Sensitivity Enhanced Model Reduction

Drayton W. Munster

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

Masters of Science
in
Mathematics

Lizette Zietsman, Chair
Jeffrey T. Borggaard
Serkan Gugercin

15 May 2013
Blacksburg, Virginia

Sensitivity Enhanced Model Reduction

Drayton W. Munster

(ABSTRACT)

In this study, we numerically explore methods of coupling sensitivity analysis to the reduced model in order to increase the accuracy of a proper orthogonal decomposition (POD) basis across a wider range of parameters. Various techniques based on polynomial interpolation and basis alteration are compared. These techniques are performed on a 1-dimensional reaction-diffusion equation and 2-dimensional incompressible Navier-Stokes equations solved using the finite element method (FEM) as the full scale model. The expanded model formed by expanding the POD basis with the orthonormalized basis sensitivity vectors achieves the best mixture of accuracy and computational efficiency among the methods compared.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Background . . . 1
  - 1.2.1 Notation . . . 1
  - 1.2.2 Finite Element Method (FEM) . . . 2
  - 1.2.3 Proper Orthogonal Decomposition (POD) . . . 4
  - 1.2.4 Sensitivity Analysis . . . 7
  - 1.2.5 Differential-Algebraic Equations . . . 7

2 Approximation Methods **11**
- 2.1 Baseline Model . . . 11
- 2.2 Polynomial Interpolation . . . 12
  - 2.2.1 Extrapolated Basis . . . 12
  - 2.2.2 Hermite Interpolation . . . 12
- 2.3 Global POD . . . 12
- 2.4 Expanded Basis . . . 13

3 Numerical Experiments **14**
- 3.1 1D Reaction-Diffusion . . . 14
  - 3.1.1 Model . . . 14
  - 3.1.2 Results . . . 15
- 3.2 2D Incompressible Navier-Stokes . . . 17
  - 3.2.1 Incompressible Navier-Stokes Equations . . . 17
  - 3.2.2 Model . . . 19
  - 3.2.3 Results . . . 19

4 Conclusions **23**

Bibliography **25**

List of Tables

3.1 Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models for Reaction-Diffusion 15
3.2 Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Reaction-Diffusion 16
3.3 Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Reaction-Diffusion 17
3.4 Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models for Navier-Stokes 20
3.5 Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Navier-Stokes 20
3.6 Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Navier-Stokes 21

List of Figures

1.1 Piecewise Linear 'Hat' Functions 4

3.1 $u(\frac{1}{2}, t)$ for varying β 14
3.2 Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models Reaction-Diffusion 15
3.3 Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Reaction-Diffusion 16
3.4 Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Reaction-Diffusion 17
3.5 Computational Domain and Sample Triangulation 19
3.6 Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models for Navier-Stokes 20
3.7 Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Navier-Stokes 21
3.8 Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Navier-Stokes 22

Chapter 1

Introduction

1.1 Motivation

In a wide variety of practical/physical problems, such as fluid flow, plasma physics, neutron transport, etc, the mathematical models used to describe these phenomena result in systems of partial differential equations (PDEs). While certain geometries and problems lend themselves to analytic solutions, it is often necessary to numerically approximate the solution to these PDEs. The PDEs are first discretized spatially using techniques such as Finite Difference Methods (FDM), Finite Element Methods (FEM), Finite Volume Methods (FVM), etc, to yield a system of ordinary differential equations (ODEs). These ODEs are then solved with a variety of techniques to yield the numerical approximation to the solution at given times.

However, due to accuracy requirements, the resulting system of ODEs may be very large (corresponding to a finer spatial discretization) and solving this system may be prohibitively expensive. Moreover, we are often interested in the behavior of the solution over a wide range of parameter values (e.g., Reynolds number in computational fluid dynamics) where the run-time of a single simulation may be very high. In these situations, it is not feasible to compute the solutions to a wide range of parameters.

In this thesis, we will explore techniques to use information at a (small) sample of parameter values to construct reduced order models that capture the dynamics of the solution over a wide range of parameter values, thereby reducing the computational requirements to explore a system's behavior.

1.2 Background

1.2.1 Notation

In this thesis, we will concern ourselves with the space $L^2(X)$, the square-integrable functions on X. The usual norm and inner product apply, that is,

$$\langle f, g \rangle = \int_X f(x)g(x)dx$$
$$||f||^2 = \int_X |f(x)|^2 \, dx = \langle f, f \rangle$$

1.2.2 Finite Element Method (FEM)

The Finite Element Method (FEM) is a popular tool for solving PDEs. The FEM involves discretizing the domain of interest, X, and approximating the PDE with a system of ordinary differential equations (ODEs).

Test Functions In order to begin our discussion of the FEM, we must first discuss test functions. We say that $\phi : X \to \mathbb{R}$ is a **test function** if:

1. ϕ has compact support ($\phi \equiv 0$ outside of a compact set),
2. $\phi \in C^d(X)$ for a suitable d.

Note that in most analysis texts, ϕ is required to be infinitely differentiable, i.e. $d = \infty$. However, we require only enough smoothness for our integrals to be defined as we will see later. Typically, $d = 0$ is sufficient.

Weak Formulation To explore the weak formulation of a PDE, consider the linear heat equation in one spatial dimension with homogeneous Dirichlet boundary conditions. We seek a function $u \in C^2([a, b])$ such that

$$u_t = \alpha u_{xx} \tag{1.2.1}$$
$$u(a) = u(b) = 0 \tag{1.2.2}$$

where subscripts refer to the derivative with respect to the subscript. If $v \in V = H_0^1([a, b])$, that is, the space of functions that vanish at a, b and have a weak spatial derivative, then we may multiply this equation by v and integrate over the domain to yield

$$\int_a^b u_t(x, t)v(x)dx = \alpha \int_a^b u_{xx}(x, t)v(x)dx.$$

On the right hand side, we may integrate by parts to yield

$$\int_a^b u_t(x, t)v(x)dx = -\alpha \int_a^b u_x(x, t)v_x(x)dx + u_x(x, t)v(x)|_a^b.$$

However, since $v \in H_0^1([a, b])$, this boundary term vanishes. Since $v \in V$ was arbitrary, we are left with the weak formulation:

$$\int_a^b u_t(x, t)v(x)dx = -\alpha \int_a^b u_x(x, t)v_x(x)dx \qquad \forall v \in H_0^1([a, b]). \tag{1.2.3}$$

Solutions to (1.2.1) are known as classical solutions and require two spatial derivatives to exist for u. Solutions to (1.2.3) are known as weak solutions. Classical solutions are also weak solutions, but there is a larger class of functions that can satisfy (1.2.3). In order for (1.2.3) to be defined, u need only have a single weak spatial derivative. Thus, we can relax our requirements on u from $C^2([a, b])$ to $H_0^1([a, b])$. Note that finding such a $u \in H_0^1([a, b])$ is still an analysis problem.

Galerkin Method The Galerkin method is a method of changing our search space from H_0^1 to a finite dimensional subspace. First, we select a finite-dimensional subspace, V^N, with basis $\{\phi_1, \ldots, \phi_N\}$, of V that we believe can accurately represent the solution u. We will discuss popular examples later. We then write $u(x,t)$ with respect to the basis of V^N:

$$u(x,t) \approx u^N(x,t) := \sum_{j=1}^{N} a_j(t)\phi_j(x) \tag{1.2.4}$$

and substitute this form into our weak formulation (1.2.3)

$$\sum_{j=1}^{N} \dot{a}_j(t) \int_a^b \phi_j(x)v(x)dx = -\alpha \sum_{j=1}^{N} a_j(t) \int_a^b \phi_j'(x)v'(x)dx \qquad \forall v \in H_0^1([a,b]) \tag{1.2.5}$$

The Galerkin method is choosing our test functions, v, to be the basis elements ϕ_i. With this, (1.2.5) becomes:

$$\sum_{j=1}^{N} \dot{a}_j(t) \int_a^b \phi_j(x)\phi_i(x)dx = -\alpha \sum_{j=1}^{N} a_j(t) \int_a^b \phi_j'(x)\phi_i'(x)dx \qquad i = 1, 2, \ldots, N \tag{1.2.6}$$

Note that $\int_a^b \phi_i(x)\phi_j(x)dx$ and $\int_a^b \phi_i'(x)\phi_j'(x)dx$ are independent of t. This system is typically represented as

$$M\dot{a}(t) = -\alpha K a(t) \tag{1.2.7}$$

where $[M]_{i,j} = \int_a^b \phi_i(x)\phi_j(x)dx, [K]_{i,j} = \int_a^b \phi_i'(x)\phi_j'(x)dx$ and $a(t) = \begin{bmatrix} a_1(t) \\ \vdots \\ a_N(t) \end{bmatrix}$. The matrix M is referred to as the mass matrix and K is known as the stiffness matrix. Thus, our infinite dimensional problem has been approximated by a system of N ODEs.

Typical Basis Functions For problems that do not require more smoothness, a popular choice for V^N are so-called 'hat' functions, piecewise linear functions. For m equally spaced nodes in $[0,1]$, these functions take the form

$$\phi_i(x) = \begin{cases} m\left(x - \dfrac{(n-1)}{m}\right) & \dfrac{(n-1)}{m} \le x < \dfrac{n}{m} \\ -m\left(x - \dfrac{(n+1)}{m}\right) & \dfrac{n}{m} \le x < \dfrac{(n+1)}{m} \\ 0 & \text{otherwise} \end{cases}$$

For $m = 3$, the 4 basis functions are shown below.

One can similarly define piecewise polynomial basis elements of higher order and elements with high conditions on smoothness, e.g. C^1 elements.

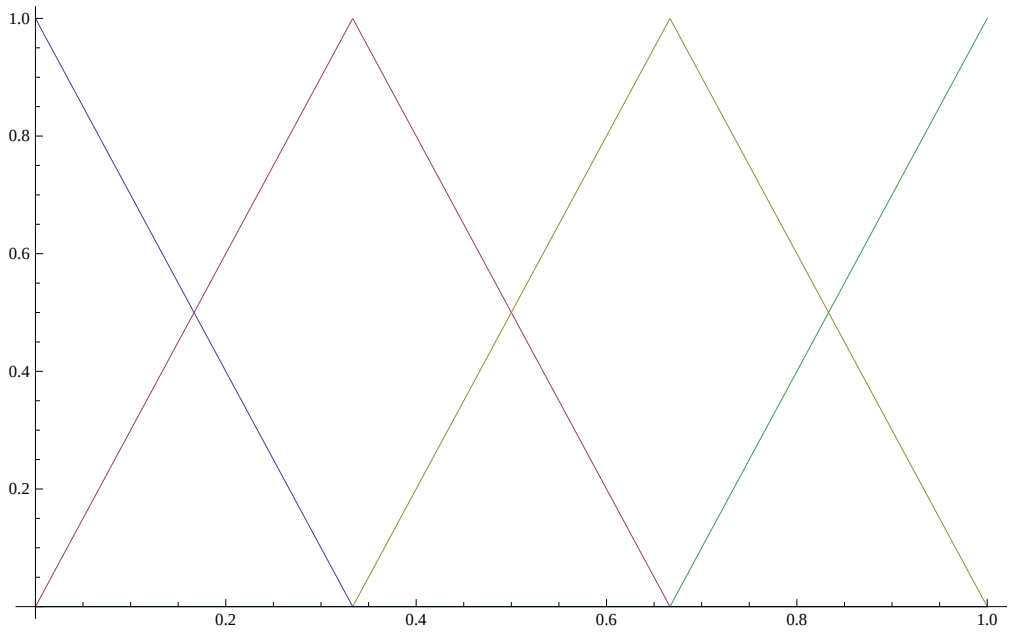

Figure 1.1: Piecewise Linear 'Hat' Functions

1.2.3 Proper Orthogonal Decomposition (POD)

The Proper Orthogonal Decomposition (POD), also known as principle component analysis or Karhunen-Loeve decomposition, is a method for optimally representing a given set of data with a lower dimensional space. Note that the POD has an infinite-dimensional analogue in a Hilbert space and L^2 functions [2]. However, we will limit our discussion to the finite-dimensional cases in this thesis. This section is modeled after the notation in [8].

POD in $\mathbb{R}^m$ Let $y_1, \ldots, y_K \in \mathbb{R}^m$. Then $\mathcal{V} = \operatorname{span}\{y_1, \ldots, y_K\}$ has dimension $d \leq \min(K, m)$. We seek a set $\Psi = \{\psi_i\}_{i=1}^{\ell}$, $\ell \leq d$, that minimizes

$$\mathcal{J}(\psi_1, \ldots, \psi_\ell) = \sum_{m=1}^{K} \left\| y_m - \sum_{n=1}^{\ell} \langle y_m, \psi_n \rangle \psi_n \right\|^2 \tag{1.2.8}$$

subject to

$$\langle \psi_i, \psi_j \rangle = \delta_{ij}$$

Ψ is known as the POD-Basis of rank ℓ. This constrained optimization problem has the associated Lagrange function:

$$L(\psi_1, \ldots, \psi_\ell, \lambda_{11}, \ldots, \lambda_{\ell\ell}) = \mathcal{J}(\psi_1, \ldots, \psi_\ell) + \sum_{i,j=1}^{\ell} \lambda_{ij} \left(\langle \psi_i, \psi_j \rangle - \delta_{ij} \right) \tag{1.2.9}$$

Optimality conditions then require (arguments omitted for brevity):

$$\frac{\partial L}{\partial \psi_i} = 0 \tag{1.2.10}$$

$$\frac{\partial L}{\partial \lambda_{ij}} = 0 \tag{1.2.11}$$

for all $1 \leq i, j \leq \ell$. However, (1.2.10) is equivalent to

$$\sum_{j=1}^{\ell} \langle y_j, \psi_i \rangle \, y_j = \lambda_{ii} \psi_i \tag{1.2.12}$$

and (1.2.11) is equivalent to

$$\langle \psi_i, \psi_j \rangle = \delta_{ij} \tag{1.2.13}$$

by [8]. Thus, letting $\lambda_i = \lambda_{ii}$ and $Y = \begin{bmatrix} y_1 & \dots & y_K \end{bmatrix}$, our minimization problem is equivalent to the symmetric eigenvalue problem

$$YY^T \psi_i = \lambda_i \psi_i \tag{1.2.14}$$

If we order λ_i such that $|\lambda_1| > |\lambda_2| > \dots$, then this is equivalent to $YY^T \psi_i = \sigma_i^2 \psi_i$ where σ_i is the ith singular value of Y and ψ_i is the associated left singular vector. Therefore, we may solve our minimization problem (1.2.8) by computing the left singular vectors of Y.

POD with Weighted Inner Product Suppose we endow $\mathbb{R}^m$ with the weighted inner product $\langle \phi, \psi \rangle_W = \psi^T W \phi$ for a symmetric, positive definite matrix $W \in \mathbb{R}^{m \times m}$ (for example, $W = M$, the mass matrix from (1.2.7)). It follows W has a Cholesky factorization R, that is, there exists an upper triangular matrix R such that $W = R^T R$. Then $\langle \phi, \psi \rangle_W = \langle R\phi, R\psi \rangle$ in the standard Euclidean inner product.

Equation (1.2.14) in this setting becomes

$$(RY)(RY)^T \psi_i = \sigma_i^2 (R\psi_i) \tag{1.2.15}$$

Thus, if $\tilde{\psi}_i$ is a left singular vector of RY, then $\psi_i = R^{-1}\tilde{\psi}_i$ is the ith POD basis vector.

Calculating the POD The above formulations yield three ways of calculating the POD basis:

1. Compute the full SVD of Y: $Y = U\Sigma V^T$.
2. Compute the eigenvectors of YY^T: $YY^T \psi_i = \sigma_i^2 \psi_i$.
3. Compute the eigenvectors of $Y^T Y$: $Y^T Y \nu_i = \sigma_i^2 \nu_i$.

1. is not used often since this can be expensive since we are typically only interested in $\ell << d$ basis vectors and we do not require the right singular vectors. 2. is computationally advantageous if $m < K$ since this is a $m \times m$ symmetric eigenvalue problem. 3. becomes computationally advantageous when $K < m$ as this is a $K \times K$ symmetric eigenvalue problem. The eigenvectors, ν_i, are the right singular vectors of Y. Thus, since $Y\nu_i = \sigma_i \psi_i$ by the SVD, we may calculate $\psi_i = \frac{1}{\sigma_i} Y \nu_i$.

Method of Snapshots A common application of the POD is called the method of snapshots. In the notation from above, let $y_1, \ldots, y_K$ be an ensemble of data at various times t_i. For example, these data vectors may represent experimental measurements or the finite element solution to a partial differential equation over time. The POD basis then represents the optimal rank ℓ approximation of the data ensemble. In particular, when Y represents the nodal coefficients over time of a finite element solution, then Ψ is the optimal representation of that solution using ℓ basis functions.

Optimality and Error Bounds In earlier sections, numerous references were made to the optimality of the POD basis vectors. We are able to state this result more formally (Cor. 1.3 in [9]).

Let $\Psi^\ell = [\psi_1, \ldots, \psi_\ell]$ and U^ℓ be any other collection of ℓ pairwise orthonormal vectors. If $\left[B^\ell\right]_{ij} = \langle \psi_i, y_j \rangle, \left[C^\ell\right]_{ij} = \langle u_i, y_j \rangle$ are the projection of Y onto the bases Ψ^ℓ and U^ℓ, respectively, then for every $1 \leq \ell \leq d$ we have

$$\left|\left|Y - \Psi^\ell B^\ell\right|\right|_F \leq \left|\left|Y - U^\ell C^\ell\right|\right|_F \tag{1.2.16}$$

where $||\cdot||_F$ denotes the Frobenius norm. Therefore, Ψ represents the optimal rank ℓ representation of Y.

Further, we can calculate the error in our representation. With $\mathcal{J}$ from (1.2.8), [8] tells us that

$$\begin{aligned} \mathcal{J}(\psi_1, \ldots, \psi_\ell) &= \sum_{m=1}^{K} \left|\left| y_m - \sum_{n=1}^{\ell} \langle y_m, \psi_n \rangle \psi_n \right|\right|^2 \\ &= \sum_{i=\ell+1}^{d} \lambda_i = \sum_{i=\ell+1}^{d} \sigma_i^2 \end{aligned} \tag{1.2.17}$$

This gives us a useful way to estimate the "energy" captured by the POD basis through [9]:

$$E(\ell) = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \tag{1.2.18}$$

However, calculating this error or the energy requires calculating all of the eigenvalues when we are typically only interested in $\ell << d$ basis vectors. Since the singular values obey $\sigma_1 > \sigma_2 > \ldots$, we may bound (1.2.8) and (1.2.18) by

$$\mathcal{J}(\psi_1, \ldots, \psi_\ell) \leq \sum_{i=\ell+1}^{d} \lambda_\ell = (d - \ell)\lambda_\ell \tag{1.2.19}$$

$$\frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{\ell} \lambda_i + \sum_{i=\ell+1}^{d} \lambda_\ell} = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{\ell} \lambda_i + (d-l)\lambda_\ell} \leq E(\ell) \tag{1.2.20}$$

In practice, ℓ is typically chosen so that (1.2.20) guarantees that $E(\ell) > \alpha$ for $\alpha \in (0, 1)$ (e.g. capturing at least 90%, 95%, or 99% of the energy).

1.2.4 Sensitivity Analysis

Sensitivity analysis refers to the study of how the solution to a given problem changes with respect to a set of parameters q. If $u(x) = u(x;q)$ is the solution to our given problem, then we can calculate the sensitivities $\dfrac{\partial u}{\partial q_i}$ when these derivatives exist.

Sensitivity of POD Modes In this thesis, we are primarily concerned with how the POD modes change with respect to a single parameter $q \in \mathbb{R}$. This material can be found in more detail in [3]. If we apply the method of snapshots, we have a matrix Y of data that depends on the parameter q. We know that the POD mode ψ is given by (1.2.14), that is

$$YY^T\psi = \lambda\psi$$

If we assume that the data, POD mode, and singular values depend continuously on q and are differentiable with respect to this parameter, then we may differentiate (1.2.14). Letting $B = YY^T$, we obtain the following:

$$\begin{aligned} B^q\psi_i + B\psi_i^q &= \lambda_i^q\psi_i + \lambda_i\psi_i^q \\ (B - \lambda_i I)\psi_i^q &= -(B^q - \lambda_i^q I)\psi_i \end{aligned} \tag{1.2.21}$$

Thus, the mode sensitivity is the solution to this linear system. However, since λ is an eigenvalue of B, the matrix $B - \lambda_i I$ is singular. Thus, we must ensure that the right hand side is in the range of $B - \lambda_i I$. This requires that the right hand side is orthogonal to the kernel of $B - \lambda_i I$. This kernel is the one dimensional subspace generated by ψ_i. Thus, we must have $\psi_i^T(B^q - \lambda_i^q I)\psi_i = 0$. Since the modes are orthonormal, this implies the following condition:

$$\lambda_i^q = \psi_i^T B^q \psi_i \tag{1.2.22}$$

With this condition in hand, we are able to solve the system in (1.2.21) for a particular solution s_i. Thus, all solutions of (1.2.21) have the form $s_i + \alpha\psi_i$ for all $\alpha \in \mathbb{R}$. Thus, we must find another condition to enforce a unique solution. This condition comes naturally from our condition that $\psi_i^T\psi_i = 1$. Differentiating this with respect to q yields $2\psi_i^T\psi_i^q = 0$ which implies the sensitivity is orthogonal to the mode. Thus, we choose $\alpha = -s_i^T\psi_i$ and we have $\psi_i^q = s_i - (s_i^T\psi_i)\psi_i$.

A similar analysis may be conducted if you calculate the POD modes through Y^TY which is available in [3].

1.2.5 Differential-Algebraic Equations

In some situations, e.g. solving Navier-Stokes equations through the finite element method, we are left with a system of differential equations subject to some algebraic constraints. This coupled system is known as a **Differential-Algebraic equation** or DAE. We will consider

semi-explicit DAEs, which have the following form:

$$\begin{aligned} \dot{u} &= F(t, u, p) \\ 0 &= G(t, u, p) \end{aligned} \tag{1.2.23}$$

where u are called the *differential variables* and p are called the *algebraic variables*. Note that if G_p is non-singular, then the implicit function theorem allows us to find a solution for p in terms of u.

The **differentiation index** of a DAE (simply index henceforth) is the number of differentiations needed to transform the system into an explicit system of ODEs [1]. For example, consider the following trivial system:

$$x(t) = g(t)$$

By differentiating once, we obtain the ODE $\dot{x}(t) = \dot{g}(t)$.

For a less trivial example, consider the system

$$\begin{aligned} \dot{x_1} &= v_1 \\ \dot{x_2} &= v_2 \\ \dot{v_1} &= -\lambda x_1 \\ \dot{v_2} &= -\lambda x_2 - g \\ 0 &= x_1^2 + x_2^2 - 1. \end{aligned}$$

This system corresponds to the motion of a pendulum in Cartesian coordinates where g represents the force due to gravity and $\lambda = \lambda(t)$ represents a Lagrange multiplier forcing the positions to the algebraic constraints. Differentiating the position constraint yields $x_1\dot{x_1} + x_2\dot{x_2} = 0$, or using the above formulas, we may derive the velocity constraint $x_1v_1 + x_2v_2 = 0$. Differentiating a second time yields $x_1\dot{v_1} + x_2\dot{v_2} + v_1^2 + v_2^2 = 0$. Substituting our formulas for $\dot{\mathbf{v}}$ yields the acceleration constraint $v_1^2 + v_2^2 - x_2g = \lambda$. This allows us to solve for x_1, x_2, v_1, v_2 but does not give us an expression for the evolution of λ. Thus, a final differentiation is necessary to yield an ODE for λ. Since three differentiations were necessary to obtain an ODE system for all variables, this DAE has index 3.

Higher indices often represent additional theoretical and numerical issues that must be addressed. In particular, higher-index DAEs include hidden constraints in addition to the explicitly stated ones [1]. These hidden constraints are the derivatives of the explicitly stated ones. For example, in the above system, the hidden constraints are the velocity and acceleration constraints derived in the process of producing an explicit ODE system.

In addition to imposing these constraints over time, it is important that the initial conditions are also given some thought. Initial conditions u_0, p_0 are said to be *consistent* if they satisfy 1.2.23 at time t_0. Often times we will only have values for u_0, in which case it becomes necessary to solve for consistent values for p_0,

Hessenberg Index 2 We will focus on a particular class of DAEs which are said to be in *Hessenberg form* of index 2 which have the following form:

$$\begin{aligned} M\dot{u} &= f(t,u,p) \\ 0 &= g(t,u) \end{aligned} \tag{1.2.24}$$

with the matrix $g_u M^{-1} f_p$ non-singular. In particular, the discretized Navier-Stokes equations have this form. Notice that g does not depend on the algebraic variables. (Note: some references will not contain the matrix M, however, this is equivalent to pre-multiplying both sides by M^{-1}.)

Solving DAEs of Hessenberg Index 2 We will consider the class of Runge-Kutta methods represented by the Butcher tableau $\begin{array}{c|c} c & A \\ \hline & b \end{array}$ with a possibly singular matrix A. Applied to a DAE of this form, an *s-stage Runge-Kutta method* at time t_n has the following form:

$$\begin{aligned} Mk_i &= f(t_n + c_i h, U_i, P_i) \\ U_i &= u_n + h\sum_{j=1}^{i} a_{ij} k_j \\ 0 &= g(t_n + c_i h, U_i) \\ P_i &= p_n + h\sum_{j=1}^{i} a_{ij} \ell_j \end{aligned} \tag{1.2.25}$$

for $i = 1, \ldots, s$ and

$$\begin{aligned} u_{n+1} &= u_n + \sum_{i=1}^{s} b_i k_i \\ p_{n+1} &= p_n + \sum_{i=1}^{s} b_i \ell_i \end{aligned} \tag{1.2.26}$$

The U_i, P_i are known as the *stage values* and k_i, ℓ_i are the *stage derivatives*. However, if A is singular, then the values ℓ_j may not be well-defined. However, we may multiply the second equation by M and simplify the equations to the following form [5]:

$$\begin{aligned} MU_i &= Mu_n + h\sum_{j=1}^{i} a_{ij} f(t_n + c_i h, U_j, P_j) \\ 0 &= g(t_n + c_i h, U_i) \end{aligned} \tag{1.2.27}$$

This allows us to find the stage values even in the case of singular A.

Selection of A, b, c When applied to index-2 DAEs, standard Runge-Kutta methods often suffer from a phenomenon known as order reduction [7]. Order reduction is where the Runge-Kutta method's accuracy does not match the expected results (e.g. an $\mathcal{O}(h^4)$ method

degrades to $\mathcal{O}(h^2)$ or $\mathcal{O}(h^1)$). In order to achieve higher orders, a Runge-Kutta method must satisfy additional order requirements. We would also like to choose our method in such a way to reduce computational expenses when possible. Additionally, we may wish to impose certain stability requirements such as L or $L(\alpha)$ stability. This requires us to select our method carefully.

In order to perform well for stiff ODEs and DAEs of index one, a Runge-Kutta method must be *stiffly accurate*, that is, $b_i = a_{s,i}$. In this case, $u_{n+1} = U_s$ and $p_{n+1} = P_s$. Stiffly accurate is a necessary condition for a method of the forms below to be L-stable.

Let n refer to the number of differential variables and m to the number of algebraic ones. A fully implicit Runge-Kutta method, i.e. one where $a_{ij} \neq 0$ for all i, j, requires solving a nonlinear system of size $s(n+m) \times s(n+m)$, which may be prohibitively expensive.

To reduce computational costs, we will consider a family of methods known as Diagonally Implicit Runge-Kutta (DIRK) methods. For such methods, we require the coefficient matrix A to be lower triangular. This allows us to replace the $s(n+m) \times s(n+m)$ system by a sequence of s systems of size $(n+m) \times (n+m)$. Additionally, if we require $a_{1,1} = a_{2,2} = \ldots = a_{s,s} \neq 0$, we obtain a family of methods known as Singly Diagonally Implicit Runge-Kutta (SDIRK) methods. SDIRK methods allow one to reuse the LU factorization of the Jacobian matrix for the nonlinear solves (possibly over more than one time-step) to further reduce computational costs. One drawback of SDIRK methods is that they are limited to stage order one, limiting their B-convergence and usefulness [6].

This limitation can be overcome by considering Explicit first-stage Singly Implicit Runge-Kutta (ESDIRK) methods. ESDIRK methods are similar to SDIRK methods but we require that $a_{1,1} = 0$ and $c_1 = 0$ and force $a_{2,2} = a_{3,3} = \ldots = a_{s,s} \neq 0$. These method may obtain stage order 2 and additionally allow for computational savings by reusing the last stage (e.g. u_n, p_n for stiffly accurate methods) as the first stage of the method. These methods enjoy high computational efficiency and a high degree of accuracy. Coefficients for a 6 stage, 4th order in differential variables, 3rd order in algebraic variables ESDIRK method can be found in [7].

Chapter 2

Approximation Methods

In many cases, we are interested in the behavior of a system over a wide range of parameters. However, due to computational expenses, sampling each point of interest in the parameter space is often unfeasible. Therefore, we seek a way to lower the cost of our simulations. We can accomplish this by reducing the dimensionality of our model. This gives us the reduced model:

$$u^R = \sum_{i=1}^{R} a_i(t)\phi_i(x) \tag{2.0.1}$$

$$\sum_{i=1}^{R} \langle A(a_i(t)\phi_i), \phi_j \rangle = \langle f, \phi_j \rangle \tag{2.0.2}$$

where R is typically much smaller than N and our test functions are chosen to better represent the dynamics of the model.

This section will focus on constructing appropriate test functions ϕ_i that allow us to perform this model reduction and approximate changes in the model parameter.

2.1 Baseline Model

Single Baseline With our optimality results discussed in 1.2.3, we see that the modes form an optimal representation of the data for a given rank. Moreover, since the POD modes are orthonormal, calculations involving the modes are often very simple. This makes the POD basis a natural choice for the test functions ϕ_i. However, this optimality only applies to the parameter point from which they were created. In order for these modes to represent data from other parameter values, we would need the dynamics to be very similar. This method requires only a single full scale model evaluation.

We will refer to the reduced model constructed from the POD at a single point as the **baseline model**.

Multiple Baselines A possible method for combining information from multiple parameter points, $\{p_1, p_2, \ldots, p_M\}$, is to orthonormalize the modes from each parameter to form the $R \cdot M$ element basis $\{\psi_{1,1}, \psi_{1,2}, \ldots, \psi_{1,R}, \psi_{2,1}, \ldots, \psi_{M,R}\}$. This method requires M full scale model evaluations, which may become prohibitively expensive as M increases (indeed, avoiding a large number of full scale model evaluations is the goal of this work).

We will refer to model constructed from this basis as the **multiple baseline model**.

2.2 Polynomial Interpolation

Due to the baseline models' weakness with respect to interpolating other parameter values, we seek a method that will allow us to improve our basis using the basis sensitivity. We can view this problem as attempting to match a function (of each basis element) given function values and derivative information. In this setting, polynomial interpolation seems very natural. Additionally, this gives us guidance on how to choose our sampling points, e.g. roots of Chebyshev polynomials on our interval of interest.

2.2.1 Extrapolated Basis

Using the POD and sensitivity from a single point, we can construct the **extrapolated basis**, $\left\{\tilde{\psi}_1, \ldots, \tilde{\psi}_R\right\}_\alpha$, by the following:

$$\tilde{\psi}_i = \psi_i + (\alpha - q)\psi_i^q \tag{2.2.1}$$

where α is the parameter of interest and the POD modes were generated at parameter value q. This method is motivated by performing a Taylor series expansion about q and truncating after the first term to obtain a linear expression in the sensitivities. This method requires only a single coupled full scale model with sensitivities evaluation. Although the sensitivities do impose additional computational costs, solving the sensitivity equations (which are linear) are typically significantly less expensive than a full scale model evaluation.

2.2.2 Hermite Interpolation

While the extrapolated basis discussed in 2.2.1 can be viewed as a special case of a Hermite interpolant at a single point, it was discussed independently due to the Taylor series motivations. If we are free to construct modes and sensitivities from multiple points in our parameter space, then we may construct higher order Hermite interpolants (e.g. cubics from 2 points, quintics from 3 points, etc.). The construction of hermite interpolants is well-studied and easily implemented. We will refer to such a basis as a **Hermite basis** with nodes a, b,etc. These methods require solving multiple coupled full scale model with sensitivity evaluations and thus, their cost increases quite significantly for higher order interpolants.

2.3 Global POD

Rather than taking the POD modes from several points and attempting to combine them, this approach attempts to find modes that represent all data points. This can be done by concatenating the data matrices:

$$\tilde{Y} = \left[\begin{array}{ccccc} Y_a & \vdots & Y_b & \vdots & \ldots \end{array}\right] \tag{2.3.1}$$

and extracting the POD modes from this $\tilde{Y}$ which will then optimally represent (for a given rank) all the given data in aggregate. We will refer to the basis constructed from this

method as a **global basis**. This method is also explored in [4]. As with the multiple baseline model (2.1), this method requires multiple full scale model evaluations but does not require sensitivity information.

2.4 Expanded Basis

It can be shown that the sensitivities, ψ_i^q span a different subspace that the original POD modes [3]. Therefore, we may construct what will be referred to as the **expanded basis** by orthonormalizing the set $\{\psi_1, \psi_2, \ldots, \psi_\ell, \psi_1^q, \ldots\}$. This basis is expected to do well since any linear combination of modes and sensitivities (e.g. the polynomial interpolants) can be represented through this basis. This method requires only a single full scale model evaluation coupled with the sensitivity equations. However, compared to the extrapolated and Hermite models (2.2), the expanded model features a larger basis which makes the reduce model evaluations more expensive (but still quite significantly less than the full model).

Chapter 3

Numerical Experiments

3.1 1D Reaction-Diffusion

3.1.1 Model

For our first numerical experiment, we will examine a 1D Reaction-Diffusion problem. These problems are of the form:

$$u_t = Cu_{xx} + R(t, u) \tag{3.1.1}$$

where $R(t, u)$ is the reaction term and u_{xx} represents the diffusion. In this particular case, we are looking at the following problem:

$$\begin{aligned} u_t &= u_{xx} + u(1-u)(u-\beta) \\ u'(0,t) &= u'(1,t) = 0 \\ u(x,0) &= \begin{cases} 1 & \frac{1}{3} \leq x \leq \frac{2}{3} \\ 0 & \text{otherwise} \end{cases} \\ x &\in [0,1] \\ \beta &\in [0,1] \end{aligned} \tag{3.1.2}$$

While this equation is used to model physical phenomena (in particular, it arises in older combustion theory [10]), this equation was primarily chosen for its behavior across parameter values. As $\alpha \to 1$, the solution decays to 0 and as $\alpha \to 0$, the solutions increases to 1. This behavior can be seen below in Figure 3.1. Therefore, this model's different behavior across parameter values provides a reasonable test problem for testing these techniques.

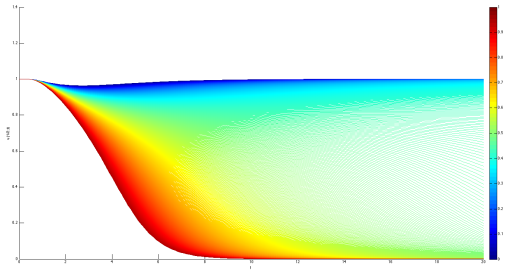

Figure 3.1: $u(\frac{1}{2}, t)$ for varying β

3.1.2 Results

For our first comparison, we will look at a comparison of the baseline, expanded, and extrapolated models.

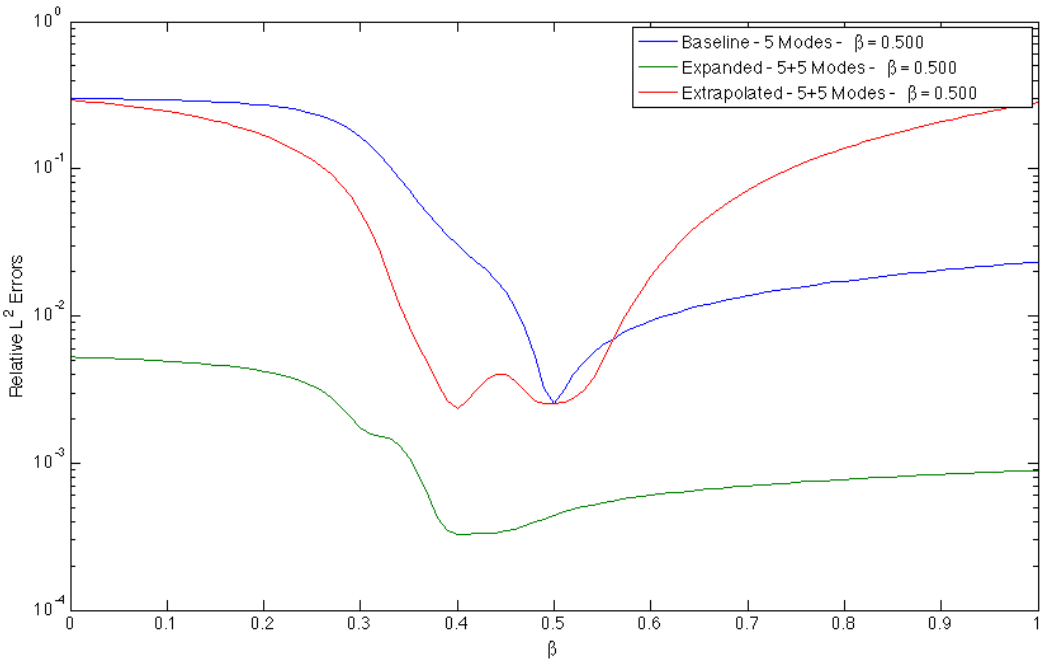

Figure 3.2: Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models Reaction-Diffusion

Reduced Model	β(s) Used	Min. Error	Max. Error	Avg. Error
Baseline	0.500	2.5360e-03	3.0053e-01	9.8830e-02
Expanded	0.500	3.2903e-04	5.2406e-03	1.7999e-03
Extrapolated	0.500	2.3264e-03	2.9234e-01	1.2009e-01

Table 3.1: Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models for Reaction-Diffusion

As we can see in Figure 3.2, the baseline reduced model performs well for the parameter value at which it was generated. However, as we get away from that parameter value (in particular, towards smaller β), the performance degrades to a 30% relative error as we can see in Table 3.1.

The extrapolated reduced model retains its accuracy over a slightly wider range of parameter values, exhibiting almost a magnitude of errors lower at $\beta = 0.4$. However, the extrapolated basis seems to be ill-equipped for the higher β values.

The expanded reduced model performed well across the entire parameter range. The maximum error is on the same order of magnitude as the minimum error for other methods.

This is expected due to the richer subspace that includes the modes generated by the other methods. (Aside: although it may seem unfair to compare the 10 modes in the expanded model to the 5 in the baseline and extrapolated models, for this problem adding additional modes did not increase the accuracy.)

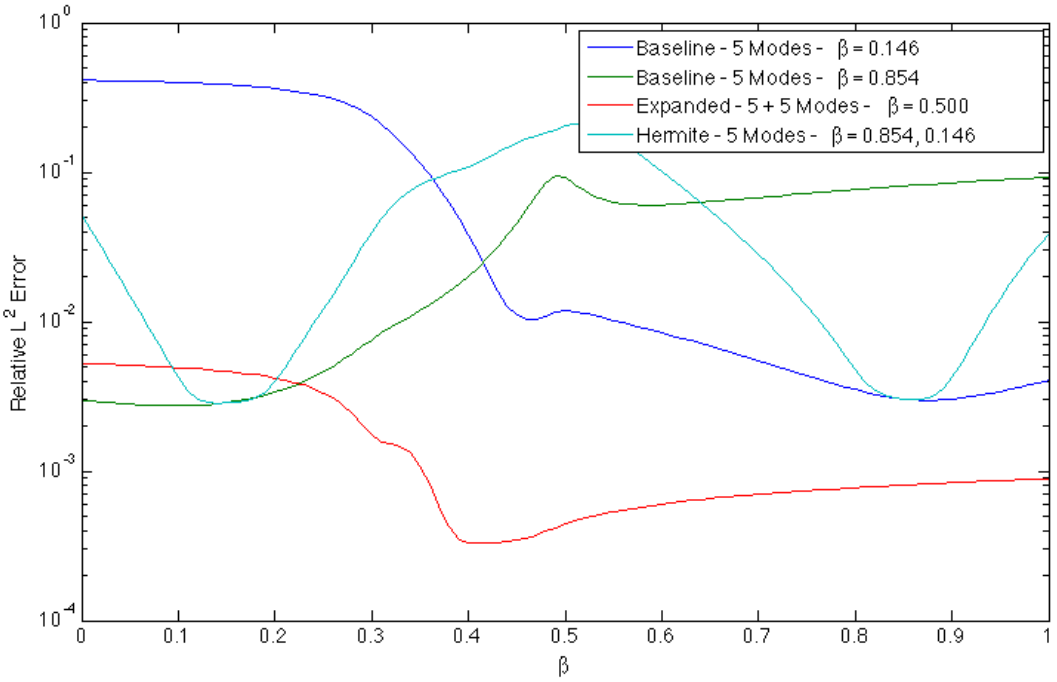

Figure 3.3: Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Reaction-Diffusion

Reduced Model	β(s) Used	Min. Error	Max. Error	Avg. Error
Baseline	0.854	2.9701e-03	4.1278e-01	1.2783e-01
Baseline	0.146	2.7406e-03	9.4108e-02	4.4861e-02
Expanded	0.500	3.2903e-04	5.2406e-03	1.7999e-03
Hermite	0.854, 0.146	2.8264e-03	2.1178e-01	5.3228e-02

Table 3.2: Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Reaction-Diffusion

Figure 3.3 and Table 3.2 demonstrate the performance of the Hermite model compared to the baseline and expanded models. The single expanded model based at $\beta = 0.5$ has been included along with the baselines at the Chebyshev nodes used for the Hermite interpolant. While the Hermite interpolant agrees with the baseline at the interpolation points, it surprisingly performs worse across a large range of β values. It exhibits smaller errors than the baseline generated by the opposite Chebyshev point, but simply switching baselines at any point from $\beta \approx 0.4$ to $\beta \approx 0.65$ would have produced better performance. The expanded model continues to perform uniformly well across the range of parameters.

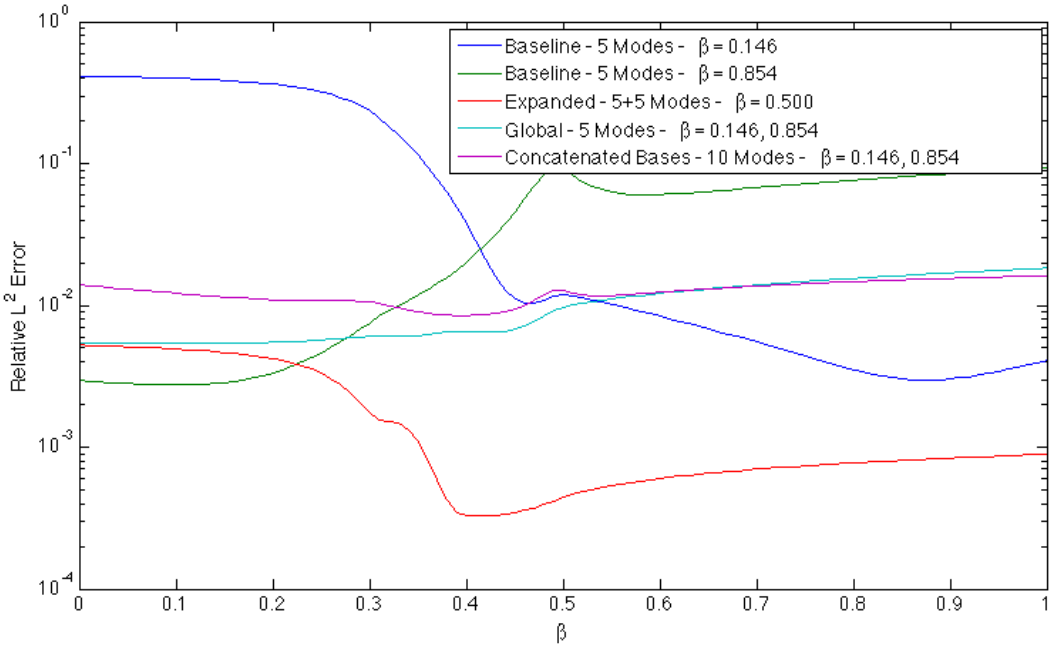

Figure 3.4: Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Reaction-Diffusion

Reduced Model	β(s) Used	Min. Error	Max. Error	Avg. Error
Baseline	0.854	2.9701e-03	4.1278e-01	1.2783e-01
Baseline	0.146	2.7406e-03	9.4108e-02	4.4861e-02
Expanded	0.500	3.2903e-04	5.2406e-03	1.7999e-03
Global	0.146, 0.854	5.4051e-03	1.8336e-02	1.0295e-02
Multiple Baselines	0.146, 0.854	8.4699e-03	1.6179e-02	1.2498e-02

Table 3.3: Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Reaction-Diffusion

Figure 3.4 and Table 3.3 demonstrate the performance of the concatenated and multiple baseline models. The concatenated and multiple baseline models perform very similarly. These models worked well across the entire range of βs considered. While they did not outperform the expanded model, these methods do not require sensitivity analysis and thus would be acceptable methods for when sensitivity information is not available.

3.2 2D Incompressible Navier-Stokes

3.2.1 Incompressible Navier-Stokes Equations

The dimensionless incompressible Navier-Stokes equations are as follows:

$$\frac{\partial U}{\partial t} + U \cdot \nabla U = \nabla \cdot (-pI + \mu(\nabla U + \nabla U^T)) + F \tag{3.2.1}$$

$$\nabla \cdot U = 0 \tag{3.2.2}$$

If $U = \begin{bmatrix} u(x,y,t) \\ v(x,y,t) \end{bmatrix}$, expanding the stress tensor term yields the following system of equations:

$$\begin{aligned} u_t + uu_x + vu_y &= -p_x + \mu\left(v_{xy} + 2u_{xx} + u_{yy}\right) + f^x \\ v_t + uv_x + vv_y &= -p_y + \mu\left(u_{xy} + 2v_{yy} + v_{xx}\right) + f^y \\ 0 &= u_x + v_y \end{aligned} \tag{3.2.3}$$

Note that the incompressiblity condition implies $v_{xy} + 2u_{xx} + u_{yy} = u_{xx} + u_{yy} + u_{xx} + v_{xy} = u_{xx} + u_{yy} + \dfrac{\partial}{\partial x}(u_x + v_y) = u_{xx} + u_{yy}$ and similarly in the v equation. This makes the equation equivalent to

$$\begin{aligned} \frac{\partial U}{\partial t} + U \cdot \nabla U &= -\nabla p + \mu(\triangle U)) + F \\ \nabla \cdot U &= 0 \end{aligned} \tag{3.2.4}$$

However, we will keep the equations as formulated in (3.2.1) because doing so allows for trivial implementation of "stress-free" boundary conditions. We then multiply by a test function and integrate (applying our boundary conditions and integrating by parts) to yield the weak formulation:

$$\begin{aligned} \int u_t\phi_i &= \int p(\phi_i)_x - \mu\left(2\int u_x(\phi_i)_x + \int u_y(\phi_i)_y + \int v_x(\phi_i)_y\right) + \int f^x\phi_i - \int (uu_x + vu_y)\phi_i \\ \int v_t\phi_i &= \int p(\phi_i)_y - \mu\left(2\int v_y(\phi_i)_y + \int v_x(\phi_i)_x + \int u_y(\phi_i)_x\right) + \int f^y\phi_i - \int (uv_x + vv_y)\phi_i \\ 0 &= \int u_x\psi_\ell + \int v_y\psi_\ell \end{aligned} \tag{3.2.5}$$

where (ϕ_i, ψ_ℓ) represent basis functions from a Taylor-Hood element pair for velocity and pressure (e.g. $P_2 - P_1$). Letting $[\hat{M}]_{ij} = \int \phi_i\phi_j, [K_{xy}]_{ij} = \int (\phi_j)_x(\phi_i)_y, [F^x]_i = \int f^x\phi_i, [N^1(u,v)]_i = \int (uu_x + vu_y)\phi_i, [P_x]_{ij} = \int (\phi_i)_x\psi_j$ and others defined analogously, we can write these equations in the following form:

$$\begin{aligned} \begin{bmatrix} \hat{M} & 0 \\ 0 & \hat{M} \end{bmatrix} \begin{bmatrix} \dot{u} \\ v \end{bmatrix} &= -\mu \begin{bmatrix} 2K_{xx} + K_{yy} & K_{xy} \\ K_{xy}^T & K_{xx} + 2K_{yy} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} F^x - N^1(u,v) \\ F^y - N^2(u,v) \end{bmatrix} + \begin{bmatrix} P_x \\ P_y \end{bmatrix} p \\ 0 &= \begin{bmatrix} P_x^T & P_y^T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned} \tag{3.2.6}$$

Let $M = \begin{bmatrix} \hat{M} & 0 \\ 0 & \hat{M} \end{bmatrix}, A = -\mu \begin{bmatrix} 2K_{xx} + K_{yy} & K_{xy} \\ K_{xy}^T & K_{xx} + 2K_{yy} \end{bmatrix}, F(t,U) = \begin{bmatrix} F^x - N^1(u,v) \\ F^y - N^2(u,v) \end{bmatrix}, \hat{P} = \begin{bmatrix} P_x \\ P_y \end{bmatrix}$. We can write this as a *differential-algebraic equation* (DAE) of Hessenberg index 2. That is, we can write it in the following form:

$$\begin{aligned} M\dot{U} &= f(t,U,p) \\ 0 &= g(U) \end{aligned} \tag{3.2.7}$$

Thus, we may use techniques discussed in Section 1.2.5 to solve 3.2.7.

3.2.2 Model

For our model, we use the above equations (3.2.1, 3.2.2) to simulate pipe flow around a cylinder. The computational domain is $\Omega = \{(x,y) : 0 \le x \le 20, -2 \le y \le 2\} \backslash \Big\{(x,y) : \sqrt{x^2+y^2} < 1/2\Big\}$. The domain and a sample triangulation can be seen in Figure 3.5. The left boundary has a parabolic inflow, the right boundary has a stress-free boundary condition and the remaining boundaries (and cylinder) have no-slip boundary conditions. The solution to the Stokes flow for this domain is used as the initial condition.

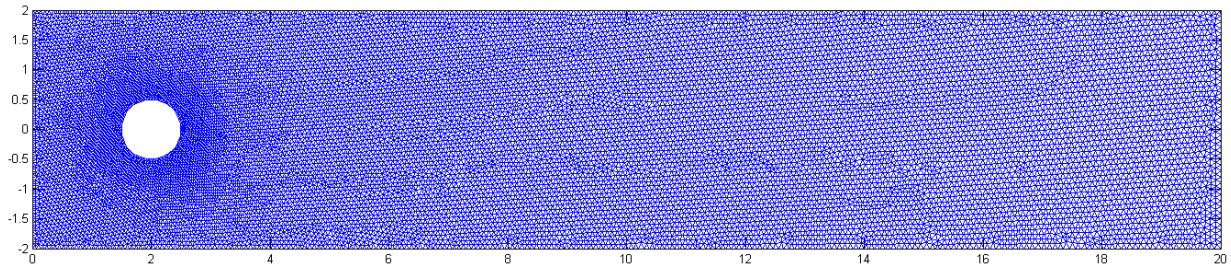

Figure 3.5: Computational Domain and Sample Triangulation

3.2.3 Results

For our first comparison, we will look at a comparison of the baseline, expanded, and extrapolated models. Note that for this section, the methods have been normalized so that each resulting reduced order model has the same dimension: 8. As a point of reference, the original system has $\approx 100,000$ degrees of freedom, so these techniques represent a quite significant reduction.

Similar to the results we saw in Section 3.1.2, the baseline model performs well for the parameter at which it was generated, but performance declines as the parameter changes.

The extrapolated model performs fairly poorly, with an average relative L^2 error of $\approx 28.9\%$. This may reflect the strong non-linear dependence of the solution with respect to the Reynolds number.

The expanded model, despite only having 4 modes from the solution, is able to capture a large portion of the dynamics using the sensitivity information. While the minimum

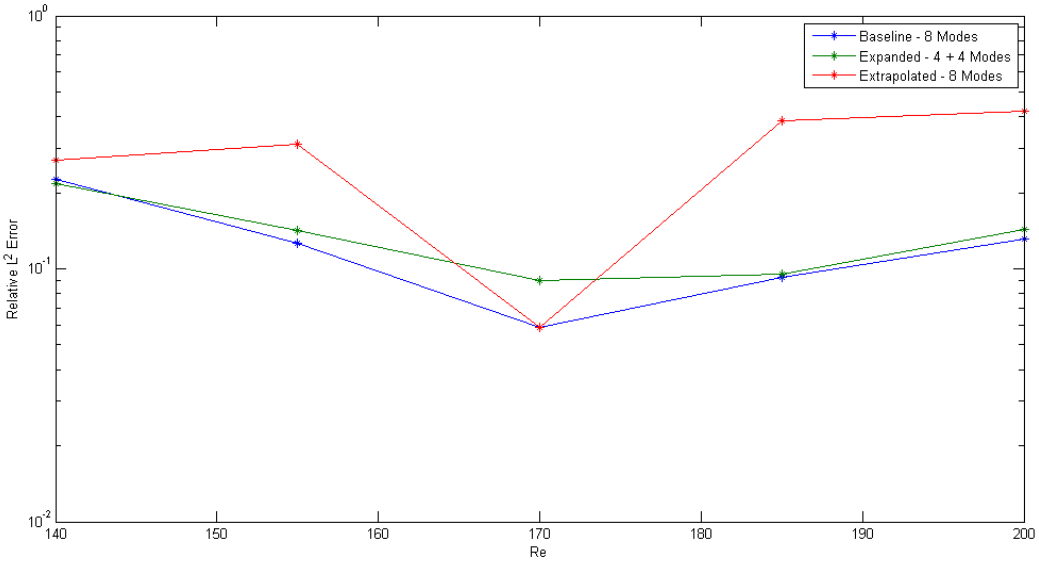

Figure 3.6: Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models for Navier-Stokes

Reduced Model	Re(s) Used	Min. Error	Max. Error	Avg. Error
Baseline	170	5.8411e-02	2.2655e-01	1.2697e-01
Expanded	170	8.9650e-02	2.1745e-01	1.3750e-01
Extrapolated	170	5.8411e-02	4.1918e-01	2.8876e-01

Table 3.4: Relative L^2 Errors for Baseline, Expanded, and Extrapolated Models for Navier-Stokes

error for the expanded model is slightly higher than the baseline, it enjoys fairly consistent performance across the range of Reynolds numbers, yielding an average relative L^2 error of $\approx 12.7\%$.

Reduced Model	Re(s) Used	Min. Error	Max. Error	Avg. Error
Baseline	170	5.8411e-02	2.2655e-01	1.2697e-01
Expanded	170	8.9650e-02	2.1745e-01	1.3750e-01
Hermite	155, 185	4.6915e-02	3.7839e-01	2.3882e-01

Table 3.5: Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Navier-Stokes

In Figure 3.7 and Table 3.5, we can see that the Hermite model again does not enjoy very good performance. At one of the interpolation nodes, $Re = 185$, we see improvement, but the method suffers from low accuracy (or at best approximately the same accuracy) compared to the baseline and expanded models, despite having two sample points.

In Figure 3.8 and Table 3.6, we can see that the global and multiple baseline models perform fairly consistently across the parameter values. However, these methods do not

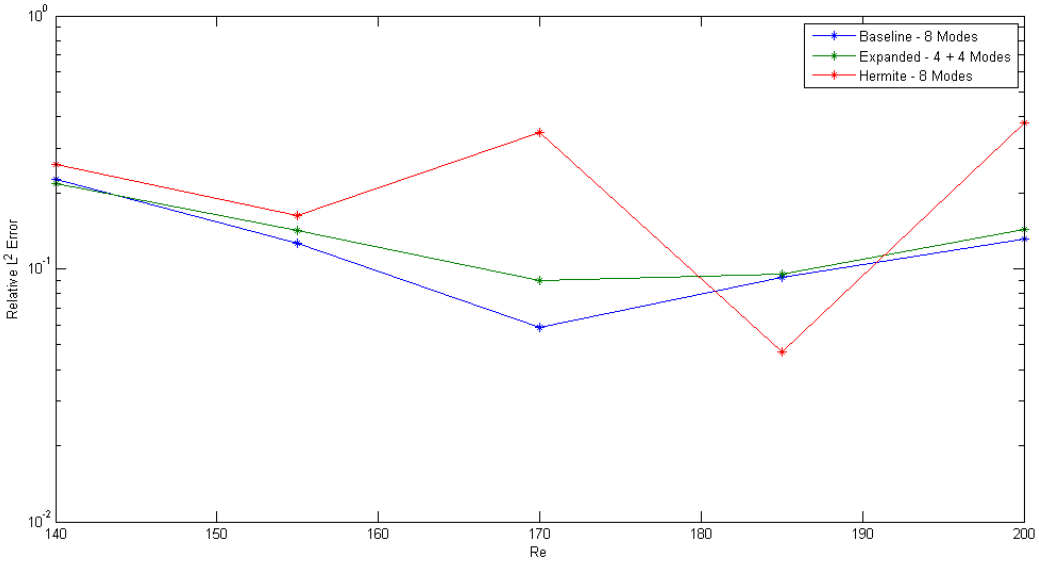

Figure 3.7: Relative L^2 Errors for Baseline, Expanded, and Hermite Models for Navier-Stokes

Reduced Model	Re(s) Used	Min. Error	Max. Error	Avg. Error
Baseline	170	5.8411e-02	2.2655e-01	1.2697e-01
Expanded	170	8.9650e-02	2.1745e-01	1.3750e-01
Global	155, 185	1.1881e-01	3.3851e-01	2.0585e-01
Multiple Baselines	155, 185	1.9274e-01	2.9776e-01	2.5310e-01

Table 3.6: Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Navier-Stokes

perform well, with maximum and average relative errors significantly higher than even the baseline model. Due to the lower number of modes used, this is not entirely unexpected for the multiple baseline model. This can be explained by only the first 4 modes from each sample point being used and these first few modes being similar. Thus, the addition of modes does not contribute much new information.

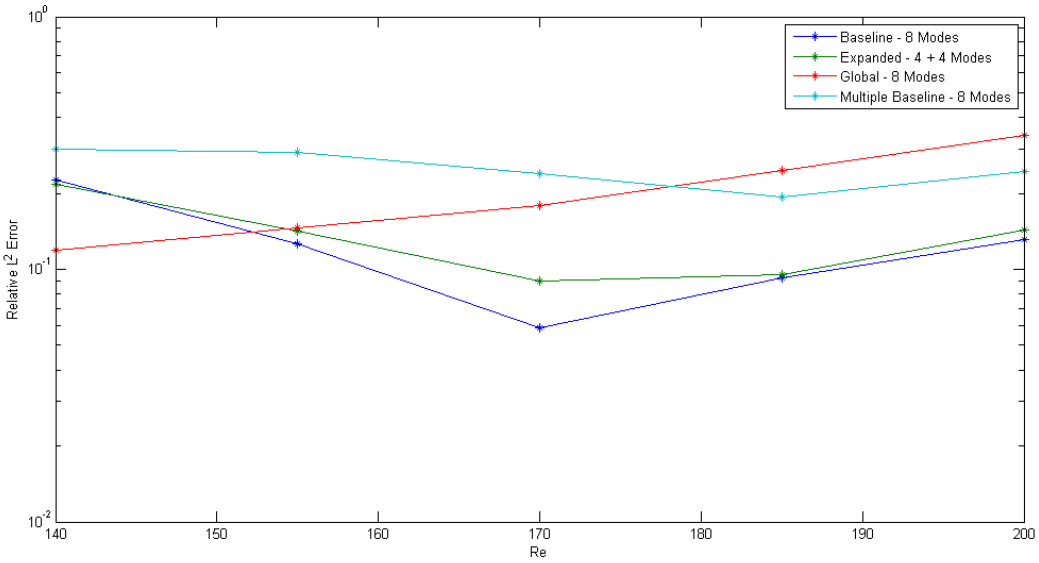

Figure 3.8: Relative L^2 Errors for Baseline, Expanded, Global, and Multiple Baseline Models for Navier-Stokes

Chapter 4

Conclusions

In this thesis, we explored several techniques for extending a reduced order model based on proper orthogonal decomposition to a range of parameter values. The full dimensional models and the corresponding sensitivities were constructed using the finite element method to perform the spatial discretization and solved using ODE/DAE solvers in time.

The baseline model is the simplest model, requiring only a single full model evaluation. However, the lack of derivative information or global information causes the baseline model to perform poorly outside of a narrow range of parameter values. The global model and multiple baseline models require several full model evaluations but do not require any sensitivity evaluations. These methods performed moderately well, but require computing the full model at several points to capture the dynamics across the range of parameters. The extrapolated and Hermite models use full models coupled with the sensitivity equations to improve on the POD basis. However, while these models remained valid for a slightly larger parameter range than the baseline models, their performance begins to degrade considerably outside of this range. Since these methods are based on Taylor series expansions and polynomial interpolation, this is explained by the non-linearity in the change in basis vectors dominating the linear or polynomial approximations. Finally, the expanded model requires only a single full model evaluation coupled with sensitivity equations. However, this method outperforms the previously mentioned techniques (in some cases significantly), even when compared against methods that sample more than one parameter value. Rather than prescribing a particular method of combining the basis vectors and sensitivities (e.g. extrapolated and Hermite models), the expanded model uses a richer subspace of basis vectors to allow for more optimal reduced order models. This performance increase is not unexpected since, in particular, the Hermite and extrapolated models can be represented in the subspace generated by the expanded model. While the multiple baseline and global models exhibit slightly better performance in some regions of the parameter, this is offset by the need for several full model simulations, which are typically more expensive than a single full model coupled with sensitivities.

These results give inspiration for future studies such as:

- Extending the expanded bases to include multiple parameter samples
- Performing model reduction across more than one parameter (perhaps utilizing sparse grid techniques)
- Utilizing sensitivity information to guide parameter selection
- Attempting to bound the error between the expanded model and the optimal projection of the solution onto that model

- Exploring properties of solution through reduced model

Bibliography

[1] Uri M. Ascher and Linda R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.

[2] B.T. Dickinson. Nonlinear Model Reduction Using the Group Proper Orthogonal Decomposition Method. Master's thesis, Oregon State University, May 2007.

[3] A. Hay, J. Borggaard, and D. Pelletier. Local Improvements to Reduced-Order Models Using Sensitivity Analysis of the Proper Orthogonal Decompositions. *Journel of Fluid Mechanics*, 629:41–72, 2009.

[4] C. Jarvis. Reduced Order Model Study of Burgers' Equation using Proper Orthogonal Decomposition. Master's thesis, Virginia Polytechnic Institute and State University, February 2012.

[5] V. John and J. Rang. Adaptive time step control for the incompressible Navier-Stokes equations. *Computational Methods in Applied Mechanical Engineering*, (199):514–524, 2010.

[6] A. Kværnø. Singly Diagonally Implicit Runge-Kutta Methods with an Explicit First Stage. *BIT Numerical Mathematics*, 44(3):489–502, 2004.

[7] L.M. Skvortsov. Diagonally Implicit Runge-Kutta Methods for Differential Algebraic Equations of Indices Two and Three. *Computational Mathematics and Mathematical Physics*, 50(6):993–1005, 2010.

[8] S. Volkwein. Proper Orthogonal Decomposition (POD) for Nonlinear Dynamical Systems. Dutch Institute of Systems and Control Summerschool, 2005.

[9] S. Volkwein. Model Reduction Using Proper Orthogonal Decomposition. 2008.

[10] Y.B. Zeldovich and D. A. Frank-Kamenetsky. A Theory of Thermal Propogation of Flame. *Acta Physicochim*, 9:341–350, 1938.