Efficient $\mathcal{H}_2$-Based Parametric Model Reduction via Greedy Search

Jon Carl Cooper

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

Master of Science
in
Mathematics

Matthias Chung, Chair
Serkan Gugercin
Mark Embree

December 11, 2020
Blacksburg, Virginia

Keywords: Parametric Model Reduction, Greedy Selection,
Rational Interpolation, $\mathcal{H}_2$ Norm

Efficient $\mathcal{H}_2$-Based Parametric Model Reduction via Greedy Search

Jon Carl Cooper

(ABSTRACT)

Dynamical systems are mathematical models of physical phenomena widely used throughout the world today. When a dynamical system is too large to effectively use, we turn to model reduction to obtain a smaller dynamical system that preserves the behavior of the original. In many cases these models depend on one or more parameters other than time, which leads to the field of parametric model reduction.

Constructing a parametric reduced-order model (ROM) is not an easy task, and for very large parametric systems it can be difficult to know how well a ROM models the original system, since this usually involves many computations with the full-order system, which is precisely what we want to avoid. Building off of efficient $\mathcal{H}_\infty$ approximations, we develop a greedy algorithm for efficiently modeling large-scale parametric dynamical systems in an $\mathcal{H}_2$-sense.

We demonstrate the effectiveness of this greedy search on a fluid problem, a mechanics problem, and a thermal problem. We also investigate Bayesian optimization for solving the optimization subproblem, and end with extending this algorithm to work with MIMO systems.

Efficient $\mathcal{H}_2$-Based Parametric Model Reduction via Greedy Search

Jon Carl Cooper

(GENERAL AUDIENCE ABSTRACT)

In the past century, mathematical modeling and simulation has become the third pillar of scientific discovery and understanding, alongside theory and experimentation. Mathematical models are used every day, and are essential to modern engineering problems. Some of these mathematical models depend on quantities other than just time, parameters such as the viscosity of a fluid or the strength of a spring. These models can sometimes become so large and complicated that it can take a very long time to run simulations with the models. In such a case, we use parametric model reduction to come up with a much smaller and faster model that behaves like the original model. But when these large models vary highly with the parameters, it can also become very expensive to reduce these models accurately.

Algorithms already exist for quickly computing reduced-order models (ROMs) with respect to one measure of how "good" the ROM is. In this thesis we develop an algorithm for quickly computing the ROM with respect to a different measure - one that is more closely tied to how the models are simulated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

1.1 Motivation

Dynamical systems offer us the mathematical tools to accurately model and simulate how the world around us changes over time, particularly through models derived by physics. Only now, in the age of high computational power and big data, has the field of model reduction really come into its own, as the need for more complex and faster models has grown substantially. Like most other pre-processing methods, the offline cost of developing a reduced-order model (a quick-to-simulate model that retains the complexity of the original full-order model) is fairly expensive. This is especially true when trying to compute reduced-order models for large parametric systems. In this thesis we develop a greedy algorithm for quickly constructing reduced-order models for parametric dynamical systems that are accurate in the so-called $\mathcal{H}_2$-sense.

1.2 Outline

In Chapter 2 we introduce the fundamental theories behind the work presented later in the thesis. We start with introducing dynamical systems, then move to discussions on the general form of a dynamical system and system norms. We end Section 2.1 with an introduction to parametric systems. We then introduce the field of model reduction, specifically working

towards the iterative rational Krylov algorithm [38], and expanding to algorithms for reducing parametric systems. We finish off this chapter with recent efficient estimates of the $\mathcal{H}_\infty$ error in parametric model reduction.

In Chapter 3 we recall the efficient $\mathcal{H}_\infty$ estimates and apply a quadrature rule to acquire a similarly efficient approximation to the $\mathcal{H}_2$ norm. We then investigate the accuracy of this approximation first on non-parametric systems, before applying this approximation to parametric model reduction. We test our resulting algorithm on a convection-diffusion flow model, a beam model, and a thermal model. Next, we investigate the benefit of using Bayesian optimization to select the next parameter sample. We conclude this chapter with a formulation for applying this efficient parametric model reduction scheme to multiple-input multiple-output (MIMO) systems.

Our contributions in this thesis are:

- Providing approximate lower and upper bounds on the $\mathcal{H}_2$ error between a full-order model and a reduced-order model that do not depend on evaluating the full-order transfer function;
- Developing a greedy algorithm making use of these efficient approximations for $\mathcal{H}_2$-based parametric model reduction;
- Investigating Bayesian optimization as an efficient means of parameter selection within the greedy algorithm;
- Expanding on the efficient error approximations and the greedy algorithm to work with MIMO systems.

Chapter 2

Background

In this chapter we provide a basic review of the material to be covered later in the thesis stated from the ground up, though this is not meant to be a comprehensive resource on any of these topics. Where applicable, additional resources are listed, allowing for a more in-depth investigation into these topics.

2.1 Dynamical Systems

A dynamical system is a mathematical model of a time-dependent process, and usually takes the form of a system of differential equations together with some output. Consider how a warm plate cools down over time when placed in a cold environment. We can represent the plate by a number of nodes, perhaps on a uniform grid, and come up with how the temperature at one node interacts with the temperatures of its neighbors. This forms the internal dynamics of the system. For the output, we might consider only evaluating the temperature at a single node, or possibly looking at the average temperature across the plate.

The mathematical representation of these systems can be formed by finite difference methods or finite element methods on a spatially-discretized domain [2, 7, 10, 12, 21, 28], or even by data directly [3, 35, 43]. Dynamical systems that accurately model the real-world phenomena they represent are integral to many fields of science and engineering. Therefore, the study of

dynamical systems and the ability to construct faithful reduced-order models (ROMs) from these systems deserves much attention.

In the following subsections, we will review the fundamentals of dynamical systems theory starting with first-order linear, time-independent (LTI) systems. We then move to a more general formulation for dynamical systems and discuss system norms.

2.1.1 Linear Systems

Consider the following linear system of differential equations:

$$\begin{aligned} \mathbf{E}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \\ y(t) &= \mathbf{c}^\top \mathbf{x}(t) + \mathbf{d}u(t), \end{aligned} \tag{2.1}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^n$, and $\mathbf{d} \in \mathbb{R}$ are constant quantities. Such a system is called a *linear time-invariant* (LTI) dynamical system. In this representation, $u : \mathbb{R} \mapsto \mathbb{R}$ represents the input to the system, or forcing term, $\mathbf{x} : \mathbb{R} \mapsto \mathbb{R}^n$ represents the internal state of the system which evolves over time, and $y : \mathbb{R} \mapsto \mathbb{R}$ represents the output of the system. This system is also called a *single-input single-output* (SISO) system, since u and y have only one output dimension each. We call the **order** of the system n, and note that because n is finite, we are working with partially discretized (spatially discretized) systems. In this thesis, we will not consider the case when $\mathbf{E}$ is singular (in which case we have a *system of differential algebraic equations*). So, without loss of generality, the $\mathbf{E}$ term can be taken to be the $n \times n$ identity matrix $\mathbf{I}$. We also make the assumption that the *feedthrough* term, $\mathbf{d}$, is zero since the majority of model reduction techniques leave the feedthrough untouched.

Assuming this system starts with some initial state $\mathbf{x}_0$ at time $t_0 = 0$, we can use the method

of variation of parameters to calculate the system output at any time $t > 0$:

$$y(t) = \mathbf{c}^\top \mathrm{e}^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t \mathbf{c}^\top \mathrm{e}^{\mathbf{A}(t-\tau)}\mathbf{b}u(\tau)\mathrm{d}\tau.$$

Letting $h(t) = \mathbf{c}^\top \mathrm{e}^{\mathbf{A}t}\mathbf{b}$, assuming $\mathbf{x}_0 = \mathbf{0}$, and taking the frequency domain representation of this solution (by either the Fourier transform or the Laplace transform), we arrive at

$$Y(s) = H(s)U(s), \quad H(s) = \mathbf{c}^\top(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}, \tag{2.2}$$

where $U(s)$ and $Y(s)$ are the frequency domain representations of $u(t)$ and $y(t)$, respectively. We call $H(s)$ the **transfer function** of the system, since it directly relates system inputs to system outputs in the frequency domain. Therefore, if we want to model this system, it suffices to model the transfer function. To derive some useful representations of $H(s)$, we first assume $\mathbf{A}$ is diagonalizable and let $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ be an eigenvalue decomposition of $\mathbf{A}$, where $\mathbf{\Lambda}$ is a diagonal matrix with elements λ_j, $j = 1, 2, ..., n$. Then, H can be equivalently written as $H(s) = \tilde{\mathbf{c}}^\top(s\mathbf{I} - \mathbf{\Lambda})^{-1}\tilde{\mathbf{b}}$, where $\tilde{\mathbf{c}} = \mathbf{V}^\top\mathbf{c}$ and $\tilde{\mathbf{b}} = \mathbf{V}^{-1}\mathbf{b}$. Since $(s\mathbf{I} - \mathbf{\Lambda})$ is a diagonal matrix, it becomes clear that H is the rational function

$$H(s) = \sum_{j=1}^{n} \frac{\phi_j}{s - \lambda_j}, \tag{2.3}$$

with $\phi_j = \tilde{\mathbf{c}}_j\tilde{\mathbf{b}}_j \in \mathbb{C}, \lambda_j \in \mathbb{C}$. This is the **pole-residue** representation of the transfer function (in complex analysis, the partial fraction expansion of H) since each λ_j is directly a pole of the transfer function with its associated residue ϕ_j. Note that the λ_js are the eigenvalues of $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ (or the generalized eigenvalues of $\mathbf{A}\mathbf{x} = \lambda\mathbf{E}\mathbf{x}$ in the case where $\mathbf{E} \neq \mathbf{I}$). If $\mathbf{A}$ does not have an eigenvalue decomposition, one can use the Jordan decomposition of $\mathbf{A}$ to see that any repeated eigenvalue of $\mathbf{A}$ with multiplicity k simply shows up as a pole of the

transfer function with order k.

One can generalize this SISO system into a *multi-input multi-output (MIMO)* system (and by extension single-input multi-output or multi-input single-output) by considering a system similar to (2.1) where $\mathbf{b}$ is replaced with $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{c}$ is replaced with $\mathbf{C} \in \mathbb{R}^{n\times \ell}$, and $\mathbf{d}$ is replaced with $\mathbf{D} \in \mathbb{R}^{m\times \ell}$ for $m > 1, \ell > 1$. Here, m is the input dimension ($u : \mathbb{R} \mapsto \mathbb{R}^{1\times m}$), and ℓ is the output dimension ($y : \mathbb{R} \mapsto \mathbb{R}^{\ell\times 1}$). In this case, the transfer function (2.3) quite clearly becomes a function $H : \mathbb{C} \mapsto \mathbb{C}^{\ell\times m}$. Then, the pole-residue representation for such a MIMO system is given by $\sum_{j=1}^{n} \frac{\hat{\mathbf{C}}_j \hat{\mathbf{B}}_j^\top}{s-\lambda_j}$, where $\hat{\mathbf{C}}_j$ is the jth column of $\mathbf{C}^\top\mathbf{V}^{-1}$ and $\hat{\mathbf{B}}_j$ is the jth row of $\mathbf{V}^{-1}\mathbf{B}$. See [27, 59] for additional theory on dynamical systems and transfer functions.

2.1.2 General Formulation

Now consider a second-order LTI system

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) &= \mathbf{b}u(t) \\ y(t) &= \mathbf{c}^\top\mathbf{x}(t), \end{aligned} \tag{2.4}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$, $\mathbf{D} \in \mathbb{R}^{n\times n}$, $\mathbf{K} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, and $\mathbf{c} \in \mathbb{R}^n$ are constant quantities. Such a system is used to describe a variety of physical processes that involve oscillatory motion, the most iconic setting being that of a mass-spring system. In such a setting, $\mathbf{M}$ represents the mass terms, $\mathbf{K}$ the spring constants, and $\mathbf{D}$ the damping applied to the system. We can follow the same steps as before to calculate the transfer function of this system as $H(s) = \mathbf{c}^\top(s^2\mathbf{M} + s\mathbf{D} + \mathbf{K})^{-1}\mathbf{b}$.

Comparing this transfer function to our first order LTI transfer function (2.3) tells us that

transfer functions can be written in the form

$$H(s) = \mathbf{c}^\top \mathbf{Q}^{-1}(s)\mathbf{b}, \tag{2.5}$$

where $\mathbf{Q} : \mathbb{C} \mapsto \mathbb{C}^{n\times n}$. This $\mathbf{Q}$ can be derived from the original system of differential equations (as in (2.1) or (2.4)) by applying a frequency domain transform to all of the terms governing the internal dynamics and collecting them together. For a first-order system such as (2.1), for example, $\mathbf{Q}(s) = s\mathbf{E} - \mathbf{A}$, and for a second-order system such as (2.4), $\mathbf{Q}(s) = s^2\mathbf{M} + s\mathbf{D} + \mathbf{K}$. This formulation boils down any dynamical system into three parts: the input-to-state mapping $\mathbf{b}$, the state-to-state transition $\mathbf{Q}$, and the state-to-output mapping $\mathbf{c}$. Although for most systems $\mathbf{b}$ and $\mathbf{c}$ are constant, in general they can also depend on the frequency s [14].

Note that the transfer function for (2.4) can also be represented in the form of (2.3), but with a state-space dimension of $2n$. Similarly, in the general form $\mathbf{Q}^{-1}(s)$ can always either be fully represented by a rational decomposition, or can be closely approximated by one (such as by the Padé approximation). We call a dynamical system *asymptotically stable* if the real part of every pole of the system is negative. For an asymptotically stable system with no input, the system state will exponentially decay to a zero state. If at least one pole of the system has a positive real part or if there is a defective eigenvalue on the imaginary axis, the system is then called *unstable*, meaning that in at least one direction, the system state can grow exponentially. If there are some semi-simple eigenvalues on the imaginary axis (none with positive real parts), the system is called *marginally stable*, and some directions of the system state can oscillate indefinitely. In this thesis, when we refer to a stable system, we are usually referring to an asymptotically sable system.

2.1.3 System Norms

Let $\mathbf{y}$ be the output of a stable MIMO dynamical system (such as (2.1), but with $\mathbf{B} \in \mathbb{R}^{n\times m}$ and $\mathbf{C} \in \mathbb{R}^{n\times \ell}$). Let $\mathbf{u}$ be the input to the system, and assume L_2 integrability for both functions. Let $\mathbf{H} : \mathbb{C} \mapsto \mathbb{C}^{\ell\times m}$ be the corresponding transfer function, $\mathbf{U} : \mathbb{C} \mapsto \mathbb{C}^m$ be the frequency domain input, and $\mathbf{Y} : \mathbb{C} \mapsto \mathbb{C}^\ell$ be the frequency domain output. For $\mathrm{i}^2 = -1$, we define the $\mathcal{H}_\infty$ norm to be

$$\|\mathbf{H}\|_{\mathcal{H}_\infty} = \sup_{\omega\in\mathbb{R}} \|\mathbf{H}(\mathrm{i}\omega)\|_2, \tag{2.6}$$

and the $\mathcal{H}_2$ norm to be

$$\|\mathbf{H}\|_{\mathcal{H}_2} = \left(\frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\|_{\mathrm{F}}^2 \,\mathrm{d}\omega\right)^{1/2}, \tag{2.7}$$

where $\|\cdot\|_{\mathrm{F}}$ is the matrix Frobenius norm. It should be immediately obvious that if any pole of the system has a zero real part, the $\mathcal{H}_\infty$ norm will be infinite. What might not be so obvious is that the $\mathcal{H}_\infty$ norm is related to the L_2 norm in the time domain:

$$\begin{aligned}
\|y\|_{\mathrm{L}_2}^2 &= \int_0^\infty \|y(t)\|^2 \,\mathrm{d}t \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathbf{Y}(\mathrm{i}\omega)\|_2^2 \,\mathrm{d}\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\mathbf{U}(\mathrm{i}\omega)\|_2^2 \,\mathrm{d}\omega \\
&\le \frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\|_2^2 \,\|\mathbf{U}(\mathrm{i}\omega)\|_2^2 \,\mathrm{d}\omega \\
&\le \sup_{\omega\in\mathbb{R}} \|\mathbf{H}(\mathrm{i}\omega)\|_2^2 \,\frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathbf{U}(\mathrm{i}\omega)\|_2^2 \,\mathrm{d}\omega \\
&= \|\mathbf{H}\|_{\mathcal{H}_\infty}^2 \|u\|_{\mathrm{L}_2}^2.
\end{aligned}$$

A similar result holds for the $\mathcal{H}_2$ and L_∞ norms:

$$\begin{aligned}
\|y\|_{\mathrm{L}_\infty}^2 &= \max_{t\geq 0} \|y(t)\|_\infty \\
&= \max_{t\geq 0} \left\| \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathbf{Y}(\mathrm{i}\omega) \mathrm{e}^{\mathrm{i}\omega t} \,\mathrm{d}\omega \right\|_\infty \\
&\leq \max_{t\geq 0} \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{Y}(\mathrm{i}\omega) \mathrm{e}^{\mathrm{i}\omega t}\|_\infty \,\mathrm{d}\omega \\
&\leq \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{Y}(\mathrm{i}\omega)\|_\infty \mathrm{d}\omega \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\mathbf{U}(\mathrm{i}\omega)\|_\infty \,\mathrm{d}\omega \\
&\leq \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\mathbf{U}(\mathrm{i}\omega)\|_2 \,\mathrm{d}\omega \\
&\leq \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\|_2 \, \|\mathbf{U}(\mathrm{i}\omega)\|_2 \,\mathrm{d}\omega \\
&\leq \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\|_\mathrm{F} \, \|\mathbf{U}(\mathrm{i}\omega)\|_2 \,\mathrm{d}\omega \\
&\leq \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{H}(\mathrm{i}\omega)\|_2 \mathrm{d}\omega \right)^{1/2} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{U}(\mathrm{i}\omega)\|_2 \,\mathrm{d}\omega \right)^{1/2} \\
&= \|\mathbf{H}\|_{\mathcal{H}_2} \|u\|_{\mathrm{L}_2}.
\end{aligned}$$

This indicates a strong relationship between the L_2 norm of the input and the norms of the output and transfer function. In particular, this allows us to derive the following relationships: for two transfer functions $\mathbf{H}_1$ and $\mathbf{H}_2$ on the same time domain input $\mathbf{u}$ with associated time domain outputs $\mathbf{y}_1$ and $\mathbf{y}_2$, we have

$$\begin{aligned}
\|\mathbf{y}_1 - \mathbf{y}_2\|_{\mathrm{L}_2} &\leq \|\mathbf{H}_1 - \mathbf{H}_2\|_{\mathcal{H}_\infty} \|\mathbf{u}\|_{\mathrm{L}_2}, \\
\|\mathbf{y}_1 - \mathbf{y}_2\|_{\mathrm{L}_\infty} &\leq \|\mathbf{H}_1 - \mathbf{H}_2\|_{\mathcal{H}_2} \|\mathbf{u}\|_{\mathrm{L}_2}.
\end{aligned}$$

This relationship is particularly useful when $\mathbf{H}_1$ and $\mathbf{H}_2$ are related, namely when $\mathbf{H}_2$ is a low-order approximation to $\mathbf{H}_1$. Note that this relies on the fact that $\mathbf{H}_1 - \mathbf{H}_2$ is a

transfer function with output $\mathbf{y}_1 - \mathbf{y}_2$ (see (A.1) for a proof). Note also because the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms are only defined for stable systems, by convention we say that both norms are infinite for unstable systems. We can still compute the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms "naïvely" from their definitions (2.6) (2.7) without regard to the stability of the system. In this case, we call these the L_∞ and L_2 norms of the system, respectively. More discussion on these relationships can be found in [1, 64].

2.2 Model Reduction

The goal of model reduction is to take a large dynamical system and construct a much smaller dynamical system, the reduced-order model (ROM), that closely approximates the original full-order model. In this thesis we will only discuss projection methods, which compress the full-order model down into a much smaller space, but there are other popular methods for model reduction. In particular, methods such as the Loewner framework [3, 43, 51], vector fitting [40], and AAA [55] are essential for when we can only evaluate the transfer function at predetermined locations. For further discussion on general model reduction, additional resources can be found in [2, 12, 21].

2.2.1 Projection Methods

Suppose we have the first-order LTI SISO system

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

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^n$, and $\mathbf{d} \in \mathbb{R}$ are constant quantities. To reduce this system, we seek some integer $r \ll n$ and some matrices $\mathbf{A}_r \in \mathbb{R}^{r\times r}$, $\mathbf{b}_r \in \mathbb{R}^r$, and $\mathbf{c}_r \in \mathbb{R}^r$ such that the system formed by these *reduced* matrices (the reduced-order model or ROM) well-approximates the original system. We make the assumption that the ROM is of the form

$$\begin{aligned} \dot{\mathbf{z}}(t) &= \mathbf{A}_r\mathbf{z}(t) + \mathbf{b}_r u(t) \\ \hat{\mathbf{y}}(t) &= \mathbf{c}_r^\top \mathbf{z}(t) + \mathbf{d}_r u(t), \end{aligned} \tag{2.8}$$

where $\mathbf{x}(t)$ is approximated by $\mathbf{V}\mathbf{z}(t)$, $\mathbf{V} \in \mathbb{R}^{n\times r}$, a linear projection from $\mathbb{R}^r$ to $\mathbb{R}^n$, and $\mathbf{z} \in \mathbb{R}^r$. The transfer function for this ROM is then given by

$$H_r(s) = \mathbf{c}_r^\top \left(s\mathbf{I}_r - \mathbf{A}_r\right)^{-1} \mathbf{b}_r + \mathbf{d}_r.$$

If we enforce the additional constraint on the ROM that there exists a matrix $\mathbf{W} \in \mathbb{R}^{n\times r}$ such that $\mathbf{W}^\top(\mathbf{V}\dot{\mathbf{z}}(t) - \mathbf{A}\mathbf{V}\mathbf{z} - \mathbf{b}u(t)) = \mathbf{0}$ for all $t > 0$, we arrive at the *Petrov-Galerkin* projection-based model reduction scheme. Taking $\mathbf{W} = \mathbf{V}$ results in the so-called *Galerkin* projection. In this case, we assume that $\mathbf{W}^\top\mathbf{V} = \mathbf{I}_r$, so $\mathbf{V}$ and $\mathbf{W}$ have bi-orthonormal columns such that $\mathbf{W}^\top\mathbf{V} = \mathbf{I}$. Multiplying through, we arrive at a simple relation between the full-order matrices and their reduced-order counterparts:

$$\begin{aligned} \mathbf{A}_r &= \mathbf{W}^\top\mathbf{A}\mathbf{V}, \quad \mathbf{b}_r = \mathbf{W}^\top\mathbf{b}, \\ \mathbf{c}_r &= \mathbf{V}^\top\mathbf{c}, \qquad \mathbf{d}_r = \mathbf{d}. \end{aligned} \tag{2.9}$$

There are a wide variety of different approaches to computing $\mathbf{V}$ and $\mathbf{W}$, and many methods include extensions to MIMO systems, nonlinear systems, or parametric systems. Balanced truncation, for example, is a popular projection method that computes a ROM that is

accurate in an $\mathcal{H}_\infty$-sense by removing subspaces that are jointly difficult for the system to reach from a zero initial condition and are difficult to observe via the output [19, 49, 53, 54, 63]. Another popular projection method is proper orthogonal decomposition (POD), which simply constructs the projection matrices $\mathbf{V}$ and $\mathbf{W}$ based on the subspaces that the internal system state frequently explores for a given input. General POD resources can be found in [5, 24, 45, 50], with extensions in [6, 36, 44, 46, 47, 48]. Other methods for constructing $\mathbf{V}$ and $\mathbf{W}$ usually involve ensuring the ROM interpolates the full-order transfer function at a set number of points.

2.2.2 Interpolatory Model Reduction and IRKA

Interpolatory model reduction has been one of the more popular model reduction frameworks and will form the basis of this thesis. In this section we will review a number of facts about interpolatory model reduction and the iterative rational Krylov algorithm (IRKA) more specifically. The reader is referred to [4] for more detailed discussions.

Suppose we have the first-order SISO LTI system (2.1) with $\mathbf{d} = \mathbf{0}$ and $\mathbf{E} = \mathbf{I}_n$, and we want to construct $\mathbf{V}$ and $\mathbf{W}$ such that the resulting reduced system is of order r and interpolates the original system at two (not necessarily distinct) sets of complex points $\{\sigma_i\}_{i=1}^r$ and $\{\mu_j\}_{j=1}^r$, i.e.,

$$H(\sigma_i) = H_r(\sigma_i) \text{ and } H(\mu_j) = H_r(\mu_j) \quad \text{for } i, j = 1, 2, ..., r.$$

Theorem 2.1. *Assume* $H(s) = \mathbf{c}^\top \mathbf{Q}(s)^{-1}\mathbf{b}$ *with* $\mathbf{Q}(\sigma_i)$ *invertible for* $i = 1, 2, ..., \mathrm{r}$. *Choose* $\mathbf{V}$ *such that the span of* $\mathbf{V}$ *contains* $\{\mathbf{Q}^{-1}(\sigma_i)\mathbf{b}\}_{i=1}^r$. *Then the transfer function of the reduced system* $H_r(s) = \mathbf{c}_r^\top \mathbf{Q}_r(s)^{-1}\mathbf{b}_r$ *created with* $\mathbf{V}$ *will interpolate* $H(s) = \mathbf{c}^\top \mathbf{Q}^{-1}(s)\mathbf{b}$ *at every* σ_i, $i = 1, 2, ..., \mathrm{r}$.

Proof. Define $\mathcal{P}(\zeta) = \mathbf{V}\mathbf{Q}_r^{-1}(\zeta)\mathbf{W}^\top\mathbf{Q}(\zeta)$, where $\mathbf{Q}(s) = s\mathbf{E} - \mathbf{A}$ and $\mathbf{Q}_r(s) = s\mathbf{E}_r - \mathbf{A}_r$, and note that the range of $\mathcal{P}(\zeta)$ is the range of $\mathbf{V}$ by construction, and that

$$\mathcal{P}^2(\zeta) = \mathbf{V}\mathbf{Q}_r^{-1}(\zeta)\mathbf{W}^\top\mathbf{Q}(\zeta)\mathbf{V}\mathbf{Q}_r^{-1}(\zeta)\mathbf{W}^\top\mathbf{Q}(\zeta) = \mathbf{V}\mathbf{Q}_r^{-1}(\zeta)\mathbf{W}^\top\mathbf{Q}(\zeta) = \mathcal{P}(\zeta),$$

and therefore $\mathcal{P}(\zeta)$ is a projector. Then,

$$\begin{aligned} H(\zeta) - H_r(\zeta) &= \mathbf{c}^\top\mathbf{Q}^{-1}(\zeta)\mathbf{b} - \mathbf{c}_r^\top\mathbf{Q}_r^{-1}(\zeta)\mathbf{b}_r \\ &= \mathbf{c}^\top\mathbf{Q}^{-1}(\zeta)\mathbf{b} - (\mathbf{V}^\top\mathbf{c})^\top\mathbf{Q}_r^{-1}(\zeta)\mathbf{W}^\top\mathbf{b} \\ &= \mathbf{c}^\top(\mathbf{Q}^{-1}(\zeta) - \mathbf{V}\mathbf{Q}_r^{-1}(\zeta)\mathbf{W}^\top)\mathbf{b} \\ &= \mathbf{c}^\top(\mathbf{Q}^{-1}(\zeta) - \mathbf{V}\mathbf{Q}_r^{-1}(\zeta)\mathbf{W}^\top)\mathbf{Q}(\zeta)\mathbf{Q}^{-1}(\zeta)\mathbf{b} \\ &= \mathbf{c}^\top(\mathbf{I} - \mathcal{P}(\zeta))\mathbf{Q}^{-1}(\zeta)\mathbf{b}. \end{aligned}$$

Therefore, if $\mathbf{Q}^{-1}(\zeta)\mathbf{b}$ is in the range of $\mathcal{P}(\zeta)$, which is equivalent to being in the range of $\mathbf{V}$, then $\mathbf{c}^\top(\mathbf{I} - \mathcal{P}(\zeta))\mathbf{Q}^{-1}(\zeta)\mathbf{b} = \mathbf{c}^\top(\mathbf{Q}^{-1}(\zeta)\mathbf{b} - \mathbf{Q}^{-1}(\zeta)\mathbf{b}) = 0$, and thus H_r interpolates H at ζ. So, if we choose $\mathbf{V}$ to be a basis of the set of vectors $\{\mathbf{Q}^{-1}(\sigma_i)\mathbf{b}\}_{i=1}^r$, then H_r will interpolate H at every σ_i, $i = 1, 2, ..., r$. □

The proof of $\mathbf{W}$ yielding interpolation at μ_j, $j = 1, 2, ..., r$ follows similarly, except we swap the roles of $\mathbf{b}$ and $\mathbf{c}$, use $\mathbf{Q}^\top$ in place of $\mathbf{Q}$, and swap $\mathbf{V}$ and $\mathbf{W}$ in forming $\mathcal{P}(\zeta)$. It should be noted that if $\sigma_k = \mu_k$ for $k = 1, 2, ..., r$, then it also holds that $H'(\sigma_k) = H_r'(\sigma)$ for $k = 1, 2, ..., r$, even without having to compute or evaluate H' [38].

Although the free derivative interpolation can increase the accuracy of a reduced model without increasing its order, note that preserving the stability of the full-order system is only guaranteed when using Galerkin projection (setting $\mathbf{W} = \mathbf{V}$) for systems with a symmetric positive definite $\mathbf{E}$ term and an $\mathbf{A}$ term that satisfies $\mathbf{A} + \mathbf{A}^\top$ being negative definite [56, 61].

Since we can choose where to interpolate, a natural question is whether there are optimal interpolation points. Meier and Luenberger [52] first showed that $\mathcal{H}_2$-optimality can be achieved when the interpolation points are chosen to be the negative poles of the transfer function, i.e., for a full-order transfer function H and a reduced order transfer function H_r with poles λ_j, $j = 1, 2, ..., r$ that satisfies

$$H_r(s) = \underset{\substack{\text{order } \widetilde{H} \leq r \\ \widetilde{H} \text{ stable}}}{\arg\min} \|H - \widetilde{H}\|_{\mathcal{H}_2},$$

then $H(-\lambda_j) = H_r(-\lambda_j)$ and $H'(-\lambda_j) = H_r'(-\lambda_j)$ for $j = 1, 2, ..., r$. Although these optimal locations cannot be known a priori, we are able to form an iterative algorithm that usually converges to them.

The iterative rational Krylov algorithm (IRKA) [38] starts with randomly initialized interpolation locations (closed under complex conjugation), constructs an interpolating reduced system, then iteratively sets the interpolation locations to be the negative eigenvalues of the $\mathbf{A}_r$ matrix, until convergence. These steps are outlined in Algorithm 1 below. Note that in this implementation we leave $\mathbf{E}$ arbitrary. IRKA uses points closed under complex conjugation to keep $\mathbf{V}$ and $\mathbf{W}$ real-valued, which ensures that the reduced system yields a real-valued output.

According to [38], for most systems the choice of the initial interpolation points does not effect the convergence of the algorithm. However, for larger and more difficult systems, the authors suggest randomly choosing the initial locations to be within the reflected spectrum of $\mathbf{A}$, since this region likely contains the reflected spectrum of $\mathbf{A}_r$ for the optimal reduced model. We will refer to using IRKA in the Galerkin setting as *single-* or *one-sided IRKA*, and IRKA in the Petrov-Galerkin setting as *double-* or *two-sided IRKA*. IRKA has been extended to many different types of systems, including MIMO systems [37, 39], bilinear

systems [16, 20], quadratic systems [23], general nonlinear systems [17, 62], delay systems [29], and parametric systems [13].

Algorithm 1 Iterative Rational Krylov Algorithm Outline

1. Select $\{\sigma_k\}_{k=1}^{r}$ to be closed under conjugation and "near" the optimal locations
2. Compute $\mathbf{V}$ and $\mathbf{W}$ such that
   $\text{Range}(\mathbf{V}) = \text{span}\{(\sigma_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{b}, ..., (\sigma_r\mathbf{E} - \mathbf{A})^{-1}\mathbf{b}\}$
   $\text{Range}(\mathbf{W}) = \text{span}\{(\sigma_1\mathbf{E} - \mathbf{A})^{-\top}\mathbf{c}, ..., (\sigma_r\mathbf{E} - \mathbf{A})^{-\top}\mathbf{c}\}$

while $\|\sigma_{\text{current}} - \sigma_{\text{previous}}\| >$ tolerance **do**

3. Compute $\mathbf{E}_r \leftarrow \mathbf{W}^\top\mathbf{E}\mathbf{V}$ and $\mathbf{A}_r \leftarrow \mathbf{W}^\top\mathbf{A}\mathbf{V}$
4. Set $\sigma_{\text{current}} \leftarrow -\lambda(\mathbf{A}_r, \mathbf{E}_r)$
5. Recompute $\mathbf{V}$ and $\mathbf{W}$ according to step 2 with the new locations σ_{current}

end while

6. Compute $\mathbf{E}_r \leftarrow \mathbf{W}^\top\mathbf{E}\mathbf{V}$, $\mathbf{A}_r \leftarrow \mathbf{W}^\top\mathbf{A}\mathbf{V}$, $\mathbf{b}_r \leftarrow \mathbf{W}^\top\mathbf{b}$, $\mathbf{c}_r^\top \leftarrow \mathbf{c}^\top\mathbf{V}$

2.2.3 Reducing Parametric Systems

Consider our original first-order LTI system (2.1) of order n, taking $\mathbf{d} = 0$, but with dependencies on additional parameters $\mathbf{p} = [\mathrm{p}_1, \mathrm{p}_2, ..., \mathrm{p}_p] \in \mathbb{R}^p$

$$\begin{aligned} \dot{\mathbf{x}}(t, \mathbf{p}) &= \mathbf{A}(\mathbf{p})\mathbf{x}(t, \mathbf{p}) + \mathbf{b}(\mathbf{p})u(t) \\ y(t, \mathbf{p}) &= \mathbf{c}(\mathbf{p})^\top\mathbf{x}(t, \mathbf{p}), \end{aligned} \tag{2.10}$$

where $\mathbf{x}(\cdot, \mathbf{p})$ denotes the parameter $\mathbf{p} \in \mathbb{R}^p$ dependent state vector. This system then has the transfer function

$$H(s, \mathbf{p}) = \mathbf{c}^\top(\mathbf{p})\left(s\mathbf{I}_n - \mathbf{A}(\mathbf{p})\right)^{-1}\mathbf{b}(\mathbf{p}).$$

Such a parametric dynamical system is often used to describe families of related systems, particularly in cases where the system is derived from a partial differential equation that contains parameters other than time. One such example would be a mass-spring system, where the damping coefficient is left as a variable until the system is simulated. For simplicity,

we assume the parameter dependence is one-dimensional, although the discussion here can be easily extended to a multi-parameter setting. We desire a reduced-order model (ROM) that is also dependent on $\mathbf{p}$ and has the same structure as the original system. This ROM should be of the form

$$\begin{aligned} \dot{\mathbf{z}}(t,\mathbf{p}) &= \mathbf{A}_r(\mathbf{p})\mathbf{z}(t,\mathbf{p}) + \mathbf{b}_r(\mathbf{p})u(t) \\ \hat{y}(t,\mathbf{p}) &= \mathbf{c}_r(\mathbf{p})^\top \mathbf{z}(t,\mathbf{p}), \end{aligned} \tag{2.11}$$

with its transfer function being

$$H_r(s,\mathbf{p}) = \mathbf{c}_r^\top(\mathbf{p})\left(s\mathbf{I}_r - \mathbf{A}_r(\mathbf{p})\right)^{-1}\mathbf{b}_r(\mathbf{p}).$$

We refer the reader to [8, 9, 11, 15, 21, 25, 26, 30, 35, 41] for detailed discussions about parametric model reduction in general, and to [4, 22, 34, 41, 42, 57] for a more comprehensive look at projection-based parametric model reduction.

For fixed $\mathbf{p}$, there are numerous methods for constructing the $\mathbf{V}$ and $\mathbf{W}$ projection matrices, as discussed in Sections 2.2.1 and 2.2.2. Although many non-parametric methods extend to the parametric setting, there are few theoretical guarantees. There are of course methods specifically designed for this setting, like MMM (2.3.1) or the p-AAA algorithm [26]. One can note that double-sided IRKA, despite not being developed for the parametric setting, is able to capture the first derivative of the transfer function with respect to the parameter for the same reason that it captures the derivative of the transfer function with respect to the frequency s. So, for frequency samples $\{\sigma_i\}_{i=1}^r$,

$$\begin{aligned}
&\text{if } \{\mathbf{Q}^{-1}(\sigma_1,\mu_1)\mathbf{b}, \mathbf{Q}^{-1}(\sigma_2,\mu_2)\mathbf{b}, ..., \mathbf{Q}^{-1}(\sigma_r,\mu_r)\mathbf{b}\} \subseteq \text{range}(\mathbf{V}) \\
&\text{and } \{\mathbf{Q}^{-\top}(\sigma_1,\mu_1)\mathbf{c}, \mathbf{Q}^{-\top}(\sigma_2,\mu_2)\mathbf{c}, ..., \mathbf{Q}^{-\top}(\sigma_r,\mu_r)\mathbf{c}\} \subseteq \text{range}(\mathbf{W}), \\
&\text{then } H(\sigma_i,\mu_i) = H_r(\sigma_i,\mu_i), \frac{\partial}{\partial s}H(\sigma_i,\mu_i) = \frac{\partial}{\partial s}H_r(\sigma_i,\mu_i) \\
&\text{and } \frac{\partial}{\partial p}H(\sigma_i,\mu_i) = \frac{\partial}{\partial p}H_r(\sigma_i,\mu_i) \text{ for } i = 1,2,...,r.
\end{aligned}$$

These results also extend to parameters with more than one dimension [10].

Despite this property, note also that guaranteeing stability for one particular value of $\mathbf{p}$ does not guarantee stability for the entire parameter range, and may even cause instability in some ranges of $\mathbf{p}$.

A common approach to extend the use of any model reduction technique for non-parametric systems to parametric ones is by assuming they can create accurate *local* ROMs for fixed p values. Once we are able to construct a $\mathbf{V}$ for a fixed p, which we call a *local basis matrix*, we can then construct the *global basis matrix* $\mathbf{V}_\text{g}$ by concatenating multiple local basis matrices together. Unfortunately, the global basis matrix loses most theoretical properties that local basis matrices may have, which is especially true of a method like balanced truncation. As long as $\mathbf{V}_\text{g}$ is effective in reducing the error between the full-order model and the local ROMs, we can iteratively build up the global basis matrix until the error is below a desired tolerance. In order to ensure a low error over the entire parameter domain, we either use a predetermined of sampling locations (generally requiring that constructing the local and global ROMs is cheap), or we use a greedy algorithm to select the next parameter location to sample at by finding the maximum error between the full-order model and the global ROM over the whole parameter domain. The second option requires access to (cheap) error measurements, but is preferable for large or difficult-to-reduce systems. The process

of iteratively selecting the next sampling location is outlined in Algorithm 2. This sort of algorithm, where in each iteration we take an optimal step towards the solution, is referred to as a *greedy search*.

Algorithm 2 Extending Local MOR Methods to Parametric Systems

1. Select $\mathbf{p}_0$ and set $\mathbf{V}_\mathrm{g} = [\,]$, maxerror $= 1$

while error $>$ tolerance **do**

2. Construct a local ROM at $\mathbf{p}_k$
3. Update $\mathbf{V}_\mathrm{g} = [\mathbf{V}_\mathrm{g}, \mathbf{V}_\mathrm{local}]$
4. Construct the global ROM using $\mathbf{V}_\mathrm{g}$, obtaining H_{r_k}
5. Solve $p_{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}} \|\mathrm{H} - \mathrm{H}_{\mathrm{r_k}}\|$

end while

Between the two most common system error measurements for step 5 above, the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norms (2.6), (2.7), the $\mathcal{H}_\infty$ norm is often preferred in this case. Computing the exact $\mathcal{H}_\infty$ norm of any system is not an easy task. Similarly, the typical method for computing the $\mathcal{H}_2$ norm involves solving a a Lyapunov problem, which quickly becomes computationally intractable for large systems. One alternative, if a minimal $\mathcal{H}_2$ error is desired, is to use quadrature to approximate (2.7). Although this might be somewhat more feasible than a Lyapunov solve for large systems (at the cost of reduced accuracy), this requires as many $\mathbf{Q}^{-1}(s, \mathbf{p})$ solves as quadrature nodes, and again becomes difficult for very large systems. This is especially true when using an adaptive quadrature method, since the quadrature nodes cannot be known beforehand and thus the full-order system evaluations cannot be precomputed. Our main goal in this thesis is to make Algorithm 2 efficient at reducing systems in the $\mathcal{H}_2$-sense by introducing computationally cheap approximations to the $\mathcal{H}_2$ error in step 5.

2.3 $\mathcal{H}_\infty$ Error Estimates

We begin to develop a sense for approximating the $\mathcal{H}_2$ error between two systems by first recalling the approximations to the $\mathcal{H}_\infty$ error developed by Feng and Benner [31, 32]. Assume we have a system of the form

$$\mathbf{x}(s,\mathbf{p}) = \mathbf{Q}^{-1}(s,\mathbf{p})\mathbf{b}(\mathbf{p}),$$
$$y(s,\mathbf{p}) = \mathbf{c}^\top(\mathbf{p})\mathbf{x}(s,\mathbf{p}),$$

where $\mathbf{Q}(s,\mathbf{p})$ can be derived by simplifying all state-to-state transition matrices in the frequency domain. We will refer to this as the *primal system*. We then similarly define the *dual system* to be

$$\mathbf{x}_{\text{du}}(s,\mathbf{p}) = \mathbf{Q}^{-\top}(s,\mathbf{p})\mathbf{c}(\mathbf{p}),$$
$$y(s,\mathbf{p}) = \mathbf{b}^\top(\mathbf{p})\mathbf{x}_{\text{du}}(s,\mathbf{p}).$$

Suppose we construct two ROMs, one to model the primal system via $\mathbf{V}_{\text{pr}}$, and one to model the dual system via $\mathbf{V}_{\text{du}}$. In this section we will only assume Galerkin projection, however the analysis can be easily extended to Petrov-Galerkin projection. Denote the solutions of the primal and dual ROMs to be $\hat{\mathbf{x}}_{\text{pr}}(s,\mathbf{p}) = \hat{\mathbf{Q}}^{-1}(s,\mathbf{p})\hat{\mathbf{b}}(\mathbf{p})$ and $\hat{\mathbf{x}}_{\text{du}}(s,\mathbf{p}) = \hat{\mathbf{Q}}^{-\top}(s,\mathbf{p})\hat{\mathbf{c}}(\mathbf{p})$, respectively. (In this section we will refer to reduced-order matrices and vectors by using hat notation, to avoid confusion with the various subscripts.) We then define the *primal residual* and *dual residual* by

$$\mathbf{r}_{\text{pr}}(s,\mathbf{p}) = \mathbf{b} - \mathbf{Q}(s,\mathbf{p})\hat{\mathbf{x}}_{\text{pr}}(s,\mathbf{p}),$$
$$\mathbf{r}_{\text{du}}(s,\mathbf{p}) = \mathbf{c} - \mathbf{Q}^\top(s,\mathbf{p})\hat{\mathbf{x}}_{\text{du}}(s,\mathbf{p}),$$

respectively. We can then use these residuals to define the *primal residual system* and *dual residual system* as follows:

$$\mathbf{x}_{\mathrm{r_{pr}}}(s,\mathbf{p}) = \mathbf{Q}^{-1}(s,\mathbf{p})\mathbf{r}_{\mathrm{pr}}(s,\mathbf{p}),$$
$$\mathbf{x}_{\mathrm{r_{du}}}(s,\mathbf{p}) = \mathbf{Q}^{-\top}(s,\mathbf{p})\mathbf{r}_{\mathrm{du}}(s,\mathbf{p}).$$

With this formulation, we have the ability to reduce these residual systems in a manner similar to reducing the primal and dual systems. This only requires using $\mathbf{r}_{\mathrm{pr}}(s,\mathbf{p})$ in place of $\mathbf{b}(\mathbf{p})$ in the primal system and $\mathbf{r}_{\mathrm{du}}(s,\mathbf{p})$ in place of $\mathbf{c}(\mathbf{p})$ for the dual system. These reduced residual systems then yield the reduced residual solutions $\hat{\mathbf{x}}_{\mathrm{r_{pr}}}(s,\mathbf{p})$ and $\hat{\mathbf{x}}_{\mathrm{r_{du}}}(s,\mathbf{p})$. Denote the $\mathbf{V}$ matrices used to reduce the residual systems by $\mathbf{V}_{\mathrm{r_{pr}}}$ and $\mathbf{V}_{\mathrm{r_{du}}}$. Note that the range of $\mathbf{V}_{\mathrm{r_{pr}}}$ **must** span at least the range of $\mathbf{V}_{\mathrm{pr}}$, and similar for $\mathbf{V}_{\mathrm{r_{du}}}$, otherwise the reduced residual systems will not be approximating the true residual systems. In particular, if $\mathbf{V}_{\mathrm{r_{pr}}} = \mathbf{V}_{\mathrm{pr}}$, then the primal residual system is trivially $\mathbf{0}$. The solution proposed by Feng and Benner [32] is to set $\mathbf{V}_{\mathrm{r_{pr}}} = \mathrm{orth}\{\mathbf{V}_{\mathrm{pr}}, \mathbf{V}_*\}$, and similar for $\mathbf{V}_{\mathrm{r_{du}}}$, where $\mathbf{V}_*$ is constructed using a separate set of interpolation points from the ones used to construct $\mathbf{V}_{\mathrm{pr}}$. Alternatively, a different projection method altogether could be used to construct $\mathbf{V}_*$ so long as $\mathrm{range}\,(\mathbf{V}_*) \not\subseteq \mathrm{range}\,(\mathbf{V}_{\mathrm{pr}})$.

With these four reduced systems together with the *residual of the primal residual system*

$$\mathbf{r}_{\mathrm{r_{pr}}}(s,\mathbf{p}) = \mathbf{r}_{\mathrm{pr}}(s,\mathbf{p}) - \mathbf{Q}(s,\mathbf{p})\hat{\mathbf{x}}_{\mathrm{r_{pr}}}(s,\mathbf{p}),$$

Feng and Benner [32] derive three pairs of approximate upper and lower bounds for the *point wise error* between the full-order and reduced-order transfer functions $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$ as in Table 2.1. The dependence on s and $\mathbf{p}$ is removed here for the simplicity of the notation.

	Estimate 1	Estimate 2	Estimate 3
Lower Bound	$\Delta_1(s,\mathbf{p}) := \lvert\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}\rvert$	$\Delta_1(s,\mathbf{p})$	$\Delta_1^{\mathrm{pr}}(s,\mathbf{p}) := \lvert\mathbf{c}^\top \hat{\mathbf{x}}_{\mathrm{r_{pr}}}\rvert$
Range	$\Delta_2(s,\mathbf{p}) := \lvert\hat{\mathbf{x}}_{\mathrm{r_{du}}}^\top \mathbf{r}_{\mathrm{pr}}\rvert$	$\Delta_2^{\mathrm{pr}}(s,\mathbf{p}) := \lvert\hat{\mathbf{x}}_{\mathrm{r_{pr}}}^\top \mathbf{r}_{\mathrm{du}}\rvert$	$\Delta_3(s,\mathbf{p}) := \lvert\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{r_{pr}}}\rvert$
Upper Bound	$\Delta_1 + \Delta_2$	$\Delta_1 + \Delta_2^{\mathrm{pr}}$	$\Delta_1^{\mathrm{pr}} + \Delta_3$

Table 2.1: Upper and lower error bound approximations to $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$ [32].

The proof that $\Delta_1(s,\mathbf{p})$ is an approximate lower bound is given below. The proofs for the remainder of the upper and lower bounds follow similarly and can be found in [31, 32].

Theorem 2.2. $\Delta_1(s,\mathbf{p}) := |\hat{\mathbf{x}}_{\mathrm{du}}^\top(s,\mathbf{p})\mathbf{r}_{\mathrm{pr}}(s,\mathbf{p})|$ *is an approximate lower bound for* $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$ *for all* s *and* $\mathbf{p}$*, where* $\hat{\mathbf{x}}_{\mathrm{du}}(s,\mathbf{p})$ *is the solution to the reduced dual system at* s *and* $\mathbf{p}$ *and* $\mathbf{r}_{\mathrm{pr}}(s,\mathbf{p})$ *is the solution to the primal-residual system at* s *and* $\mathbf{p}$*.*

Proof. For simplicity of notation, we omit the dependence of $\hat{\mathbf{x}}_{\mathrm{pr}}$ on $(s,\mathbf{p})$, and similar for $\mathbf{x}_{\mathrm{du}}, \hat{\mathbf{x}}_{\mathrm{du}}$, and $\mathbf{r}_{\mathrm{pr}}$. Denote $\hat{\mathbf{Q}}(s,\mathbf{p})$ as the reduced form of $\mathbf{Q}(s,\mathbf{p})$, obtained via the projection $\mathbf{V}$. Then

$$
\begin{aligned}
\Delta_1(s,\mathbf{p}) &= |\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| \\
&= |\mathbf{x}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| - |\mathbf{x}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| + |\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| \\
&= |\mathbf{c}\mathbf{Q}^{-1}(s,\mathbf{p})\mathbf{r}_{\mathrm{pr}}| - |\mathbf{x}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| + |\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| \\
&= |\mathbf{c}\mathbf{Q}^{-1}(s,\mathbf{p})(\mathbf{b} - \mathbf{Q}(s,\mathbf{p})\hat{\mathbf{x}}_{\mathrm{pr}})| - |\mathbf{x}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| + |\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| \\
&= |\mathbf{c}\mathbf{Q}^{-1}(s,\mathbf{p})(\mathbf{b} - \mathbf{Q}(s,\mathbf{p})\mathbf{V}\hat{\mathbf{Q}}^{-1}(s,\mathbf{p})\hat{\mathbf{b}})| - |\mathbf{x}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| + |\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| \\
&= |\mathbf{c}\mathbf{Q}^{-1}(s,\mathbf{p})\mathbf{b} - \hat{\mathbf{c}}\hat{\mathbf{Q}}^{-1}(s,\mathbf{p})\hat{\mathbf{b}}| - |\mathbf{x}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| + |\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| \\
&= |H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})| - |\mathbf{x}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}| + |\hat{\mathbf{x}}_{\mathrm{du}}^\top \mathbf{r}_{\mathrm{pr}}|.
\end{aligned}
$$

Then, by the triangle inequality,

$$\Delta_1(s,\mathbf{p}) \leq |H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})| + |(\mathbf{x}_{\mathrm{du}} - \hat{\mathbf{x}}_{\mathrm{du}})^\top \mathbf{r}_{\mathrm{pr}}|.$$

Therefore $\Delta_1(s,\mathbf{p})$ is an approximate lower bound to $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$, with the error in the approximation being $|(\mathbf{x}_{\mathrm{du}}(s,\mathbf{p}) - \hat{\mathbf{x}}_{\mathrm{du}}(s,\mathbf{p}))^\top \mathbf{r}_{\mathrm{pr}}(s,\mathbf{p})|$. □

It is clear that the error term above is small when either the reduced primal system or reduced dual systems are accurate. With our own analysis, we will now show that the error in this approximation is generally much smaller than the error in the primal state vector $\|\mathbf{x}_{\mathrm{pr}}(s,\mathbf{p}) - \hat{\mathbf{x}}_{\mathrm{pr}}(s,\mathbf{p})\|_2$, and therefore makes $\Delta_1(s,\mathbf{p})$ a useful approximation to $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$.

Theorem 2.3. *Let $g_r(s,\mathbf{p}) \geq \|\mathbf{x}_{\mathrm{du}} - \hat{\mathbf{x}}_{\mathrm{du}}\|_2$ be an upper bound for the primal state-vector error associated with the model reduction scheme used to create $\hat{\mathbf{x}}$. Then, as r increases, the term $\|\mathbf{x}_{\mathrm{pr}}(s,\mathbf{p}) - \hat{\mathbf{x}}_{\mathrm{pr}}(s,\mathbf{p})\|_2$ decreases proportional to $g_r^2(s,\mathbf{p})$.*

Proof. Assume the full order is n and the order of the reduced primal system is r. Suppose the reduced system was generated by a model reduction scheme that allows us to vary r. Furthermore, suppose that the error in the primal state vector $\|\mathbf{x}_{\mathrm{pr}}(s,\mathbf{p}) - \hat{\mathbf{x}}_{\mathrm{pr}}(s,\mathbf{p})\|_2$ decreases as r approaches n. We expect the same properties to hold for the reduced dual system when using the same model reduction scheme as for the primal system. We can then write

$$\max_{s,\mathbf{p}} \{\|\mathbf{x}_{\mathrm{pr}}(s,\mathbf{p}) - \hat{\mathbf{x}}_{\mathrm{pr}}(s,\mathbf{p})\|_2, \|\mathbf{x}_{\mathrm{du}}(s,\mathbf{p}) - \hat{\mathbf{x}}_{\mathrm{du}}(s,\mathbf{p})\|_2\} \leq g_r(s,\mathbf{p}).$$

Noting that

$$\begin{aligned}\mathbf{r}_{\text{pr}}(s,\mathbf{p}) &= \mathbf{b} - \mathbf{Q}(s,\mathbf{p})\hat{\mathbf{x}}_{\text{pr}}(s,\mathbf{p}) \\ &= \mathbf{b} - \mathbf{Q}(s,\mathbf{p})\hat{\mathbf{x}}_{\text{pr}}(s,\mathbf{p}) - (\mathbf{b} - \mathbf{Q}(s,\mathbf{p})\mathbf{x}_{\text{pr}}(s,\mathbf{p})) \\ &= \mathbf{Q}(s,\mathbf{p})\left(\mathbf{x}_{\text{pr}}(s,\mathbf{p}) - \hat{\mathbf{x}}_{\text{pr}}(s,\mathbf{p})\right),\end{aligned}$$

we can then show

$$\begin{aligned}|(\mathbf{x}_{\text{du}} - \hat{\mathbf{x}}_{\text{du}})^\top \mathbf{r}_{\text{pr}}| &\leq \|\mathbf{x}_{\text{du}} - \hat{\mathbf{x}}_{\text{du}}\|_2 \, \|\mathbf{r}_{\text{pr}}\|_2 \\ &\leq \|\mathbf{x}_{\text{du}} - \hat{\mathbf{x}}_{\text{du}}\|_2 \, \|\mathbf{Q}(s,\mathbf{p})\|_2 \, \|\mathbf{x}_{\text{pr}} - \hat{\mathbf{x}}_{\text{pr}}\|_2 \\ &\leq g_r^2(s,\mathbf{p})\|\mathbf{Q}(s,\mathbf{p})\|_2.\end{aligned}$$

□

In Theorem 2.3, note that because $g_n(s,\mathbf{p})$ must be identically zero, we have that $g_r(s,\mathbf{p}) \to 0$ as r approaches n. So from the result of the Theorem, when $\hat{\mathbf{x}}$ approximates $\mathbf{x}$ well, $\Delta_1(s,\mathbf{p})$ becomes a very close approximation to $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$. Because of this, even poor primal and dual ROMs in practice still yield good approximate bounds on the point wise transfer function error $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$. Each approximate upper and lower bound in Table 2.1 incurs a similar error term which decays quickly as the reduced order r is increased.

Note that for symmetric systems (systems that satisfy $\mathbf{Q}(s,\mathbf{p}) = \mathbf{Q}^\top(s,\mathbf{p})$ and $\mathbf{b}(\mathbf{p}) = \mathbf{c}(\mathbf{p})$ for all s and $\mathbf{p}$), according to [32, Remark 4.1] $\Delta_1 \equiv 0$ for all s and $\mathbf{p}$. This can be seen by fully expanding $|\hat{\mathbf{x}}_{\text{du}}^\top \mathbf{r}_{\text{pr}}|$ and using the fact that $\mathbf{V}_{\text{pr}} = \mathbf{W}_{\text{du}}$ and $\mathbf{V}_{\text{du}} = \mathbf{W}_{\text{pr}}$. Having $|\hat{\mathbf{x}}_{\text{du}}^\top \mathbf{r}_{\text{pr}}| \equiv 0$ suggests that the *true* error in Δ_1 is entirely determined by $|\mathbf{x}_{\text{du}}^\top \mathbf{r}_{\text{pr}}|$, which only decreases linearly with the order of the reduced model given our prior assumptions. One proposed solution is to construct $\mathbf{V}_{\text{pr}}$ differently from $\mathbf{V}_{\text{du}}$, in a similar manner to how

$\mathbf{V}_{\mathrm{r_{pr}}}$ is constructed differently than $\mathbf{V}_{\mathrm{pr}}$. [32, Remark 4.4] suggests that $\Delta_1^{\mathrm{pr}} \approx 0$ as well if $\mathbf{V}_{\mathrm{pr}} \approx \mathbf{V}_{\mathrm{r_{pr}}}$, although this is unlikely considering the intentionally different construction of the two projections. Note that even if an approximate lower bound is numerically zero, the approximate upper bound remains unaffected by the system being symmetric.

2.3.1 Multi-Moment Matching with Efficient Error Estimates

Feng and Benner [32] propose a version of Algorithm 2 which efficiently approximates the $\mathcal{H}_\infty$ norm in step 5 of Algorithm 2 in a method we will refer to as the moment matching method (MMM). Using the point-wise estimates of the transfer function error given in Table 2.1, Feng and Benner [32] sample the approximations on a user-defined grid in the joint frequency domain and parameter space. They then approximate the $\mathcal{H}_\infty$ norm (2.6) by taking the maximum of the 2-norm of their samples. Their method for constructing the local projection matrices $\mathbf{V}_{\mathrm{local}}$ is a numerically stable method for implicit parametric moment matching that they introduced in 2014 [18]. We will first outline the details of their method for constructing $\mathbf{V}_{\mathrm{local}}$ before discussing how they adapted it to the framework of Algorithm 2.

Assume that $\mathbf{Q}(s, \mathbf{p})$ has the affine decomposition $\mathbf{Q}(s, \mathbf{p}) = \mathbf{Q}_0 + h_1(s, \mathbf{p})\mathbf{Q}_1 + \ldots + h_m(s, \mathbf{p})\mathbf{Q}_m$. For example, given the system (2.10) with $\mathbf{A}(\mathbf{p}) = \mathbf{A}_c + \mathbf{p}\mathbf{A}_{\mathbf{p}}$, we might write $\mathbf{Q}(s, \mathbf{p}) = -\mathbf{A}_c + s\mathbf{E} - \mathbf{p}\mathbf{A}_{\mathbf{p}}$. We can then expand the solution to the primal system around the location

$(h_1(s_0, \mathbf{p}_0), ..., h_m(s_0, \mathbf{p}_0))$ as follows:

$$\begin{aligned}
\mathbf{x}(s, \mathbf{p}) &= \mathbf{Q}^{-1}(s, \mathbf{p})\mathbf{b}(\mathbf{p}) \\
&= [\mathbf{Q}_0 + h_1(s, \mathbf{p})\mathbf{Q}_1 + ... + h_m(s, \mathbf{p})\mathbf{Q}_m]^{-1}\mathbf{b}(\mathbf{p}) \\
&= [\mathbf{I} - (-\sigma_1\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_1 - ... - \sigma_m\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_m)]^{-1}\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{b}(\mathbf{p}) \\
&= \sum_{k=0}^{\infty}(-\sigma_1\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_1 - ... - \sigma_m\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_m)^k\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{b}(\mathbf{p}),
\end{aligned}$$

where $\sigma_j = h_j(s, \mathbf{p}) - h_j(s_0, \mathbf{p}_0)$. Under the simplifying assumption that $\mathbf{b}$ does not depend on $\mathbf{p}$, we can recursively write the terms in this power series by

$$\begin{aligned}
\mathbf{R}_0 &= \mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{b} \\
\mathbf{R}_1 &= [-\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_1\mathbf{R}_0, ..., -\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_m\mathbf{R}_0] \\
\mathbf{R}_2 &= [-\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_1\mathbf{R}_1, ..., -\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_m\mathbf{R}_1] \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
\mathbf{R}_q &= [-\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_1\mathbf{R}_{q-1}, ..., -\mathbf{Q}^{-1}(s_0, \mathbf{p}_0)\mathbf{Q}_m\mathbf{R}_{q-1}].
\end{aligned}$$

It is clear that if $\mathbf{V}$ contains $\mathbf{R}_0$ in its span, then the ROM's transfer function constructed with $\mathbf{V}$ will interpolate the full-order at $(s_0, \mathbf{p}_0)$, just as single-sided IRKA does. For $\mathbf{R}_q$, note that this is equivalently enforcing interpolation on

$$\mathbf{x}(s, p) = (-1)^q\mathbf{Q}^{-(q+1)}(s, \mathbf{p})\mathbf{Q}_{k_1} \cdots \mathbf{Q}_{k_m}\mathbf{b}$$

for all combinations of $k_1, k_2, ..., k_m$ that sum to q, which are exactly the m th order derivatives of the full-order model with respect to $h_1, h_2, ..., h_m$, up to a scalar. Then, any $\mathbf{V}$ whose span is equal to span$\{\mathbf{R}_0, \mathbf{R}_1, ..., \mathbf{R}_q\}$ will cause the ROM's transfer function to interpolate the full-order system's transfer function at $(s_0, \mathbf{p}_0)$, as well as all derivatives of the transfer

function up to order q. In practice, we set

$$\mathbf{V} = \text{orth}\{\text{real}\{\mathbf{R}_0, \mathbf{R}_1, ...\mathbf{R}_q\}, \text{imag}\{\mathbf{R}_0, \mathbf{R}_1, ...\mathbf{R}_q\}\}$$

to keep the ROM real-valued and preserve the interpolation properties. Note that $\mathbf{R}_q \in \mathbb{C}^{n \times m^q}$ (unless some columns are removed in the orthogonalization), so q is usually set to be 0 or 1 to limit the exponential growth in the number of columns of $\mathbf{V}$. See [18, 32] for a more in-depth discussion of forming the local basis matrix $\mathbf{V}$, including the steps needed when $\mathbf{b}$ is parameter-dependent.

With MMM now defined in a local ROM construction sense, we move on to how the point wise error approximations were used with MMM to come up with a parametric model reduction scheme. Four different algorithms are presented in [32] - two for non-parametric model reduction, two for parametric, two for generic systems, and two specifically for improving the approximation behavior on symmetric systems. We will broadly discuss their algorithm for general parametric systems.

Feng and Benner [32] begin with a mesh on the joint parameter and frequency space, and select a point from the mesh to be the first local ROM. After updating all four global ROMs, they compute the upper bound estimates given in Table 2.1. They select where to construct the next local ROM based on which location maximizes the 2-norm of the point wise errors across all upper bound estimates. After identifying this location in the parameter space as well as which upper bound estimate was maximized, they perform a search along the imaginary axis to find the frequency that maximizes the corresponding range variable in Table 2.1. In this way, they use separate expansion points for their different global ROMs, while intentionally sampling to decrease the ranges of the approximations.

Chapter 3

$\mathcal{H}_2$ PROM with Reduced-Order Surrogates

In this chapter, we adapt the point wise transfer function error approximations given in Table 2.1 to form efficient approximations to the $\mathcal{H}_2$ error in Section 3.1. We test these approximations on a convection-diffusion flow model with a parameter dependency in the convection term. In Section 3.2, we implement these approximations in our own version of Algorithm 2, using the mean plus standard deviation of the approximations as our objective function. In Section 3.2.1, we demonstrate that the estimated improvement algorithm (a type of Bayesian optimization) is an effective and efficient choice of optimization algorithm to use for step 5 of Algorithm 2. We then apply our algorithm (Algorithm 3) to the convection-diffusion flow model in Section 3.2.2, as well as an Euler-Bernoulli cantilever beam model with variable stiffness in Section 3.2.3, and a multiple-parameter thermal model with variable film coefficients in Section 3.2.4. We end this chapter with a simple extension of our computations to the MIMO case, and demonstrate on a MIMO variation of the convection-diffusion flow model in Section 3.3.

3.1 Efficient $\mathcal{H}_2$ Error Estimates for Non-Parametric Systems

In this section we propose a method of quickly computing an approximation to the $\mathcal{H}_2$ error between a full-order model and its reduced-order model by using several upper and lower bound estimates provided in Table 2.1. First, we only work with non-parametric systems (parametric systems with fixed $\mathbf{p}$'s), but later expand upon this into a scheme for efficient parametric model reduction in Section 3.2.

Recall the definition of the $\mathcal{H}_2$ norm (2.7) and note that for SISO systems, the $\mathcal{H}_2$ error becomes an integral of an absolute value of a complex scalar-valued function. We can then use a quadrature rule $\{\mathrm{i}x_j, w_j\}_{j=0}^m$ to approximate this integral:

$$\begin{aligned} \|H(s,\mathbf{p}) - H_r(s,\mathbf{p})\|_{\mathcal{H}_2} &= \left(\frac{1}{2\pi}\int_{-\infty}^{\infty} |H(\mathrm{i}\omega,\mathbf{p}) - H_r(\mathrm{i}\omega,\mathbf{p})|^2 \mathrm{d}\omega\right)^{1/2} \\ &\approx \left(\frac{1}{2\pi}\sum_{j=0}^{m} w_j |H(\mathrm{i}x_j,\mathbf{p}) - H_r(\mathrm{i}x_j,\mathbf{p})|^2\right)^{1/2}. \end{aligned} \tag{3.1}$$

From this, it directly follows that we can integrate approximate lower bounds to $|H(\mathrm{i}\omega,\mathbf{p}) - H_r(\mathrm{i}\omega,\mathbf{p})|$ to arrive at an approximate lower bound for $\|H(s,\mathbf{p}) - H_r(s,\mathbf{p})\|_{\mathcal{H}_2}$, and similar for upper bounds. We show this more formally in the following proposition.

Proposition 3.1. *Let H_r be the transfer function for a reduced system modelling H with reduced order r, and let $\Delta_\ell(s,\mathbf{p})$ be a lower bound as in Table 2.1, satisfying $\Delta_\ell(s,\mathbf{p}) \leq |H(s,\mathbf{p}) - H_r(s,\mathbf{p})| + \varepsilon(s,\mathbf{p})$, where $\Delta_\ell(s,\mathbf{p}) \geq 0$ and $\varepsilon(s,\mathbf{p}) \geq 0$ for all s and p. Similarly, let $\Delta_u(s,\mathbf{p})$ be an upper bound as in Table 2.1, satisfying $\Delta_u(s,\mathbf{p}) \geq |H(s,\mathbf{p}) - H_r(s,\mathbf{p})| - \varepsilon(s,\mathbf{p})$, where $\Delta_u(s,\mathbf{p}) \geq 0$ and $\varepsilon(s,\mathbf{p}) \geq 0$ for all s and p. Suppose $\varepsilon(s,\mathbf{p}) \leq M_\varepsilon$ for all s and $\mathbf{p}$. Then, for any exact quadrature rule for integrating Δ_ℓ and Δ_u with respect to s over the*

imaginary axis given by $\{\mathrm{i}x_j, w_j\}_{j=0}^m$*, it holds that*

$$\left(\frac{1}{2\pi}\sum_{j=0}^{m} w_j \Delta_\ell^2(\mathrm{i}x_j, \mathbf{p})\right)^{1/2} \leq \|H(s,\mathbf{p}) - H_r(s,\mathbf{p})\|_{\mathcal{H}_2} + \mathcal{O}(M_\varepsilon), \tag{3.2}$$

$$\|H(s,\mathbf{p}) - H_r(s,\mathbf{p})\|_{\mathcal{H}_2} \geq \left(\frac{1}{2\pi}\sum_{j=0}^{m} w_j \Delta_u^2(\mathrm{i}x_j, \mathbf{p})\right)^{1/2} + \mathcal{O}(M_\varepsilon), \tag{3.3}$$

with the error in the norm approximation being dependent on the accuracy of the reduced models required to compute Δ_ℓ *and* Δ_u *and the quadrature rule.*

Proof. Let $\mathbf{p}$ be fixed. Assume that we have a quadrature rule $\{\mathrm{i}x_j, w_j\}_{j=0}^m$ as in (3.1). Note that for SISO systems,

$$\|H(\mathrm{i}x_j,\mathbf{p}) - H_r(\mathrm{i}x_j,\mathbf{p})\|_{\mathrm{F}}^2 = |H(\mathrm{i}x_j,\mathbf{p}) - H_r(\mathrm{i}x_j,\mathbf{p})|^2.$$

Then,

$$\|H(s,\mathbf{p}) - H_r(s,\mathbf{p})\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi}\sum_{j=0}^{m} w_j \|H(\mathrm{i}x_j,\mathbf{p}) - H_r(\mathrm{i}x_j,\mathbf{p})\|_{\mathrm{F}}^2 := \hat{H}_e(\mathbf{p}).$$

By the fact that $(\Delta(s,\mathbf{p}) - \varepsilon(s,\mathbf{p}))^2$ is a lower bound for

$$|H(s,\mathbf{p}) - H_r(s,\mathbf{p})|^2 = \|H(\mathrm{i}x_j,\mathbf{p}) - H_r(\mathrm{i}x_j,\mathbf{p})\|_{\mathrm{F}}^2,$$

we have that

$$\hat{H}_e(\mathbf{p}) \geq \frac{1}{2\pi}\sum_{j=0}^{m} w_j (\Delta_\ell(\mathrm{i}x_j,\mathbf{p}) - \varepsilon(\mathrm{i}x_j,\mathbf{p}))^2.$$

If we expand the squared term and collect the ε terms, then

$$\begin{aligned}
\hat{H}_e(\mathbf{p}) &\geq \frac{1}{2\pi}\sum_{j=0}^{m} w_j \left(\Delta_\ell^2(\mathrm{i}x_j,\mathbf{p}) - w_j\varepsilon(\mathrm{i}x_j,\mathbf{p})(2\Delta_\ell(\mathrm{i}x_j,\mathbf{p}) - \varepsilon(\mathrm{i}x_j,\mathbf{p}))\right) \\
&= \frac{1}{2\pi}\sum_{j=0}^{m} w_j\Delta_\ell^2(\mathrm{i}x_j,\mathbf{p}) \\
&\qquad - \frac{1}{2\pi}\sum_{j=0}^{m} w_j\varepsilon(\mathrm{i}x_j,\mathbf{p})(2\Delta_\ell(\mathrm{i}x_j,\mathbf{p}) - \varepsilon(\mathrm{i}x_j,\mathbf{p})) \\
\hat{H}_e(\mathbf{p}) &\geq \frac{1}{2\pi}\sum_{j=0}^{m} w_j\Delta_\ell^2(\mathrm{i}x_j,\mathbf{p}) - \frac{M_\varepsilon}{2\pi}\sum_{j=0}^{m} w_j(2\Delta_\ell(\mathrm{i}x_j,\mathbf{p}) - \varepsilon(\mathrm{i}x_j,\mathbf{p})) \\
&= \frac{1}{2\pi}\sum_{j=0}^{m} w_j\Delta_\ell^2(\mathrm{i}x_j,\mathbf{p}) - \mathcal{O}(M_\varepsilon).
\end{aligned}$$

The proof for Δ_u is similar. □

Using this proposition, we can then form three approximate pairs of upper and lower bounds for the $\mathcal{H}_2$ error using the approximations given in Table 2.1. Integrating Δ_1 and Δ_1^{pr} yields two approximate lower bounds, and integrating $\Delta_1 + \Delta_2$, $\Delta_1 + \Delta_2^{\mathrm{pr}}$, and $\Delta_1^{\mathrm{pr}} + \Delta_3$ yields three approximate upper bounds. Note that the bound on the point wise error M_ε decays rather quickly with respect to increasing the reduced order r (see Theorem 2.2). In practice though, the approximate bounds are rarely strict, since the quadrature rule used is rarely exact. Despite this, these approximations are not only accurate, but cheap. Specifically, each $\mathcal{H}_2$ error approximation requires in the worst case four evaluations of reduced-order solutions and two evaluations of multiplying $\mathbf{Q}(s,\mathbf{p})$ against a vector, for each quadrature node. This entirely eliminates the need to compute full-order Lyapunov solutions (for the true $\mathcal{H}_2$ error) or to compute $\mathbf{Q}^{-1}(s,\mathbf{p})$ (for quadrature directly on the primal system). This comes at the cost of constructing four different reduced models, with the two residual ROMs necessarily being of a larger order than the primal and dual ROMs. We demonstrate the effectiveness of these approximations on a convection-diffusion flow model.

3.1.1 A Convection-Diffusion Flow Model

A convection-diffusion flow model is often used as a benchmark for studying dynamical systems due to its first-order LTI structure. The partial differential equation (PDE) governing the convection and diffusion of a viscous fluid in two dimensions is given by

$$\frac{\partial \mathbf{x}}{\partial t}(t,\xi) = a\Delta \mathbf{x}(t,\xi) + \mathbf{p}\nabla \mathbf{x}(t,\xi) + b(\xi)u(t), \quad \xi \in \Omega, t \in (0,\infty),$$
$$\mathbf{x}(t,\xi) = \mathbf{0}, \quad \xi \in \partial\Omega,$$

where $\mathbf{p}$ is the parameter controlling the convection in both dimensions, and a is the parameter controlling the diffusion. We fix a at 2.5. Using a square mesh on Ω with $n = 400$ total elements, we can use a centered five-point finite difference stencil to approximate the diffusion (second order terms), and a centered three-point stencil to approximate the convection in each direction (the first order spacial derivatives). Although $n = 400$ is small for a full-order, we chose it such that we could easily compute the true $\mathcal{H}_2$ errors and compare them with our approximations. Taking $\mathbf{E} = \mathbf{I}_n$, and introducing a $\mathbf{b}$ vector and a $\mathbf{c}$ vector where the first element of each vector is one and all other elements are zeros, we arrive at the SISO first-order model

$$\mathbf{E}\dot{\mathbf{x}}(t,\mathbf{p}) = (\mathbf{A}_d + \mathbf{p}\mathbf{A}_c)\mathbf{x}(t,\mathbf{p}) + \mathbf{b}u(t),$$
$$y(t,\mathbf{p}) = \mathbf{c}^\top \mathbf{x}(t,\mathbf{p}),$$

where $\mathbf{A}_d$ governs the diffusion of the system, and $\mathbf{A}_c$ governs the convection of the system. This system's transfer function is given by

$$H(s,\mathbf{p}) = \mathbf{c}^\top \mathbf{Q}(s,\mathbf{p})^{-1}\mathbf{b},$$

where $\mathbf{Q}(s,\lambda) = s\mathbf{E} - (\mathbf{A}_d + \lambda\mathbf{A}_c)$. Note that with a diffusion coefficient of 1 and a discretization of 10 points in each dimension on Ω, we calculate the cell Reynolds number for this system to be $\frac{0.1\mathbf{p}}{2}$. To ensure meaningful results, we set the range of $\mathbf{p}$ to be $[10^{-3}, 10^{2}]$ to keep the cell Reynolds number below 1. Similarly, we note that both the mesh Péclet number and the CFL condition number for this system remain below 1 for the entire parameter range. When using IRKA on this model, we use a logarithmic frequency sampling range between $\mathrm{i} \cdot 10^{-4}$ and $\mathrm{i} \cdot 10^{4}$, including its conjugate, for IRKA's initial frequency samples.

3.1.2 Non-Parametric Convection-Diffusion Flow Results

In order to demonstrate the $\mathcal{H}_2$ error approximations, we took our convection-diffusion flow model and reduced it via IRKA at two separate parameter values. We then constructed the four global ROMs necessary to form the error approximations, and compared these approximate $\mathcal{H}_2$ estimates to the true $\mathcal{H}_2$ error when we vary the size of the reduced order. In our computations, we worked with the relative $\mathcal{H}_2$ error by dividing all errors by the $\mathcal{H}_2$ norm of the full-order system at the given p values. Dividing by the norm of the full system is only computationally tractable for full-orders that are not too large, and in Section 3.2 we replace this term with the norm of the reduced model for exactly this reason.

We found that for single-sided IRKA, the error approximations were almost exactly within 1% of the true relative $\mathcal{H}_2$ error for every order tested. As for double-sided IRKA, although the true error converged to zero more quickly (since double-sided IRKA is a more accurate method), the first two lower bounds were numerically zero and the third pair of bounds proved to not be very accurate past an order of 6. This suggests that with double-sided IRKA, the $\mathbf{V}_{\mathrm{r_{pr}}}$ and $\mathbf{V}_{\mathrm{r_{du}}}$ matrices were not sufficiently distinct from $\mathbf{V}_{\mathrm{pr}}$ and $\mathbf{V}_{\mathrm{du}}$ for using the third pair of bounds. Although the strategy for constructing $\mathbf{V}_{\mathrm{r_{pr}}}$ and $\mathbf{V}_{\mathrm{r_{du}}}$ is fairly

low-effort, the residual systems were evidently able to model the residual quite well in the single-sided IRKA case. A comparison of all three upper and lower bounds, as well as their average error across both parameters for both single-sided IRKA and double-sided IRKA is given in Figure 3.1. The corresponding upper and lower bounds are plotted with the same colors.

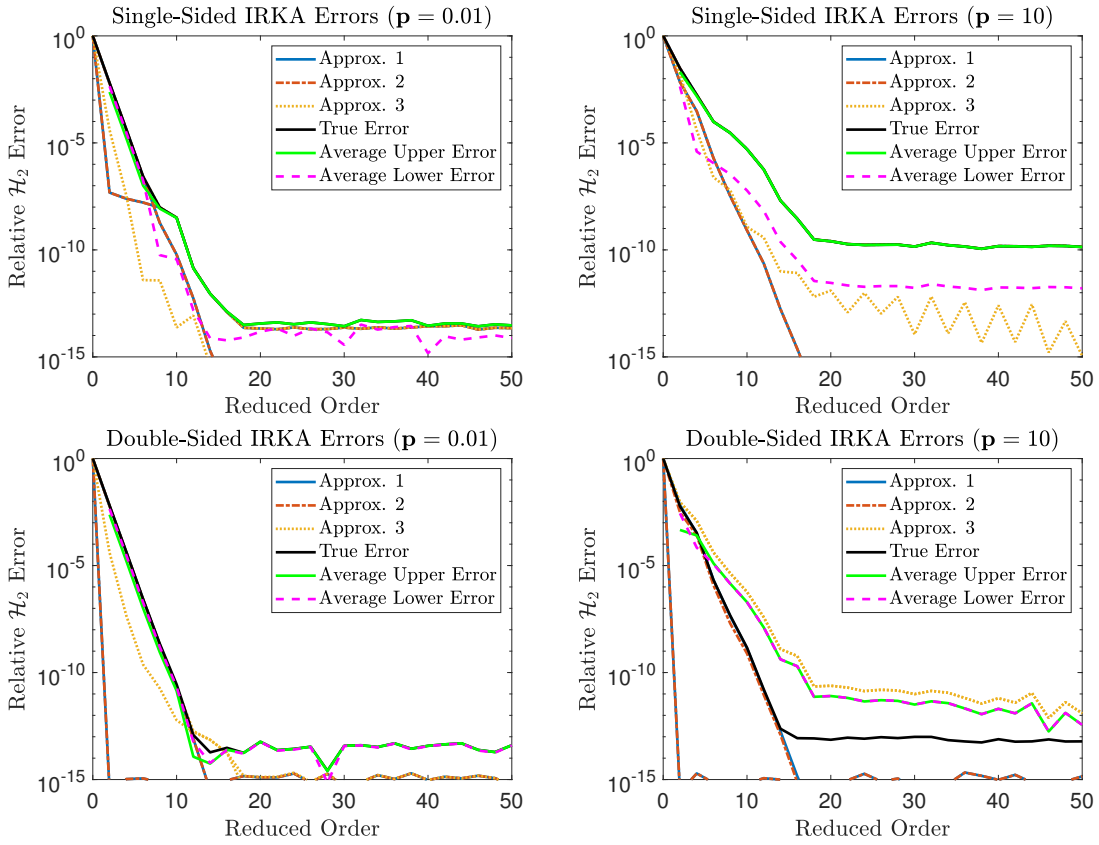


Figure 3.1: Non-parametric error approximations on convection-diffusion flow

We next symmetrized our convection-diffusion flow model by taking $\mathbf{A}_c \leftarrow \frac{1}{2}\mathbf{A}_c + \frac{1}{2}\mathbf{A}_c^\top$, and similar for $\mathbf{A}_d$. Although this is no longer a convection-diffusion model, our goal was to test the effectiveness of these various approximations even when some of the lower bounds would be numerically zero, as was proven would happen for symmetric systems in Section 2.3. As expected, the lower bounds for the first two approximations were numerically zero throughout, but the upper bounds remained tight. The results for $\mathbf{p} = 10$ are given in Figure

3.2. Note that for symmetric systems, double-sided IRKA is the same as single-sided IRKA, which is why the error converges to zero rather quickly.

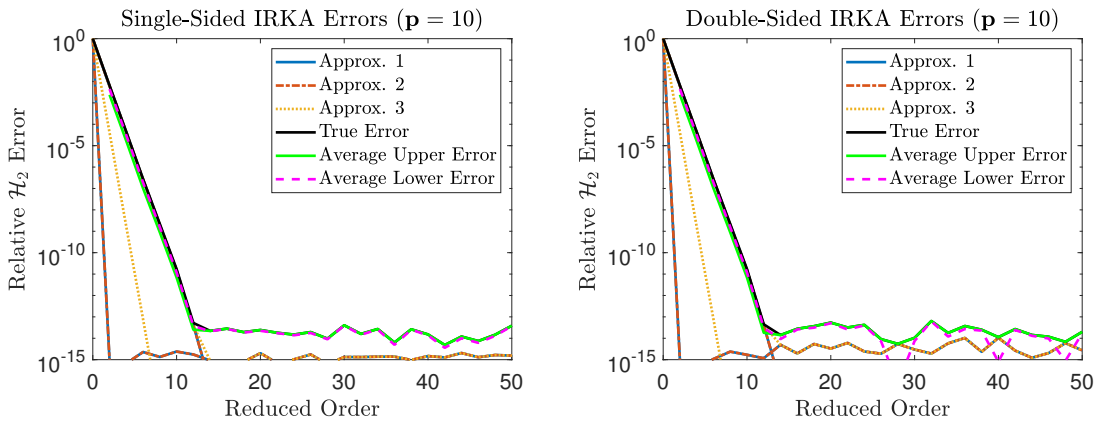


Figure 3.2: Non-parametric error approximations on symmetrized convection-diffusion flow

We also tested the error approximations on a cantilever beam model of full-order 200 after lifting (see Section 3.2.3 for more details), at a parameter value of 10^{-1}. We only tested this model with single-sided IRKA. This example proved to be much more difficult than the convection-diffusion flow model, as a reduced-order of 10 in the beam model accounted for a relative $\mathcal{H}_2$ error of 1%, whereas in the convection-diffusion flow model, a reduced order of 10 led to a relative error of 10^{-6} in the more challenging parameter range. The first two pairs of approximations performed quite well, despite the slow decrease in relative error. The third pair of approximations frequently overestimated the error, which resulted in a number of spikes in the error at certain reduced orders. This is due to the value of the transfer function at the first indexed finite element node being innaccurate (recall that the third lower bound is $|\mathbf{c}^\top \hat{\mathbf{x}}_{\mathrm{r_{pr}}}|$, and $\mathbf{c}$ in this example only observes the first indexed node). The results are shown in Figure 3.3. Again, the corresponding upper and lower bounds are plotted in the same colors.

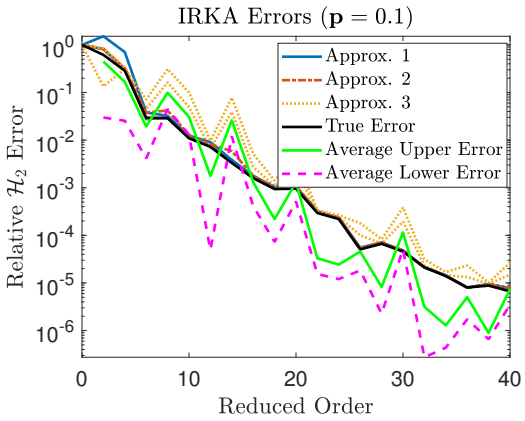


Figure 3.3: Non-parametric error approximations on cantilever beam model

3.2 Efficient $\mathcal{H}_2$ Error Estimates for Parametric Systems

Now that we have a way of approximating the $\mathcal{H}_2$ error efficiently, we will use this to modify the greedy selection step (step 5) of Algorithm 2 in our own implementation of this algorithm. Considering that the error approximations in Table 2.1 were developed for usage in parametric model reduction, we expect our approximations to the $\mathcal{H}_2$ norm to also be well-suited for the parametric model reduction case. For constructing the projection matrices, we will make use of single-sided IRKA (and double-sided where applicable). Note that IRKA (Algorithm 1) requires more than a few full-order linear system solves in order to generate even a single local basis matrix. Other methods can be used instead, however we used IRKA here because of its local $\mathcal{H}_2$-optimality. See, for example, [34, 42], which construct non-greedy projection-based parametric reduced models to minimize a joint error $\mathcal{H}_2 \times L_2$ measure in the frequency and parameter domain. Moreover, see [8, 15], which construct a sampling-free parametric reduced model for special parametric structures.

To begin, we will need to keep track of four different ROMs - the primal, dual, primal residual, and dual residual ROMs. Assuming IRKA accepts the following input format, we

can compute the appropriate projection matrices by the following:

$$\begin{aligned}
&\mathbf{V}_{\mathrm{pr}} \leftarrow \mathrm{IRKA}(Q, b, c, s);\\
&\mathbf{V}_{\mathrm{du}} \leftarrow \mathrm{IRKA}(Q^\top, c^\top, b^\top, s);\\
&\mathbf{V}_{\mathrm{r_{pr}}} \leftarrow \mathrm{IRKA}(Q, b, c, s_{\mathrm{alt}}); \qquad \mathbf{V}_{\mathrm{r_{pr}}} \leftarrow \mathrm{orth}\{\mathbf{V}_{\mathrm{r_{pr}}}, \mathbf{V}_{\mathrm{pr}}\};\\
&\mathbf{V}_{\mathrm{r_{du}}} \leftarrow \mathrm{IRKA}(Q^\top, c^\top, b^\top, s_{\mathrm{alt}}); \quad \mathbf{V}_{\mathrm{r_{du}}} \leftarrow \mathrm{orth}\{\mathbf{V}_{\mathrm{r_{du}}}, \mathbf{V}_{\mathrm{du}}\};
\end{aligned} \tag{3.4}$$

where s and s_{alt} are the initial frequency distributions. Recall that $\mathbf{V}_{\mathrm{r_{pr}}}$ and $\mathbf{V}_{\mathrm{r_{du}}}$ need to not only include the information in $\mathbf{V}_{\mathrm{pr}}$ and $\mathbf{V}_{\mathrm{du}}$, respectively, but must have additional information in order for the residual systems to be nonzero. We achieve this by giving IRKA a different initial frequency distribution when constructing the parts of $\mathbf{V}_{\mathrm{r_{pr}}}$ and $\mathbf{V}_{\mathrm{r_{du}}}$ that need to be distinct, in addition to terminating IRKA early. With this particular method for constructing the projection matrices, note that the residual system projection matrices will be up to twice the reduced order of the primal ROM (the only ROM we ultimately care about). Although this might not seem ideal, considering we desire $r \ll n$, $2r$ should not be much of an additional burden.

Note that in Algorithm 2, the reduced orders of the local projection matrices are not specified. When using IRKA, this translates to the number of frequency points given in s and s_{alt}. Since we have the ability then to easily vary the reduced order of each local ROM, two fundamental approaches to parametric model reduction become open to us. We can either perform a *depth-first* order reduction, or a *breadth-first* order reduction. In the depth-first setting, we create a small local ROM at parameter p, and if the global ROM does not achieve the desired tolerance at p, we gradually increase the order of the local ROM at p until tolerance is achieved in the global ROM. In the breadth-first setting, we always construct local ROMs of the minimum desired (or minimum possible) order, unless the parameter that maximizes the error is one we have already sampled (or is sufficiently close to a previous sample), in

which case we increase the order of the next local ROM until an unsampled parameter is the next to maximize the error. This behavior of repeated sampling is only likely to occur for low dimensional parameter spaces, and usually on the boundaries.

Comparing these two approaches, the depth-first approach would require fewer optimization calls since the majority of the time we would only need to sample the error at the same parameter repeatedly. However, we intuitively expect that the breadth-first approach yields a more evenly-distributed error, and possibly a smaller final reduced order of the global model due to having richer projection matrices. Of course, how much information a projection matrix holds might be highly dependent on the problem. We ultimately decided on the breadth-first approach due to our belief that without a guarantee of errors decreasing at all parameter locations with each iteration, the depth-first approach has more potential backtracking to do than the breadth-first approach. A comparison of the two approaches across a variety of model reduction techniques should be investigated in the future.

It now only remains to define the optimization problem. Our goal is to solve

$$\arg\max_{\mathbf{p}\in\mathcal{P}} \|H(s,\mathbf{p}) - H_r(s,\mathbf{p})\|_{\mathcal{H}_2}$$

. Define the mean operator to be $\mu(X_1,...,X_n) = \frac{1}{n}\sum_{i=1}^{n} X_i$ and the standard deviation operator to be $\sigma(X_1,...,X_n) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu(X_1,...,X_n))^2}$, where n is finite and the X_is are random variables. Let $\ell_1(\mathbf{p}), \ell_2(\mathbf{p}), \ell_3(\mathbf{p})$ be the three lower bounds for the $\mathcal{H}_2$ norm defined by integrating the three lower bounds in Table 2.1 with respect to s using some fixed quadrature rule $\{\mathrm{i}x_j, w_j\}_{j=0}^{m}$, and similar for $u_1(\mathbf{p}), u_2(\mathbf{p}), u_3(\mathbf{p})$ with the three upper bounds. We define the objective function to then be

$$f_{\text{obj}}(\mathbf{p}) = \mu(\ell_1(\mathbf{p}), \ell_2(\mathbf{p}), \ell_3(\mathbf{p}), u_1(\mathbf{p}), u_2(\mathbf{p}), u_3(\mathbf{p})) + \sigma(\ell_1(\mathbf{p}), \ell_2(\mathbf{p}), \ell_3(\mathbf{p}), u_1(\mathbf{p}), u_2(\mathbf{p}), u_3(\mathbf{p})). \tag{3.5}$$

This objective function allows the optimization algorithm to converge at not only parameters that have actual large errors but also at regions of the parameter space over which the error estimates are poor. Although sampling in such a region might not help the accuracy of the primal ROM by much, it will allow for more accurate error estimations and more precise selection of parameters with actual large errors in future iterations.

This objective function has one other very useful property, which relates to some of the lower bounds being zero in special cases. Since all lower and upper bounds are positive, setting one or more lower bounds to zero will decrease the value of the mean. It can be shown that the mean plus standard deviation is an upper bound to using the mean by itself that is somewhat robust to changes in the approximations. The details for the robustness of the objective function can be found in Section A.2. Recalling Section 2.3, some of the lower bounds for $|H(s,\mathbf{p}) - \hat{H}(s,\mathbf{p})|$ can be numerically zero for all s and $\mathbf{p}$, and therefore not very useful, if certain conditions about the full-order model and the ROMs are met. With our choice of objective function, we do not have to introduce a different algorithm for handling such a case. Compare this to the two separate implementations of MMM for symmetric and non-symmetric systems [32]. Our complete algorithm for parametric model reduction that makes use of our efficient $\mathcal{H}_2$ error approximations is given in Algorithm 3. The main difference between Algorithm 2 and Algorithm 3 is that we replace step 5 in Algorithm 2 with our objective function (3.5), which operates on efficient approximations to the $\mathcal{H}_2$ error.

3.2.1 Utilizing Bayesian Optimization

Our initial choice of optimization algorithm for solving step 5 of Algorithm 3 was MATLAB's local optimization algorithm `fminbnd` (or in the case of a multi-parameter model, `fmincon`). To coax this local optimization algorithm into behaving like a global optimization algorithm,

Algorithm 3 A Parametric IKRA Scheme with Efficient $\mathcal{H}_2$ Errors

1. Select $\mathbf{p}_0$ and set $\mathbf{V}_\mathrm{g} = [\,]$, minorder, increment
while error > tolerance **do**
  if already sampled at $\mathbf{p}_k$ **then**
    minorder $\leftarrow$ minorder + increment
  else
    reset minorder
  end if
  2. Construct all 4 local ROMs at $\mathbf{p}$ with IRKA and order minorder ▷ (3.4)
  3. Update $\mathbf{V}_\mathrm{g} = \mathrm{orth}\{\mathbf{V}_\mathrm{g}, \mathbf{V}_\mathrm{local}\}$ for each global projection matrix
  4. Construct the global primal ROM using $\mathbf{V}_\mathrm{g}$, obtaining H_{r_k} ▷ (2.9)
  5. Solve $\mathbf{p}_{k+1} = \arg\max_{\mathbf{p}\in\mathcal{P}} f_\mathrm{obj}(\mathbf{p})/\|H_{r_k}(\mathbf{p})\|_{\mathcal{H}_2}$ ▷ (3.5)
end while

we first sample the objective function on a coarse grid and initialize the optimization algorithm at the maximum on the grid, similar to modifying a local optimization algorithm by allowing it multiple starts. We used 10 logarithmically spaced points in each dimension to form our initial grid. In this section we will investigate using expected improvement, a Bayesian optimization algorithm, in place of our initial choice of optimization algorithm.

Bayesian optimization is a class of global optimization and sequential design strategies that falls into the categories of derivative-free optimization and surrogate optimization. They usually fit a Gaussian process [58] to values of the objective function in a process known as Kriging, then use statistical properties of the Gaussian process to determine where to sample next. Each Bayesian optimization algorithm depends on its *acquisition function*, which determines where to sample next by balancing *exploring* new regions of the domain and *exploiting* regions where we already know the value of the objective function is low. As a brief reminder, the standard for formulating an optimization problem is minimize the objective function. For our usage, it suffices to minimize the negative of the error function. For more information on Bayesian and surrogate optimization schemes, see [33].

The expected improvement algorithm [60] is one common example of Bayesian optimization.

In Figure 3.4 we show MATLAB's `bayesopt` implementing expected improvement as it fits and optimizes a Gaussian process to the negative of the relative $\mathcal{H}_2$ error. On the bottom of Figure 3.4, we observe that occasionally the Gaussian process does not fit the data very well and models all the observations as random noise, although this is rare. The example model being used here is the convection-diffusion flow model (Section 3.1.1) with a full-order of $n = 400$. We apply single-sided IRKA to reduce this system.

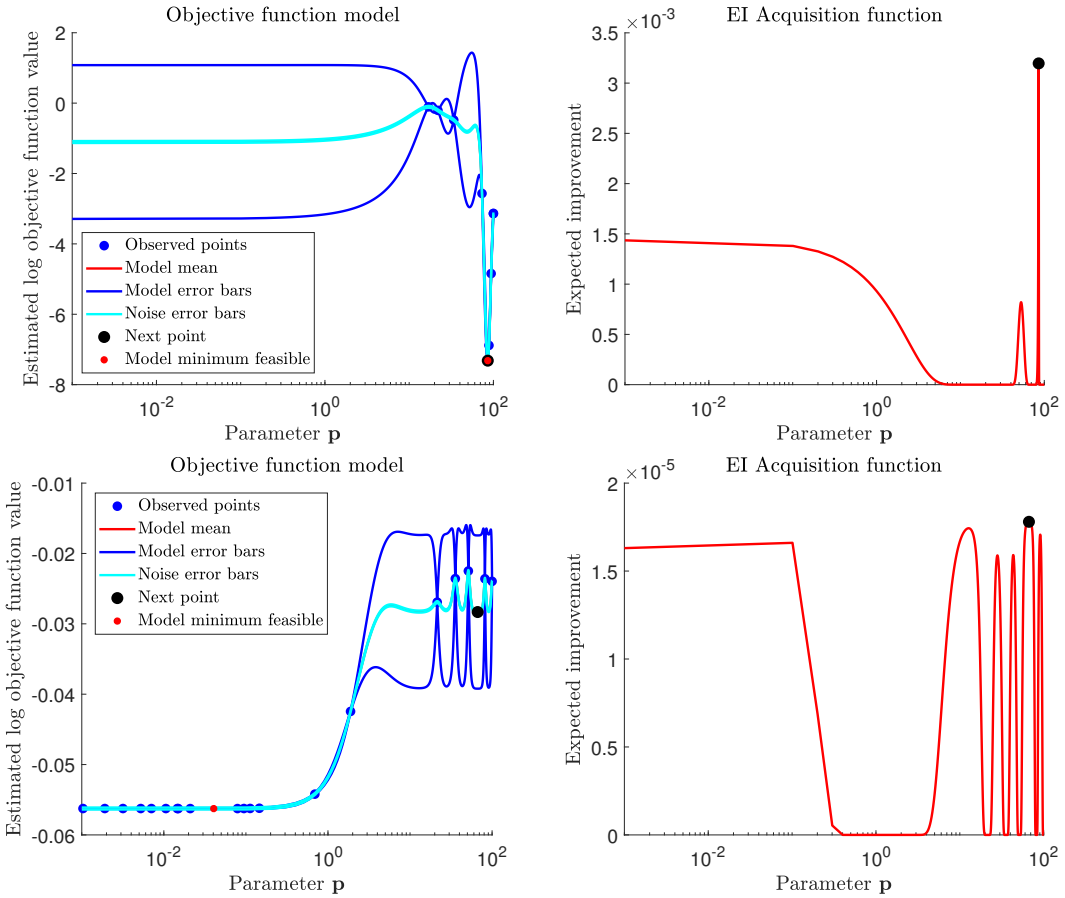


Figure 3.4: The expected improvement algorithm and acquisition function.

In Figure 3.5 we compare the final models formed by our original optimization algorithm, which uses a limit of 25 $\mathcal{H}_2$ approximations per optimization (including the coarse grid), expected improvement with a limit of 20 function evaluations, and expected improvement with a limit of 10 function evaluations. We see pretty clearly that there is a negligible

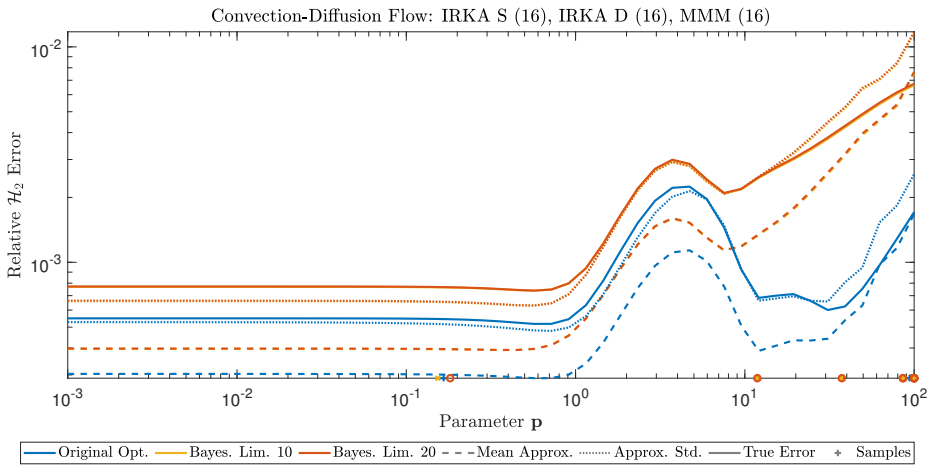
different between the two expected improvement models, and they even match the model created with our rather expensive optimization algorithm up until the final step. Recall that when a location is resampled, the order of the local ROM is increased. Expected improvement selected a slightly different location and did not expand the order of the local ROM, which is why the original optimization algorithm model performed better in the final step. This can be easily fixed by considering two locations to be the same if they are within a small radius of each other. These results indicate that, in the single parameter case at least, we can get away with only using 10 or fewer function evaluations in a search for the next parameter value. So, for increased efficiency in solving step 5 of Algorithm 3, we will use the expected improvement algorithm for our optimization algorithm.

3.2.2 Convection-Diffusion Flow Results

We first tested Algorithm 3 on the convection-diffusion flow example outlined in Section 3.1.1 with a full-order chosen to be $n = 400$. The structure of the system is restated for convenience:

$$\begin{aligned} \mathbf{E}\dot{\mathbf{x}}(t,\mathbf{p}) &= (\mathbf{A}_d + \mathbf{p}\mathbf{A}_c)\mathbf{x}(t,\mathbf{p}) + \mathbf{b}u(t), \\ y(t,\mathbf{p}) &= \mathbf{c}^\top \mathbf{x}(t,\mathbf{p}). \end{aligned}$$

We selected a minimum order of 2 to be added with each iteration to the global ROMs, and if we resampled a parameter value we attempted to then add a local ROM of order 4, then 6, and so on until the location of the maximum error changed. Note that all of these orders are even, since IRKA requires frequencies to be closed under conjugation. We selected a goal of achieving a relative error of 10^{-4}.



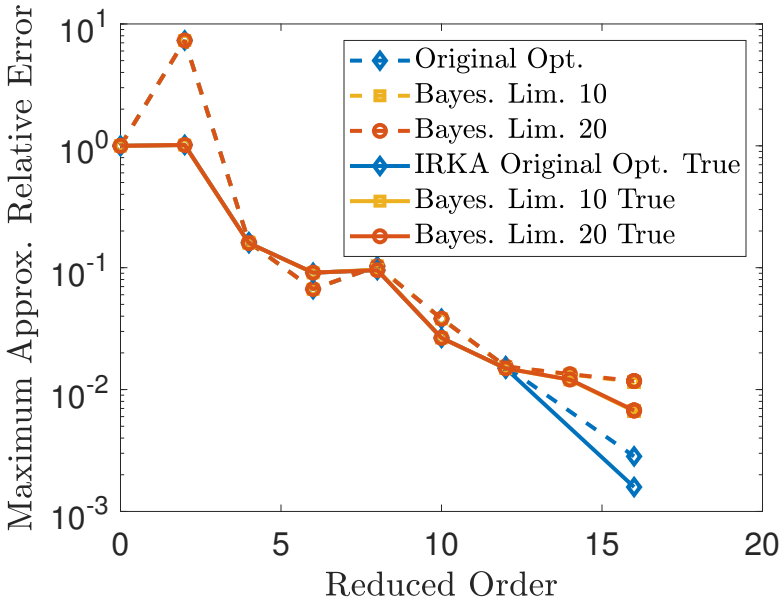


Figure 3.5: Comparing our expensive optimization algorithm to Bayesian optimization.

With this setup, we were able to produce a single-sided IRKA ROM of order 32 and a double-sided IRKA ROM of order 28. To guarantee stability of the double-sided IRKA model, we set $\mathbf{V} \leftarrow [\mathbf{V}, \mathbf{W}]$ and then used Galerkin projection with $\mathbf{W} \leftarrow \mathbf{V}$. Note that MMM is designed to minimize the $\mathcal{H}_\infty$ norm, not the $\mathcal{H}_2$ norm. All upper and lower bounds plus the true relative $\mathcal{H}_2$ errors are given in Figure 3.6. Note that all three approximate upper bounds are able to closely approximate the true error in shape, and are very close to

each other as well. The lower bounds are not as tight, but again capture the basic shape of the true error. When compared to the nonparametric results in Section 3.1.2, we see that sampling at multiple parameter values was able to dramatically increase the accuracy of the approximations in the higher parameter values. In Figure 3.7, we plot the single-sided IRKA and double-sided IRKA true errors, approximation means, and standard deviations next to each other along with the sampling locations. We also compare these results to our implementation of MMM on the same model. We made MMM comparable to our $\mathcal{H}_2$-based approach by slightly modifying MMM to have its exit condition be dependent on the $\mathcal{H}_2$ norm instead of the $\mathcal{H}_\infty$ norm, which only affects the number of iterations it takes.

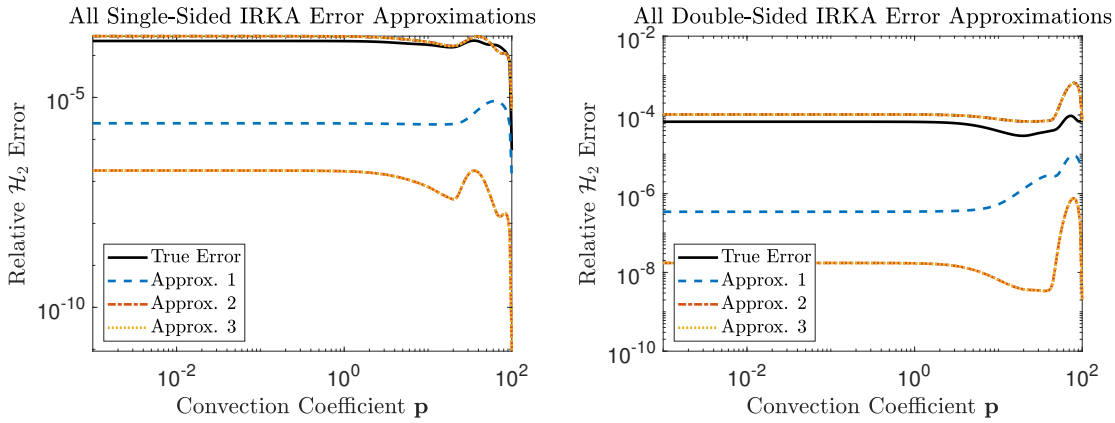


Figure 3.6: Comparison of the error approximations when using IRKA.

Note that our implementation of MMM did not place any samples at the more difficult end of the parameter space. Again, this is because MMM is an $\mathcal{H}_\infty$-based approach, not an $\mathcal{H}_2$-based approach. However, note that the placement of the samples by MMM suggests that the largest point wise errors occur at lower parameter values. Since the $\mathcal{H}_2$ errors are still on the order of 10^{-2} in this parameter range, the point wise errors along the imaginary axis at these lower parameter values must be low and very spread out. One more thing to observe from Figure 3.7 is that the true error of the double-sided IRKA model remains within one standard deviation of the mean of the approximations, even though two of the lower

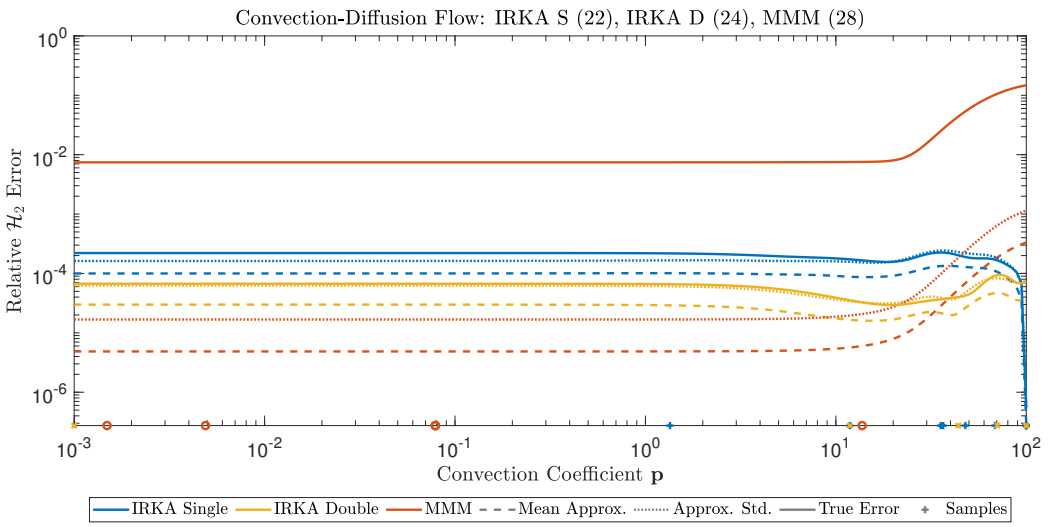


Figure 3.7: Direct comparison of the three models on the convection-diffusion flow example.

bounds are zero. This agrees with our theoretical expectations (See Section A.2). Despite the advantages of using IRKA over MMM in this setting, we note that IRKA (Algorithm 1) requires many more full-order linear system solves than MMM (see Section 2.3.1).

In Figure 3.8, we ran each of our three approaches until they could no longer add to the dimensions of $\mathbf{V}_{\text{pr}}$, then plotted the maximum of the objective function (3.5) versus the true error at that location for each of the models. In this regard, single-sided IRKA and double-sided IRKA performed similarly, with double-sided IRKA adding twice as many orders to the reduced system with each iteration. With MMM, we first note that the error approximations after the first and second iterations are greatly underpredicting the true error. This is due to how slowly MMM builds up its residual system ROMs. We also note that MMM stalls rather early, since our implementation of MMM cannot sample the same location twice and reduce the error. Finally, note that after the first few iterations, both IRKA methods were two orders of magnitude better than MMM. As a reminder, we should have expected some amount of this improvement of IRKA over our implementation of MMM, since IRKA is an

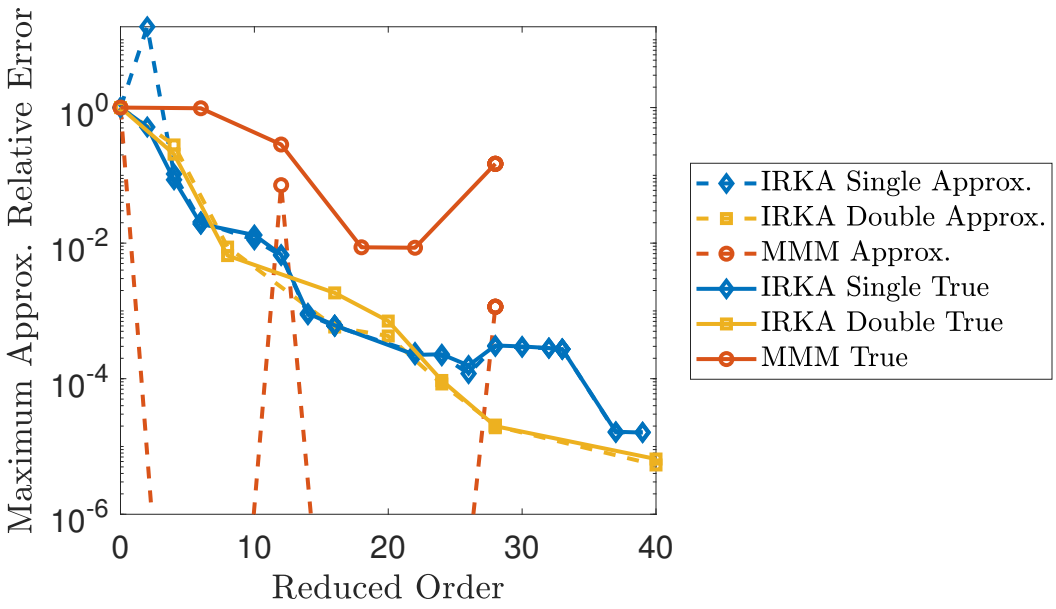


Figure 3.8: True and approximate errors of each model changing with each iteration.

$\mathcal{H}_2$-optimal model reduction scheme in the non-parametric setting.

Recall that in Algorithm 3, we replaced the division of the objective function by the norm of the true system with the norm of the reduced system at each iteration. Therefore, it was also in our interests to see how this function converged (or perhaps did not converge) across our three methods. The results are plotted in Figure 3.9, and show that all three methods do indeed cause the norm of the ROM to converge to the norm of the full model, and at a rather quick rate. MMM performs the worst of these, but still provides an accurate estimate within 1% of the true system norm after just 3 iterations. Once again, both IRKA methods perform similarly, and both stay roughly two orders of magnitude better than the MMM approximation after the first several iterations.

To finish off the analysis of this convection-diffusion flow example, we compared single-sided IRKA using optimal choices for the parameter samples that maximize the error at each iteration to single-sided IRKA using randomly selected parameter samples at each iteration.

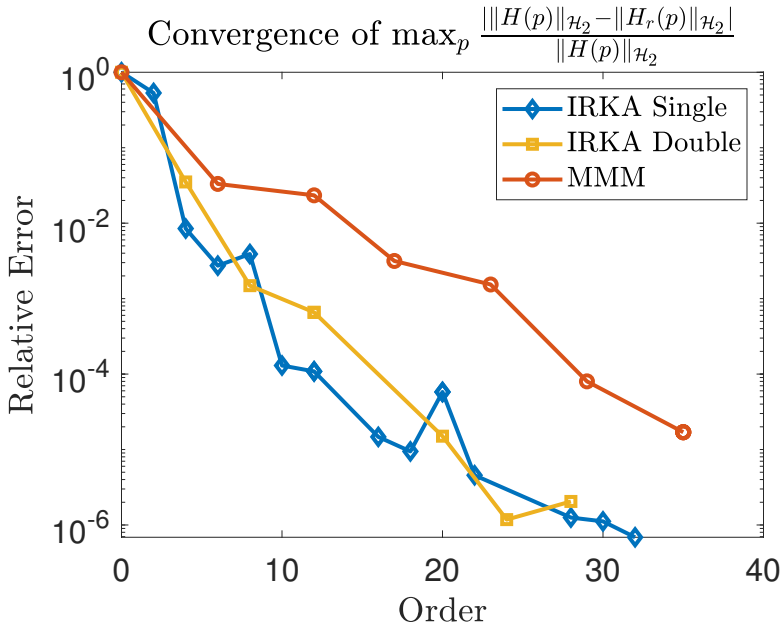
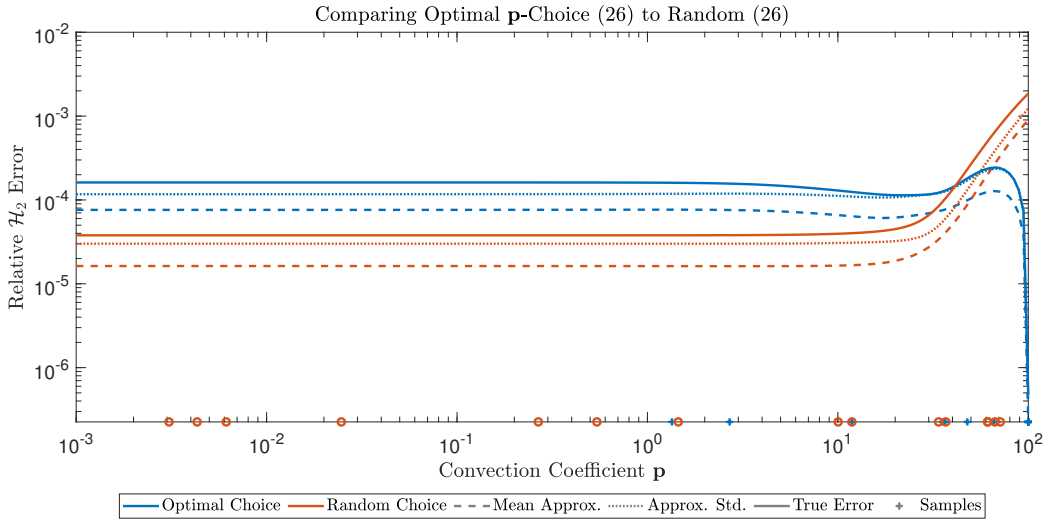


Figure 3.9: Norm of the reduced system compared to the norm of the true system.

We directly compare the two models at a reduced-order of 26 in Figure 3.10. When comparing their errors and error approximations, the random-choice model performs slightly better over the majority of the range, where the model is fairly easy to reduce and where the domain is likely to be sampled from, but otherwise performs worse at 10^2, where the model is most difficult to reduce, despite having sampled very close by.

When we look at how the maximum error changes from iteration to iteration, we see that the optimal-choice model's error decrease nearly monotonically and at a somewhat steady rate. We only sometimes observe these properties with the random-choice model, and we also observe that the random-choice model seems to stall at a relative error of around 10^{-4}. In general, the optimal-choice model is usually one order of magnitude more accurate than the random-choice model after the first few iterations. Since the optimization problem in Algorithm 2 is generally the most expensive part of each iteration, the results here suggest that perhaps sub-optimal parameter selections might provide a good balance of reducing the cost of the optimization while still yielding a very accurate ROM.



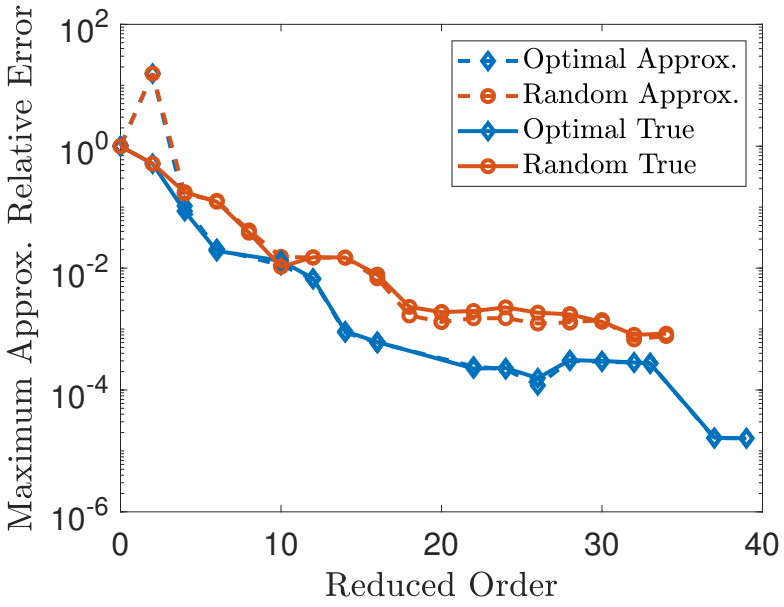


Figure 3.10: Comparing optimal parameter selection to random selection.

3.2.3 Euler-Bernoulli Cantilever Beam Results

For our second example, we use an Euler-Bernoulli cantilever beam model. The system is derived from a finite element discretization of a two-dimensional beam with homogeneous mass and stiffness, and is given by

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{x}}(t,\mathbf{p}) + (a\mathbf{M}+\mathbf{p}\mathbf{K})\dot{\mathbf{x}}(t,\mathbf{p}) + \mathbf{K}\mathbf{x}(t,\mathbf{p}) &= \mathbf{b}u(t),\\ y(t,\mathbf{p}) &= \mathbf{c}^\top\mathbf{x}(t,\mathbf{p}),\end{aligned}$$

where $\mathbf{p}\cdot a \leq 1$. Recalling (2.4), $\mathbf{M}$ represents the masses of nodes in a network, $\mathbf{K}$ represents the stiffness matrix connecting these nodes, and $\mathbf{D} := (a\mathbf{M}+\mathbf{p}\mathbf{K})$ is the damping term. $\mathbf{M}$ and $\mathbf{K}$ are symmetric, as we expect of the mass and stiffness matrices, and $\mathbf{b}$ and $\mathbf{c}$ are taken to have 1 in their first entries, and zeros everywhere else. Although this is a two-parameter model, we fix a at 1 to keep this model in a single parameter. Note that if $\mathbf{p} = a = 0$, there would be no damping in the system, resulting in all of the poles being on the imaginary axis. Such a model would not be stable, so we must be careful to not choose $\mathbf{p}$ too low. We therefore restrict the parameter space to be between 10^{-3} and 1. The transfer function for this system is given by

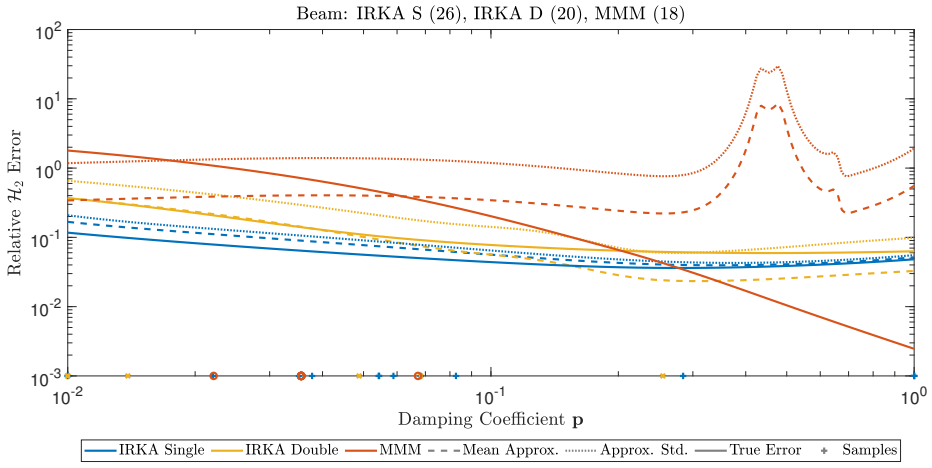
$$H(s,\mathbf{p}) = \mathbf{c}^\top Q(s,\mathbf{p})^{-1}\mathbf{b},$$

where $Q(s,\mathbf{p}) = s^2\mathbf{M} + s(a\mathbf{M}+\mathbf{p}\mathbf{K}) + \mathbf{K}$. In order to apply IRKA to this setting, we first transform the system to a first order system of twice the full-order via a lifting transformation

$$\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}\end{bmatrix}\dot{\mathbf{q}}(t,\mathbf{p}) = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{K} & -(a\mathbf{M}+\mathbf{p}\mathbf{K})\end{bmatrix}\mathbf{q}(t,\mathbf{p}) + \begin{bmatrix}\mathbf{0}\\ \mathbf{b}\end{bmatrix}u(t), \tag{3.6}$$

$$y(t,\mathbf{p}) = \begin{bmatrix}\mathbf{c}^\top & \mathbf{0}\end{bmatrix}\mathbf{q}(t,\mathbf{p}), \tag{3.7}$$

where $\mathbf{q}(t,\mathbf{p}) = \begin{bmatrix} \mathbf{x}(t,\mathbf{p}) \\ \dot{\mathbf{x}}(t,\mathbf{p}) \end{bmatrix}$, and $\mathbf{0}$ and $\mathbf{I}$ are zero and identity matrices of the appropriate sizes, respectively. For this problem, we use a logarithmic frequency sampling range between $\mathrm{i} \cdot 10^{-8}$ to $\mathrm{i} \cdot 10^{8}$, closed under conjugation. The matrices in this problem were generated automatically by a finite element method code that allows for the full-order to be specified. We chose to test with a full-order (after lifting) of $n = 200$.



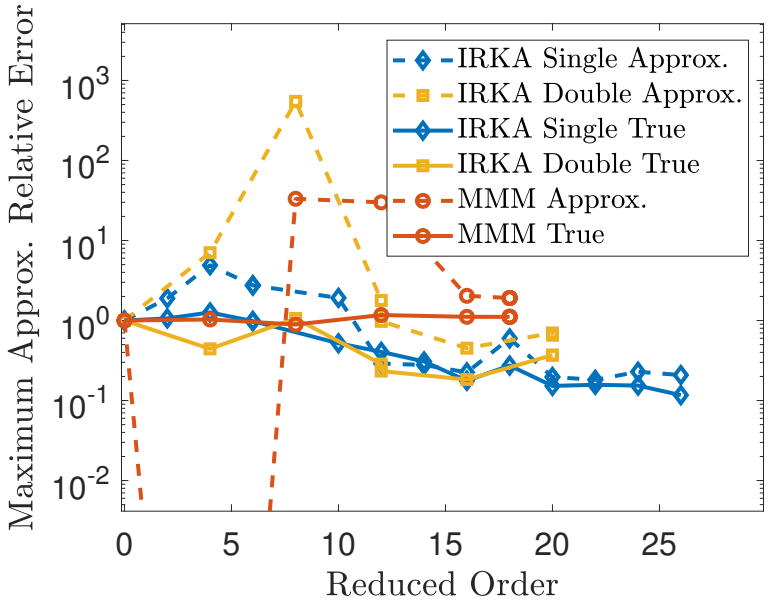


Figure 3.11: Direct comparison of the three models on the cantilever beam example.

We again use single-sided IRKA, double-sided IRKA with $\mathbf{V} \leftarrow [\mathbf{V}, \mathbf{W}]$ for stability, and our implementation of MMM. This beam model proved difficult to reduce, as we see in Figure 3.11. This was primarily due to the special structure of the model's lifted form. All three models were run until the ROMs became unstable. Single-sided IRKA performed the best of the three, having the tightest approximate error bounds and the lowest maximum $\mathcal{H}_2$ error. Double-sided IRKA was marginally worse in this example, but again note that the true error in the double-sided IRKA ROM was always within one standard deviation of the mean of the approximations. Our implementation of MMM was not able to sufficiently reduce the primal system at the low end of the parameter range, and was also not able to resolve a pole that existed in the dual system model around a value of $\mathbf{p} = 0.45$. This pole was close to the imaginary axis in the reduced model, causing the spike in the error approximations, but no such pole existed in the full-order model. Additional sampling around this region was not done by MMM, since this spike in the error is only the maximal error in the $\mathcal{H}_2$-sense.

3.2.4 Thermal Model Results

A thermal model for the heat exchange of a microchip with three variable film coefficients is a parametric first-order model given by the PDE

$$\begin{aligned} \nabla \cdot (\kappa(r)\nabla T(r,t)) + Q(r,t) - \rho(r)C_p(r)\frac{\partial T(r,t)}{\partial t} = 0, \quad r \in \Omega, \\ q(r,t) = h_i(T(r,t) - T_{bulk}), \quad r \in \partial\Omega_i, \end{aligned}$$

where r is the position, T is the unknown temperature distribution, κ is the thermal conductivity, Q is the heat generation rate, ρ is the mass density, C_p is the specific heat capacity, q is the heat flow through a given point, h_i is the film coefficient associated with boundary $\partial\Omega_i$, and T_{bulk} is the bulk temperature in the neighboring phase. The boundary is made up

of a top region, a bottom region, and a side region, and their respective film coefficients are h_t, h_b, and h_s. After discretizing with a finite element method, we obtain the dynamical system

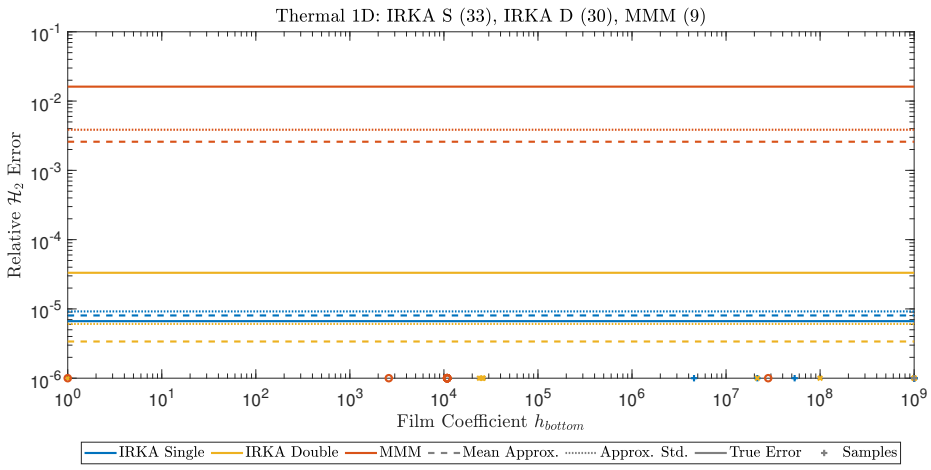
$$\begin{aligned}\mathbf{E}\dot{\mathbf{x}}(t, h_\mathrm{t}, h_\mathrm{b}, h_\mathrm{s}) &= (\mathbf{A} - h_\mathrm{t}\mathbf{A}_\mathrm{t} - h_\mathrm{b}\mathbf{A}_\mathrm{b} - h_\mathrm{s}\mathbf{A}_\mathrm{s})\mathbf{x}(t, h_\mathrm{t}, h_\mathrm{b}, h_\mathrm{s}) + \mathbf{b}u(t),\\ y(t, h_\mathrm{t}, h_\mathrm{b}, h_\mathrm{s}) &= \mathbf{C}^\top \mathbf{x}(t, h_\mathrm{t}, h_\mathrm{b}, h_\mathrm{s}),\end{aligned}$$

where $\mathbf{A}_\mathrm{t}, \mathbf{A}_\mathrm{b}, \mathbf{A}_\mathrm{s}$ are diagonal, $\mathbf{p} := [h_\mathrm{t}, h_\mathrm{b}, h_\mathrm{s}]$, and $h_\mathrm{t}, h_\mathrm{b}, h_\mathrm{s} \in [1, 10^9]$. The full-order of the system is fixed at $n = 4257$, which is rather large for a parametric system. $\mathbf{C}$ originally contains 7 columns, but we only take the first column for the sake of working with SISO systems. The model is part of a benchmark collection found on the Model Order Reduction Wiki[1].

We begin testing with this model by fixing $h_\mathrm{t} = h_\mathrm{s} = 10^9$ in order to work with a one-dimensional parametric model. These particular values were chosen after manually exploring the other parameter boundaries and discovering that most of the boundaries were difficult to construct parametric ROMs over, regardless of the method used. This particular side, however, proved to not be too difficult to reduce.

We compare single-sided IRKA, double-sided IRKA (using Galerkin projection with $\mathbf{V} \leftarrow [\mathbf{V}, \mathbf{W}]$), and our implementation of MMM over this one-dimensional parameter space. The results are shown in Figure 3.12. All three methods used were able to construct ROMs that reduced the error almost perfectly uniformly over the region. Single-sided IRKA ended with an order of 33, double-sided IRKA an order of 30, and MMM an order of 9. In this example, double-sided IRKA performed one order of magnitude worse than single-sided IRKA and also had poor error approximations, but both were able to reduce the error to

[1] https://morwiki.mpi-magdeburg.mpg.de/morwiki/index.php/Thermal_Model



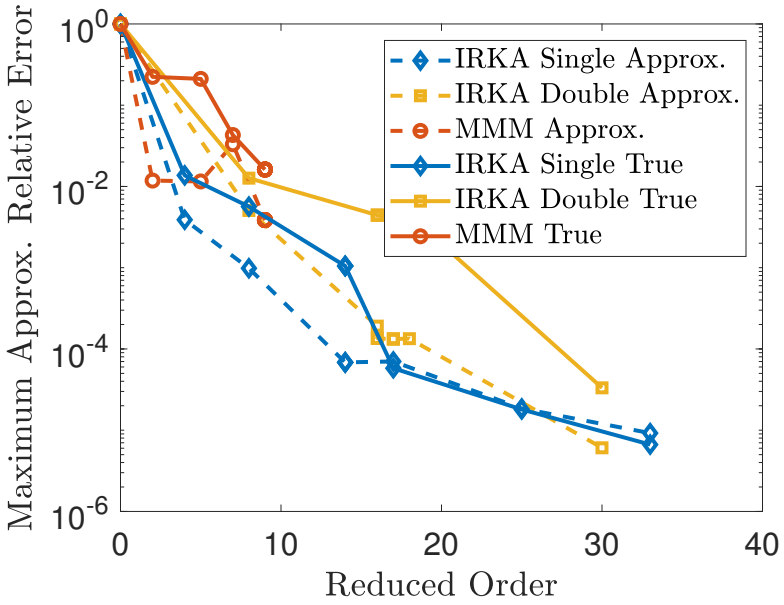


Figure 3.12: Direct comparison of the models on the thermal example.

any desired tolerance. Our implementation of MMM was unable to perform more than five iterations before attempting to sample the same maximum error location twice. Since in our implementation of MMM we chose to match only up to two moments, MMM was not able to increase the accuracy of the local ROM, which one can probably prevent by allowing higher order moment matching, i.e., by choosing $q \geq 2$ in MMM.

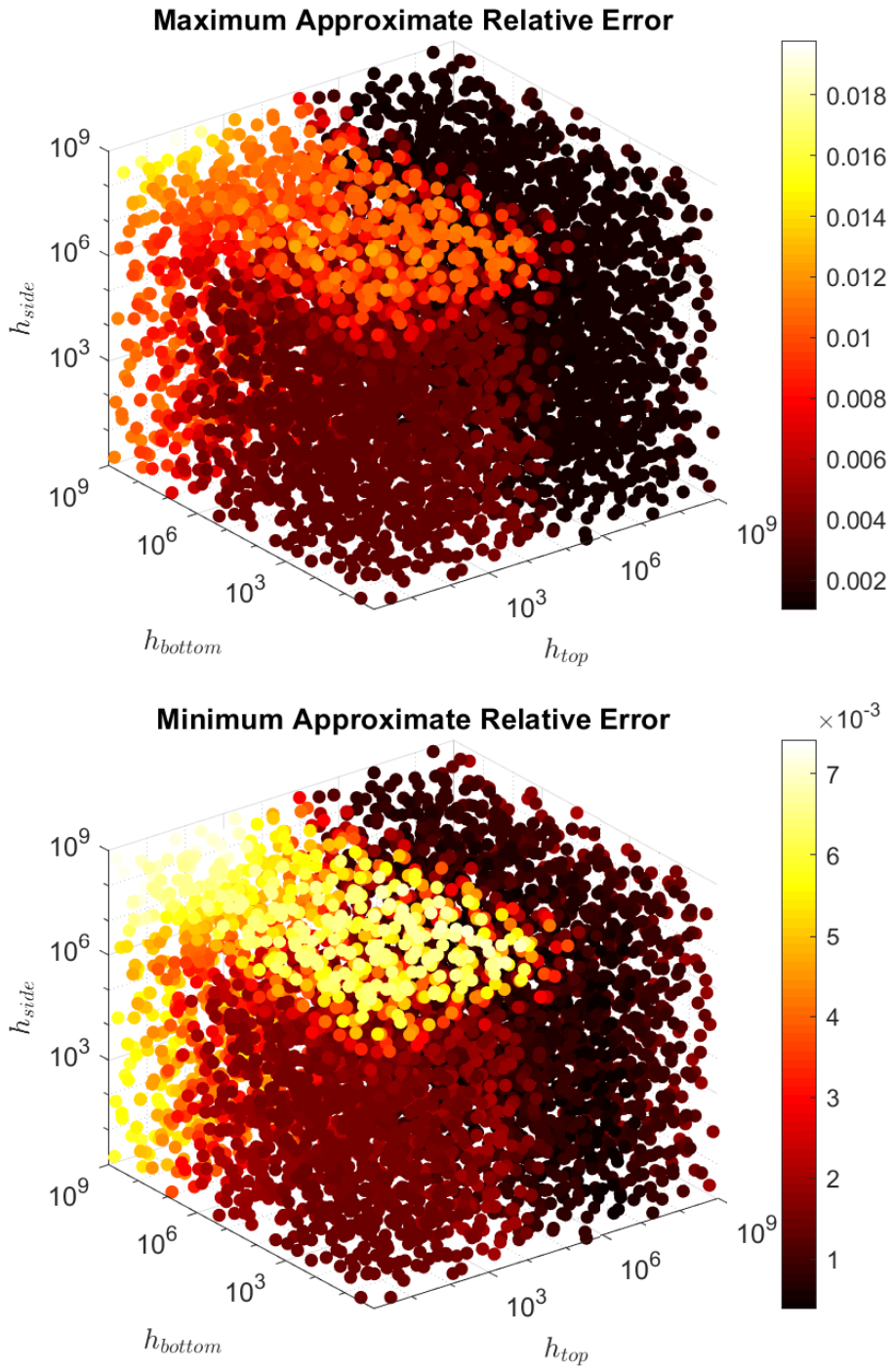


Figure 3.13: 3D ROM of the thermal model.

Although the model reduction methods had problems on the boundaries, we were able to efficiently compute a ROM for the interior of the parameter space without much trouble. In Figure 3.13, we expand our algorithm to operate in three parameter dimensions, construct a ROM using single-sided IRKA, and plot the minimum and maximum relative approximate errors over the entire parameter domain. The order of the ROM constructed was 28. Note that we see larger errors along some of the boundaries where the model was harder to reduce, however none of the points sampled in creating the figure are sufficiently close enough to the boundary to see what the largest upper bound on the error was. We were able to validate some of the error estimates on the interior and the boundaries, and found that the error estimates were accurate, with a maximum error over the whole domain being around 2% relative error along the side $h_{\mathrm{s}} = 10^9$.

3.3 Extension to MIMO Systems

Recall that a MIMO system has input-to-state and state-to-output matrices with multiple columns, and that the $\mathcal{H}_2$ norm for a MIMO system (2.7) includes computing the Frobenius norm of the transfer function along the imaginary axis. Looking through the various upper and lower bounds (2.1), we see that they can be applied in the MIMO case with very few modifications. Assuming we have m inputs and ℓ outputs to a MIMO system, we see that $\hat{\mathbf{x}}_{\mathrm{pr}}, \mathbf{r}_{\mathrm{pr}}, \hat{\mathbf{x}}_{\mathrm{r}_{\mathrm{pr}}}, \mathbf{r}_{\mathrm{r}_{\mathrm{pr}}} \in \mathbb{R}^{n\times m}$, and $\hat{\mathbf{x}}_{\mathrm{du}}, \mathbf{r}_{\mathrm{du}}, \hat{\mathbf{x}}_{\mathrm{r}_{\mathrm{du}}} \in \mathbb{R}^{n\times \ell}$, so all of the Δ measurements in Table 2.1 have dimensions $\ell \times m$, where the i,jth element of these matrices is the error approximation of the SISO system formed by only using column i of $\mathbf{C}$ and column j of $\mathbf{B}$. Thus, the only change we need to make is to compute the Frobenius norm of the approximations.

We use single-sided IRKA for MIMO, as developed in [37, 39], together with these MIMO $\mathcal{H}_2$ estimates to reduce the convection-diffusion flow model (Section 3.1.1) with a full order

of 400. The SISO **b** and **c** vectors were 1 in the first element and zeros elsewhere. In the MIMO setting, we add a second column of all ones to both vectors. The results are shown in Figure 3.14. We are able to reduce the MIMO system to a reduced-order of 60, and we can even clearly see the locations where the local ROMs were constructed. We abbreviate the dimensions of the system by $2 \times 60 \times 2$ to represent the two input dimensions, the reduced order of 60, and the two output dimensions. It is currently unknown why the approximate relative $\mathcal{H}_2$ error overestimates the true error by one to two orders of magnitude, and is worth investigating in the future.

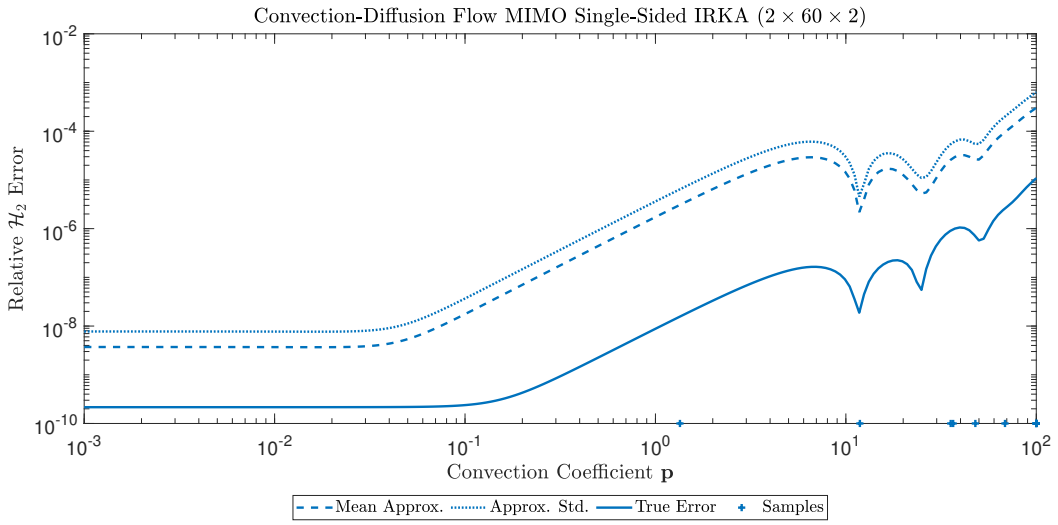


Figure 3.14: MIMO convection diffusion flow final error plot.

Chapter 4

Conclusions and Future Work

In this thesis we reviewed the basics of dynamical systems and model reduction, system errors, and efficient estimates for the $\mathcal{H}_\infty$ error between a full-order model and a subsequent reduced-order model. We have used this basis to develop comparatively efficient estimates to the $\mathcal{H}_2$ error between a full-order model and a subsequent reduced-order model. We showed that these estimates are useful in both the non-parametric setting and the parametric setting. We developed a greedy algorithm (Algorithm 3) for modeling parametric systems that uses IRKA and our efficient error estimates. We tested this algorithm on a suite of parametric models including fluid flow models, mechanical models, and thermal models. We also showed that in the $\mathcal{H}_2$-sense, our algorithm outperformed MMM, an algorithm for parametric model reduction that uses efficient $\mathcal{H}_\infty$ error estimates, usually by two orders of magnitude. We then concluded by investigating Bayesian optimization as a more efficient means of solving the underlying optimization problem, and expanded the algorithm to the MIMO case.

In the future, there are several areas that should be investigated further. First, the effect that the choice of the quadrature scheme has on the ability to approximate the quantity $\left(\frac{1}{2\pi}\int_{-\infty}^{\infty} \|H(\mathrm{i}\omega)\|_{\mathrm{F}}^2 \,\mathrm{d}\omega\right)^{1/2}$, both for generic systems and for when we want to approximate the $\mathcal{H}_2$ error of a reduced-order model.

Second, analysis should be done on the effectiveness of optimal choices of parameter samples. In our experiments, we noted that a random selection of samples worked surprisingly well, but did not have the nearly regular monotonic descent in the maximum $\mathcal{H}_2$ error that we

observed when using optimal choices for the parameter samples.

Third, a more rigorous computational comparison should be made between different optimization schemes within the greedy algorithm for parametric model reduction (Algorithm 2). The preliminary results given in this thesis suggest that Bayesian optimization works very well in this setting, however there are many other algorithms that are worth testing against the efficiency of Bayesian optimization.

Fourth, our algorithm (Algorithm 3) can be extended further to cover more types of systems. Although we heavily use IRKA in our implementation for its $\mathcal{H}_2$-optimality, this method can easily be swapped out for any other type of model reduction scheme, easily allowing extensions to other types of systems not discussed in this thesis, such as bilinear systems, quadratic systems, and general nonlinear systems.

Bibliography

[1] Athanasios C Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[2] Athanasios C Antoulas, Christopher A Beattie, and Serkan Gugercin. Interpolatory model reduction of large-scale dynamical systems. In *Efficient modeling and control of large-scale systems*, pages 3–58. Springer, 2010.

[3] Athanasios C Antoulas, Sanda Lefteriu, A Cosmin Ionita, P Benner, and A Cohen. A tutorial introduction to the loewner framework for model reduction. *Model Reduction and Approximation: Theory and Algorithms*, 15:335, 2017.

[4] Athanasios Constantinos Antoulas, Christopher Andrew Beattie, and Serkan Güğercin. *Interpolatory Methods for Model Reduction*. SIAM, 2020.

[5] Jeanne A Atwell. *Proper orthogonal decomposition for reduced order control of partial differential equations*. PhD thesis, Virginia Tech, 2000.

[6] Jeanne A Atwell and Belinda B King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and computer modelling*, 33 (1-3):1–19, 2001.

[7] Zhaojun Bai and Daniel Skoogh. A projection method for model reduction of bilinear dynamical systems. *Linear algebra and its applications*, 415(2-3):406–425, 2006.

[8] U. Baur, C. Beattie, and P. Benner. Mapping parameters across system boundaries: parameterized model reduction with low rank variability in dynamics. *PAMM*, 14(1): 19–22, 2014.

[9] Ulrike Baur and Peter Benner. Model reduction for parametric systems using balanced truncation and interpolation. *at-Automatisierungstechnik*, 57(8):411–419, 2009.

[10] 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.

[11] Ulrike Baur, Christopher Beattie, and Peter Benner. Mapping parameters across system boundaries: parameterized model reduction with low rank variability in dynamics. *PAMM*, 14(1):19–22, 2014.

[12] Ulrike Baur, Peter Benner, and Lihong Feng. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Archives of Computational Methods in Engineering*, 21(4):331–358, 2014.

[13] Ulrike Baur, Peter Benner, Bernard Haasdonk, Christian Himpe, Immanuel Martini, and Mario Ohlberger. Comparison of methods for parametric model order reduction of time-dependent problems. *Model Reduction and Approximation: Theory and Algorithms*, 15:377, 2017.

[14] Christopher Beattie and Serkan Gugercin. Interpolatory projection methods for structure-preserving model reduction. *Systems & Control Letters*, 58(3):225–232, 2009.

[15] Christopher Beattie, Serkan Gugercin, and Zoran Tomljanović. Sampling-free model reduction of systems with low-rank parameterization. *Advances in Computational Mathematics*, 46(6):1–34, 2020.

[16] Peter Benner and Tobias Breiten. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control systems. *SIAM Journal on Matrix Analysis and Applications*, 33(3):859–885, 2012.

[17] Peter Benner and Tobias Breiten. Krylov-subspace based model reduction of nonlinear circuit models using bilinear and quadratic-linear approximations. In *Progress in Industrial Mathematics at ECMI 2010*, pages 153–159. Springer, 2012.

[18] Peter Benner and Lihong Feng. A robust algorithm for parametric model order reduction based on implicit moment matching. In *Reduced order methods for modeling and computational reduction*, pages 159–185. Springer, 2014.

[19] Peter Benner and Pawan Goyal. Balanced truncation model order reduction for quadratic-bilinear control systems. *arXiv preprint arXiv:1705.00160*, 2017.

[20] Peter Benner, Tobias Breiten, and Tobias Damm. Generalised tangential interpolation for model reduction of discrete-time mimo bilinear systems. *International Journal of Control*, 84(8):1398–1407, 2011.

[21] 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.

[22] Peter Benner, Mario Ohlberger, Albert Cohen, and Karen Willcox. *Model Reduction and Approximation: Theory and Algorithms*. SIAM, 2017.

[23] 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(2):983–1032, 2018.

[24] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[25] Tan Bui-Thanh, Karen Willcox, and Omar Ghattas. Model reduction for large-scale

systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[26] Andrea Carracedo Rodriguez. *Approximation of Parametric Dynamical Systems*. PhD thesis, Virginia Tech, 2020.

[27] Ruth Curtain and Kirsten Morris. Transfer functions of distributed parameter systems: A tutorial. *Automatica*, 45(5):1101–1116, 2009.

[28] Ruth F Curtain and Hans Zwart. *An Introduction to Infinite-Dimensional Linear Systems Theory*, volume 21. Springer Science & Business Media, 2012.

[29] I Pontes Duff, Pierre Vuillemin, Charles Poussot-Vassal, Cédric Seren, and Corentin Briat. Approximation of stability regions for large-scale time-delay systems using model reduction techniques. In *2015 European Control Conference (ECC)*, pages 356–361. IEEE, 2015.

[30] Lihong Feng. Parameter independent model order reduction. *Mathematics and Computers in Simulation*, 68(3):221–234, 2005.

[31] Lihong Feng and Peter Benner. A new error estimator for reduced-order modeling of linear parametric systems. *IEEE Transactions on Microwave Theory and Techniques*, 67(12):4848–4859, 2019.

[32] Lihong Feng and Peter Benner. On error estimation for reduced-order modeling of linear non-parametric and parametric systems. *arXiv preprint arXiv:2003.14319*, 2020.

[33] R.B. Gramacy. *Surrogates*. CRC Press, 2020.

[34] A. R. Grimm. *Parametric Dynamical Systems: Transient Analysis and Data Driven Modeling*. PhD thesis, Virginia Tech, 2018.

[35] Alexander Rudolf Grimm. *Parametric Dynamical Systems: Transient Analysis and Data Driven Modeling*. PhD thesis, Virginia Tech, 2018.

[36] Martin Gubisch and Stefan Volkwein. Proper orthogonal decomposition for linear-quadratic optimal control. *Model reduction and approximation: theory and algorithms*, 5:66, 2017.

[37] S Gugercin, C Beattie, and AC Antoulas. A rational Krylov iteration for optimal $\mathcal{H}_2$ model reduction. In *Proceedings of MTNS*, 2006.

[38] Serkan Gugercin, Athanasios C Antoulas, and Christopher Beattie. H_2 model reduction for large-scale linear dynamical systems. *SIAM journal on matrix analysis and applications*, 30(2):609–638, 2008.

[39] Serkan Gugercin, Tatjana Stykel, and Sarah Wyatt. Model reduction of descriptor systems by interpolatory projection methods. *SIAM Journal on Scientific Computing*, 35(5):B1010–B1033, 2013.

[40] Bjorn Gustavsen and Adam Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on power delivery*, 14(3):1052–1061, 1999.

[41] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, volume 590. Springer, 2016.

[42] M. Hund, P. Mlinarić, and J. Saak. An $\mathcal{H}_2 \otimes \mathcal{L}_2$-optimal model order reduction approach for parametric linear time-invariant systems. *Proc. Appl. Math. Mech.*, 18(1): e201800084, 2018. doi: 10.1002/pamm.201800084.

[43] Antonio Cosmin Ionita and Athanasios C Antoulas. Data-driven parametrized model

reduction in the Loewner framework. *SIAM Journal on Scientific Computing*, 36(3): A984–A1007, 2014.

[44] Pierre Kerfriden, Pierre Gosselet, Sondipon Adhikari, and Stephane Pierre-Alain Bordas. Bridging proper orthogonal decomposition methods and augmented newton–krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):850–866, 2011.

[45] Gaëtan Kerschen and Jean-Claude Golinval. Physical interpretation of the proper orthogonal modes using the singular value decomposition. *Journal of Sound and vibration*, 249(5):849–865, 2002.

[46] Gaetan Kerschen, Jean-claude Golinval, Alexander F Vakakis, and Lawrence A Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear dynamics*, 41(1-3):147–169, 2005.

[47] Boris Kramer and Karen E Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, 57(6):2297–2307, 2019.

[48] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.

[49] Sanjay Lall, Jerrold E Marsden, and Sonja Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 12(6):519–535, 2002.

[50] Hung V Ly and Hien T Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and computer modelling*, 33(1-3):223–236, 2001.

[51] AJ Mayo and AC2343060 Antoulas. A framework for the solution of the generalized realization problem. *Linear algebra and its applications*, 425(2-3):634–662, 2007.

[52] Lewis Meier and D Luenberger. Approximation of linear constant systems. *IEEE Transactions on Automatic Control*, 12(5):585–588, 1967.

[53] Bruce Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.

[54] C Mullis and RA Roberts. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Transactions on Circuits and Systems*, 23(9):551–562, 1976.

[55] Yuji Nakatsukasa, Olivier Sète, and Lloyd N Trefethen. The AAA algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, 2018.

[56] Stephen Prajna. POD model reduction with stability guarantee. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 5, pages 5254–5258. IEEE, 2003.

[57] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[58] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[59] Wilson J Rugh. *Mathematical Description of Linear Systems*. Number 2. Marcel Dekker, 1975.

[60] Matthias Schonlau. Computer experiments and global optimization. 1997.

[61] Rosa Castañé Selga, Boris Lohmann, and Rudy Eid. Stability preservation in projection-based model order reduction of large scale systems. *European journal of control*, 18(2): 122–132, 2012.

[62] Klajdi Sinani, Serkan Gugercin, and Christopher Beattie. A structure-preserving model reduction algorithm for dynamical systems with nonlinear frequency dependence. *IFAC-PapersOnLine*, 49(9):56–61, 2016.

[63] Liqian Zhang, James Lam, Biao Huang, and Guang-Hong Yang. On gramians and balanced truncation of discrete-time bilinear systems. *International Journal of Control*, 76(4):414–427, 2003.

[64] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and Optimal Control*, volume 40. Prentice hall New Jersey, 1996.

Appendices

Appendix A

Omitted Proofs

A.1 Transfer Function Additivity

Claim: The difference of two transfer functions of the same form is another transfer function of the same form, and the under the same input, the associated output is the difference of the two outputs.

Proof. We will mainly focus on the first-order LTI system for simplicity. Let

$$H_1(s) = \mathbf{C}_1^\top (s\mathbf{E}_1 - \mathbf{A}_1)^{-1}\mathbf{B}_1 + \mathbf{D}_1$$

and

$$H_2(s) = \mathbf{C}_2^\top (s\mathbf{E}_2 - \mathbf{A}_2)^{-1}\mathbf{B}_2 + \mathbf{D}_2$$

be two first-order LTI systems' transfer functions, with the order of the first system n not necessarily equal to the order of the second system m.

$$
\begin{aligned}
H_1(s) - H_2(s) &= \mathbf{C}_1^\top (s\mathbf{E}_1 - \mathbf{A}_1)^{-1}\mathbf{B}_1 + \mathbf{D}_1 - \mathbf{C}_2^\top (s\mathbf{E}_2 - \mathbf{A}_2)^{-1}\mathbf{B}_2 - \mathbf{D}_2 \\
&= \mathbf{C}_1^\top (s\mathbf{E}_1 - \mathbf{A}_1)^{-1}\mathbf{B}_1 - \mathbf{C}_2^\top (s\mathbf{E}_2 - \mathbf{A}_2)^{-1}\mathbf{B}_2 + (\mathbf{D}_1 - \mathbf{D}_2) \\
&= \begin{bmatrix} \mathbf{C}_1 \\ -\mathbf{C}_2 \end{bmatrix}^\top \begin{bmatrix} (s\mathbf{E}_1 - \mathbf{A}_1)^{-1} & 0 \\ 0 & (s\mathbf{E}_2 - \mathbf{A}_2)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} + (\mathbf{D}_1 + \mathbf{D}_2) \\
&= \begin{bmatrix} \mathbf{C}_1 \\ -\mathbf{C}_2 \end{bmatrix}^\top \begin{bmatrix} (s\mathbf{E}_1 - \mathbf{A}_1) & 0 \\ 0 & (s\mathbf{E}_2 - \mathbf{A}_2) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} + (\mathbf{D}_1 + \mathbf{D}_2) \\
&= \begin{bmatrix} \mathbf{C}_1 \\ -\mathbf{C}_2 \end{bmatrix}^\top \left(s \begin{bmatrix} \mathbf{E}_1 & 0 \\ 0 & \mathbf{E}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} + (\mathbf{D}_1 + \mathbf{D}_2) \\
&:= H_3(s).
\end{aligned}
$$

From this formulation, it is clear what $\mathbf{E}_3, \mathbf{A}_3, \mathbf{B}_3, \mathbf{C}_3$, and $\mathbf{D}_3$ should be. Thus, the difference of any two first-order LTI transfer functions, even of different orders, is again a transfer function for a first-order LTI system with order $n + m$.

Following the same steps as above but with $\mathbf{Q}_1 = s\mathbf{E}_1 - \mathbf{A}_1$ and $\mathbf{Q}_2 = s\mathbf{E}_2 - \mathbf{A}_2$, it is also clear than the difference of any two transfer functions of the same form (e.g., two second-order LTI systems) yield a transfer function of the same form, so long as all of the matrices that make up $\mathbf{Q}$ allow for zero blocks in the upper right and lower left.

Now let y_1 be the output associated with H_1 under input u and y_2 be the output associated with H_2 under the same input u. Then

$$
Y_3(s) = H_3(s)U(s) = \left(H_1(s) - H_2(s)\right) U(s) = H_1(s)U(s) - H_2(s)U(s) = Y_1(s) - Y_2(s).
$$

Since frequency domain transforms are linear, we have that $y_3(t) = y_1(t) - y_2(t)$. □

A.2 Mild Robustness of $\mu + \sigma$

For context in the following proofs, assume we have six observed nonnegative data points $\tilde{X}$, but up to two of our six observations are "bad" data points, and are zero when they should be nonzero. Because the data might "naturally" be zero, we cannot simply throw away the zeros. Supposing we know which data points are "bad", we then impute these values with draws from a Gaussian distribution with a mean and standard deviation computed from the "good" data points (or any other distribution with the same mean). The set of "good" data is given by X, and the "corrected" data set is given by $\hat{X}$. In the following proofs, we show that the mean plus standard deviation of the "good" data is an upper bound (in expectation) for the mean of the imputed data. Since we cannot tell the "good" data points from the "bad" data points, these proofs show that if we desire an upper bound on the "corrected" data set's mean, we can use the mean and standard deviation of the "bad" data set without needing to know which, if any, of the data points are "bad".

Case 0: Let $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ be a collection of six fixed nonnegative numbers, with a computed mean μ and a computed standard deviation σ. It is trivial to see that $\mu + \sigma \geq \mu$.

Case 1: Let $X = \{x_1, x_2, x_3, x_4, x_5\}$ be a collection of five fixed nonnegative numbers, with a computed mean μ and a computed standard deviation σ. Let $\tilde{X} = \{x_1, x_2, x_3, x_4, x_5, 0\}$, with corresponding mean $\tilde{\mu}$ and corresponding standard deviation $\tilde{\sigma}$. Let $\hat{x}_6 \sim \mathcal{N}(\mu, \sigma)$, and define $\hat{X} = \{x_1, x_2, x_3, x_4, x_5, \hat{x}_6\}$, with mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$. Then,

$$\tilde{\mu} + \tilde{\sigma} \geq \mathbb{E}[\hat{\mu}].$$

Proof. Because $\mathbb{E}[\hat{\mu}] = \mu$ and $\tilde{\mu} = \frac{5}{6}\mu$, it suffices to show $\tilde{\sigma}^2 \geq \frac{1}{36}\mu^2$.

$$\tilde{\sigma}^2 = \left(\frac{1}{6}\sum_{i=1}^{5}(x_i - \tilde{\mu})^2\right) + \frac{1}{6}\tilde{\mu}^2 \geq \frac{1}{6}\tilde{\mu}^2 = \frac{25}{216}\mu \geq \frac{1}{36}\mu$$

□

Case 2: Let $X = \{x_1, x_2, x_3, x_4\}$ be a collection of four fixed nonnegative numbers, with a computed mean μ and a computed standard deviation σ. Let $\tilde{X} = \{x_1, x_2, x_3, x_4, 0, 0\}$, with corresponding mean $\tilde{\mu}$ and corresponding standard deviation $\tilde{\sigma}$. Let $\hat{x}_5, \hat{x}_6 \sim \mathcal{N}(\mu, \sigma)$, and define $\hat{X} = \{x_1, x_2, x_3, x_4, \hat{x}_5, \hat{x}_6\}$, with mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$. Then,

$$\tilde{\mu} + \tilde{\sigma} \geq \mathbb{E}[\hat{\mu}].$$

Proof. Because $\mathbb{E}[\hat{\mu}] = \mu$ and $\tilde{\mu} = \frac{2}{3}\mu$, it suffices to show $\tilde{\sigma}^2 \geq \frac{1}{9}\mu^2$.

$$\tilde{\sigma}^2 = \left(\frac{1}{6}\sum_{i=1}^{4}(x_i - \tilde{\mu})^2\right) + \frac{2}{6}\tilde{\mu}^2 \geq \frac{1}{3}\tilde{\mu}^2 = \frac{4}{27}\mu \geq \frac{1}{9}\mu$$

□