Improved Reduced Order Modeling Strategies for Coupled and Parametric Systems

Daniel Joseph Sutton

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

Master of Science
in
Mechanical Engineering

Jeffrey T. Borggaard, Co-Chair
Daniel J. Inman, Co-Chair
Harry H. Robertshaw

August 16, 2005
Blacksburg, Virginia

Keywords: Proper Orthogonal Decomposition, Reduced Order Modeling,
Domain Decomposition, Coupled Systems, Feedback Control

Improved Reduced Order Modeling Strategies for Coupled and Parametric Systems

Daniel J. Sutton

(ABSTRACT)

This thesis uses Proper Orthogonal Decomposition to model parametric and coupled systems. First, Proper Orthogonal Decomposition and its properties are introduced as well as how to numerically compute the decomposition. Next, a test case was used to show how well POD can be used to simulate and control a system. Finally, techniques for modeling a parametric system over a given range and a coupled system split into subdomains were explored, as well as numerical results.

Dedication

for Diddy

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reduced order models have a number of practical uses including building feedback control laws for complex systems and fast predictions. A popular approach is to use the Proper Orthogonal Decomposition (POD) [9], also known as the Karhunen-Loève Expansion [18]. This decomposition has a number of different origins due to its independent formulation in diverse fields, such as statistics (Karhunen-Loève Expansion)[18, 21], controls (Principle Component Analysis)[24], fluids (Proper Orthogonal Decomposition)[23, 27], and weather forecasting (Empirical Eigenfunctions)[22]. These approaches have seen a wide range of applications including the design of feedback control systems [2, 3, 7, 8, 15, 19, 25], dynamic compensators [6, 11], optimal design [16, 17], and identification of fundamental mechanisms in physical phenomena (e.g., Langmuir circulations on the ocean surface [20]). Using a few basis vectors from this decomposition, one can construct very good low dimensional dynamical systems (usually by Galerkin projection) to model complex phenomena such as flow transition [2], turbulence [9, 27], and large circuit simulations [4].

This decomposition is derived from a set of simulation data. The quality of the resulting reduced order model depends on how well this data can represent the dynamics. Strategies for producing good data (known as an "input collection") are needed. Currently, this is an "art" guided by physical intuition and in the control setting, by a closed-loop response [7]. Methods which have been tried either for parametric systems or in more specific but generalizable situations involved an approach to pick the best parameters from an already given larger set [5], discretizing the parameter range and weighting the parameters [26], creating a larger parameter set from a smaller

one by interpolation [10], and picking your parameters randomly from the range and then using some kind of convergence scheme to know when you have enough snapshots. In this monograph we study a systematic approach to guide the generation of a suitable input collection for parametric systems based on numerically approximating an integral.

In many cases, it is desirable to produce a reduced order model for a coupled system. These systems, such as aero-elasticity or combustion, are modeled using two (or more) separate systems where outputs of either system are inputs to the other. In some cases, reduced order models may be available for the subsystems, or software may only be available for the individual certain classes of coupled systems. These are systems where the coupling occurs through a spatial boundary. This covers the case of domain decomposition and may lead to algorithms which have good parallel efficiency. To our knowledge POD has not been tried on subsystems which are then coupled together with the desire of having the POD modes accurately predict the entire system, and without ever having simulation data from the entire system, which is done in this monograph.

This work contributes to the scientific community by demonstrating that POD can be used equally well on systems which are dependent on parameters without drastically increasing the number of necessary POD modes. Also it shows that POD can be used on coupled systems to produce a model for the entire system without every having to simulate the computationally expensive entire system.

The remainder of this thesis is organized as follows. In Chapter 2, we give an overview of POD and provide motivation for our study. An example of POD for prediction as well as for constructing feedback control laws is developed in Chapter 3. We then present our main results in Chapter 4 in which we describe our strategy for choosing a good input collection for systems with parameters. Our initial results led us to try natural extensions which placed more samples in regions (in parameter space) where the dynamics of the original system are more difficult to predict. Also, we outline a strategy to treat coupled systems. A numerical example shows that this can be effective. Finally, in Chapter 5, we present our conclusions and outline some areas for future work.

Chapter 2

Background

2.1 Motivation of Reduced Order Modeling

Many of today's problems in engineering and science are described by partial differential equations (PDE's), of which only the most simple have known analytic solutions. In order to have any kind of solution it is thus necessary to numerically approximate the solution. However; partial differential equations are infinite-dimensional and hence cannot be solved exactly, no matter how much time is spent or how large the computing power available.

To circumvent this handicap, the partial differential equation is projected onto a finite-dimensional subspace. Although this subspace cannot represent the PDE exactly, by enlarging the subspace the PDE can be solved within any error bound, no matter how small. Reduced order modeling is precisely this, using some suitable subspace on which to solve the PDE, which must also be simplified, usually into a finite number of ordinary differential equations.

Two of the many different reduced order modeling methods are finite differences and finite elements. The former uses previously chosen points in the domain of interest and an approximation of the PDE (usually obtained through Taylor series expansions) to obtain ordinary differential equations. These equations are then prescribed some suitable initial values, and the differential equations are solved numerically. The error from this method is controlled by the number of points used and their placement, the order of the approximation to the PDE, and the order of the differential equation numerical solver. However; no information about the PDE is used in picking any of these parameters, and there is no guarantee that, for example, the

points are chosen in an optimal way.

The other method mentioned, finite elements, splits the domain into pairwise disjoint subdomains, whose union is the original domain. On these subdomains the PDE is assumed to be approximated by a linear combination of previously chosen basis functions. Next, a linear system of equations is obtained, usually by multiplying by a test function and integrating over the entire domain. This system is then solved numerically as above. In this method the error is controlled by the subdomains chosen, basis functions, test functions, and the order of the differential equation numerical solver. Again, no information about the PDE is used in choosing these parameters, and the model is not optimized.

Although many problems can be solved by either of the two methods above to suitable tolerances in an acceptable amount of time on today's computers, the harder problems, such as Navier-Stokes, cannot be solved well enough by today's methods. Because of this it is necessary to improve today's methods; one way to do this is to use information from the PDE, such as is done by the Proper Orthogonal Decomposition (POD).

2.2 Motivation of Coupled Systems

The physical systems considered may not be very large, and sometimes may be normalized to be inside a square or cube of unit length. However; to obtain a solution to within specified tolerances it is usually necessary to specify thousands of points in the finite difference method or thousands of elements in the finite element method. Each of these may represent multiple variables to be solved, and in setting up the system of equations to be solved, the matrix involved is usually proportional to n^2, where n is the number of unknowns. It is therefore easily possible to have a matrix with over one million entries. One way to reduce the number of computations necessary is to split up the domain into disjoint subdomains and couple the subdomains by passing the boundary information across the boundaries of adjacent subdomains, i.e. prescribe boundary conditions on a subdomain to match its neighbor, so that the solution is continuous. If the amount of information passed across the boundaries is large enough, the tolerance of the coupled system may still be relatively low. The main advantage comes from the decrease in computational time. Consider, for example, splitting up the domain into two equally sized subdomains. Before the computational effort was proportional

to n^2, but now each subdomain only has $\frac{n}{2}$ unknowns, for a computational effort proportional to $\frac{n^2}{4}$, for a total effort over the entire domain proportional to $\frac{n^2}{2}$, or half the effort.

Another reason to use coupled systems is when the entire system cannot be modeled with the same equations. It is usually obvious that this is necessary, for example with a vibrating beam that is surrounded by a fluid. The beam is forced by the pressure from the fluid, and the fluid's velocity must agree with the velocity of the beam on the boundary of the beam. Also these two systems adhere to different PDE's inside their respective domains and hence only affect each other on their mutual boundaries.

2.3 Introduction to the Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition seeks to use information from a solution to the PDE, such as a numerical solution or experimental data, to find N functions, called POD modes, which will best represent the solution of the partial differential equation in a time-averaged sense. To be more precise, if $S(\vec{x}, t)$ is the solution defined on some spatial domain Ω and a temporal interval of length T, the first POD mode $\phi(\vec{x})$ is defined to be

$$\max_{\phi} \left\{ \frac{1}{T} \int_T \frac{\langle S(\vec{x}, t), \phi(\vec{x}) \rangle^2}{\langle \phi(\vec{x}), \phi(\vec{x}) \rangle} \, dt \right\} \tag{2.1}$$

Here, $\langle , \rangle$ denotes the L^2 inner product over Ω; the second POD mode can be computed by using equation (2.1) with $S(\vec{x}, t)$ replaced by $S(\vec{x}, t) - \langle S(\vec{x}, t), \phi_1(x) \rangle \phi_1(\vec{x})$, and subsequent POD modes can be found iteratively in a similar manner. From this it can be seen that the first POD mode is the direction in state-space on which the solution $S(\vec{x}, t)$ is largest, in a time-averaged sense. Correspondingly, another way of viewing the first POD mode is

$$\min_{\phi, \psi} \left\{ \frac{1}{T} \int_T \| S(\vec{x}, t) - \psi(t) \phi(\vec{x}) \|^2 \, dt : \| \phi(\vec{x}) \| = 1 \right\} \tag{2.2}$$

The solution is then assumed to be a linear combination of the POD modes. One of the consequences of this formulation is that given the first N POD modes, these functions give the best representation of the solution out of any set of N orthogonal functions, in a time-averaged sense. In other

words, if S_N is the approximation of the first N POD modes to the solution, i.e.

$$S_N(\vec{x},t) = \sum_{i=1}^{N} \psi_i(t)\phi_i(\vec{x}) \tag{2.3}$$

where $\psi_i(t)$ is the projection of the solution onto $\phi_i(\vec{x})$, or

$$\psi_i(t) = \langle S(\vec{x},t), \phi_i(\vec{x}) \rangle \tag{2.4}$$

then

$$\frac{1}{T}\int_T \|S(\vec{x},t) - S_N(\vec{x},t)\|^2 \, dt \leq \frac{1}{T}\int_T \left\| S(\vec{x},t) - \sum_{i=1}^{N} \alpha_i(t) f_i(\vec{x}) \right\|^2 dt \tag{2.5}$$

where $\{f_i(\vec{x})\}$ is any set of N orthonormal functions.

Viewing $S_N(\vec{x},t)$ as the projection of $S(\vec{x},t)$ onto the N-dimensional subspace with basis given by $\{\phi_k\}$, $S_N(\vec{x},t) = P(S(\vec{x},t))$, where P is the operator that projects onto the subspace. Due to the identity

$$\|S\|^2 = \|S - P(S)\|^2 + \|P(S)\|^2 \tag{2.6}$$

which is true for any orthogonal projection, it is also true that

$$\min_{S_N} \left\{ \frac{1}{T}\int_T \|S(\vec{x},t) - S_N(\vec{x},t)\|^2 \, dt \right\} = \max_{S_N} \left\{ \frac{1}{T}\int_T \|S_N(\vec{x},t)\|^2 \, dt \right\} \tag{2.7}$$

where the max and min are taken over the set of all N^{th}-order orthonormal approximations.

Writing out the S_N in terms of the basis functions:

$$\begin{aligned}
&\frac{1}{T}\int_T \|S_N(\vec{x},t)\|^2\,dt \\
&= \frac{1}{T}\int_T \langle S_N(\vec{x},t), S_N(\vec{x},t)\rangle\,dt \\
&= \frac{1}{T}\int_T \left\langle \sum_{i=1}^{N}\langle S(\vec{x},t),\phi_i(\vec{x})\rangle\phi_i(\vec{x}), \sum_{j=1}^{N}\langle S(\vec{x},t),\phi_j(\vec{x})\rangle\phi_j(\vec{x}) \right\rangle dt \\
&= \frac{1}{T}\int_T \sum_{i=1}^{N}\langle S(\vec{x},t),\phi_i(\vec{x})\rangle \left\langle \phi_i(\vec{x}), \sum_{j=1}^{N}\langle S(\vec{x},t),\phi_j(\vec{x})\rangle\phi_j(\vec{x}) \right\rangle dt \\
&= \frac{1}{T}\int_T \sum_{i=1}^{N}\langle S(\vec{x},t),\phi_i(\vec{x})\rangle\langle S(\vec{x},t),\phi_i(\vec{x})\rangle\,dt \\
&= \sum_{i=1}^{N}\frac{1}{T}\int_T \langle S(\vec{x},t),\phi_i(\vec{x})\rangle^2\,dt \\
&= \sum_{i=1}^{N} E(\langle S(\vec{x},t),\phi_i(\vec{x})\rangle^2)
\end{aligned} \tag{2.8}$$

where $E(\cdot)$ is defined by

$$E(f) = \frac{1}{T}\int_T f\,dt \tag{2.9}$$

and $E(\langle S(\vec{x},t),\phi_i(\vec{x})\rangle^2)$ is a measure of the "energy" of the solution in the i^{th} POD mode. Now we have a variational problem: Maximize

$$\sum_{i=1}^{N} E(\langle S(\vec{x},t),\phi_i(\vec{x})\rangle^2) \tag{2.10}$$

subject to the constraints $\|\phi_i(\vec{x})\| = 1$. Therefore the POD modes are the extremals of the function

$$J[\phi(\vec{x})] = E(\langle S(\vec{x},t),\phi(\vec{x})\rangle^2) - \lambda(\|\phi(\vec{x})\|^2 - 1) \tag{2.11}$$

A necessary condition for a function to be an extremal is that the functional derivative must vanish for all variations $\phi(\vec{x}) + \epsilon\psi(\vec{x})$, $\epsilon \in \mathbb{R}$:

$$\frac{d}{d\epsilon}J[\phi + \epsilon\psi]\bigg|_{\epsilon=0} = 0 \tag{2.12}$$

Expanding (2.12) we obtain:

$$\begin{aligned}
&\frac{d}{d\epsilon}\left(\begin{array}{l}\frac{1}{T}\int_T\int_\Omega S(\vec{x},t)(\phi(\vec{x})+\epsilon\psi(\vec{x}))\,dx\int_\Omega S(\vec{y},t)(\phi(\vec{y})+\epsilon\psi(\vec{y}))\,dy\,dt\\ -\lambda\left(\int_\Omega(\phi^2(\vec{x})+2\epsilon\phi(\vec{x})\psi(\vec{x})+\epsilon^2\psi^2(\vec{x}))\,dx-1\right)\end{array}\right)\Bigg|_{\epsilon=0}\\
&\quad\begin{array}{l}\frac{1}{T}\int_T\int_\Omega S(\vec{x},t)(\phi(\vec{x})+\epsilon\psi(\vec{x}))\,dx\int_\Omega S(\vec{y},t)\psi(\vec{y})\,dy\\ +\int_\Omega S(\vec{y},t)(\phi(\vec{y})+\epsilon\psi(\vec{y}))\,dy\int_\Omega S(\vec{x},t)\psi(\vec{x})\,dx\,dt\\ -\lambda\int_\Omega 2\phi(\vec{x})\psi(\vec{x})+2\epsilon\psi^2(\vec{x})\,dx\end{array}\Bigg|_{\epsilon=0}\\
=\;&\frac{1}{T}\int_T\int_\Omega S(\vec{x},t)\phi(\vec{x})\,dx\int_\Omega S(\vec{y},t)\psi(\vec{y})\,dy\\
&+\int_\Omega S(\vec{y},t)\phi(\vec{y})\,dy\int_\Omega S(\vec{x},t)\psi(\vec{x})\,dx\,dt\\
&-\lambda\int_\Omega 2\phi(\vec{x})\psi(\vec{x})\,dx\Rightarrow\\
&\frac{1}{T}\int_T\int_\Omega S(\vec{y},t)\phi(\vec{y})\,dy\int_\Omega S(\vec{x},t)\psi(\vec{x})\,dx\,dt-\lambda\int_\Omega\phi(\vec{x})\psi(\vec{x})\,dx=0
\end{aligned}\tag{2.13}$$

Changing the order of integration produces

$$\int_\Omega\left[\int_\Omega\frac{1}{T}\int_T S(\vec{x},t)S(\vec{y},t)\phi(\vec{y})\psi(\vec{x})\,dt\,dy-\lambda\phi(\vec{x})\psi(\vec{x})\right]dx=0\tag{2.14}$$

Since the variation $\psi(\vec{x})$ is arbitrary, this implies that the integrand of the outermost integral is identically zero, or

$$\begin{aligned}
&\int_\Omega\frac{1}{T}\int_T S(\vec{x},t)S(\vec{y},t)\phi(\vec{y})\psi(\vec{x})\,dt\,dy=\lambda\phi(\vec{x})\psi(\vec{x})\\
&=\psi(\vec{x})\int_\Omega\frac{1}{T}\int_T S(\vec{x},t)S(\vec{y},t)\phi(\vec{y})\,dt\,dy\Rightarrow\\
&\int_\Omega\frac{1}{T}\int_T S(\vec{x},t)S(\vec{y},t)\phi(\vec{y})\,dt\,dy=\int_\Omega E(S(\vec{x},t)S(\vec{y},t))\phi(\vec{y})\,dy\\
&=\int_\Omega R(\vec{x},\vec{y})\phi(\vec{y})\,dy=\lambda\phi(\vec{x})
\end{aligned}\tag{2.15}$$

Therefore the POD modes are the eigenfunctions of the integral operator with kernel $R(\vec{x},\vec{y})$ given in (2.15), and since $R(\vec{x},\vec{y})$ is self-adjoint, the eigenvalues are real and the eigenfunctions are orthogonal, and hence can be chosen to be orthonormal. Also note that by taking the inner product of (2.15) with ϕ we obtain

$$\begin{aligned}\lambda &= \langle\lambda\phi,\phi\rangle = \left\langle \int_\Omega R(\vec{x},\vec{y})\phi(\vec{y})\,dy, \phi(\vec{x})\right\rangle \\ &= \int_\Omega\int_\Omega \frac{1}{T}\int_T S(\vec{x},t)S(\vec{y},t)\,dt\phi(\vec{y})\,dy\phi(\vec{x})\,dx \\ &= \frac{1}{T}\int_T\int_\Omega\int_\Omega S(\vec{x},t)(S(\vec{y},t)\phi(\vec{y})\phi(\vec{x})\,dy\,dx\,dt \\ &= \frac{1}{T}\int_T\int_\Omega S(\vec{x},t)\phi(\vec{x})\,dx\int_\Omega S(\vec{y},t)\phi(\vec{y})\,dy\,dt = E(\langle S(\vec{x},t),\phi(\vec{x})\rangle^2)\end{aligned} \tag{2.16}$$

so the eigenvalue is a measure of how much "energy" is captured by the corresponding eigenfunction, or POD mode.

2.4 Computation of the Proper Orthogonal Decomposition

In most applications a known solution is not available, and therefore there is no $R(\vec{x},\vec{y})$ that can be used to solve (2.15). Instead, an approximation to $R(\vec{x},\vec{y})$, $\tilde{R}(\vec{x},\vec{y})$, can be obtained by using the solution values at different points, or snapshots, from an approximation to the PDE:

$$R(\vec{x},\vec{y}) = \frac{1}{T}\int_T S(\vec{x},t)S(\vec{y},t)\,dt \approx \frac{1}{N}\sum_{i=1}^{N} u(\vec{x},t_i)u(\vec{y},t_i) = \tilde{R}(\vec{x},\vec{y}) \tag{2.17}$$

where $u(\vec{x},t_i)$ is the approximation to the PDE at t_i and N is the number of snapshots, or the number of times at which the solution is known, assumed to be spaced at equal intervals.

Plugging our approximation $\tilde{R}(\vec{x},\vec{y})$ into (2.15) we obtain

$$\begin{aligned}\int_\Omega \tilde{R}(\vec{x},\vec{y})\phi(\vec{y})\,dy &= \int_\Omega \frac{1}{N}\sum_{i=1}^{N} u(\vec{x},t_i)u(\vec{y},t_i)\phi(\vec{y})\,dy \\ &= \frac{1}{N}\sum_{i=1}^{N} u(\vec{x},t_i)\int_\Omega u(\vec{y},t_i)\phi(\vec{y})\,dy = \sum_{i=1}^{N}\alpha_i u(\vec{x},t_i)\end{aligned} \tag{2.18}$$

According to (2.15) this is equal to $\lambda\phi(\vec{x})$, which implies that the POD modes are linear combinations of the snapshots, or

$$\phi(\vec{x}) = \sum_{i=1}^{N} a_i u(\vec{x}, t_i) \tag{2.19}$$

Returning to (2.15) and plugging in (2.19) for the POD mode:

$$\begin{aligned} &\int_\Omega \frac{1}{N} \sum_{i=1}^{N} u(\vec{x}, t_i) u(\vec{y}, t_i) \sum_{j=1}^{N} a_j u(\vec{y}, t_j)\, dy \\ &= \frac{1}{N} \sum_{i=1}^{N} u(\vec{x}, t_i) \sum_{j=1}^{N} a_j \int_\Omega u(\vec{y}, t_i) u(\vec{y}, t_j)\, dy = \lambda \sum_{k=1}^{N} a_k u(\vec{x}, t_k) \end{aligned} \tag{2.20}$$

This can be written in the form $(Ca)^T U = \lambda a^T U$ where U is the matrix with components $U_{i,j} = u(x_i, t_j)$ where $x_i, i = 1, 2, ..., m$ is a discretization of Ω, and [14, 7]

$$C_{ij} = \frac{1}{N} \int_\Omega u(\vec{x}, t_i) u(\vec{x}, t_j)\, dx \tag{2.21}$$

and if the snapshots are linearly independent, i.e. U has maximal rank, then this is equivalent to solving $Ca = \lambda a$, a matrix eigenvector problem. Substituting the components of each eigenvector as the constants into (2.19) gives the N POD modes, ranked according to the size of their corresponding eigenvalue.

Another method of computing the POD Modes is by approximating the POD modes by an M^{th} order approximation: $\phi(\vec{x}) \approx \phi^M(\vec{x}) = \sum_{i=1}^{M} \phi_i h_i(\vec{x})$ where $h_i(\vec{x})$ is an M^{th} order basis to approximate functions on Ω. Using this basis the approximate solution to the PDE can be projected onto the basis by $u(\vec{x}, t) \approx u^M(\vec{x}, t) = \sum_{i=1}^{M} u_i(t) h_i(\vec{x})$. Similarly to (2.17)

$$\begin{aligned} R(\vec{x}, \vec{y}) &= \frac{1}{T} \int_T S(\vec{x}, t) S(\vec{y}, t)\, dt \approx \frac{1}{N} \sum_{k=1}^{N} u(\vec{x}, t_k) u(\vec{y}, t_k) \\ &\approx \sum_{k=1}^{N} \sum_{i=1}^{M} u_i(t_k) h_i(\vec{x}) \sum_{j=1}^{M} u_j(t_k) h_j(\vec{y}) \end{aligned} \tag{2.22}$$

and therefore approximating $\int_\Omega R(\vec{x}, \vec{y}) \phi(\vec{y})$ and $\lambda\phi(\vec{x})$ by vectors from evaluation at the points $x_i, i = 1, 2, ...M$, assumed to be equally spaced, equation

(2.15) becomes

$$HUU^T H^T H\phi = \lambda H\phi \tag{2.23}$$

where $U_{i,j} = u_i(t_j)$, $H_{i,j} = h_j(x_i)$ and the i^{th} component of the vector ϕ is ϕ_i; this is again an eigenvector problem (for $H\phi$, which is the vector approximation to $\phi(\vec{x})$).

Chapter 3

Demonstration of the Usefulness of POD: A Control Application

To show some of the applications of using Proper Orthogonal Decomposition, a test problem was investigated. This involved numerically solving a PDE on a domain with an obstacle in order to get an approximation to $S(\vec{x}, t)$, and using this to derive the POD modes. Then the reduced order model from the POD modes was used to control the full order model by forcing the PDE on the boundary of the obstacle and its effectiveness was studied.

3.1 Problem Description

In this section, we develop a partial differential equation model for the system under consideration. To that end, let $\Omega_1 \subset \mathbb{R}^2$ be the open rectangle given by $(a, b) \times (c, d)$. Let $\Omega_2 \subset \mathbb{R}^2$ be the rectangle given by $[a_1, a_2] \times [b_1, b_2]$ where $a < a_1 < a_2 < b$ and $c < b_1 < b_2 < d$, i.e., $\Omega_2 \subset \Omega_1$. The problem domain, Ω, is given by $\Omega = \Omega_1 \setminus \Omega_2$. In this configuration, Ω_2 is the obstacle on which we implement Dirichlet boundary control.

The dynamics of the system are given by the two-dimensional Burgers' equation

$$\frac{\partial}{\partial t} w(x, y, t) + (K_1, K_2) \cdot \frac{1}{2} \nabla w^2(x, y, t) = \frac{1}{Re} \Delta w(x, y, t) \tag{3.1}$$

(the ∇ and Δ operators are with respect to the spatial variables only) for $t > 0$ and $(x, y) \in \Omega$. Behaviors described by Burgers' equation include shock formation, shock propagation, and rarefaction wave formation. In (3.1), K_1 and K_2 are constants used to scale the nonlinear terms; K_2 will be zero for simplicity, and K_1 will henceforth be known as K. The quantity Re is a nonnegative constant, and is analogous to the Reynolds number in the Navier-Stokes equations; it will be equal to 300 in all simulations.

In order to fully specify the model, we need to enumerate conditions on $\partial\Omega_1$ and $\partial\Omega_2$, as well as an initial condition. As we are interested in implementing control on $\partial\Omega_2$, denote the sides of $\partial\Omega_2$ as

$$\begin{aligned} \Gamma_1 &= \{(x, b_1) : a_1 \leq x \leq a_2\}, \qquad \Gamma_2 = \{(a_1, y) : b_1 \leq y \leq b_2\}, \\ \Gamma_3 &= \{(x, b_2) : a_1 \leq x \leq a_2\}, \qquad \Gamma_4 = \{(a_2, y) : b_1 \leq y \leq b_2\}. \end{aligned} \tag{3.2}$$

For simplicity, we assume that the controls on $\partial\Omega_2$ are separable, i.e., they are the product of a function of time and a function of the spatial variables. With this assumption, we specify conditions on $\partial\Omega_2$ of the form

$$\begin{aligned} w(\Gamma_1, t) &= u_1(t)\Psi_1(x), \qquad w(\Gamma_2, t) = u_2(t)\Psi_2(y), \\ w(\Gamma_3, t) &= u_3(t)\Psi_3(x), \qquad w(\Gamma_4, t) = u_4(t)\Psi_4(y). \end{aligned} \tag{3.3}$$

Only $u_1(t)$ and $u_3(t)$ will be nonzero; the function $\Psi_i(\cdot)$ is a function describing the influence of the i^{th} control on Γ_i.

To complete the model, we specify conditions on $\partial\Omega_1$. Analogous to the no-slip conditions enforced in many flow configurations, we specify that

$$w(x, c, t) = 0, \qquad w(x, d, t) = 0 \tag{3.4}$$

also,

$$w(a, y, t) = f(y), \qquad \frac{\partial}{\partial x} w(b, y, t) = 0 \tag{3.5}$$

In (3.5), $f(y)$ is a parabolic inlet condition on the left, and is analogous to the inflow condition specified in many channel flow problems. To complete the model, we specify an initial condition of the form

$$w(x, y, 0) = w_0(x, y) \in L^2(\Omega) \tag{3.6}$$

3.2 Control Law Design

By substituting the approximation for $w(x, y, t)$ from the POD modes (of the form in (2.3)) into (3.1), taking an inner product with $\phi_j(x, y)$, posing the equation weakly, and substituting forward and backward difference approximations for the partial derivatives, equations for $\dot{\psi}_j(t)$ are obtained of the form $\dot{\psi}(t) = A\psi + Bu(t) + F + N(\psi, t)$ [13] where

$$A_{i,j} = -\frac{1}{Re \cdot h}\left(\begin{array}{l}\displaystyle\int_{a_1}^{a_2} (\phi_j(x, b_1 - h)\phi_i(x, b_1) + \phi_j(x, b_2 + h)\phi_i(x, b_2))\, dx + \\ \displaystyle\int_c^d \phi_j(a + h, y)\phi(a, y)\, dy + h\int_\Omega \nabla\phi_i \cdot \nabla\phi_j\, d\Omega\end{array}\right), \tag{3.7}$$

$$B_i = \left[\int_{a_1}^{a_2} \frac{\Psi_1(x)\phi_i(x, b_1)}{Re \cdot h}\, dx \quad \int_{a_1}^{a_2} \frac{\Psi_3(x)\phi_i(x, b_2)}{Re \cdot h}\, dx\right], \tag{3.8}$$

$$F_i = \int_c^d \frac{f(y)}{Re \cdot h}\phi_i(a, y)\, dy, \text{ and} \tag{3.9}$$

$$N(\psi, t)_i = \frac{K}{2}\int_\Omega w^2 \frac{\partial}{\partial x}\phi_i\, d\Omega + \frac{K}{2}\int_c^d \left(f^2(y)\phi_i(a, y) - w^2(b, y)\phi_i(b, y)\right)\, dy \tag{3.10}$$

The initial conditions are found by projecting the initial condition for $w(x, y, t)$ onto the POD basis.

In this subsection we are concerned with constructing boundary feedback control laws for finite dimensional system models with state-space equations of the form

$$\dot{\alpha} = A\alpha + Bu(t) + F + N(\alpha, t),\ t > 0, \tag{3.11}$$

$$\alpha(0) = \alpha_0(\vec{x}) \tag{3.12}$$

We consider the tracking control problem for (3.11)-(3.12). A fixed reference signal $z(\vec{x})$ is specified, and we desire that solutions of (3.11)-(3.12) track $z(\vec{x})$ as time evolves. As the tracking signal is time-invariant, the dynamics of the system under tracking control are given by

$$\begin{aligned}\begin{bmatrix}\dot{\alpha} \\ \dot{z}\end{bmatrix} &= \begin{bmatrix}A & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\alpha \\ z\end{bmatrix} + \begin{bmatrix}B \\ 0\end{bmatrix}u + \begin{bmatrix}F \\ 0\end{bmatrix} + \begin{bmatrix}N \\ 0\end{bmatrix} \\ &= \bar{A}X + \bar{B}u + \bar{F} + \bar{N}\end{aligned} \tag{3.13}$$

where we have defined the augmented state X as $X = [\alpha\ z]^T$. The initial data of the system under tracking control is given by

$$X_0 = \begin{bmatrix} \alpha_0(\vec{x}) \\ z(\vec{x}) \end{bmatrix} \tag{3.14}$$

To formulate the control problem, we consider the ω-shifted linear quadratic regulator (ω-LQR) cost functional

$$J_\omega(\alpha, u) = \int_0^\infty \left((\alpha - z)^T Q(\alpha - z) + u^T R u\right) e^{2\omega t}\, dt \tag{3.15}$$

In (3.15) Q is a diagonal, positive semi-definite matrix consisting of state weights and R is a diagonal, positive definite matrix of control weights. The optimal control problem we consider is to minimize (3.15) over all controls $u \in L^2(0, \infty)$ subject to the constraints (3.13)-(3.14).

For an ω-controllable linear system, the tracking LQR problem has a unique solution of the form [12, 13]

$$u_{opt} = -\begin{bmatrix} R^{-1}B^T\Pi_1 & R^{-1}B^T\Pi_2 \end{bmatrix} X \tag{3.16}$$

where Π_1 is the unique symmetric, non-negative solution of the algebraic Riccati equation

$$(A + \omega I)^T\Pi_1 + \Pi_1(A + \omega I) - \Pi_1 B R^{-1} B^T \Pi_1 + Q = 0 \tag{3.17}$$

The matrix Π_2 in (3.16) satisfies the equation

$$\left((A + \omega I)^T - \Pi_1 B R^{-1} B^T\right)\Pi_2 = Q \tag{3.18}$$

Once the gain matrix K is obtained, the feedback control law is placed into the augmented state-space equation. The resulting closed-loop system is of the form

$$\dot{X} = (\bar{A} - \bar{B}K)X + \bar{F} + \bar{N} \tag{3.19}$$

The tracking ω-LQR formulation discussed above can be utilized to control the full order model of the system as well as the reduced order POD model with explicit control input. In the reduced order case the reference signal $z(x)$ is projected onto the POD modes to obtain reference coefficients $\{\tilde{\alpha}_j\}_{j=1}^N$. The resulting reduced order tracking problem is constructed in exactly the same way as for the full order case.

3.3 Snapshot Generation and Pod Modes

The desired outcome of this section is to control a full order model using only information from some POD modes. For a given solution $S(x,t)$ there are well-defined POD modes with properties as in Chapter 2, however these POD modes are for a given solution. We desire to control the solution of the PDE to be an arbitrary smooth reference function by forcing it on the boundary, and since we do not already know what the forcing function will be we cannot accurately simulate the forced solution without prior knowledge of the POD modes. This then defines the best POD modes implicitly, and to overcome this difficulty the solution is simulated with different forcing functions and then the POD modes used are those from the combination of the simulations (i.e. if two solutions are found for two different forcing functions, then the POD modes taken are those from adding the beginning of the second simulation to the end of the first and effectively doubling the simulation time). It is then assumed that with the proper choice of forcing functions the reference function will be in the span of the POD modes. The forcing functions chosen were sinusoidal inputs since they include polynomials of all positive integer orders and therefore assumed to excite most of the dynamics of the PDE, which will then be captured in the POD modes. The inputs were of the form

$$u_1(t) = \sin(\frac{3.92}{35^{.6} \cdot 1.6} t^{1.6}) \qquad u_3(t) = \sin(\frac{8}{50^2 \cdot 3} t^3) \tag{3.20}$$

This along with a run with u_1 and u_3 switched were each simulated for 50 seconds. The controls were chosen so that one would start off slower than the other, catch up at 35 seconds, and then for the remainder of the time be faster than the other. Also, the range of the frequency of the inputs was from 0 to 8 rad/s, a run from the zero initial condition to steady state was included, and all of the forcing function snapshots started from steady state with no controls.

The first eight pod modes are plotted in Figure 3.1. The initial condition is plotted in Figure 3.2 and the projected initial conditions using the first 20 POD modes are plotted in Figure 3.3 to compare how well the projected space captures the actual solution space.

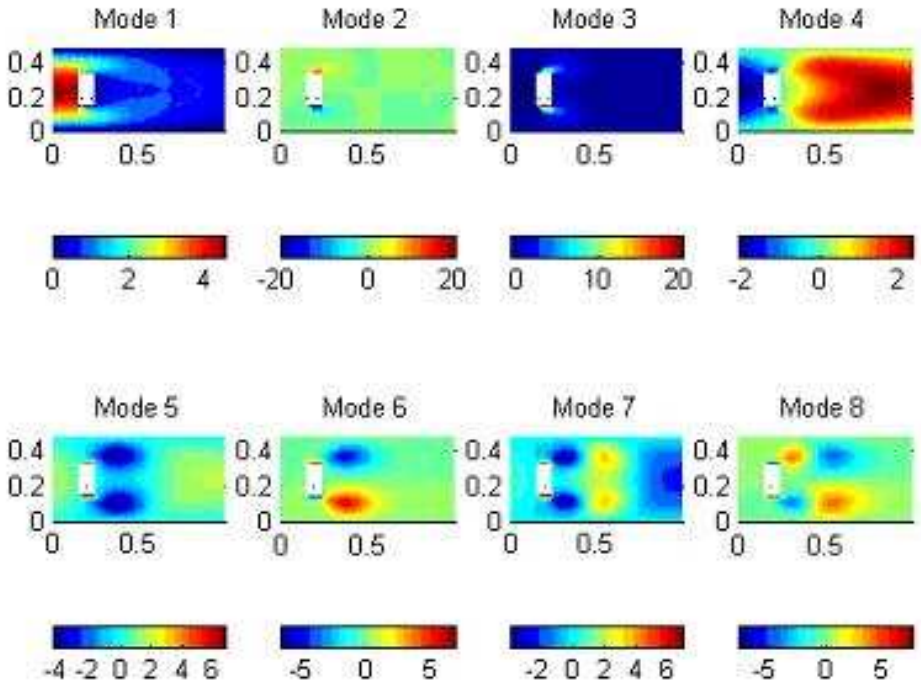

Figure 3.1: First eight POD modes for the sinusoidal snapshots

3.4 Open Loop Testing

To test the validity of the POD model, open loop controls were used on the full order model of the form

$$u_1(t) = u_3(t) = \sin(3t) \tag{3.21}$$

This input was chosen since it is not explicitly in the snapshots, but is complicated enough to excite an adequate amount of the dynamics. Figure 3.4 shows the full order response at 2 second intervals, and Figure 3.5 shows the reduced order response for the sinusoidal snapshots. Unless otherwise specified, all reduced order plots from here on in this section used 20 POD modes.

To see how well the reduced order model compared with the full order

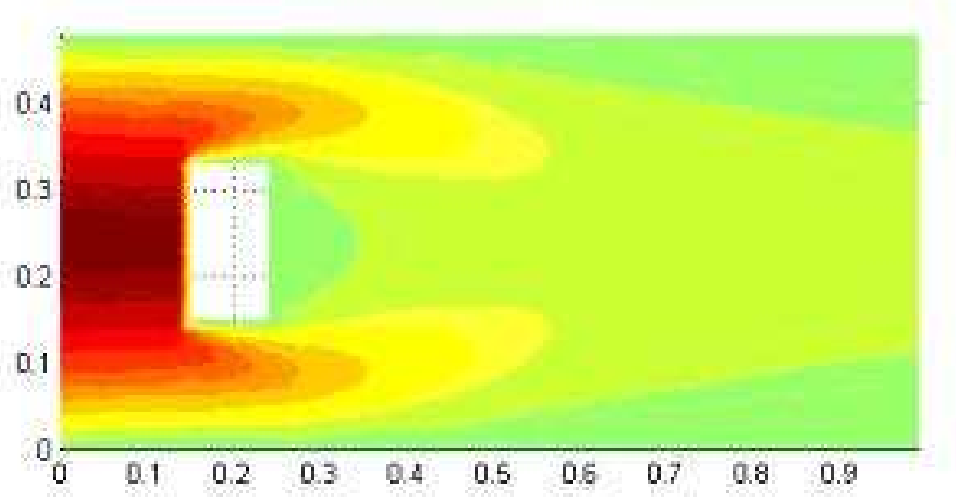

Figure 3.2: Steady-state (initial condition) for the full order solution

model it was compared with the best possible simulation when confined to the POD basis. This was done by plotting the first five POD mode coefficients versus the coefficients obtained by projecting the full order solution onto the POD basis. Figure 3.6 shows this comparison as a function of time for the sinusoidal snapshots. The projected coefficients are plotted in blue and the POD mode coefficients are plotted as red dashes.

Also compared was how well the POD modes capture the controls. Figure 3.7 shows the full order controls versus the reduced order controls as a function of time for the sinusoidal snapshots. Again, the full order control is plotted in blue and the reduced order control is plotted as red dashes.

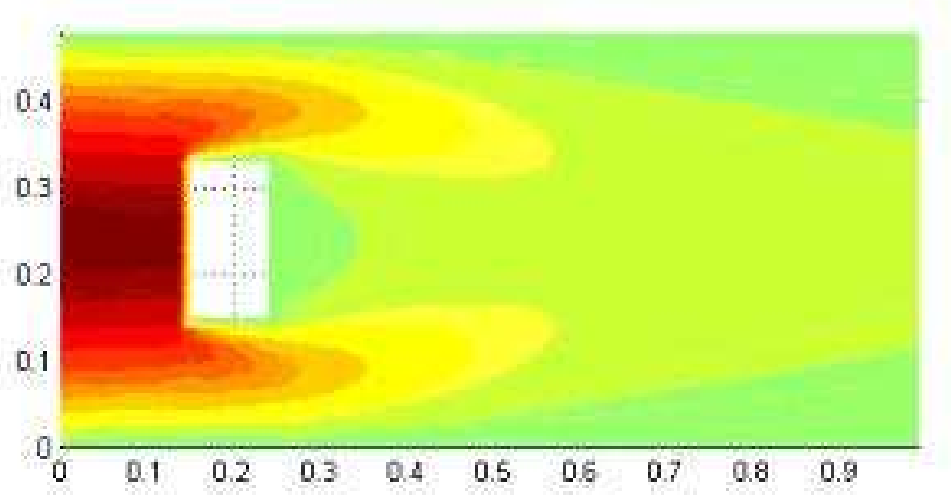

Figure 3.3: Initial condition for the sinusoidal snapshots

3.5 Closed Loop Tracking

Next, the POD model was used to track the full order model to a reference function. The reference function chosen was the steady-state solution with no controls and $Re = 50$. Figure 3.8 shows the reference function.

To see how well the closed loop system works, the assumed optimal solution was found by running the full order system with controls between 0 and -2. The controls were changed slowly enough so the the system was in quasi-equilibrium. Then the solution for each value of the control was subtracted from the reference function, which was squared and integrated over the domain. Figure 3.9 shows the values of the integrals for the range of controls previously specified. From this plot it appears that the optimal open-loop controls are $u_1 = u_3 = -1.135$.

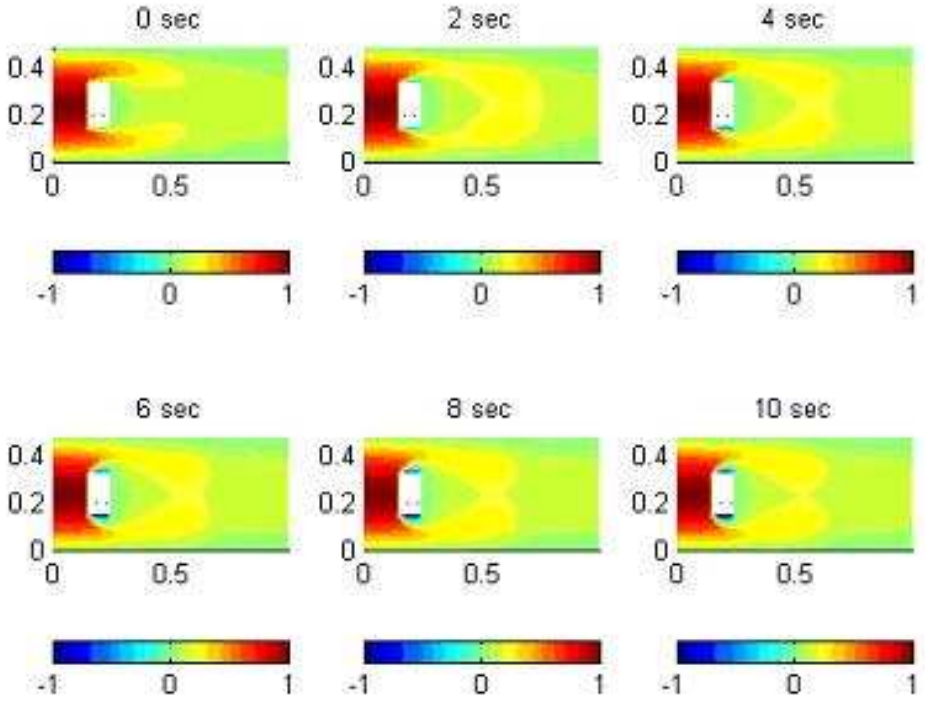

Figure 3.4: Full order response to a sinusoidal input

In order to track the reference function ω-LQR was used. The values used in the ω-LQR were the identity for R, 1000I for Q, and ω was chosen the largest possible so that $A + \omega I$ had all negative eigenvalues. For the sinusoidal snapshots $\omega = .17$. To see how well the tracking was achieved the reduced order solution was plotted in Figure 3.10, and the full order solution was plotted in Figure 3.11.

As another comparison the reduced order controls were plotted versus the full order controls for the tracking problem in Figure 3.12. Again, the full order controls are plotted in blue and the reduced order controls are plotted as red dashes.

From these plots it can be seen that the POD modes do a decent job of controlling the full order solution, and of course the values of the R and Q matrices place limitations on how well the full order solution can be con-

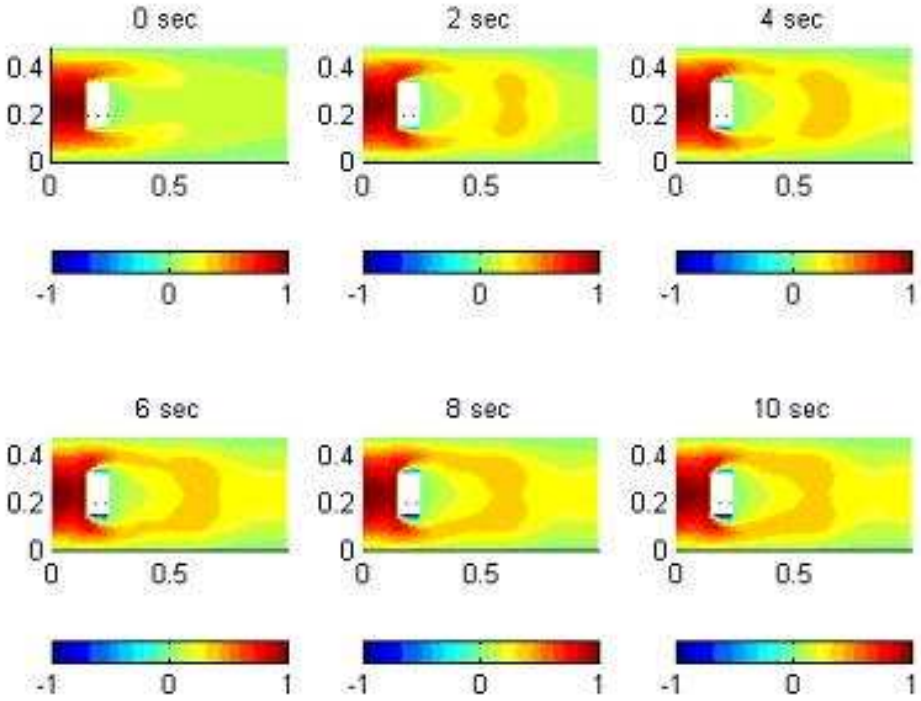

Figure 3.5: Reduced order response to a sinusoidal input for the sinusoidal snapshots

trolled, i.e. by keeping the controls from getting too big, which may be required to perfectly control the full order solution.

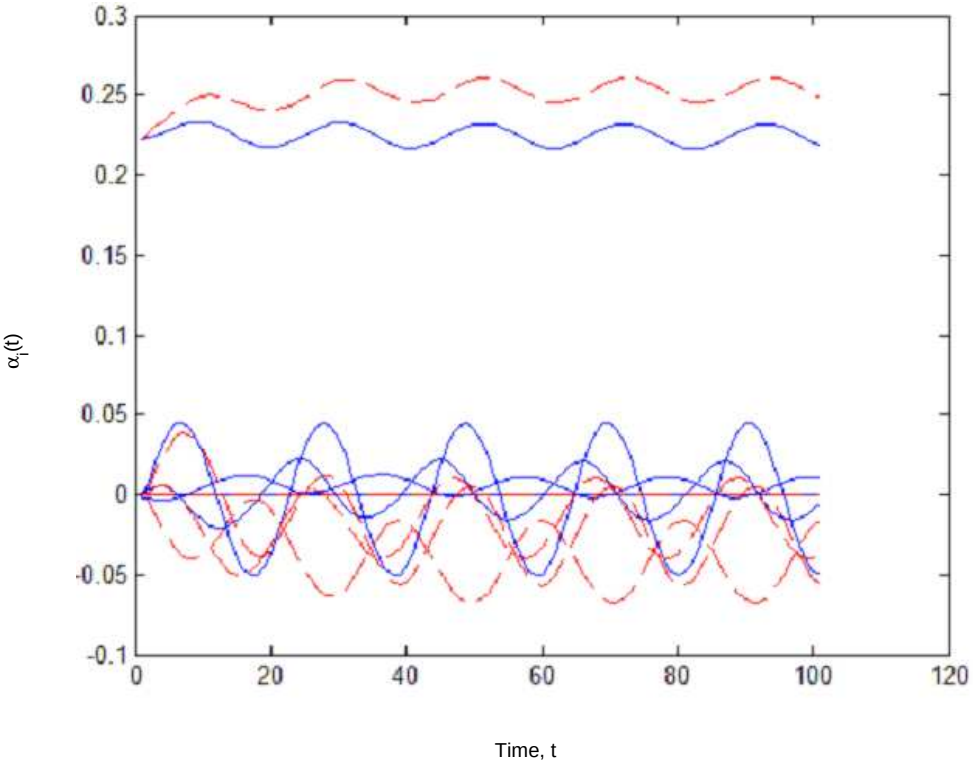

Figure 3.6: POD mode and projected coefficients as a function of time for the sinusoidal snapshots

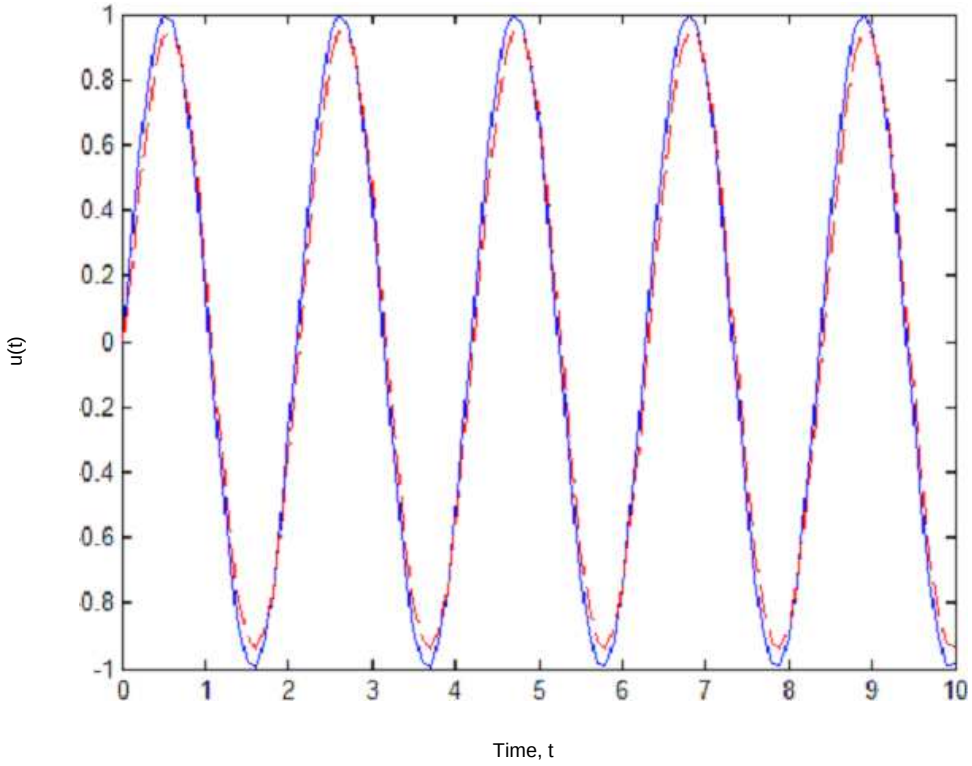

Figure 3.7: Full order and reduced order controls as a function of time for the sinusoidal snapshots

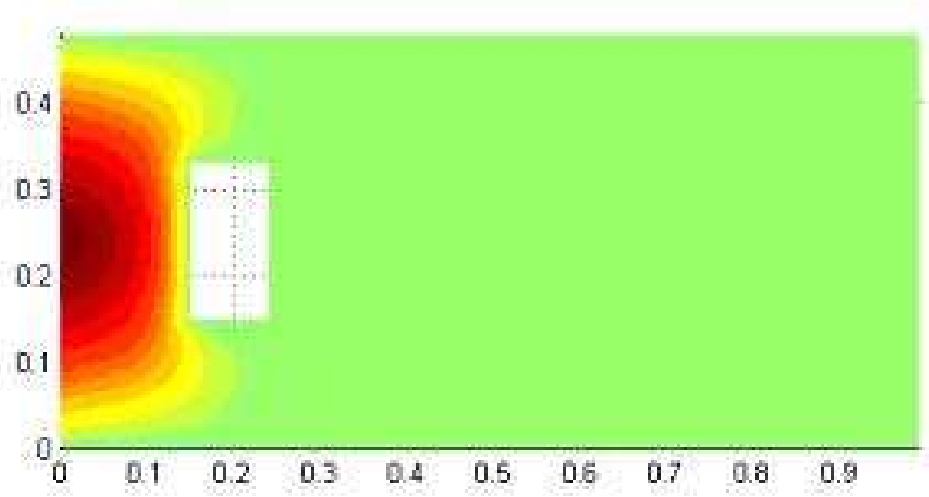

Figure 3.8: Full order reference function

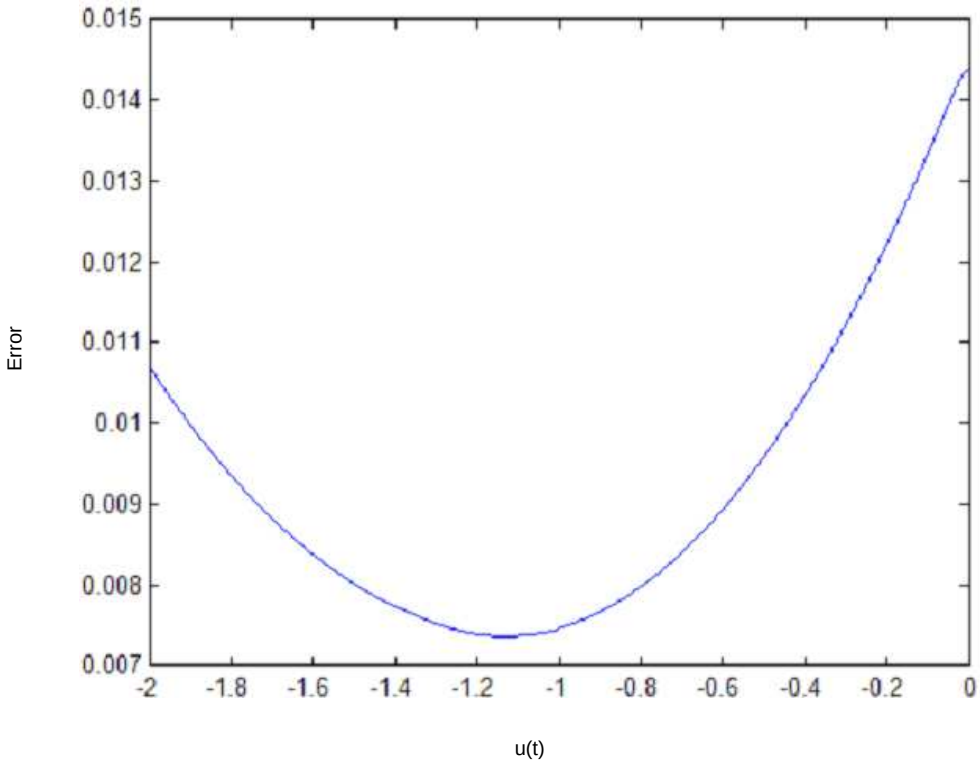

Figure 3.9: Metric of how close the full order model is to the reference function for a range of input values

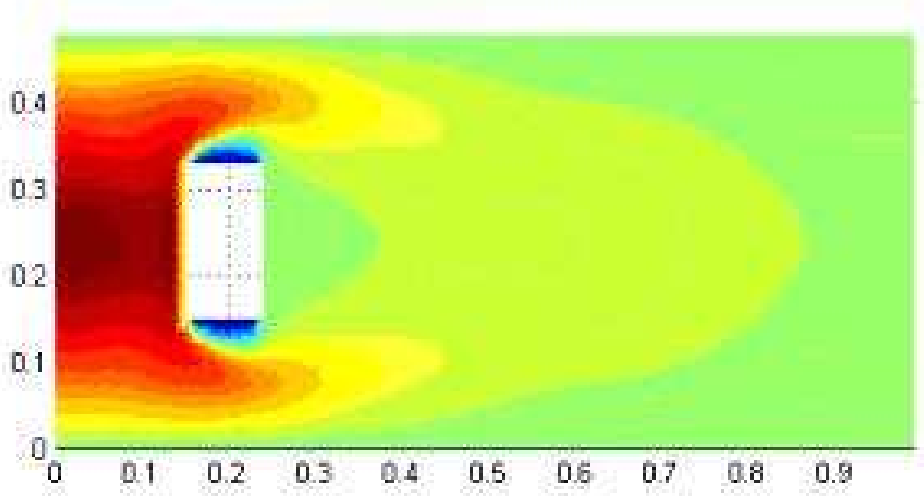

Figure 3.10: Reduced order tracking solution for the sinusoidal snapshots

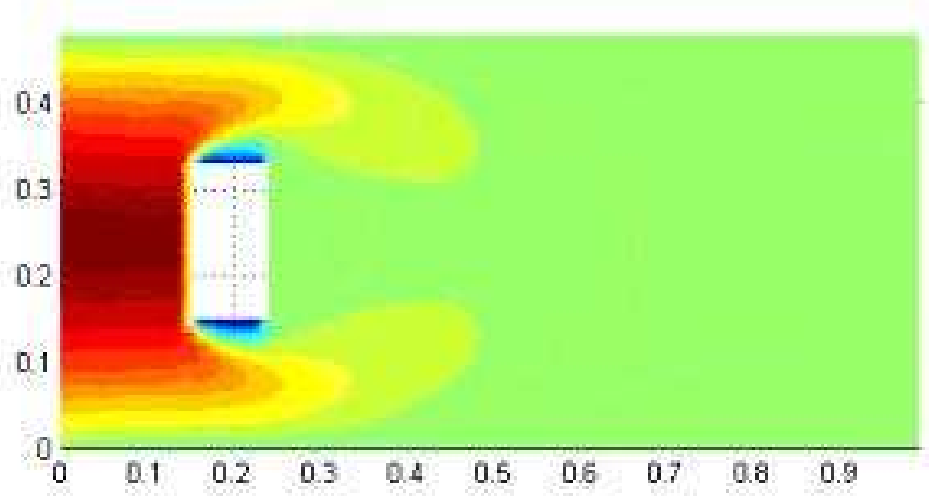

Figure 3.11: Full order tracking solution for the sinusoidal snapshots

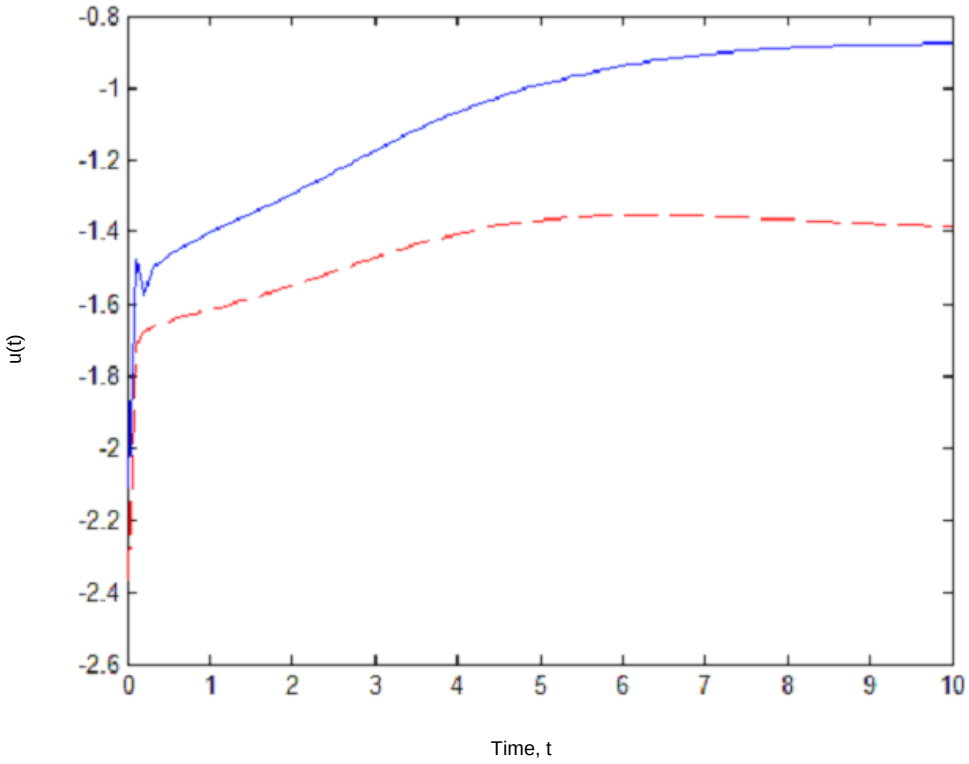

Figure 3.12: Full order and reduced order controls for the tracking problem with the sinusoidal snapshots

Chapter 4

Numerical Results

4.1 Partial Differential Equations with Parameters

In a previous chapter the spatial POD modes were found from the solution of a PDE that depended only on space and time. In many applications the PDE that describes a process may be dependent on other parameters, and POD modes that work independently of the parameter values given may be desired. Suppose that the PDE is dependent on one parameter, ϵ, which is within $[\epsilon_1, \epsilon_2]$, then returning to equation (2.7) and generalizing yields the problem

$$\max_{S_N} \left\{ \frac{1}{\Delta\epsilon} \int_{\epsilon_1}^{\epsilon_2} \frac{1}{T} \int_T \|S_N(\vec{x}, t, \epsilon)\|^2 \, dt \, d\epsilon \right\} \tag{4.1}$$

This then becomes the generalized form of equation (2.15):

$$\int_{\epsilon_1}^{\epsilon_2} \int_\Omega R(\vec{x}, \vec{y}, \epsilon)\phi(\vec{y}) \, dy \, d\epsilon = \lambda\phi(\vec{x}) \tag{4.2}$$

where $R(\vec{x}, \vec{y}, \epsilon)$ is defined as $R(\vec{x}, \vec{y})$ with $S(\vec{x}, t)$ replaced by $S(\vec{x}, t, \epsilon)$.

Since the solution to the PDE is not known, solutions to equation (4.2) must be approximated by a numerical approximation for $\int_{\epsilon_1}^{\epsilon_2} f(\vec{x}, \epsilon) \, d\epsilon \approx \sum_{i=1}^{n} w_i f(\vec{x}, \epsilon_i)$. The main problem in this section is investigating how to choose the weights and evaluation points of the numerical approximation.

4.1.1 The One Dimensional Heat Equation

To start, a test case was chosen: the one-dimensional heat equation on $[0, 1]$ with one end fixed and the other given a sinusoidal displacement, or

$$\frac{\partial}{\partial t}w(x,t) = \epsilon\frac{\partial^2}{\partial x^2}w(x,t) \tag{4.3}$$

$$w(0,t) = \sin(t) \tag{4.4}$$

$$w(1,t) = 0 \tag{4.5}$$

Next, different numerical schemes were compared starting with the Midpoint rule with one point and Gaussian Quadrature, using five points. To measure the error of the different approximations they were compared with the full order solution as given by a finite difference method. Figure 4.1 shows the error of the POD modes over the interval $[.01, 1]$ along with the error of the POD modes evaluated by using only the data from the given ϵ, which barring numerical error should be the best possible error.

The Midpoint rule was chosen because it is the simplest rule possible and amounts to assuming that if only one set of POD modes can be used for different epsilons then perhaps the one corresponding to the epsilon in the middle of the interval would be best. The error involved in the Midpoint rule is $\frac{f''(\xi)}{24}h^3$ where ξ is some number in $[\epsilon_1, \epsilon_2]$. The Gaussian Quadrature method is a popular, accurate, and easily used method which is of $O(\epsilon^{10})$. The Gaussian Quadrature method was compared with the Midpoint rule and the POD modes from the given epsilon to compare how well a simple rule and an accurate rule can capture the dominant system characteristics over a given range. From Figure 4.1 it can be seen that the Gaussian Quadrature method is notably better than the Midpoint rule as would be expected from normal integration, but there is no noticeable difference for epsilons larger than three tenths and both of them show growing error as epsilon gets closer to one hundredth. The error from the POD modes from the given epsilon, assumed to be the best possible, also shows this trend of growing error near epsilon equal to one hundredth but it is not as sharp and hence also shows that improvement is possible.

This trend is intuitive from a physical perspective as for large epsilon an arbitrary point inside the domain is affected locally by points farther away than for small epsilons. Thus for large epsilon the solution is much closer to linear which can be very easily captured by one POD mode and hence the error is not expected to be large. As the value of epsilon is decreased

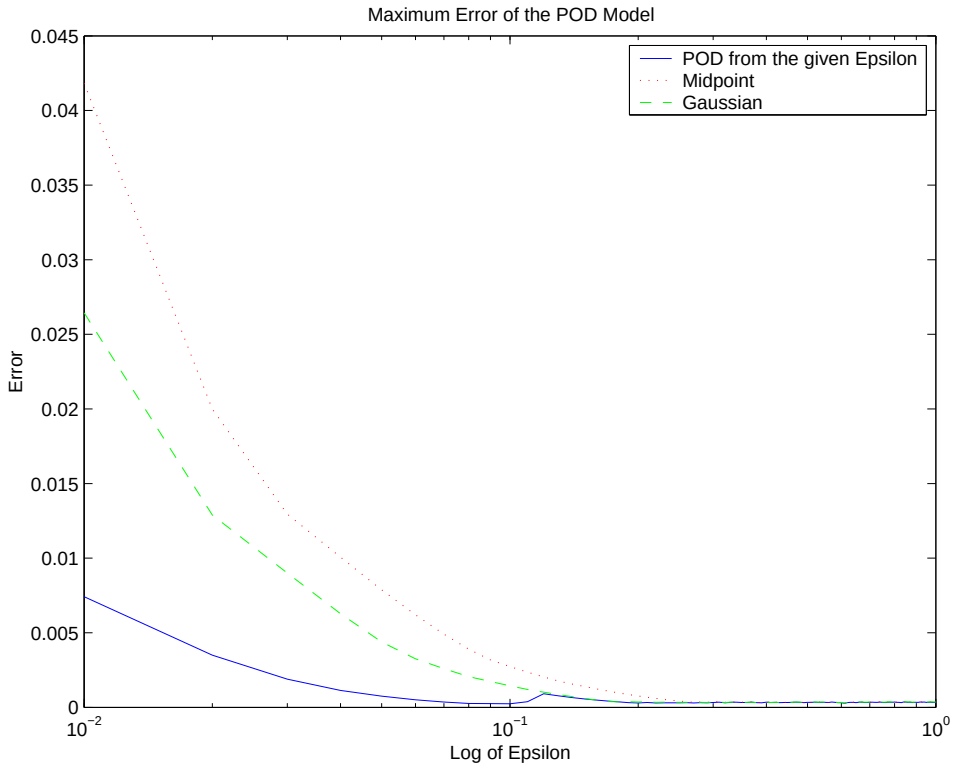

Figure 4.1: Maximum error of the Midpoint rule and Gaussian Quadrature POD models

an arbitrary point is only affected by points which are close by, which allows a solution to have much more curvature. Added to this is the fact that the forcing function on the left boundary is sinusoidal, which generates traveling waves, and since the points on the right side are not affected by those on the left, which is where the action is, for smaller epsilon the right side is essentially zero. This kind of solution is not easily captured by POD modes which are close to linear, which are the kind generated when epsilon is larger than approximately three tenths, or over seventy percent of the desired epsilon range. Figure 4.2 shows the first POD modes at different values of epsilon, which reaffirms the fact that for larger values of epsilon the dominant motion is linear whereas for smaller epsilons the motion has more curvature. Based on this analysis, only a few POD modes are needed when epsilon is large, but many more are needed when epsilon is small, so the points used to gather

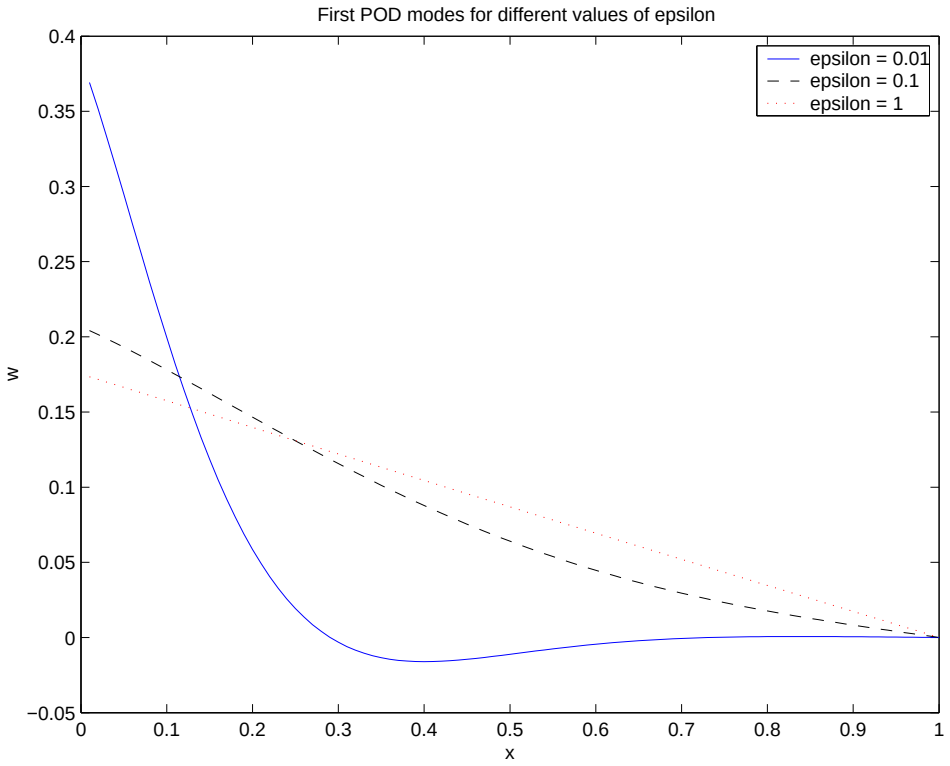

Figure 4.2: First POD modes for different values of epsilon

data for the POD modes should be clustered closer to epsilon equals one hundredth.

Continuing in our search for a better numerical integration method, since the dependence of the solution of the PDE on ϵ is not known and from figure 4.1 the error is largest near $\epsilon = .01$, different numerical integration methods were tried. One method for comparison with the Gaussian Quadrature using five points was the midpoint rule with five points. This is only $O(\epsilon^2)$, but again since the dependence of the PDE on epsilon is not known this is a way of seeing how the error depends on the number of points used to evaluate the integral. Also, to see how the error depends on the distribution of the points used, a second midpoint type method was used, except with the points chosen from those used in Gaussian Quadrature (i.e. both the second midpoint rule and Gaussian Quadrature were of the form $\sum_{i=1}^{n} w_i f(\vec{x}, \epsilon_i)$ with the same $\epsilon_i's$ but with different $w_i's$, or weights). The error for these three methods is

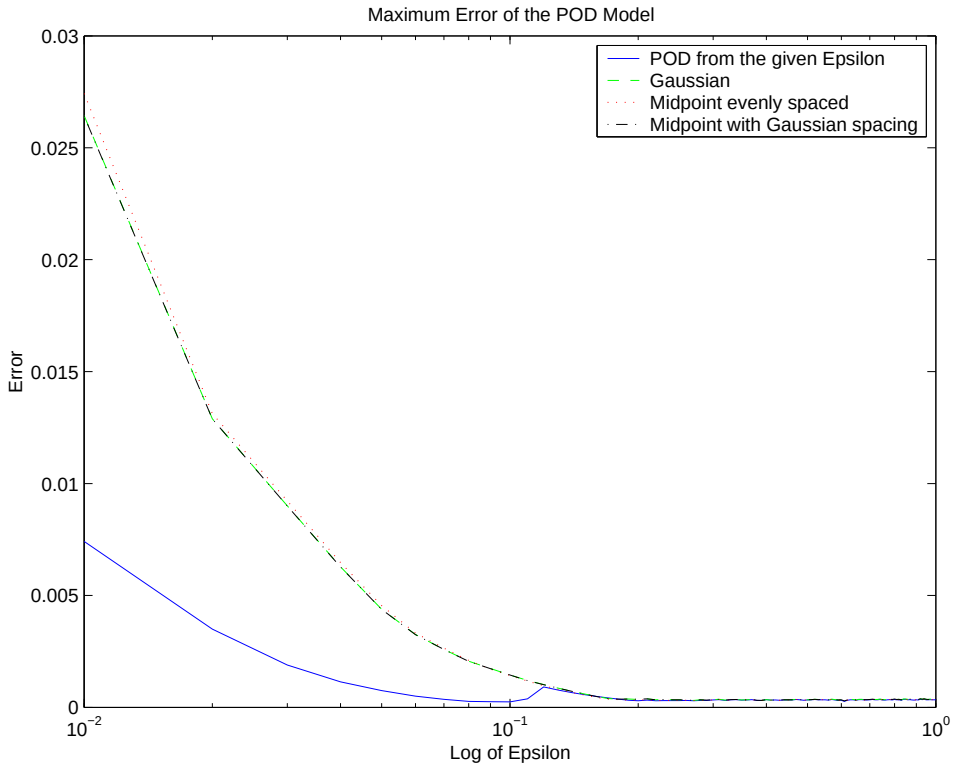

Figure 4.3: Maximum error of the five point Midpoint rule, Midpoint rule with Gaussian spacing, and Gaussian Quadrature POD models

shown in Figure 4.3, which shows that they all have almost identical errors throughout the entire range. This signifies that perhaps the weighting is not the most important but the distribution of the points is.

In an attempt to use our knowledge that the solution is hardest to capture when epsilon is small, a second Gaussian Quadrature method was tried, this time breaking the integral into two, one on the range $[.01, .11]$ with three points and the other on the range $[.11, 1]$ with two points. This effectively clusters the data closer to epsilon equals one hundredth, which was previously argued to intuitively be a better approximation. Upon closer examination of Figure 4.3 the Midpoint rule with Gaussian spacing is seen to be the best out of those methods tried in Figure 4.3, and the best so far, so it was compared with our new Gaussian Quadrature rule, which can be seen in Figure 4.4. This shows a drastic improvement over previous methods tried and the error

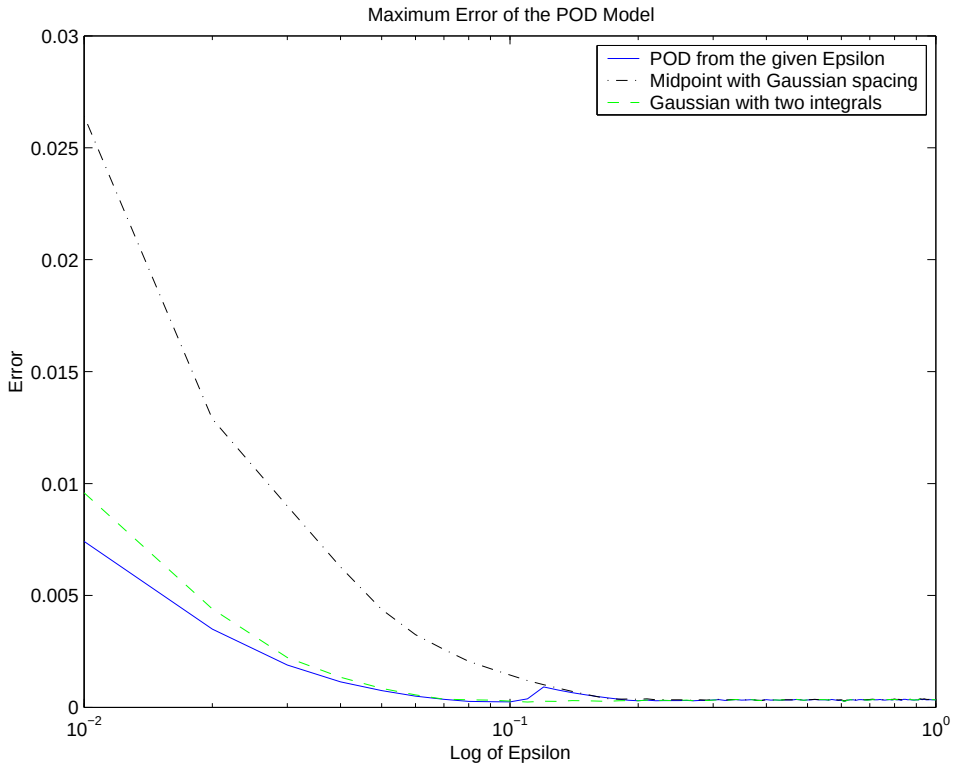

Figure 4.4: Maximum error of the Midpoint rule with Gaussian spacing and double integral Gaussian Quadrature POD models

is close to the best possible over the entire range.

For one last attempt an integration method was tried which is meant for functions with a singularity at one end, since the error appears to be blowing up as epsilon approaches one hundredth. This method is meant for functions with a singularity of the order $\frac{1}{\sqrt{x}}$ and has an error of $\frac{2^{4n+1}}{4n+1}\frac{[(2n)!]^3}{[(4n)!]^2}f^{(2n)}(\xi)$ where n is the number of points used and ξ is some point inside the interval[1]. For the given case of five points this error is $8.1 \times 10^{-13} f^{(10)}(\xi)$. To try to most accurately compare with the Gaussian method of two integrals, a second singular method was tried, this time breaking the integral into two exactly as before, using the singular method for the first and Gaussian Quadrature for the second. These are shown in Figure 4.5 which shows that the singular method is not as good as the double Gaussian, but the double singular is slightly better than the double Gaussian, as might be expected. For com-

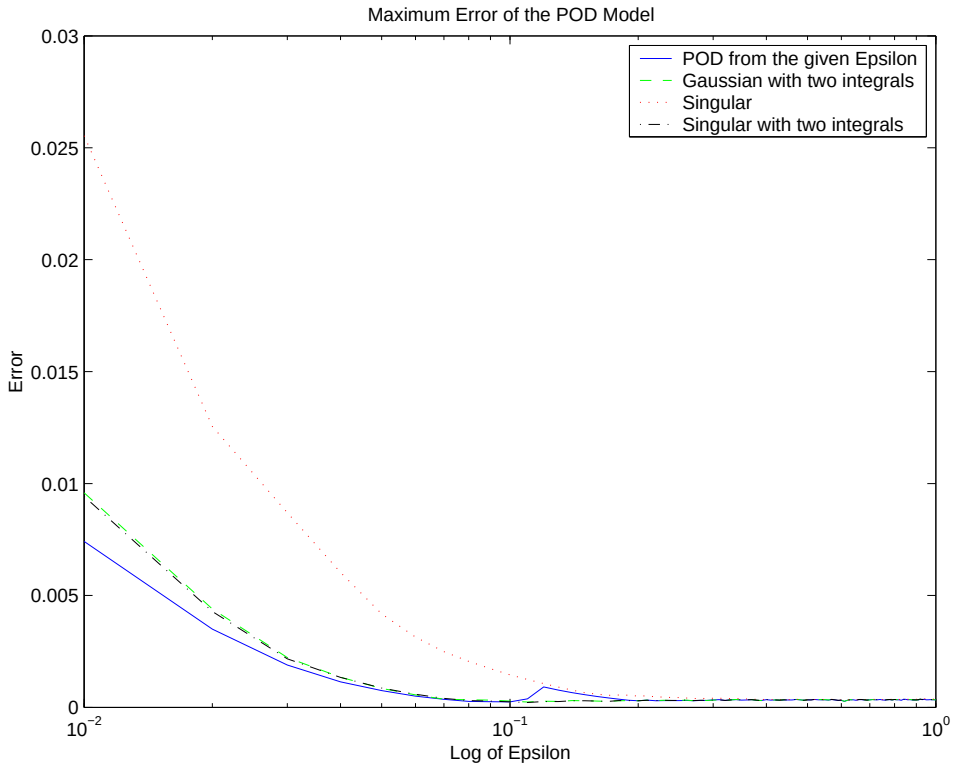

Figure 4.5: Maximum error of the double integral Gaussian Quadrature, singular rule, and double integral mixed singular/Gaussian Quadrature POD models

parison the singular method is slightly better than the Midpoint rule with Gaussian spacing.

4.1.2 The One Dimensional Heat Equation with a Gradient Term

In an attempt to see how robust this method is at representing the solution to a PDE over a range of parameter values, another, harder, test case was tried. This was the same as the PDE in the last subsection except that a gradient term was added, which has the affect of adding convection to the

system. The PDE and boundary conditions were

$$\frac{\partial}{\partial t}w(x,t) + \epsilon\frac{\partial}{\partial x}w(x,t) = 0.1\frac{\partial^2}{\partial x^2}w(x,t) \tag{4.6}$$

$$w(0,t) = \sin(t) \tag{4.7}$$

$$w(1,t) = 0 \tag{4.8}$$

This time the value of the coefficient of the laplacian term was kept fixed at 0.1 which is approximately 10% from the left of the range of values used in the previous subsection, for reference. The parameter was identified with the coefficient of the gradient term, and the range of values was again $[.01, 1]$. To see how the gradient term changes the POD modes, the first POD modes for different values of epsilon are shown in Figure 4.6, which shows that dominant motion has been changed and the linearity of the solution for larger epsilons has been lost.

Preceding as before the Midpoint rule with one point and Gaussian Quadrature were compared with the POD modes using only the data from the given ϵ, which can be seen in Figure 4.7. As expected and as before the Midpoint rule is better at the middle of the interval but Gaussian Quadrature is better for all values except those very close to the middle. Contrary to before, the error does not appear to be blowing up anywhere although the error for Gaussian Quadrature does appear to be linear with the most error towards epsilon equals one hundredth.

As before the five point Midpoint rule and the Midpoint rule with Gaussian spacing were tried. These can be seen in Figure 4.8, which shows that they are almost identical, having the same shape and the difference between each appearing almost constant.

Since the last POD models had the largest error towards epsilon equals one hundredth, a double integral was again tried, using three points on $[.01, .11]$ and two points on $[.11, 1]$. The results can be seen in Figure 4.9. This time the double integral method is not the best one and is the worst out of the five point methods.

As a comparison the singular method and double integral mixed singular/Gaussian Quadrature were again tried, although they were not expected to perform as well as before since the error does not display the same behavior for small epsilons. Figure 4.10 shows the errors from which it can be seen that although the double integral method is better than the single integral, the singular method is not an improvement.

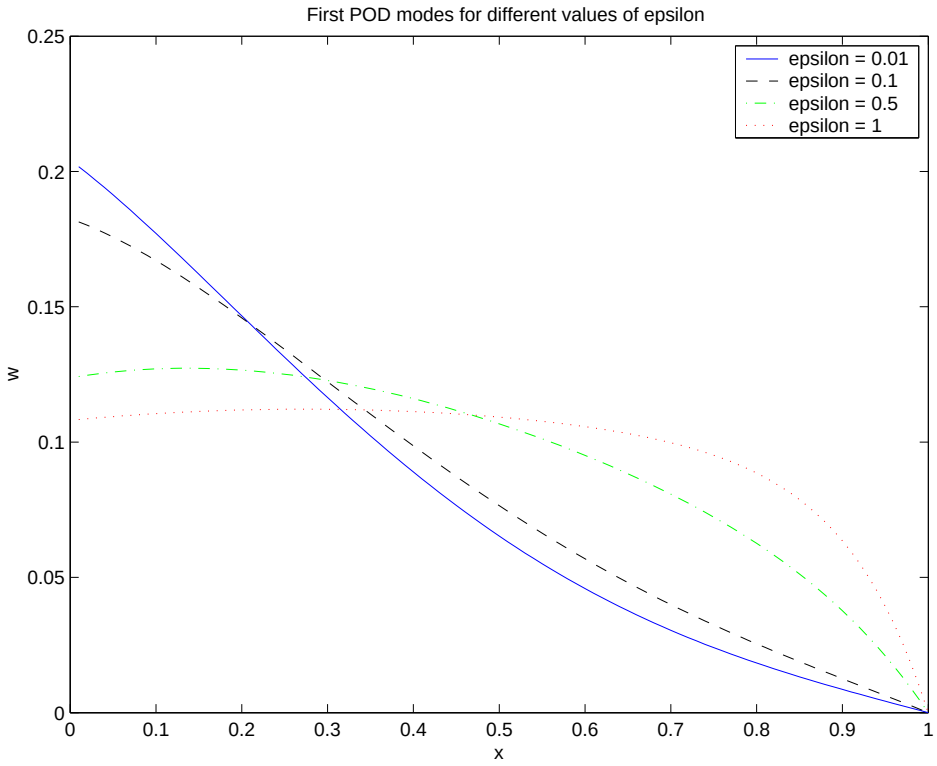

Figure 4.6: First POD modes for different values of epsilon

4.2 Approximation by Subdomains

In Section 2.2 we discussed the computational advantages that can be obtained by splitting a system up into subdomains. This raises the question of whether or not the solutions on the subdomains converge to the solution on the entire domain. If the solution is known on suitable surfaces inside the domain that can be used as boundaries for subdomains, then the solution can be approximated by some full order approximation (or some other method of generating the solution data), and this data can be used to generate POD modes for each subdomain. Next the POD modes can be used to generate a reduced order model of each subdomain and the accuracy of this model can be checked versus some solution of the entire domain restricted to the subdomain. However; since the boundary data is already known and prescribed into the reduced order model, assuming that the solution to the PDE is unique, this amounts to the usual question of whether or not the POD

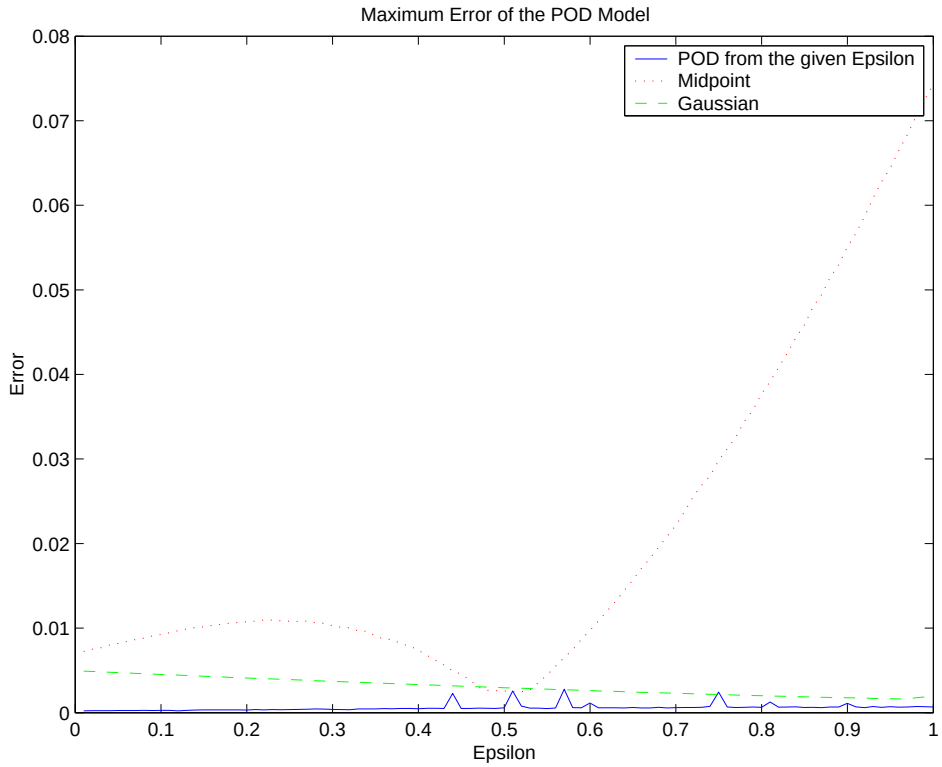

Figure 4.7: Maximum error of the Midpoint rule and Gaussian Quadrature POD models

modes of a domain can accurately describe the solution, and the fact that the POD modes are defined on a subdomain really does not matter. Also, in most cases the reason for using domain decomposition is to avoid having to solve the PDE on the entire domain at once, which may be computationally impossible.

A more reasonable problem with actual convergence issues would be one which never simulates the full order method over the entire domain. One such problem was investigated, namely to split the domain into some arbitrary, well chosen subdomains, and prescribe arbitrary POD modes to each subdomain (in this case the initial POD modes were all zero functions). Next, some sort of Red-Black scheme was used to divide the subdomains into two disjoint groups; in the cases chosen to investigate, the subdomains were capable of being split into two groups such that each subdomain in one group was

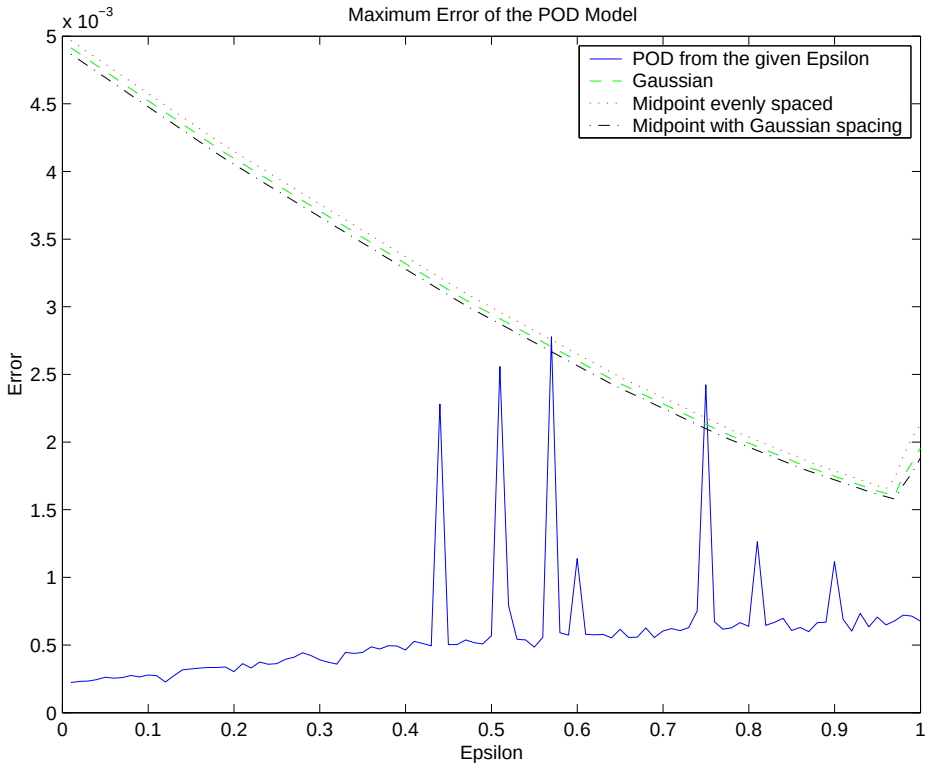

Figure 4.8: Maximum error of the five point Midpoint rule, Midpoint rule with Gaussian spacing, and Gaussian Quadrature POD models

surrounded completely by subdomains in the other group. This is intuitively the best type of separation for the iterative procedure described below as at each iteration it passes along the most information to the next iterate. Then the solutions on the red subdomains were simulated with full order solutions while the solutions on the black subdomains were simulated with reduced order models using their respective POD modes. The boundary data on the red subdomains was prescribed to be the boundary data from the black subdomains and the boundary data on the black subdomains was prescribed to be the boundary data from the red subdomains. This created small discontinuities on the boundaries between the red and black subdomains but helped keep the solution over the entire domain relatively continuous. The data that was obtained from the full order simulations of the red subdomains was then used to obtain new POD modes for the red subdomains, and then the process

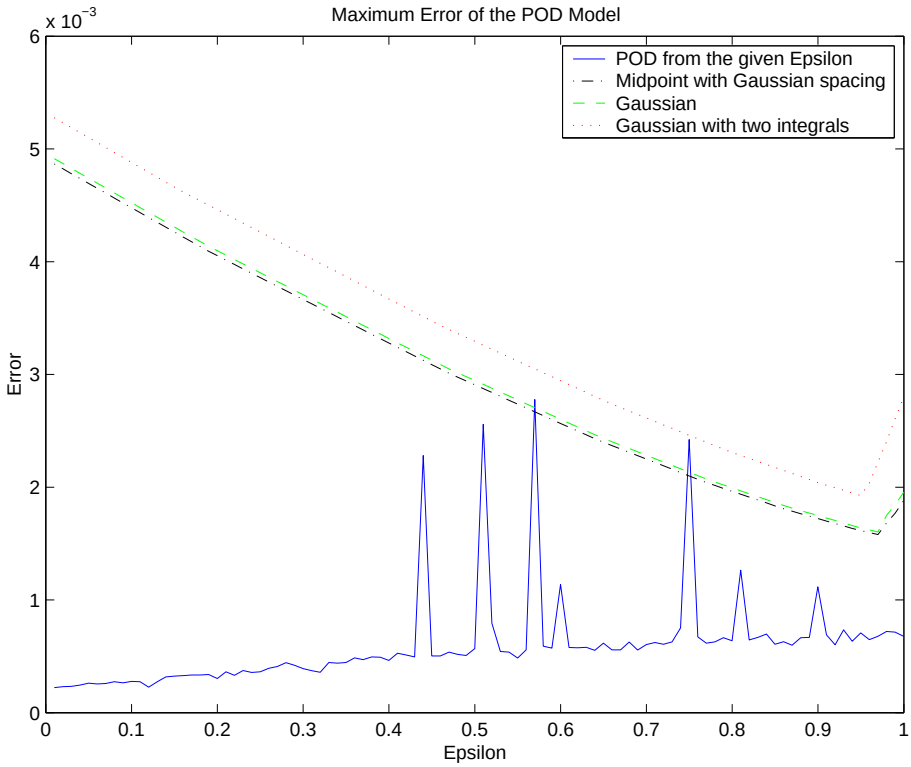

Figure 4.9: Maximum error of the Midpoint rule with Gaussian spacing, Gaussian Quadrature, and double integral Gaussian Quadrature POD models

was reversed so that the full order simulations were on the black subdomains and the reduced order models were on the red subdomains (using the newly found POD modes), with the end result of new POD modes on each black subdomain. This process was then iterated except that each time the full order model was run on a subdomain the data obtained was combined with the data obtained on all the previous runs using the full order model and this combination of data was used to generate the next sequence of POD modes.

Two test cases were chosen: the first was the two dimensional heat equation on $[0,1] \times [0,1] = \Omega \subset \mathbb{R}^2$ with the left, right, and top boundaries fixed at zero for all time and the bottom boundary given a forcing function. This

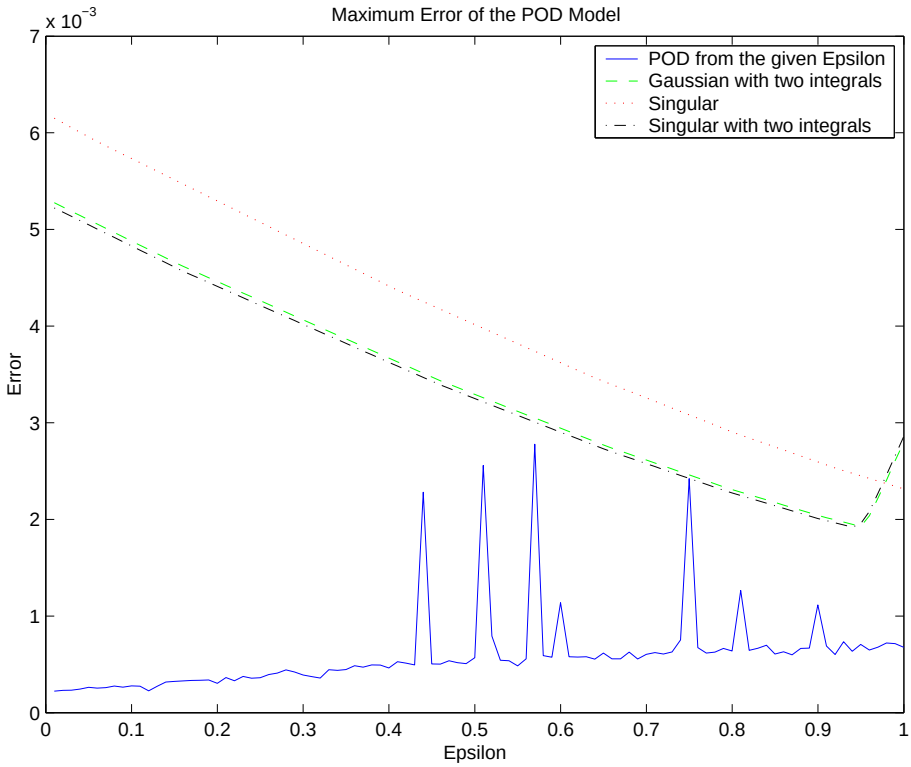

Figure 4.10: Maximum error of the double integral Gaussian Quadrature, singular rule, and double integral mixed singular/Gaussian Quadrature POD models

corresponds to

$$\frac{\partial}{\partial t}w(x,y,t) = \epsilon\left(\frac{\partial^2}{\partial x^2}w(x,y,t) + \frac{\partial^2}{\partial y^2}w(x,y,t)\right) \tag{4.9}$$

$$w(x,0,t) = f(x,t) = c\sin(\omega t) \tag{4.10}$$

$$w(x,1,t) = w(0,y,t) = w(1,y,t) = 0 \tag{4.11}$$

for $t > 0$ and $(x,y) \in (0,1)\times(0,1)$, where ϵ, c, and ω were positive constants and a zero initial condition was given. Ω was divided into Ω_i $i = 1,2,...,9$, each with $\frac{1}{9}$ of the total area and with Ω_i $i = 1,2,3$ on the x-axis, Ω_1 at the origin, and Ω_i $i = 7,8,9$ on the line $y = 1$ as shown in Figure 4.11. Then the Red-Black iteration method was implemented with the odd numbered domains as one color and the even numbered domains as the other.

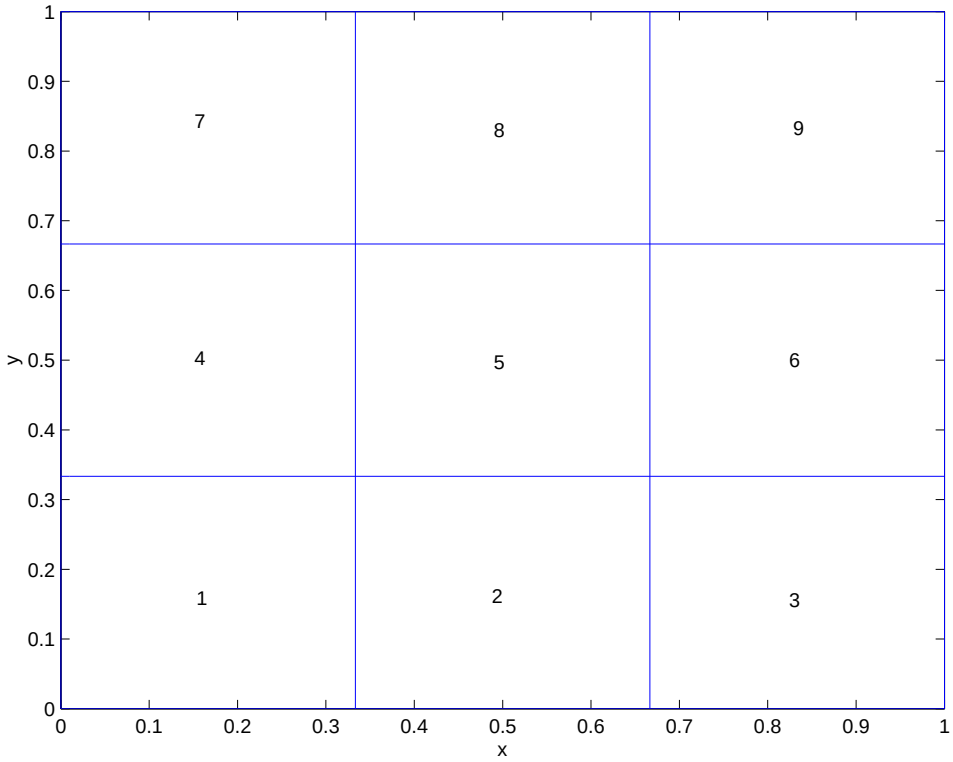

Figure 4.11: Illustration of the chosen domain decomposition

The second test case was the wave equation on $\Omega = [0, 1]$ with the left boundary given a forcing function and the right boundary given conditions to simulate the string being attached to a spring on the right end. This corresponds to

$$\rho \frac{\partial^2}{\partial t^2} w(x,t) = \frac{\partial}{\partial x} \tau \frac{\partial}{\partial x} w(x,t) \tag{4.12}$$

$$w(0,t) = f(t) = c \sin(\omega t) \tag{4.13}$$

$$m \frac{\partial^2}{\partial t^2} w(1,t) = -\tau \frac{\partial}{\partial x} w(1,t) - k w(1,t) \tag{4.14}$$

for $t > 0$ and $x \in (0, 1)$, where ρ, τ, c, ω, m, and k were positive constants and a zero initial condition was given. Ω was divided into four equal partitions Ω_i $i = 1, 2, 3, 4$ with Ω_1 at the origin and Ω_4 at $x = 1$. Then the Red-Black iteration method was implemented with the odd numbered domains as one color and the even numbered domains as the other.

Table 4.1: The maximum error in the reduced order model for the heat equation using different forcing functions

Heat equation	Test one	Test two	Test three
1^{st} iteration	0.5900	589.82	0.6889
2^{nd}	0.0473	47.245	0.0638
3^{rd}	0.0186	18.639	0.0563
4^{th}	0.0118	11.815	0.0583
5^{th}	0.0098	9.795	0.0585

To check the convergence of the POD modes, after each iteration a simulation of the PDE over Ω was run with all the $\Omega_i's$ using their latest POD modes. This simulation was then compared with the simulation with all the $\Omega_i's$ using a full order model. For the heat equation three different tests were run, each with t_1 (the length of the simulation) equal to 10 seconds and a different forcing function. They were $f_1(x,t) = 10\sin(\frac{2}{3}t)$, $f_2(x,t) = 10,000\sin(\frac{2}{3}t)$, and $f_3(x,t) = 10$; also, for all cases $\epsilon = \frac{5}{81}$. Table 4.1 shows the results for each of the three cases by displaying the maximum absolute value of the difference between the reduced order model and the full order model. For the wave equation three different test were run, each with t_1 equal to 30 seconds and a different forcing function, $f_i(t)$ $i = 1, 2, 3$ the same as above; also $\rho = 1$, $\tau = 1$, $m = 1$, and $k = 10$. Table 4.2 shows the results for each of the three cases.

From the tables it can be seen that the amount of error in the reduced order model appears to be proportional to the magnitude of the forcing function. Also, the convergence at first is rapid and monotonic, but after the error has "bottomed out" it rises and falls slowly, and appears to converge as the number of iterations increases due to the large number of snapshots.

Table 4.2: The maximum error in the reduced order model for the wave equation using different forcing functions

Wave equation	Test one	Test two	Test three
1^{st} iteration	50.139	50151	18.096
2^{nd}	8.484	8490.2	4.6030
3^{rd}	0.0134	13.556	0.3841
4^{th}	0.0134	13.594	0.3701
5^{th}	0.0135	13.527	0.4179

Chapter 5

Conclusions and Future Work

Proper Orthogonal Decomposition is a decomposition which seeks a specified number of orthogonal functions that best represent a solution in a time-averaged sense. To find these functions an approximation to the solution was used which had values at discrete points in time, known as snapshots, of discrete points in space. Proper Orthogonal Decomposition was used as a reduced order modeling scheme to control a system and to simulate a parametric and a coupled system. The controlled system showed that POD can accurately and cheaply keep a system close to its desired state. The parametric system showed that with a proper choice of approximations in the desired interval the system can be well represented over its range, without drastically increasing the number of necessary POD modes, and it was seen that the choice of approximations may be more important then the weighting values used. The coupled system showed that POD can be used on subdomains with no prior knowledge of the complete system, and without ever fully simulating the system, the POD modes can converge to the POD modes on the subdomains.

Problems which remain are estimating error bounds in all cases and developing a convergence theory for the coupled system. A more systematic and less intuitive approach to picking the approximation points to the parametric system is desired, such as a statistical sampling technique. Also, systems with higher order parameters should be investigated as well as ways to incorporate the PDE into the selection process of picking the approximation points. Finally, all of these methods should be tried on more complicated problems.

Bibliography

[1] Milton Abramowitz and Irene A. Stegun, editors. *Title Handbook of mathematical functions with formulas, graphs, and mathematical tables.* Applied mathematics series. U.S. Dept. of Commerce, 10th edition, 1972.

[2] K. Afanasiev, M. Morzyński, B. R. Noack, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[3] A.C. Antoulas, C.A. Beattie, E. Gildin, and S. Gugercin. Krylov-based controller reduction for large-scale systems. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, December 2004.

[4] A.C. Antoulas, S. Gugercin, and D.C. Sorensen. A survey of model reduction methods for large-scale systems, 2001.

[5] P. Astrid, T. Backx, S. Weiland, and K. Willcox. Missing point estimation in models described by proper orthogonal decomposition. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[6] Jeanne A. Atwell, J. Borggaard, and Belinda B. King. Reduced order controllers for Burgers' equation with a nonlinear observer. *International Journal of Applied Mathematics and Computer Science*, 11(6):1311–1330, 2001.

[7] Jeanne A. Atwell and Belinda B. King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and Computer Modeling*, 33:1–19, 2001.

[8] H.T. Banks, S.C. Beeler, G.M. Kepler, and H.T. Tran. Reduced order modeling and control of thin film growth in an HPCVD reactor. *SIAM Journal on Applied Mathematics*, 62(4), 2002.

[9] G. Berkooz, P. Holmes, and J.L. Lumley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge University Press, 1996.

[10] T. Bui-Thanh, M. Damodaran, and K. Willcox. Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics. In *Proceedings of the 15th AIAA Computational Fluid Dynamic Conference*, Orlando, FL, June 2003. Paper 2003-4213.

[11] John A. Burns and Belinda B. King. A reduced basis approach to the design of low order compensators for nonlinear partial differential equation systems. *Journal of Vibration & Control*, 4:297–323, 1998.

[12] R. Chris Camphouse. Boundary control of a 2D parabolic process.

[13] R. Chris Camphouse. A boundary feedback control method for proper orthogonal decomposition system models. Control Theory Optimization Branch, Air Vehicles Directorate, Wright-Patterson Air Force Base.

[14] Edgar Caraballo, James DeBonis, and Mo Samimy. Low dimensional modeling of flow for closed-loop flow control. In *Proceedings of the 41st AIAA Aerospace Science Meeting*, Reno, NV, January 2003. AIAA-2003-0059.

[15] T. Colonius, R.M. Murray, and C.W. Rowley. POD based models of self-sustained oscillations in the flow past an open cavity. In *Proceedings of the 6th AIAA/CEAS Aeroacoustics Conference*, 2000. AIAA Paper Number 2000-1969.

[16] M. Fahl and E.W. Sachs. Reduced order modeling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In L.T. Biegler, O. Ghattas, M. Heinkenschloss, and B. Van Bloemen Waanders, editors, *Large Scale PDE-Constrained Optimization.* Springer-Verlag, 2003.

[17] D. Hömberg and S. Volkwein. Laser surface hardening using proper orthogonal decomposition for a three-dimensional example. In *Proceedings in Applied Mathematics and Mechanics*, volume 3, pages 1–4, 2003.

[18] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientarum Fennicae*, 37, 1946.

[19] Belinda B. King. Nonuniform grids for reduced basis design of low order feedback controllers for nonlinear continuous time systems. *Mathematical Models & Methods in Applied Sciences*, 8:1223–1241, 1998.

[20] S. Liebovich. The form and dynamics of Langmuir circulations. *Annual Review of Fluid Mechanics*, 15:391–427, 1983.

[21] M. Loève. *Probability Theory.* Van Nostrand, 1955.

[22] E.N. Lorenz. Empirical orthogonal functions and statistical weather prediction. In *Statistical Forecasting Project*, Cambridge, MA, 1956. MIT Press.

[23] J. L. Lumley. The structure of inhomogeneous turbulence. In V. I. Tatarski and A. M. Yaglom, editors, *Atmospheric Turbulence and Wave Propagation*, pages 166–78. Nauka, Moscow, 1967.

[24] B. C. Moore. Principal component analysis in linear system: controllability, observability and model reduction. *IEEE Transactions on Automatic Control*, 26:17–32, 1981.

[25] M. Morzyński, B. R. Noack, and B. Tadmor. Actuation models and dissipative control in empirical Galerkin models of fluid flows. In *The 2004 American Control Conference*, pages 0001–0006, Boston, MA, U.S.A., June 30–July 2, 2004, 2004. Paper **FrP15.6**.

[26] Morten Brøns, Erik A. Christensen, and Jens N. Sørensen. Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows. *SIAM Journal of Scientific Computing*, 21(4):1419–34, 2000.

[27] L. Sirovich. Turbulence and the dynamics of coherent structures, part I: Coherent structures. *Quarterly Applied Mathematics*, 45(3):561–571, 1987.