(t) \sum_{i=1}^r \phi_i e^{\lambda_i t} dt \\
&= \int_0^{t_f} \sum_{i=1}^r \phi_i e^{\lambda_i t} \sum_{j=1}^n \psi_j e^{\rho_j t} dt \\
&= \sum_{i=1}^r \phi_i \int_0^{t_f} \sum_{j=1}^n \psi_j e^{\rho_j t} e^{\lambda_i t} dt \\
&= \sum_{i=1}^r \phi_i \sum_{j=1}^n \psi_j \int_0^{t_f} e^{\rho_j t} e^{\lambda_i t} dt \\
&= \sum_{i=1}^r \phi_i \sum_{j=1}^n \psi_j \frac{e^{(\rho_j + \lambda_i)t_f} - 1}{\lambda_i + \rho_j} \\
&= \sum_{j=1}^n \phi_k \psi_j \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} + \sum_{j=1}^n \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \psi_j \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j}.
\end{aligned}$$

□

Lemma 4.3. *If $\mathbf{h}_r(t) = \sum_{i=1}^r \phi_i e^{\lambda_i t}$ is the impulse response of the reduced model, then*

$$\begin{aligned}
\|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2 &= \phi_k^2 \frac{e^{(2\lambda_k)t_f} - 1}{2\lambda_k} + \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_k \frac{e^{(\lambda_i + \lambda_k)t_f} - 1}{\lambda_i + \lambda_k} \\
&\quad + \sum_{\substack{j=1 \\ j \neq k}}^r \phi_j \phi_k \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} + \sum_{\substack{j=1 \\ j \neq k}}^r \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_j \frac{e^{(\lambda_i + \lambda_j)t_f} - 1}{\lambda_i + \lambda_j}.
\end{aligned}$$

Proof. From Definition 3.2 we have

$$\begin{aligned}
\|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2 &= \int_0^{t_f} \left(\sum_{i=1}^r \phi_i e^{\lambda_i t} \right)^2 dt \\
&= \int_0^{t_f} \sum_{i=1}^r \phi_i^2 e^{2\lambda_i t} dt + 2 \int_0^{t_f} \sum_{i < m} \phi_i \phi_m e^{(\lambda_i + \lambda_m)t} dt \\
&= \int_0^{t_f} \phi_k^2 e^{(2\lambda_k)t} + \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_k e^{(\lambda_i + \lambda_k)t} + \sum_{\substack{j=1 \\ j \neq k}}^r \phi_j \phi_k e^{(\lambda_k + \lambda_j)t} + \sum_{\substack{j=1 \\ j \neq k}}^r \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_j e^{(\lambda_i + \lambda_j)t} dt \\
&= \phi_k^2 \frac{e^{(2\lambda_k)t_f} - 1}{2\lambda_k} + \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_k \frac{e^{(\lambda_i + \lambda_k)t_f} - 1}{\lambda_i + \lambda_k} \\
&\quad + \sum_{\substack{j=1 \\ j \neq k}}^r \phi_j \phi_k \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} + \sum_{\substack{j=1 \\ j \neq k}}^r \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_j \frac{e^{(\lambda_i + \lambda_j)t_f} - 1}{\lambda_i + \lambda_j}.
\end{aligned}$$

□

Remark 4.4. The k -th index is no different than any other index. We only isolate the k -th index to clarify later derivations. In Lemma 4.3, if we add up all the individual sums in the right-hand side, we obtain the usual summation over all the indices.

For an infinite horizon, Corollary 2.9 tells us that a locally $\mathcal{H}_2$ optimal reduced transfer function fully interpolates the transfer function of the original system at the mirror images of the poles of the reduced model. In the finite horizon case, we obtain an equivalent result for the SISO system, even though the full and reduced transfer functions are not interpolants of each other at the mirror images of the poles. The following lemma is essential in proving the interpolation conditions for the finite horizon case.

Lemma 4.5. Let $\mathbf{G}(s) = -e^{-st_f} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{b} + \mathbf{H}(s)$. Then,

$$\mathbf{G}(-\lambda_j) = \int_0^{t_f} \mathbf{h}(t) e^{\lambda_j t} dt. \quad (4.1.7)$$

Proof.

$$\begin{aligned}
\int_0^{t_f} \mathbf{h}(t) e^{\lambda_j t} dt &= \int_0^{t_f} e^{\lambda_j t} \sum_{k=1}^n \psi_k e^{\rho_k t} dt \\
&= \int_0^{t_f} \sum_{k=1}^n \psi_k e^{(\rho_k + \lambda_j) t} dt \\
&= \sum_{k=1}^n \psi_k \frac{e^{(\rho_k + \lambda_j) t_f} - 1}{\rho_k + \lambda_j} \\
&= -e^{\lambda_j t_f} \sum_{k=1}^n \frac{\psi_k e^{\rho_k t_f}}{-\lambda_j - \rho_k} + \sum_{k=1}^n \frac{\psi_k}{-\lambda_j - \rho_k} \\
&= -e^{\lambda_j t_f} \mathbf{c}^T ((-\lambda_j) \mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A} t_f} \mathbf{b} + \mathbf{H}(-\lambda_j) \\
&= \mathbf{G}(-\lambda_j).
\end{aligned}$$

□

Writing the relevant terms of the error in (4.1.5) in terms of the poles and residues of the reduced system enable us to differentiate the error. This representation of the error is essential in proving the following theorem, which establishes the necessary optimality conditions with respect to the $\mathcal{H}_2$ norm.

Theorem 4.6. *Let $\mathbf{G}(s) = -e^{-st_f} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A} t_f} \mathbf{b} + \mathbf{H}(s)$ and $\mathbf{G}_r(s) = -e^{-st_f} \mathbf{c}_r^T (s\mathbf{I}_r - \mathbf{A}_r)^{-1} e^{\mathbf{A}_r t_f} \mathbf{b}_r + \mathbf{H}_r(s)$. If $\mathbf{H}_r$ is the best r^{th} order approximation of $\mathbf{H}$ with respect to the $\mathcal{H}_2(t_f)$ norm, then*

$$\mathbf{G}(-\lambda_k) = \mathbf{G}_r(-\lambda_k) \text{ and}$$

$$\mathbf{G}'(-\lambda_k) = \mathbf{G}_r'(-\lambda_k),$$

where λ_k 's for $k = 1, 2, \dots, r$ are the poles of the reduced system.

Proof. Let $\mathbf{J}$ be the square $\mathcal{H}_2(t_f)$ error between the full and reduced models, i.e.,

$$\mathbf{J} = \|\mathbf{h} - \mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2.$$

From Definition 3.2, Lemma 4.2, and Lemma 4.3, we infer

$$\begin{aligned} \mathbf{J} = & \int_0^{t_f} \mathbf{h}(t)^T \mathbf{h}(t) dt - 2 \left(\sum_{j=1}^n \phi_k \psi_j \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} + \sum_{j=1}^n \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \psi_j \frac{e^{(\lambda_i + \rho_j)t_f} - 1}{\lambda_i + \rho_j} \right) + \phi_k^2 \frac{e^{(2\lambda_k)t_f} - 1}{2\lambda_k} \\ & + \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_k \frac{e^{(\lambda_i + \lambda_k)t_f} - 1}{\lambda_i + \lambda_k} + \sum_{\substack{j=1 \\ j \neq k}}^r \phi_j \phi_k \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} + \sum_{\substack{j=1 \\ j \neq k}}^r \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_j \frac{e^{(\lambda_i + \lambda_j)t_f} - 1}{\lambda_i + \lambda_j}. \end{aligned} \quad (4.1.8)$$

If we differentiate the error $\mathbf{J}$ with respect to the k -th residue ϕ_k , and set it equal to 0, we have

$$\begin{aligned} \frac{\partial \mathbf{J}}{\partial \phi_k} &= -2 \sum_{j=1}^n \psi_j \frac{e^{(\lambda_k + \rho_j)t_f} - 1}{\lambda_k + \rho_j} + 2 \sum_{j=1}^r \phi_j \frac{e^{(\lambda_k + \lambda_j)t_f} - 1}{\lambda_k + \lambda_j} \\ &= 0. \end{aligned}$$

This implies

$$\sum_{i=1}^r \phi_i \frac{e^{(\lambda_i + \lambda_k)t_f} - 1}{\lambda_i + \lambda_k} = \int_0^{t_f} \mathbf{h}(t) e^{\lambda_k t} dt. \quad (4.1.9)$$

Equation (4.1.9) and Lemma 4.5 imply

$$\mathbf{G}(-\lambda_k) = \mathbf{G}_r(-\lambda_k).$$

If we differentiate the error $\mathbf{J}$ with respect to the k -th pole λ_k , and set it equal to 0, we have

$$\begin{aligned} \frac{\partial \mathbf{J}}{\partial \lambda_k} &= -2\phi_k \sum_{j=1}^n \psi_j \frac{t_f(\lambda_k + \rho_j)e^{(\rho_j + \lambda_k)t_f} - e^{(\rho_j + \lambda_k)t_f} + 1}{(\lambda_k + \rho_j)^2} + \phi_k^2 \frac{4t_f \lambda_k e^{2\lambda_k t_f} - 2e^{2\lambda_k t_f} + 2}{4\lambda_k^2} \\ &\quad + 2 \sum_{\substack{i=1 \\ i \neq k}}^r \phi_i \phi_k \frac{t_f(\lambda_i + \lambda_k)e^{(\lambda_i + \lambda_k)t_f} - e^{(\lambda_i + \lambda_k)t_f} + 1}{(\lambda_i + \lambda_k)^2}. \\ &= -2\phi_k \sum_{j=1}^n \psi_j \frac{t_f(\lambda_k + \rho_j)e^{(\rho_j + \lambda_k)t_f} - e^{(\rho_j + \lambda_k)t_f} + 1}{(\lambda_k + \rho_j)^2} \\ &\quad + 2\phi_k \sum_{i=1}^r \phi_i \frac{t_f(\lambda_i + \lambda_k)e^{(\lambda_i + \lambda_k)t_f} - e^{(\lambda_i + \lambda_k)t_f} + 1}{(\lambda_i + \lambda_k)^2}. \end{aligned}$$

Thus, if $\phi_k \neq 0$, then, we have

$$\sum_{j=1}^n \psi_j \frac{t_f(\lambda_k + \rho_j)e^{(\rho_j + \lambda_k)t_f} - e^{(\rho_j + \lambda_k)t_f} + 1}{(\lambda_k + \rho_j)^2} = \sum_{i=1}^r \phi_i \frac{t_f(\lambda_i + \lambda_k)e^{(\lambda_i + \lambda_k)t_f} - e^{(\lambda_i + \lambda_k)t_f} + 1}{(\lambda_i + \lambda_k)^2}. \quad (4.1.10)$$

Consider

$$\begin{aligned} \mathbf{G}(s) &= -e^{-st_f} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{b} + \mathbf{H}(s) \\ &= -e^{-st_f} \sum_{j=1}^n \frac{\psi_j e^{\rho_j t_f}}{s - \rho_j} + \sum_{j=1}^n \frac{\psi_j}{s - \rho_j} \\ &= -\sum_{j=1}^n \frac{\psi_j e^{(-s + \rho_j)t_f}}{s - \rho_j} + \sum_{j=1}^n \frac{\psi_j}{s - \rho_j} \end{aligned}$$

and

$$\begin{aligned}
\mathbf{G}_r(s) &= -e^{-st_f} \mathbf{c}_r^T (s\mathbf{I}_r - \mathbf{A}_r)^{-1} e^{\mathbf{A}_r t_f} \mathbf{b}_r + \mathbf{H}_r(s) \\
&= -e^{-st_f} \sum_{i=1}^r \frac{\phi_i e^{\lambda_i t_f}}{s - \lambda_i} + \sum_{i=1}^r \frac{\phi_i}{s - \lambda_i} \\
&= -\sum_{i=1}^r \frac{\phi_i e^{(-s+\lambda_i)t_f}}{s - \lambda_i} + \sum_{i=1}^r \frac{\phi_i}{s - \lambda_i}.
\end{aligned}$$

If we differentiate $\mathbf{G}(s)$ with respect to s , we get

$$\begin{aligned}
\mathbf{G}'(s) &= -\sum_{j=1}^n \psi_j \frac{(-t_f)(s - \rho_j) e^{(-s+\rho_j)t_f} - e^{(-s+\rho_j)t_f}}{(s - \rho_j)^2} - \sum_{j=1}^n \frac{\psi_j}{(s - \rho_j)^2} \\
&= -\sum_{j=1}^n \psi_j \frac{(-t_f)(s - \rho_j) e^{(-s+\rho_j)t_f} - e^{(-s+\rho_j)t_f} + 1}{(s - \rho_j)^2} \\
&= \sum_{j=1}^n \psi_j \frac{t_f(s - \rho_j) e^{(-s+\rho_j)t_f} + e^{(-s+\rho_j)t_f} - 1}{(s - \rho_j)^2}.
\end{aligned}$$

Similarly, if we differentiate $\mathbf{G}_r(s)$ with respect to s , we get

$$\mathbf{G}'_r(s) = \sum_{i=1}^r \phi_i \frac{t_f(s - \lambda_i) e^{(-s+\lambda_i)t_f} + e^{(-s+\lambda_i)t_f} - 1}{(s - \lambda_i)^2}.$$

For any pole λ_k of the reduced system $\mathbf{H}_r$, the left side of (4.1.10) equals $-\mathbf{G}'(-\lambda_k)$ and the right side of (4.1.10) equals $-\mathbf{G}'_r(-\lambda_k)$. Hence, for any pole λ_k of the reduced system $\mathbf{H}_r$,

$$\mathbf{G}'(-\lambda_k) = \mathbf{G}'_r(-\lambda_k).$$

□

The following corollary deals with a specific case of Theorem 4.6. If we know the poles of the reduced system, we can establish the necessary and sufficient optimality conditions for the

residues. In other words, given the poles of a reduced system we can find the best residues so that we minimize the error between the full and reduced systems.

Corollary 4.7. *Let $\mathbf{H}(s)$ and $\mathbf{H}_r(s)$ be as given in (4.2.1), and $\mathbf{G}(s)$ and $\mathbf{G}_r(s)$ as defined in Theorem 4.6. Assume the reduced poles $\{\lambda_i\}_{i=1}^r$ are fixed. Then, $\mathbf{H}_r(s)$ is the best r^{th} order approximation of $\mathbf{H}(s)$ with respect to the $\mathcal{H}_2(t_f)$ norm if and only if $\mathbf{M}\boldsymbol{\phi} = \mathbf{z}$, or equivalently, $\mathbf{G}(-\lambda_k) = \mathbf{G}_r(-\lambda_k)$, for $k = 1, 2, \dots, r$, where $\boldsymbol{\phi} = [\phi_1 \ \phi_2 \ \dots \ \phi_r]^T \in \mathbb{C}^r$ is the vector of residues; $\mathbf{z} \in \mathbb{C}^r$ is the vector with entries*

$$\mathbf{z}_j = e^{\lambda_j t_f} \mathbf{c}^T (-\lambda_j \mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A} t_f} \mathbf{b} - \mathbf{H}(-\lambda_j), i = 1, 2, \dots, r;$$

and $\mathbf{M} \in \mathbb{C}^{r \times r}$ is the matrix with entries

$$\mathbf{M}_{i,j} = \frac{e^{(\lambda_i + \lambda_j)t_f} - 1}{\lambda_i + \lambda_j}, \quad \text{for } i, j = 1, 2, \dots, r.$$

Proof. Using the the results from Lemmas 4.2 and 4.3 and applying some algebraic manipulation, the cost functional $\mathbf{J}$ can be written as

$$\mathbf{J} = \|\mathbf{h}\|_{\mathcal{H}_2(t_f)}^2 - 2\boldsymbol{\phi}^T \mathbf{w} + \boldsymbol{\phi}^T \mathbf{M} \boldsymbol{\phi},$$

where $\mathbf{w} \in \mathbb{C}^{r \times 1}$ has the entries

$$\mathbf{w}_i = \sum_{k=1}^n \psi_k \frac{e^{(\rho_k + \lambda_i)t_f} - 1}{\lambda_i + \rho_k} \quad \text{for } i = 1, 2, \dots, r.$$

Note that $\mathbf{M}$ is positive definite, $\boldsymbol{\phi}^T \mathbf{M} \boldsymbol{\phi} = \|\mathbf{h}_r\|_{\mathcal{H}_2(t_f)}^2 > 0$, and the cost function is quadratic in $\boldsymbol{\phi}$. Thus, the (global) minimizer is obtained by solving $\mathbf{M}\boldsymbol{\phi} = \mathbf{z}$, which corresponds to rewriting $\mathbf{G}(-\lambda_k) = \mathbf{G}_r(-\lambda_k)$ for $k = 1, 2, \dots, r$ in a compact way. $\square$

The result is analogous to the regular infinite-horizon $\mathcal{H}_2$ problem where the Lagrange optimality becomes necessary and sufficient once the poles are fixed [24, 71]. What is important

here is that once the reduced poles are fixed, the best residues can be computed directly by solving an $r \times r$ linear system $\mathbf{M}\boldsymbol{\phi} = \mathbf{z}$. This is the property that we will exploit in the numerical scheme next.

Remark 4.8. If we let $t_f \rightarrow \infty$ we retrieve the conditions for the infinite-horizon case for SISO systems. In other words if t_f grows arbitrarily large, the reduced transfer function interpolates the full transfer function at the mirror images of the poles, which is expected.

Proposition 4.9. Let $\mathbf{G}(s) = -e^{-st_f} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{b} + \mathbf{H}(s)$ where $\mathbf{H}(s) = \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{b}$ is the transfer function of (3.1.1). Then, $\|\mathbf{G}\|_{\mathcal{H}_2(t_f)} = \|\mathbf{H}\|_{\mathcal{H}_2(t_f)}$.

Proof. Note

$$\begin{aligned} \mathbf{G}(s) &= -e^{-st_f} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{b} + \mathbf{H}(s) \\ &= -e^{-st_f} \sum_{i=1}^n \frac{\phi_i e^{\lambda_i t_f}}{s - \lambda_i} + \mathbf{H}(s), \end{aligned}$$

where the λ_i 's and the ϕ_i 's are the poles and the residues of the system, respectively. Consider the inverse Laplace transform of $\mathbf{G}(s)$:

$$\begin{aligned} \mathcal{L}^{-1}\{\mathbf{G}(s)\} &= \mathbf{h}(t) + \sum_{i=1}^n u_{t_f}(t) \phi_i e^{\lambda_i t_f} e^{\lambda_i (t-t_f)} \\ &= \mathbf{h}(t) + \sum_{i=1}^n u_{t_f}(t) \phi_i e^{\lambda_i t}, \end{aligned}$$

where $u_{t_f}(t)$ is the unit step function

$$u_{t_f}(t) = \begin{cases} 0, & t < t_f; \\ 1, & t \geq t_f. \end{cases}$$

Thus,

$$\begin{aligned}
\|\mathbf{G}\|_{\mathcal{H}_2(t_f)} &= \int_0^{t_f} \mathbf{h}(t) + \sum_{i=1}^n u_{t_f}(t) \phi_i e^{\lambda_i t} dt \\
&= \int_0^{t_f} \mathbf{h}(t) dt \\
&= \|\mathbf{H}\|_{\mathcal{H}_2(t_f)}.
\end{aligned}$$

□

Remark 4.10. The transfer function of a dynamical system $\mathbf{H}(s)$ is closely connected to the pseudo-transfer function $\mathbf{G}(s)$ with respect to the $\mathcal{H}_2(t_f)$ norm, i.e., $\|\mathbf{G}\|_{\mathcal{H}_2(t_f)} = \|\mathbf{H}\|_{\mathcal{H}_2(t_f)}$. Proposition 4.9 enriches the discussion in Subsection 3.5.1 about the implications of $\mathcal{H}_2(t_f)$ optimal finite horizon reduced order modeling.

4.2 A Descent-type Algorithm for the SISO Case

In this section, we briefly discuss a numerical framework to construct a reduced model that satisfies the optimality conditions in Theorems 3.10 and 4.6. Let $\mathbf{H}(s)$ and $\mathbf{H}_r(s)$ be SISO full- and reduced-model transfer functions, respectively, i.e.,

$$\begin{aligned}
\mathbf{H}(s) &= \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{b} = \sum_{i=1}^n \frac{\psi_i}{s - \rho_i} \text{ and} \\
\mathbf{H}_r(s) &= \mathbf{c}_r^T (s\mathbf{I}_r - \mathbf{A}_r)^{-1} \mathbf{b}_r = \sum_{i=1}^r \frac{\phi_i}{s - \lambda_i},
\end{aligned} \tag{4.2.1}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$, $\mathbf{A}_r \in \mathbb{R}^{r \times r}$, and $\mathbf{b}_r, \mathbf{c}_r \in \mathbb{R}^r$. Note that the residues ψ_i and ϕ_i are scalar valued.

As stated before, the $\mathcal{H}_2(t_f)$ minimization problem is a non-convex optimization problem and Theorems 3.10 and 4.6 give the necessary conditions for optimality when both poles and

residues are treated as variables. However, if the poles are fixed, Corollary 4.7 establishes the necessary and sufficient optimality conditions for the residues and suggests a method to find the global minimizer, the optimal residues, by solving a linear system.

The numerical algorithm described here produces a reduced model that satisfies the necessary $\mathcal{H}_2(t_f)$ optimality conditions upon convergence. Let $\boldsymbol{\lambda} \in \mathbf{C}^r$ denote the vector of reduced poles. Thus, the error $\mathbf{J}$ is a function of $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$. Since we explicitly know the gradients of the cost function with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$ (and indeed we can compute the Hessians as well), one can (locally) minimize $\mathbf{J}$ using well established optimization tools. However, as Corollary 4.7 shows, we can easily compute the globally optimal $\boldsymbol{\phi}$ for fixed $\boldsymbol{\lambda}$. Therefore, we treat the reduced poles $\boldsymbol{\lambda}$ as the optimization parameter, and once $\boldsymbol{\lambda}$ are updated at the k -th step of an optimization algorithm, we find/update the corresponding optimal residues $\boldsymbol{\phi}$ based on Corollary 4.7, and then repeat the process. Similar strategies have been successfully employed in the regular $\mathcal{H}_2$ optimal approximation problem [22, 24] and for nonlinear least squares [80, 104, 105]. In summary, we use a quasi-Newton type optimization with $\boldsymbol{\lambda}$ being the parameter, and in each optimization step, we update the residues, $\boldsymbol{\phi}$, by solving the $r \times r$ linear system $\mathbf{M}\boldsymbol{\phi} = \mathbf{z}$ as in Corollary 4.7. Since we are enforcing interpolation at every step of the algorithm, yet tackling the model reduction problem over a finite horizon, we name this algorithm Finite Horizon IRKA, denoted by FHIRKA. Unlike regular IRKA, FHIRKA is a descent algorithm, thus indeed mimics [22] more closely. Upon convergence, the locally optimal reduced model satisfies the first-order necessary conditions of Theorem 4.6.

4.3 Numerical Comparisons

In this section we compare the proposed algorithm FHIRKA with Proper Orthogonal Decomposition (POD), Time-Limited Balanced Truncation (TLBT), and the recently introduced $\mathcal{H}_2(t_f)$ -based algorithm by Goyal and Redmann (GR) [82], as briefly discussed in Section

Algorithm 4 Pseudocode of FHIRKA

Input: Original state space matrices: $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and final simulation time t_f

Output: Reduced state space matrices: $\mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r$

- Pick an r -fold initial shift set that is closed under conjugation.
 - while (not converged)
 - Find the optimal residues ϕ for the given shift set using Corollary 4.7.
 - Update the shifts λ by minimizing the $\mathcal{H}_2(t_f)$ error $\mathbf{J}$.
 - With the converged poles λ and residues ϕ construct the matrices $\mathbf{A}_r, \mathbf{B}_r$ and $\mathbf{C}_r$.
-

3.5.1. We use three models: a heat model of order $n = 197$ [51], a model of the International Space Station 1R Module (ISS 1R) of order $n = 270$ [85], and a toy unstable model of order $n = 402$. The ISS 1R model has 3-inputs and 3-outputs. We focus on the SISO subsystem from the first-input to the first-output. We have created the unstable system such that it has 400 stable poles and 2 unstable poles (positive real part). For all three models, we choose $t_f = 1$, first reduce the original model using POD, GR or TLBT, and then use the resulting reduced model to initialize FHIRKA. Thus, we are trying to investigate how these different initializations affect the final reduced model via FHIRKA and how much improvement one might expect. We trained POD with input $u(t) = \cos(5t)$. We generated the POD snapshot by simulating the time response of the dynamical system using the MATLAB function `lsim`. For details on our implementation of POD, we refer the reader to Section 5.1.2.

The results are shown in Figures 4.1–4.3, where we show the $\mathcal{H}_2(t_f)$ approximation error for different values of r , the order of the reduced model. All three initializations are used for the heat model (Figure 4.1) where the order is reduced from $r = 2$ to $r = 10$ with increments of one. For some r values, certain initializations are excluded (e.g., the GR initialization for $r = 6$) since the algorithm either did not converge or produced poor approximations.

However, this happened only rarely. For the ISS model (Figure 4.2), we use TLBT and GR initializations, since POD approximations were very poor and are excluded. In this case, we pick reduced orders from $r = 2$ to $r = 14$ with increments of 2. For the unstable model (Figure 4.3), we use POD and GR initializations; for this model TLBT produced poor results and is avoided. In this case, we reduce the order from $r = 2$ to $r = 12$ with increments of 2. The first observation is that, since FHIRKA is a descent-method and drives the initialization to a local minimizer, it improves the accuracy of the reduced model for all three initializations as expected. The improvements could be dramatic. For example, FHIRKA is able to outperform POD as much as by an order of magnitude, see, for example, Figure 4.1, the $r = 4$ and $r = 5$ cases. While FHIRKA improves the TLBT and GR initializations as well, the improvements for the heat model are not as significant. However, for the ISS model, FHIRKA is able to improve the TLBT performance as much as 50%; see, e.g., Figure 4.2, the $r = 8$ case. The best improvement of the GR initialization has occurred for the unstable model. For example, for $r = 8$, for the unstable model, FHIRKA improved the reduced model by more than 40%. Gains were significantly better for POD, especially for larger r values. Finally, in Figure 4.4, 4.5, 4.6 we compare the error in the impulse responses due to POD and FHIRKA for the three models. For both methods, POD and FHIRKA, the reduced model was of order $r = 14$ for the ISS model. As we can see from the graphs, FHIRKA clearly outperforms POD on the time interval $[0, 1]$.

In the tables in this section we present some raw numerical results for the same models as above. In addition to comparing the relative errors from different model reduction techniques with FHIRKA, we also include the number of iterations that it takes for FHIRKA to converge. In each table r denotes the order of the reduced model; t_f denotes the final simulation time; *Iterations* denotes the number of iterations it takes for FHIRKA to converge. When FHIRKA fails to converge, we indicate it by writing (NC) next to the number of iterations for which we allowed FHIRKA to run. Similar to the figures above, Table 4.1 shows FHIRKA outperforms POD in terms of accuracy under the $\mathcal{H}_2(t_f)$ norm for a heat model of order $n = 197$. We

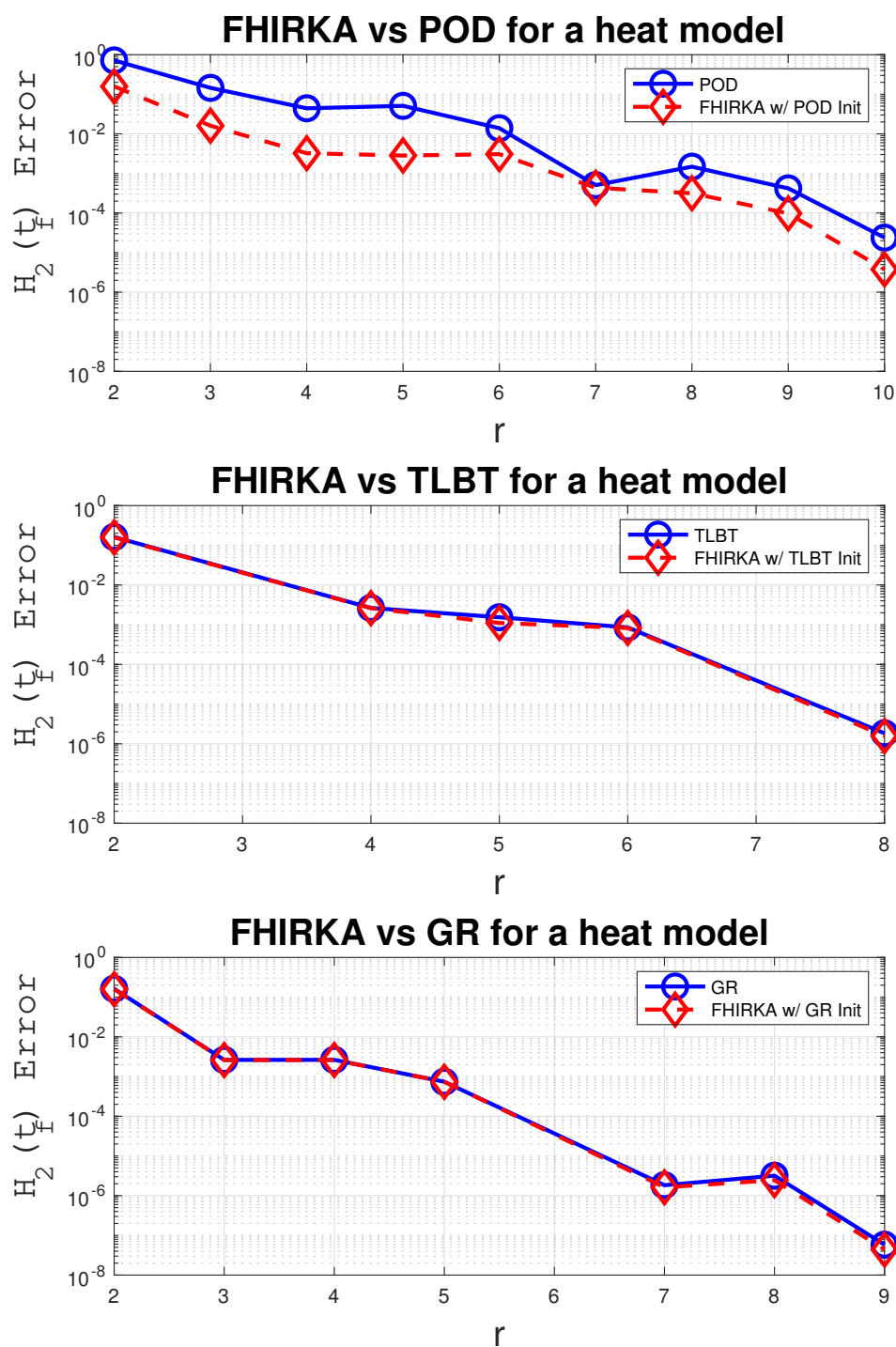


Figure 4.1: FHIRKA and other algorithms for the heat model

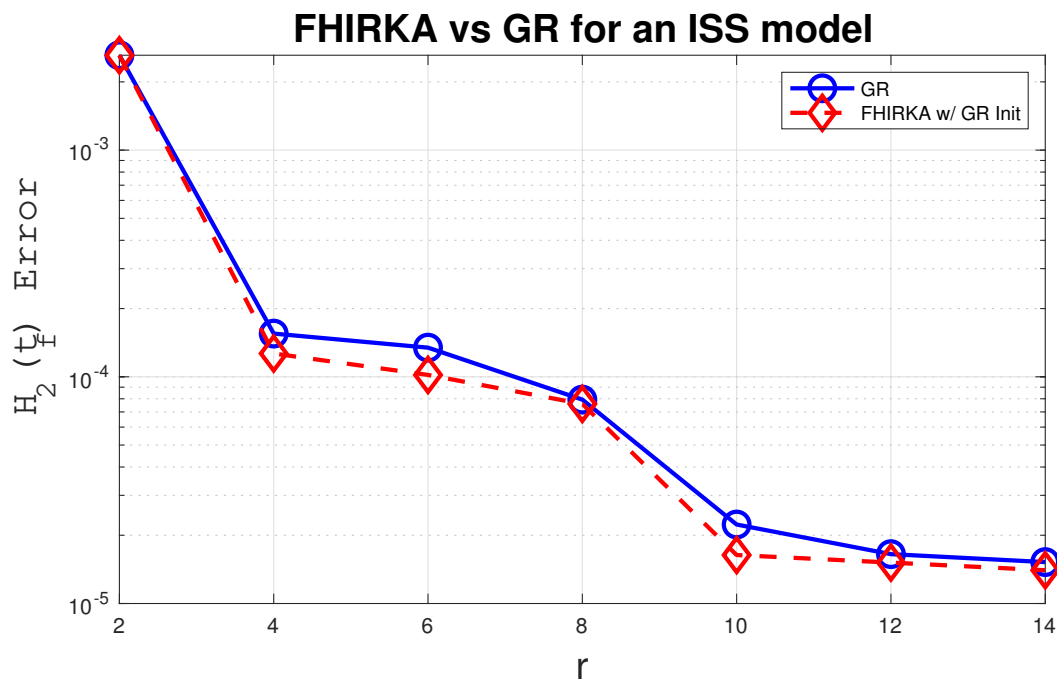
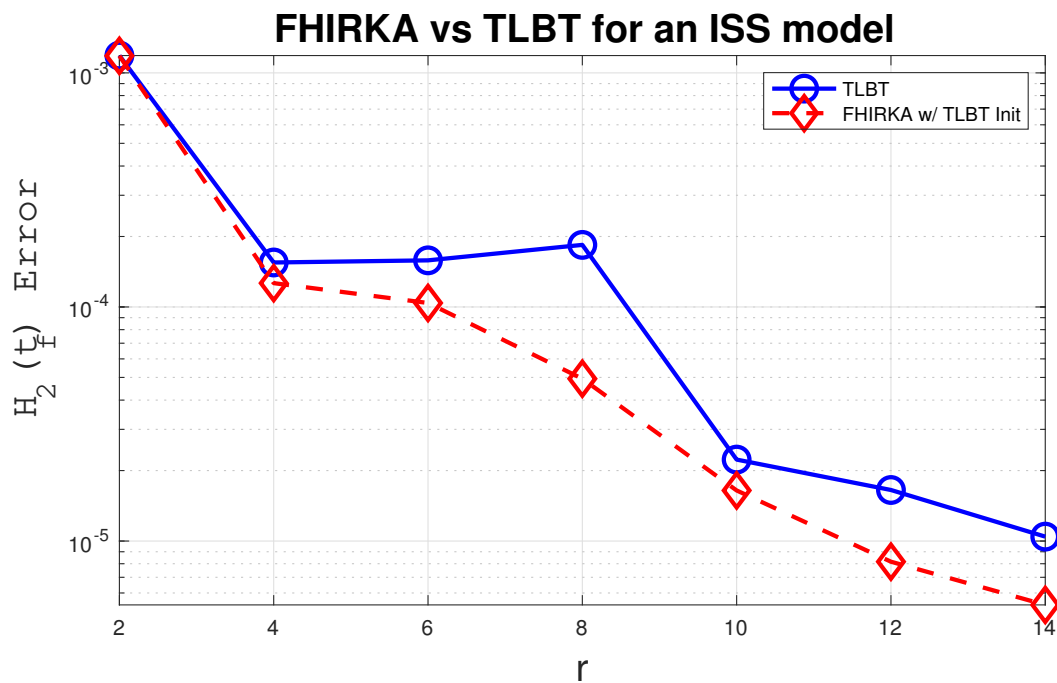


Figure 4.2: FHIRKA and other algorithms for the ISS model

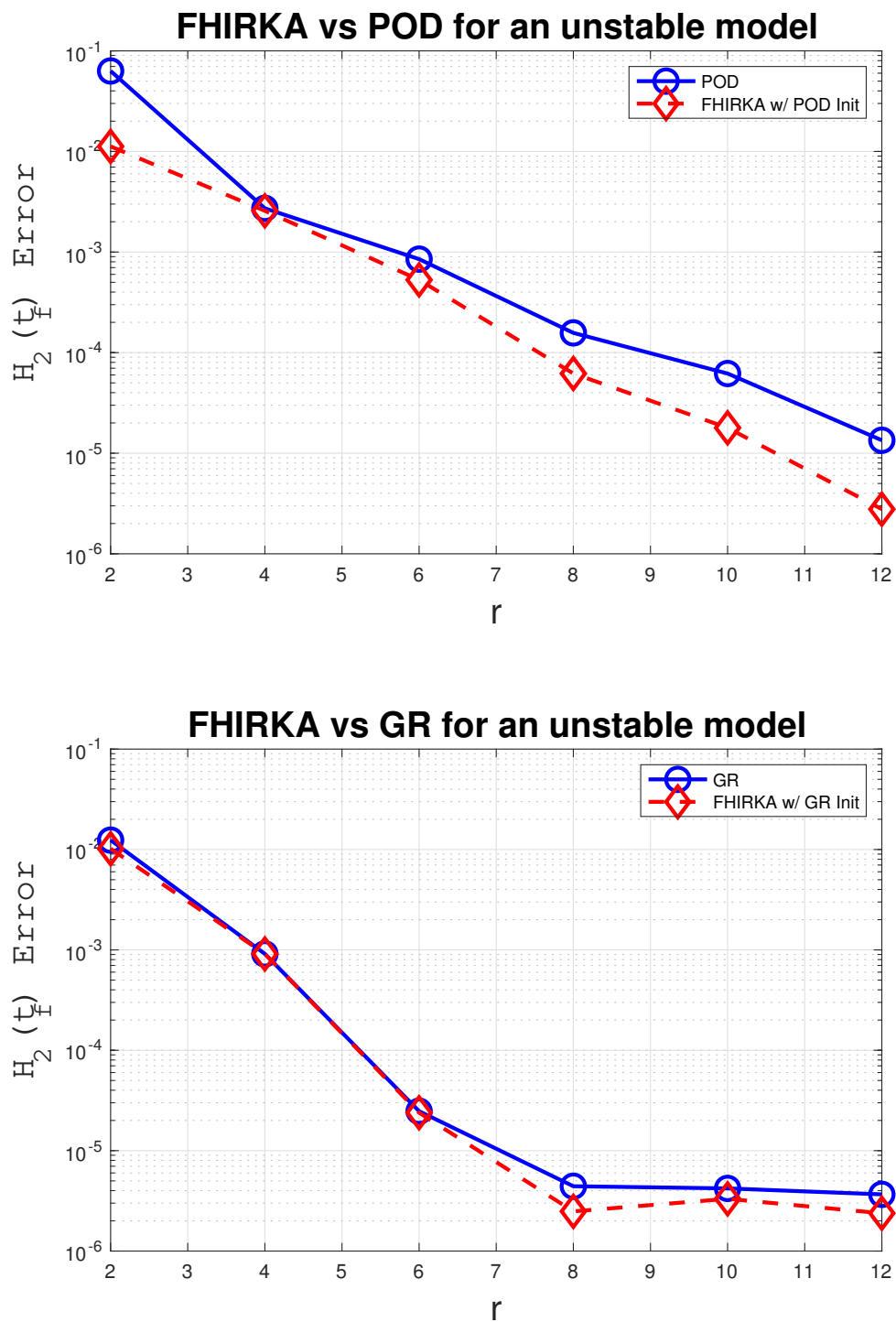


Figure 4.3: FHIRKA and other algorithms for the unstable model

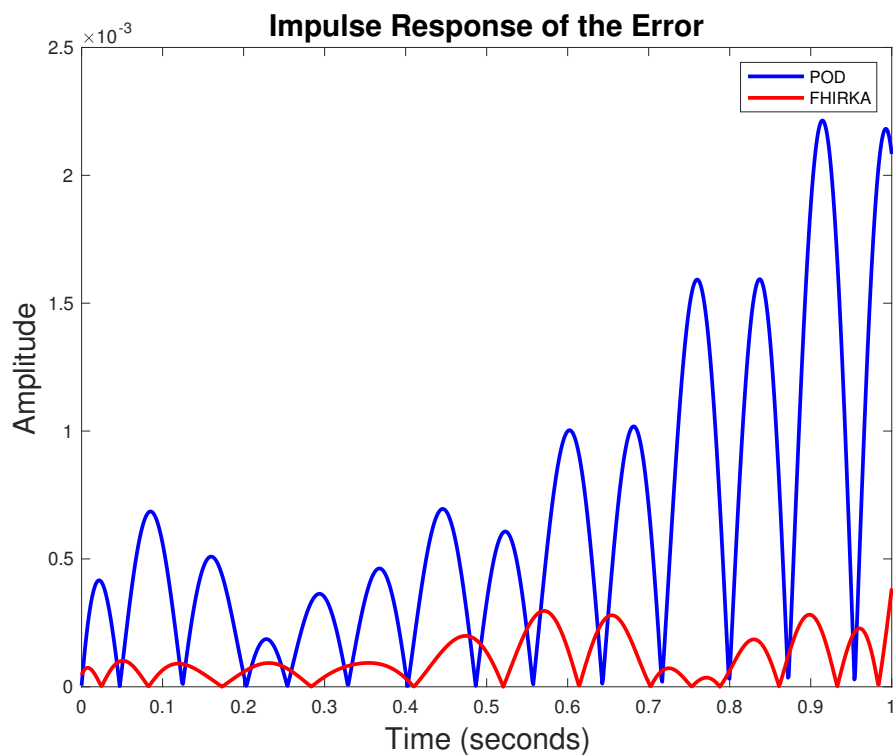


Figure 4.4: FHIRKA and POD for the ISS model

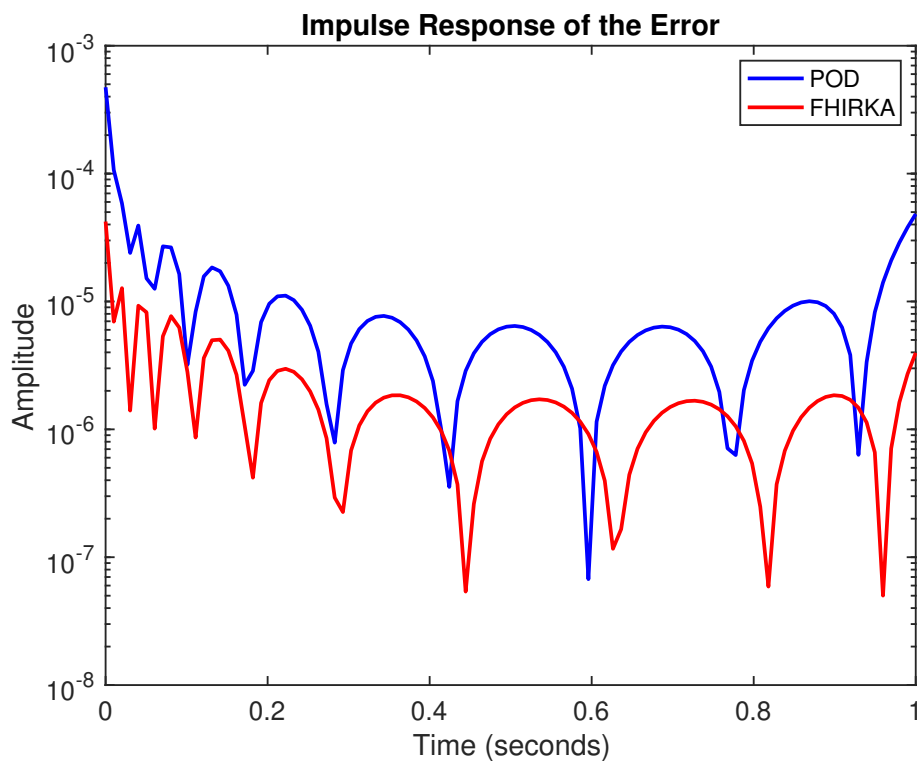


Figure 4.5: FHIRKA and POD for the heat model

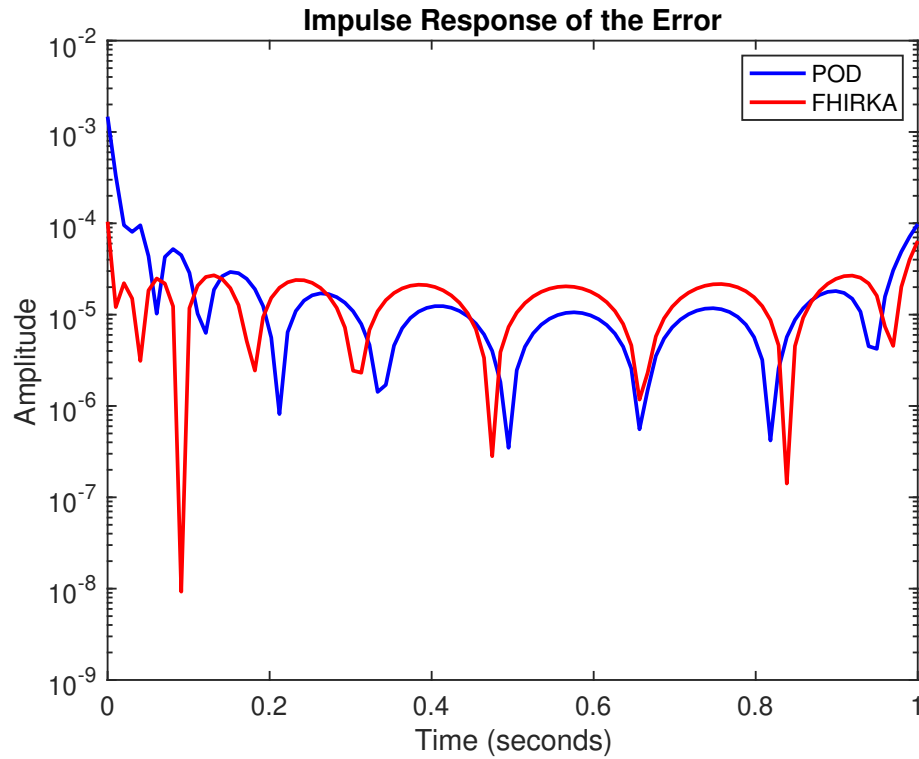


Figure 4.6: FHIRKA and POD for the unstable model

also notice faster convergence for FHIRKA as the order of the reduced model increases. We speculate the faster convergence could be due to a better initialization model for FHIRKA.

In Table 4.2 we show the comparison of the performances of FHIRKA and POD when reducing an ISS model of order $n = 270$. Even in this case, FHIRKA approximates the original more accurately than POD, although we do not observe a trend of faster convergence in this case.

Table 4.1: $\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for a heat model ($n = 197$)

r	t_f	POD Error	FHIRKA Error	Iterations
4	1	4.43×10^{-2}	9.0×10^{-3}	19
5	1	5.11×10^{-2}	2.32×10^{-2}	6
6	1	1.39×10^{-2}	8.5×10^{-3}	3
7	1	5.0674×10^{-4}	4.3467×10^{-4}	3
8	1	1.4813×10^{-3}	5.9447×10^{-4}	3
9	1	4.1931×10^{-4}	1.7433×10^{-4}	3
10	1	2.3879×10^{-5}	4.5018×10^{-6}	3

Table 4.2: $\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for an ISS model ($n = 270$)

r	t_f	POD Error	FHIRKA Error	Iterations
4	1	2.8713×10^{-3}	2.614×10^{-3}	11
6	1	3.148×10^{-3}	2.5749×10^{-3}	200(NC)
8	1	3.2879×10^{-3}	2.4203×10^{-3}	3
10	1	3.1631×10^{-3}	2.3463×10^{-3}	7
12	1	2.8524×10^{-3}	2.354×10^{-3}	3

Tables 4.3 and 4.4 illustrate the superiority of FHIRKA for unstable systems compared to POD. We consider two unstable systems: one consists of 400 poles in the left half plane, and 2 poles in the right half plane, hence it has order $n = 402$, and the other consists of 4000 poles in the left plane, and 2 poles in the right half plane, thus the second system has order $n = 4002$.

Table 4.3: $\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for an unstable system ($n = 402$)

r	t_f	POD Error	FHIRKA Error	Iterations
4	1	2.8713×10^{-3}	2.614×10^{-3}	11
5	1	1.4859×10^{-3}	1.1144×10^{-3}	3
6	1	8.504×10^{-4}	5.2878×10^{-4}	3
7	1	2.2721×10^{-4}	9.4445×10^{-5}	2
8	1	3.2879×10^{-3}	2.4203×10^{-3}	3
9	1	3.1449×10^{-3}	2.4583×10^{-3}	6
10	1	6.1985×10^{-5}	1.7889×10^{-5}	2
11	1	3.1519×10^{-3}	2.3127×10^{-3}	3
12	1	2.8524×10^{-3}	2.354×10^{-3}	3

Table 4.4: $\mathcal{H}_2(t_f)$ errors for POD and FHIRKA for an unstable system ($n = 4002$)

r	t_f	POD Error	FH Error	Iterations
5	1	0.0511	0.0232	6
6	1	1.39×10^{-2}	8.5×10^{-3}	3
7	1	5.0674×10^{-4}	4.3467×10^{-4}	3
8	1	1.4813×10^{-3}	5.9447×10^{-4}	3
9	1	4.1931×10^{-4}	1.7433×10^{-4}	3

Next, we present the raw values for the relative errors when comparing FHIRKA to time-limited balanced truncation. In this case we reduce the full models via TLBT, and then use the reduced model obtained through TLBT to initialize FHIRKA. As Table 4.5 and Table 4.6 show, FHIRKA outperforms TLBT when we reduce a heat model of order $n = 197$ and an ISS model of order $n = 270$, respectively. This is expected since FHIRKA is locally optimal under the $\mathcal{H}_2(t_f)$ norm.

Table 4.5: $\mathcal{H}_2(t_f)$ errors for TLBT and FHIRKA for a heat model ($n = 197$)

r	t_f	TLBT Error	FHIRKA Error	Iterations
4	1	3.2876×10^{-1}	1.3734×10^{-1}	2
5	1	1.4359×10^{-1}	1.8329×10^{-2}	2
6	1	8.5492×10^{-2}	1.9707×10^{-2}	2
7	1	1.2954×10^{-1}	1.4427×10^{-2}	2
8	1	1.2415×10^{-1}	1.0454×10^{-2}	2
9	1	8.4319×10^{-2}	8.3239×10^{-4}	2
10	1	1.7938×10^{-2}	6.4013×10^{-4}	2

Table 4.6: $\mathcal{H}_2(t_f)$ errors for TLBT and FHIRKA for an ISS model ($n = 270$)

r	t_f	TLBT Error	FHIRKA Error	Iterations
4	1	2.6247×10^{-3}	2.6135×10^{-3}	200(NC)
6	1	1.916×10^{-4}	9.7942×10^{-5}	200(NC)
8	1	1.6636×10^{-4}	1.0124×10^{-4}	4
10	1	1.7279×10^{-4}	1.0772×10^{-4}	2
12	1	1.4254×10^{-4}	1.1624×10^{-4}	2
14	1	9.5064×10^{-5}	7.6996×10^{-5}	2

Finally we compare the conventional Iterative Rational Krylov Algorithm (IRKA), which yields a locally $\mathcal{H}_2$ optimal reduced model over an infinite time horizon to FHIRKA. Since FHIRKA produces a reduced model which is locally $\mathcal{H}_2(t_f)$ optimal over a specific finite time interval, we expect FHIRKA to return a more accurate model than IRKA with respect to the $\mathcal{H}_2(t_f)$ norm over the time interval $[0, t_f]$. In the examples shown below, we reduce the full model via IRKA, and then use the reduced model we obtained through IRKA as an initialization for FHIRKA. The following tables, namely Table 4.7, Table 4.8, and Table 4.9 not only demonstrate the better performance of FHIRKA compared to IRKA with respect

to the $\mathcal{H}_2(t_f)$ norm, but also illustrates the advantage of a general finite horizon framework over an infinite horizon one, provided the interval of interest is finite.

Table 4.9 shows a drastic improvement of the FHIRKA over IRKA when dealing with an unstable system. We expect this drastic improvement due to IRKA's inability to produce a locally $\mathcal{H}_2$ optimal reduced system if the full system is unstable.

Table 4.7: $\mathcal{H}_2(t_f)$ errors for IRKA and FHIRKA for a heat model ($n = 197$)

r	t_f	IRKA Error	FHIRKA Error	Iterations
4	1	1.6087×10^{-1}	1.4717×10^{-1}	2
5	1	5.8176×10^{-3}	4.1439×10^{-3}	2
6	1	2.5614×10^{-3}	2.5408×10^{-3}	2
7	1	7.9903×10^{-4}	7.4239×10^{-4}	2
8	1	8.1294×10^{-4}	7.3622×10^{-4}	2
9	1	6.6808×10^{-5}	4.4417×10^{-5}	2
10	1	2.1285×10^{-6}	2.1006×10^{-6}	3

Table 4.8: $\mathcal{H}_2(t_f)$ errors for IRKA and FHIRKA for an ISS model ($n = 270$)

r	t_f	IRKA Error	FHIRKA Error	Iterations
4	1	2.6247×10^{-3}	2.6135×10^{-3}	200(NC)
6	1	1.8958×10^{-4}	9.9932×10^{-5}	191
8	1	1.6483×10^{-4}	9.2441×10^{-5}	13
10	1	1.6354×10^{-4}	1.0912×10^{-4}	6
12	1	1.3247×10^{-4}	1.0485×10^{-4}	2
14	1	1.3084×10^{-4}	1.6309×10^{-6}	2

Table 4.9: $\mathcal{H}_2(t_f)$ errors for IRKA and FHIRKA for an unstable model ($n = 402$)

r	t_f	IRKA Error	FHIRKA Error	Iterations
6	1	2.0949×10^{-1}	3.2466×10^{-3}	3
8	1	1.0856×10^{-1}	8.212×10^{-3}	3
12	1	6.5629×10^{-2}	1.1665×10^{-5}	4

Overall, as expected, FHIRKA yields a better approximation compared to the other algorithms for each model. We find that GR provided the best initialization for FHIRKA. This is not surprising, since GR produces a reduced-model that approximately satisfies the $\mathcal{H}_2(t_f)$ optimality conditions.

4.4 Matrix Exponential Approximation

As we have observed throughout this chapter both the auxiliary interpolated function $\mathbf{G}(s)$ and the time-limited gramians require the computation of the action of a matrix exponential to a vector, i.e., we want to compute $e^{\mathbf{A}t_f}\mathbf{b}$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$. A direct computation of the matrix exponential would be very expensive; fortunately, a plethora of numerical methods to approximate $e^{\mathbf{A}t_f}\mathbf{b}$ exist [6, 26, 48, 60, 61, 69, 90, 99, 116, 159]. For FHIRKA we use the approach proposed in [6], which is an adaptation of the scaling and squaring method for computing the matrix exponential $e^{\mathbf{A}}$ [129, 138]. Since we just need to compute $e^{\mathbf{A}t_f}\mathbf{b}$ we do not need to explicitly compute the matrix exponential itself. We take advantage of a truncated Taylor series approximation and the relation:

$$e^{\mathbf{A}t_f}\mathbf{b} = (e^{z^{-1}\mathbf{A}t_f})^z\mathbf{b}. \quad (4.4.1)$$

Let

$$T_q(z^{-1}\mathbf{A}t_f) = \sum_{i=0}^q \frac{(z^{-1}\mathbf{A}t_f)^i}{i!} \quad (4.4.2)$$

be a truncated Taylor series approximation for $e^{z^{-1}\mathbf{A}t_f}$. Then the following recurrence

$$\begin{aligned} \mathbf{b}_0 &= \mathbf{b} \\ \mathbf{b}_{i+1} &= T_q(z^{-1}\mathbf{A}t_f)\mathbf{b}_i \text{ for } i = 0, 1, \dots, z-1 \end{aligned} \quad (4.4.3)$$

produces the approximation $\mathbf{b}_z \approx e^{\mathbf{A}t_f}\mathbf{b}$. The parameters z and q are chosen based on the backward error analysis in [6, 98, 100]. The same analysis demonstrates that

$$\mathbf{b}_z = e^{\mathbf{A}t_f + \Delta\mathbf{A}t_f}\mathbf{b} + \mathbf{r}, \quad (4.4.4)$$

with

$$\|\Delta\mathbf{A}t_f\| \leq tol \|\mathbf{A}t_f\|,$$

and

$$\|\mathbf{r}\| \leq \frac{c \cdot q \cdot n \cdot \epsilon}{1 - c \cdot q \cdot n \cdot \epsilon} e^{\|\mathbf{A}t_f\|} \|\mathbf{b}\|,$$

where tol is the backward error of the approximate computation of the exponential, ϵ is the machine precision, and c is some small constant. Note $\|\Delta t_f\| = t_f \|\Delta\|$. For more details on this upper bound and an implementation of the method, see [6].

As already mentioned in this section, we can choose from a plethora of methods that guarantee a small approximation error, i.e., $\mathbf{b}_z = e^{\mathbf{A}t_f}\mathbf{b} + \Delta$, where Δ is small. Therefore, instead of interpolating

$$\mathbf{G}(s) = e^{-st_f} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} e^{\mathbf{A}t_f} \mathbf{b} + \mathbf{H}(s),$$

we interpolate

$$\hat{\mathbf{G}}(s) = -e^{-st_f} \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{b}_z + \mathbf{H}(s).$$

Corollary 4.11. *Suppose $\mathbf{b}_z = e^{\mathbf{A}t_f}\mathbf{b} + \Delta$. Let $\hat{\mathbf{G}}(s) = -e^{-st_f}\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}_z + \mathbf{H}(s)$. Then,*

$$\left\| \mathbf{G}(s) - \hat{\mathbf{G}}(s) \right\| \leq \left\| e^{-st_f}\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1} \right\| \|\Delta\|. \quad (4.4.5)$$

Proof. Note that

$$\begin{aligned} \hat{\mathbf{G}}(s) &= -e^{-st_f}\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}(e^{\mathbf{A}t_f}\mathbf{b} + \Delta) + \mathbf{H}(s) \\ &= -e^{-st_f}\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}e^{\mathbf{A}t_f}\mathbf{b} - e^{-st_f}\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\Delta + \mathbf{H}(s). \end{aligned}$$

As a result,

$$\mathbf{G}(s) - \hat{\mathbf{G}}(s) = e^{-st_f}\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\Delta$$

and

$$\left\| \mathbf{G}(s) - \hat{\mathbf{G}}(s) \right\| \leq \left\| e^{-st_f}\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1} \right\| \|\Delta\|.$$

□

If the method chosen for the computation of the matrix exponential $e^{\mathbf{A}t_f}\mathbf{b}$ yields a small value for $\|\Delta\|$, the approximation $\mathbf{b}_z \approx e^{\mathbf{A}t_f}\mathbf{b}$ does not change our algorithm drastically; however, it does speed it up. The only aspect of FHIRKA that changes is the definition of the function that needs to be minimized. Instead of computing the matrix exponential explicitly, we approximate its action on the vector $\mathbf{b}$. Indeed, the MATLAB code used for `expm` is modified so that we do not compute the dense matrix exponential; instead, we benefit from only storing a vector yielded by the action of the matrix exponential on another vector.

Remark 4.12. Unlike $\mathbf{G}(s)$, the approximation $\hat{\mathbf{G}}(s)$ in Corollary 4.11 loses analyticity. Thus, the approximation and the error bound will have singularities that get unboundedly poor for some values of s .

As Table 4.10 shows, the error introduced from this approximation is minimal. Moreover FHIRKA requires only a single computation of $e^{\mathbf{A}t_f}\mathbf{b}$. We tested the approximation with the three full order models described in Section 4.3.

Table 4.10: Matrix Exponential Computation

Model	Order	$\ \mathbf{b}_s - e^{\mathbf{A}t_f}\mathbf{b}\ $
Heat	197	9.4252×10^{-14}
ISS1R	270	4.5551×10^{-15}
Unstable	402	4.8908×10^{-12}

To further illustrate the accuracy and efficiency of the approximation in [6] we consider a convection-diffusion heat equation of order $n = 10000$ [147]. For problems of moderate size such as those referenced in Table 4.10, the MATLAB matrix exponential function `expm` yields satisfactory results even in terms of computational speed. However, for larger problems, the approximation in (4.4.3) is drastically faster, e.g., for the convection-diffusion model, the MATLAB function computes $e^{\mathbf{A}t_f}\mathbf{b}$ in 293.07 seconds, while it takes only 1.25 seconds for the approach in [6]. Moreover, $\|\mathbf{b}_z - e^{\mathbf{A}t_f}\mathbf{b}\| = 7.7869 \times 10^{-13}$, where $\mathbf{b}_z$ is the result of the (4.4.3), and $e^{\mathbf{A}t_f}\mathbf{b}$ is the result of the MATLAB computation. We incorporate the approach in [6] into FHIRKA and reduce the Convection Diffusion model from order $n = 10000$ to order $r = 12$. We initialize FHIRKA with a POD-reduced order model. As we observe from the output and error plots in Figures 4.7 and 4.8, FHIRKA produces accurate approximations even when implemented with the matrix exponential approximation in (4.4.3).

4.5 Summary of Finite Horizon MOR

In Chapter 3, we defined a finite horizon $\mathcal{H}_2(t_f)$ norm, which enables us to measure the approximation error over a finite interval. We reviewed existing techniques for time-limited

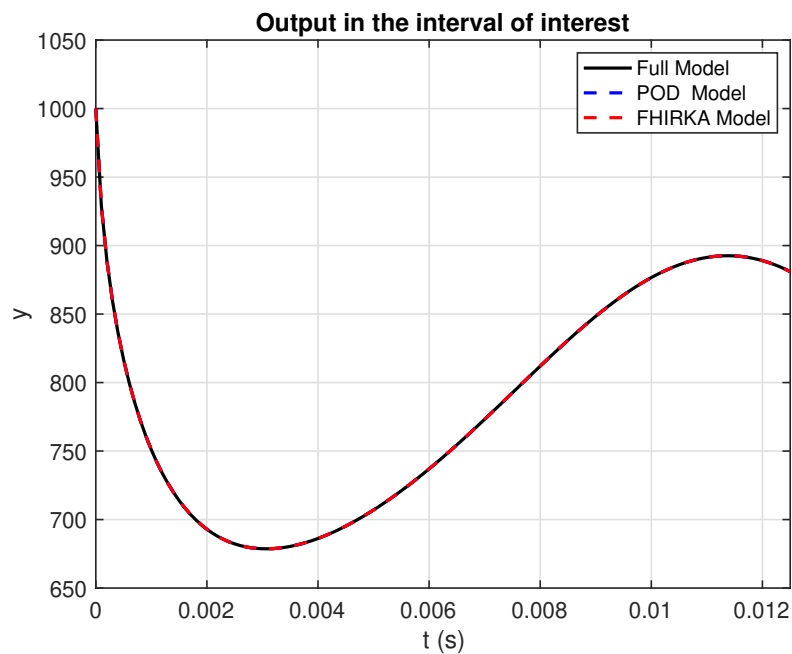


Figure 4.7: Output Plots for Convection Diffusion Model

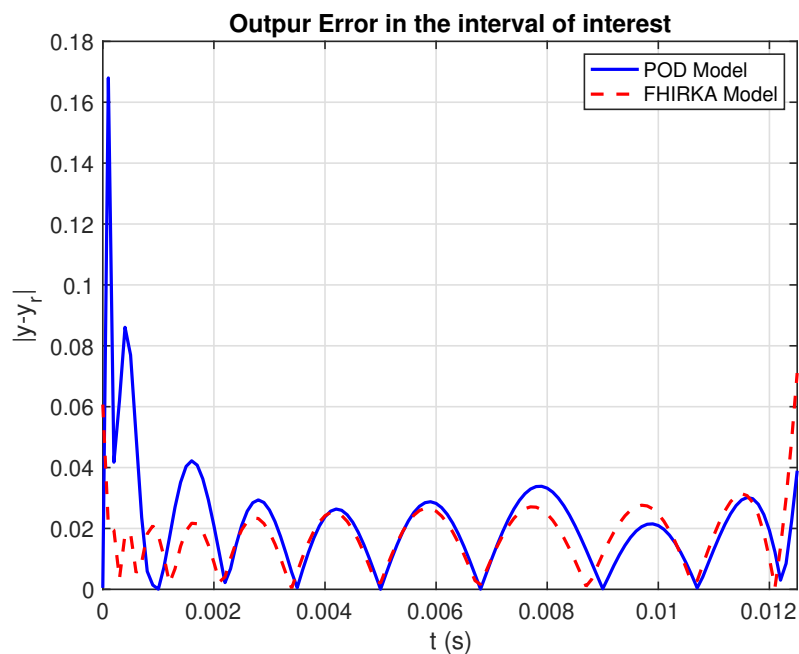


Figure 4.8: Error Plots for Convection Diffusion Model

model reduction such as time-limited balanced truncation and an IRKA-type algorithm which approximately satisfies the time-limited gramian based $\mathcal{H}_2(t_f)$ optimality conditions. Writing the impulse response of the reduced system in terms of the poles and residues of the ROM allowed us to establish interpolatory $\mathcal{H}_2(t_f)$ -optimality conditions for model order reduction of dynamical systems over a finite time horizon. We derived the conditions for both the MIMO and the SISO cases. Even though the optimal interpolation points and tangential directions are still determined by the reduced model, we showed that unlike the regular $\mathcal{H}_2$ problem, a modified reduced-transfer function should interpolate a modified full-order transfer function.

For the special case of SISO models, we have studied a numerical algorithm, FHIRKA, and illustrated that it performs effectively. FHIRKA outperforms POD, IRKA, TLBT, and the GR algorithm [82] for the examples discussed in this dissertation. Numerical experiments were consistent with our theoretical results.

Chapter 5

Operator Splitting with Model Reduction

As we have already discussed in this dissertation, a plethora of powerful and robust methods exists for reducing linear dynamical systems. However, the nonlinear model reduction problem is more difficult.

The Iterative Rational Krylov Algorithm (IRKA) has inspired algorithms that produce high fidelity nonlinear reduced order models for nonlinearities that can be converted to a bilinear and a quadratic bilinear form, respectively Bilinear IRKA (B-IRKA) [1, 27] and Quadratic Bilinear IRKA (QB-IRKA) [2, 3, 34]. Balanced Truncation (BT) has also been extended for bilinear and quadratic bilinear systems [5, 33, 63]. Since not every system can be converted into a bilinear, or quadratic bilinear form, QB-IRKA and BT for quadratic bilinear systems are not feasible in every case.

Proper Orthogonal Decomposition (POD) provides an alternative approach [40, 42, 93, 112, 119, 121, 160, 182]. POD has been implemented successfully in many applications, such as optimal control [120], fluid dynamics [15, 121], compressible flow [157], and aerodynamics [46]. While POD can be very useful, one of its main downsides resides in the fact that the reduced order model is dependent on the selected input. In this chapter, we propose a method to mitigate the impact of the input on the reduced system by incorporating model reduction into operator splitting.

Operator splitting is a numerical method widely used for solving differential equations in-

volving terms with different physics [4, 76, 103, 107, 132, 161, 162, 169, 171]. A few other instances of splitting include dimensional splitting, domain decomposition, and splitting of objective functions in optimization [132]. In fact, operator splitting generalizes to any differential equation that involves two or more operators. The main motivation behind operator splitting is that it is faster than the direct computation of a solution. This comes at the cost of accuracy. In this chapter we discuss how to integrate system theoretic methods like the Iterative Rational Krylov Algorithm (IRKA) and Balanced Truncation (BT) with trajectory based techniques like Proper Orthogonal Decomposition (POD) in order to solve nonlinear ODEs more efficiently. First, we review model reduction techniques for nonlinear systems. Then, we discuss operator splitting in the context of separating linear terms from nonlinearities and propose an algorithm which incorporates model order reduction into operator splitting. In the last two sections of this chapter we perform some theoretical and numerical analysis for the proposed algorithm.

5.1 Nonlinear Model Reduction

In this section we discuss existing model reduction methods for nonlinear system. First, we describe methods for quadratic bilinear systems, which constitute a special class of nonlinear systems. The structure of quadratic bilinear systems enables the extension of IRKA and BT to QB systems. Then, we cover POD and DEIM, which are popular methods for reducing general nonlinearities.

5.1.1 Quadratic Bilinear Systems

For Quadratic Bilinear systems, a class of nonlinear systems, the authors in [34] propose an approach that yields an $\mathcal{H}_2$ -quasi-optimal reduced order model. Quadratic Bilinear systems

have the following form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{H} \cdot \mathbf{x}(t) \otimes \mathbf{x}(t) + \sum_{k=1}^m \mathbf{N}_k \cdot \mathbf{x}(t) \mathbf{u}_k(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned}\tag{5.1.1}$$

where $\mathbf{A}, \mathbf{N}_k \in \mathbb{R}^{n \times n}$, $\mathbf{H} \in \mathbb{R}^{n \times n^2}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{p \times n}$ are constant matrices. Consistent with our previous notation $\mathbf{x}(t) \in \mathbb{R}^n$ are the states, $\mathbf{u}(t) \in \mathbb{R}^m$ are the inputs, and $\mathbf{y}(t) \in \mathbb{R}^p$ are the outputs. The symbol $\otimes$ indicates Kronecker product. Many models from engineering and physics contain a quadratic nonlinearity, as is the case with spatial discretizations of Burgers' equation and the Allen-Cahn equation. Furthermore, many smooth nonlinearities can be converted into a Quadratic-Bilinear form [28, 84]. However, many of this conversions give rise to differential algebraic equations (DAEs) or descriptor systems [122], which introduce a complete new set of challenges [32, 89].

The Quadratic Bilinear Iterative Rational Krylov Algorithm (QB-IRKA), similar to IRKA, is a projection based algorithm. Thus, the reduced matrices are obtained via projection. QB-IRKA produces a high fidelity reduced model under a truncated $\mathcal{H}_2$ -norm for quadratic bilinear systems which is defined with the help of the Volterra series [34, 158].

Balanced truncation has also been extended to the bilinear and quadratic bilinear approaches [31, 33]. As discussed in previous chapters, Balanced Truncation generates a reduced order model by eliminating states that are simultaneously hard to reach and hard to observe. Balanced Truncation for QB systems relies on the computation of the truncated reachability and observability Gramians, $\mathcal{P}$ and $\mathcal{Q}$, which are the solutions to the following Lyapunov equations:

$$\begin{aligned}\mathbf{A}\mathcal{P}_\tau + \mathcal{P}_\tau\mathbf{A}^T + \mathbf{H}(\mathcal{P}_l \otimes \mathcal{P}_l)\mathbf{H}^T + \sum_{k=1}^m \mathbf{N}_k\mathcal{P}_l\mathbf{N}_k^T + \mathbf{B}\mathbf{B}^T &= 0 \text{ and} \\ \mathbf{A}^T\mathcal{Q}_\tau + \mathcal{Q}_\tau\mathbf{A} + \mathcal{H}^{(2)}(\mathcal{P}_l \otimes \mathcal{Q}_l)(\mathcal{H}^{(2)})^T + \sum_{k=1}^m \mathbf{N}_k^T\mathcal{Q}_l\mathbf{N}_k + \mathbf{C}^T\mathbf{C} &= 0,\end{aligned}\tag{5.1.2}$$

where $\mathcal{P}_l$ and $\mathcal{Q}_l$ are the solutions to

$$\begin{aligned}\mathbf{A}\mathcal{P}_l + \mathcal{P}_l\mathbf{A}^T + \mathbf{B}\mathbf{B}^T &= 0 \\ \mathbf{A}^T\mathcal{Q}_l + \mathcal{Q}_l\mathbf{A} + \mathbf{C}^T\mathbf{C} &= 0,\end{aligned}\tag{5.1.3}$$

and $\mathcal{H}^{(2)}$ denotes the mode-2 matricization of the Hessian. For more details on the tensor properties and how they relate to quadratic bilinear systems, see [34, 109, 117].

Similar to the original Balanced Truncation and time limited Balanced Truncation (TLBT), the efficient computation of the low-rank Cholesky factors of the truncated Gramians is essential; see [38, 164]. After computing the Cholesky factors, the procedure is the same as in the case of regular BT and TLBT. For further details on Balanced Truncation for QB systems, including error bounds, we refer the reader to [33].

Quadratic Bilinear IRKA and BT for Quadratic Bilinear systems yield high-fidelity reduced systems independent of the control inputs since they depend only on state-space quantities. However, for systems that cannot be written in a quadratic bilinear form, we need to consider other tools such as POD for example.

5.1.2 Proper Orthogonal Decomposition

Let $f(x, t)$ be some function of interest that can be approximated as follows over some domain:

$$f(x, t) \approx \sum_{i=1}^k \alpha_i(t) g_i(x).\tag{5.1.4}$$

We expect that this approximation gets closer and closer to the true function as k becomes larger. In other words, the limit of the infinite sum in (5.1.4) as k goes to infinity is the original function $f(x, t)$. The approximation in (5.1.4) is not necessarily unique. We can pick $g_i(x)$ to be Chebyshev polynomials, Legendre polynomials or some other set that could serve as a basis. For each basis, we would have different functions $\alpha_i(t)$. For POD it makes

sense to choose an orthonormal basis. In this section we discuss how to accomplish this in the finite dimensional case. POD originated in statistical analysis. In statistics it is more commonly known as Principal Component Analysis (PCA) [114, 123, 139]. Consider the case where we take measurements for a state variable of order n at N instants of time. We store the data in a $n \times N$ snapshot matrix $\mathbf{X}$. In statistical analysis, POD (or PCA) is useful when dealing with large sets of data because it enables us to extract the most “important” information from the dataset. In the dynamical system setting, which is our area of interest, the snapshot matrix $\mathbf{X}$ is obtained by simulating the dynamical system. In other words, we have

$$\mathbf{X} = [\mathbf{x}(t_1) \quad \mathbf{x}(t_2) \cdots \mathbf{x}(t_N)],$$

where $\mathbf{x}(t_i)$ denotes the evaluation of the state variable $\mathbf{x}$ at time t_i . Once we have constructed the snapshot $\mathbf{X}$, we get its singular value decomposition as follows:

$$\mathbf{X} = \Phi \Sigma \Psi^T. \quad (5.1.5)$$

Then, we pick the the dominant r left singular vectors to form the POD basis $\mathbf{V}$. Let us illustrate POD by considering the nonlinear dynamical system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{f}(\mathbf{x}, t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned} \quad (5.1.6)$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times n}$ are constant matrices. Consistent with the notation from the linear case, in this system $\mathbf{x}(t) \in \mathbb{R}^n$ is the internal variable. Obviously, the dimension of the system is n . The input is denoted by $\mathbf{u}(t)$, while $\mathbf{y}(t)$ is the output. Just as in the linear case, if $m = p = 1$, then the dynamical system is called single-input/single-output (SISO). If $m > 1$ and $p > 1$, the system is called multi-input/multi-output (MIMO). For large n , e.g., $n > 10^6$, we want to replace this nonlinear model with a reduced order

nonlinear model

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{B}_r \mathbf{u}(t) + \mathbf{V}^T \mathbf{f}(\mathbf{V} \mathbf{x}_r, t) \\ \mathbf{y}_r(t) &= \mathbf{C}_r \mathbf{x}_r(t)\end{aligned}\tag{5.1.7}$$

where $\mathbf{A}_r = \mathbf{V}^T \mathbf{A} \mathbf{V}$, $\mathbf{C}_r = \mathbf{C} \mathbf{V}$, and $\mathbf{B}_r = \mathbf{V}^T \mathbf{B}$. Algorithm 5 presents a synthesized sketch of the POD algorithm.

As we notice, the reduction basis $\mathbf{V}$ is contingent on the snapshot matrix $\mathbf{X}$, which in turn depends on a particular simulation of the system. Since for different inputs, we have different simulations, hence, different snapshots, the POD generated reduced models could be different depending on the training input.

Algorithm 5 Pseudocode for POD

Input: Original system

Output: Reduced system

- Construct snapshot $\mathbf{X}$ by simulating the ODE
- Obtain the singular value decomposition of $\mathbf{X}$: $\mathbf{X} = \Phi \Sigma \Psi^T$
- Select the basis $\mathbf{V} = \Phi(:, 1 : r)$.
- Use projection to reduce the state space matrices: $\mathbf{A}_r = \mathbf{V}^T \mathbf{A} \mathbf{V}$, $\mathbf{B}_r = \mathbf{V}^T \mathbf{B}$, $\mathbf{C}_r = \mathbf{C} \mathbf{V}$.
- The reduced system is:

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= \mathbf{A}_r \mathbf{x}_r(t) + \mathbf{B}_r \mathbf{u}(t) + \mathbf{V}^T \mathbf{f}(\mathbf{V} \mathbf{x}_r, t) \\ \mathbf{y}_r(t) &= \mathbf{C}_r \mathbf{x}_r(t)\end{aligned}$$

Even though (5.1.7) is a reduced order model, we still could have costly computations due to

the lifting bottleneck for the evaluation of nonlinearity at a “full order” vector. In order to deal with this bottleneck we use Discrete Empirical Interpolation Method (DEIM) [53, 54] which is a discretized version of the Empirical Interpolation Method (EIM) [17, 97, 133].

5.1.3 Discrete Empirical Interpolation Method

As we have extensively discussed throughout this dissertation, model reduction aims to produce lower dimension systems that approximate the original outputs. POD is one popular method. However, when reducing a nonlinear system via POD, the computational complexity of the nonlinearity remains unchanged. For example, the nonlinear term in (5.1.6) is approximated as follows:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{V}\mathbf{x}_r, t).$$

Even for the reduced model, we need to evaluate the nonlinearity for a vector of the same length as the order of the full model. In order to reduce the cost of evaluating the nonlinear terms, we approximate the nonlinearity $\mathbf{f}$ using the Discrete Empirical Interpolation Method (DEIM) [53]. The DEIM algorithm selects a set of indices which determine what components of the state variable to evaluate. We initialize the DEIM algorithm with a linearly independent set $\{\mathbf{u}_l\}_1^m \subset \mathbb{R}^n$ obtained from snapshots of the nonlinearity $\mathbf{f}$. Once we construct a set of indices using the DEIM algorithm, the constructed set of indices informs us where to evaluate the nonlinearity. The DEIM approximation of order m for $\mathbf{f}(\mathbf{x})$ is

$$\hat{\mathbf{f}}(\mathbf{x}) := \mathbf{U}(\mathbf{P}^T \mathbf{U})^{-1} \mathbf{P}^T \mathbf{f}(\mathbf{x}) \quad (5.1.8)$$

where the columns of $\mathbf{U}$ are $\{\mathbf{u}_l\}_1^m$ and $\mathbf{P}$ is a permutation of some columns of the identity according to the DEIM indices. Incorporating DEIM into POD generates the reduced

nonlinearity

$$\mathbf{f}_r(\mathbf{x}_r(t)) = \mathbf{V}^T \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1} \mathbf{f}(\mathbf{P}^T \mathbf{V} \mathbf{x}_r(t), \mathbf{u}(t)), \quad (5.1.9)$$

where $\mathbf{V}$ is a POD basis. As we see from (5.1.8), the permutation matrix $\mathbf{P}$ selects the components at which the nonlinearity is evaluated. To obtain $\mathbf{U}$ we select the first m singular vectors in the singular value decomposition of the snapshots of $\mathbf{f}$. Once we have the input bases $\{\mathbf{u}_l\}_1^m$, DEIM constructs the set of indices $\mathcal{P}_1, \dots, \mathcal{P}_m$ which determine the permutation matrix $\mathbf{P}$. The first index $\mathcal{P}_1$ corresponds to the component of $\mathbf{u}_1$ that has the largest magnitude. The rest of the indices are selected based on the largest magnitude entries of the error between the corresponding input basis and its approximation from interpolating the basis as shown in the DEIM algorithm below.

For other variations of DEIM see [57, 146, 167, 185].

5.2 Linear Operator Splitting

Let us consider the following ordinary differential equation:

$$\dot{\mathbf{x}}(t) = (\mathbf{M} + \mathbf{N})\mathbf{x}(t) \quad (5.2.1)$$

with solution

$$\mathbf{x}(t) = e^{t(\mathbf{M}+\mathbf{N})}\mathbf{x}(0). \quad (5.2.2)$$

In many applications, computing $e^{t(\mathbf{M}+\mathbf{N})}$ is not always cheap or easy. Nevertheless, there exist techniques which enable us to compute $e^{t\mathbf{M}}$ and $e^{t\mathbf{N}}$ separately. This is one reason to employ operator splitting, which yields

$$e^{t(\mathbf{M}+\mathbf{N})} \approx e^{t\mathbf{M}}e^{t\mathbf{N}}. \quad (5.2.3)$$

Algorithm 6 Pseudocode of DEIM [53]

Input: Original nonlinearity

Output: DEIM indices

- Construct snapshot $\mathbf{F}$
 - Obtain the singular value decomposition of $\mathbf{F}$: $\mathbf{F} = \Phi \Sigma \Psi^T$
 - Select the basis $\mathbf{U} = \Phi(:, 1 : m)$.
 - $[|\rho|, \mathcal{P}_1] = \max\{|\mathbf{u}_1|\}$
 - $\mathbf{U} = \mathbf{u}_1, \mathbf{P} = e_{\mathcal{P}_1}, \vec{\mathcal{P}} = [\mathcal{P}_1]$
 - **for** $l = 2 : m$
 - Solve $(\mathbf{P}^T \mathbf{U}) \mathbf{c} = \mathbf{P}^T \mathbf{u}_l$ for $\mathbf{c}$
 - $\mathbf{r} = \mathbf{u}_l - \mathbf{U} \mathbf{c}$
 - $[|\rho|, \mathcal{P}_l] = \max\{|\mathbf{r}|\}$
 - $U = [U \ \mathbf{u}_l], \ P = [P \ e_{\mathcal{P}_l}], \ \vec{\mathcal{P}} = \begin{bmatrix} \vec{\mathcal{P}} \\ \mathcal{P}_l \end{bmatrix}$
-

If $\mathbf{M}\mathbf{N} = \mathbf{N}\mathbf{M}$, equality is attained [79]. The splitting in Equation (5.2.3) is known as first order splitting, named after the order of accuracy. Consider the following Taylor expansions:

$$\begin{aligned} e^{t\mathbf{M}} &= \mathbf{I} + t\mathbf{M} + \frac{1}{2}t^2\mathbf{M}^2 + \dots \\ e^{t\mathbf{N}} &= \mathbf{I} + t\mathbf{N} + \frac{1}{2}t^2\mathbf{N}^2 + \dots \\ e^{t(\mathbf{M}+\mathbf{N})} &= \mathbf{I} + t(\mathbf{M} + \mathbf{N}) + \frac{1}{2}t^2(\mathbf{M} + \mathbf{N})^2 + \dots \end{aligned} \quad (5.2.4)$$

Note that

$$e^{t\mathbf{M}}e^{t\mathbf{N}} = \mathbf{I} + t(\mathbf{M} + \mathbf{N}) + t^2\left(\frac{1}{2}\mathbf{M}^2 + \mathbf{N}\mathbf{M} + \frac{1}{2}\mathbf{N}^2\right) + \dots \quad (5.2.5)$$

Therefore, if $\mathbf{M}$ and $\mathbf{N}$ do not commute, then the Taylor series of $e^{t(\mathbf{M}+\mathbf{N})}$ and $e^{t\mathbf{M}}e^{t\mathbf{N}}$ have only the identity matrix $\mathbf{I}$ and the first order term in common. This implies the splitting has $\mathcal{O}(h^2)$ accuracy on a subinterval of length h , and $\mathcal{O}(h)$ accuracy over the entire interval, i.e., the local error is $\mathcal{O}(h^2)$ and the global error is $\mathcal{O}(h)$.

The symmetric Strang splitting generates a more accurate approximation with $\mathcal{O}(h^2)$ global error [132, 170, 171]. However, for numerical experiments in this chapter, a first order splitting is sufficient.

5.3 Nonlinear Operator Splitting

For nonlinear splitting we separate the linear and nonlinear terms. This means that over each time step, we compute the solutions of the linear and nonlinear parts separately. Consider the following system of differential equations:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t)), \quad (5.3.1)$$

where $\mathbf{A}$ is a constant matrix and $\mathbf{f}(\mathbf{x}(t))$ is some nonlinear function. When applying operator splitting, first, we numerically integrate $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ over $[t_n, t_{n+1}]$. If we use, e.g., the Forward Euler method, we have

$$\mathbf{x}_s = \mathbf{x}_n + \Delta t \mathbf{A} \mathbf{x}_n, \quad (5.3.2)$$

where $\Delta t = t_{n+1} - t_n$.

Then, we use the result, $\mathbf{x}_s$ as an initial condition for the next “half step”, i.e., we numerically integrate $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t))$ over $[t_n, t_{n+1}]$ by setting the starting point $\mathbf{x}(t_n) = \mathbf{x}_s$, i.e., we have

$$\mathbf{x}_{n+1} = \mathbf{x}_s + \Delta t f(\mathbf{x}_s). \quad (5.3.3)$$

We repeat this process for every step. A simple operator splitting scheme is illustrated visually in Figures 5.1, 5.2, and 5.3.

The convergence analysis for nonlinear operator splitting is not as robust as in the linear case [132]. We revisit the error/convergence analysis for nonlinear systems in Section 5.5.

5.4 Operator Splitting and MOR for General Nonlinearities

As we have broadly discussed throughout this dissertation, modeling and simulation for large scale nonlinear systems can be very costly. Thus, the need for efficient computational methods arises. Operator splitting and model order reduction, even as separate approaches, are very effective in reducing costs of numerical computations. In this section we combine these strategies in order to approximate nonlinear dynamical systems with high fidelity

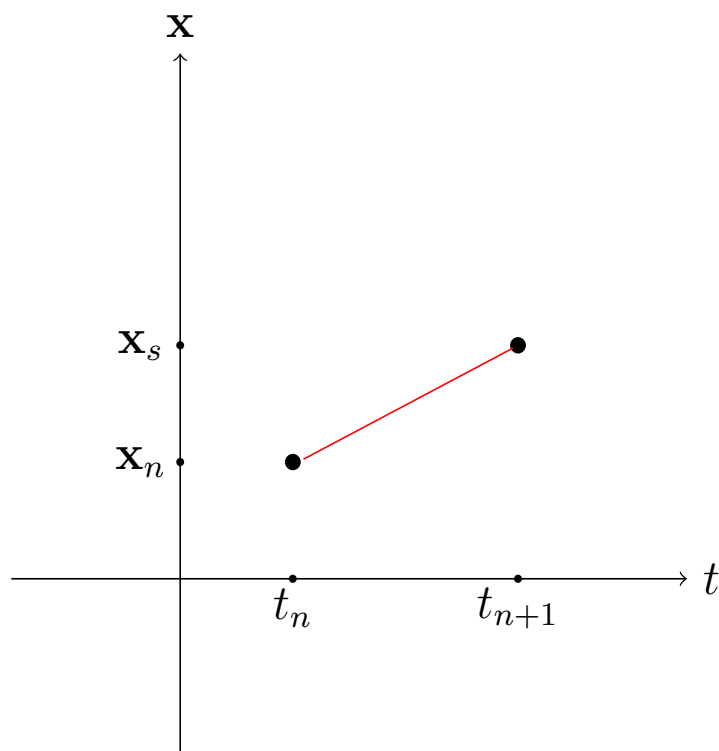


Figure 5.1: Operator Splitting: Step 1

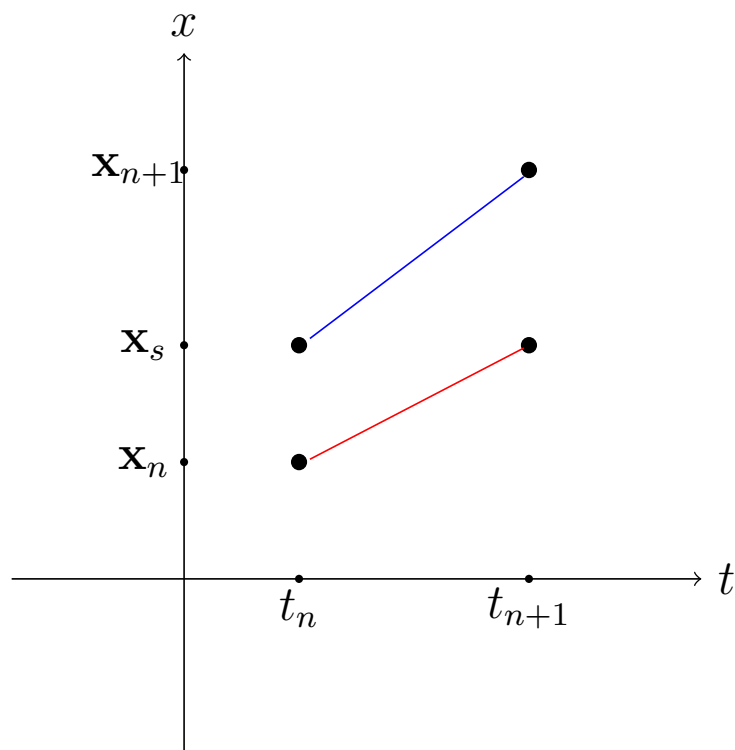


Figure 5.2: Operator Splitting: Step 2

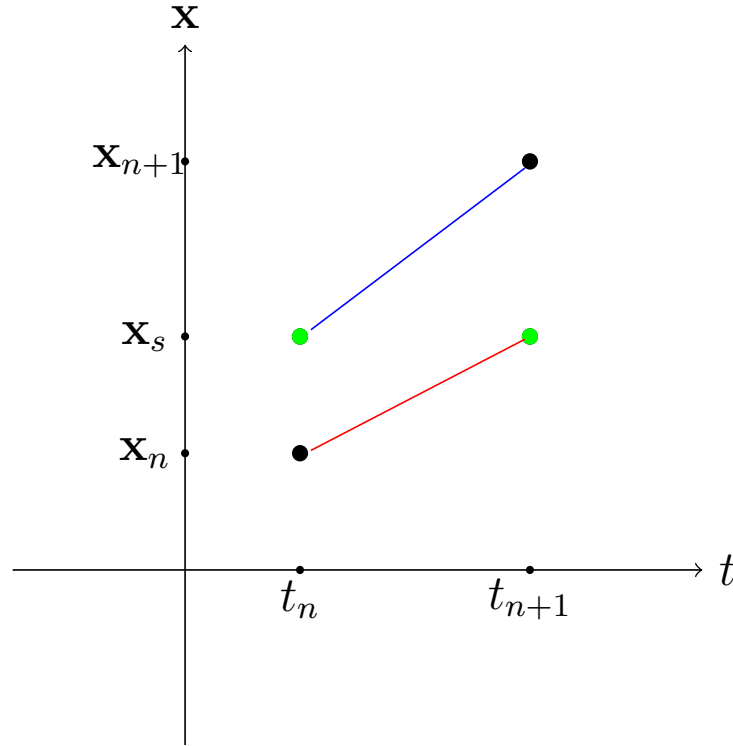


Figure 5.3: Operator Splitting: Step 3

reduced models which are minimally dependent on the control inputs. Consider a general nonlinear system,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{5.4.1}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{p \times n}$ are constant matrices, and $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ is some nonlinear function. Similar to the linear case we discussed in previous chapters, the variable $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the internal variables, $\mathbf{u}(t) \in \mathbb{R}^m$ denotes the control inputs, and $\mathbf{y}(t) \in \mathbb{R}^p$ denotes the outputs. The length of the internal variable $\mathbf{x}(t)$, i.e., n , is called the order of the full model that we would like to reduce. As we mentioned in Section 5.1, many existing methods, e.g., Quadratic Bilinear IRKA (QB-IRKA) and Proper Orthogonal Decomposition (POD) could approximate nonlinear systems such as (5.4.1) with a reduced order model. Both QB-IRKA and POD are very effective model reduction techniques and produce accurate approximations in many cases. QB-IRKA yields a

system that approximately satisfies necessary optimality conditions under the truncated $\mathcal{H}_2^{(\tau)}$ norm and POD generates an optimal approximation with respect to the observed state data. However, it is not always feasible to convert a system into quadratic bilinear form. POD is input-dependent and relies on sampled trajectories, hence, it cannot capture what it has not observed. In this chapter, by taking advantage of operator splitting, we propose a numerical method that integrates the best features of system theoretic approaches with trajectory based techniques. Operator splitting enables us to consider the linear and nonlinear terms in (5.4.1) separately. First, we reduce the linear terms of (5.4.1) via IRKA (alternatively via balanced truncation, or your favorite system theoretic method) to obtain $\mathbf{A}_{r1}, \mathbf{B}_{r1}$, and $\mathbf{C}_{r1}$. Then, we approximate $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ with $\mathbf{V}^T \mathbf{f}(\mathbf{V} \mathbf{x}(t), \mathbf{u}(t))$ where $\mathbf{V}$ is a POD basis. We incorporate the Discrete Empirical Interpolation Method (DEIM) to speed up the POD reduction. Therefore, we have the following reduced ODE systems:

$$\dot{\mathbf{x}}_{r1}(t) = \mathbf{A}_{r1} \mathbf{x}_{r1}(t) + \mathbf{B}_{r1} \mathbf{u}(t) \quad (5.4.2)$$

$$\dot{\mathbf{x}}_{r2}(t) = \mathbf{V}^T \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1} \mathbf{f}(\mathbf{P}^T \mathbf{V} \mathbf{x}(t), \mathbf{u}(t)). \quad (5.4.3)$$

Once we have reduced the the linear and nonlinear terms individually, we apply operator splitting, i.e., in each step we numerically integrate the linear and nonlinear parts separately.

For the n -th step of our numerical scheme, first, we obtain an approximation of $\mathbf{x}_{r1}(t_{n+1})$ by evolving (5.4.2) over the interval $[t_n, t_{n+1}]$. Then using the computed approximation of $\mathbf{x}_{r1}(t_{n+1})$ as the starting point, we get an approximate value for $\mathbf{x}_{r2}(t_{n+1})$ by evolving the nonlinear reduced model (5.4.3) over the interval $[t_n, t_{n+1}]$. Even though we can choose different ROM techniques to reduce the linear and strictly nonlinear parts of the dynamical system of interest, for the sake simplicity and clarity, we assume we perform linear model reduction via IRKA, and nonlinear model reduction via POD. We refer to this numerical scheme as IRKA-POD Splitting (IPS). Algorithm 7 describes the pseudocode for IPS.

Algorithm 7 Pseudocode of IRKA-POD Splitting (IPS)

Input: Original ODE system

Output: Approximate reduced solution of the ODE system

- Given a nonlinear dynamical system,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) + \mathbf{f}(\mathbf{x}(t), u(t)) \quad (5.4.4)$$

isolate the nonlinear residual to obtain

$$\dot{\mathbf{x}}_1(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \quad (5.4.5)$$

$$\dot{\mathbf{x}}_2(t) = \mathbf{f}(\mathbf{x}(t), u(t)) \quad (5.4.6)$$

- Reduce (5.4.5) via IRKA, and (5.4.6) via POD to obtain

$$\dot{\mathbf{x}}_{r1}(t) = \mathbf{A}_{r1}\mathbf{x}_{r1}(t) + \mathbf{b}_{r1}\mathbf{u}(t) \quad (5.4.7)$$

$$\dot{\mathbf{x}}_{r2}(t) = \mathbf{V}^T \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1} \mathbf{f}(\mathbf{P}^T \mathbf{V} \mathbf{x}(t), \mathbf{u}(t)) \quad (5.4.8)$$

- At the k -th step:
 - Evolve (5.4.7) from $\mathbf{x}_{r1}(t_k)$ to $\mathbf{x}_{r1}(t_{k+1})$ to obtain $\mathbf{x}_s$.
 - Lift and reduce $\mathbf{x}_s$ to ensure matching orders.
 - Evolve (5.4.8) from $\mathbf{x}_{r2}(t_k) = \mathbf{x}_{r1}(t_{k+1})$ to $\mathbf{x}_{r2}(t_{k+1})$ to obtain $\mathbf{x}_{k+1}$.
 - Lift and reduce $\mathbf{x}_{k+1}$ to ensure matching orders.
-

5.5 Error Analysis for IPS

In this section we explore error bounds for the error between the full solution and the IPS solution. We know that individually both operator splitting and model reduction are very successful approaches, however, we do not know how they behave together. Let us analyze the error in the state variable. We treat IPS as any numerical method for ODEs and exploit concepts like truncation and global error in our analysis [74, 176].

First, we investigate the truncation and global errors for operator splitting without model reduction in the context of Forward Euler. We use Forward Euler for the sake of simplicity and clarity of presentation.

Definition 5.1. [74, 176] The truncation error of the method Φ (approximate increment per unit step) at the point $(t_k, \mathbf{x}(t_k))$ is defined by

$$T_k(t_k, \mathbf{x}(t_k); h) = \Phi(t_k, \mathbf{x}(t_k); h) - \frac{1}{h}[\mathbf{x}(t_k + h) - \mathbf{x}(t_k)].$$

Theorem 5.2. *Given a full order nonlinear dynamical system as in (5.4.1) where the nonlinearity $\mathbf{f} \in C^3$ is Lipschitz continuous in the first variable $\mathbf{x}$ with Lipschitz constant L_f , let h be the time step size for a Forward Euler operator splitting algorithm that isolates the nonlinearity. Then, the truncation error is*

$$T_k(t_k, \mathbf{x}(t_k); h) = \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{A}\mathbf{x}(t_k) + \mathbf{b}\mathbf{u}(t_k), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) - h\frac{\ddot{\mathbf{x}}(t_k)}{2}) + \mathcal{O}(h^2) \quad (5.5.1)$$

and

$$\|T_k(t_k, \mathbf{x}(t_k); h)\| \leq h(L_f \|\mathbf{A}\mathbf{x}(t_k) + \mathbf{b}\mathbf{u}(t_k)\| + \frac{1}{2} \|\ddot{\mathbf{x}}(t_k)\|) + \mathcal{O}(h^2). \quad (5.5.2)$$

Proof. On the $k + 1$ -th step of Forward Euler with operator splitting we have the following:

$$\begin{aligned}\mathbf{x}_s &= \mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)) \text{ and} \\ \mathbf{x}_{k+1} &= \mathbf{x}_s + h(\mathbf{f}(\mathbf{x}_s, \mathbf{u}(t_k))).\end{aligned}\tag{5.5.3}$$

Combining the two “half steps” in (5.5.3), we infer the $k + 1$ -th step of Forward Euler with operator splitting is implemented as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k) + \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)), \mathbf{u}(t_k))).\tag{5.5.4}$$

This implies that the approximate increment per unit step is

$$\Phi(t_k, \mathbf{x}(t_k); h) = \mathbf{Ax}(t_k) + \mathbf{bu}(t_k) + \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)), \mathbf{u}(t_k)).\tag{5.5.5}$$

The truncation error on the $k + 1$ -th subinterval is given by the equation

$$T_k(t_k, \mathbf{x}(t_k); h) = \Phi(t_k, \mathbf{x}(t_k); h) - \frac{1}{h}[\mathbf{x}(t_k + h) - \mathbf{x}(t_k)].\tag{5.5.6}$$

Using Taylor expansion for $\mathbf{x}(t_k + h)$ we get

$$\mathbf{x}(t_k + h) = \mathbf{x}(t_k) + h\dot{\mathbf{x}}(t_k) + \frac{h^2}{2}\ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^3).\tag{5.5.7}$$

Substituting (5.5.7) into (5.5.6) we get

$$\begin{aligned}T_k(t_k, \mathbf{x}(t_k); h) &= \Phi(t_k, \mathbf{x}(t_k); h) - \frac{1}{h}[\mathbf{x}(t_k) + h\dot{\mathbf{x}}(t_k) + \frac{h^2}{2}\ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^3) - \mathbf{x}(t_k)] \\ &= \Phi(t_k, \mathbf{x}(t_k); h) - \frac{1}{h}[h\dot{\mathbf{x}}(t_k) + \frac{h^2}{2}\ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^3)] \\ &= \Phi(t_k, \mathbf{x}(t_k); h) - \dot{\mathbf{x}}(t_k) - \frac{h}{2}\ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^2).\end{aligned}\tag{5.5.8}$$

Plugging (5.5.5) and (5.4.1) into (5.5.8) we get:

$$\begin{aligned} T_k(t_k, \mathbf{x}(t_k); h) &= \mathbf{Ax}(t_k) + \mathbf{bu}(t_k) + \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)), \mathbf{u}(t_k)) \\ &\quad - (\mathbf{Ax}(t_k) + \mathbf{bu}(t_k) + \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k))) - \frac{h}{2}\ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^2). \end{aligned} \quad (5.5.9)$$

Therefore,

$$T_k(t_k, \mathbf{x}(t_k); h) = \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) - \frac{h}{2}\ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^2). \quad (5.5.10)$$

Since $\mathbf{f}$ is Lipschitz continuous in the first variable $\mathbf{x}$, we have

$$\begin{aligned} &\|\mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k))\| \\ &\leq L_f \|\mathbf{x}(t_k) + h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)) - \mathbf{x}(t_k)\| \\ &= L_f \|h(\mathbf{Ax}(t_k) + \mathbf{bu}(t_k))\| \\ &= hL_f \|\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)\| \end{aligned} \quad (5.5.11)$$

Then, from (5.5.10) and (5.5.11) we infer

$$\|T_k(t_k, \mathbf{x}(t_k); h)\| \leq h \left(L_f \|\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)\| + \frac{1}{2} \|\ddot{\mathbf{x}}(t_k)\| \right) + \mathcal{O}(h^2). \quad (5.5.12)$$

□

For our analysis of IPS we assume we implement operator splitting with model reduction in the context of Forward Euler as well, i.e., in each “half step” we are evolving the reduced linear and nonlinear terms with Forward Euler.

Theorem 5.3. *Let*

$$\alpha_k = \left\| \mathbf{V}_p \mathbf{V}_p^T \mathbf{V}_i (\mathbf{V}_i^T \mathbf{Ax}(t_k) + \mathbf{b}_r \mathbf{u}(t_k)) - (\mathbf{Ax}(t_k) + \mathbf{bu}(t_k)) \right\|$$

be the model reduction error for the linear terms, and

$$\beta_k = \|\mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k))\|$$

the model reduction error for the nonlinear terms, where $\mathbf{V}_i$ is the IRKA basis and $\mathbf{V}_p$ is the POD basis used in Algorithm 7. Also assume the nonlinearity $\mathbf{f} \in C^3$ is Lipschitz continuous in the first variable $\mathbf{x}$. Then, we have

$$\|T_k(t_k, \mathbf{x}(t_k); h)\| \leq \alpha_k + \beta_k + h \left(L_f \|\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)\| + \frac{1}{2} \|\ddot{\mathbf{x}}(t_k)\| \right) + \mathcal{O}(h^2),$$

where h is the time step size.

Proof. The approximate increment per unit step for the IPS algorithm is

$$\begin{aligned} \Phi(t_k, \mathbf{x}(t_k); h) &= \mathbf{V}_p \mathbf{V}_p^T \mathbf{V}_i (\mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{b}_r \mathbf{u}(t_k)) \\ &\quad + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)), \end{aligned} \quad (5.5.13)$$

where $\mathbf{V}_p$ is the POD basis used to reduce the nonlinearity and $\mathbf{V}_i$ is the IRKA basis used to reduce the linear terms. Since the truncation error can be written as

$$\begin{aligned} T_k(t_k, \mathbf{x}(t_k); h) &= \Phi(t_k, \mathbf{x}(t_k); h) - \frac{1}{h} [\mathbf{x}(t_k) + h\dot{\mathbf{x}}(t_k) + \frac{h^2}{2} \ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^3) - \mathbf{x}(t_k)] \\ &= \Phi(t_k, \mathbf{x}(t_k); h) - \frac{1}{h} [h\dot{\mathbf{x}}(t_k) + \frac{h^2}{2} \ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^3)] \\ &= \Phi(t_k, \mathbf{x}(t_k); h) - \dot{\mathbf{x}}(t_k) - \frac{h}{2} \ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^2), \end{aligned} \quad (5.5.14)$$

we infer

$$\begin{aligned} T_k(t_k, \mathbf{x}(t_k); h) &= \mathbf{V}_p \mathbf{V}_p^T \mathbf{V}_i (\mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{b}_r \mathbf{u}(t_k)) - (\mathbf{A} \mathbf{x}(t_k) + \mathbf{b} \mathbf{u}(t_k)) \\ &\quad + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) \\ &\quad - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) - \frac{h}{2} \ddot{\mathbf{x}}(t_k) + \mathcal{O}(h^2). \end{aligned} \quad (5.5.15)$$

Note that for the nonlinear terms in (5.5.15) we have

$$\begin{aligned}
& \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \\
&= \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) \\
&\quad - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) - \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \\
&= \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) \\
&\quad - \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)).
\end{aligned} \tag{5.5.16}$$

By the triangle inequality we have

$$\begin{aligned}
& \left\| \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right. \\
& \quad \left. + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\| \leq \left\| \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) \right. \\
& \quad \left. - \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\| + \left\| -\mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\|.
\end{aligned} \tag{5.5.17}$$

Furthermore,

$$\begin{aligned}
& \left\| \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\| \\
&= \left\| \mathbf{V}_p \mathbf{V}_p^T (\mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k))) \right\| \\
&\leq \left\| \mathbf{V}_p \mathbf{V}_p^T \right\| \left\| \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\|.
\end{aligned} \tag{5.5.18}$$

Since $\mathbf{V}_p$ is orthonormal,

$$\begin{aligned}
& \left\| \mathbf{V}_p \mathbf{V}_p^T (\mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k))) \right\| \\
&\leq \left\| \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\|.
\end{aligned} \tag{5.5.19}$$

The Lipschitz continuity of $\mathbf{f}$ implies

$$\begin{aligned}
& \left\| \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\| \\
&\leq L_f \left\| h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)) \right\| \\
&= h L_f \left\| \mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k) \right\|.
\end{aligned} \tag{5.5.20}$$

Plugging (5.5.20) into (5.5.17) and substituting β_k for

$$\left\| -\mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\| ,$$

we obtain

$$\begin{aligned} & \left\| \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k)), \mathbf{u}(t_k)) - \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right. \\ & \quad \left. - \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{x}(t_k), \mathbf{u}(t_k)) \right\| \leq \beta_k + hL_f \left\| \mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k) \right\|. \end{aligned} \quad (5.5.21)$$

Thus,

$$\begin{aligned} \|T_k(t_k, \mathbf{x}(t_k); h)\| & \leq \alpha_k + \beta_k + hL_f \left\| \mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k) \right\| + \frac{h}{2} \|\ddot{\mathbf{x}}(t_k)\| + \mathcal{O}(h^2) \\ & = \alpha_k + \beta_k + h \left(L_f \left\| \mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k) \right\| + \frac{1}{2} \|\ddot{\mathbf{x}}(t_k)\| \right) + \mathcal{O}(h^2) \end{aligned}$$

□

We have determined the local truncation error of the IPS method. However, in order to determine the global error, we need to establish that Φ is Lipschitz continuous, and if so, to find its Lipschitz constant.

Remark 5.4. The assumption of Lipschitz continuity for the nonlinear function $\mathbf{f}$ is a reasonable one, given that the nonlinearities appearing in most applications are Lipschitz continuous.

Theorem 5.5. *If the nonlinearity $\mathbf{f}$ is a Lipschitz continuous function with Lipschitz constant L_f , then Φ is also a Lipschitz continuous function with Lipschitz constant L_Φ where*

$$L_\Phi = \|\mathbf{A}\| + L_f \left\| \mathbf{I} + h\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \right\|. \quad (5.5.22)$$

Proof. For simplicity of notation let $\mathbf{y}_k = \mathbf{x}(t_k) + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t_k))$. Then,

$$\begin{aligned} \Phi(t_k, \mathbf{x}(t_k); h) &= \mathbf{V}_p \mathbf{V}_p^T \mathbf{V}_i (\mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{b}_r \mathbf{u}(t_k)) + \mathbf{V}_p \mathbf{V}_p^T \mathbf{f}(\mathbf{y}_k, \mathbf{u}(t_k)) \\ &= \mathbf{V}_p \mathbf{V}_p^T (\mathbf{V}_i (\mathbf{V}_i^T \mathbf{A} \mathbf{x}(t_k) + \mathbf{b}_r \mathbf{u}(t_k)) + \mathbf{f}(\mathbf{y}_k, \mathbf{u}(t_k))). \end{aligned}$$

Note that

$$\begin{aligned} \Phi(t_k, \mathbf{x}_\alpha; h) - \Phi(t_k, \mathbf{x}_\beta; h) &= \mathbf{V}_p \mathbf{V}_p^T (\mathbf{V}_i (\mathbf{V}_i^T \mathbf{A} \mathbf{x}_\alpha) + \mathbf{f}(\mathbf{y}_\alpha, \mathbf{u}(t_k)) - \mathbf{V}_i (\mathbf{V}_i^T \mathbf{A} \mathbf{x}_\beta) - \mathbf{f}(\mathbf{y}_\beta, \mathbf{u}(t_k))) \\ &= \mathbf{V}_p \mathbf{V}_p^T (\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}_\alpha - \mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}_\beta) + \mathbf{f}(\mathbf{y}_\alpha, \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{y}_\beta, \mathbf{u}(t_k)) \\ &= \mathbf{V}_p \mathbf{V}_p^T (\mathbf{V}_i \mathbf{V}_i^T (\mathbf{A} \mathbf{x}_\alpha - \mathbf{A} \mathbf{x}_\beta)) + \mathbf{f}(\mathbf{y}_\alpha, \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{y}_\beta, \mathbf{u}(t_k)). \end{aligned}$$

Since $\mathbf{V}_p$ is an orthonormal POD basis we have,

$$\begin{aligned} \|\Phi(t, \mathbf{x}_\alpha; h) - \Phi(t, \mathbf{x}_\beta; h)\| &\leq \|\mathbf{V}_p \mathbf{V}_p^T\| \|(\mathbf{V}_i \mathbf{V}_i^T (\mathbf{A} \mathbf{x}_\alpha - \mathbf{A} \mathbf{x}_\beta)) + \mathbf{f}(\mathbf{y}_\alpha, \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{y}_\beta, \mathbf{u}(t_k))\| \\ &= \|\mathbf{V}_i \mathbf{V}_i^T (\mathbf{A} \mathbf{x}_\alpha - \mathbf{A} \mathbf{x}_\beta) + \mathbf{f}(\mathbf{y}_\alpha, \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{y}_\beta, \mathbf{u}(t_k))\|. \end{aligned}$$

By the triangle inequality we have

$$\|\Phi(t, \mathbf{x}_\alpha; h) - \Phi(t, \mathbf{x}_\beta; h)\| \leq \|\mathbf{V}_i \mathbf{V}_i^T (\mathbf{A} \mathbf{x}_\alpha - \mathbf{A} \mathbf{x}_\beta)\| + \|\mathbf{f}(\mathbf{y}_\alpha, \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{y}_\beta, \mathbf{u}(t_k))\|. \quad (5.5.23)$$

Since $\mathbf{V}_i$ is an orthonormal IRKA basis

$$\begin{aligned} \|\mathbf{V}_i \mathbf{V}_i^T (\mathbf{A} \mathbf{x}_\alpha - \mathbf{A} \mathbf{x}_\beta)\| &\leq \|\mathbf{A} \mathbf{x}_\alpha - \mathbf{A} \mathbf{x}_\beta\| \\ &\leq \|\mathbf{A}\| \|\mathbf{x}_\alpha - \mathbf{x}_\beta\|. \end{aligned} \quad (5.5.24)$$

Since the nonlinearity $\mathbf{f}$ is Lipschitz, we have

$$\|\mathbf{f}(\mathbf{y}_\alpha, \mathbf{u}(t_k)) - \mathbf{f}(\mathbf{y}_\beta, \mathbf{u}(t_k))\| \leq L_f \|\mathbf{y}_\alpha - \mathbf{y}_\beta\|, \quad (5.5.25)$$

where $L_{\mathbf{f}}$ is the Lipschitz constant for $\mathbf{f}$. Further,

$$\begin{aligned}
\|\mathbf{y}_\alpha - \mathbf{y}_\beta\| &= \|\mathbf{x}_\alpha + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}_\alpha + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t)) - \mathbf{x}_\beta - h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}_\beta + \mathbf{V}_i \mathbf{b}_r \mathbf{u}(t))\| \\
&= \|\mathbf{x}_\alpha + h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}_\alpha) - \mathbf{x}_\beta - h(\mathbf{V}_i \mathbf{V}_i^T \mathbf{A} \mathbf{x}_\beta)\| \\
&= \|(\mathbf{I} + h \mathbf{V}_i \mathbf{V}_i^T \mathbf{A})(\mathbf{x}_\alpha - \mathbf{x}_\beta)\| \\
&\leq \|\mathbf{I} + h \mathbf{V}_i \mathbf{V}_i^T \mathbf{A}\| \|\mathbf{x}_\alpha - \mathbf{x}_\beta\|.
\end{aligned} \tag{5.5.26}$$

Plugging (5.5.26) into (5.5.25), and then substituting (5.5.24) and (5.5.25) into (5.5.23), we obtain

$$\|\Phi(t, \mathbf{x}_\alpha; h) - \Phi(t, \mathbf{x}_\beta; h)\| \leq \|\mathbf{A}\| \|\mathbf{x}_\alpha - \mathbf{x}_\beta\| + L_{\mathbf{f}} \|\mathbf{I} + h \mathbf{V}_i \mathbf{V}_i^T \mathbf{A}\| \|\mathbf{x}_\alpha - \mathbf{x}_\beta\|. \tag{5.5.27}$$

As a result,

$$\|\Phi(t, \mathbf{x}_\alpha; h) - \Phi(t, \mathbf{x}_\beta; h)\| \leq L_\Phi \|\mathbf{x}_\alpha - \mathbf{x}_\beta\|, \tag{5.5.28}$$

where

$$L_\Phi = \|\mathbf{A}\| + L_{\mathbf{f}} \|\mathbf{I} + h \mathbf{V}_i \mathbf{V}_i^T \mathbf{A}\|. \tag{5.5.29}$$

Thus, Φ is Lipschitz, with Lipschitz constant L_Φ . □

Therefore, the global error for operator splitting with model reduction is

$$\|e_k\| \leq \frac{\|T_k\|}{L_\Phi} (e^{L_\Phi(t_k - t_0)} - 1) \tag{5.5.30}$$

where L_Φ is the Lipschitz constant for Φ and t_k the final time. We refer the reader to [74, 176] for more details on the concepts of global error, truncation error and the relation between them.

The global error $\|e_k\|$ in (5.5.30) provides an upper bound for the error in the state variable, i.e., $\|\mathbf{x} - \mathbf{V} \mathbf{x}_r\| \leq \|e_k\|$. We are also interested in an error bound for the output. Let $\mathbf{y}$ be

the output obtained by solving the system via Forward Euler and $\mathbf{y}_r$ the output when we solve the system via IPS. We have

$$\begin{aligned}\|\mathbf{y} - \mathbf{y}_r\| &= \|\mathbf{C}\mathbf{x} - \mathbf{C}\mathbf{V}\mathbf{x}_r\| \\ &\leq \|\mathbf{C}\| \|\mathbf{x} - \mathbf{V}\mathbf{x}_r\|.\end{aligned}$$

As a result

$$\|\mathbf{y} - \mathbf{y}_r\| \leq \|\mathbf{C}\| \|e_k\|. \quad (5.5.31)$$

As expected, IPS is not, and it cannot be, a consistent numerical method since model reduction introduces inaccuracies that are independent of the step size. Even as h approaches zero, the model reduction error is still present. However, if we have a small model reduction error, we can obtain an accurate approximate solution for a system of ordinary differential equations by picking a small step size. Since both system theoretic methods and POD generate high fidelity reduced models, we are confident our results will yield accurate approximations. Next, we investigate numerically the effects of the order of the reduced model and of the time-step choice.

5.6 Numerical Results

In this section we investigate the IRKA-POD splitting algorithm numerically on several nonlinear models. As described previously in this chapter, the implemented operator splitting first evolves the linear terms, and then uses the result to evolve the nonlinear terms. Even though, our error analysis was conducted in the context of Forward Euler, due to the stiffness of some of the problems we use the Backward Euler method to numerically integrate the ODE. Even though the reduced linear terms and the reduced nonlinearity can be of different

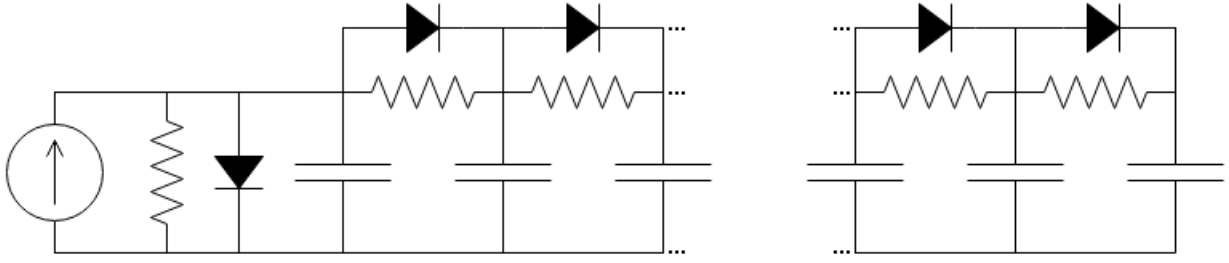


Figure 5.4: RC Ladder Circuit [177]

reduced orders, in our numerical experiments below they are of the same order r for simplicity and clarity of comparisons.

5.6.1 Nonlinear RC Ladder

The nonlinear RC-ladder is an electronic test circuit [177]. This system models a resistor-capacitor network that exhibits a distinct nonlinear behavior caused by nonlinear resistors. The nonlinearity models a diode as nonlinear resistor similar to the Shockley model [155]. The nonlinear RC ladder system is structured as follows:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) - e^{40\mathbf{A}_0\mathbf{x}(t)} + e^{40\mathbf{A}_1\mathbf{x}(t)} - e^{40\mathbf{A}_2\mathbf{x}(t)} + 1 \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t).\end{aligned}\tag{5.6.1}$$

The order of the original model is $n = 400$. The 2-norm condition number of the matrix $\mathbf{A}$ is $3.208 \cdot 10^5$ and Figure 5.5 shows the plot of the 2-norm condition number of the Jacobian of the nonlinearity $\mathbf{f}$ at given time steps.

Using IRKA to reduce the linear terms, we get

$$\dot{\mathbf{x}}_r(t) = \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{B}_r\mathbf{u}(t),\tag{5.6.2}$$

and the 2-norm condition number of $\mathbf{A}_r$ is $3.5213 \cdot 10^3$.

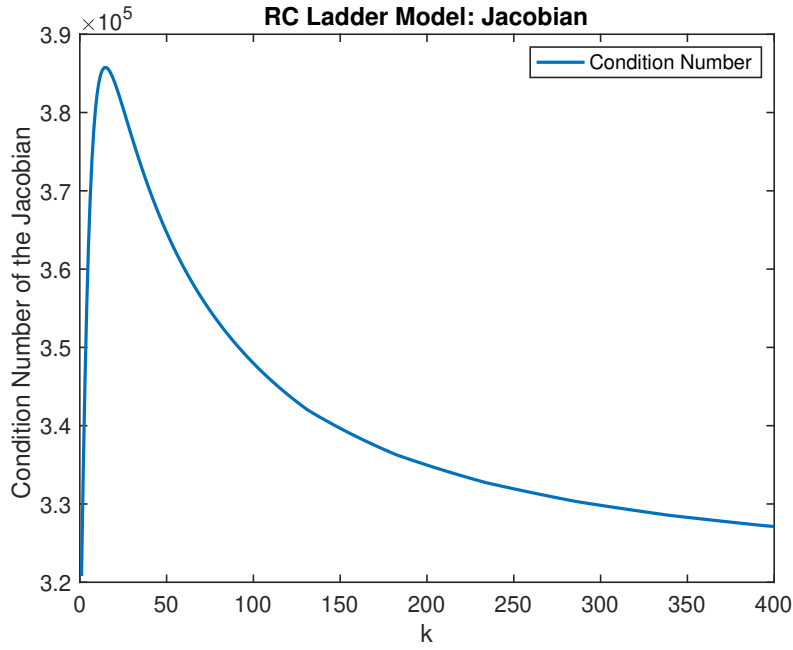


Figure 5.5: Jacobian of the Nonlinearity

POD generates the following reduced nonlinearity:

$$\dot{\mathbf{x}}_p(t) = \mathbf{V}^T(e^{40\mathbf{A}_0}\mathbf{V}\mathbf{x}(t) + e^{40\mathbf{A}_1}\mathbf{V}\mathbf{x}(t) - e^{40\mathbf{A}_2}\mathbf{V}\mathbf{x}(t) + 1) \quad (5.6.3)$$

where $\mathbf{V}$ is a POD basis. If we reduce (5.6.1) using only POD, we obtain:

$$\begin{aligned} \dot{\mathbf{x}}_p(t) &= \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\mathbf{u}(t) - \mathbf{V}^T(e^{40\mathbf{A}_0}\mathbf{V}\mathbf{x}(t) + e^{40\mathbf{A}_1}\mathbf{V}\mathbf{x}(t) - e^{40\mathbf{A}_2}\mathbf{V}\mathbf{x}(t) + 1) \\ \mathbf{y}_p(t) &= \mathbf{C}_p\mathbf{x}_p(t). \end{aligned} \quad (5.6.4)$$

Any time we reduced a system or a nonlinearity via POD we approximated the nonlinear terms via DEIM to make our computations more efficient. For all the numerical experiments with the RC Ladder model POD was trained with the input $\mathbf{u}(t) = e^{-t}$ and the snapshots were generated via Backward Euler.

First, we show some numerical results where we compare the solution of the reduced order model obtained via a combination of IRKA and POD and the solution of the reduced order

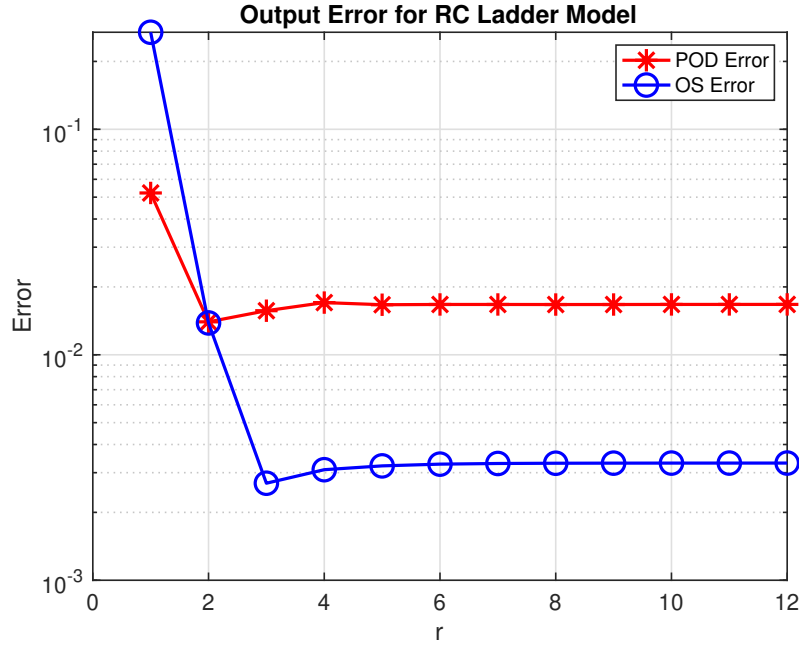


Figure 5.6: Output Error: IPS vs POD

model obtained using solely POD, for both the linear and nonlinear parts. Specifically, we compare $\|\mathbf{y} - \mathbf{y}_r\|_2$ with $\|\mathbf{y} - \mathbf{y}_p\|_2$ and $\|\mathbf{x} - \mathbf{V}_p \mathbf{x}_r\|_2$ with $\|\mathbf{x} - \mathbf{V}_p \mathbf{x}_p\|_2$. The outputs $\mathbf{y}_r$, $\mathbf{y}_p$ and the states $\mathbf{x}_r$, $\mathbf{x}_p$ are obtained after solving the reduced systems via operator splitting.

For the comparison between IPS and POD we used $N = 100$ steps for the Backward Euler method. Figure 5.6 and Figure 5.7 show the results of the comparisons for various values of r . Even though POD appears to do better for very small values of r , IPS clearly outperforms POD when $r \geq 4$. The control input we use for the comparisons is the same as the control input we use for training POD, so these comparisons are biased towards POD.

Next, we explore how the order of the reduced model and the step size influence the accuracy of IPS. Figure 5.10 illustrates the accuracy of IPS for the reduced order $r = 8$ and the time step size $h = 0.0025$ for the RC Ladder model through the plots of the impulse responses obtained via Backward Euler and IPS. For this case, we also computed the values of α_k and β_k from the analysis in Section 5.5 for each step of the method. These values are plotted in Figures 5.8 and 5.9. As anticipated, the values of α_k and β_k are small.

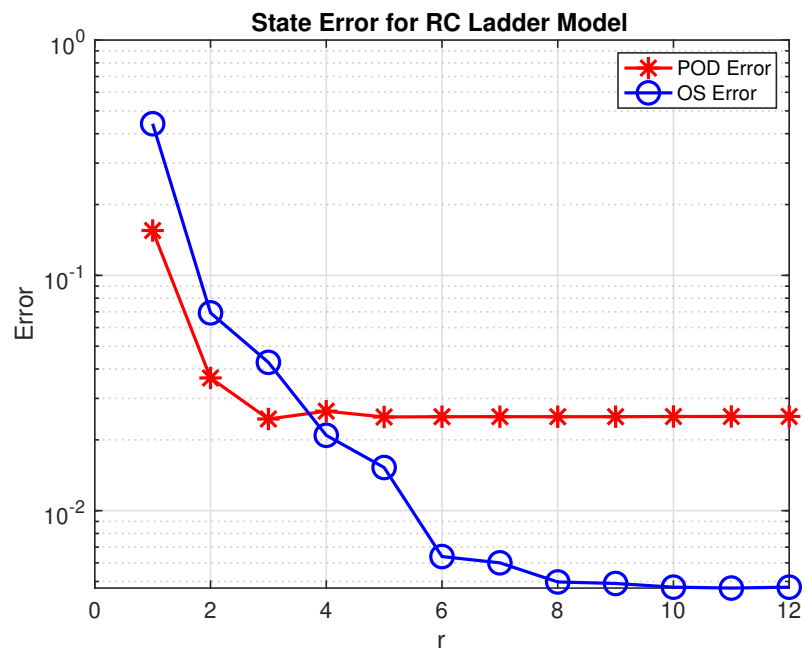


Figure 5.7: State Error: IPS vs POD

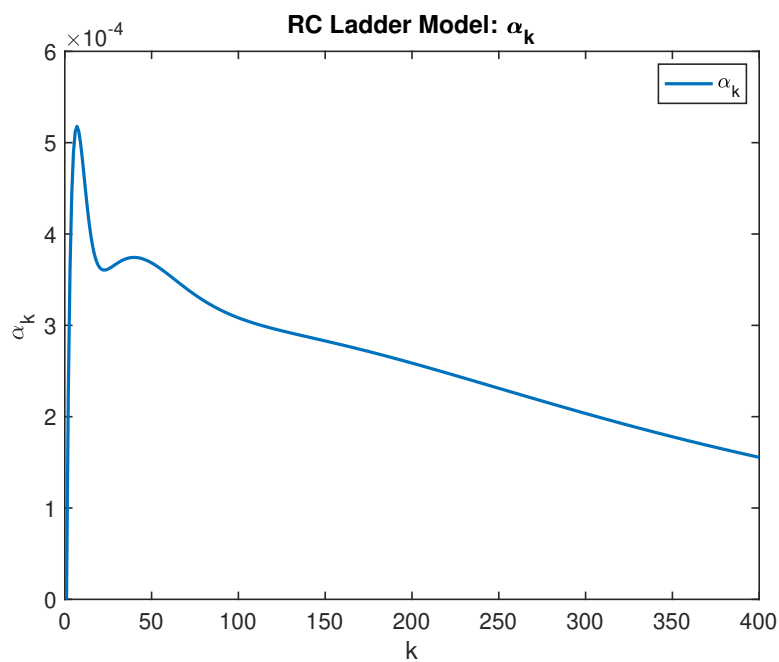


Figure 5.8: ROM Error on the Linear Terms

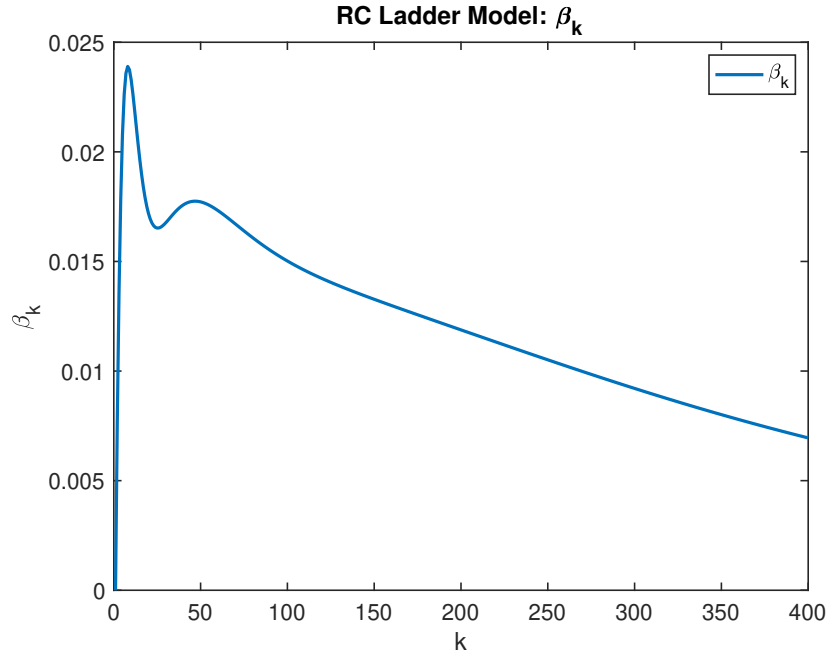
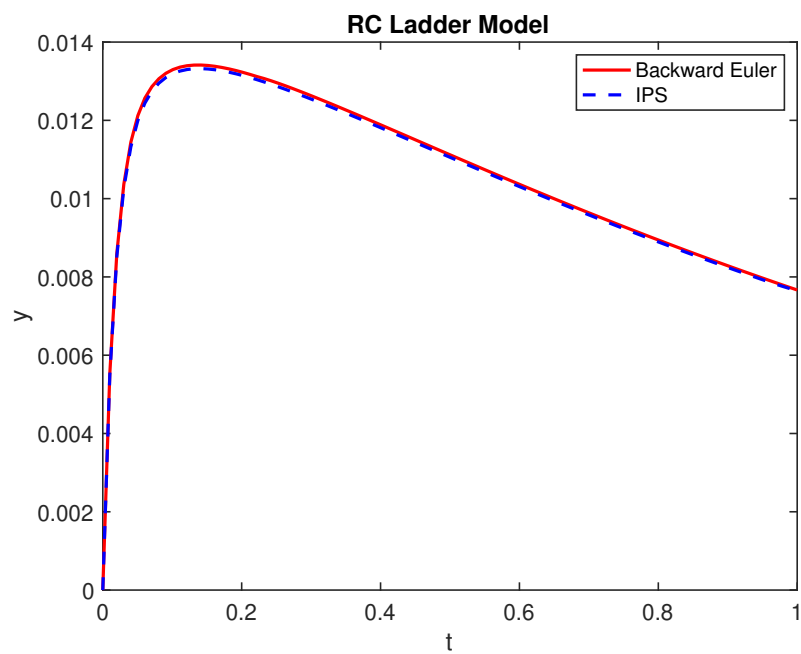
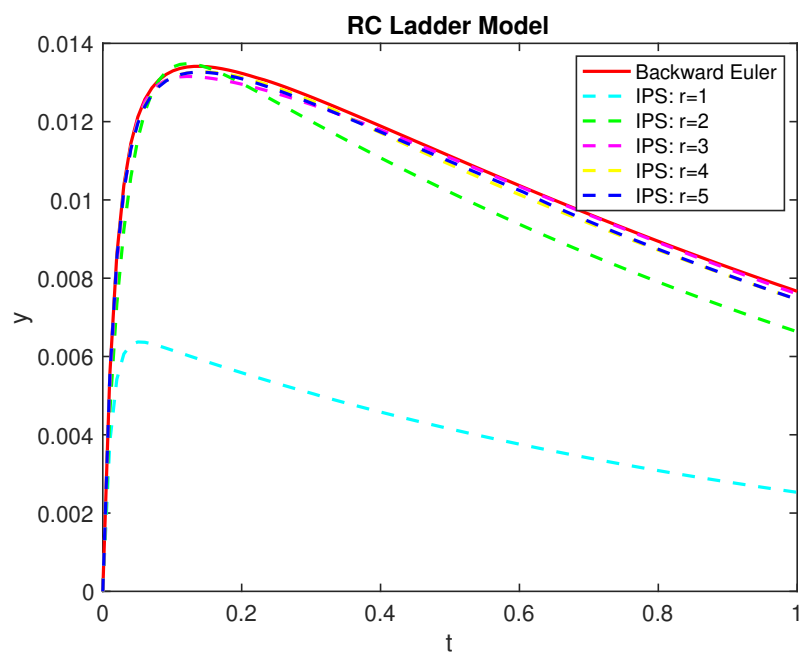
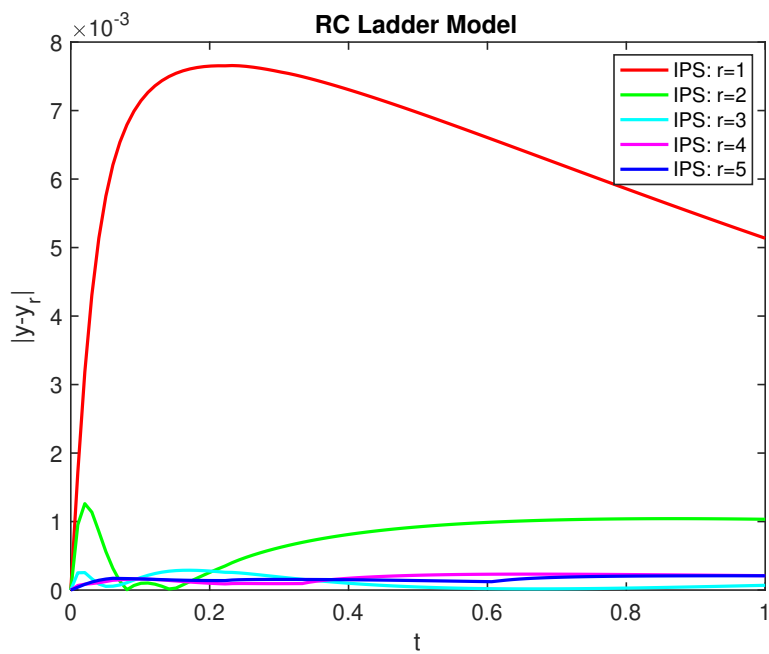
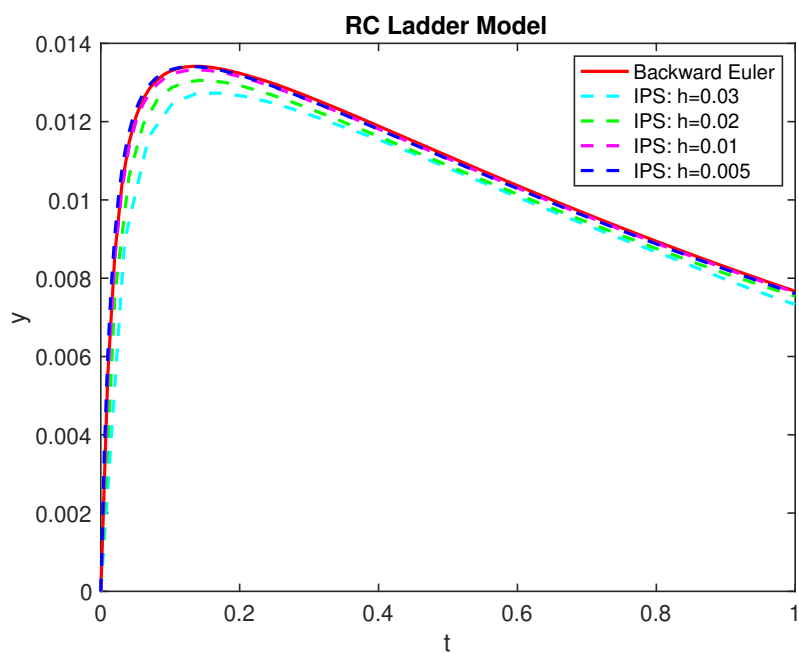
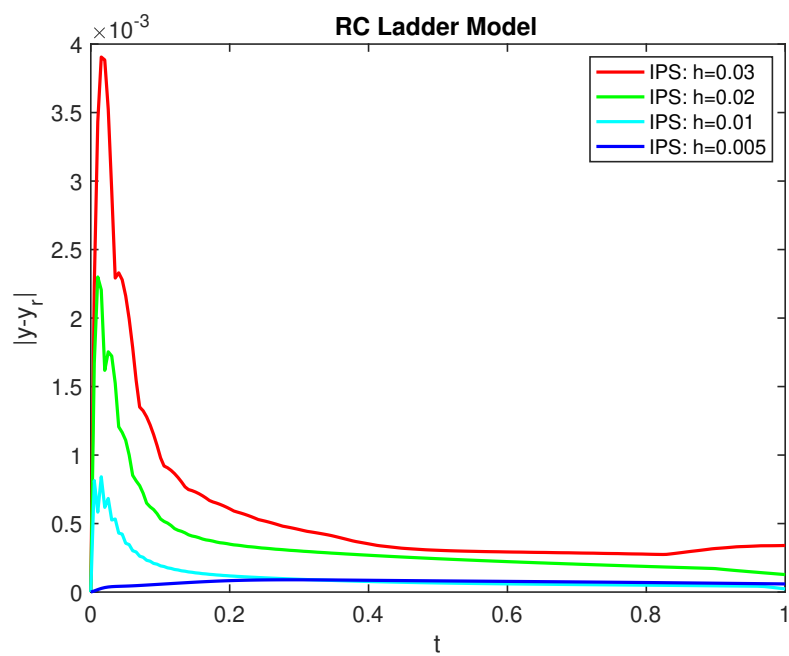
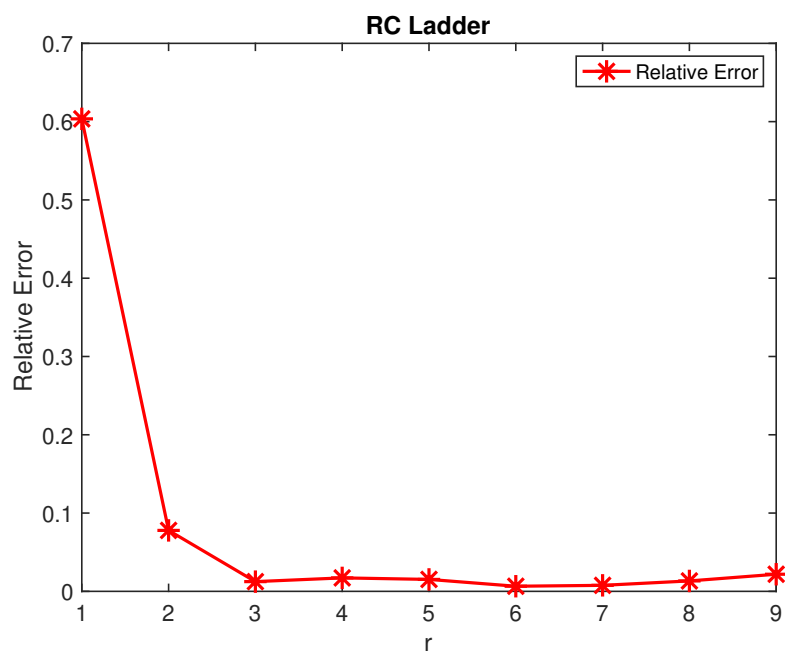


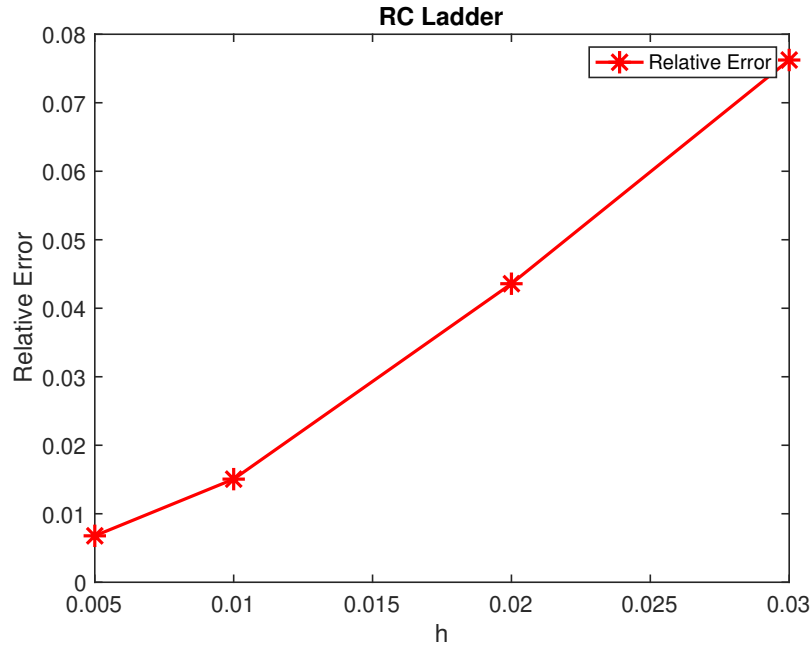
Figure 5.9: ROM Error on the Nonlinear Terms

Figure 5.11 shows various impulse responses for various r values as we keep the time step $h = 0.01$ constant, and Figure 5.12 illustrates the same results with the output error plots. As expected, a higher order reduced model produces a more accurate solution. The impulse responses plotted in Figure 5.13 and the error plots in Figure 5.14 show the outcomes of numerical experiments where we fix the order of the reduced order model at $r = 6$. In Figure 5.15 we plot the values of the output error $\|\mathbf{y} - \mathbf{y}_r\|_2$ over the time interval $[0, 1]$ for a constant value of $h = 0.01$ and varying values of r . Figure 5.16 depicts the relationship between the output error $\|\mathbf{y} - \mathbf{y}_r\|_2$ for a fixed value of $r = 6$ and changing values of h . As we observe, the error diminishes as r increases and h decreases. These results suggest we could play some balancing act between the time step and the order of the reduced model in order to maximize the accuracy of the solution and minimize the cost of the computations.

Figure 5.10: IPS vs Backward Euler; $r = 8$, $h = 0.0025$ Figure 5.11: IPS vs Backward Euler; $h = 0.01$

Figure 5.12: IPS Errors; $h = 0.01$ Figure 5.13: IPS vs Backward Euler; $r = 6$

Figure 5.14: IPS Errors; $r = 6$ Figure 5.15: Error vs r ; $h = 0.01$

Figure 5.16: Error vs h ; $r = 6$

5.6.2 Chafee-Infante Model

The one-dimensional Chafee-Infante system is a diffusion-reaction model first introduced in [50] and has been used as a benchmark for nonlinear model reduction [28, 34, 35]. The system is described by the following equations:

$$\begin{aligned} \frac{dv}{dt} + v^3 &= \frac{\partial^2 v}{\partial x^2} + u, & (x, t) \in (0, 1) \times (0, T), & \quad v(0, T) = u(t), \quad t \in (0, T), \\ \frac{dv}{dx}(1, t) &= 0, & t \in (0, T), & \quad v(x, 0) = 0, \quad x \in (0, 1). \end{aligned} \quad (5.6.5)$$

We are interested in the output at the right boundary. After discretizing this system using 250 grid points, we obtain the nonlinear dynamical system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) + \mathbf{f}(\mathbf{x}(t)), \\ \mathbf{y}(t) &= \mathbf{c}\mathbf{x}(t), \end{aligned} \quad (5.6.6)$$

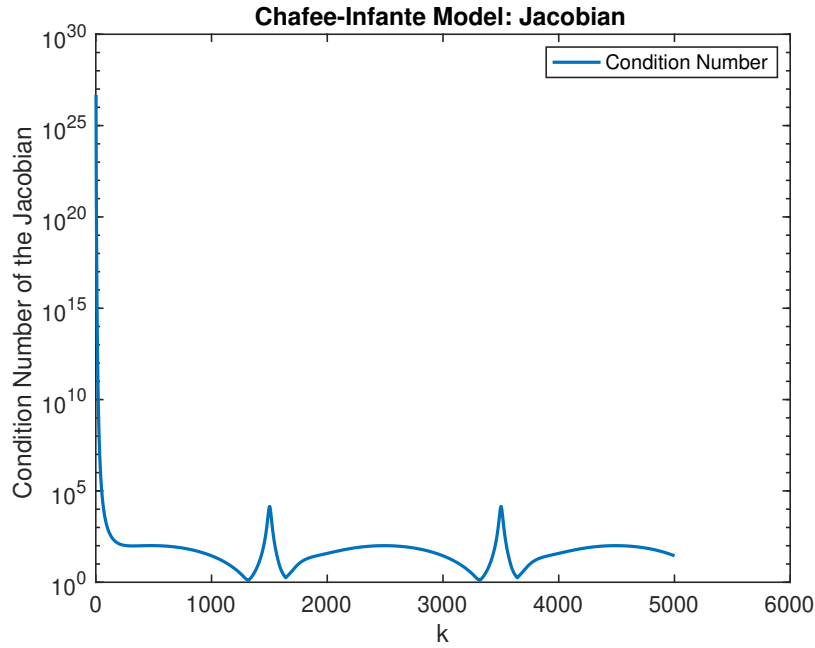


Figure 5.17: Jacobian of the Nonlinearity

where the nonlinearity $\mathbf{f}(\mathbf{x}(t))$ is described by the element-wise third power of the state vector $\mathbf{x}(t)$. The 2-norm condition number of the full order matrix $\mathbf{A}$ and the IRKA-generated reduced order matrix $\mathbf{A}_r$ are $1.7058 \cdot 10^5$ and $1.5265 \cdot 10^5$, respectively. In Figure 5.17, we plot the 2-norm condition number of the Jacobian of the nonlinearity $\mathbf{f}$ versus the given time steps.

Similarly to the RC Ladder system, we use the Chafee-Infante model to numerically investigate the connection between the time step h , the reduced order r , and the error between the true and the approximate solution obtained via IPS. All the simulations for this model were performed on the time interval $[0, 5]$. For this system we trained POD with the input $\mathbf{u}(t) = 10(\sin(\pi t) + 1)$ and the snapshot simulations were generated via Backward Euler.

In Figure 5.18, we observe that IPS captures the solution of the Chafee-Infante system very accurately for the time step size $h = 0.0001$ and the order of the reduced model $r = 16$. The impulse response plots in Figure 5.19 and the error plots in Figure 5.20 illustrate how the error decreases when we fix the time step size $h = 0.0001$ and increase the order of the

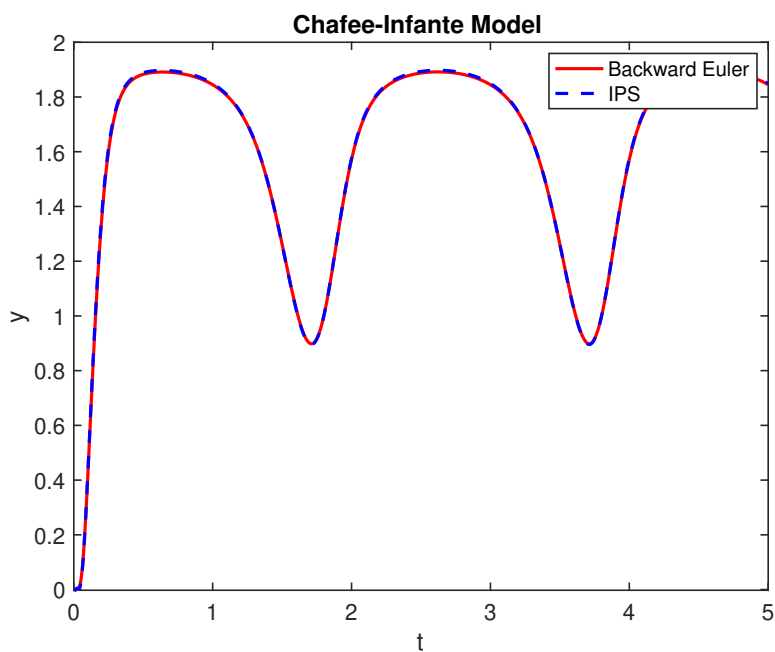
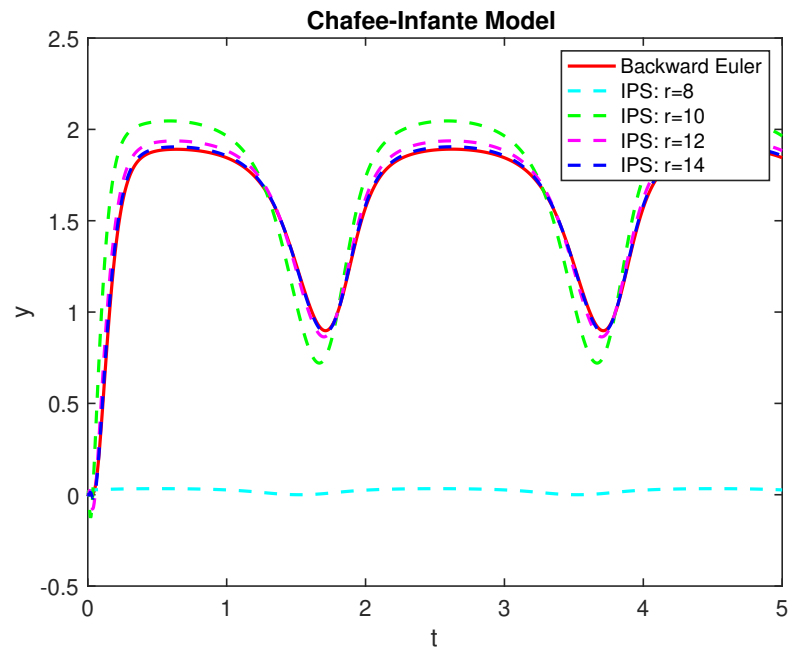
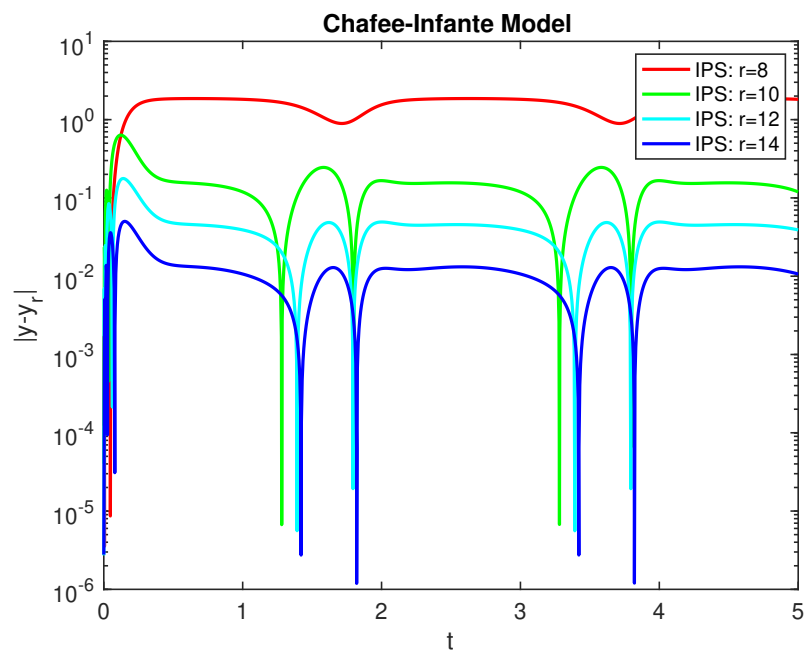
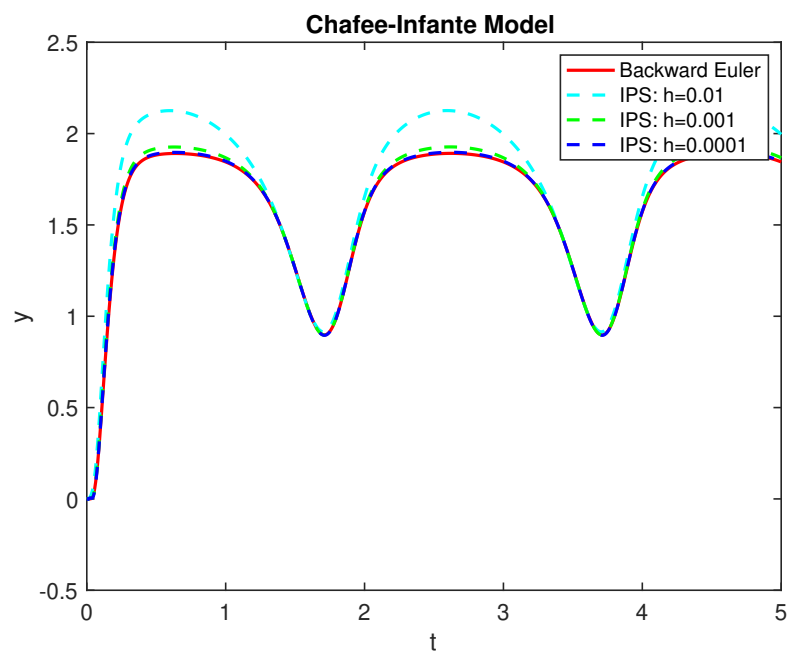
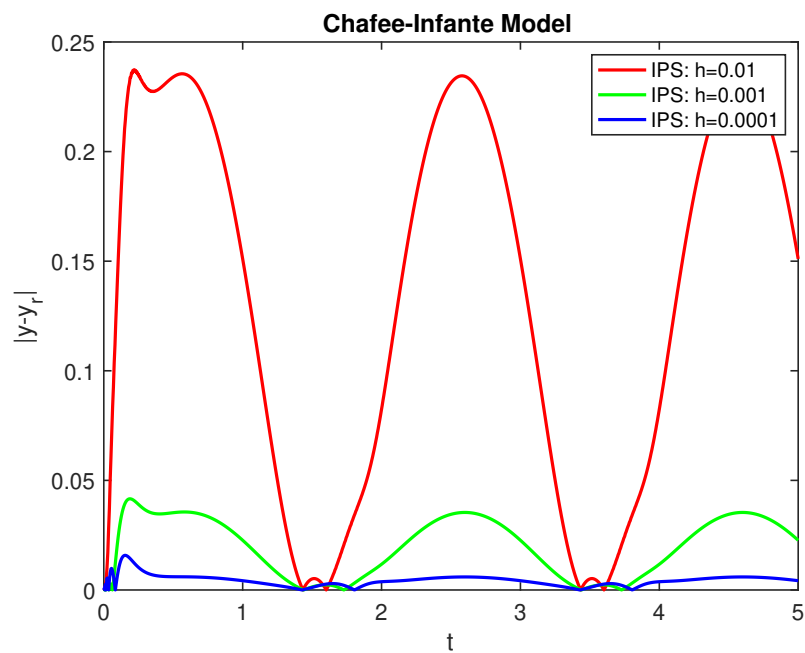
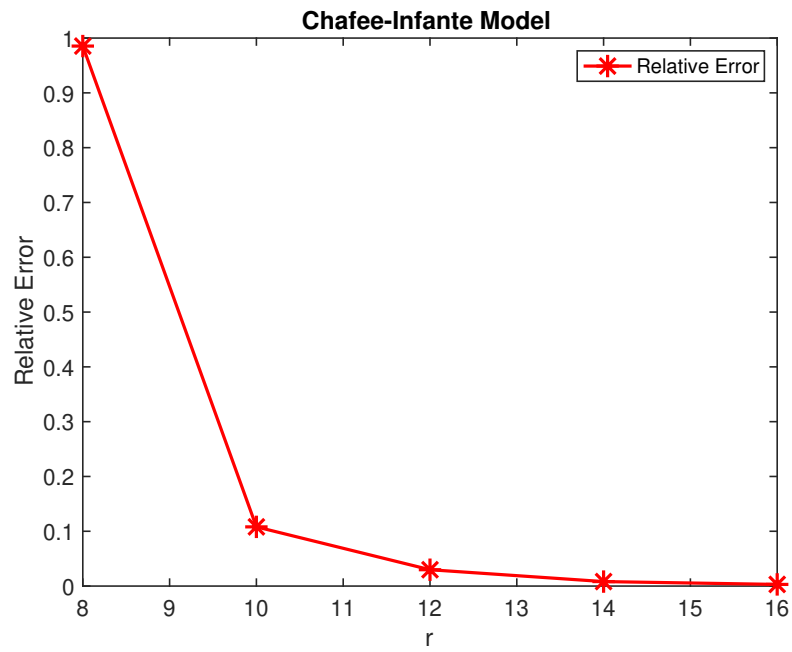
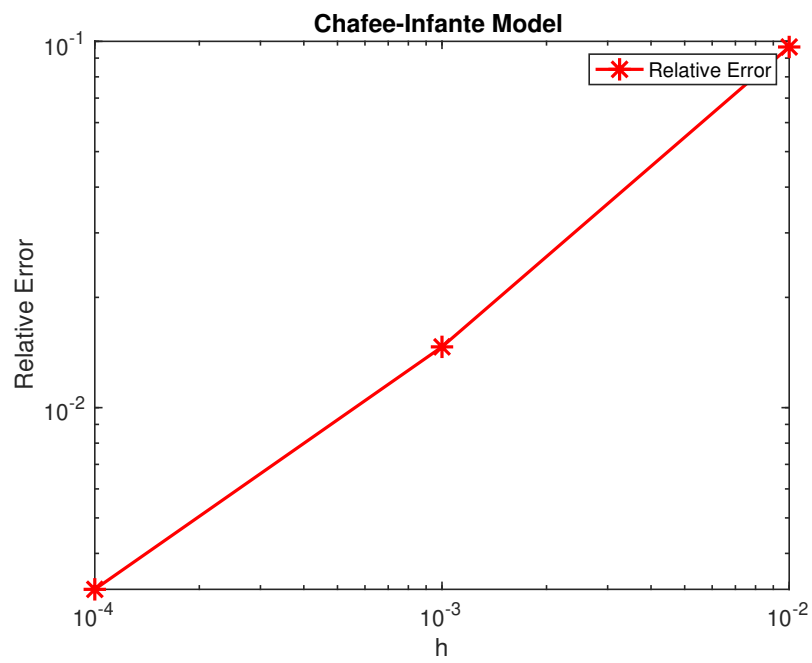


Figure 5.18: IPS vs Backward Euler; $r = 16$, $h = 0.0001$

reduced model from $r = 8$ to $r = 16$ in increments of 2. In Figures 5.21 and 5.22, the plots show the dependence of the error on the time step h as we keep the order of the reduced model fixed to $r = 16$. In Figures 5.23 and 5.24 we plot the output errors $\|\mathbf{y} - \mathbf{y}_r\|_2$ versus the values of r and h , respectively. Since r and h affect the accuracy simultaneously, we can choose to vary their values according to the problem at hand.

Figure 5.19: IPS vs Backward Euler; $h = 0.0001$ Figure 5.20: IPS Errors; $h = 0.0001$

Figure 5.21: IPS vs Backward Euler; $r = 16$ Figure 5.22: IPS Errors; $r = 16$

Figure 5.23: Error vs r ; $h = 0.0001$ Figure 5.24: Error vs h ; $r = 16$

5.6.3 Warning: Tubular Reactor

While IPS can be very successful with systems such as the RC Ladder and the Chafee-Infante model from the previous sections, it does not always work. To illustrate this point we describe a non-adiabatic tubular reactor system with a single reaction that models the evolution of the concentration $\psi(x, t)$ and temperature $\theta(x, t)$ [35, 94]. The system is governed by the following PDEs:

$$\begin{aligned}\frac{\partial \psi}{\partial t} &= \frac{1}{Pe} \frac{\partial^2 \psi}{\partial x^2} - \frac{\partial \psi}{\partial x} - \mathcal{D}\mathcal{F}(\psi, \theta; \gamma); \\ \frac{\partial \theta}{\partial t} &= \frac{1}{Pe} \frac{\partial^2 \theta}{\partial x^2} - \frac{\partial \theta}{\partial x} - \beta(\theta - \theta_{ref}) + \mathcal{B}\mathcal{D}\mathcal{F}(\psi, \theta; \gamma),\end{aligned}$$

where $x \in (0, 1)$, $t > 0$, the Damkohler number $\mathcal{D} = 0.167$, the Peclet number $Pe = 5$, the reaction rate $\gamma = 25$, the constants $\mathcal{B} = 0.5$, $\beta = 2.5$, the reference temperature $\theta_{ref}(x, t) = 1$, and the Arrhenius reaction term is

$$\mathcal{F}(\psi, \theta; \gamma) = \psi \exp\left(\gamma - \frac{\gamma}{\theta}\right). \quad (5.6.7)$$

We impose the following boundary conditions:

$$\begin{aligned}\frac{\partial \psi}{\partial x}(0, t) &= Pe(\psi(0, t) - 1), & \frac{\partial \theta}{\partial x}(0, t) &= Pe(\theta(0, t) - 1), \\ \frac{\partial \psi}{\partial x}(1, t) &= 0, & \frac{\partial \theta}{\partial x}(1, t) &= 0.\end{aligned}$$

The initial conditions are given as $\psi(x, 0) = \psi_0(x)$ and $\theta(x, 0) = \theta_0(x)$. Discretizing this PDE model we obtain the following ODE system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B} + \mathbf{f}(\mathbf{x}(t)), \quad (5.6.8)$$

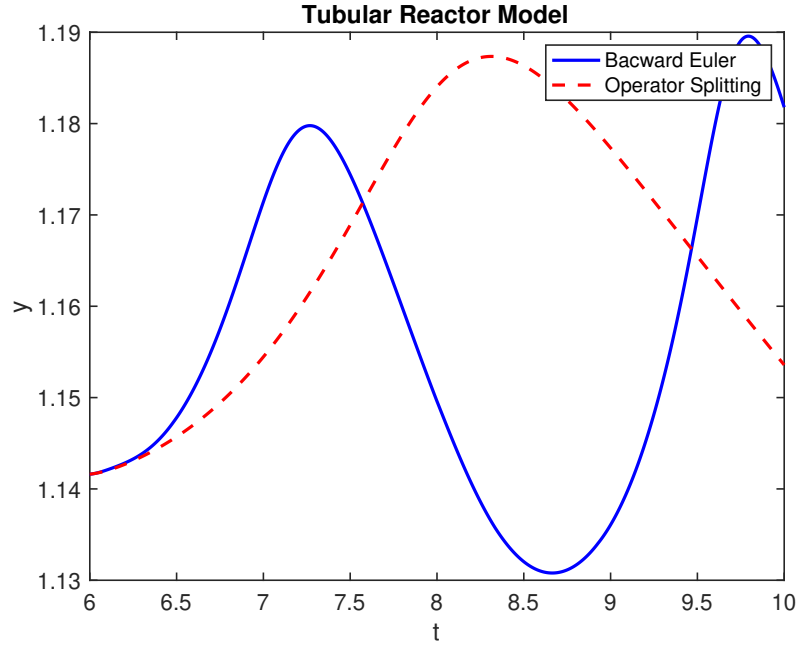


Figure 5.25: Operator Splitting vs Backward Euler

where

$$\mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} -0.167\mathbf{x}_1(t)e^{25-\frac{25}{\mathbf{x}_2(t)}} \\ 0.0835\mathbf{x}_1(t)e^{25-\frac{25}{\mathbf{x}_2(t)}} \end{bmatrix}$$

and

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix}.$$

Note that the input of the dynamical system (5.6.8) is constant, i.e., $\mathbf{u}(t) = 1$. The nonlinearity comes from the Arrhenius term and it requires pointwise evaluations.

The problem with IPS for this system is with the splitting of the terms rather than model reduction. Even when we attempted to solve the full system with operator splitting, we could not obtain an accurate solution for various splitting schemes. We speculate the issue is structural. Since problem is near a bifurcation, it is possible the operator splitting ap-

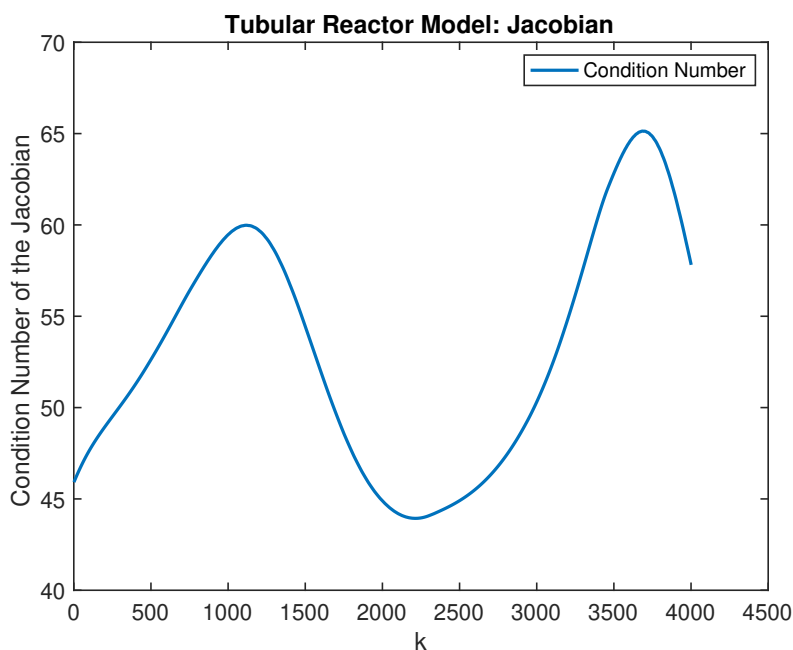


Figure 5.26: Condition Number of the Jacobian of the Nonlinearity

proximation is very different from the trajectory yielded by the simulation of the coupled problem. Furthermore, Figure 5.25 shows that operator splitting is capturing the true solution in the beginning, but deviates as soon as the true solution starts to rapidly increase. In summary, for the problems where operator splitting produced accurate, we could incorporate model reduction. If operator splitting failed, then IPS cannot succeed.

Chapter 6

Conclusions and Outlook

In this dissertation we have explored model reduction of linear systems on a finite horizon, and the integration of system theoretic methods like Balanced Truncation and IRKA with trajectory based techniques like POD.

First, we reviewed existing model reduction techniques for linear systems that produce high fidelity reduced models on an infinite interval. Then, we established a framework for locally optimal reduced order modeling by deriving interpolation based conditions on a finite time horizon. Based on the derived $\mathcal{H}_2(t_f)$ optimality conditions, we constructed a descent algorithm, FHIRKA, that produces a high fidelity reduced order model upon convergence. Furthermore, FHIRKA reduces unstable systems optimally in a finite horizon. Our numerical experiments further supported our theoretical results and showed that FHIRKA outperforms many existing methods such as POD, Time-Limited Balanced Truncation, and an IRKA-type time-limited algorithm based on Sylvester equations.

These results have spawned many interesting questions that remain to be investigated in the future. FHIRKA can be improved further in order to become more efficient. Establishing a connection between the gramian based optimality conditions and the interpolation conditions would further illuminate our understanding of the problem, and possibly facilitate the extension to the bilinear and quadratic bilinear cases.

We also examined current model reduction approaches for nonlinear systems such as QB-IRKA, BT for QB systems, and POD. Since QB-IRKA and BT for QB systems cannot be used to reduce systems that cannot be converted into a quadratic bilinear form, and POD

is highly dependent on the selected trajectory, we propose a different approach. Operator splitting enables us to combine method like IRKA and POD to construct an algorithm like IPS. Our error analysis of IPS showed that as long as we keep the model reduction errors under control, IPS can yield highly accurate approximate solutions and our numerical experiments provided further justification for this conclusion. For the RC Ladder model, IPS even outperformed POD; however, the main contribution of IPS consists in the mitigation of the POD input dependence. Nonetheless, we must exercise caution when employing IPS, since it is not appropriate for every nonlinear system, e.g., the tubular reactor system.

In the future, establishing a relationship between the time step h and the reduced order r could enable us to make a priori choices for these parameters. A more rigorous investigation of the effects of lifting the state variable every step of the method could provide insights on how to further improve the IPS algorithm. Constructing an algorithm that separates quadratic bilinear terms from the strictly nonlinear term would reduce the input dependence that results from POD even further. Another interesting research direction to pursue would be the integration of FHIRKA with POD via operator splitting.

Bibliography

- [1] Mian Ilyas Ahmad, Ulrike Baur, and Peter Benner. Implicit Volterra series interpolation for model reduction of bilinear systems. *Journal of Computational and Applied Mathematics*, 316, 10 2016.
- [2] Mian Ilyas Ahmad, Peter Benner, and Lihong Feng. Interpolatory model reduction for quadratic-bilinear systems using error estimators. *Engineering Computations*, 36, 01 2019.
- [3] Mian Ilyas Ahmad, Peter Benner, and I.M. Jaimoukha. Krylov subspace projection methods for model reduction of quadratic-bilinear systems. *IET Control Theory & Applications*, 10, 06 2016.
- [4] Giorgos Akrivis, Michel Crouzeix, and Charalambos Makridakis. Implicit-explicit multistep finite element methods for nonlinear parabolic problems. *Mathematics of Computation*, 67(222):457–477, 1998.
- [5] S.A. Al-Baiyat and Maamar Bettayeb. New model reduction scheme for k-power bilinear systems. In *Proceedings of the IEEE Conference on Decision and Control*, volume 1, pages 22 – 27 vol.1, 01 1994.
- [6] Awad Al-Mohy and Nicholas Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM Journal on Scientific Computing*, 33, 01 2011.
- [7] B. Aniç, C. Beattie, S. Gugercin, and A. C. Antoulas. Interpolatory weighted- $\mathcal{H}_2$ model reduction. *Automatica*, 49:1275–1280, 2013.
- [8] A. C. Antoulas. *Approximation of large-scale dynamical systems (Advances in Design*

- and Control*). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.
- [9] A. C. Antoulas, C. A. Beattie, and S. Gugercin. Interpolatory model reduction of large-scale dynamical systems. *Efficient Modeling and Control of Large-Scale Systems*, pages 3–58, 2010.
- [10] A.C. Antoulas, C. Beattie, and S. Gugercin. Interpolatory model reduction of large-scale dynamical systems. In J. Mohammadpour and K. Grigoriadis, editors, *Efficient Modeling and Control of Large-Scale Systems*. Springer-Verlag, 2010.
- [11] A.C. Antoulas, C. A. Beattie, and S. Gugercin. *Interpolatory Methods for Model Reduction*. SIAM, Philadelphia, 2020.
- [12] A.C. Antoulas, Ion Gosea, and A.C. Ionita. Model reduction of bilinear systems in the Loewner framework. *SIAM Journal on Scientific Computing*, 38:B889–B916, 01 2016.
- [13] Zhaojun Bai, P. Feldmann, and R. W. Freund. How to make theoretically passive reduced-order models passive in practice. In *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference (Cat. No.98CH36143)*, pages 207–210, 1998.
- [14] Zhaojun Bai and Roland Freund. A partial Padé-via-Lanczos method for reduced-order modeling. *Linear Algebra and its Applications*, 332-334:139–164, 08 2001.
- [15] Francesco Ballarin and Gianluigi Rozza. POD-Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems: POD-Galerkin monolithic ROM for parametrized FSI problems. *International Journal for Numerical Methods in Fluids*, 82, 12 2016.
- [16] L. Baratchart, M. Cardelli, and M. Olivi. Identification and rational ℓ_2 approximation: A gradient algorithm. *Automat.*, 27:413–418, 1991.

- [17] Maxime Barrault, Yvon Maday, Ngoc Nguyen, and Anthony Patera. An ‘empirical interpolation’ method: Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339:667–672, 11 2004.
- [18] R.H. Bartels and GW Stewart. Algorithm 432: Solution of the matrix equation $AX+XB=C$. *Communications of the ACM*, 15(9):820–826, 1972.
- [19] Ulrike Baur, Christopher Beattie, Peter Benner, and Serkan Gugercin. Interpolatory projection methods for parameterized model reduction. *SIAM Journal on Scientific Computing*, 33(5):2489–2518, 2011.
- [20] C. Beattie and S. Gugercin. Interpolatory projection methods for structure-preserving model reduction. *Systems and Control Letters*, 58:225–232, 2009.
- [21] C. A. Beattie and S. Gugercin. Model reduction by rational interpolation. In P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, editors, *To appear in Model Reduction and Approximation: Theory and Algorithms*. Available as <http://arxiv.org/abs/1409.2140>. SIAM, Philadelphia, 2015.
- [22] C.A. Beattie and S. Gugercin. A trust region method for optimal $\mathcal{H}_2$ model reduction. *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009.
- [23] Christopher Beattie, Zlatko Drmač, and Serkan Gugercin. Quadrature-based IRKA for optimal $\mathcal{H}_2$ model reduction. *IFAC-PapersOnLine*, 48:5–6, 12 2015.
- [24] Christopher Beattie and Serkan Gugercin. Realization-independent $\mathcal{H}_2$ -approximation. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4953–4958, 12 2012.
- [25] Christopher Beattie, Serkan Gugercin, and Volker Mehrmann. Model reduction for systems with inhomogeneous initial conditions. *Systems & Control Letters*, 99:99–106, 01 2017.

- [26] Bernhard Beckermann and Lothar Reichel. Error estimates and evaluation of matrix functions via the Faber transform. *SIAM J. Numerical Analysis*, 47:3849–3883, 01 2009.
- [27] Peter Benner and Tobias Breiten. Interpolation-based $\mathcal{H}_2$ -model reduction of bilinear control systems. *SIAM Journal on Matrix Analysis and Applications*, 33:859–885, 2012.
- [28] Peter Benner and Tobias Breiten. Two-sided projection methods for nonlinear model order reduction. *SIAM Journal on Scientific Computing*, 37:B239–B260, 03 2015.
- [29] Peter Benner, Tobias Breiten, and Tobias Damm. Generalised tangential interpolation for model reduction of discrete-time MIMO bilinear systems. *International Journal of Control*, 84:1398–1407, 08 2011.
- [30] Peter Benner, Zvonimir Bujanović, Patrick Kürschner, and Jens Saak. RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations. *Numerische Mathematik*, 138, 07 2017.
- [31] Peter Benner and Tobias Damm. Lyapunov equations, energy functionals, and model order reduction. *SIAM J. Control and Optimization*, 49, 01 2011.
- [32] Peter Benner and Pawan Goyal. Multipoint interpolation of Volterra series and $\mathcal{H}_2$ -model reduction for a family of bilinear descriptor systems. *Systems & Control Letters*, 97, 11 2016.
- [33] Peter Benner and Pawan Goyal. Balanced truncation model order reduction for quadratic-bilinear control systems. *Preprint*, 04 2017.
- [34] Peter Benner, Pawan Goyal, and Serkan Gugercin. $\mathcal{H}_2$ -quasi-optimal model order reduction for quadratic-bilinear control systems. *SIAM Journal on Matrix Analysis and Applications*, 39, 10 2016.

- [35] Peter Benner, Pawan Goyal, Boris Kramer, Benjamin Peherstorfer, and Karen Wilcox. Operator inference for non-intrusive model reduction of system with non-polynomial nonlinear terms. *Preprint*, 2020.
- [36] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.
- [37] Peter Benner, Patrick Kürschner, and Jens Saak. An improved numerical method for balanced truncation for symmetric second-order systems. *Mathematical and Computer Modelling of Dynamical Systems*, 0:1–23, 12 2013.
- [38] Peter Benner and Jens Saak. Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: A state of the art survey. *GAMM-Mitteilungen*, 36, 08 2013.
- [39] Mouhacine Benosman, Jeff Borggaard, and Boris Krämer. Robust POD model stabilization for the 3D Boussinesq equations based on Lyapunov theory and extremum seeking. In *2017 American Control Conference (ACC)*, pages 1827–1832, 05 2017.
- [40] G Berkooz, PJ Holmes, and John Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 11 2003.
- [41] B.N. Bond and L. Daniel. A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(12):2116–2129, 2007.
- [42] Jeff Borggaard, Traian Iliescu, and Zhu Wang. Artificial viscosity proper orthogonal decomposition. *Mathematical and Computer Modelling*, 53:269–279, 01 2011.
- [43] Tobias Breiten, Christopher Beattie, and Serkan Gugercin. Near-optimal frequency-weighted interpolatory model reduction. *Systems & Control Letters*, 78, 08 2013.

- [44] Angelika Bruns and Peter Benner. Parametric model order reduction of thermal models using the bilinear interpolatory rational Krylov algorithm. *Mathematical and Computer Modelling of Dynamical Systems*, 21:1–27, 11 2014.
- [45] A. E. Bryson and A. Carrier. Second-order algorithm for optimal model order reduction. *J. Guidance Control Dynam.*, 13:887–892, 1990.
- [46] T. Bui-Thanh, M. Damodaran, and K. Wilcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42:1505–1516, 2004.
- [47] A. Bunse-Gerstner, D. Kubalińska, G. Vossen, and D. Wilczek. $\mathcal{H}_2$ -optimal model reduction for large scale discrete dynamical MIMO systems. *Journal of Computational and Applied Mathematics*, 233(5):1202–1216, 2010.
- [48] Marco Caliari, Peter Kandolf, Alexander Ostermann, and Stefan Rainer. Comparison of software for computing the action of the matrix exponential. *BIT Numerical Mathematics*, 54, 03 2014.
- [49] A. Castagnotto and A. Lohmann. A new framework for $\mathcal{H}_2$ -optimal model reduction. *Mathematical and Computer Modeling of Dynamical Systems*, pages 1–22, 2018.
- [50] N. Chafee and E. F. Infante. A bifurcation problem for a nonlinear partial differential equation of parabolic type. *Applicable Analysis*, 4(1):17–37, 1974.
- [51] Y. Chahlaoui and P. Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. Technical report, SLICOT Working Note 2002-2, 2002.
- [52] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.

- [53] Saifon Chaturantabut and Danny Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Scientific Computing*, 32:2737–2764, 01 2010.
- [54] Saifon Chaturantabut and Danny Sorensen. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM J. Numerical Analysis*, 50:46–63, 01 2012.
- [55] E. De Sturler, S. Gugercin, M. Kilmer, S. Chaturantabut, C.A. Beattie, and M. O’Connell. Nonlinear parametric inversion using interpolatory model reduction. *SIAM Journal on Scientific Computing*, 37(3):B495–B517, 2015.
- [56] Z. Drmac, S. Gugercin, and Christopher Beattie. Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation. *SIAM Journal on Scientific Computing*, 37:A625–A652, 01 2015.
- [57] Zlatko Drmac and Serkan Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38, 05 2015.
- [58] Zlatko Drmac, Serkan Gugercin, and Christopher Beattie. Vector fitting for matrix-valued rational approximation. *SIAM Journal on Scientific Computing*, 37, 03 2015.
- [59] V. Druskin, V. Simoncini, and M. Zaslavsky. Solution of the time-domain inverse resistivity problem in the model reduction framework part I. One-dimensional problem with SISO data. *SIAM Journal on Scientific Computing*, 35(3):A1621–A1640, 2013.
- [60] Vladimir Druskin, Leonid Knizhnerman, and Mikhail Zaslavsky. Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts. *SIAM Journal on Scientific Computing*, 31:3760–3780, 01 2009.
- [61] Vladimir Druskin, Chad Lieberman, and Mikhail Zaslavsky. On adaptive choice of shifts in rational Krylov subspace reduction of evolutionary problems. *SIAM Journal on Scientific Computing*, 32:2485–2496, 01 2010.

- [62] I. P. Duff, P. Goyal, and P. Benner. Balanced truncation for a special class of bilinear descriptor systems. *IEEE Control Systems Letters*, 3(3):535–540, 2019.
- [63] Paolo D’Alessandro, Alberto Isidori, and Antonio Ruberti. Realization and structure theory of bilinear dynamical systems. *Siam Journal on Control*, 12, 08 1974.
- [64] Mark Embree. Unstable modes in projection-based reduced-order models: How many can there be, and what do they tell you? *Systems and Control Letters*, 124:49–59, 12 2019.
- [65] Dale Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Proceedings of the IEEE Conference on Decision and Control*, pages 127 – 132, 01 1985.
- [66] Peter Feldmann and Roland Freund. Efficient linear analysis by Padè approximation via the Lanczos process. *IEEE Trans. Computer-Aided Design*, 14, 04 1994.
- [67] G. Flagg, C.A. Beattie, and S. Gugercin. Convergence of the Iterative Rational Krylov Algorithm. *Systems and Control Letters*, 61(6):688–691, 2012.
- [68] Garret Flagg and Serkan Gugercin. Multipoint Volterra series interpolation and $\mathcal{H}_2$ optimal model reduction of bilinear systems. *SIAM Journal on Matrix Analysis and Applications*, 36:549–579, 01 2015.
- [69] Andreas Frommer and Valeria Simoncini. *Matrix Functions*, volume 13, pages 275–303. Model Order Reduction: Theory, Research Aspects and Applications, 01 2008.
- [70] P. Fulcheri and M. Olivi. Matrix rational $\mathcal{H}_2$ approximation: A gradient algorithm based on schur analysis. *SIAM J. Control Optim.*, 36:2103–2127, 1998.
- [71] D. Gaier. *Lectures on Complex Approximation*. Birkhauser, Cambridge, MA, 1987.
- [72] Katie Gallivan, E. Grimme, and Paul Van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Applied Mathematics Letters*, 7:75–80, 02 1994.

- [73] Kyle Gallivan, A. Vandendorpe, and Paul Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Analysis Applications*, 26:328–349, 01 2005.
- [74] Walter Gautschi. *Numerical Analysis*. Springer, 01 2012.
- [75] Wodek Gawronski and Jer-Nan Juang. Model reduction in limited time and frequency intervals. *International journal of systems science*, 21(2):349–376, 1990.
- [76] Amir Gholami, Andreas Mang, and George Biros. An inverse problem formulation for parameter estimation of a reaction diffusion model of low grade gliomas. *Journal of Mathematical Biology*, 05 2015.
- [77] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ -error bounds. *Internat. J. Control*, 39(6):1115–1193, 1984.
- [78] Keith Glover. A tutorial on Hankel-norm approximation. Technical report, International Institute for Applied Systems Analyses, 01 1989.
- [79] Gene Golub and Charles Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 2012.
- [80] G.H. Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10:413–432, 04 1973.
- [81] Ion Gosea, Qiang Zhang, and Athanasios Antoulas. Preserving the DAE structure in the Loewner model reduction and identification framework. *Advances in Computational Mathematics*, 46, 02 2020.
- [82] Pawan Goyal and Martin Redmann. Time-limited $\mathcal{H}_2$ -optimal model order reduction. *Applied Mathematics and Computation*, 355:184–197, 08 2019.

- [83] Eric Grimme. *Krylov projection methods for model reduction*. PhD thesis, University of Illinois at Urbana-Champaign, 05 1997.
- [84] Chenjie Gu. QImor: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 30:1307–1320, 09 2011.
- [85] S. Gugercin, A.C. Antoulas, and N Bedrossian. Approximation of the International Space Station 1R and 12A models. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 2, pages 1515–1516. IEEE, 2001.
- [86] S. Gugercin, C. Beattie, and A. C. Antoulas. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *Siam J. Matrix Anal. Appl.*, 30(2):609–638, 2008.
- [87] Serkan Gugercin and Athanasios C. Antoulas. A time-limited balanced reduction method. *Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI*, 2003.
- [88] Serkan Gugercin and Athanasios Antoulasy. A survey of model reduction by balanced truncation and some new results. *International Journal of Control - INT J CONTR*, 77, 05 2004.
- [89] Serkan Gugercin, Tatjana Stykel, and Sarah Wyatt. Model reduction of descriptor systems by interpolatory projection methods. *SIAM Journal on Scientific Computing*, 35, 01 2013.
- [90] Stefan Güttel. Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM Mitteilungen*, 36, 08 2013.
- [91] Y. Halevi. Frequency weighted model reduction via optimal projection. *IEEE Transactions on Automatic Control*, 37(10), 1992.

- [92] S. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA Journal of Numerical Analysis*, 2, 07 1982.
- [93] Alexander Hay, Jeff Borggaard, Akhtar Imran, and Dominique Pelletier. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis of the POD. *Journal of Computational Physics*, 229:1327–1352, 02 2010.
- [94] Robert Heinemann and Aubrey Poore. Multiplicity, stability, and oscillatory dynamics of the tubular reactor. *Chemical Engineering Science - CHEM ENG SCI*, 36:1411–1419, 12 1981.
- [95] Martin Hess, Annalisa Quaini, and Gianluigi Rozza. Reduced basis model order reduction for Navier–Stokes equations in domains with walls of varying curvature. *International Journal of Computational Fluid Dynamics*, pages 1–8, 07 2019.
- [96] M.W. Hess and P. Benner. A reduced basis method for microwave semiconductor devices with geometric variations. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 33(4):1071–1081, 2014.
- [97] Jan Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *The Empirical Interpolation Method*, pages 67–85. Springer International Publishing, 01 2016.
- [98] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4):1179–1193, 2005.
- [99] Nicholas J. Higham. *Functions of Matrices*. Society for Industrial and Applied Mathematics, 2008.
- [100] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Review*, 51(4):747–764, 2009.
- [101] Christian Himpe, Tobias Leibner, and Stephan Rave. Hierarchical approximate proper

- orthogonal decomposition. *SIAM Journal on Scientific Computing*, 40:A3267–A3292, 10 2018.
- [102] Michael Hinze and Stefan Volkwein. *Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems: Error Estimates and Suboptimal Control*, volume 45, pages 261–306. *Dimension Reduction of Large-scale Systems*, 01 2005.
- [103] Cosmina Hoguea, Christos Davatzikos, and George Biros. An image-driven parameter estimation problem for a reaction–diffusion glioma growth model with mass effects. *Journal of Mathematical Biology*, 56:793–825, 07 2008.
- [104] Jeffrey Hokanson. *Numerically Stable and Statistically Efficient Algorithms for Large Scale Exponential Fitting*. PhD thesis, Rice University, 2013.
- [105] Jeffrey Hokanson. Projected nonlinear least squares for exponential fitting. *SIAM Journal on Scientific Computing*, 39:A3107–A3128, 01 2017.
- [106] Jeffrey Hokanson and Caleb Magruder. $\mathcal{H}_2$ -optimal model reduction using projected nonlinear least squares. *Preprint*, 2018.
- [107] Helge Holden, Kenneth H. Karlsen, Nils Henrik Risebro, and Knut Andreas Lie. *Splitting methods for partial differential equations with Rough Solutions*. European Mathematical Society, Zurich, Switzerland, 2010.
- [108] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry (Cambridge Monographs on Mechanics)*. Cambridge University Press, Cambridge, UK, 1996.
- [109] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1990.
- [110] Y.S. Hung and K. Glover. Optimal Hankel-norm approximation of stable systems with first-order stable weighting functions. *Systems & Control Letters*, 7:165–172, 06 1986.

- [111] D.C. Hyland and D.S. Bernstein. The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton, and Moore. *IEEE Trans. Automat. Control.*, 30:1201–1211, 1985.
- [112] Akhtar Imran, Jeff Borggaard, and John Burns. Computing functional gains for designing more energy-efficient buildings using a model reduction framework. *Fluids*, 3:97, 11 2018.
- [113] I.M. Jaimoukha and Mohsen Ebrahim. Krylov Subspace Methods for Solving Large Lyapunov Equations. *Siam Journal on Numerical Analysis - SIAM J NUMER ANAL*, 31, 02 1994.
- [114] Ian Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016.
- [115] A.R. Kellems, D. Roos, N. Xiao, and S.J. Cox. Low-dimensional, morphologically accurate models of subthreshold membrane potential. *Journal of Computational Neuroscience*, 27(2):161, 2009.
- [116] Leonid Knizhnerman. Calculation of functions of unsymmetric matrices using Arnoldi’s method. *Computational Mathematics and Mathematical Physics*, 31:1–9, 01 1992.
- [117] Tamara Kolda and Brett Bader. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 08 2009.
- [118] Daniel Kressner, Stefano Massei, and Leonardo Robol. Low-rank updates and a divide-and-conquer method for linear matrix equations. *SIAM Journal on Scientific Computing*, 41, 12 2017.
- [119] Boris Krämer and Karen Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, pages 1–11, 04 2019.

- [120] K. Kunisch and S. Volkwein. Control of the Burgers equation by a reduced-order approach using Proper Orthogonal Decomposition. *Journal of Optimization Theory and Applications*, 102:345–371, 1999.
- [121] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40:539–575, 11 2002.
- [122] Peter Kunkel and Volker Mehrmann. Differential-algebraic equations. Analysis and numerical solution. *European Mathematical Society*, 01 2006.
- [123] Takio Kurita. *Principal Component Analysis (PCA)*, pages 1–4. Springer International Publishing, Cham, 2019.
- [124] J. Kutz, Steven Brunton, Bingni Brunton, and Joshua Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 11 2016.
- [125] J. Kutz, Jonathan Tu, Joshua Proctor, and Steven Brunton. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2:165–191, 12 2016.
- [126] Patrick Kürschner. Balanced truncation model order reduction in limited time intervals for large systems. *Advances in Computational Mathematics*, 06 2018.
- [127] Patrick Kürschner. Approximate residual-minimizing shift parameters for the low-rank ADI iteration. *Electronic Transactions on Numerical Analysis ETNA*, 51:240–261, 09 2019.
- [128] Alan Lattimer, Jeff Borggaard, Serkan Gugercin, Kray Luxbacher, and Brian Lattimer. Computationally efficient wildland fire spread models. In *Interflam 2016 - 14th International Fire Science & Engineering Conference, At Royal Holloway College, University of London, UK*, 07 2016.

- [129] J. Lawson. Generalized Runge-Kutta processes for stable systems with large Lipschitz constants. *Siam Journal on Numerical Analysis - SIAM J NUMER ANAL*, 4:372–380, 09 1967.
- [130] A. Lepschy, G.A. Mian, G. Pinato, and U. Viaro. Rational l_2 approximation: A nongradient algorithm. *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 2321–2323, 1991.
- [131] Stefano Lorenzi, Antonio Cammi, Lelio Luzzi, and Gianluigi Rozza. POD-Galerkin method for finite volume approximation of Navier-Stokes and RANS equations. *Computer Methods in Applied Mechanics and Engineering*, 311, 08 2016.
- [132] Shev MacNamara and Gilbert Strang. *Operator Splitting*, pages 95–114. Springer, 01 2016.
- [133] Yvon Maday and Olga Mula. A generalized empirical interpolation method: Application of reduced basis techniques to data assimilation. *Springer INdAM Series*, 4, 01 2013.
- [134] C. Magruder, C. Beattie, and S. Gugercin. Rational Krylov methods for optimal $\mathcal{L}_2$ model reduction. *49th IEEE Conference on Decision and Control, Atlanta, GA*, 2010.
- [135] AJ Mayo and AC Antoulas. A framework for the solution of the generalized realization problem. *Linear algebra and its applications*, 425(2):634–662, 2007.
- [136] L. Meier and D.G. Luenberger. Approximation of linear constant systems. *IEE. Trans. Automat. Contr.*, 12:585–588, 1967.
- [137] S. A. Melchior, P. Van Dooren, and K. A. Gallivan. Model reduction of linear time-varying systems over finite horizons. *Applied Numerical Mathematics*, 77:72–81, 2014.
- [138] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential

- of a matrix, twenty-five years later. *Society for Industrial and Applied Mathematics*, 45:3–49, 03 2003.
- [139] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.
- [140] Clifford T Mullis, Richard Roberts, et al. Synthesis of minimum roundoff noise fixed point digital filters. *Circuits and Systems, IEEE Transactions on*, 23(9):551–562, 1976.
- [141] Yuji Nakatsukasa, Olivier Sète, and Lloyd Trefethen. The AAA algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40, 12 2016.
- [142] Ngoc Nguyen, Gianluigi Rozza, D. Huynh, and A. Patera. *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Parabolic PDEs: Application to Real-Time Bayesian Parameter Estimation*, pages 151 – 177. John Wiley and Sons, ltd, 10 2010.
- [143] Ngoc Nguyen, Gianluigi Rozza, and Anthony Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers’ equation. *Calcolo*, 46:157–185, 09 2009.
- [144] Meghan O’Connell, Misha Kilmer, Eric de Sturler, and Serkan Gugercin. Computing reduced order models via inner-outer Krylov recycling in diffuse optical tomography. *SIAM Journal on Scientific Computing*, 39, 01 2016.
- [145] H.K.F. Panzer, S. Jaensch, T. Wolf, and B. Lohmann. A greedy rational Krylov method for $\mathcal{H}_2$ -pseudooptimal model order reduction with preservation of stability. In *American Control Conference (ACC), 2013*, pages 5512–5517, 2013.
- [146] Benjamin Peherstorfer, Daniel Butnaru, Karen Willcox, and Hans-Joachim Bungartz.

- Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36, 01 2014.
- [147] T. Penzl. LYAPACK - A MATLAB Toolbox for Large Lyapunov and Riccati Equations, Model Reduction Problems, and Linear–Quadratic Optimal Control Problems. netlib, 1999. Version 1.0.
- [148] Lars Pernebo and Leonard Silverman. Model reduction via state space representation. *Automatic Control, IEEE Transactions on*, AC-27:382 – 387, 05 1982.
- [149] Lawrence Pillage and Ronald Rohrer. Asymptotic waveform evaluation for timing analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 9:352 – 366, 05 1990.
- [150] Elizabeth Qian, Martin Grepl, Karen Veroy, and Karen Willcox. A certified trust region reduced basis approach to PDE-constrained optimization. *SIAM Journal on Scientific Computing*, 39, 02 2017.
- [151] Elizabeth Qian, Boris Krämer, Benjamin Peherstorfer, and Karen Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 02 2020.
- [152] V Raghavan, R.A. Rohrer, L.T. Pillage, J.Y. Lee, Eric Bracken, and M.M. Alaybeyi. Awe-inspired. In *Proceedings of IEEE Custom Integrated Circuits Conference - CICC '93*, pages 18.1.1 – 18.1.8, 06 1993.
- [153] Martin Redmann. An l_t^2 -error bound for time-limited balanced truncation. *Systems & Control Letters*, 136:104620, 02 2020.
- [154] Martin Redmann and Patrick Kürschner. An output error bound for time-limited balanced truncation. *Systems and Control Letters*, 121, 11 2018.

- [155] Timo Reis. *Mathematical Modeling and Analysis of Nonlinear Time-Invariant RLC Circuits*, pages 125–198. Springer International Publishing, Cham, 2014.
- [156] Julius Reiss, Philipp Schulze, Jörn Sesterhenn, and Volker Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40, 12 2015.
- [157] Clarence Rowley, Tim Colonius, and Richard Murray. Model reduction for compressible flow using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189:115–129, 01 2003.
- [158] W. J. Rugh. *Nonlinear System Theory*. The Johns Hopkins University Press, Baltimore, MD, USA, 1981.
- [159] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis - SIAM J NUMER ANAL*, 29, 02 1992.
- [160] Omer San and Jeff Borggaard. Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows. *International Journal for Numerical Methods in Fluids*, 78, 05 2015.
- [161] Michelle Schatzman. Numerical integration of reaction-diffusion systems. *Numerical Algorithms*, 31:247–269, 2002.
- [162] Michelle Schatzman. Toward non commutative numerical analysis: high order integration in time. *Journal of Scientific Computing*, 17(1-4):99–116, 2002.
- [163] Peter Schmid and Jörn Sesterhenn. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656, 11 2008.
- [164] Valeria Simoncini. Analysis of the rational Krylov subspace projection method for

- large-scale algebraic Riccati equations. *SIAM Journal on Matrix Analysis and Applications*, 37, 02 2016.
- [165] Klajdi Sinani. Iterative Rational Krylov Algorithm for unstable dynamical systems and genaralized coprime factorizations. Master’s thesis, Virginia Tech, 2015.
- [166] Klajdi Sinani and Serkan Gugercin. $\mathcal{H}_2(t_f)$ optimality conditions for a finite-time horizon. *Automatica*, 110:108604, 12 2019.
- [167] D. Sorensen and Mark Embree. A DEIM induced CUR factorization. *SIAM Journal on Scientific Computing*, 38:A1454–A1482, 05 2016.
- [168] J.T. Spanos, M.H. Milman, and D.L. Mingori. A new algorithm for $\mathcal{L}_2$ optimal model reduction. *Automat.*, 28:897–909, 1992.
- [169] Raymond Speth, William Green, Shev Macnamara, and Gilbert Strang. Balanced splitting and rebalanced splitting. *SIAM Journal on Numerical Analysis*, 51, 01 2013.
- [170] Gilbert Strang. Essays in linear algebra. *Wellesley-Cambridge Press*, 2012.
- [171] W.G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numerical Analysis*, 5, 01 1968.
- [172] Tatjana Stykel. Gramian-based model reduction for descriptor systems. *MCSS*, 16:297–319, 03 2004.
- [173] Tatjana Stykel and Valeria Simoncini. Krylov subspace methods for projected Lyapunov equations. *Applied Numerical Mathematics*, 62, 01 2012.
- [174] Renee Swischuk, Boris Krämer, Cheng Huang, and Karen Willcox. Learning physics-based reduced-order models for a single-injector combustion process. *AIAA Journal*, pages 1–15, 03 2020.

- [175] Renee Swischuk, Laura Mainini, Benjamin Peherstorfer, and Karen Willcox. Projection-based model reduction: Formulations for physics-based machine learning. *Computers & Fluids*, 08 2018.
- [176] Endre Süli and David F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.
- [177] The MORwiki Community. Nonlinear RC Ladder. MORwiki – Model Order Reduction Wiki, 2018.
- [178] P. van Dooren, K.A. Gallivan, and P.A. Absil. $\mathcal{H}_2$ -optimal model reduction of MIMO systems. *Applied Mathematics Letters*, 21(12):1267–1273, 2008.
- [179] Christian Villemagne and Robert Skelton. Model reduction using a projection formulation. In *International Journal of Control*, volume 46, pages 461–466, 12 1987.
- [180] P. Vuillemin, C. Poussot-Vassal, and D. Alazard. Poles residues descent algorithm for optimal frequency-limited $\mathcal{H}_2$ model approximation. In *Control Conference (ECC), 2014 European*, pages 1080–1085. IEEE, 2014.
- [181] D. A. Wilson. Optimum solution of model-reduction problem. *Proceedings of the Institution of Electrical Engineers*, 117(6), 1970.
- [182] Xuping Xie, Muhammad Mohebujjaman, L. Rebholz, and Traian Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing*, 40, 09 2017.
- [183] Boyuan Yan and Peng Li. Reduced order modeling of passive and quasi-active dendrites for nervous system simulation. *Journal of computational neuroscience*, 31:247–71, 10 2011.
- [184] W. Y. Yan and J. Lam. An approximate approach to $\mathcal{H}_2$ optimal model reduction. *IEEE Trans. Automat. Control*, 44:1341–1358, 1999.

- [185] W. Yao and Simão Marques. Nonlinear aerodynamic and aeroelastic model reduction using a discrete empirical interpolation method. *AIAA Journal*, 55:1–14, 01 2017.
- [186] Ajmal Yousuff and Robert Skelton. Covariance equivalent realizations with application to model reduction of large-scale systems. *Control and Dynamic Systems*, 22:273–348, 12 1985.
- [187] Ajmal Yousuff, D.A Wagie, and Robert Skelton. Linear system approximation via covariance equivalent realizations. *Journal of Mathematical Analysis and Applications*, 106:91–115, 02 1985.
- [188] Kemin Zhou. Frequency weighted $\mathcal{L}_\infty$ norm optimal Hankel norm model reduction. *Automatic Control, IEEE Transactions on*, 40:1687 – 1699, 11 1995.
- [189] Kemin Zhou, Gregory Salomon, and Eva Wu. Balanced realization and model reduction for unstable systems. *International Journal of Robust and Nonlinear Control*, 9:183 – 198, 03 1999.