Parameter Dependent Model Reduction for Complex Fluid Flows

Christopher H. Jarvis

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

Doctor of Philosophy
in
Mathematics

John A. Burns, Chair
Jeffrey T. Borggaard
Eugene M. Cliff
Lizette Zietsman

03 March 2014
Blacksburg, Virginia

Keywords: Model Reduction, Nonlinear Systems, Fluid Flows

Copyright 2014, Christopher H. Jarvis
When applying optimization techniques to complex physical systems, using very large numerical models for the solution of a system of parameter dependent partial differential equations (PDEs) is usually intractable. Surrogate models are used to provide an approximation to the high fidelity models while being computationally cheaper to evaluate. Typically, for time dependent nonlinear problems a reduced order model is built using a basis obtained through proper orthogonal decomposition (POD) and Galerkin projection of the system dynamics. In this thesis we present theoretical and numerical results for parameter dependent model reduction techniques. The methods are motivated by the need for surrogate models specifically designed for nonlinear parameter dependent systems. We focus on methods in which the projection basis also depends on the parameter through extrapolation and interpolation. Numerical examples involving 1D Burgers’ equation, 2D Navier-Stokes equations and 2D Boussinesq equations are presented. For each model problem comparison to traditional POD reduced order models will also be presented.
To my Dad, for always being proud of what my brother, sister and I could do and believing in us even when we weren’t sure.
Acknowledgments

First, I would like to express my sincere appreciation to my advisor and committee chairman Dr. John Burns. His support, patience, encouragement and profound ability to keep focused on real problems is invaluable for graduate students beginning research. Furthermore, he has always treated me as an equal and I am truly blessed to know him as a friend. Thank you to Dr. Jeff Borggaard for his many helpful discussions with me on model reduction and numerical computing in general. Thank you to Dr. Eugene Cliff and Dr. Lizette Zietsman for also serving on my committee. Their discussions and recommendations have helped improve my research and dissertation. Thank you to the entire group of faculty, staff and students of the Interdisciplinary Center of Applied Mathematics (ICAM). ICAM is an incredible environment in which to perform research, have discussions and truly learn.

I also wish to acknowledge my support received through the Department of Defense Science, Mathematics and Research for Transformation (SMART) scholarship and my sponsorship from the United States Air Force. Specifically, thank you to Ken Kennedy, Kent Watson and Bob Kressin for the opportunity and support to go back to school.

I am and will be forever grateful to my family for their love and encouragement in everything. Thank you particularly to my wife, Vanessa. There are no words to express my gratitude for her love and support guiding me all the way. Without her, achieving the dream of a PhD would not have been possible.

During my studies I have been supported by so many family, friends and co-workers not mentioned here. I am deeply indebted to more people than I can count.
## Contents

1 Introduction ......................................................... 1
   1.1 Motivation ..................................................... 1
   1.2 Literature Review ............................................. 2
   1.3 Outline ........................................................ 4
   1.4 Notation ......................................................... 4

2 Proper Orthogonal Decomposition ......................... 6
   2.1 Overview ......................................................... 6
   2.2 POD Basis ......................................................... 6
   2.3 Direct POD ROM ................................................. 10
      2.3.1 Heat Equation ............................................. 11
      2.3.2 Abstract POD Problem .................................... 12
   2.4 Centering Trajectory .......................................... 14

3 POD for Parameter Dependent Systems ................. 18
   3.1 POD Basis Sensitivity ....................................... 19
   3.2 Approximating the Optimal POD Basis .................... 22
      3.2.1 Extrapolation ............................................ 22
      3.2.2 Interpolation ............................................. 23
      3.2.3 Expansion ................................................ 27
## 4 Error Analysis

4.1 Standard POD ROM Error ................................................. 28  
4.2 Fixed Basis POD ROM Error ........................................... 33  
4.3 Extrapolation POD ROM Error .................................... 35  
4.4 Interpolation POD Error .............................................. 37  
  4.4.1 Linear Interpolation ............................................... 37  
  4.4.2 Piecewise Cubic Hermite Interpolation Error .......... 40  

## 5 Numerical Experiments ............................................. 43  
5.1 Burgers’ Equation ...................................................... 47  
  5.1.1 POD ROM .......................................................... 49  
  5.1.2 Fixed Parameter Results ...................................... 52  
  5.1.3 Parameter Dependent Results ................................. 55  
5.2 Navier-Stokes Equations ........................................... 64  
  5.2.1 POD ROM .......................................................... 67  
  5.2.2 Flow around a cylinder ...................................... 68  
5.3 Boussinesq Equations ............................................... 78  
  5.3.1 POD ROM .......................................................... 79  
  5.3.2 Radiator ........................................................... 80  

## 6 Conclusions and Future Research ............................................. 88  
6.1 Review of Results .................................................... 88  
6.2 Conclusions ............................................................. 89  
6.3 Future Work ............................................................. 90  

**Bibliography** .......................................................... 92
## List of Figures

5.1 FEM solution for Burgers’ equation with 1024 elements and $Re = 100$ . . . . . . . 48

5.2 FEM solution of sensitivity equation with respect to $1/Re$ for Burgers’ equa-

5.3 POD ROM Relative Error for various $r$ . . . . . . . . . . . . . . . . . . . . . . . . . 53

5.4 Standard POD Error for various $r$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

5.5 POD ROM times for various $r$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

5.6 Parameter Dependent Error for Fixed POD basis . . . . . . . . . . . . . . . . 57

5.7 Parameter Dependent Error for Extrapolated POD basis . . . . . . . . . . . . 58

5.8 Parameter Dependent Error for Linear Interpolation POD basis . . . . . . . . 59

5.9 Parameter Dependent Error for PCHIP POD basis . . . . . . . . . . . . . . . . 60

5.10 Parameter Dependent Error using single data . . . . . . . . . . . . . . . . . . . 61

5.11 Parameter Dependent Error using multiple data . . . . . . . . . . . . . . . . . 63

5.12 Model Problem Domain and Finite Element Mesh . . . . . . . . . . . . . . . . 69

5.13 Typical Solution Snapshot . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

5.14 POD ROM relative error for various $r$ . . . . . . . . . . . . . . . . . . . . . . . . . 72

5.15 Comparison of mean and first uncentered POD mode . . . . . . . . . . . . . . 73

5.16 POD ROM times for various $r$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74

5.17 Parameter dependent ROM relative error using single data shown for various

$r$ and $Re$ values . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
5.18 Parameter dependent ROM relative error using multiple data shown for various $r$ and $Re$ values ........................................ 77

5.19 Model Problem Domain and Finite Element Mesh ........................................ 81

5.20 Typical Solution Snapshot ................................................................. 82

5.21 POD ROM relative error for various $r$ .................................................... 84

5.22 POD ROM times for various $r$ ............................................................... 85

5.23 Parameter Dependent Error using single data ........................................... 86

5.24 Parameter Dependent Error using multiple data ....................................... 87
Chapter 1

Introduction

1.1 Motivation

During a system design process, either for a completely new system or a modification to an existing system there are generally many technical solutions that meet the requirements. Ultimately, a single design must be selected that meets the cost, schedule and performance requirements of the end user. Each of the potential solutions must be explored and understood within the system tradespace. This analysis of potential designs may require the evaluation of complex models to determine the characteristics of each piece of the overall system design. Furthermore, when subsystems are coupled, the complete system performance may require the iteration of solutions for subcomponent models. To meet a required accuracy, the numerical solutions of the underlying dynamical systems for a single set of parameters can be computationally expensive due to both spatial and temporal discretizations. Thus, when applying optimization techniques to identify the optimal system design, using large numerical models is usually intractable. For this reason, in recent years considerable attention has been devoted to the understanding and design of Reduced Order Modeling (ROM) techniques. The goal of reduced order modeling is to develop a model with a high percentage of accuracy of a full order model while saving computational time and storage.
Informally, given a dynamical system of the form

\[
\dot{y}(t) = Ay(t) + F(t, y(t)) \tag{1.1}
\]

the goal of reduced order modeling is to find an approximate system

\[
\dot{y}_r(t) = A_r y_r(t) + F_r(t, y_r(t)) \tag{1.2}
\]

such that \(y_r(t)\) is close to \(y(t)\) either in some norm or output functional and much less computationally expensive to obtain. Then the approximate system can be used to rapidly provide reliable results over many sample points in the system design tradespace.

1.2 Literature Review

For a thorough introduction to model reduction techniques the reader is referred to Antoulas [1]. Within the model reduction framework, there are two major classes of methods. Singular Value Decomposition (SVD) and Krylov subspace based methods. Krylov subspace methods which include Arnoldi and Lanczos methods as well as Iterative Rational Krylov Algorithm (IRKA) are iterative in nature and are well suited for application to very large dimensional linear problems. While recent research has been extended to bilinear and quadratic-bilinear systems [2, 3, 4], fully nonlinear systems are still out of reach and must be modified or linearized first. Consequently, SVD based methods are the primary techniques used for model reduction of nonlinear systems.

SVD based methods use a dataset to identify an optimal subspace of the full dimensional space. The dynamics are then projected onto the lower dimensional subspace to create reduced order models. The resulting reduced order models have much smaller dimension than the full order model and the state space is the space of coefficients of the reduced basis. SVD based methods can be applied to any type of linear or nonlinear dynamical system.
In some cases though the dimension of the system is reduced, the overall complexity can be increased. In this case, the complexity refers to the computational cost to evaluate e.g. FLOPS. Extensions have been developed to reduce both dimension and complexity of the system [5, 6, 7, 8].

The application of SVD based methods are naturally decoupled into offline and online stages. Finding the appropriate reduced basis allows rapid evaluations of the reduced order models in the “online” stage. This is extremely useful in real time applications or when a high accuracy model is a component of a larger optimization process. In these situations a surrogate model is desired that can be evaluated for many parameter values quickly and accurately.

The computation of the appropriate reduced basis is accomplished by decomposition of a representative dataset. It is assumed that the data results from high fidelity models that are not feasible to run in real or near-real time. Furthermore, in parameter dependent problems a representative dataset may require data from many sample points in the intended parameter space. This expensive “offline” stage can be completed once, and the reduced basis extracted from the data. The literature contains an extensive body of research on reduced basis methods for steady linear parametrically dependent problems [9, 10, 11]. In these cases efficient a posteriori error estimators can be developed that allow for the creation of “certified” reduced order models in which the reduced order modeling error over a parameter range can be bounded. Results have also been published for some steady fluid flow applications [12, 11] and more recently the unsteady fluid flows [13, 14]. In these methods a single reduced basis is found that is valid over the parameter range. In this thesis we turn to another technique that allows the development of a reduced order model where the basis depends on the parameter. It is shown that this allows for smaller reduced order models that can have better performance across the parameter ranges.

Proper Orthogonal Decomposition (POD) is a widely used SVD based model reduction technique that can be applied to reduce the dimension of any dynamical system, including
nonlinear systems. In the case of Navier-Stokes equations, the quadratic nonlinear term yields a fast reduced complexity form and hence, fast reduced order model. This coupled with the ease of implementation of the overall POD method have helped the technique become especially common in fluid applications. Furthermore, it has been demonstrated that the use of sensitivity information can provide reduced order models with better performance over a wider range of parameters than traditional methods alone [15, 16]. The contribution of this thesis is an error analysis for POD reduced order models that use improvements based on sensitivity analysis. In addition, a new approach is presented to estimate the optimal POD basis using POD basis functions and respective sensitivities at nearby values through Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) POD.

1.3 Outline

This thesis is organized as follows. The final section of this chapter defines the notation that will be used throughout this thesis. In Chapter 2, a description of the standard POD technique for a single parameter is given for both the generation of the POD basis and resulting projected reduced order model. The POD method for parameter dependent systems is presented in Chapter 3, which includes POD methods that utilize sensitivity information. In Chapter 4, the error analysis for the standard and sensitivity based POD methods is developed. Chapter 5 contains numerical experiments that demonstrate and validate the methods as well as provide practical insight for the use of the methods. Finally, Chapter 6 summarizes the results found herein and gives concluding remarks and future directions for possible research.

1.4 Notation

Since the goal of this thesis is to develop new theory and analysis for POD, the continuous version is presented. In the continuous case the solution to the full order system can be
considered as a collection of functions representing the solution at the specified time instances \[17\] or as a single function that varies over time. The latter interpretation is taken here. The notation follows the framework used by Singler \[18\].

Let \( X \) be a Hilbert space with inner product \((\cdot, \cdot)_X\). The inner product is assumed to be linear in the first argument and conjugate linear in the second, therefore given \( \alpha, \beta \in \mathbb{C} \) and \( x, y \in X \)

\[
(\alpha x, \beta y)_X = \alpha \bar{\beta} (x, y)_X.
\]

The induced norm is given by \( \|x\|_X^2 = (x, x)_X \). For an interval \( I = [0, T] \), let \( L^2(I; X) \) be the Hilbert space of functions \( f \) such that for \( t \in I, f(t) \in X \). Also, for \( f, g \in L^2(I; X) \), the inner product on \( L^2(I; X) \) be given as

\[
(f, g)_{L^2(I; X)} = \int_I (f(t), g(t))_X dt,
\]

and hence the induced norm of \( L^2(I; X) \) is given by \( \|f\|_{L^2(I; X)}^2 = (f, f)_{L^2(I; X)} \).

For compactness of presentation, subscripts are used to denote partial derivatives where the context is clear, e.g.

\[
\phi_{xy}(t, x) = \frac{\partial^2}{\partial x \partial y} \phi(t, x)
\]

and when indexing is present the derivative subscripts follow a comma, hence

\[
\phi_{i,xy}(t, x) = \frac{\partial^2}{\partial x \partial y} \phi_i(t, x).
\]
Chapter 2

Proper Orthogonal Decomposition

2.1 Overview

Proper Orthogonal Decomposition (POD) also known as Karhunen-Loève Decomposition or Principal Component Analysis, is a tool to analyze data. Indicative of its applicability and performance, it has been independently rediscovered in many areas of research ranging from image processing to Computational Fluid Dynamics (CFD) [19]. Given data, POD extracts an orthonormal basis that is optimal to represent the data in a mean squared error sense. In the context of model reduction for dynamical systems, the POD basis, which is determined by a relevant data set, determines a subspace upon which model dynamics are projected into. In this work the focus is on the method itself, to this end, the approach is formulated in a continuous setting without concern for the details of specific discretization schemes. It is assumed that the spatial and temporal discretization errors will be negligible compared to the error introduced by the model reduction.

2.2 POD Basis

Let $y \in L^2(I;X)$ be a given function. The data can represent the solution to a dynamical system of interest. The POD problem is to find an orthonormal set $\{\phi_i\}_{i=1}^r$ with $\phi_i \in X$ for
all \( i = 1,2,\ldots,r \) that satisfies the constrained optimization problem given by

\[
\min_{\{\phi_i\}_{i=1}^r} \|y - P_r y\|_{L^2(I,X)}^2 = \int_I \|y(t) - P_r y(t)\|_X^2 dt
\]

\( \text{s.t.} \quad (\phi_j, \phi_i)_X = \delta_{ij}, \ 1 \leq i,j \leq r \) \hspace{1cm} (2.1)

where

\[ P_r y(t) = \sum_{i=1}^r (y(t), \phi_i)_X \phi_i. \]

The orthonormal set is a basis for the \( r \)-dimensional space \( X_r \subset X \). The orthonormal basis that minimizes the projection error maximizes the inner product above. To find an optimal solution we define the Lagrangian functional \( L(\phi, \lambda) : X \times \mathbb{R} \rightarrow \mathbb{R} \) as

\[
L(\phi, \lambda) = \int_I (y(t), \phi)_X^2 \ dt + \int_I \lambda (1 - (\phi, \phi)_X) \ dt.
\]

By taking the partial derivative of \( L(\phi, \lambda) \) with respect to \( \phi \) we obtain

\[
\left[ \frac{\partial}{\partial \phi} L(\phi, \lambda) \right] \hat{\phi} = \int_I \phi y(t) y(t) \hat{\phi} - \lambda \phi \hat{\phi} dt
\]

for all \( \hat{\phi} \in X \). By examining the necessary conditions for the existence of the solution of the optimization problem, we obtain the eigenvalue problem

\[
\mathcal{R} \phi_i = \lambda_i \phi_i
\] \hspace{1cm} (2.2)

where \( \mathcal{R} : X \rightarrow X \) is given by

\[
\mathcal{R} v = \int_I (v, y(t))_X y(t) \ dt.
\] \hspace{1cm} (2.3)

The operator \( \mathcal{R} \) is referred to as the POD operator and is compact, self-adjoint and Hilbert-Schmidt \cite{20}. Since \( \mathcal{R} \) is self-adjoint and compact there exists orthonormal eigenvectors with corresponding eigenvalues that decay to zero \cite{21} Theorem VI.17. Using the natural ranking
by the eigenvalues, the solution to the POD problem is given by the first $r$ eigenvectors of the $\mathcal{R}$ operator. In practical applications solving for the eigenvectors and eigenvalues of $\mathcal{R}$ can be difficult. As mentioned, when $X$ is real, the eigenvectors and eigenvalues of $\mathcal{R}$ are real. However, due to numerical error, the computed eigenvectors and eigenvalues may have nonzero imaginary parts. This can be corrected by using a singular value decomposition of an appropriate operator where the singular values are guaranteed to be real. Furthermore, the spatial discretization on $X$ may still lead to solving a very large SVD problem. In these cases the method of snapshots provides an alternate method by first computing a reduced basis for $L^2(I)$, where the discretization will likely provide a much smaller numerical problem, and then by solving for the POD basis in $X$.

Define the operator $\mathcal{Y} : L^2(I) \to X$ by

$$\mathcal{Y}w = \int_I w(t) y(t) \, dt$$

for $y \in L^2(I; X)$ and all $w \in L^2(I)$ with the adjoint operator $\mathcal{Y}^* : X \to L^2(I)$ given by

$$[\mathcal{Y}^* z](t) = (y(t), z)_X$$

for all $z \in X$. It is also known that $\mathcal{Y}$ is a bounded, compact, Hilbert-Schmidt operator \cite{20}. Since $\mathcal{Y}$ is a compact operator we know $\mathcal{Y}$ has a singular value decomposition \cite{21}, i.e. there exists nonnegative singular values $\{\sigma_i\}_{i=1}^\infty$ that can be ordered such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$ and corresponding singular vectors $\{\psi_i\}_{i=1}^\infty \in L^2(I)$ and $\{\varphi_i\}_{i=1}^\infty \in X$ such that

$$\mathcal{Y}\psi_i = \sigma_i \varphi_i, \quad \mathcal{Y}^* \varphi_i = \sigma_i \psi_i.$$
Here we note, since $\mathcal{R} = \mathcal{Y}\mathcal{Y}^*$ the singular vectors $\varphi_i$ of $\mathcal{Y}$ satisfy

$$\mathcal{Y}\mathcal{Y}^*\varphi_i = \mathcal{Y}[\mathcal{Y}^*\varphi_i] = \mathcal{Y}[\sigma_i\psi_i] = \sigma_i^2\varphi_i$$

and are the eigenvectors of the operator $\mathcal{R}$, which satisfy the constrained optimization problem. In addition, the square of the singular values $\sigma_i^2$ of $\mathcal{Y}$ are the eigenvalues of $\mathcal{R}$. Furthermore, the singular vectors $\psi_i$ of $\mathcal{Y}$ satisfy

$$\mathcal{Y}^*\mathcal{Y}\psi_i = \mathcal{Y}^*[\mathcal{Y}\psi_i] = \mathcal{Y}^*[\sigma_i\varphi_i] = \sigma_i^2\psi_i$$

and are the eigenvectors of the operator $\mathcal{K} : L^2(I) \to L^2(I)$ defined by $\mathcal{K} = \mathcal{Y}^*\mathcal{Y}$. Therefore solving the SVD problem for either $\mathcal{Y}$ or $\mathcal{Y}^*$ can yield $\{\varphi_i\}$. The size of eigenvalue problem for $\mathcal{K}$ is determined by the temporal discretization which is typically much smaller than the spatial discretization for problems in 2D or higher. Once the vectors $\{\psi_i\}$ have been found, $\{\varphi_i\}$ can be obtained from the SVD relation as

$$\varphi_i = \frac{1}{\sigma_i} \mathcal{Y}\psi_i. \quad (2.4)$$

We define the POD basis of dimension $r$ to be the spatial functions $\phi_i = \varphi_i$ for $i = 1, \ldots, r$. Note that the POD basis is the result of a decomposition of the solution function, and thus the basis is directly dependent on the solution. The POD basis vectors, which are functions in $X$ are commonly referred to as the POD modes. The square of the singular values, which are also the eigenvalues of the operator $\mathcal{R}$, are referred to as the POD eigenvalues [18]. The POD eigenvalues provide the criterion for the ranking of the POD basis vectors. Relying on
the fact that $\mathcal{Y}$ is Hilbert-Schmidt we know that $\{\sigma_i\}_{i=1}^\infty$ is square summable and hence the sum of the eigenvalues of $\mathcal{R}$ is finite i.e. $\mathcal{R}$ is nuclear. Therefore the choice of $r$ can be made such that the relative percentage of modeled “energy” given by

$$\mathcal{E}(r) = \frac{\sum_{i=1}^r \sigma_i^2}{\sum_{i=1}^\infty \sigma_i^2}$$

is sufficiently close to 1. In the case that computing all the singular values is burdensome, one can use the alternative measure to estimate the energy by picking $r$ such that

$$\hat{\mathcal{E}}(r) = 1 - \frac{\sigma_r^2}{\sigma_1^2}$$

is sufficiently close to 1. This attempts to balance the increased computational cost of a larger dimension reduced system with the additional accuracy provided by each successive vector.

### 2.3 Direct POD ROM

Once a basis has been selected the reduced order model is built by projecting the dynamics of the full order system into the subspace spanned by the reduced basis. This process is called the POD Galerkin technique and is illustrated for the abstract system of the form

$$\dot{y}(t) = Ay(t) + F(t, y(t)) \quad (2.5)$$

where $A : \mathcal{D}(A) \subset X \to X$ is a linear operator and $F : \mathcal{D}(F) \subset \mathbb{R} \times X \to X$ is nonlinear.

For a partial differential equation (PDE) the operator $A$ is usually an (unbounded) differential operator that generates a $C_0$-semigroup and in practical problems the desired solutions are actually weak solutions. For in-depth information on this subject refer to \[22, 23, 24\]. As an example the 1D heat equation is presented.
2.3.1 Heat Equation

Let \( X = [0, 1] \) and \( I = [0, T] \), then the homogeneous 1D heat equation is given by

\[
\begin{align*}
\dot{y}(t, x) &= \mu y_{xx}(t, x) \quad \text{for } x \in (0, 1), \ t \in (0, T] \\
y(t, 0) &= y(t, 1) = 0 \\
y(0, x) &= y_0(x). \\
\end{align*}
\]  

(2.6)

The classical solution to this problem is a function \( y(t, x) \) that satisfies (2.6). Specifically, the solution must be twice differentiable in space, differentiable in time and match the boundary conditions and initial conditions. For many practical applications we wish to relax this notion of a solution to allow for more general problems. For instance, the initial condition may have less than two spatial derivatives or may be discontinuous. Secondly, the numerical solution of PDEs is best suited for Hilbert spaces. To address these issues we first restate the PDE initial value problem as an Abstract Cauchy Problem (ACP)

\[
\begin{align*}
\dot{z}(t) &= \mathcal{A}_0 z(t) \\
z(0) &= z_0
\end{align*}
\]  

(2.7)

where \( z_0 \in Z = L^2(X) \) and \( \mathcal{A}_0 : \mathcal{D}(\mathcal{A}_0) = H^1_0(X) \cap H^2(X) \subset Z \rightarrow Z \) is a linear operator defined by \( \mathcal{A}_0 \phi = \mu \phi_{xx} \) for \( \phi \in \mathcal{D}(\mathcal{A}_0) \). In this framework the classical solution to the IVP is a function \( z : I \rightarrow Z \) such that \( z(t) \in \mathcal{D}(\mathcal{A}_0) \) for all \( t > 0 \).

To consider a larger class of potential solutions we seek a function \( z(t) \) that satisfies the weak form of (2.7) for appropriate test functions \( v \), given by

\[
\begin{align*}
(\dot{z}(t), v)_X &= (\mathcal{A}_0 z(t), v)_X \\
z(0) &= z_0.
\end{align*}
\]  

(2.8)

Using the definition of \( \mathcal{A}_0 \) and integration by parts and the homogeneous boundary
conditions we obtain

\[(\mathbf{A}_0 z(t), v)_X = (\mu z_{xx}(t), v)_X \]

\[= -\mu (z_x(t), v_x)_X.\]

We define \(a(\cdot, \cdot) : H^1_0(X) \times H^1_0(X) \to \mathbb{R}\) as \(a(z(t), v) = -\mu(z_x(t), v_x)_X\). It is well known that \(a(\cdot, \cdot)\) is a sesquilinear form hence by the Riesz representation theorem there exists an operator \(\mathbf{A} : \mathcal{D}(\mathbf{A}) = H^1_0(X) \to (H^1_0(X))' = H^{-1}(X)\) such that \([\mathbf{A}z](v) = a(z(t), v)\).

Consequently, (2.7) can be relaxed to

\[\dot{z}(t) = \mathbf{A}z(t) \]

\[z(0) = z_0\] (2.9)

where \(z_0 \in Z = H^{-1}(X)\).

Throughout this thesis, the dynamical systems are to be interpreted in the weak sense.

### 2.3.2 Abstract POD Problem

Using the POD basis for the \(r\)-dimensional subspace \(X_r = \text{span} \{\phi_1, \phi_2, ..., \phi_r\}\) and given any function \(f \in X\), we can represent the approximation \(f_r \in X_r\) by its Fourier coefficients \(\{\alpha_i\}_{i=1}^r\) with respect to the POD basis \(\{\phi_i\}_{i=1}^r\). Hence, for \(f \in X\) we have the approximation

\[f \approx f_r = \sum_{i=1}^r \alpha_i \phi_i\]

where \(\alpha_i = (f, \phi_i)_X\) for \(i = 1, ..., r\).
Using this we define the approximate abstract problem as

\[ \dot{y}_r(t) = Ay_r(t) + F(t, y_r(t)) \]

\[
\frac{d}{dt} \left( \sum_{i=1}^{r} \alpha_i(t) \phi_i \right) = A \left( \sum_{i=1}^{r} \alpha_i(t) \phi_i \right) + F \left( t, \sum_{i=1}^{r} \alpha_i(t) \phi_i \right)
\]

\[
\begin{bmatrix}
\phi_1 & \phi_2 & \cdots & \phi_r
\end{bmatrix}
\frac{d}{dt} \begin{bmatrix}
\alpha_1(t) \\
\alpha_2(t) \\
\vdots \\
\alpha_r(t)
\end{bmatrix} = \begin{bmatrix}
A\phi_1 & A\phi_2 & \cdots & A\phi_r
\end{bmatrix} \begin{bmatrix}
\alpha_1(t) \\
\alpha_2(t) \\
\vdots \\
\alpha_r(t)
\end{bmatrix} + F \left( t, \sum_{i=1}^{r} \alpha_i(t) \phi_i \right)
\]

Recall that the POD basis vectors are a linear combination of the snapshots. Consequently, the approximate system must be interpreted in the same sense as the original system. Taking the inner product with each \( \phi_i \) for \( i = 1, \ldots, r \) we obtain the \( r \)-dimensional system

\[ \dot{\alpha}(t) = A_r \alpha(t) + F_r(t, \alpha(t)) \]

where

\[
A_r = \begin{bmatrix}
(\phi_1, A\phi_1)_X & \cdots & (\phi_1, A\phi_r)_X \\
\vdots & \ddots & \vdots \\
(\phi_r, A\phi_1)_X & \cdots & (\phi_r, A\phi_r)_X
\end{bmatrix}
\]

and

\[
F_r(t, \alpha(t)) = \begin{bmatrix}
(\phi_1, F(t, \sum_{i=1}^{r} \alpha_i(t) \phi_i))_X \\
(\phi_2, F(t, \sum_{i=1}^{r} \alpha_i(t) \phi_i))_X \\
\vdots \\
(\phi_r, F(t, \sum_{i=1}^{r} \alpha_i(t) \phi_i))_X
\end{bmatrix}
\]
Once the $r$-dimensional model has been solved the full state approximation is given by

$$\hat{y}(t) = \sum_{i=1}^{r} \alpha_i(t) \phi_i.$$  

In the case of nonlinear systems, even though the dimension of the system is reduced, the overall complexity may not be reduced. However for linear, bilinear and even quadratic nonlinearities such as the nonlinearity present in the Navier-Stokes equations, an exact reduced form can be found. For more general nonlinearities there exist methods to address complexity reduction in addition to the dimension reduction $[5, 7, 8]$. These methods will not be addressed in this thesis.

## 2.4 Centering Trajectory

It is common in the SVD based model reduction community to use a centering trajectory. The centering trajectory approach first removes the mean of the data set and the POD basis is computed from the “centered” data set. Within the statistical framework, the operator $\mathcal{R}$ is then interpreted to be the 2nd central moment operator as opposed to the 2nd moment operator. As illustrated in $[25]$, it is easy to see the relationship between the mean of a data set and the first POD mode. To that end, given a function $y(t) \in X$ let $\overline{y} \in X$ be the time average of $y(t)$ on $I$, i.e.

$$\overline{y} = \frac{1}{T} \int_I y(s) \, ds.$$  

Consequently, we can write

$$y(t) = \overline{y} + w(t) \quad (2.10)$$
where \( w(t) = y(t) - \overline{y} \) is the perturbation around the time independent mean. Using the operator \( R \) with eigenvector \( \phi \) we have that

\[
R\phi = \lambda \phi
\]

and substituting (2.10) into the definition of \( R \) in (2.3) we obtain

\[
\begin{align*}
\lambda \phi &= \mathcal{R}\phi \\
&= \int_I (\phi, y(t))_X y(t) \, dt \\
&= \int_I (\phi, (\overline{y} + w(t)))_X (\overline{y} + w(t)) \, dt \\
&= (\phi, \overline{y})_X \overline{y} \int_I \, dt + (\phi, \overline{y})_X \int_I w(t) \, dt \\
&\quad + \overline{y} \int_I (\phi, w(t))_X \, dt + \int_I (\phi, w(t))_X w(t) \, dt.
\end{align*}
\]

Thus, if the perturbations are small compared to the mean, then

\[
\lambda \phi \approx c\overline{y}
\]

i.e. the dominant eigenvector is approximately a scaling of the mean.

Recall the POD method determines the optimal subspace in which the data resulting from the dynamical system is represented. In the application of the reduced order model however, it is the dynamical operators that are projected and integrated. At each time step the coefficients of each POD mode are computed from the projected dynamical system. In the simple case, when the mean of the data set is the first POD mode, the coefficient should be constant. Any numerical errors in computing the constant coefficient can result in substantial errors in the approximate solution of the reduced order model. The centering technique attempts to avoid this problem by first subtracting the mean from the data set. The POD method is applied to the perturbation data and a dynamical system generated to
model the perturbation data only.

To illustrate the centered POD ROM we begin with the dynamical system (2.5) and use the expansion (2.10), to obtain

\[ \dot{w}(t) = A\bar{y} + Aw(t) + F(t, \bar{y} + w(t)). \]  

(2.11)

The POD basis is computed from the centered data \( w(t) \) and we use the approximation

\[ w(t) \approx w_r(t) = \sum_{i=1}^{r} \beta_i(t) \phi_i \]

where \( \beta_i(t) = (w(t), \phi_i)_X \) for \( i = 1, \ldots, r \).

Then, the approximate dynamical system is written as

\[ \dot{w}_r(t) = Aw_r(t) + A\bar{y} + F(t, \bar{y} + w_r(t)) \]

and an \( r \)-dimensional system is given in terms of the coefficients by

\[ \dot{\beta}(t) = A_r \beta(t) + [A\bar{y}]_r + F_r(t, \bar{y}, \beta(t)) \]

where

\[ [A\bar{y}]_r = \begin{bmatrix} (\phi_1, A\bar{y})_X \\ \vdots \\ (\phi_r, A\bar{y})_X \end{bmatrix} \]

and

\[ F_r(t, \bar{y}, \beta(t)) = \begin{bmatrix} (\phi_1, F(t, \bar{y} + \sum_{i=1}^{r} \beta_i(t) \phi_i))_X \\ (\phi_2, F(t, \bar{y} + \sum_{i=1}^{r} \beta_i(t) \phi_i))_X \\ \vdots \\ (\phi_r, F(t, \bar{y} + \sum_{i=1}^{r} \beta_i(t) \phi_i))_X \end{bmatrix} \]

Here, once the centered \( r \)-dimensional model has been solved, the full state approximation
is given by

\[ \hat{y}(t) = \bar{y} + \sum_{i=1}^{r} \beta_i(t) \phi_i. \]
Chapter 3

POD for Parameter Dependent Systems

In many realistic applications we are interested in the design or optimization of a system that will be used in a range of conditions governed by parameters. The parameters influence the dynamical system, the structure of the solution and consequently the appropriate subspace for reduced order modeling. As was seen in Section 2.2, the POD basis is the result of a decomposition of the solution to a dynamical system. Our goal is to develop a reduced order model that provides a reasonably accurate solution for parameter values not used in the ROM generation.

Other methods have been developed to address this issue. The simplest is the Global POD (GPOD) method. In the GPOD method, the POD basis is extracted from a collection of solutions instead of a single solution \cite{26}. If the collection of solutions is a representative sampling of the parameter space, then the GPOD basis should be well suited to represent the unknown solution for any parameter in the parameter space. Thus, the GPOD approach can be used to generate a ROM valid for a range of parameters. Here the basis is still optimal, but only to represent the whole data, not necessarily the solution resulting from a single parameter value. However, the solution subspaces for any two parameters may be
very different. The GPOD basis may contain vectors that are orthogonal to the solution for certain parameter values where the ROM is evaluated. Furthermore, an accurate sampling of the parameter space may require many full order solutions to develop an appropriate representative dataset. Finally, the resulting GPOD ROM may either provide only marginal performance over the entire parameter range for a fixed dimension, or require the dimension of the reduced order model to be large to meet desired error tolerances across the parameter range of interest.

In view of these shortcomings, we focus on methods that treat the basis functions as parameter dependent. This procedure allows for small reduced order models that are “minimal” realizations in the sense that the basis is not orthogonal to the solution at the parameter. The optimal POD basis at any parameter is estimated using several methods to include Taylor series approximation and interpolation methods using data from sample points in the parameter space. Some of the methods require the sensitivity of the POD basis with respect to a parameter. In the next section the derivation is shown following the ideas presented in [15].

3.1 POD Basis Sensitivity

If the given function $y \in L^2(I; X)$ from Section 2.2 also depends on a parameter $q \in Q \subset \mathbb{R}$ then resulting POD basis also depends on the parameter and we are interested in how the basis will change as the parameter changes. To facilitate the analysis we redefine the continuous POD operator as $\mathcal{R}(q) : \mathbb{R} \to \mathcal{L}(X)$ given by

$$[\mathcal{R}(q)]v = \int_I \left[ v, y(t, q) \right]_X y(t, q) \; dt$$

where $\mathcal{L}(X)$ is the space of bounded linear operators on $X$.

If we assume that the data and resulting operators depend smoothly on the parameter
then the derivative of \( R (q) \) at \( \hat{q} \in \mathbb{R} \) in the direction of \( \eta \) is given by

\[
\left( [R'(\hat{q})] \eta \right) v = \eta \left( \int_I \left( v, y_q (t, \hat{q}) \right)_x y (t, \hat{q}) + \left( v, y (t, \hat{q}) \right)_x y_q (t, \hat{q}) \, dt \right).
\]

As is common in the case where the domain is one-dimensional, we abuse notation by writing the gradient in place of the differential in the direction of \( \eta \),

\[
[R_q (q)] = \left. [R'(\hat{q})] \eta \right|_{\eta=1} \in \mathcal{L} (X)
\]

where \([R_q (q)]\) is given by

\[
[R_q (q)] v = \int_I \left( v, y_q (t, q) \right)_x y (t, q) + \left( v, y (t, q) \right)_x y_q (t, q) \, dt,
\]

and the term \( y_q (\cdot, q) \) is the derivative of the data with respect the parameter. This can be found by solving the continuous sensitivity equation that corresponds to the dynamical system \([27]\).

From \([2.2]\) for each \( i = 1, \ldots, r \), \( \phi_i (q) \) solves the eigenvalue problem given by

\[
R (q) \phi_i (q) = \lambda_i (q) \phi_i (q).
\]

Differentiating the eigenvalue problem with respect to the parameter yields

\[
\left( R (q) - \lambda_i (q) \right) \phi_{i,q} (q) = \left( - R_q (q) + \lambda_{i,q} (q) \right) \phi_i (q).
\]

Taking the inner product of both sides with \( \phi_i (q) \) we have

\[
\left( \left( R (q) - \lambda_i (q) \right) \phi_i (q), \phi_{i,q} (q) \right)_X = \left( \phi_i (q), \left( - R_q (q) + \lambda_{i,q} (q) \right) \phi_i (q) \right)_X.
\]

where the self-adjoint property of \( R (q) - \lambda_i (q) \) has been used. On the left hand side, \( \phi_i (q) \)
is the eigenvector that satisfies the eigenvalue problem (3.1), hence (3.3) reduces to

\[ \lambda_{i,q}(q) = \left( \phi_i(q), R_q(q) \phi_i(q) \right)_X. \]

Since \( \phi_i(q) \) is in the null space of \( R(q) - \lambda_i(q) \) for any \( \alpha \in \mathbb{R} \), \( \psi = \alpha \phi_i(q) + \phi_{i,q}(q) \) is a solution to (3.2). To find a suitable solution recall the constraint of the POD problem was that the POD vectors should be orthonormal i.e.

\[ \left( \phi_i(q), \phi_j(q) \right)_X = \delta_{ij}. \]

Differentiating this condition respect to the parameter, for \( j = i \) yields

\[ \left( \phi_{i,q}(q), \phi_i(q) \right)_X + \left( \phi_i(q), \phi_{i,q}(q) \right)_X = 0. \]

Due to the symmetric nature of the inner product, the proper choice for \( \phi_{i,q}(q) \) should be orthogonal to \( \phi_i(q) \). To achieve this, the least squares solution is found and then orthogonalized against \( \phi_i(q) \).

The approach using the method of snapshots is similar, exchanging \( R \) for \( K \) and \( \psi_i(q) \) for \( \phi_i(q) \). After obtaining \( \psi_{i,q}(q) \) the POD basis sensitivities \( \phi_{i,q}(q) \) can be found from the differentiated SVD relationship by solving (suppressing the dependence on \( q \))

\[
\frac{\partial}{\partial q} \phi_i = \frac{\partial}{\partial q} \left[ \frac{1}{\sigma_i} Y \psi_i \right] = -\frac{\sigma_{i,q}}{\sigma_i} Y \psi_i + \frac{1}{\sigma_i} Y q \psi_i + \frac{1}{\sigma_i} Y \psi_{i,q}.
\]

In the case that the centering trajectory has been applied, the sensitivity of the center trajectory can also be found. Since the center trajectory is defined as the mean of the data
due to linearity the sensitivity of the mean $\frac{\partial}{\partial q} \bar{y}(q)$ can be found as

$$\frac{\partial}{\partial q} \bar{y}(q) = \frac{\partial}{\partial q} \left[ \frac{1}{T} \int_I y(t, q) \, dt \right] = \left[ \frac{1}{T} \int_I \frac{\partial}{\partial q} y(t, q) \, dt \right] = \bar{y}_i(q).$$

Just as the POD basis computed from the centered data, the sensitivity of the POD basis is computed from the centered solution sensitivity data.

### 3.2 Approximating the Optimal POD Basis

Several methods can be used to estimate the optimal basis for new parameter from existing bases and respective sensitivities. The methods covered include extrapolation, interpolation of the basis and additional improvements utilizing the sensitivities. When implementing a POD ROM with a centering trajectory, the methods are applied in the same fashion to estimate the optimal center trajectories. Throughout this chapter and the next we assume the solution and the POD basis have as many derivatives with respect to the parameter as necessary.

#### 3.2.1 Extrapolation

Given a POD basis for a single parameter and respective basis vector sensitivities, the optimal POD basis can be estimated using a Taylor series expansion of the basis functions. Thus for a nominal parameter $\tilde{q} \in \mathbb{R}$, for each $i = 1, ..., r$, the optimal POD basis functions $\phi_i(q)$ can be written as

$$\phi_i(q) = \phi_i(\tilde{q}) + \phi_{i,q}(\tilde{q})(q - \tilde{q}) + \frac{1}{2} \phi_{i,qq}(\tilde{q})(q - \tilde{q})^2 + O((q - \tilde{q})^3).$$

When $q$ is in a local neighborhood of $\tilde{q}$ it is assumed the higher order terms can be
neglected giving the first order approximation of the optimal basis as

\[ \hat{\phi}_i(q) = \phi_i(\bar{q}) + \phi_{i,q}(\bar{q})(q - \bar{q}). \]  

(3.4)

### 3.2.2 Interpolation

Given POD bases and respective sensitivities for multiple points in the parameter space, the optimal POD basis can be estimated by interpolating the known optimal bases and respective sensitivities. The interpolation method is reviewed briefly here, for more details see [28].

**Linear Interpolation**

Given distinct points \(x_0, x_1, ..., x_n \in \mathbb{R}\), the Lagrange polynomials \(l_k(x)\) are defined as

\[ l_k(x) = \prod_{j=0, j \neq k}^{n} \frac{x - x_j}{x_k - x_j}, \quad k = 0, 1, ..., n. \]  

(3.5)

From the definition it follows that

\[ l_k(x_m) = \begin{cases} 1 & m = k \\ 0 & m \neq k. \end{cases} \]

Given a function \(f(x)\) and the data points \(x_0, x_1, ..., x_n \in \mathbb{R}\), the polynomial \(p_n(x)\) of degree \(n\) that satisfies \(p_n(x_j) = f(x_j)\) for all \(j = 0, 1, ..., n\) is unique and given by [28 Th74 0em 4.5]

\[ p_n(x) = \sum_{k=0}^{n} f(x_k) l_k(x). \]

The Lagrange interpolation result could be used to approximate the optimal POD basis for parameters over a parameter range. However, a well known problem of polynomial interpolation is Runge’s phenomenon in which the maximum error in approximating a general
function by a polynomial can be unbounded. Specifically, for a general function \( f \) defined on a closed interval \([a, b]\) ⊂ \( \mathbb{R} \) with polynomial interpolant \( p_n \) of order \( n \), using equally spaced points \( x_0, x_1, ..., x_n \in [a, b] \) then it is possible that the error
\[
\| f - p_n \|_\infty \to \infty
\]
as \( n \to \infty \) \[28\]. For this reason we utilize piecewise polynomial approximations. Piecewise polynomial approximations use lower order polynomials defined over smaller intervals. Thus the interpolant is not susceptible to wild fluctuations that arise with high order polynomials defined over a large interval. The piecewise linear Lagrange interpolant of a general function \( f \) has the form
\[
p(x) = \begin{cases}
p_0(x) & x \in [x_0, x_1] \\
p_1(x) & x \in [x_1, x_2] \\
\vdots & \vdots \\
p_{n-1}(x) & x \in [x_{n-1}, x_n]
\end{cases}
\]
where \( p_j(x) \) for each \( j = 0, 1, ..., n - 1 \) is the linear polynomial on \([x_j, x_{j+1}]\) given by
\[
p_j(x) = f(x_j) \frac{(x - x_{j+1})}{(x_j - x_{j+1})} + f(x_{j+1}) \frac{(x - x_j)}{(x_{j+1} - x_j)}.
\]

In the context of POD, the basis can be viewed as an \([n \times r]\) matrix and the Lagrange interpolation is carried out component wise.

Using the POD bases \( \Phi(q) \) at the parameter sample points, \( q_1, ..., q_n \in \mathbb{R} \), the piecewise
linear Lagrange interpolant of the POD basis has the form

$$\Phi(q) = \begin{cases} 
\Phi_0(q) & q \in [q_0, q_1] \\
\Phi_1(q) & q \in [q_1, q_2] \\
& \vdots \\
\Phi_{n-1}(q) & q \in [q_{n-1}, q_n]
\end{cases}$$

where $\Phi_j(q)$ for each $j = 0, 1, ..., n-1$ is the linear polynomial of the matrices on $[q_j, q_{j+1}]$ given by

$$\Phi_j(q) = \Phi(q_j) \frac{(q - q_{j+1})}{(q_j - q_{j+1})} + \Phi(q_{j+1}) \frac{(q - q_j)}{(q_{j+1} - q_j)}.$$  

**Hermite Interpolation**

The concept of the interpolation can be extended to include derivative information at the sample points as well. Given the distinct data points $x_1, x_2, ..., x_n \in \mathbb{R}$ and a function $f(x) \in C^1[x_1, x_n]$ the polynomial $p(x)$ of degree $2n - 1$ that satisfies

$$p(x_j) = f(x_j)$$

$$p'(x_j) = f'(x_j)$$

for all $j = 1, ..., n$ is unique [28]. The Hermite interpolant can be constructed by

$$p(x) = \sum_{k=1}^{n} f(x_k) h_k(x) + \sum_{k=1}^{n} f'(x_k) \tilde{h}_k(x)$$

where

$$h_k(x) = [1 - 2l_k'(x)(x-x_k)] (l_k(x))^2$$

$$\tilde{h}_k(x) = (x-x_k) (l_k(x))^2$$
and \( l_k (x) \) are defined as in (3.5).

The Hermite polynomial that interpolates functions and derivatives between two points \( x_j \) and \( x_{j+1} \) is a cubic polynomial. A Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) interpolates a function and derivatives at a set of points \( x_1, x_2, ..., x_n \in \mathbb{R} \) and is a cubic polynomial on each interval \([x_j, x_{j+1}]\). Using the basis functions

\[
N_{00} (x) = \frac{(x - x_{j+1})^2 (2x - 3x_j + x_{j+1})}{(x_{j+1} - x_j)^3}
\]

\[
N_{01} (x) = \frac{(x - x_j) (x - x_{j+1})^2}{(x_{j+1} - x_j)^2}
\]

\[
N_{10} (x) = -\frac{(x - x_j)^2 (2x + x_j - 3x_{j+1})}{(x_{j+1} - x_j)^3}
\]

\[
N_{11} (x) = \frac{(x - x_j)^2 (x - x_{j+1})}{(x_{j+1} - x_j)^2}
\]

the Hermite interpolant is given by

\[
p_j (x) = f (x_j) N_{00} (x) + f' (x_j) N_{01} (x) + f (x_{j+1}) N_{10} (x) + f' (x_{j+1}) N_{11} (x).
\]

These ideas can be used to build a Hermite interpolating polynomial for a single parameter interval \([q_j, q_{j+1}]\) using the known POD bases and corresponding sensitivities. Then, given a set of parameter sample points, \( q_1, ..., q_n \in \mathbb{R} \) where the POD basis functions \( \phi_i (q_j) \) and respective sensitivities \( \phi_{i,q} (q_j) \) are known for each \( j = 1, ..., n \). We construct the piecewise cubic Hermite interpolating polynomial \( H_i (q) \) of the the basis functions \( \phi_i (q) \) that...
satisfies for all $i = 1, \ldots, r$, $j = 1, \ldots, n$

$$H_i(q) = \begin{cases} H_i(q_j) = \phi_i(q_j) \\ H_{i,q}(q_j) = \phi_{i,q}(q_j) \end{cases}$$ \hspace{1cm} (3.7)

### 3.2.3 Expansion

By construction, the POD basis sensitivity functions are orthogonal to the corresponding basis functions and span different subspaces. In the basis extrapolation method, for a nearby parameter a precise linear combination of the nominal POD basis functions and sensitivities are used to estimate the optimal POD basis using to the Taylor series expansion. In the expanded basis method, the POD basis is increased by inclusion of the sensitivity vectors. With the sensitivity vectors as part of the basis, any linear combination of the nominal POD basis and respective sensitivity vectors can be used to estimate the unknown optimal POD basis for a particular parameter. At the nominal parameter the overall performance increase from doubling the basis may be minimal. Even though the POD basis and sensitivity vectors are orthogonal, the sensitivity vectors may not be “next” set of POD basis functions. In contrast, for off nominal parameters the expanded POD basis performance should be greater than the extrapolated basis since it allows for any linear combination of the basis and sensitivity vectors to estimate the optimal POD basis.

When the centering trajectory is used, the sensitivity of the center is added as an additional POD basis function. Here, the coefficients of the center sensitivity vector can be viewed as a correction factor for the center trajectory based on the parameter value and hence constant in time. In actuality however, the coefficient may not be constant due to numerical errors.
Chapter 4

Error Analysis

In this thesis we focus on model reduction where the projection basis depends on a parameter. The methods outlined in Chapter 3 seek to estimate the optimal POD subspace for a parameter using data already known from other parameters. To aid the comparison of the methods, a common framework is used in the error estimation. Due to optimality, the POD basis generated from a single parameter provides the best subspace for the model reduction [17]. Therefore, the error bounds are derived with respect to the optimal POD error estimate at a given parameter value.

4.1 Standard POD ROM Error

First we present the error given by the standard uncentered POD method which is a well known result.

Theorem 4.1. Given a fixed $q \in \mathbb{R}$ and the POD basis $\{\phi_i(q)\}_{i=1}^r$ that satisfies (2.1) for $y(t,q) \in L^2(I;X)$. The error given by the POD reduced order model is given by

$$E_{\text{opt}}(r) = \left\| \sum_{i>r} \phi_i(q) (y(t,q), \phi_i(q))_X \right\|_{L^2(I;X)}$$

$$= \left( \sum_{i>r} \sigma_i(q)^2 \right)^{1/2}.$$
Proof. Using the POD vectors, which are the left singular vectors of $\mathcal{Y}^*$, as a basis for $X$ we can write

$$y(t, q) = \sum_{i=1}^{\infty} \langle y(t, q), \phi_i(q) \rangle_X \phi_i(q).$$

Then, from the definition of the POD problem, the optimal POD error resulting from the truncation of the basis is given by

$$E_{\text{opt}}(r) = \left\| y(t, q) - P_r(q)y(t, q) \right\|_{L^2(I;X)}$$

$$= \left\| \sum_{i=1}^{\infty} \langle y(t, q), \phi_i(q) \rangle_X \phi_i(q) - \sum_{i=1}^{r} \langle y(t, q), \phi_i(q) \rangle_X \phi_i(q) \right\|_{L^2(I;X)}$$

$$= \left\| \sum_{i>r} \phi_i(q) \langle y(t, q), \phi_i(q) \rangle_X \right\|_{L^2(I;X)}$$

$$= \left\| \sum_{i>r} \phi_i(q) \sigma_i(q) \psi_i(t, q) \right\|_{L^2(I;X)}$$

$$= \left( \int \sum_{i>r} \sum_{j>r} \sigma_i(q) \sigma_j(q) \psi_i(t, q) \psi_j(t, q) \langle \phi_i(q), \phi_j(q) \rangle_X dt \right)^{1/2}$$

$$= \left( \int \sum_{i>r} \sigma_i(q)^2 \psi_i(t, q)^2 dt \right)^{1/2}$$

$$= \left( \sum_{i>r} \sigma_i(q)^2 \right)^{1/2}.$$

This result shows that the error resulting from the POD reduced order model is due to the truncation of the POD basis. Note that when using the centering trajectory, the mean of the data set is used directly and the POD model found for the perturbations about the center, thus we have the following result.

**Corollary 4.2.** Using a centering trajectory the standard POD reduced order model error is
given by

\[
E_{\text{cent, opt}} (r) = \left\| \sum_{i>r} \tilde{\phi}_i (q) \left( w (t, q) , \tilde{\phi}_i (q) \right) \right\|_{L^2(I,X)}
= \left( \sum_{i>r} \tilde{\sigma}_i (q)^2 \right)^{1/2}.
\]

**Proof.** Using the expansion (2.10), and the POD basis of the centered data set we have

\[
E_{\text{cent, opt}} (r) = \left\| y (t, q) - \hat{y} (t, q) \right\|_{L^2(I,X)}
= \left\| \left( \mathcal{Y} (q) + w (t, q) \right) - \left( \mathcal{Y} (q) + P_r (q) w (t, q) \right) \right\|_{L^2(I,X)}
= \left\| w (t, q) - P_r (q) w (t, q) \right\|_{L^2(I,X)}
= \left\| \sum_{i>r} \tilde{\phi}_i (q) \left( w (t, q) , \tilde{\phi}_i (q) \right) \right\|_{L^2(I,X)}
= \left( \sum_{i>r} \tilde{\sigma}_i (q)^2 \right)^{1/2}.
\]

**Remark.** Here the singular values \( \tilde{\sigma}_i \) and vectors \( \tilde{\phi}_i \) are associated with the centered data set which has inherently less energy. Specifically, the singular values are smaller so the error resulting from a centered ROM is consequently lower.

For an arbitrary set of independent functions \( \{ \hat{\phi}_i (q) \}_{i=1}^r \), the rank \( r \) projection into the span of the set is given as

\[
\hat{P}_r (q) w (t, q) = \sum_{i=1}^r \left( w (t, q) , \hat{\phi}_i (q) \right) X \hat{\phi}_i (q),
\]

and the subsequent projection error is given by

\[
\hat{E} (r) = \left\| w (t, q) - \hat{P}_r (q) w (t, q) \right\|_{L^2(I,X)}.
\] (4.1)
Here, the projection basis $\hat{\phi}_i(q)$ is an approximation to the optimal POD basis $\phi_i(q)$. Writing the projection basis in terms of the optimal basis plus the approximation error gives

$$\hat{\phi}_i(q) = \phi_i(q) + \psi_i(q),$$

where $\psi_i(q) = \hat{\phi}_i(q) - \phi_i(q)$. Consequently, the projection can be written as

$$\hat{P}_r(q) w(t,q) = \sum_{i=1}^r (w(t,q), \hat{\phi}_i(q))_X \hat{\phi}_i(q)$$

$$= \sum_{i=1}^r (w(t,q), (\phi_i(q) + \psi_i(q)))_X (\phi_i(q) + \psi_i(q))$$

$$= \sum_{i=1}^r (w(t,q), \phi_i(q))_X \phi_i(q) + \sum_{i=1}^r (w(t,q), \psi_i(q))_X \phi_i(q)$$

$$+ \sum_{i=1}^r (w(t,q), \phi_i(q))_X \psi_i(q) + \sum_{i=1}^r (w(t,q), \psi_i(q))_X \psi_i(q)$$

$$= P_r(q) w(t,q) + \tilde{P}_r(q) w(t,q) ,$$

where

$$\tilde{P}_r(q) w(t,q) = \sum_{i=1}^r \left( (w(t,q), \psi_i(q))_X \phi_i(q) + (w(t,q), \phi_i(q))_X \psi_i(q) \right)$$

$$+ \sum_{i=1}^r (w(t,q), \psi_i(q))_X \psi_i(q) .$$

With this formulation the error can be written as

$$\hat{E}(r) = \left\| w(t,q) - \hat{P}_r(q) w(t,q) \right\|_{L^2(I,X)}$$

$$= \left\| (w(t,q) - P_r(q) w(t,q)) - \tilde{P}_r(q) w(t,q) \right\|_{L^2(I,X)}$$

$$\leq E_{opt}(r) + \left\| \tilde{P}_r(q) w(t,q) \right\|_{L^2(I,X)} .$$

When using a centering trajectory the center must also be approximated. Using $\bar{y}(q)$ as the center, and $\hat{y}(q)$ as the approximate center, the total error given by the approximate
reduced order model is given by

\[
\hat{E}(r) = \left\| y(t, q) - \hat{y}(t, q) \right\|_{L^2(I; X)} \\
= \left\| (\overline{y}(q) + w(t, q)) - (\hat{\overline{y}}(q) + \hat{P}_r(q) w(t, q)) \right\|_{L^2(I; X)} \\
= \left\| (\overline{y}(q) - \hat{\overline{y}}(q)) + (w(t, q) - \hat{P}_r(q) w(t, q)) \right\|_{L^2(I; X)} \\
= \left\| (\overline{y}(q) - \hat{\overline{y}}(q)) + (w(t, q) - P_r(q) w(t, q)) - \hat{P}_r(q) w(t, q) \right\|_{L^2(I; X)} \\
\leq E_{\text{opt}}(r) + \left\| \overline{y}(q) - \hat{\overline{y}}(q) \right\|_{L^2(I; X)} + \left\| \hat{P}_r(q) w(t, q) \right\|_{L^2(I; X)}.
\]

This establishes the following lemma which provides the framework for the error analysis in the remainder of this chapter.

**Lemma 4.3.** Given an approximate POD basis \( \{ \hat{\phi}(q) \}_{i=1}^r \) for a parameter \( q \) the resulting ROM error is bounded by

\[
\hat{E}(r) \leq E_{\text{opt}}(r) + \left\| \overline{y}(q) - \hat{\overline{y}}(q) \right\|_{L^2(I; X)} + \left\| \hat{P}_r(q) w(t, q) \right\|_{L^2(I; X)},
\]

where \( E_{\text{opt}}(r) \) is the optimal POD error from Corollary 4.2.

Remark: From this point forward the POD models are assumed to use a centering trajectory approach where \( y(t, q) \) represents the full data set with temporal mean \( \overline{y}(q) \) and perturbations \( w(t, q) \). For each method, the terms \( \left\| \hat{P}_r(q) w(t, q) \right\|_{L^2(I; X)} \) and \( \left\| \overline{y}(q) - \hat{\overline{y}}(q) \right\|_{L^2(I; X)} \) will be described. In the case that a centering trajectory is not applied, the results still hold when \( \overline{y}(q) \) and \( \hat{\overline{y}}(q) \) are taken to be zero and \( w(t, q) = y(t, q) \). The POD basis is then computed for the full uncentered data and the error is bounded by

\[
\hat{E}(r) \leq E_{\text{opt}}(r) + \left\| \hat{P}_r(q) w(t, q) \right\|_{L^2(I; X)},
\]

where \( E_{\text{opt}}(r) \) is the optimal POD error from Theorem 4.1.
4.2 Fixed Basis POD ROM Error

Using the fixed POD method, the optimal POD basis at a parameter is estimated using the POD basis from a single known solution. The assumption is that the optimal POD basis does not depend on the parameter. For the fixed POD method we have the following results.

**Theorem 4.4.** If for each \( i = 1, \ldots, r \), \( \phi_i(q) \in C^1(Q) \), then the fixed POD ROM error is given by

\[
E_{\text{fixed}}(r, q; \tilde{q}) \leq E_{\text{opt}}(r) + \left\| -\bar{y}_q(\xi)(q - \tilde{q}) \right\|_{L^2(I;X)} + \left\| - (q - \tilde{q}) \sum_{i=1}^{r} \left( \left( w(t, q), \phi_{i,q}(\xi_i) \right)_X \phi_i(q) + \left( w(t, q), \phi_i(q) \right)_X \phi_{i,q}(\xi_i) \right) \right\|_{L^2(I;X)} + \left( q - \tilde{q} \right)^2 \sum_{i=1}^{r} \left( w(t, q), \phi_{i,q}(\xi_i) \right)_X \phi_{i,q}(\xi_i) \right\|_{L^2(I;X)},
\]

for some \( \xi, \xi_i \) between \( q, \tilde{q} \).

**Proof.** Using the Taylor series expansion of the center with respect to a parameter at \( \tilde{q} \), we have

\[
\bar{y}(q) = \bar{y}(\tilde{q}) + \bar{y}_q(\xi)(q - \tilde{q})
\]

for some \( \xi \) between \( q, \tilde{q} \). The fixed basis approximation of the center is

\[
\hat{y}(q; \tilde{q}) = \bar{y}(\tilde{q})
\]

with an error of

\[
\bar{y}(q) - \hat{y}(q; \tilde{q}) = -\bar{y}_q(\xi)(q - \tilde{q}).
\]

Likewise, using the Taylor series expansion of the POD basis with respect to a parameter at
\( \tilde{q} \) is given for each \( i = 1, \ldots, r \) by

\[
\phi_i (q) = \phi_i (\tilde{q}) + \phi_{i,q} (\xi_i) (q - \tilde{q})
\]

for some \( \xi_i \) between \( q, \tilde{q} \). The approximation of the optimal POD basis is given by

\[
\hat{\phi}_i (q; \tilde{q}) = \phi_i (\tilde{q})
\]

and we can write

\[
\hat{\phi}_i (q; \tilde{q}) = \phi_i (q) + \psi_i (q; \tilde{q})
\]

with

\[
\psi_i (q; \tilde{q}) = -\phi_{i,q} (\xi_i) (q - \tilde{q}) .
\]

By Lemma 4.3, the error of the fixed basis POD ROM method is

\[
E_{\text{fixed}} (r; q, \tilde{q}) \leq E_{\text{opt}} (r) + \left\| \mathbf{y} (q) - \hat{\mathbf{y}} (q) \right\|_{L^2(I; X)} + \left\| \tilde{P}_r (q; \tilde{q}) w (t, q) \right\|_{L^2(I; X)}
\]

with

\[
\tilde{P}_r (q; \tilde{q}) w (t, q) = - (q - \tilde{q}) \sum_{i=1}^{r} \left( \left( w (t, q) , \phi_{i,q} (\xi_i) \right)_X \phi_i (q) + \left( w (t, q) , \phi_i (q) \right)_X \phi_{i,q} (\xi_i) \right)
\]

\[
+ (q - \tilde{q})^2 \sum_{i=1}^{r} \left( w (t, q) , \phi_{i,q} (\xi_i) \right)_X \phi_{i,q} (\xi_i) .
\]
Therefore, it follows that

\[ E_{\text{fixed}} (r; q; \tilde{q}) \leq E_{\text{opt}} (r) + \left\| -\gamma_q (\xi) (q - \tilde{q}) \right\|_{L^2(I;X)} + \left\| -(q - \tilde{q}) \sum_{i=1}^{r} \left( \left( w(t, q) , \phi_{i,q} (\xi_i) \right) \right) \phi_i (q) + \left( w(t, q) , \phi_i (q) \right) X \phi_{i,q} (\xi_i) \right\|_{L^2(I;X)} + \left\| (q - \tilde{q})^2 \sum_{i=1}^{r} \left( \left( w(t, q) , \phi_{i,q} (\xi_i) \right) \right) X \phi_{i,q} (\xi_i) \right\|_{L^2(I;X)}, \]

for some \( \xi, \xi_i \) between \( q, \tilde{q} \).

\[ \square \]

### 4.3 Extrapolation POD ROM Error

Using the extrapolation approach from Section 3.2.1, the POD basis at a parameter is estimated from a single known solution and solution sensitivity with respect to the parameter using a first order Taylor series approximation. Thus, we have the following results for the extrapolated POD method.

**Theorem 4.5.** If for each \( i = 1, \ldots, r \), \( \phi_i (q) \in C^2(Q) \), then the extrapolation POD ROM error is given by

\[ E_{\text{extrap}} (r; q; \tilde{q}) \leq E_{\text{opt}} (r) + \left\| -\gamma_q (\xi) (q - \tilde{q})^2 \right\|_{L^2(I;X)} + \left\| -(q - \tilde{q})^2 \sum_{i=1}^{r} \left( \left( w(t, q) , \phi_{i,q,q} (\xi_i) \right) \right) \phi_i (q) + \left( w(t, q) , \phi_i (q) \right) X \phi_{i,q,q} (\xi_i) \right\|_{L^2(I;X)} + \left\| (q - \tilde{q})^4 \sum_{i=1}^{r} \left( \left( w(t, q) , \phi_{i,q,q} (\xi_i) \right) \right) X \phi_{i,q,q} (\xi_i) \right\|_{L^2(I;X)}, \]

for some \( \xi, \xi_i \) between \( q, \tilde{q} \).
Proof. Using the Taylor series expansion of the center with respect to a parameter at \( \tilde{q} \), we have

\[
\gamma(q) = \gamma(\tilde{q}) + \gamma_q(\tilde{q})(q - \tilde{q}) + \frac{1}{2} \gamma_{qq}(\xi)(q - \tilde{q})^2,
\]

for some \( \xi \) between \( q, \tilde{q} \). The extrapolation approximation of the center is

\[
\hat{\gamma}(q; \tilde{q}) = \gamma(\tilde{q}) + \gamma_q(\tilde{q})(q - \tilde{q})
\]

with an error of

\[
\gamma(q) - \hat{\gamma}(q; \tilde{q}) = -\frac{1}{2} \gamma_{qq}(\xi)(q - \tilde{q})^2.
\]

Likewise, using the Taylor series expansion of the POD basis with respect to a parameter at \( \tilde{q} \), is given for each \( i = 1, ..., r \) by

\[
\phi_i(q) = \phi_i(\tilde{q}) + \phi_{i,q}(\tilde{q})(q - \tilde{q}) + \frac{1}{2} \phi_{i,qq}(\xi_i)(q - \tilde{q})^2,
\]

for some \( \xi_i \) between \( q, \tilde{q} \). The approximation of the optimal POD basis is given by

\[
\hat{\phi}_i(q; \tilde{q}) = \phi_i(\tilde{q}) + \phi_{i,q}(\tilde{q})(q - \tilde{q})
\]

and we can write

\[
\hat{\phi}_i(q; \tilde{q}) = \phi_i(q) + \psi_i(q; \tilde{q})
\]

with

\[
\psi_i(q; \tilde{q}) = -\frac{1}{2} \phi_{i,qq}(\xi_i)(q - \tilde{q})^2.
\]

By Lemma 4.3, the error of the Extrapolation POD ROM method is

\[
E_{\text{extrap}}(r, q; \tilde{q}) \leq E_{\text{opt}}(r) + \| \gamma(q) - \hat{\gamma}(q) \|_{L^2(I;X)} + \| \hat{P}_r(q; \tilde{q}) w(t, q) \|_{L^2(I;X)}
\]
with
\[
\tilde{P}_r(q; \tilde{q}) w(t, q) = -\frac{(q - \tilde{q})^2}{2} \sum_{i=1}^{r} \left( (w(t, q), \phi_{i,qq}(\xi_i))_X \phi_i(q) \right) + \left( w(t, q), \phi_i(q) \right)_X \phi_{i,qq}(\xi_i)
\]
\[+ \frac{(q - \tilde{q})^4}{4} \sum_{i=1}^{r} \left( w(t, q), \phi_{i,qq}(\xi_i) \right)_X \phi_{i,qq}(\xi_i).\]

Therefore, it follows that
\[
E_{\text{extrap}}(r, q; \tilde{q}) \leq E_{\text{opt}}(r) + \left\| \frac{1}{2} g_{qq}(\xi) (q - \tilde{q})^2 \right\|_{L^2(I; X)}
\]
\[+ \left\| -\frac{(q - \tilde{q})^2}{2} \sum_{i=1}^{r} \left( (w(t, q), \phi_{i,qq}(\xi_i))_X \phi_i(q) \right) + \left( w(t, q), \phi_i(q) \right)_X \phi_{i,qq}(\xi_i) \right\|_{L^2(I; X)}
\]
\[+ \frac{(q - \tilde{q})^4}{4} \sum_{i=1}^{r} \left( w(t, q), \phi_{i,qq}(\xi_i) \right)_X \phi_{i,qq}(\xi_i) \right\|_{L^2(I; X)},
\]

for some $\xi, \xi_i$ between $q, \tilde{q}$.

\[
\square
\]

4.4 Interpolation POD Error

While interpolation of POD basis has been used before in the literature e.g. [29, 30], to the authors knowledge error estimates for the parameter dependent POD bases have not been proven.

4.4.1 Linear Interpolation

Beginning with the linear interpolation from Section 3.2.2, the POD basis at a parameter can be estimated using a piecewise linear polynomial. We make use of the following well known result for the Lagrange polynomial interpolant $p(x)$ [28, Theorem 4.6].
Theorem 4.6. If \( f \in C^{n+1}[a,b] \), then
\[
f(x) - p(x) = \frac{\prod_{k=0}^{n} (x - x_k)}{(n+1)!} f^{(n+1)}(\xi) \text{ for some } \xi = \xi(x) \in [a,b].
\]

In the case of piecewise linear polynomials, \( n = 1 \) and on each interval \([x_i, x_{i+1}]\) the following holds.

Corollary 4.7. If \( f \in C^2[x_i, x_{i+1}] \), then on the interval \([x_i, x_{i+1}]\) there is some \( \xi_i \in (x_i, x_{i+1}) \) such that
\[
f(x) - p(x) = \frac{(x - x_i)(x - x_{i+1})}{2} f^{(2)}(\xi_i).
\]

In the context of the POD basis with parameters \( q_1, q_2, ..., q_n \), on each approximation interval for \( q \in [q_k, q_{k+1}] \) the linear interpolated POD basis vectors can be written as
\[
\hat{\phi}_i(q; q_k, q_{k+1}) = \phi_i(q) - \frac{(q - q_k)(q - q_{k+1})}{2}\phi_{i,qq}(\xi_i)
\]
for \( i = 1, ..., r \). This leads to the following results for the linear interpolating polynomial of the POD basis.

Theorem 4.8. If \( q_k, q_{k+1} \in \mathbb{R} \) and \( \phi_i \in C^2([q_k, q_{k+1}]) \) for each \( i = 1, ..., r \), then on \([q_k, q_{k+1}]\) there are some \( \xi, \xi_i \in (q_k, q_{k+1}) \) such that
\[
E_{LI}(r,q; q_k, q_{k+1}) \leq E_{opt}(r) + \left\| \frac{(q - q_k)(q - q_{k+1})}{2} y_{qq}(\xi) \right\|_{L^2(I;X)}
+ \left\| \frac{(q - q_k)(q - q_{k+1})}{2} \sum_{i=1}^{r} \left( (w(t,q), \phi_{i,qq}(\xi_i))_X \phi_i(q) + (w(t,q), \phi_i(q))_X \phi_{i,qq}(\xi_i) \right) \right\|_{L^2(I;X)}
+ \frac{(q - q_k)^2(q - q_{k+1})^2}{4} \sum_{i=1}^{r} \left( (w(t,q), \phi_{i,qq}(\xi_i))_X \phi_{i,qq}(\xi_i) \right) \right\|_{L^2(I;X)}.
\]
Proof. From Corollary 4.7, the approximation error of the center is given as

$$\| \mathcal{G}(q) - \hat{\mathcal{G}}(q; q_k, q_{k+1}) = \frac{(q - q_k)(q - q_{k+1})}{2} \mathcal{G}_{qq}(\xi)$$

for some $\xi \in (q_k, q_{k+1})$. Similarly, for each $i = 1, \ldots, r$ there exists a $\xi_i \in (q_k, q_{k+1})$ such that the optimal basis approximation error is given by

$$\psi_i(q; q_k, q_{k+1}) = \frac{(q - q_k)(q - q_{k+1})}{2} \phi_{i,qq}(\xi_i).$$

Using Lemma 4.3 the error for the Linear Interpolation based POD ROM on the interval $(q_k, q_{k+1})$ is

$$E_{LI}(r, q; q_k, q_{k+1}) \leq E_{opt}(r) + \left\| \mathcal{G}(q) - \hat{\mathcal{G}}(q) \right\|_{L^2(I; X)} + \left\| \tilde{P}_r(q; q_k, q_{k+1}) w(t, q) \right\|_{L^2(I; X)},$$

where

$$\tilde{P}_r(q; q_k, q_{k+1}) w(t, q) = \frac{(q - q_k)(q - q_{k+1})}{2} \sum_{i=1}^{r} \left( (w(t, q), \phi_{i,qq}(\xi_i))_X \phi_i(q) \right.$$

$$+ (w(t, q), \phi_i(q))_X \phi_{i,qq}(\xi_i) \bigg)$$

$$+ \frac{(q - q_k)^2(q - q_{k+1})^2}{4} \sum_{i=1}^{r} (w(t, q), \phi_{i,qq}(\xi_i))_X \phi_{i,qq}(\xi_i).$$

Therefore we have

$$E_{LI}(r, q; q_k, q_{k+1}) \leq E_{opt}(r) + \left\| \frac{(q - q_k)(q - q_{k+1})}{2} \mathcal{G}_{qq}(\xi) \right\|_{L^2(I; X)}$$

$$+ \left\| \frac{(q - q_k)(q - q_{k+1})}{2} \sum_{i=1}^{r} \left( (w(t, q), \phi_{i,qq}(\xi_i))_X \phi_i(q) \right.$$

$$+ (w(t, q), \phi_i(q))_X \phi_{i,qq}(\xi_i) \bigg)$$

$$+ \frac{(q - q_k)^2(q - q_{k+1})^2}{4} \sum_{i=1}^{r} (w(t, q), \phi_{i,qq}(\xi_i))_X \phi_{i,qq}(\xi_i) \right\|_{L^2(I; X)}.$$
for some \( \xi, \xi_i \in (q_k, q_{k+1}) \).

The piecewise linear interpolation POD error for the full parameter range can now be bounded by taking the maximum over all the intervals.

**Corollary 4.9.** If \( q_1, ..., q_n \in \mathbb{R} \) and \( \phi_i \in C^2([q_1, q_n]) \) for each \( i = 1, ..., r \), then on \([q_1, q_n]\)

\[
E_{LI}(r, q; q_1, ..., q_n) \leq \max_{k=1, ..., n-1} E_{LI}(r, q; q_k, q_{k+1}).
\]

### 4.4.2 Piecewise Cubic Hermite Interpolation Error

Using the Hermite interpolation approach from Section 3.2.2, the POD basis at a parameter can be estimated using a piecewise Hermite cubic polynomial. The following well known result for the general Hermite interpolant \( p(x) \) is used [28, Theorem 4.8].

**Theorem 4.10.** If \( f \in C^{2n}[a, b] \), then

\[
f(x) - p(x) = \frac{(\Psi_n(x))^2}{(2n)!} f^{(2n)}(\xi) \quad \text{for some } \xi = \xi(x) \in [a, b],
\]

where \( \Psi_n(x) = \prod_{k=1}^n (x - x_k) \).

In the case of piecewise cubic Hermite polynomials \( n = 2 \) and on each interval \([x_i, x_{i+1}]\) we have

\[
\Psi_2(x) = (x - x_i)(x - x_{i+1}).
\]

Therefore, on each interval we have the following.

**Corollary 4.11.** If \( f \in C^4[x_i, x_{i+1}] \), then on the interval \([x_i, x_{i+1}]\)

\[
f(x) - p(x) = \frac{(x - x_i)(x - x_{i+1})^2}{24} f^{(4)}(\xi).\]
In the context of the POD basis with parameters $q_1, q_2, ..., q_n$, on each approximation interval for $q \in [q_k, q_{k+1}]$ the PCHIP POD basis vectors can be written as

$$\hat{\phi}_i(q; q_k, q_{k+1}) = \phi_i(q) - \frac{(q - q_k) (q - q_{k+1})^2}{24} \phi_i^{(4)}(\xi_i)$$

for $i = 1, ..., r$. With this corollary we have the following results for the Hermite cubic interpolating polynomial.

**Theorem 4.12.** If $q_k, q_{k+1} \in \mathbb{R}$ and $\phi_i \in C^4([q_k, q_{k+1}])$ for each $i = 1, ..., r$, then on $[q_k, q_{k+1}]$ there is some $\xi, \xi_i \in (q_k, q_{k+1})$ such that

$$E_{\text{PCHIP}}(r, q; q_k, q_{k+1}) \leq E_{\text{opt}}(r) + \left\| \frac{(q - q_k)^2 (q - q_{k+1})^2}{24} y_{qqqq}(\xi) \right\|_{L^2(I;X)}$$

$$+ \left\| \frac{(q - q_k)^2 (q - q_{k+1})^2}{24} \sum_{i=1}^{r} \left( (w(t, q), \phi_i^{qqqq}(\xi_i)) X \phi_i(q) \right) + (w(t, q), \phi_i(q)) X \phi_i^{qqqq}(\xi_i) \right\|_{L^2(I;X)}$$

$$+ \frac{(q - q_k)^4 (q - q_{k+1})^4}{576} \sum_{i=1}^{r} \left( (w(t, q), \phi_i^{qqqq}(\xi_i)) X \phi_i^{qqqq}(\xi_i) \right) \right\|_{L^2(I;X)}.$$

**Proof.** From Corollary 4.11 the approximation error of the center is given as

$$y(q) - \hat{y}(q; q_k, q_{k+1}) = \frac{(q - q_k)^2 (q - q_{k+1})^2}{24} y_{qqqq}(\xi)$$

for some $\xi \in (q_k, q_{k+1})$. Likewise, for each $i = 1, ..., r$ there exists a $\xi_i \in (q_k, q_{k+1})$ such that the optimal basis approximation error is given by

$$\psi_i(q; q_k, q_{k+1}) = \frac{(q - q_k)^2 (q - q_{k+1})^2}{24} \phi_i^{qqqq}(\xi_i).$$

Using Lemma 4.3 the error for the cubic Hermite Interpolation based POD ROM on the
interval \((q_k, q_{k+1})\) is

\[
E_{\text{PCHIP}} (r, q; q_k, q_{k+1}) \leq E_{\text{opt}} (r) + \left\| \mathbf{y} (q) - \hat{\mathbf{y}} (q) \right\|_{L^2(I,X)} + \left\| \tilde{P}_r (q; q_k, q_{k+1}) w (t, q) \right\|_{L^2(I,X)},
\]

where

\[
\tilde{P}_r (q; q_k, q_{k+1}) w (t, q) = \frac{(q - q_k)^2 (q - q_{k+1})^2}{24} \sum_{i=1}^r \left[ \left( w (t, q), \phi_{i,qqqq} (\xi_i) \right)_X \phi_i (q) + \left( w (t, q), \phi_i (q) \right)_X \phi_{i,qqqq} (\xi_i) \right] + \frac{(q - q_k)^4 (q - q_{k+1})^4}{576} \sum_{i=1}^r \left( w (t, q), \phi_{i,qqqq} (\xi_i) \right)_X \phi_{i,qqqq} (\xi_i).
\]

It follows that

\[
E_{\text{PCHIP}} (r, q; q_k, q_{k+1}) \leq E_{\text{opt}} (r) + \left\| \frac{(q - q_k)^2 (q - q_{k+1})^2}{24} \mathbf{y}_{qqqq} (\xi) \right\|_{L^2(I,X)} + \left( \frac{(q - q_k)^4 (q - q_{k+1})^4}{576} \sum_{i=1}^r \left( w (t, q), \phi_{i,qqqq} (\xi_i) \right)_X \phi_{i,qqqq} (\xi_i) \right),
\]

for some \(\xi, \xi_i \in (q_k, q_{k+1})\).

The PCHIP POD error for the full parameter range can now be bounded by taking the maximum over all the intervals.

**Corollary 4.13.** If \(q_1, \ldots, q_n \in \mathbb{R}\) and \(\phi_i \in C^4 ([q_1, q_n])\) for each \(i = 1, \ldots, r\), then on \([q_1, q_n]\)

\[
E_{\text{PCHIP}} (r, q; q_1, \ldots, q_n) \leq \max_{k=1, \ldots, n-1} E_{\text{PCHIP}} (r, q; q_k, q_{k+1})
\]

In the next chapter numerical examples will be presented to demonstrate the methods presented so far. Where possible, error bounds will be presented.
Chapter 5

Numerical Experiments

In this chapter we present numerical experiments that demonstrate the application of the POD model reduction techniques for parameter dependent systems. Three parameter dependent systems are used, the 1D Burgers’ equation, 2D Navier-Stokes equation applied to a flow around a cylinder and finally the Boussinesq equations for thermal driven flow. These models are common benchmark problems in fluid dynamics both because their ability to showcase interesting phenomena and their close resemblance to real physical problems.

For each model problem the finite element method is used to discretize the abstract system (2.5) into an ODE system of the form

\[ \dot{y}_N(t, q) = A_N(q)y_N(t, q) + F_N(t, y_N, q). \]  

(5.1)

To obtain the solution sensitivity with respect to the parameter we formally differentiate (2.5) with respect to the parameter to obtain the sensitivity equation. Using \( s(t, q) = y_q(t, q) \) the sensitivity equation is given by

\[ \dot{s}(t, q) = \mathcal{A}(q)s(t, q) + \mathcal{A}_q(q)y(t, q) + \mathcal{F}_y(t, y, q)s(t, q) + \mathcal{F}_q(t, y, q). \]

We note that the sensitivity equation is linear in \( s(t, q) \). Again, using finite elements the
sensitivity ODE system is given as

\[ \dot{s}_N(t, q) = A_N(q) s_N(t, q) + [A_q]_N(q) y_N(t, q) + [F_y]_N(t, y_N, q) s_N(t, q) + [F_q]_N(t, y_N, q). \]  

(5.2)

The ODE systems can be coupled and solved concurrently in the case of a non-iterative explicit method like the Runge-Kutta methods. Or as is the case here, the sensitivity equation is solved after sufficient convergence of the nonlinear iteration scheme for \( y_N(t, q) \). Solving the sensitivity equation in this manner provides an efficient approach with very little additional computational cost. Usually, the matrices required for the sensitivity equation are built as part of a Newton method to solve the nonlinear equation, thus in theory the sensitivity solution requires only a single extra linear solve per time step.

There are methods to obtain the sensitivities of the solution with respect to the parameter other than directly solving the sensitivity equation. The two major methods include finite differencing of nearby solutions and automatic differentiation (AD) methods. For finite differencing of nearby solutions, multiple full order simulations for nearby parameters are used to estimate the sensitivity of the solution with respect to the parameter. This approach has several disadvantages. If the solution and sensitivities are required at many sample points in the parameter space for the ROM basis generation, the computational cost of obtaining the full order solution data may be burdensome. In addition, if the parameters affect the domain, as in the case of with a parameter that determines the location or shape of an object, finite differencing of solutions on different domains may not be meaningful or possible. Furthermore, the choice of the nearby parameters must be made carefully to provide accurate sensitivity measurements while preventing numerical errors from rendering the sensitivity approximations useless.

The other major method that shows considerable promise is the use of automatic differentiation methods. These methods compute the derivatives of complex functions by exploiting the chain rule applied to basic operations \([31]\). Thus the sensitivity can be obtained using
only one full order solution. However, overhead associated with automatic differentiation computations running in addition to the original code may incur a significant computational penalty. In a forward approach the sensitivity solution scheme depends directly on the scheme used in the numerical model for the original solution. When using existing software that cannot be easily modified, AD can provide a non-intrusive method to obtain the solution sensitivity. One code that has been tested in this application is AD_Deriv [32]. The results using AD_Deriv agree very well with the sensitivity equation implementation. Furthermore, coupling the solution and solution sensitivity using AD_Deriv provides a robust method to compute the POD basis and corresponding basis sensitivity vectors concurrently using existing code for computing the POD basis.

In this chapter, the reduced order modeling techniques are tested for each model problem. First, the methods that utilize only a single full order solution are compared. These include a standard fixed basis for a single nominal parameter $\tilde{q}$ and the extrapolation of the fixed basis using the sensitivities. Next, we compare the methods that utilize multiple data sets these include Global POD, linear interpolation of the basis functions and Piecewise Cubic Hermite Interpolating Polynomials (PCHIP). In both cases the errors are compared against the optimal POD error for the tested parameter.

Prior to beginning the specific numerical examples, it is worth discussing some practical considerations associated with generating the POD basis for each ROM. In almost every method used to estimate the optimal POD basis from known bases, the transformations are not guaranteed to preserve orthonormality of the basis. The orthonormality is important since the initial condition for the reduced order model is obtained by projection of the appropriate vector from the full dimensional approximation space. Without the orthogonality, the POD basis does not represent a true projection basis. For this reason, an extra Gram-Schmidt step is performed to re-orthonormalize the new basis prior to implementation in the reduced order model.

Furthermore, care should be exercised when interpolating bases. The interpolation meth-
ods in this thesis interpolate between the vectors of the POD bases by index. Since the basis vectors are ranked according to their POD energy similar vectors arise at different parameter values, yet have different rankings within their respective basis. Additionally, the solution of the SVD is not unique in the sense that multiple vectors associated with the same singular values may come in any order. Moreover, the negative of a singular vector is also a singular vector. To overcome the ambiguity, a congruence transformation is used \[30, 4\]. The congruence transformation problem seeks to find a common coordinate system in which to represent the bases. Given two \([n \times r]\) matrices \(\Phi_1, \Phi_2\) we seek the \([r \times r]\) matrix \(Q\) that solves

\[
\min_Q \|\Phi_2 Q - \Phi_1\| \text{ subject to } Q^T Q = I_r
\]

where \(\Phi_1\) has been used as the reference basis. If the two bases were identical up to a permutation of the columns then \(Q\) should be a permuted identity matrix. When the bases are associated with different parameter values, a vector from one basis may be a linear combination of the vectors the other basis. Thus, the entries of \(Q\) will not be integers in practice. This is the classical orthogonal Procrustes problem whose general solution can be obtained by SVD \[33\]. In the reduced order modeling application, the interpolation methods only seek to find an approximation of the subspace spanned by the optimal POD basis at a new parameter, not the necessarily the optimal POD vectors themselves. The PCHIP POD which is a Hermite interpolant also requires the same congruence transformation be applied to the sensitivities of the basis in addition to the basis vectors. In this way the PCHIP POD can be applied in the same generalized coordinate system as a linear interpolation allowing for a direct comparison of the improvement resulting from the use of the sensitivity information.
5.1 Burgers’ Equation

Consider the unsteady viscous Burgers’ equation in 1D given by

\[ y_t (t, x, q) + y (t, x, q) y_x (t, x, q) - \nu y_{xx} (t, x, q) = 0 \quad x \in (0, 1), \ t > 0 \]  

(5.3)

with homogeneous Dirichlet boundary conditions \( y (t, 0, q) = 0 = y (t, 1, q) \) for \( t > 0 \) and \( q \in Q \) and initial condition given by \( y (0, \cdot, q) = y_0 (\cdot, q) \in L^2 (0, 1) \).

Burgers’ equation is a second order nonlinear PDE containing both convection and diffusion. Lighthill derived Burgers’ equation as a second order approximation to the 1D Navier-Stokes equations [34]. Burgers’ equation is also a prototypical PDE that can exhibit steeping fronts and even discontinuous solutions depending on the chosen parameter value. Thus, Burgers’ equation in 1D is a widely used test problem in both fluid dynamics and in numerical methods for PDEs. Here the parameter \( q = \nu \) represents the viscosity of the fluid and in a normalized setting we have \( \nu = 1/Re \), where \( Re \) plays the role of the non-dimensional Reynolds number. For our consideration the boundary conditions and mesh properties do not depend on the Reynolds number. Thus the sensitivity of the solution with respect the changes in the parameter are bounded and smooth.

Using the finite element method, the PDE is discretized into an ODE system of the form

\[
M \dot{y} (t, q) + C (y (t, q)) y (t, q) + \nu K y (t, q) = 0
\]  

(5.4)

where \( M_{ij} = \langle \phi_j, \phi_i \rangle \) is the mass matrix, \( K_{ij} = \langle \phi_{j,x}, \phi_{i,x} \rangle \) is the stiffness matrix and \( C_{ij} (\xi) = \langle \xi \cdot \phi_{j,x}, \phi_i \rangle \) is the nonlinear (trilinear) convective matrix.

The system of nonlinear ODEs is then solved using a trapezoidal time stepping method with Picard iteration. A typical solution is shown below in Figure 5.1.

To obtain the solution sensitivity with respect to the parameter we differentiate (5.3) with respect to the parameter to obtain the sensitivity equation. Using \( s (t, x, q) = y_q (t, x, q) \) the
sensitivity equation is given by

\[ s_t (t, x, q) + s (t, x, q) y_x (t, x, q) + y (t, x, q) s_x (t, x, q) - y_{xx} (t, x, q) - \nu s_{xx} (t, x, q) = 0 \]

with homogeneous Dirichlet boundary conditions \( s (t, 0, q) = 0 = s (t, 1, q) \) and initial condition given by \( s (0, x, q) = \frac{\partial}{\partial q} y_0 (x, q) \in L^2 (0, 1) \). We note that the sensitivity equation uses the same bilinear and trilinear forms as the weak form of Burgers’ equation [35]. However in this case the PDE is linear in \( s (t, x, q) \). Again using finite elements the sensitivity ODE system is given as

\[ M \dot{s} (t, q) + \left( \tilde{C} (y (t, q)) + C (y (t, q)) \right) s (t, q) + Ky (t, q) + \nu Ks (t, q) = 0. \]

Here, \( \tilde{C}_{ij} (\xi) = \langle \phi_j \cdot \xi_x, \phi_i \rangle \) represents the weak derivative of the convection velocity and is
used in case of Newton iteration \[36\]. The solution to the sensitivity equation corresponding to the Figure 5.1 is shown below in Figure 5.2.

![Figure 5.2: FEM solution of sensitivity equation with respect to \(1/Re\) for Burgers’ equation with 1024 elements and \(Re = 100\)](image)

### 5.1.1 POD ROM

We generate the POD based reduced order model as follows. Starting from the spatially discretized system (5.4), let \(\overline{y}\) be the temporal mean of the snapshots and write

\[
y(t, q) = \overline{y}(q) + w(t, q)
\]

to obtain the centered system

\[
M \dot{w}(t, q) + C(w(t, q)) \overline{y}(q) + C(w(t, q)) w(t, q) + (C(\overline{y}(q)) + \nu K) w(t, q) = -\nu K + C(\overline{y}(q)) \overline{y}(q)
\]
Then let \( \hat{\Phi}_r \) be the basis for the reduced subspace for the model reduction. Substituting in the approximation \( w(t,q) \approx \hat{\Phi}_r w(t,q) \) we obtain the low rank approximation system

\[
M \hat{\Phi} \dot{w}_r(t,q) + C \left( \hat{\Phi}_r w(t,q) \right) \bar{y}(q) + C \left( \hat{\Phi}_r w(t,q) \right) \hat{\Phi}_r w(t,q) \\
+ (C \left( \bar{y}(q) \right) + \nu K) \hat{\Phi}_r w(t,q) = - (\nu K + C \left( \bar{y}(q) \right)) \bar{y}(q).
\]

Here the trilinearity of the nonlinear term has been used and the constant terms involving only the mean have been moved to the right hand side. Multiplying through by \( \hat{\Phi}_r^T \) we obtain the low dimensional model given by

\[
M_r \dot{w}_r(t,q) + C_1 (w_r(t,q)) \bar{y}(q) + C_2 (w_r(t,q)) w_r(t,q) \\
+ (C_3 + \nu K_r) w_r(t,q) = - \hat{\Phi}_r^T \left( \nu K + C \left( \bar{y}(q) \right) \right) \bar{y}(q)
\]

(5.5)

where

\[
M_r = \hat{\Phi}_r^T M \hat{\Phi},
\]

\[
K_r = \hat{\Phi}_r^T K \hat{\Phi},
\]

and

\[
C_3 = \hat{\Phi}_r^T C \left( \bar{y}(q) \right) \hat{\Phi}.
\]

For the other terms, tensor forms can be precomputed offline by taking advantage of the linearity in each argument of the trilinear forms. The tensor terms allow for rapid online assembly of the nonlinear matrices. Recall that

\[
w(t,q) \approx \sum_{i=1}^{r} [w_r(t,q)]_i \hat{\phi}_i
\]
so that

\[
C \left( \hat{\Phi} w_r \left( t, q \right) \right) = \sum_{i=1}^{r} \left[ w_r \left( t, q \right) \right]_i C \left( \hat{\phi}_i \right)
\]

and we can generate an \([r \times 1 \times r]\) matrix given by

\[
\left[ C^r_1 \right]_{ijk} = \hat{\phi}_i^T C \left( \hat{\phi}_k \right) y \left( q \right)
\]

and an \([r \times r \times r]\) matrix given by

\[
\left[ C^r_2 \right]_{ijk} = \hat{\phi}_i^T C \left( \hat{\phi}_k \right) \hat{\phi}_j.
\]

During the online phase, the two terms

\[
C_1 \left( w_r \left( t, q \right) \right) y \left( q \right) = \hat{\Phi}^T C \left( \hat{\Phi} w_r \left( t, q \right) \right) y \left( q \right),
\]

\[
C_2 \left( w_r \left( t, q \right) \right) w_r \left( t, q \right) = \hat{\Phi}^T C \left( \hat{\Phi} w_r \left( t, q \right) \right) \hat{\Phi} w_r \left( t, q \right)
\]

can be quickly assembled as

\[
C_1 \left( w_r \left( t, q \right) \right) y \left( q \right) = \sum_{i=k}^{r} \left[ w_r \left( t, q \right) \right]_k \left[ \tilde{C}^1 \right]_{ijk}
\]

\[
C_2 \left( w_r \left( t, q \right) \right) w_r \left( t, q \right) = \left( \sum_{i=k}^{r} \left[ w_r \left( t, q \right) \right]_k \left[ \tilde{C}^2 \right]_{ijk} \right) w_r \left( t, q \right).
\]

The initial conditions are projected into the space the same way, so that

\[
w_r \left( 0, q \right) = \hat{\Phi}^T \left( y \left( 0, q \right) - \bar{y} \left( q \right) \right).
\]
Thus, the Burgers’ equation POD IVP becomes

\[
M_r \dot{w}_r (t, q) = - \hat{\Phi}^T (\nu K + C (\bar{y} (q))) \bar{y} (q) - C_1 (w_r (t, q)) \bar{y} (q) - C_2 (w_r (t, q)) w_r (t, q) \\
- (C_3 + \nu K_r) w_r (t, q)
\]

\[
w_r (0, q) = \hat{\Phi}^T (y (0, q) - \bar{y} (q)).
\]  

(5.6)

Once the solution to (5.6) is computed, the ROM solution in the original approximation space coordinates can be recovered by

\[
\hat{y}_r (t, q) = \bar{y} (q) + \hat{\Phi} w_r (t, q).
\]

We note here that the online order of complexity according to the operation count of the POD convective terms \(\hat{\Phi}^T C (\hat{\Phi} w_r (t, q)) \hat{\Phi} w_r (t, q)\), actually increases when naively implemented. The specific structure of the nonlinear term can be exploited to produce \(r\)-dimensional tensors. With these reduced expressions, the online complexity is reduced from \(O(N)\) to \(O(r)\). Although the speed up may not be readily apparent in the 1D case, in higher dimensions where \(N\) is significantly larger the cost of assembling matrices grows considerably.

### 5.1.2 Fixed Parameter Results

The performance of the POD reduced order model is tested for the nominal parameter \(\nu = 1/100\). We test the convergence of the POD ROM to the high fidelity model as \(r\) increases. To accomplish this, the full order model is solved using \(N = 1024\) uniform elements in space and integrated over the time interval \((0, 1]\) using the trapezoidal scheme with a time step \(\Delta t = 0.0005\). At each time step the iteration tolerance is \(10^{-10}\) between iterations. The data over the complete simulation is used to generate the POD basis and the reduced order model is integrated using the same method for the same time interval. The error from the reduced order model is measured against the high fidelity model. Additionally the “truth” solution data is projected into the subspace of dimension \(r\). The difference between the ROM
error and projection error demonstrates the error associated with the numerical integration and projection of dynamical operators. In Figure 5.3 the results are shown for the cases where a centering trajectory has and has not been used.

![Figure 5.3: POD ROM Relative Error for various $r$](image)

From this scenario it is evident the results are comparable with and without using a centering trajectory. This is likely attributable to the fact that the mean of the solution is on the order of the perturbations. By increasing the size of the POD basis, the relative error in the ROMs can be reduced to the iterative tolerance that is used in both the high fidelity and reduced order model. The error in the projected data continues to decrease down to machine precision which indicates that the ROM solution error is dominated by the integration error at approximately $r = 35$. Recall in theorem 4.1 the standard POD error for any $r$ is based on the truncated singular values. In Figure 5.4 the analytic relative error bound is shown with the performance of the uncentered POD ROM. The use of the uncentered approach is to provide a clear picture of the errors in the full data set. In this case however, performance is similar between the centered and uncentered ROMs. There is
good agreement between the performance of the ROM and the analytic error bound up to the iterative tolerance. Moreover, there is good agreement between the analytic error and error resulting from the projection of the data onto the POD basis up to machine precision. The conclusion here is the basis obtained from the POD method and ultimately the reduced order model achieves the performance indicated by the singular values.

In Figure 5.5 the computation times are shown. The times are the same regardless of the choice of employing a centering trajectory or not. The full order model required less than 5 seconds to run which is shown by the blue line. The red line represents the time need to precompute the tensor terms, while the light blue line represents the time required to solve the ROM. The total time is shown in the green line. For this 1D problem the full FEM model has a sparse tridiagonal structure and only a single convective matrix needs to be assembled at every iteration step. The convective matrix assembly can be vectorized to allow for fast online construction. This can be seen in the ROM precompute time, where two tensors are built using $r$ projections of the convective matrix built using the POD basis. The time to
complete this offline stage is negligible compared to the online solve time. Recall, since the POD basis functions are a linear combination of the FEM basis functions, in general they have global support. After projection into the $r$-dimensional subspace the sparsity of the full system matrices is lost. The combination of a fast sparse full order model and dense reduced order model in this case causes the ROM to require more time to solve than the full order model for $r > 25$.

5.1.3 Parameter Dependent Results

Next the POD ROM is tested for a variety of parameters using the various bases described in Chapter 3. The nominal parameter is chosen to be $\nu = 1/100$, and the parameter range is chosen such that the endpoints of the parameter range represent a 30% difference from the nominal parameter. This equates to a parameter range of $\nu \in [1/142.86, 1/76.92]$. For the interpolation based methods, data from full order models evaluated at the end points of the parameter range are also used for estimation. The tests were run for a variety of
dimension reduced order models. The computed POD error using the parameter dependent methods are compared against their analytic error bound and the optimal POD error at each sample parameter. The analytic error bound depends on knowing the optimal POD basis, POD basis sensitivities and the optimal POD error. The first sensitivity of the POD basis is computed using the CSE method \cite{27}. Higher order derivatives can be computed in the same way, though numerical scaling becomes an issue. To avoid this a finite difference approach is used to compute the second and higher order sensitivities of the POD basis. Since the true $\xi, \xi_i$ values are not known, for consistency, the POD basis sensitivities used in the analytic error bounds are computed for the nominal parameter. The optimal POD basis and ROM error are not known a priori, however it is possible to use the same methods to estimate the optimal POD basis and use an estimation of the optimal POD error to yield an a priori analytic error bound. Finally, we note that the analytic error bound derived using Lemma 4.3 utilized the triangle inequality and therefore are likely conservative.

The structure of the error plots is the same for each method, the blue line represents the optimal POD error at each parameter for a specified value of $r$. The green line represents the error resulting from the particular parameter dependent POD ROM and the red line represents the computed analytic error bound. The light blue line represents the analytic error bound a priori estimate using only data known at the sample points.

In Figure 5.6 the fixed basis method errors are compared over the parameter range for several values of $r$. The fixed basis is the optimal basis at the nominal parameter however as the parameter is changed, the performance of the basis quickly diminishes. It is worth noting that using a larger dimension reduced order model does not improve the error at the off-nominal parameter points. It is possible that the higher order (lower energy) POD modes can not compensate for the error in the lower order (higher energy) POD modes. For all the samples of $r$ the computed error from the optimal ROM and the a priori estimation of the analytic error bound compare well with the analytic error bound.

In Figure 5.7 the extrapolated basis method errors are compared over the parameter
range for several values of $r$. Due to the use of sensitivity information the extrapolated basis ROMs have near optimal performance for parameter samples near the nominal parameter value. As the distance from the nominal parameter increases the degrades.

In Figure 5.8 the linear interpolation basis method errors are compared over the parameter range for several values of $r$. Since the basis is interpolated this method produces the optimal basis at the sample parameters used for estimation. While the method performs reasonably well for small dimension ROMs, as $r$ is increased the performance for parameters away from the known data decreases quickly.
In Figure 5.7 the piecewise Hermite cubic interpolation basis method errors are compared over the parameter range for several values of $r$. Here again the method produces the optimal basis at the sample parameters used for estimation. However, the Hermite nature of the interpolating polynomials provides better performance near the known data points. Here, the computation of the 4th derivative of the POD basis becomes an issue. The parameter perturbation used for reasonable 3rd derivatives of the POD basis leads to a potentially unreliable computed 4th derivative. This is indicated by the estimated conservative POD error being lower than the actual computed error. In practice a suitable parameter perturbation
Figure 5.8: Parameter Dependent Error for Linear Interpolation POD basis

must be tested so that the resulting sensitivities are meaningful.

Next, we compare the methods for their performance against the optimal POD basis errors. In Figure 5.10 the results are shown for bases that require only one full order solution, i.e. the fixed, extrapolated and expanded POD methods. The blue line again represents the optimal POD error where at every parameter point the optimal POD basis has been computed from a full order solution and the optimal POD ROM of dimension $r$ solved. The green line represents the fixed basis ROM error and the red line the extrapolated basis ROM error.
Recall the expanded basis POD method described in Section 3.2.3. In this method the POD basis is augmented by the POD sensitivity vectors. Two possible choices exist for comparison, the $Exp_1$ results reflect maintaining a reduced order model of dimension $r$. Here, half of POD vectors are taken and combined with their respective sensitivity vectors to form a basis. Since the sensitivity vectors are in general different than the optimal POD vectors that they replace, some performance has been lost, at least at the nominal parameter. The assumption is that by including the sensitivity vectors the resulting ROM is more robust to the parameter changes and can provide more consistent performance across the entire range of
parameters. The other choice recognizes that the majority of the offline cost is associated with solving the high fidelity model and the corresponding sensitivity equations. If the dimension of the reduced order model is not restricted, then the sensitivities represent a different set of information than what is contained in the optimal POD basis. The $Exp_2$ results reflect using a ROM of dimension $2r$ where $r$ POD basis and $r$ corresponding sensitivity vectors are used as a basis for the ROM.

As noted before, the fixed basis errors are nearly identical for any off nominal parameter once the dimension of the reduced order model is greater than 10. The extrapolation method
performs well across the parameter range. It is noted that as $r$ increases the higher order POD modes are much more dependent on the parameter. This is seen in plots (c) and (d) of Figure 5.10, when $r = 12$ and $r = 16$, at the end points of the parameter range the relative errors for the extrapolation method are close despite increasing the basis by 4 more vectors. In contrast, near the nominal parameter the relative error drops by a full order of magnitude with the addition of the same 4 vectors. The expansion method also provides consistent performance that is more robust with respect to perturbations in the parameter. The sensitivity vectors contain information different from the optimal POD vectors. This can be seen by comparing the results for $r = 6$ and $r = 12$, by construction the $Exp_2$ results for $r = 6$ are also the $Exp_1$ results for $r = 12$. The $Exp_2$ ROM for $r = 6$ (a dimension 12 ROM) increases the performance at the nominal parameter by about half an order of magnitude, while this same result falls about a full order of magnitude short of the optimal POD performance for $r = 12$. In addition, when $r = 12$, for most of the parameter range we see that the extrapolated basis ROM performs better than the expanded basis ROM of dimension 12. In contrast for $r = 16$ the expanded basis ROM ($Exp_1$) has better performance over more of the parameter range than the extrapolated basis ROM except in a very local neighborhood of the nominal parameter. Thus, the neighborhood for which the extrapolated basis ROM is valid decreases as $r$ increases due to the stronger dependence on the parameter of the higher order POD modes.

Now the results are compared for the POD ROM methods that use multiple data points, Global POD, piecewise linear and piecewise cubic Hermite interpolation methods. In this case the full order solution data are taken from the nominal parameter and the endpoints of the parameter interval. The results are shown in Figure 5.11.

Again, the blue line represents the optimal POD error. This is the best performance that can be accomplished with a dimension $r$ POD ROM. The green line represents global POD approach where the data from the 3 high fidelity solutions are concatenated together and a single POD basis is found that is optimal for the larger data set. This basis is also fixed
Figure 5.11: Parameter Dependent Error using multiple data over the parameter range. In comparison to the fixed basis generated from just the solution at the nominal parameter, the GPOD ROM is more robust across the parameter range. However, the robustness gained comes at the cost of performance at any single parameter in the parameter space. The red and light blue lines are associated with the piecewise linear and piecewise cubic Hermite interpolants of the known optimal POD bases at the same 3 parameter test points. The performance of the interpolation methods is similar with the PCHIP POD method performing better due to the inclusion of the sensitivity information. As the dimension of the reduced order models increases the performance of the interpolation...
ROMs varies by several orders of magnitude over the parameter range. As was mentioned for the extrapolation method the higher order POD modes are more sensitive to changes in the parameter. Furthermore, the ordering of the similar basis functions with respect the parameter may change significantly. The two issues lead to the deteriorating performance of the interpolants between the sample parameters. In contrast, the performance of the GPOD ROM is consistent across the entire parameter range and for many parameter values better than an interpolation approach.

In this section the 1D Burgers’ equation has been used to demonstrate the parameter dependent POD methods and error analysis. This allows for testing where the full order models still can run within a manageable time. In the next section we consider a 2D fluid flow application. In this model the full order model dimensional is much larger than the reduced order model dimension. Also, the 2D model uses an unstructured mesh. These two facts will illustrate the efficiency that is possible through the use of reduced order modeling.

5.2 Navier-Stokes Equations

The Navier-Stokes equations are among the most studied equations in mathematics. The equations describe the motion of fluids and have ranging applications from oceans and atmospheres to economics. Numerical simulations of the Navier-Stokes equations is the subject of ongoing research. In contrast to their widespread applicability in everyday life, fundamental understanding of the properties of the solutions to the equations is still elusive. The Clay Mathematics Institute has listed the “Existence and smoothness of Navier-Stokes solutions on $\mathbb{R}^3$” as one of the Millennium Prize Problems [37].

Let $\Omega \subset \mathbb{R}^n$ for $n = 2$ or 3, be an open bounded connected domain with boundary $\Gamma = \Gamma_D \cup \Gamma_N$ and let $q \in Q \subset \mathbb{R}$. Let $u(t, x, q)$ be the velocity field and $p(t, x, q)$ be the pressure which both depend on $q$. The time dependent incompressible Navier-Stokes initial
boundary value problem on \( \Omega \times (0, T] \) is given as \[38\]

\[
\begin{align*}
\mathbf{u}_t (t, \mathbf{x}, q) + (\mathbf{u} (t, \mathbf{x}, q) \cdot \nabla) \mathbf{u} (t, \mathbf{x}, q) - \nu \Delta \mathbf{u} (t, \mathbf{x}, q) + \nabla p(t, \mathbf{x}, q) &= 0 \\
\nabla \cdot \mathbf{u} (t, \mathbf{x}, q) &= 0
\end{align*}
\] (5.7)

with boundary conditions

\[
\begin{align*}
\mathbf{u} (t, \mathbf{x}, q) \big|_{\Gamma_D \times (0, T]} &= \tilde{\mathbf{u}} \\
\nu \frac{\partial}{\partial n} \mathbf{u} (t, \mathbf{x}, q) - p(t, \mathbf{x}, q) \mathbf{n} \bigg|_{\Gamma_N \times (0, T]} &= 0
\end{align*}
\]

and initial condition

\[
\mathbf{u} (0, \mathbf{x}, q) = \mathbf{u}_0 (\mathbf{x}, q)
\]

where \( \nu \) represents the viscosity of the fluid.

Despite the differences in domains and scales, two flows may be dynamically similar. By removing the scaling effects and nondimensionalizing the systems, the dimensionless Reynolds number is introduced. The Reynolds number is defined as

\[
Re = \frac{VL}{\nu}
\]

where \( V \) and \( L \) are characteristic velocity and length scales respectively and \( \nu \) is the fluid viscosity \[39\]. We also note that here we have made the incompressibility assumption that the density of the fluid is uniform, constant with value 1. Just as in the Burgers’ equation, the Reynolds number describes the balance of inertial and viscous forces within the fluid. When the Reynolds number is small, e.g. near 1, the fluid flow is smooth due a domination of viscous forces. As the Reynolds number increases, the inertial forces become more important and ultimately lead to flows that are turbulent. Turbulence in fluid flows, typically characterized by mixing and irregularity of the flow, is a major source of ongoing research with many books
and conferences focusing solely on the topic. Understanding turbulence has been described by Nobel Laureate Richard Feynman as “the most important unsolved problem of classical physics”.

Due to the incompressibility constraint

$$\nabla \cdot \mathbf{u}(t, x, q) = 0,$$

the Navier-Stokes equations are a system of (nonlinear) Differential Algebraic Equations (DAEs). Specific numerical schemes must be used that satisfy the conditions associated with the algebraic constraints. One of the most popular classes of finite element schemes that satisfy the conditions is the Taylor-Hood mixed finite element pair $P_k - P_{k-1}$ \[38\]. Using the Finite Element Method the PDE is discretized into an ODE system of the form

$$\mathbf{M} \dot{\mathbf{u}}_N(t, q) + \mathbf{C}(\mathbf{u}_N(t, q)) \mathbf{u}_N(t, q) + \nu \mathbf{K} \mathbf{u}_N(t, q) + \mathbf{B}^T \mathbf{p}_N(t, q) = 0$$

$$\mathbf{B} \mathbf{u}_N(t, q) = 0.$$ (5.8)

For the Navier-Stokes equations, we are interested in the dependence of the solution with respect to the Reynolds number. In general the sensitivity with respect to mesh parameters or Dirichlet boundary conditions may be infinite. However, for our problem boundary conditions and the mesh are fixed thus we can define sensitivity of the solution with respect to the parameter $q$ as

$$\mathbf{s}(t, x, q) = \frac{\partial}{\partial q} \mathbf{u}(t, x, q)$$

and the corresponding sensitivity initial boundary value problem as

$$\mathbf{s}_t(t, x, q) + (\mathbf{s}(t, x, q) \cdot \nabla) \mathbf{u}(t, x, q) + (\mathbf{u}(t, x, q) \cdot \nabla) \mathbf{s}(t, x, q)$$

$$-\nu \Delta \mathbf{s}(t, x, q) + \nabla p_q(t, x, q) = \Delta \mathbf{u}(t, x, q)$$

$$\nabla \cdot \mathbf{s}(t, x, q) = 0.$$
with boundary conditions

\[
\left. s(t, x, q) \right|_{\Gamma_D \times [0,T]} = 0 \\
\nu \left. \frac{\partial}{\partial n}s(t, x, q) - p_q(t, x, q) n \right|_{\Gamma_N \times [0,T]} = 0
\]

and initial condition

\[
s(t, x, q) = \frac{\partial}{\partial q}u_0(x, q).
\]

This yields the coupled sensitivity system given by

\[
M \dot{s}_N(t, q) + \tilde{C}(u_N(t, q)) s_N(t, q) + C(u_N(t, q)) s_N(t, q) \\
+ \nu K s_N(t, q) + B^T p_N(t, q) = -K u_N(t, q) \\
B s_N(t, q) = 0.
\] (5.9)

### 5.2.1 POD ROM

This section opens with some preliminary remarks on the POD method for the Navier-Stokes equations. The POD method extracts the left singular vectors out of the data set as the projection basis. The data are snapshots of the solution with velocity and pressure components. Due to the algebraic constraint of the PDE, the velocity components satisfy the incompressibility constraint, i.e. the velocity field snapshots are discretely divergence free. If the velocity components are considered together, the POD basis which are in the span of the snapshots will also inherit this characteristic. The solutions to the POD ROM are divergence free and automatically satisfy the incompressibility constraint. Since the incompressibility condition is enforced in the momentum equation by the pressure term, the pressure term and the conservation of mass equation can be neglected in the POD ROM. In contrast, if a basis is found for the velocity components separately, there is no guarantee that the POD ROM subspace is divergence free hence, the pressure term and incompressibility constraint can not be neglected.
We consider the case where the velocity data are considered together and consequently the projection space is discretely divergence free. Within this framework the POD based reduced order model for the Navier-Stokes equations has the same form as the Burgers’ equation POD ROM. Starting from the spatially discretized system (5.8) we let \( \bar{u} \) be the mean of the snapshots, using \( \hat{\Phi} \) as the projection basis, the POD approximation is given by

\[
\mathbf{u}(t,q) = \bar{u}(q) + \hat{\Phi}y_r(t,q).
\]

Substituting this into (5.8) and dropping the pressure and incompressibility constraint gives the centered ROM system

\[
M\hat{\Phi}y_r(t,q) + C\hat{\Phi}y_r(t,q)\bar{u}(q) + C\hat{\Phi}y_r(t,q)\hat{\Phi}y_r(t,q) + (C(\bar{u}(q)) + \nu K)\hat{\Phi}y_r(t,q) = -\nu K + C(\bar{u}(q))\bar{u}(q).
\]

The nonlinear terms are handled in the same manner as described for Burgers’ equation which allows for pre-computation of tensor terms which can then be quickly contracted during the online portion of the ROM solve.

The specific model problem we consider is the flow around a cylinder. This problem is a common benchmark for the Navier-Stokes equations. It has a short time interval period and very large structures that as shown later allow for very accurate replication by POD reduced order models.

### 5.2.2 Flow around a cylinder

For purposes of comparison we use the same formulation as Hay et al. [15]. This is a typical square cylinder in a free stream flow. The length of the computational domain has been normalized which yields a height of \( 13/22 \). The length of the sides of the cylinder are \( 1/22 \) and the constant horizontal inflow velocity is also normalized to 1. The walls of the cylinder
have the ‘no-slip’ condition applied, and the top, bottom, and right outer domain boundaries are outflow boundaries. Thus, the viscosity is given by

$$\nu = \frac{1}{22 \, Re}.$$  \hspace{1cm} (5.10)

The domain with an unstructured triangular mesh consisting of 10660 elements is shown in Figure 5.12.

![Model Problem Domain and Finite Element Mesh](image)

**Figure 5.12: Model Problem Domain and Finite Element Mesh**

The discretization chosen is the ‘div-stable’ Taylor-Hood $P_2$–$P_1$ mixed finite element pair [38]. The time stepping is accomplished using the trapezoidal scheme in conjunction with Newton’s method to solve the nonlinear problem at every time step. The time step chosen is $\Delta t = 0.005$ and the Newton’s iteration tolerance is $10^{-10}$. The model is initialized with a zero steady state solution and the inflow velocity linearly ramps to the constant inflow velocity over one second.

For the reduced order modeling of this problem we are interested in only the periodic vortex shedding. To ensure the resulting snapshot data is only from the periodic orbit, the model is evolved for a specified length of time before data is collected. For this problem 30
seconds is used, this seems sufficient since by observation the system has reached its limit cycle after about approximately 5 seconds. In Figure 5.13, the vorticity of a typical solution and corresponding sensitivity with respect the parameter \( q = \nu \) are shown for \( Re = 100 \).
Fixed Parameter Results

The performance of the POD reduced order model is tested for the nominal parameter $Re = 100$. At this value the flow exhibits the periodic von Karman vortex street and based on the characteristic length and velocity, the length of the period is 0.31 seconds. Data from the full order solution is collected over one full period beginning from $t = 30$. The standard POD ROM is then applied and the relative error associated with the choice of $r$ is shown below in Figure 5.14. Due to the particular structure of the problem, the rank of the data set is no more than 63 which is the number of snapshots collected over the period. The number of spatial degrees of freedom is 42932. In this case the method of snapshots provides a much more efficient method for computing the POD basis. The singular values are computed to reach machine precision after about $r = 40$ at this point singular values and corresponding singular vectors are no longer accurate. Recall the method of snapshots first solves for the singular vectors in the time domain, then computes the spatial POD basis using (2.4). Thus, the POD basis is susceptible to large numerical error when the error in computing the singular values is comparable to the singular values themselves.

The figure shows the error resulting from both the integration of the reduced order model and the direct projection of the full order model data into the POD subspace for both the centered and uncentered approaches. The relative error resulting from the projection of the full order model data is the optimal for any choice of $r$ up to the rank of the data set. From the results we see that the error obtained by solving the centered POD ROM follows the projection error of the solution and the integration error is negligible up to about $r = 40$. At this point the integration error accumulation in the lower energy POD modes begins to dominate the overall ROM performance and as more modes are added the error increases. Specifically, the coefficients associated with the initial condition for POD modes greater than $r > 40$ have magnitude less than about $10^{-8}$ decaying down to less than $10^{-14}$. After the first time step the computed coefficients all have magnitude greater than $10^{-9}$. Thus, the iteration error tolerance has introduced artificial energy into the ROM. After many time
steps, this artificial energy accumulates affects the overall performance of the ROM solution. Methods to address this issue have been investigated and are based on turbulence modeling principles. Specifically, Wang gives a thorough review of methods based on the Kolmogorov’s energy cascade theory and proposes several POD closure models [40].

The error obtained by evaluating the uncentered POD ROM decreases for the first few POD modes and the performance stops at about 0.007 relative error. As discussed in Section 2.4, if the mean of the data set is large compared to the POD modes, the first POD mode should be a scaled version of the mean. In Figure 5.15 the mean of the data set and the difference in the normalized mean and first uncentered POD mode are compared. In (c) the computed coefficient for the first POD mode in the uncentered POD ROM is shown compared to the coefficient obtained by projection of the full order model data. Since the first mode in the uncentered POD ROM case represents more energy than the other modes, any numerical errors associated with this coefficient could dominate the ROM performance. Here, the first uncentered POD mode represents about 98% of the POD energy according
to the POD eigenvalues.

Figure 5.15: Comparison of mean and first uncentered POD mode

In Figure 5.16, the computation times for the ROM models for various \( r \) are shown. The full order model required approximately 14 minutes to compute the solution over one period in the full order model approximation space. In contrast the ROM can be solved in less than 30 seconds with an relative error less than \( 10^{-8} \) for \( r = 40 \). Also in the same figure we see the actual ROM solve time takes about 1.3 seconds, with the remaining time devoted to the pre-computation of the tensor terms and some setup overhead. Thus, after the initial pre-computation, the reduced order model could be integrated over a very long time interval.
very quickly.

**Parameter Dependent Results**

As for Burgers’ equation example, for a single parameter the standard POD ROM is able to accurately replicate the high fidelity model as $r$ increases. We now test the POD ROM for a variety of parameters using the various bases described in Chapter 3. The nominal parameter is chosen to be the $\nu$ value corresponding to $Re = 100$, and the parameter range is chosen such that the endpoints of the parameter range represent a 30% difference from the nominal parameter. This equates to a parameter range of $Re \in [76.92, 142.86]$. To collect data for each parameter value the full order model is started from a consistent initial condition and evolved for some time so that the transient dynamics between orbits in not captured. Once on the orbit, the solution sensitivity with respect to $\nu$ is computed beginning from a zero initial condition. Data is collected over the time interval of 0.31 seconds, the length of the period at the nominal parameter, which again contains 63 snapshots. This
allows the solution sensitivity and ultimately the POD basis sensitivity to contain only the information from the periodic regime for each parameter sample point. For this problem it is difficult to obtain sensitivities higher than first order due to scaling and roundoff error. In this parameter range the Navier-Stokes solution is periodic and therefore bounded. However, the length of the period depends on the parameter, hence the sensitivity solution grows over time as solutions for different parameters diverge from a common initial condition. For these reasons, we only compare single and multiple data methods for various $r$ values. The plots have the same structure as before. In the case where multiple POD bases are used for interpolation the congruence transformation was applied and was indeed necessary for meaningful results.

First, the bases are tested that require only one full order solution, to include the sensitivity based POD ROMs. The results are shown in Figure 5.17. We note a similar structure as the Burgers’ equation plots. The fixed basis POD ROM performs well only at the nominal parameter for which the data is collected. At the end points of the parameter interval again the fixed basis POD performance is similar for the various dimension ROMs tested, so there is no benefit from adding additional basis functions. The next major observation relates to the Expanded basis POD ROM, $Exp_2$. In contrast to the Burgers’ equation results that showed the expanded POD method increased the performance at the nominal parameter, here the expanded POD method does not appreciably increase the performance of the Fixed POD ROM at the nominal parameter. Since the sensitivity solution begins at zero and grows, the second period of solution sensitivity data has larger scaling than the first. After some trials the data from approximately the third period was found to provide the best performance. Consequently, the extrapolation and expanded POD ROM $Exp_2$ results are similar over the entire parameter range. Again, the higher POD modes are more sensitive to perturbations in the parameter and as $r$ increases the region of validity for all the single data POD ROMs shrinks.

Next the results are compared for the POD ROM methods that use multiple data points.
Figure 5.17: Parameter dependent ROM relative error using single data shown for various \( r \) and \( Re \) values

In this case the full order solution data are taken from the nominal parameter and the endpoints of the parameter interval to build the interpolating polynomials. The results are shown in Figure 5.18. From the figure it is seen that the GPOD error is consistent across the parameter range, however the performance increases very slowly as \( r \) increases. The addition of the 4 additional basis function from \( r = 12 \) to \( r = 16 \) only decreases the average error slightly across the entire parameter range. The LI and PCHIP methods are interpolation based and so their performance is optimal at the sample parameter points. While there is an increase in the performance by using a Hermite interpolant the increase is much less
Figure 5.18: Parameter dependent ROM relative error using multiple data shown for various \( r \) and \( Re \) values

noticeable than for the Burgers’ equation results. We note that this problem has a sensitive
dependence on the parameter and the higher order POD modes that depend on the solution
sensitivity are numerically difficult to compute. Thus, the analytic error bounds which were
shown in the Burgers’ equation example to be conservative are in this case not useful.

In this section a 2D periodic flow was investigated. The change from 1D to 2D necessi-
tated the use of the method of snapshots and the use of the centering trajectory. Furthermore,
the improvement of the POD ROMs based of incorporation of the sensitivity information
was demonstrated. In the next section a non periodic 2D application is investigated.
5.3 Boussinesq Equations

The Navier-Stokes equations describe the motion of a fluid alone with no consideration for the influence of temperature differences. A temperature variation leads to a nonuniform density in the fluid and the buoyancy forces in turn drive the motion of the fluid in the vertical direction. In the general form of the problem, the fluid parameters become dependent on the temperature and are unnecessarily difficult when the temperature differences are small. The Boussinesq approximation assumes that the fluid density is constant, the viscous dissipation is negligible and the fluid properties are constant with respect to the temperature. These assumptions give the incompressible Navier-Stokes system coupled to a convection diffusion equation by a gravity induced forcing term proportional to the temperature difference. The system is referred to as the Boussinesq or Oberbeck-Boussinesq equations which are suitable for natural convection type systems.

The time dependent nondimensionalized Boussinesq equations are given as [38]

\[
\begin{align*}
  u_t (t, x, q) + (u (t, x, q) \cdot \nabla) u (t, x, q) & - \frac{1}{Re} \nabla u (t, x, q) + \nabla p (t, x, q) = \frac{Gr}{Re^2} T (t, x, q) \\
  T_t (t, x, q) + (u (t, x, q) \cdot \nabla) T (t, x, q) & - \frac{1}{RePr} \nabla T (t, x, q) = 0 \\
  \nabla \cdot u (t, x, q) & = 0.
\end{align*}
\]

(5.11)

Here \( Re \) is the Reynolds number, \( Gr \) is the Grashof number which represents the ratio of buoyancy to viscous forces and \( Pr \) is the Prandtl number which represents the ratio of momentum to thermal diffusion of the fluid. The vector \( e \) is the canonical unit vector for the vertical plane, hence the buoyancy affects the vertical component of the velocity directly.

Since the Boussinesq equations contain the Navier-Stokes equations, the Boussinesq equations are also a system of Differential Algebraic Equations (DAEs). Due to coupled nature of the velocity and temperature, it is common to use the same discretization for both. Using
the finite element method the PDE is discretized into an ODE system of the form

\[
\begin{align*}
M\dot{u}_N(t,q) + C(u_N(t,q))u_N(t,q) + \frac{1}{Re}Ku_N(t,q) + B^T p_N(t,q) &= eG_r MT_N(t,q) \\
M\dot{T}_N(t,q) + C(u_N(t,q))T_N(t,q) + \frac{1}{RePr}KT_N(t,q) &= 0 \\
Bu_N(t,q) &= 0
\end{align*}
\]

(5.12)

### 5.3.1 POD ROM

Recall the case of the Navier-Stokes equations the horizontal and vertical velocity components are treated as a single block of data. Doing so ensures the POD basis and the resulting POD ROM are a within a (discretely) divergence free subspace of the finite element approximation space. For the Boussinesq equations, the velocity components are also treated as single block of data to maintain the divergence free subspace. The temperature data is treated separately requiring a different POD basis for the velocity components and temperature components.

The POD based reduced order model for the Boussinesq equations resembles Navier-Stokes equation POD ROM. Here again the centering trajectory is applied, both for the velocity and temperature components. Let \( \bar{u} \) be the mean of the velocity snapshots with perturbations \( y \) and \( \bar{T} \) be the mean of the velocity snapshots with perturbations \( w \). We denote the projection basis for the velocity as \( \hat{\Phi}_y \) and the projection basis for the velocity as \( \hat{\Phi}_w \). Thus the POD approximation for \( u \) and \( T \) are given by

\[
\begin{align*}
\\bar{u}(t,q) &= \bar{u}(q) + \hat{\Phi}_y y_r(t,q) \\
\bar{T}(t,q) &= \bar{T}(q) + \hat{\Phi}_w w_r(t,q)
\end{align*}
\]

Then starting from the spatially discretized system (5.12), dropping the pressure and incompressibility constraint and using the above expansions gives the coupled centered POD
ROM system (suppressing dependencies)

\[
M \dot{\hat{\Phi}} y_r + \left( C (\bar{u}) + C (\dot{\hat{\Phi}} y_r) + \frac{1}{Re} K \right) \dot{\hat{\Phi}} y_r + C (\dot{\hat{\Phi}} y_r) \bar{u} = - \left( \frac{1}{Re} K + C (\bar{u}) \right) \bar{u} + e \frac{Gr}{Re^2} M T
\]

\[
M \dot{\hat{\Phi}} w_r + \left( C (\bar{u}) + C (\dot{\hat{\Phi}} y_r) + \frac{1}{Re Pr} K \right) \dot{\hat{\Phi}} w_r + C (\dot{\hat{\Phi}} y_r) \bar{T} = - \left( \frac{1}{Re Pr} K + C (\bar{u}) \right) \bar{T}.
\]

Next the POD ROM for the Boussinesq will be tested for a natural convection type problem.

### 5.3.2 Radiator

Natural convection is a ubiquitous phenomenon. Many buildings use natural convection systems either as the only method for heating and cooling or to supplement other existing heating or cooling systems. Chilled beams, heated floors and radiators alone are simple heat exchangers, heating or cooling the local medium. However if large enough, the local temperature differences in turn provide the buoyant forcing necessary to begin circulation of the flow in a large region.

To investigate this we utilize a radiator in a room under a chilled wall. This setup represents a typical unforced convection loop driven by a radiator. In many instances the radiator is placed directly under a window. During the time that a radiator would be used for heat, the window is usually cooler than the walls of a room. This is due to less insulation in the window or by small gaps that allow the cool air to enter the room. In this simplistic model the room is assumed to be sealed with no slip boundary conditions for the velocity. The temperature uses Dirichlet boundary conditions on the wall and the radiator and zero flux everywhere else.
The domain with an unstructured triangular mesh consisting of 9704 elements is shown in Figure 5.19. The discretization chosen is the ‘div-stable’ Taylor-Hood $P_2$–$P_1$ mixed finite element pair. The time stepping is accomplished using the trapezoidal scheme in conjunction with Newton’s method to solve the nonlinear problem at every time step. The time step is $\Delta t = 0.005$ and the Newton’s iteration tolerance is $10^{-10}$. The model initializes to a zero steady state solution and the inflow velocity linearly ramps to the constant temperature boundary conditions over 0.1 second. In Figure 5.13, a typical temperature solution and corresponding sensitivity for $Gr = 3 \times 10^7$ are shown.

Figure 5.19: Model Problem Domain and Finite Element Mesh
(a) Temperature

(b) Sensitivity of temperature with respect to $Gr$

Figure 5.20: Typical Solution Snapshot
**Fixed Parameter Results**

The performance of the POD reduced order model is tested for the nominal parameters $Re = 1000$, $Pr = 1$, $Gr = 3 \times 10^7$. Though these parameter values may not be realistic, the model is only intended to demonstrate a reasonable natural convection properties.

The model is started from zero and one full second of data collected beginning from $t = 0.1$. The solution has 201 snapshots and 59352 combined velocity and temperature degrees of freedom. In Figure 5.21 the standard POD ROM performance is shown compared to the projection of the data. Here two cases are considered, keeping the velocity and temperature data together and generating a POD basis, denoted by “uvT ROM”. The other case computes a POD basis for the velocity and temperature data separately resulting in a block POD basis, denoted “uv_T ROM”. In the former case where the data is considered as one block, each POD basis vector contains a horizontal and vertical velocity component as well as a temperature component. Consequently, when computing the POD ROM, at each time step the coefficients of the POD basis constrain the velocity and temperature approximations to vary together. For independent but coupled variables this may not be appropriate. As was seen in the case of the Navier-Stokes equations, treating the directional velocity components in this manner is justified due to the incompressibility constraint. In the case of natural convection the velocity and temperature variables are strongly coupled. However, decoupling the velocity and temperature data and computing basis separately allows the variables to evolve independently according to the dynamical system. Specifically, as the parameter of the dynamical system is changed the relationship between the variable fields is expected to change. In Figure 5.21 it is observed that using a separate basis provides a better approximation. To be clear, when separating the data $r$ POD basis vectors are computed for the velocity data and an additional $r$ POD basis vectors are computed for the temperature data. The resulting POD basis used for the ROM is an $r + r$ block structure POD basis. When the data velocity and temperature are considered together a single $r$ dimensional POD basis is computed. In both cases it is evident that this model is harder to
reduce. Comparing to the flow past a cylinder model, in Figure 5.14 it requires only about 40 POD basis vectors to achieve a relative error of $10^{-8}$. Neither of the methods here reach that level of performance. Furthermore, though the POD error should be monotonically decreasing as seen in the projection error, low $r$ values where the addition of a vector can slightly increases the error. This is due to the conditioning of the ROM linear solves and numerical error. This ROM error occurs more frequently for the uvT POD.

![Figure 5.21: POD ROM relative error for various $r$](image)

In Figure 5.22 the computation times for the ROM models for various $r$ are shown. The full order model required approximately 30 minutes to compute the solution. Recalling that the uv_T POD uses $r$ basis vectors for both the velocity and temperature separately the computation times for the two methods agree. The high number of POD modes required for accuracy in this problem drive the large precompute and online computation times. For the remainder of this section we use only the split velocity and temperature basis method uv_T POD.
Parameter Dependent Results

Now the POD ROM methods are tested for a variety of parameters. The parameter of interest here is the Grashof number $Gr$. This parameter describes the ratio of buoyant to viscous forces in a fluid. A change in the Grashof number can represent changing either the fluid viscosity or changing the temperature difference in a fixed fluid. We consider a nominal parameter value of $Gr = 3 \times 10^7$ and a parameter range of 10% in either direction. The sensitivity of the POD basis with respect to the parameter is computed using finite differences.

First, we test the bases that require only one full order solution, to include the sensitivity based POD ROMs. The results are shown in Figure 5.23. From the figure we see in every case the extrapolation method is able to improve the results of the reduced order model for the entire parameter range. In the $Exp_2$ results we see that for the higher dimension ROMs the addition of the sensitivity vectors provides a better performance across the parameter range, though for the low $r$ values the addition of sensitivity vectors directly to the POD basis increases the error. This is likely due to the conditioning and numerical error of the system. The error is also non-symmetric for expansion methods. As the Grashof number decreases the solution changes less rapidly. Hence, the region of applicability of the sensitivity vectors
is larger than the even more convection dominated case of higher Grashof numbers.

Next the results are compared for the POD ROM methods that can use multiple data points. In this case the full order solution data are taken from the nominal parameter and the endpoints of the parameter interval to build the interpolating polynomials. The results are shown in Figure 5.24. As in the prior examples, the interpolation methods in most cases provide a significant improvement over the fixed GPOD basis of similar dimension. For this problem as well we see that though the sensitivity information can provide an improvement to the interpolation, the advantage may be small. Also, if the parameter range is large the
Hermite interpolation may have larger errors than the standard linear interpolation.

It general not every problem is a good candidate for model reduction. Due to the non-periodic structures and fine detail, this problem is challenging from a POD ROM standpoint. Nevertheless, the sensitivity information provides an improvement over the standard POD ROM. As a final remark, when the POD basis was developed for the velocity and temperature as a single block of data (the uvT POD method), the resulting POD ROM was not stable when being evaluated at moderately off-nominal parameters. The constrained coupling of the velocity and temperature coefficients is not suitable as a parameter dependent model.
Chapter 6

Conclusions and Future Research

6.1 Review of Results

The focus of this dissertation is to provide error analysis for various Proper Orthogonal
Decomposition (POD) reduced order modeling methods. In Chapter 3 the Fixed POD,
Extrapolated POD, Linear Interpolation POD and Piecewise Cubic Hermite Interpolation
(PCHIP) POD methods are described. In Chapter 4, the error for each of these methods is
represented in terms of the optimal POD error of a given dimension plus an error term that
depends on the optimal POD basis and POD basis sensitivities with respect to a parameter.
The particular framework presented is abstract so the results are independent of specific
spaces and numerical schemes. The new results are specifically Theorems 4.4, 4.5, 4.8 and
4.12 as well as Corollaries 4.9 and 4.13. In Chapter 5, numerical experiment demonstrate the
methods for the 1D Burgers’ equation, 2D Navier-Stokes equations and finally 2D Boussinesq
equations, common PDEs used in fluid applications. In the 1D Burgers’ equation analytic
error bounds were demonstrated that are conservative, moreover a priori estimations of
the error bounds were shown to agree well with the computed error bounds. In the 2D
Navier-Stokes example, the need to apply the centering trajectory was demonstrated and
investigated for the case when the mean represents a significant portion of the system energy.
In the 2D Boussinesq example, a PDE consisting of two independent but coupled variables was demonstrated. The need to identify a separate basis for each was also shown.

### 6.2 Conclusions

This study of parameter dependent reduced order models is driven by need for surrogate models that have sufficient accuracy yet are computationally much cheaper to evaluate. This need arises in many wide ranging applications of optimization, control and uncertainty quantification. It is well known that POD produces an optimal surrogate model of dimension \( r \) for a given dataset and the projection basis is depends strongly on the data. This dependence provides the performance but also the potential lack of robustness for systems that involve parameters. It has previously been demonstrated that the use of sensitivity information can provide a parameter dependent POD basis that is more robust with respect to changes in parameters. In this thesis error bounds were derived to quantify the performance that can be expected as the parameter is changed. The results shown are conservative because of the triangle inequality used in the analysis. However, the estimated a priori error bounds can provide indications where the parameter dependent POD ROM may no longer be valid. From this work, it is seen that the sensitivity based reduced order models can provide surrogate models that are robust with respect to moderate parameter changes while providing near optimal POD performance for a single parameter. Furthermore, the methods are optimal in the sense that the projection basis is parameter dependent and not static over a range of parameters. This serves to allow the reduced order models to be truly low order models. This contrasts directly with the GPOD method which uses a larger data set and a fixed global basis. In many of the examples shown, using the static GPOD basis performed worse over the parameter range than using a dynamic parameter dependent basis. The severe dependence of the lower energy POD modes to the parameter also means that maintaining very high accuracy for a wide parameter range is unlikely when the solution changes rapidly.
with respect to the parameter.

The ability to compute the higher order sensitivities necessary for the a priori error may be problematic depending on the scaling of the problem and how quickly the parameter changes. Automatic differentiation based methods can be employed but in some cases can significantly increase the computational cost of a single full order solution. The underlying advantage of automatic differentiation methods is usually high accuracy since the basic computational operations are well conditioned. In cases where the data is obtained from an experiment or a priori the parameter of interest may not be known, finite difference methods can be used to compute sensitivities after the fact.

6.3 Future Work

In spite of the vast amount of research to develop and improve model reduction for parameter dependent systems there are many open areas of research. A logical next step is to extend the parameter dependent basis ideas further. Research has been done by Farhat to extend reduced order bases interpolation in a more general framework using ideas from manifold interpolation [41]. These methods make interpolation in higher dimension parameter spaces possible while preserving properties like orthogonality of the resulting basis. It may be possible to develop higher order interpolations to include Hermite interpolations in this framework as well.

While the results proved in the thesis require the true POD basis and sensitivities to compute the analytic error bound it was shown in Section 5.1 that an estimated a priori error bound could provide useful information regarding the performance of the methods proposed here. The error bounds themselves were most likely conservative leaving room for future improvement. The reduced basis methods utilize both adjoint methods and statistical estimation techniques to achieve “certified” error bounds of a parameter range for linear and some nonlinear problems [9,11,13]. Improvement in the a priori error bounds presented
here may be improved using similar methods.

Ultimately, there are many opportunities relating to the application of the parameter dependent reduced order models. For control purposes, the solution of nonlinear Riccati equations lead to state feedback controllers which are ubiquitous in modern life. Controllers for nonlinear systems are designed to stabilize around a particular operating point that can depend on parameters. To avoid the solution for large scale Riccati equations, which may be computationally intractable, reduced order methods is a active research area. Methods that utilize projection for the reduction steps may benefit from a parameter dependent projection basis. Thus the time to develop an appropriate projection basis can be reduced. This could allow for near real time development of feedback controllers based on parameter or state dependent operating conditions.

In the case of optimization, the advantage of cheap and accurate surrogate models is clear. Moreover, the sensitivity of the projection basis can provide additional information that can speed up the search. For transition between steady and unsteady flows, the extrapolation using POD eigenvalues and corresponding sensitivities for lower energy modes can help identify the critical bifurcation values. When a nominal operating condition is known beforehand the robustness of the parameter dependent POD ROMs developed here can be used to perform uncertainty quantification based on potential operating conditions that the physical system may encounter.

The POD ROM methods here seek to represent the entire state optimally, however when a system output is the primary interest a different form of projection basis may be desired. When the system is linear, the transfer function is used to understand the input/output relationship. Balanced POD provides one method of generating an approximately balanced reduced order model [42]. It may be possible to use the POD basis in conjunction with the sensitivity information to develop a parameter dependent approximately balanced system.
Bibliography


