University of California
Los Angeles

POD Investigations

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Frances Henrietta Hammock

2013

Abstract of the Dissertation

POD Investigations

by

Frances Henrietta Hammock

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2013

Professor Christopher Anderson, Chair

Large-scale dynamical systems are an intrinsic part of many areas of science and engineering. Frequently, they are simulated using high-dimensional discretized equations whose accurate solutions often can only be obtained at very high computational cost. For this reason, there has been a lot of research on the development of reduced order models (ROM's) in the last few decades. Ideally, these models are low-dimensional but still manage to replicate the important characteristics of the system. One popular model is POD which provides the optimally ordered, orthonormal basis from a set of data. A reduced order model is then constructed by representing the solution using a subset of this basis, e.g. projecting the problem onto a lower dimensional subspace in order to obtain a lower dimensional model. In this thesis, we explore both how to evaluate the response of a POD model to perturbations and how to improve its ability to handle perturbations. We also investigate the use of sensitivity analysis on POD reduced order models. Additionally, we look at the POD method's dependence on the type of system being modeled, and ways to improve computational efficiency.

We first present results comparing the outcomes of using essentially the same POD basis for different types of PDE's that demonstrate the accuracy of the POD model depends not only on the ability of the POD basis to represent the solution as set by the tolerance used in the POD selection procedure, but also on the type of equation. Additionally, the results demonstrate that the way in which inaccuracies arise in POD reduced order models are dependent on the type of equation being modeled.

Next, we introduce an error indicator that detects when the POD reduced order model stops performing well on a perturbed system as the numerical solution evolves. This has the advantage that the POD basis can be adjusted as needed during the evolution of the ROM. The error indicator is shown to effectively detect the error as it arises during the evolution of the model on four different types of PDE's. Additionally, we found that sensitivity analysis applied to POD models with significantly reduced dimension compared to their full dimensional counterparts still generated accurate sensitivities.

We address the related issue of how to improve the ability of a POD model to handle perturbations in its parameters. To do this, we introduce a number of basis augmentation strategies. Our tests of these strategies demonstrate that using augmented bases can substantially improve the accuracy of POD reduced order models on perturbed systems.

Lastly, we consider how to improve the computational efficiency of POD. We propose that the implicit filtering associated with POD enables the use of larger timesteps in the POD model. Our computational experiments support this idea that POD based ROM's will generally have reduced stability timestep restrictions. The consequence of this observation is that the ability to use a larger timestep will typically lead to greater computational efficiency, and thus help to offset the additional cost per timestep that evolving a POD based ROM requires for problems with nonlinear terms that cannot be precomputed.

The dissertation of Frances Henrietta Hammock is approved.

Alan Laub

Robert Fovell

Luminita Vese

Christopher Anderson, Committee Chair

University of California, Los Angeles

2013

To my friends and family

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Chris Anderson, for all his advice, support, patience, kindness, and encouragement over the years–you have been a great mentor. I would also like to thank my committee members: Luminita Vese, Robert Fovell, and Alan Laub for their guidance and suggestions. Lastly, thank you to my friends and family. This work was partially supported by a NSF VIGRE Fellowship.

Vita

2004	B.S. (Applied Mathematics), UCB, Berkeley, California.
2006	M.A. (Mathematics), UCLA, Los Angeles, California.
2005–2009	NSF VIGRE Fellowship, UCLA.
2005–2011	Teaching Assistant, Mathematics Department, UCLA.

CHAPTER 1

Introduction

1.1 Background

Large-scale dynamical systems are an intrinsic part of many areas of science and engineering. Frequently, they are simulated using high-dimensional discretizated equations whose accurate solutions often can only be obtained at very high computational cost. For this reason, there has been a lot of research on the development of reduced order models (ROM's) in the last few decades. Ideally, these models are low-dimensional but still manage to replicate the important characteristics of the system [Pin08]. In this thesis, we will focus on a specific type of reduced order model known as proper orthogonal decomposition (POD).

The original idea for proper orthogonal decomposition (POD), was from Pearson [Pea01]. POD is a valuable tool used in a wide variety of fields such as optimal control of partial differential equations ([KV02], [KV01], [KV99], [Rav02]) and inverse problems in structural dynamics [BJW00]. It is known by a number of different names depending on the field of research. In fluid dynamics and meteorology, it goes by empirical orthogonal functions ([ECS97], [Sel97]), in statistics, it is called principal component analysis ([BJS02], [HPS05]), while in signal analysis, it is known as Karhunen-Loueve decomposition ([MM03], [LMM09]).

POD is a projection method that provides the optimally ordered, orthonormal basis (in a least squares sense) of a given size, n, from a set of data [HJB98]. The basis preserves, on average, the greatest amount of kinetic energy possible from the data set used to create it. A reduced order model is then constructed by representing the solution using a subset of this basis, e.g. projecting the problem onto a lower dimensional subspace in order to obtain a lower dimensional model [Pea01]. Often, a surprising amount of information about the system

is preserved in a small number of basis elements. Frequently, the low order model enables the researcher to determine information about the system which would otherwise have been impossible.

There are a number of methods for selecting the data used to make the POD basis. In this thesis, we use the most prominent technique ([Pea01])–the method of snapshots. This was introduced by Sirovich [Sir87] and involves constructing the basis from time snapshots taken from the numerical simulation. It is clear that the use of a basis formed from snapshots enables one to reproduce the solution used to generate them. It is unclear, however, how to assess the accuracy of the reduced order model they are used to create, particularly the ability of the reduced order model to accurately capture nearby (or perturbed) solutions. Furthermore, when the collection of snapshots does not provide a rich enough basis to capture nearby solutions, it is unclear how to augment the basis inorder to capture nearby behavior. In this thesis, we will be addressing both of these questions.

While POD can be used for both ODE's and semi-discretized partial differential equations (PDE's), in this thesis, we focus on the use of POD methods for PDE's. Dimensional reduction of discretized PDE's is incredibly important in many large scale applications such as atmospheric [LP08] and neuron [Cha11] modeling. In the following (1.1.1), we review the construction of a POD reduced order model using the method of snapshots.

1.1.1 POD Reduced Order Model

Consider the dynamical system ,

$$\frac{dy}{dt} = f(y,t), \qquad y(t_0) = y_0 \tag{1.1}$$

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, $t \in [t_0, t_f]$ and $y, y_0 \in \mathbb{R}^n$. The initial condition at t_0 is y_0. This system corresponds to either an ordinary differential equation (ODE), or a semi-discretized partial differential equation (PDE). Throughout the thesis, we will be applying POD to PDE's by using finite difference (FD) discretization to convert the PDE's to ODE's using method of lines. In this context, the dimension, n, is the number of spatial grid points used to form the FD discretization.

Let $Y = [y(t_1), ..., y(t_m)] \in \mathbb{R}^n \times \mathbb{R}^m$ represent the matrix of snapshots collected over the time interval $[t_0, t_f]$. POD is used to find the projection matrix, P, onto a subspace of dimension, k, that minimizes

$$\sum_{i=1}^{m} \|y(t_i) - Py(t_i)\|^2 = \|Y - PY\|^2 \tag{1.2}$$

Given the singular value decomposition (SVD) [Atk88] of Y: $V^T Y U = \Sigma$ with $\sigma_1 \geq \sigma_1 \geq ... \geq \sigma_m \geq 0$ being the ordered singular values of Y, then the minimal value of 1.2 over all k-dimensional subspaces is

$$\sum_{i=k+1}^{m} \sigma_i^2$$

([Vol08]). The corresponding POD projection matrix is the product, $P = \tilde{V}\tilde{V}^T$, where $\tilde{V} = [v_1, ..., v_k] \in \mathbb{R}^n \times R^k$. Here, $v_1, ..., v_k$ are the first k, left singular vectors of the snapshot matrix, Y.

Using the POD matrix, we can construct the reduced order model of 1.1 in full dimensional space

$$\frac{d\tilde{y}}{dt} = Pf(\tilde{y}, t) \quad \tilde{y}(t_0) = Py_0 \tag{1.3}$$

or in the subspace spanned by k largest, left singular vectors

$$\frac{dz}{dt} = \tilde{V}^T f(\tilde{V}z, t) \quad z(0) = \tilde{V}^T y_0 \tag{1.4}$$

Additional information on POD can be found in [HPS05], [Cha11], and [Pea01].

1.2 Thesis Overview

The thesis is organized as follows. In Chapter 2, we look at two basic questions about POD. First, we examine the degree to which the success of the POD reduced order model depends on the type of equation being used. We find that although the procedure for constructing the POD basis from snapshots is independent of the equation (e.g. it is based on the SVD), the accuracy of the corresponding reduced order model is not. We present results comparing the outcomes of using essentially the same POD basis for different types of PDE's that demonstrate the accuracy of the POD model depends not only on the ability of the POD basis to represent

the solution as set by the tolerance used in the POD selection procedure, but also on the type of equation. Additionally, the results demonstrate that the way in which inaccuracies arise in POD reduced order models are dependent on the type of equation being modeled.

One of the most important questions related to the use of a POD model is how to evaluate its ability to handle variations in problem parameters and initial conditions. Although it would be best to know this before applying the a POD model to perturbed systems, it is difficult to predict consistently (particularly for anything not local). Chapter 3 introduces an alternative approach in which we use an error indicator to detect when the POD reduced order model stops performing well on a perturbed system as the numerical solution evolves. This has the advantage that the POD basis can be adjusted as needed during the evolution. The error indicator that we propose is based upon a comparison of the relative $\|.\|_2$ error due to the projection of the operator between the original system and the perturbed system. We apply it to numerical experiments on four different PDE's with a variety of perturbations.

The numerical results from Chapters 2 and 3 reflect that reaction-diffusion type equations are particularly challenging to model using POD, so we focus on these types of equations in the subsequent two chapters. Ideally, one would like to be able to not only accurately reproduce the solution whose snapshots give rise to the POD basis, but also to accurately reproduce the solutions and sensitivities associated with perturbed initial conditions or model parameters. We introduce some new approaches to acheive these goals in Chapters 4 and 5.

In Chapter 4, we investigate how to carry out sensitivity analysis efficiently by applying it to POD reduced order models on reaction-diffusion type equations. Next, in Chapter 5, we introduce a variety of strategies for basis augmentation. We investigate both how best to pick components of perturbed systems in order to improve the POD basis, and how to improve the computational efficiency of the calculations.

A major limitation of POD models is their size, especially their inability to reduce the computational complexity of nonlinear terms. Recently, there has been a lot of research to improve the efficiency of the ROM with regard to the nonlinear terms. In Chapter 6, we look at an additional way to improve the efficiency. We find that using a POD reduced order model

relaxes the stability requirement on the numerical examples considered. This enables one to take coarser time steps in the ROM compared to the original system.

CHAPTER 2

An Investigation Into The Equation Dependence of POD Model Accuracy

As mentioned in the introduction, the POD basis is commonly determined by creating a collection of basis vectors from the singular value decomposition of the snapshots. A tolerance is then picked for the total square distance between the original solution snapshots and their projection onto the POD subspace under the assumption that the error in the solution, computed using the POD basis, will have an error close to this tolerance. As we demonstrate in this chapter, this assumption is not necessarily a vaild one. Specifically, we present results comparing the outcomes of using essentially the same POD basis for different types of PDE's. The results demonstrate that, for the systems considered, the accuracy of the POD method depends not only on the ability of the POD basis to represent the solution as set by the tolerance used in the POD selection procedure, but also on the type of equation. The general experimental design is presented in Section 2.1. This is followed by the numerical details in 2.2. We then examine the effects of POD induced dimensional reduction for the three types of equations with similar solutions 2.2.1.

2.1 Set Up

We investigate a basic question about the proper orthogonal decomposition (POD) method as it is applied to three different types of PDE's. Namely, is the success of POD dependent only on the form of the solution, or on the type of equation itself.

In order to answer this question, we apply POD to three standard types of 1-D PDE's: a PDE for reaction-diffusion,a PDE for transport, and a hyperbolic conservation law exhibiting

a shock.

Reaction-Diffusion

$$\frac{dy}{dt} = ay^2(2-y) + by_{xx} \tag{2.1}$$

$$y(0,t) = 2,\ y(2,t) = 0,\ y(x,0) = 0$$

where a and b are constants.

Transport

$$\frac{dy}{dt} = -y_x \tag{2.2}$$

$$y(0,t) = 2,\ y(x,0) = 0$$

Hyperbolic conservation law (Burgers' Equation)

$$\frac{dy}{dt} = -(y^2/2)_x \tag{2.3}$$

$$y(0,t) = 2,\ y(x,0) = 0$$

For all three equations, $t \in [0,1]$, and $x \in [0,2]$. The equation parameters and numerical parameters are selected such that the equations have nearly identical numerical solutions which consist of a single moving front moving from left to right. Since the solution snapshots are nearly identical, doing singular value analysis on the time slices will produce very similar singular values and vectors despite the equations all being different. Thus, the POD bases formed from the singular value decompositions of the snapshots of the solutions will yield almost identical POD bases for all three equations.

The exact solutions to the shock and transport equations (2.3 and 2.2) are identical: fronts moving to the right with ninety degree angles at a height of 2 and a speed of 1. A numerical solution, however, is highly dependent on both the type of equation, and the method used to solve it. For example, the fronts generated by using the Lax-Friedrichs method for the transport equation are diffuse (unless a very fine grid is used) because the corresponding discretization of the y_x term looks more like the discretization of $y_x - \varepsilon y_{xx}$. In contrast, if ENO ([SO88], [Fed96]) is used to solve Burgers' equation, then the fronts are almost vertical.

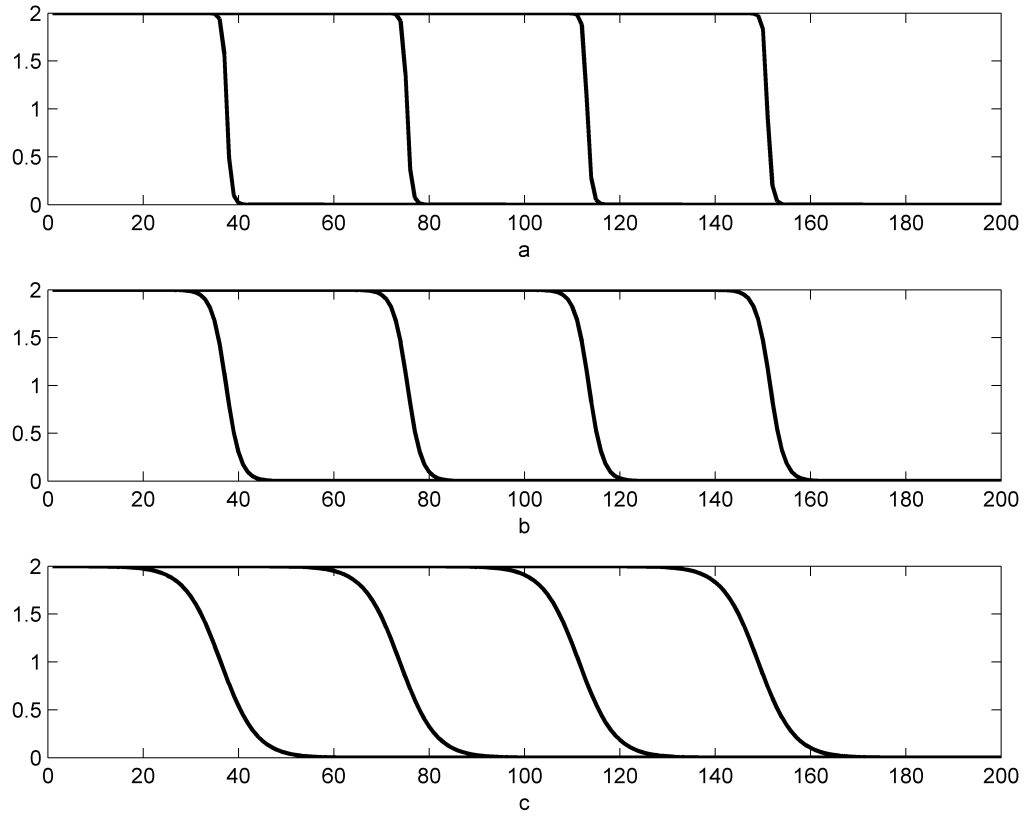

Figure 2.1: The three types of moving fronts listed from left to right in order of decreasing slope. The solutions are shown for $t = \{0.25, 0.5, 0.75, 1.0\}$.

Because we want to investigate how the equations behave with essentially the same POD basis, we used standard PDE methods and parameters so that the numerical solutions obtained for each equation were approximately equal, rather than using methods and parameters which created the most accurate solution for each equation.

In particular, for each equation we select a numerical method and problem parameters so that the resulting computed solutions consist of a moving front with a leading edge which we classify as: steep, medium diffuse, and most diffuse. The three different types of fronts are shown in Fig. 2.1.

In Fig. 2.2, we show the first 100 singular values for the three different types of fronts. There is a large difference in the rate of decay of the singular values between the different types

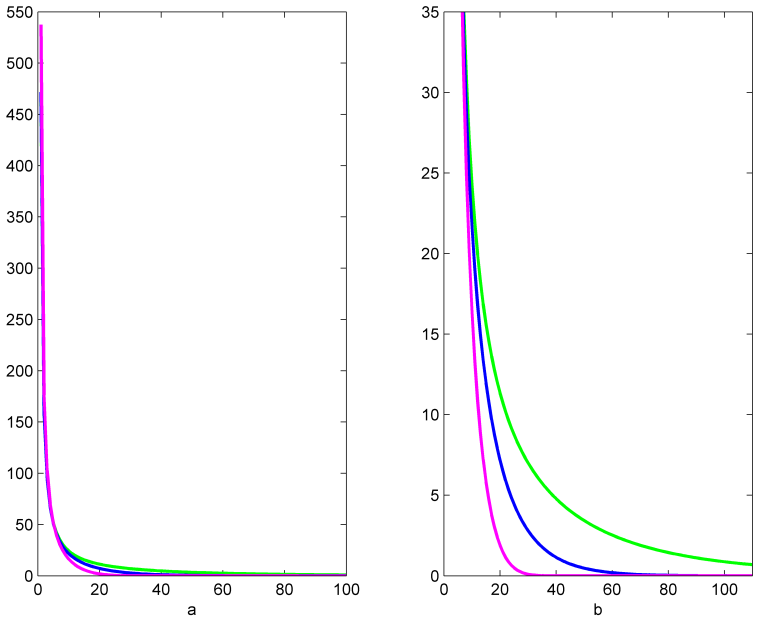

Figure 2.2: The singular values for the three different types of fronts. b) provides a close up of the singular values where their behavior differs the most. The following colors correspond to the different slopes: green for the steepest, blue for medium diffuse, and magenta for the most diffuse.

of fronts. This is expected since the steeper the slope, the more the differences between the snapshots resemble the unit basis and thus a larger number of basis elements are necessary to acheive good resolution of the fronts. So, the steeper the slopes, the more basis elements are required for good resolution of the fronts.

2.2 Numerical Results

We first go through the numerical methods and discretizations used, followed by a comparison of the effects of dimensional reduction on the POD reduced order models for the three types of equations in Section 2.2.1.

The numerical details for each type of front and equation are listed below:

- Steepest Fronts

For this case, transport 2.2 is solved using 3rd order ENO discretization in space and 3rd order Runge Kutta in time. We solve the 1-D hyperbolic conservation law, 2.3, using a 3rd order ENO approximation of the convection term and 3rd order Runge Kutta in time. 2nd order central differencing is used for reaction-diffusion with 3rd order Runge Kutta in time. In order to get steep fronts with the correct speed, the reaction and diffusion parameters in equation 2.1 are selected to be $a = 180.0$ and $b = .003$. We evolve the shock and reaction-diffusion systems until time $t = 1.0$ using 900 time steps on a grid of size 300. In order to get the transport fronts to be sufficiently steep, it was necessary to use 1200 steps on a grid of size 900. At the final time step, there is a relative $\|.\|_2$ difference of 2.5% between reaction-diffusion and the hyperbolic conservation law. Between reaction-diffusion and transport, the relative $\|.\|_2$ difference is 2.3% and between shock and transport, it is 4.9%

- Medium Diffuse Fronts

  We use central differencing for the reaction-diffusion equation and select the reaction and diffusion terms to be $a = 51.0$ and $b = .0101$. We solve transport, 2.2, with 3rd order ENO discretization of the advection term. The fronts are more diffuse then in the steepest cases because a less fine grid is used. We add a diffusion term, εu_{xx} where $\varepsilon = .01$, to the 1-D hyperbolic conservation law (2.3) in order to get the fronts to have medium diffuse. The diffusion term is dealt with using 2nd order central differencing. We evolve the systems until $t = 1.0$ using 900 time steps with 3rd order Runge Kutta on a grid of size 300. At the final time step, we get a relative, $\|.\|_2$ difference of .1% between the the reaction-diffusion and hyperbolic conservation law solutions. Between the transport and shock solutions, the relative $\|.\|_2$ difference is 4.3% and 5.1% between transport and reaction-diffusion.

- Most Diffuse Fronts

  Here, 2.2 is solved using Lax Friedrichs. We add a larger diffusion term, εu_{xx}, to the 1-D hyperbolic conservation law (2.3) with $\varepsilon = .022$ in order to get the fronts to have diffuse fronts. The diffusion term is approximated with 2nd order central differencing. We also use central differencing for the reaction-diffusion equation. The coefficients of the reaction

and diffusion terms are selected to be $a = 20.0$ and $b = .025$ respectively.

We evolve the systems until $t = 1.0$ using 1200 time steps with 3rd order Runge Kutta (smaller time steps are required because of the increased amount of diffusion) on a grid of size 300. At the final time step, we get a relative, $\|.\|_2$ difference of 6.3% between the reaction-diffusion and hyperbolic conservation law solutions. Between the transport and shock solutions, the relative $\|.\|_2$ difference is 5.5%. Between transport and reaction-diffusion the relative $\|.\|_2$ difference is 3.0%.

For the range of model parameters and equations used, an explicit, 3rd order Runge-Kutta was sufficient to ensure that the numerical solution could be stably computed with an acceptable timestep. It was not necessary to use an implicit method because the time step restriction is not as severe as in the case of pure diffusion (the time scales involved in both reaction-diffusion and advection-diffusion are different than pure diffusion), and, when a diffusion term was present, its' coefficient was small.

2.2.1 Comparison of dimension reduction for the three types of equations

We would like to answer the following two questions using these results. Firstly, how does the accuracy of the POD model depend on the particular PDE when using essentially the same basis? Secondly, what is the nature of the inaccuracies that arise?

To answer the first question, we looked at the errors generated by the reduced systems for the different equations and slopes as the size of the POD basis is varied. In Figures 2.3- 2.5 below, results are shown for each equation for each of the three types of slopes. The relative, $\|.\|_2$ error between the full dimensional solution, and the POD reduced order model solution are plotted as a function of the dimension of the POD basis.

It is clear from the figures that reaction-diffusion's behavior differs significantly from the other two equations. As the size of the POD basis is decreased, the error grows much faster for the reaction-diffusion equation than for either shock or transport. For example, with a basis of dimension 80 for the steepest fronts, reaction-diffusion has a relative, $\|.\|_2$ error of 31% as opposed to the conservation law which has a relative error of 4.7% and transport, which has a

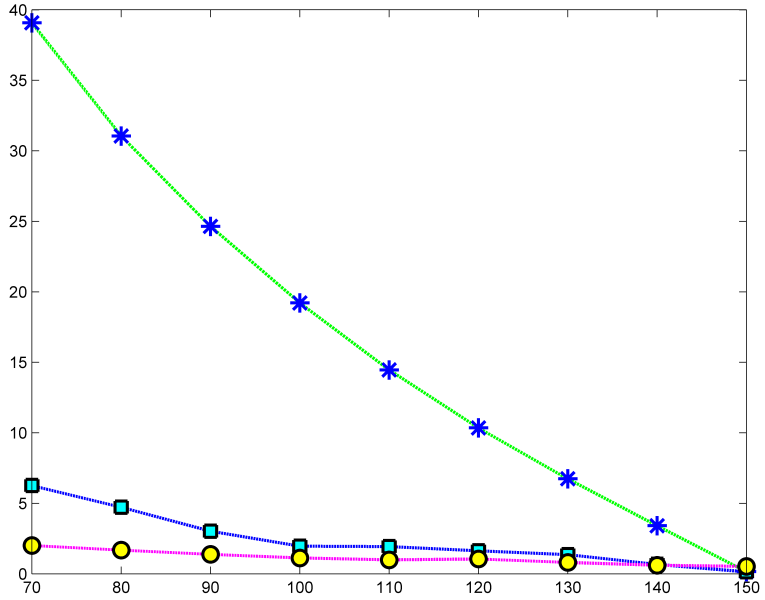

Figure 2.3: The relative, $\|.\|_2$ error between full dimensional solution and the POD reduced order model recorded in percentages as a function of the size of the POD basis. The results are for the fronts with the steepest slopes. The following markers correspond to the different equations: stars for reaction-diffusion, circles for transport, and squares for shock.

relative error 1.7%. Similarly, for the medium diffuse case, using a basis of dimension 20 results in errors of 35.9% for reaction-diffusion while only 4.3% and 5.2% for shock and transport respectively. Finally, for the diffuse fronts, using 6 POD basis elements generated errors of 13.5% and 11.1% for shock and transport respectively. In contrast, reaction-diffusion was off by 47.3%.

Although the fronts are not exactly identical for the different types of equations for each type of slope, the striking difference in the behavior of the error for reaction-diffusion compared to the hyperbolic conservations law and transport makes it clear that a POD reduced order model of reaction-diffusion behaves very differently than a POD reduced order model for the other two equations. The fact that overall, more basis elements are needed for steeper fronts, regardless of the type of equation, is due to the SVD producing many more nontrivial singular values as shown in Fig. 2.2. For each equation, the more diffusion (whether by an added term

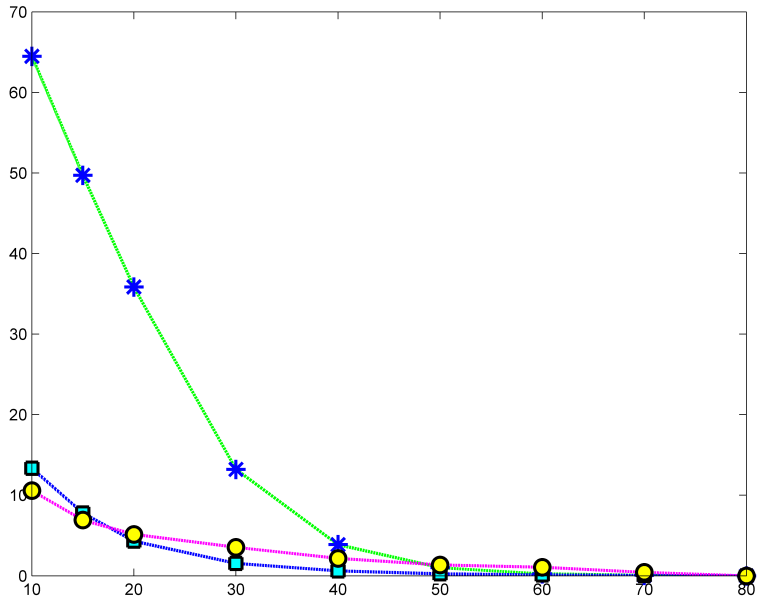

Figure 2.4: The relative, $\|.\|_2$ error between full dimensional solution and the POD reduced order model recorded in percentages as a function of the size of the POD basis. The results are for the fronts with medium diffuse slopes. The following markers correspond to the different equations: stars for reaction-diffusion, squares for shock, and circles for transport.

or inherent to the numerical method such as with transport solved with Lax Friedrichs), the better the POD method worked

The errors associated with a reduced order model for the hyperbolic conservation law and transport did not differ as significantly in these experiments as with reaction-diffusion. The relaxation term added to the shock equation in the medium and most diffuse cases improved the success of the POD basis. It caused the ROM to behave very similarly to transport in both the medium and most diffuse cases. This result is an apparent contradiction to work showing that models which contain shocks are only successfully represented by a POD reduced order model if one of the snapshots used to build the model had a discontinuity in the same location as the shock [LKO01]. This generally requires a large number of snapshots to achieve acceptable accuracy in the final solution. The resolution to this contradiction is that the addition of diffusion renders the shock profile more diffuse and causes the size of its POD basis necessary

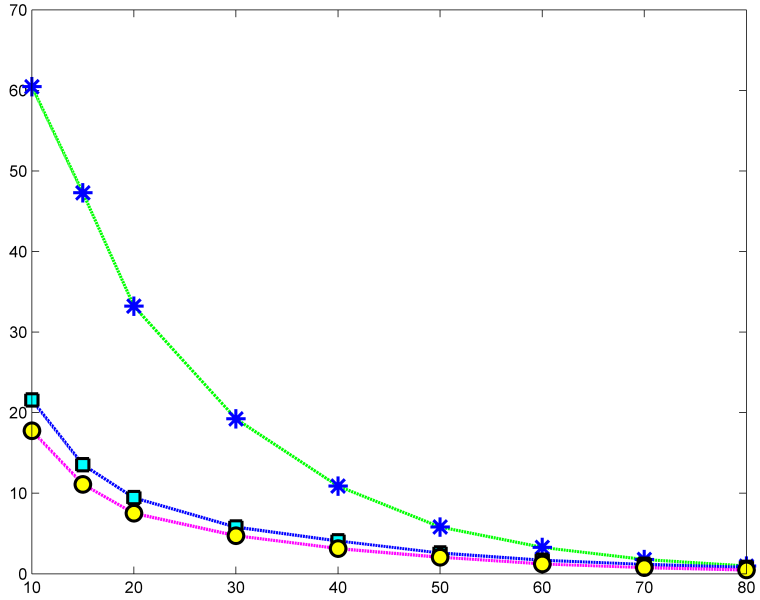

Figure 2.5: The relative, $\|.\|_2$ error between full dimensional solution and the POD reduced order model recorded in percentages as a function of the size of the POD basis. The results are for the fronts with the most diffuse slopes. The following markers correspond to the different equations: stars for reaction-diffusion, squares for shock, and circles for transport.

to achieve a given accuracy to be very similar to transport's.

However, we found that when no diffusion was added to the shock equation, the relative, $\|.\|_2$ errors were generally between 2 and 3 times larger with the shock equation than with the transport equation (here, the $\|.\|_2$ errors are assumed to be $> 1\%$) for the case with the steepest fronts. This is consistent with the previous literature and again demonstrates that different equations with essentially the same solutions respond differently to the same bases.

We demonstrated above that the success of POD can be very subtle. It depends not only on the ability of the POD basis to resolve the original snapshots, but on the type of equation itself.

Next, we consider the second question: what is the nature of the inaccuracies that arise when using POD on the different types of PDE's? We found that not only did the magnitude of the error (as the dimension of the basis was reduced) differ greatly between reaction-diffusion and

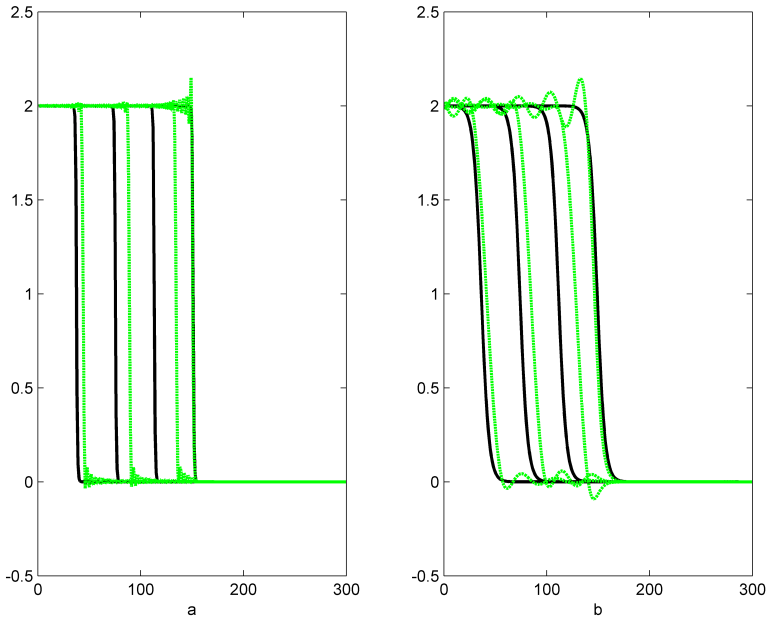

Figure 2.6: Solutions of both the full dimensional and reduced order model of the reaction-diffusion equation for the steepest and most diffuse cases are shown at $t = \{0.25, 0.5, 0.75, 1.0\}$. The green dashed lines show the POD solution. a)The POD basis has size 80 and the solutions correspond to the steepest slope. b) The POD basis has size 10 and the solutions correspond to the most diffuse slope.

the other two equations, but the way in which the solutions deteriorated also differed greatly. For both the hyperbolic conservation law and transport, the shape and speed were generally preserved as the size of the POD basis was reduced. The majority of the error for these equations comes from the oscillations (these are easily removed with some sort of post processing such as thresholding). This is shown in Figs. 2.7 and 2.8.

In contrast, for all three types of slopes, reaction-diffusion formed only some small oscillations as can be seen in Fig. 2.6. These contributed very little to the final error, unlike the other two equations. The bulk of the error, regardless of the steepness of the fronts, is due to the fact that as the dimension of the basis is reduced, the fronts move at dramatically different speeds–the smaller the basis, the more incorrect the speeds. Not only does this cause the error to grow (as the basis dimension is decreased) much faster than with the other two equations,

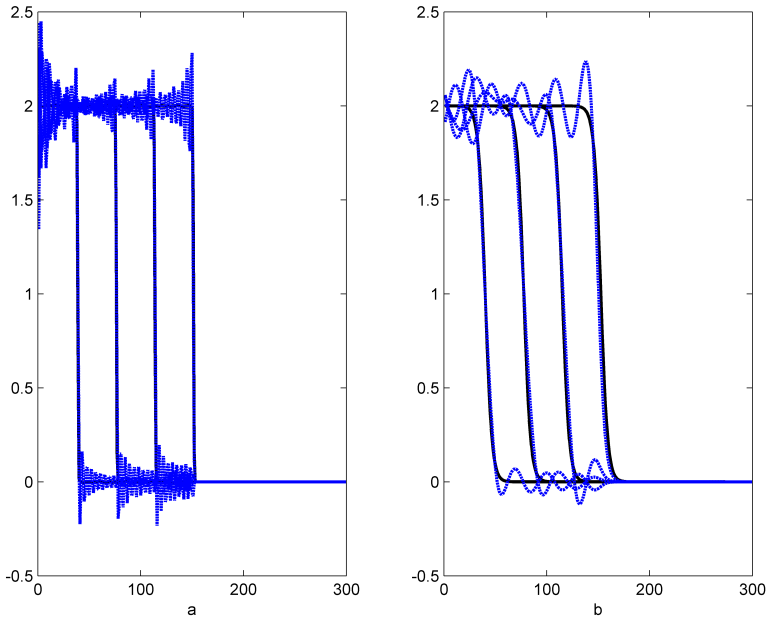

Figure 2.7: Solutions of both the full dimensional and reduced order model of the shock equation for the steepest and most diffuse cases shown at $t = \{0.25, 0.5, 0.75, 1.0\}$. The blue dashed lines show the POD solution. a)The POD basis has size 70 for the steepest slope and the solutions correspond to the steepest slope. b)The POD basis has size 10 and the solutions correspond to the most diffuse slope.

but it is difficult to detect visually (as opposed to something like spatial oscillations).

The results indicate that POD models for reaction-diffusion behave very differently when compared to either transport or the hyperbolic conservation law even though essentially the same POD basis is used for the three different types of fronts. Not only does its accuracy deteriorate much more rapidly as the basis size is diminished, but the nature of the errors are much more problematic than those that developed with the other two equations. The behavior of the shock and transport equations also differed significantly when the fronts were steepest. These results demonstrate that even if different equations have nearly identical solutions and POD bases, the accuracy that can be obtained with a POD reduced order model will vary greatly. Further, they indicate that it is difficult to predict the quality of a POD reduced order model based on a threshold used in selecting the POD basis.

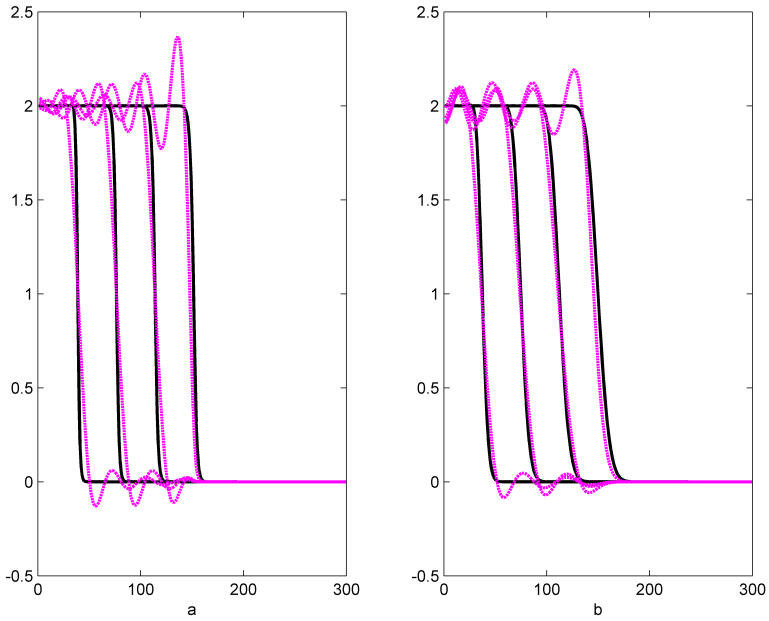

Figure 2.8: Solutions of both the full dimensional and reduced order model of the transport equation for the medium diffuse and most diffuse cases shown at $t = \{0.25, 0.5, 0.75, 1.0\}$. The purple dashed lines shows the POD solution. a)The POD basis has size 10 and the solutions correspond to the medium diffuse slope. b)The POD basis has size 8 and the solutions correspond to the most diffuse slope.

2.3 Conclusion

By comparing the effects of applying POD to three standard PDE's with essentially the same POD bases, we showed that the accuracy of a POD model is equation dependent. The accuracy of the POD model for reaction-diffusion equation was strikingly different when compared to the hyperbolic conservation law and transport. Similarly, in the case of the steepest fronts, the hyperbolic conservation law equation generated relative $\|.\|_2$ errors 2 to 3 times larger than transport.

With reaction-diffusion, not only did the magnitude of the error (as the dimension of the basis was reduced) grow much faster than with the other equations, but the way in which the solution deteriorated was also distinctive. Although there were not many oscillations and the

shape of the fronts were preserved, the solutions at different times went at different speeds if the POD basis was too small. This type of error would be challenging to detect numerically if the true solution is not already known. It furthermore suggests that applying a POD basis derived for specific parameters in a reaction-diffusion system to the same system, but with perturbed parameters, must be done with caution. Additionally, reaction-diffusion's poor performance when compared to the other two equations highlights the importance of equation specific corrections for POD reduced order models.

CHAPTER 3

Dynamic Error Detection for POD Reduced Order Models

The number of singular vectors included in a POD basis is commonly determined by picking a threshold for the error in the representation of the solution snapshots for a specific solution by the POD basis. One can ascertain an estimate of this threshold based on seeing how well the reduced order model based on this POD basis reproduces the solution that provided the basis snapshots. However, such estimates are not necessarily reliable indicators of the performance of a POD model when they are used to create nearby or "perturbed" solutions. Moreover, any theoretical results will likely be strongly equation dependent (see Chapter 2). In the absence of theoretical a-priori estimates of the error associated with a model solution, it is very useful to have a computational procedure that indicates when a model solution is likely to have a significant error.

This chapter introduces metrics that can be used to detect error as it arises in POD reduced order models where the snapshots are taken from systems which are different from the system under consideration. This error indicator has the advantage that it does not depend on specific inequalities or linearization. In Section 3.1, the metrics are introduced. In Section 3.2, numerical experiments are carried out on the 1-D reaction-diffusion equation, 1-D transport equation, 1-D Fitz-Hugh Nagumo model, and 2-D nondivergent barotropic vorticity equation. A variety of perturbations are applied to the PDE's and the metrics are used to see how sensitive they are to errors in the PDE's ROM. We find that the behavior of the metrics accurately mimics the behavior of the true relative error in the PDE.

3.1 Error Detection

One of the most important questions related to the use of POD is how to evaluate the quality of the approximation. For non-linear systems, little work on error estimation of POD reduced order models can be found in the literature. In [UCS85], the authors present a method for computing the errors of a variety of model reduction methods. The technique relies on first-order estimates around the initial time which are typically valid for only a short period of time. In order to evaluate if a POD reduced order model is able to successfully represent a shock in a given location, [LKO01] computes the first order change in the solution due to the augmentation of the original POD basis with top hat basis functions. The results are then compared to the accuracy of a known solution such as one of the snapshots used to create the POD basis. In [MM03] , Meyer and Matthies use the dual-weighted-residual method to obtain estimates of the error due to the reduced order model for a specific functional using an adjoint system.

K. Kunisch and S. Volkwein introduce error estimates for POD approximation methods for a class of nonlinear parabolic equations arising in fluid dynamics in [KV02] and [KV01]. Their approximation errors were composed of the time discretization error (due to backwards Euler) and the POD subspace approximation error. The estimates rely on inequalities, however, that may not be satisfied for other types of equations. Additionally, they require that the snapshots used to make the POD approximation are taken from the system being analyzed.

In [SPH07], Serban, Petzold, and Homescu analyze the approximation properties of POD based schemes where the snapshots are taken from systems which are different from the system under consideration. This is an important tool for applications such as control and inverse problems where the POD basis is used both to approximate the original system's solution, and to model systems perturbed from the original one. They introduce an error estimation technique based on the combination of the adjoint method with small sample statistical condition estimation. The approach determines ranges of perturbations in the original system over which the reduced order model is still appropriate without it being necessary to solve the perturbed system. Their analysis has somewhat limited scope because it is based on linearization. This

means that the perturbations must be sufficiently small and the models run only as long as the linearization remains accurate.

Ideally, one would like a POD basis to reproduce the dynamics of the full system for variations in problem parameters and initial conditions. It would be great to know how well the ROM will perform for perturbed parameters without having to evolve the full, perturbed system until the final time. It is, however, difficult to predict the accuracy of the ROM solution consistently (particularly for anything not local). Another approach is to use an error indicator to detect when the POD basis stops performing well on a perturbed system as the numerical solution evolves. This has the advantage that the basis can be adjusted as needed during the evolution. The error indicator that we propose is based upon a comparison of the relative $\|.\|_2$ error due to the projection of the operator between the original system and the perturbed system. The idea is that when the projection error associated with the reduced model is greater than the average projection error of the original system, then this is a strong indication of significant error in the solution being constructed with the reduced order model.

Specifically, consider the dynamical system

$$\frac{dy}{dt} = f(y, p, t), \qquad y(t_0) = y_0(p) \tag{3.1}$$

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$, $t \in [t_0, t_f]$ and $y, y_0 \in \mathbb{R}^n$. The initial conditions at t_0 are y_0. p are the model parameters. This system corresponds to either an ordinary differential equation (ODE), or a semi-discretized partial differential equation (PDE). Let $y(t)$ and $x(t)$ be particular solutions of 3.1

$$\frac{dy}{dt} = f(y, p, t) \quad y(0) = y_0(p)$$
$$\frac{dx}{dt} = f(x, q, t) \quad x(0) = x_0(q)$$

and p and q be their respective problem parameters. Here, we assume that $y(t)$ is the original system, and $x(t)$ is the perturbed system. Let $P = \tilde{V}\tilde{V}^T \in \mathbb{R}^n \times R^n$ be the POD projection matrix derived from the timeslices of $y(t)$. This projection matrix is then used to make the POD reduced order models for both $y(t)$ and $x(t)$. Let $\tilde{y}(t)$ be the solution of the ROM, 3.2, for $y(t)$

$$\frac{d\tilde{y}}{dt} = Pf(\tilde{y}, p, t) \quad \tilde{y}(0) = Py_0(p) \tag{3.2}$$

and $\tilde{x}(t)$ be the solution of the ROM, 3.3, for $x(t)$.

$$\frac{d\tilde{x}}{dt} = Pf(\tilde{x}, q, t) \quad \tilde{x}(0) = P\tilde{x}_0(q) \tag{3.3}$$

We then evaluate the relative $\|.\|_2$ error due to the projection of the operator from both the original system

$$e_1 = \frac{\|f(\tilde{y}(t_j), p) - Pf(\tilde{y}(t_j), p)\|_2}{\|f(\tilde{y}(t_j), p)\|_2}$$

and the perturbed.

$$e_2 = \frac{\|f(\tilde{x}(t_j), q) - Pf(\tilde{x}(t_j), q)\|_2}{\|f(\tilde{x}(t_j), q)\|_2}$$

Assuming that a tolerance has been chosen so that the POD reduced order model reproduces y sufficiently, then the average value of e_1, $\bar{e}_1$, is an estimate of the size of the error associated with the projection that can occur without adversly effecting the solution accuracy. Thus, by observing the size of e_2 relative to $\bar{e}_1$, we obtain an error indicator. In particular, when $e_2 >> \bar{e}_1$, this strongly suggests that the solution of the perturbed system will be inaccurate. Similarly, if the ratio between e_2 and $\bar{e}_1$ is close to or less than 1, then the perturbed POD model is likely functioning well.

To test our proposed error indicator, we calculate it while creating approximate perturbed solutions to a variety of different types of PDE's using a POD based ROM. In particular, we apply the indicator on 1-D transport and reaction-diffusion. Perturbations are introduced that affect the speed, size, and shape of the fronts. We also apply the metrics to the FitzHugh-Nagumo model, and a 2-D Nondivergent Barotropic Vorticity Equation.

In all the cases considered, this approach turns out to be effective at detecting how well the ROM is performing. e_2 is small like e_1 when the basis is performing well, and its growth is generally proportional to the relative error of the true solution of the perturbed system's ROM. Specifically, we look at the ratio between e_2, and the average value of $\bar{e}_1$ taken over the interval considered. When $\frac{e_2}{\bar{e}_1}$ is around one, the ROM works extremely well, but it deteriorates as the ratio gets larger. As $\frac{e_2}{\bar{e}_1}$ approaches ten, the accuracy of the ROM becomes very inaccurate. In the calculation of $\bar{e}_1$, we ignore transients.

3.2 Numerical Examples

In the examples below, the error indicator is used to evaluate the quality of the ROM under a variety of perturbation for four PDE's. In 3.2.1,the metrics are used to evaluate the 1-D reaction-diffusion equation and the 1-D transport equation in Section 3.2.2. Subsequently, the metrics are applied to the Fitz-Hugh Nagumo equation in Section 3.2.3. Finally, we consider the 2-D nondivergent barotropic vorticity equation 3.2.4.

3.2.1 1-D Reaction-Diffusion

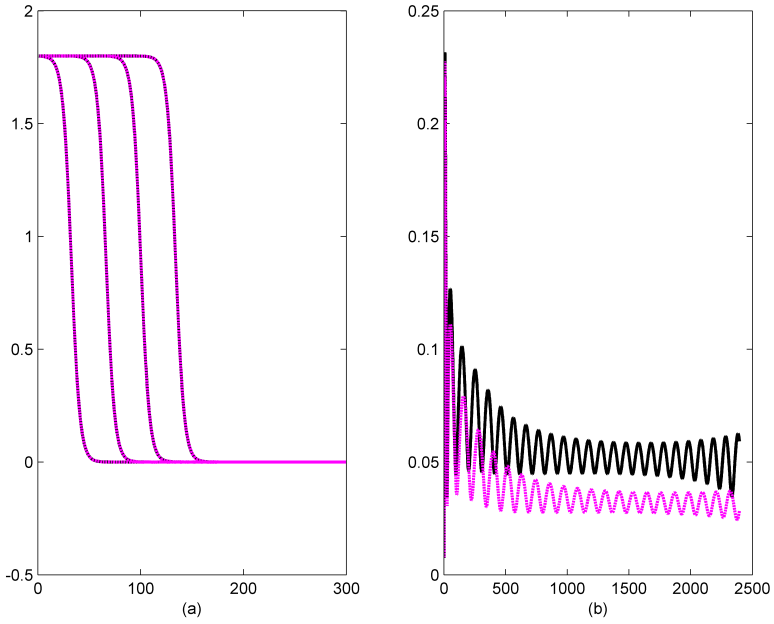

Figure 3.1: The initial condition, $y(0,t) = 2$ is perturbed to 1.8 (a) Black is the full dimensional, perturbed system, purple dashed is the ROM for the perturbed system. The solutions are shown at $t = \{.25, .5, .75, 1.0\}$. (b) e_1 is black, e_2 is purple dashed, both are plotted as a function of the timestep.

Consider the reaction-diffusion equation

$$\frac{dy}{dt} = ay^2(2-y) + by_{xx} \tag{3.4}$$

$$y(0,t) = 2,\ y(2,t) = 0,\ y(x,0) = 0$$

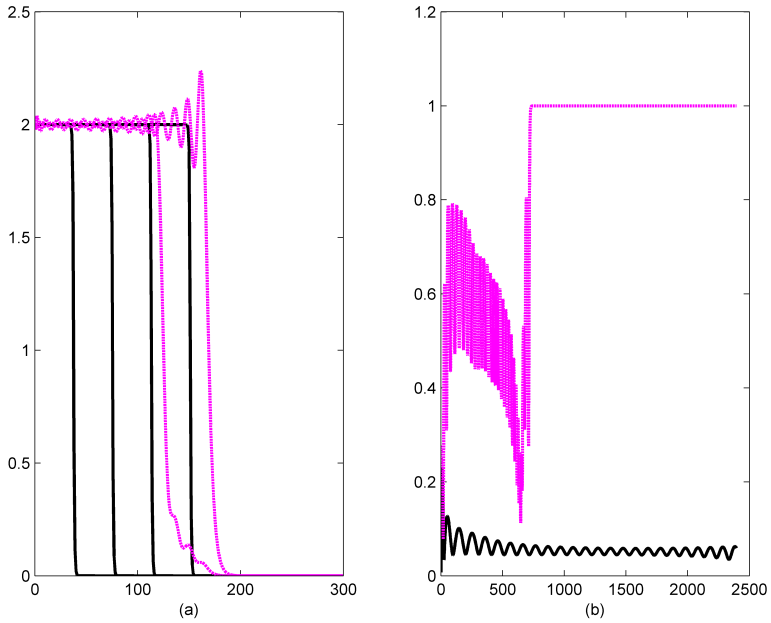

Figure 3.2: The reaction and diffusion coefficients are perturbed to respectively $a = 180.0$, $b = .003$. (a) Black is the full dimensional, perturbed system, purple dashed is the ROM for the perturbed system. The solutions are shown at $t = \{.25, .5, .75, 1.0\}$. (b) e_1 is black, e_2 is purple dashed, both are plotted as a function of the timestep.

We form the POD basis (see Section 2.1 for the numerical details) with the diffusion coefficient, $b = .025$, the reaction coefficient, $a = 20.0$, and the boundary condition $y(0,t) = 2$. Using method of lines, we convert the PDE to an ODE using standard centered difference in space to approximate the spatial derivatives on a grid of 300 points. The resulting ODE has dimension $n = 300$. The POD model is constructed from 2400 equally-spaced snapshots collected over the time $[0, 1.1]$. The solutions are evolved until t=1.0. The projection matrix is constructed from 28 modes. The relative error between the full and reduced models at the final time, $t = 1.0$, is

$$\frac{||y - \tilde{y}||}{||y||} < .0013$$

.

We first perturb the problem by reducing the boundary condition, $y(0,t) = 2$, by .2 which reduces the height of the fronts and decreases their speed. The ROM using the original POD

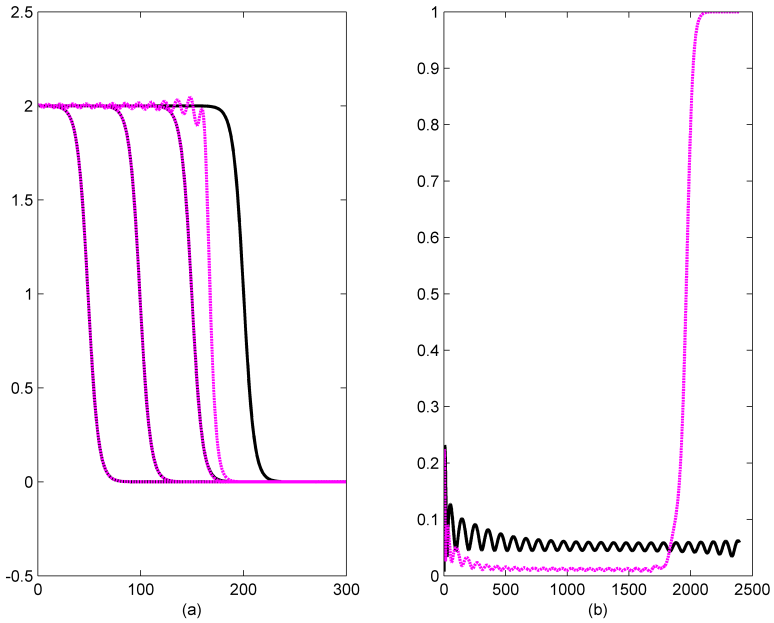

Figure 3.3: Diffusion coefficient increased to $b = .045$ (a) Black is the full, perturbed system, purple dashed is the ROM for the perturbed system. The solutions are shown at $t = \{.25, .5, .75, 1.0\}$. (b) e_1 is black, e_2 is purple dashed, both are plotted as a function of the timestep.

basis captures the solutions of the perturbed system very well (Fig. 3.1 (a)). As shown in Fig. 3.1 (b), e_2 correctly indicates that the POD basis successfully models the perturbed solutions by staying close to e_1 throughout the interval ($\frac{e_2}{\bar{e}_1} \cong 1.5$).

In the second test, we create a perturbed problem by modifying the reaction and diffusion coefficients. This makes the fronts steeper while maintaining their speed by changing the reaction coefficient to $a = 180.0$ and the diffusion coefficient to $b = .003$. Note that this solution does not move faster than the solution to the original system so there is no problem with it overtaking the basis. As shown in Fig. 3.2, e_2 indicates that the ROM using the original basis is unable to model this situation. It shows correctly that the error between the true perturbed solution and its ROM is large until about 650 time steps at which point the ROM completely breaks down and the relative size of e_2 to $\bar{e}_1$ increases dramatically to about 11.

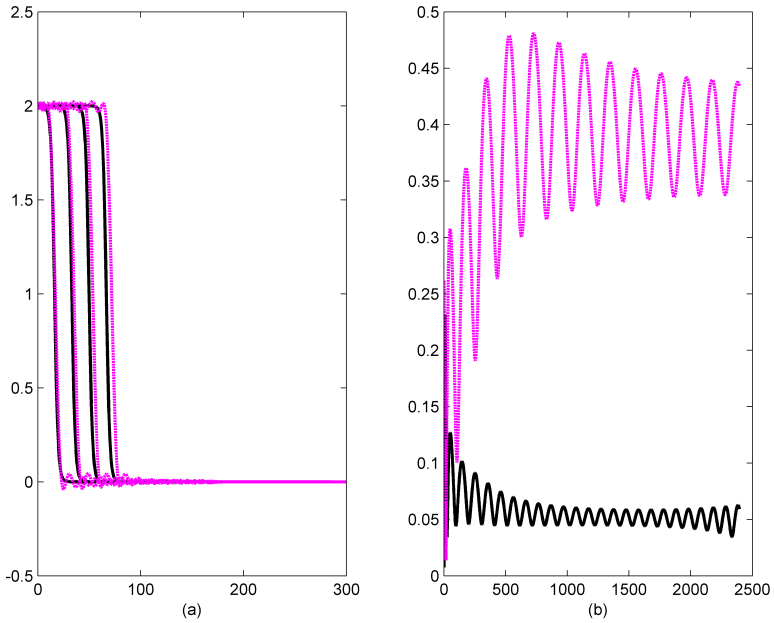

Figure 3.4: The diffusion coefficient is decreased to $b = .005$. (a) Black is the full dimensional, perturbed system, purple dashed is the ROM for the perturbed system. The solutions are shown at $t = \{.25, .5, .75, 1.0\}$. (b) e_1 is black, e_2 is purple dashed, both are plotted as a function of the timestep.

For our third test, we perturb the system by increasing the diffusion to $b = .045$. Increasing diffusion causes the solution to overrun the basis before $t = 1.0$ so the fronts are unable to progress past where the original solution reached. However, before this happens, the fronts are well represented by the ROM. This is reflected by the ratio between e_2 and $\bar{e}_1$ being very small until the perturbed solution reaches the end of the basis at which point the ratio shoots up to 12.5. This is depicted in Fig. 3.3.

Finally, our last test case for this problem consists of decreasing the diffusion to $b = .005$. Reducing diffusion causes the solution both to move more slowly and to make the fronts become steeper. The ROM has trouble because of the increased slope of the fronts, and has a relative error of 19% at $t = 1.0$. This situation is reflected by the metrics in Fig. 3.4 where it shows that $\bar{e}_1$ is around 7% and e_2 is 43%. So, the ratio is about 6, and thus, as expected, gives larger errors than occur with a ratio of 1 or 2, but definitely better than the errors associated with a

perturbed solution when the ratio is about 10.

3.2.2 Transport

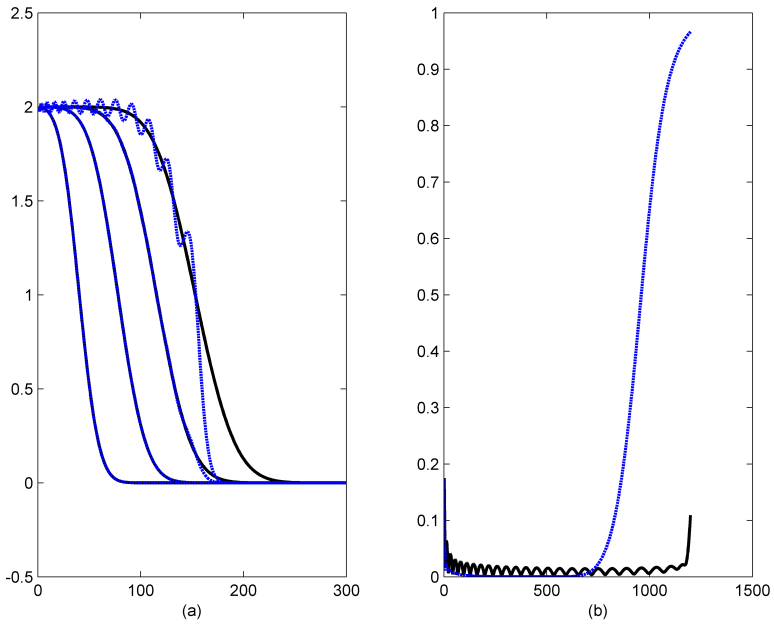

Figure 3.5: A diffusion term with coefficient .04 is added to transport(a) Black is the full dimensional, perturbed system, blue dashed is the ROM for the perturbed system. The solutions are shown at $t = \{.25, .5, .75, 1.0\}$. (b) e_1 is black, e_2 is blue dashed, both are plotted as a function of time

We look at the tranport equation

$$\frac{dy}{dt} = -y_x \tag{3.5}$$
$$y(0,t) = 2,\ y(x,0) = 0$$

Using method of lines, we convert the PDE to an ODE on a grid of 300 points (see Section 2.1 for the numerical details). The POD basis is formed from 1200 snapshots collected over the time interval $[0, 1]$ with speed, $a = 1$ and initial condition, $y(0,t) = 2$. It has dimension 25. Neither varying the boundary condition nor reducing the speed change the effectiveness of the ROM. In all the computations, this was reflected by observing a ratio of e_2 to $\bar{e}_1$ which was less than one.

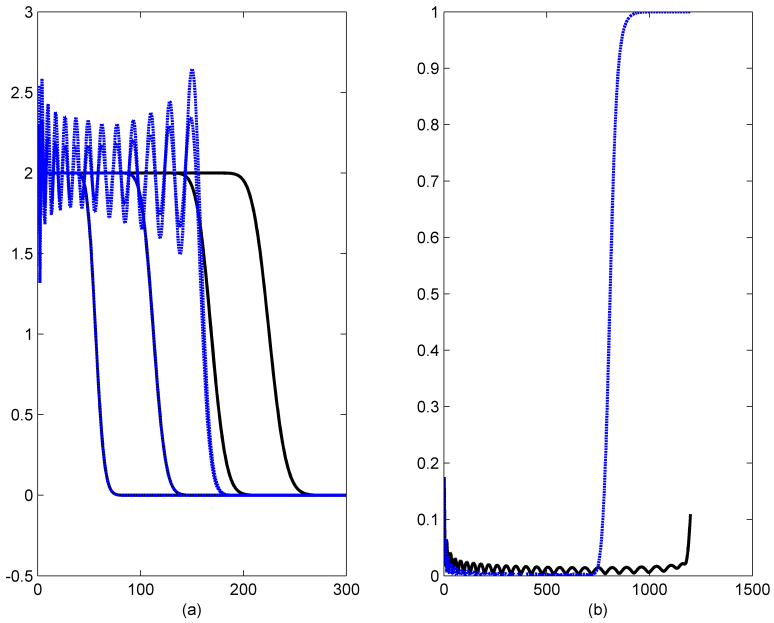

Figure 3.6: The speed is increased to $a = 1.5$ (a) Black is the full dimensional, perturbed system, blue dashed is the ROM for the perturbed system. The solutions are shown at $t = \{.25, .5, .75, 1.0\}$. (b) e_1 is black, e_2 is blue dashed, both are plotted as a function of time.

Increasing the speed, however, preserves the shape of the front, but causes the solution to overrun the original basis while adding in dissipation (there is already some diffusion in the discretization due to the numerical scheme) both changes the shape of the front (the slope becomes less steep), and causes it to overrun the basis.

In both of these cases, the ratios of e_2 to $\bar{e}_1$, correctly detect both when the ROM's capture the perturbed solutions well, and when they stop being effective. Initially, the ROMs work well and the ratios of e_2 to $\bar{e}_1$ are bounded above by one for both perturbed problems. When the ROM's stop functioning well, the ratios increase rapidly to about 100.

The results are shown in Figs. 3.5 and 3.6. Figure 3.5 shows transport with diffusion added where the coefficient is .04 while Fig. 3.6 shows transport with the speed increased to $a = 1.5$. The perturbed systems and their ROM's are shown on the left at equally spaced time intervals, while the values for e_1 and e_2 are on the right. In Fig. 3.6, the first two sets of fronts

plotted for $t = \{.25, .5\}$ are nearly identical and overlay eachother. The osscillations observed in the graph are associated with the two time slices evaluated for $t = \{.75, 1.0\}$.

3.2.3 FitzHugh-Nagumo

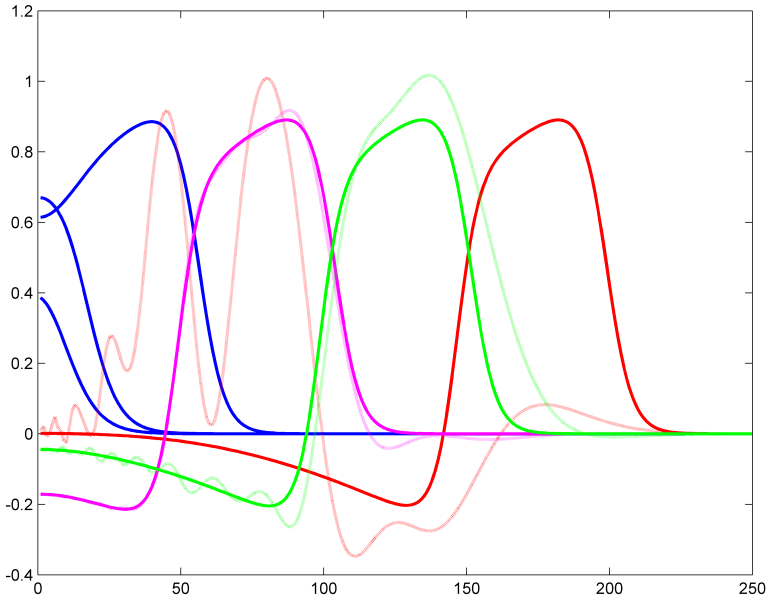

Figure 3.7: The basis for the ROM is constructed for $m = 12,000$. The solid lines are u for the full dimensional system where $m = 12666$ and the dashed are u derived from the projected system. u is shown every 500 time steps from 500 to 3000.

Next, we consider the FitzHugh-Nagumo model.

$$\varepsilon \frac{du}{dt} = \varepsilon^2 u_{xx} + u(u - .1)(1 - u) - y \quad on\ [0, 1.3] \tag{3.6}$$

$$\frac{dy}{dt} = bu - \lambda y$$

$$u(x, 0) = 0 \quad y(x, 0) = 0$$

$$u_x(l, t)) = 0 \quad u_x(0, t) = -i_0(t)$$

We will use the scaling of Keener and Sneyd [KS98] that

$$\varepsilon = 0.015, \quad b = 0.5, \quad and \quad \lambda = 2$$

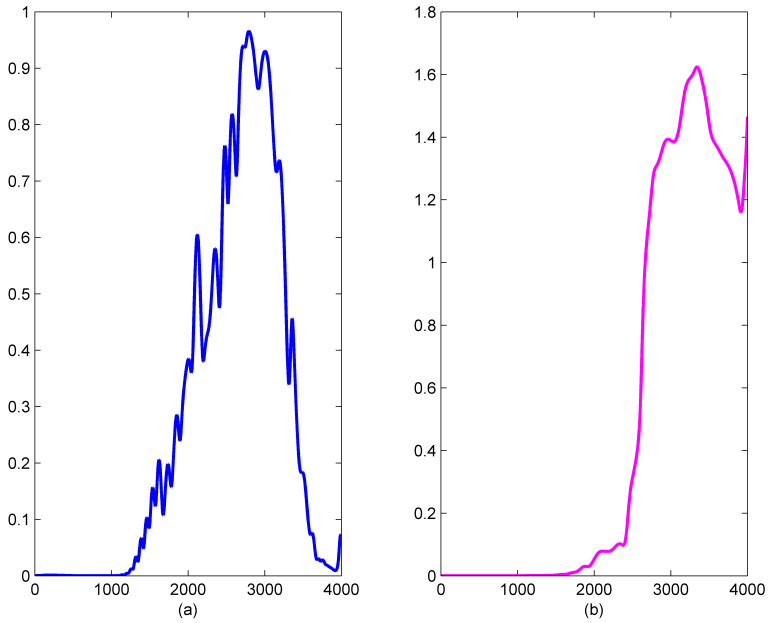

Figure 3.8: The basis for the ROM is constructed for $m = 12000$ and the system is evaluated for $m = 12666$ which produces a traveling wave solution. (a) e_2 is blue; (b) purple is the relative $\|.\|_2$ error between the full dimensional and the ROM system.

The stimulus is $i_0 = mt^3e^{-15t}$ where m is a constant. The size of the stimulus determines whether the system will produce a stable traveling wave or will go to a rest state. As above, using method of lines, we convert the PDE to an ODE and the spatial derivatives are approximated on a grid of 250 points (see Section 5.3.1 for the numerical details). Varying the size of the stimulus causes a bifurcation to occur for m between 12665 and 12666 (see Section 5.3.1 for a more detailed discussion). We form two different POD bases–one from a traveling wave solution (where $m = 13500$) and the other from a decaying wave solution (where $m = 12000$). For each type of basis, we change the size of the stimulus so that the solution from the corresponding ROM will just cross the bifurcation away from the type of solutions well represented by the basis. The ROM is unable to capture the move across the bifurcation threshold in either case. In other words, the basis made when $m = 12000$ works well only for decaying wave solutions, while the basis construced where $m = 13500$ only models successfully for traveling wave solutions.

For our first test, we form the POD basis from the decaying wave solution where $m = 12000$

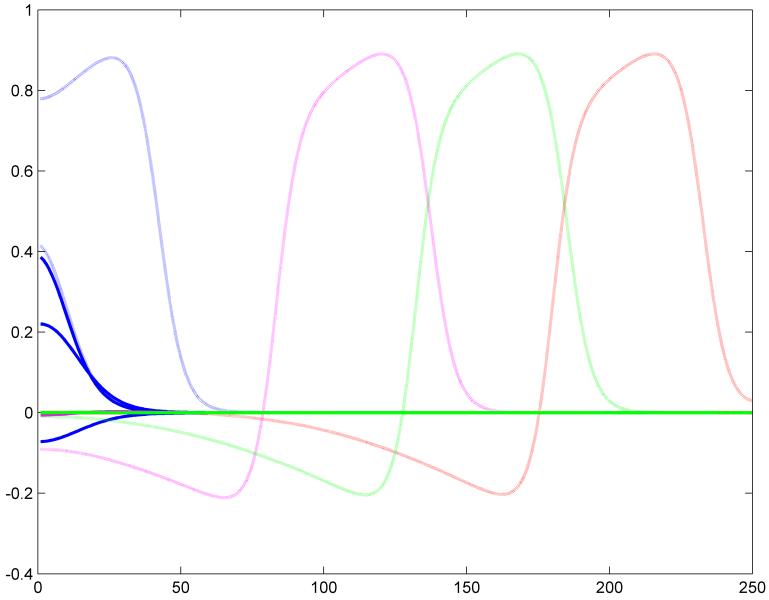

Figure 3.9: The basis for the ROM is constructed for $m = 13,500$. The solid lines are u for the full dimensional system where $m = 12665$ and the dashed are u derived from the projected system. u is shown every 500 time steps from 500 to 3000.

. Specifically, the POD model is constructed using 40 modes selected from 4000 equally-spaced snapshots collected over the time interval $[0, 4]$. The relative errors of u and y between the full and reduced models sampled over the domain (where the solution is greater than .001)

$$\frac{||u - \tilde{u}||}{||u||} < .001$$

$$\frac{||y - \tilde{y}||}{||y||} < .001$$

The POD basis models the decaying solutions well for $m \leqslant 12665$. For $m \geqslant 12665$, it is able to represent the traveling wave solutions only while the decaying solutions used to form the basis are not too small. The fronts are initially well represented, but for longer times, as the wave moves forward, the solutions to the ROM are completely inaccurate and even start moving backwards. This can be seen in Fig. 3.7 for $m = 12666$.

This is correctly indicated by e_2 which is shown in Fig. 3.8 where e_2 increases as the exact error increases. Initially, e_2 and $\bar{e}_1$ have a ratio close to one. This starts increasing after about

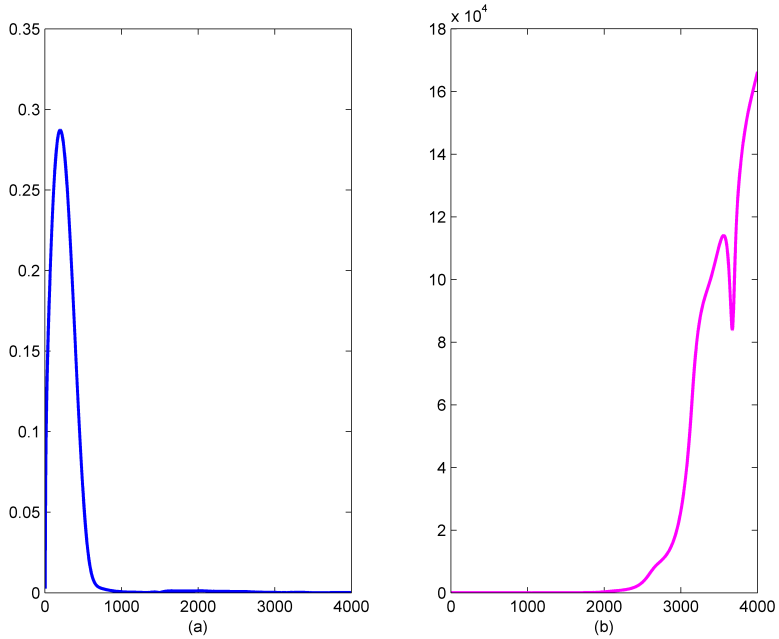

Figure 3.10: The basis for the ROM is constructed for $m = 13,500$ and the system is evaluated for $m = 12665$ which produces a decaying solution. (a) e_2 is blue; (b) purple is the relative $\|.\|_2$ error between the full dimensional and the ROM system.

1500 time steps and the ratio between e_2 and $\bar{e}_1$ becomes very large (about 940) when the true relative error is at its largest. Interestingly the ratio does not stay as large near the end of the time interval, it still is generally greater than 10 indicating that the model is not functioning well. The fact that it become so large around 2500 time steps should be an indicator that the basis in the ROM is performing poorly and the simulation ought to be stopped or the POD basis needs to be enlarged.

In the second test, we create a basis from the traveling wave solution where $m = 13500$. The POD model is constructed from the 4000 equally-spaced snapshots collected over the time interval $[0, 4]$. The projection matrix is constructed from 130 modes. The relative errors between the full and reduced models sampled over the domain are

$$\frac{||u - \tilde{u}||}{||u||} < .011$$

$$\frac{||y - \tilde{y}||}{||y||} < .010$$

This suggests that the ROM of dimension 130 represents the full order model well for $m = 13500$.

The POD basis models traveling wave solutions well, but is unable to capture the behavior of the decaying solutions for $12665 \geqslant m \geqslant 12631$. For m in this range, the ROM generates traveling waves instead of decaying solutions (see Fig. 3.9).

Here, the ratio of e_2 to $\bar{e}_1$ increases rapidly to about 30 and then falls back down to less than one for the remainder of the time as shown in Fig. 3.10. Although the size of the ratio increases before the actual relative error increases, it does correctly indicate that the ROM does not model the system well.

3.2.4 2-D Nondivergent Barotropic Vorticity Equation

For our last system, we considered a 2-D nondivergent barotropic vorticity equation used to model tropical cyclone motion [Cha05]. To apply the error detection method to this model, we selected equations and parameter values taken from Chan and Williams [CW87] while the boundary conditions are taken from [FE89]. The governing equations are

$$\frac{\partial \zeta}{\partial t} + \mathbf{V} \cdot \nabla \zeta + \beta v \; on \; [1010, -1010] \times [1010, -1010] km^2 \tag{3.7}$$

$$\Delta \psi = \zeta$$

$$\mathbf{V} = (u, v) \; , \quad u = \frac{\partial \psi}{\partial y} \; , \quad v = -\frac{\partial \psi}{\partial x}$$

where ζ is the vertical component of the relative vorticity and $\mathbf{V}$ is the nondivergent wind vector. The Rossby parameter, β is defined to be

$$\beta = \frac{df}{dy} = \frac{d(2\omega sin\phi)}{d\phi}\frac{1}{a} = \frac{2\omega cos\phi}{a}$$

where f is the Coriolus parameter, y is the meridional distance, ϕ is the latitude, ω is the angular speed of the earth's rotation, and a is the mean radius of the earth.

Following Chan and Williams ([CW87]), the tangential wind $V(r)$ profile is given by

$$V(r) = V_m(\frac{r}{r_m}) \exp(\frac{1}{b}(1 - (\frac{r}{r_m})^b))$$

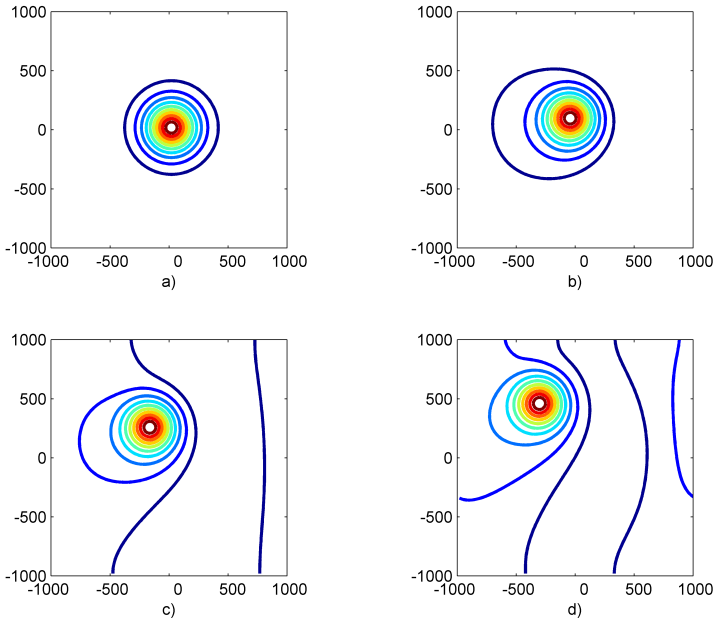

Figure 3.11: a) 0 hr b) 24 hr c)48 hr d) 72 hr.

where r_m is the radius of maximum wind, b is the factor that determines the shape of the vortex, and V_m is the value of $V(r)$ at the radius of the maximum wind intensity. The initial vorticity profile is then [CW87]

$$\zeta(r) = 2\frac{V_m}{r_m}(1 - \frac{1}{2}(\frac{r}{r_m})^b)\exp(\frac{1}{b}(1 - (\frac{r}{r_m})^b))$$

The system is initialized by using this to solve for ψ. The boundary conditions for ψ are periodic from East to West, and Neumann North to South [FE89]. Specifically,

$$\nabla\psi(x, 1010, t) = 0\ , \quad \nabla\psi(x, -1010, t) = 0$$

$$\psi(1010, y, t) = \psi(-1010, y, t)$$

A 100×100 grid is used. Initially, we set $V_m = 40\frac{m}{s}$, $r_m = 100km$, $b = 1.0$ as in [CW87], [FE89]. $\beta = 10^\circ$ latitude. The system is solved using 2nd ordered centered differencing for the velocity and gradient terms and with 4th order Runge Kutta for time. The laplacian of the stream function is solved using a fast poisson solver [SS75]. In Fig 3.11, we show the evolution of the stream function every 24 hours over a 72 hour period.

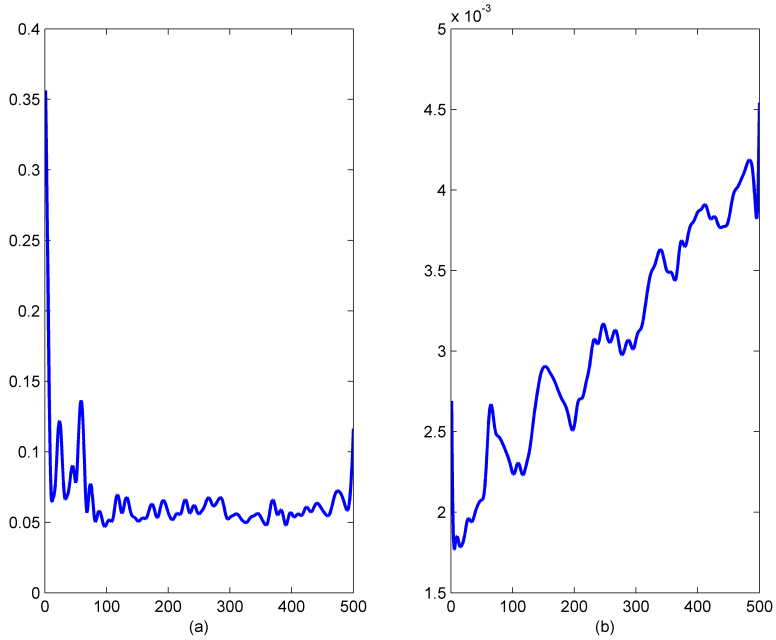

Figure 3.12: The system is evaluated for $V_m = 40$. a) e_1 b) The relative error in ζ as a function of time.

We apply the error detection method over the first 24 hour period. 500 timesteps are used to model the original system, which generates a relative $\|.\|_2$ error of 3.2% in the vorticity at the final time (here, the true solution was estimated by refining the grid and the time steps). The POD model is then constructed from these 500 equally-spaced snapshots collected over the time interval $[0, 24]$ hours.

The relative error in the vorticity between the full and reduced models at the final time, $t = 24.0$, is .5% with 25 basis elements in the reduced order model. We perturb the initial wind profile by varying the value of V_m (the value of $V(r)$ at the radius of maximum wind intensity). This changes the speed, shape and path of the vortex. In Fig. 3.12, the values for, e_1 (the $\|.\|_2$ error due to the projection of the operator for the original system with $V_m = 40$) and the relative $\|.\|_2$ error of the vortex, are recorded at each timestep. e_1 generally maintains an average value of around 7% over the time interval.

In Figs. 3.13 and 3.14, the behavior of e_2 (the $\|.\|_2$ error due to the projection of the operator for the perturbed system) are shown. In Fig. 3.13, where V_m is reduced to 35, the growth of

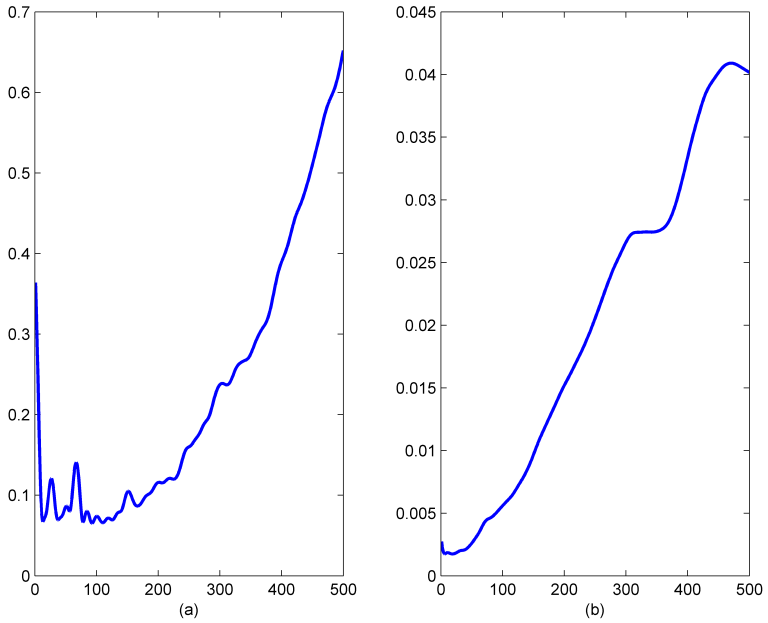

Figure 3.13: The system is perturbed with $V_m = 35$. a) e_2 b) The relative error in ζ as a function of time.

the ratio between e_2 and $\bar{e}_1$ resembles the growth of the relative error of the perturbed system at each time step. It initially starts around 1.4 and rises steadily to about 9 at the final time, which resembles the rise of the relative error from .3% to 4%.

In Fig. 3.14, V_m is increased to 45. Here, again, the error due to the projection of the operator for the perturbed system, e_2, grows proportionally to the growth of the relative error of the vorticity solution, ζ while $\bar{e}_1$ stays small. Initially, the ratio between $\bar{e}_1$ and e_2 is 1.4. At the final time, the ratio is around 13 which reflects the large (12%) relative error in the perturbed solution. $\bar{e}_1$ provides an accurate reference point for both these cases.

The comparison of e_2 to $\bar{e}_1$ correctly detects both when the POD basis is functioning well, and when it stops being effective for the perturbed ROM for all the perturbations considered.

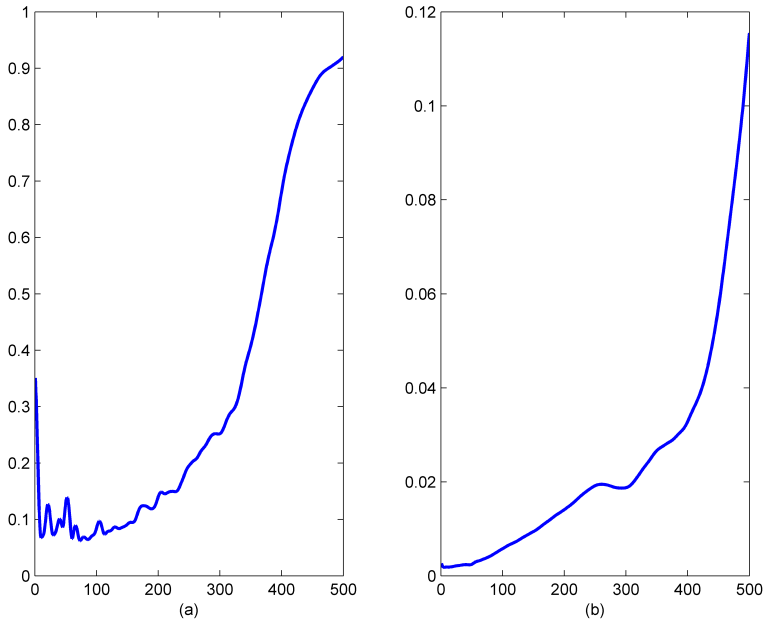

Figure 3.14: The system is perturbed with $V_m = 45$. a) e_2 b) The relative error in ζ as a function of time.

3.3 Conclusion

POD based reduced order models are constructed using snapshots of the time evolution of a system with specific parameters and initial conditions. While relatively easy to assess the capability of the POD based ROM to approximate the specfic system used to construct it, it becomes difficult to assess the capabilities of the POD method for accurately approximating solutions of systems whose parameters or initial conditions are perturbed. In this chapter we have proposed and demonstrated with several examples that an error indicator based on the relative size of the projection error is a reliable predictor of the errors in the solution created with the ROM.

The reliability of the error indicator was demonstrated by applying it to four different PDE's with a variety of perturbations. On 1-D reaction-diffusion, the boundary conditions and diffusion coefficients were varied. For transport, we increased both the speed and diffusion. The initial wind profile was perturbered for the 2-D nondivergent barotropic vorticity equation. In

all three cases, the error indicator worked extremely well at detecting the behavior of the error in the perturbed ROM. It was able to detect both when the POD system was function well, and when the accuracy of the model deteriorated.

For the Fitz-Hugh Nagumo system, the stimulus was adjusted so that the solution from the ROM would just cross the bifurcation away from the type of solutions well represented by the POD basis. The ROM was unable to capture the move across the bifurcation threshold with either basis. In both of these situations, the error indicator was able to detect that the POD basis was not modeling the solution correctly.

The error indicator introduced offers a useful tool for detecting error as it arises during the evolution of a perturbed POD model. Combined with the ability to augment the basis with singular vectors from perturbed systems as needed, which is discussed in Chapter 5, it may greatly improve both the adaptability and efficiency of POD.

CHAPTER 4

Sensitivity Analysis and Proper Orthogonal Decomposition when Applied to Reaction-Diffusion Type Equations

An additional observation that can be made from the investigation of POD based reduced order models in Chapters 2 and 3, is that one of the more difficult systems to model is the reaction-diffusion equation. In this and the next chapter, we propose and investigate methods for improving the ability of a POD based ROM to capture the behavior of these systems. The key idea is that one needs to not only be able to accurately reproduce the solution whose snapshots give rise to the POD basis, but also to accurately reproduce the solutions and sensitivities associated with perturbed initial conditions or model parameters.

This chapter presents some limitations of the linearized error equations associated with sensitivity analysis for reaction-diffusion type problems. It then explores the alternative of doing sensitivity analysis on a POD reduced order model. Sensitivity analysis is reviewed in Section 4.1. This is followed by a discussion of the problems associated with sensitivity analysis for reaction-diffusion type equations in 4.2. In Section 4.2.1, these limitations are exhibited by applying sensitivity analysis to two simple numerical examples, 1-D Reaction-Diffusion and the Fitz-Hugh Nagumo equation.

An alternative to the usual sensitivity analysis methods is discussed in 4.3. We demonstrate that one can obtain good sensitivity estimates directly by solving forward in time on POD reduced order models with dimensions significantly smaller than those in the original systems. Results from sensitivity analysis done on three POD reduced order models are included in 4.3.1.

4.1 Sensitivity Analysis

Quantifying uncertainty is of great interest in scientific computing. The ability of an ODE or PDE to deal with variations is commonly investigated using sensitivity analysis. This was pioneered by Lorenz in the 1960's and has been used extensively in many fields. The technique helps determine the directions of error growth and decay. We begin with a review of the fundamental sensitivity analysis construction due to Lorenz ([Lor65]).

Consider the dynamical system

$$\frac{dy}{dt} = f(y, p, t), \qquad y(t_0) = y_0(p) \tag{4.1}$$

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$, $t \in [t_0, t_f]$ and $y, y_0 \in \mathbb{R}^n$. y_0 is the initial condition at time t_0 and p denotes the model parameters. This system corresponds to either an ordinary differential equation (ODE), or a semi-discretized partial differential equation (PDE).

We first consider the estimate of the behavior of two solutions with differing initial conditions. Later, we explain why estimating this behavior is also sufficient for the case of parameter perturbations.

Let $y(t)$ and $\tilde{y}(t)$ be two solutions of 4.1 such that $y(t) = \tilde{y}(t) + \varepsilon z(t)$ where ε is small and $z(t_0)$ has unit measure, then

$$\begin{aligned} \frac{d(\varepsilon z)}{dt} &= \frac{dy}{dt} - \frac{d\tilde{y}}{dt} \\ &= f(y) - f(\tilde{y}) \\ &\cong Jf(y - \tilde{y}) \\ &\cong Jf_{|y}(\varepsilon z) \end{aligned}$$

The difference between the two solutions can be approximated by the homogeneous linear equation (also called the tangent-linear system).

$$\frac{d(\varepsilon z)}{dt} = Jf_{|y}(\varepsilon z), \qquad \varepsilon z(t_0) = \varepsilon z_0 \tag{4.2}$$

The Jacobian, Jf, is evaluated along the reference trajectory, $y(t)$. Let us denote the solution operator of 4.2, as $L(t_0, t)$ where $z(t) = L(t_0, t)\varepsilon z(t_0)$. This is referred to as the tangent-linear propagator or simply the propagator in atmospheric science literature ([LP08]).

The norm, $||\varepsilon z(t)||$, is a measure of the deviation caused by perturbing the initial condition with εz_0.

Using the solution operator, $L(t_0, t)$, it is possible to determine the directions that are the most sensitive to perturbations. The magnitude of the error between the solutions at time t_0 is

$$||\varepsilon z_0|| = \varepsilon$$

and at some later time, $t_1 = t_0 + \delta t$, the difference in solution values satisfies the equation

$$\begin{aligned} \langle \varepsilon z(t), \varepsilon z(t) \rangle &= \langle L(t_0, t)\varepsilon z(t_0), L(t_0, t)\varepsilon z(t_0) \rangle \\ &= \langle \varepsilon z(t_0), L(t_0, t)^* L(t_0, t)\varepsilon z(t_0) \rangle \end{aligned}$$

The behavior of $||z(t)||$ can then be determined by looking at the singular value decomposition of L. The singular values indicate the extent to which the difference between the trajectories, in the direction of the corresponding right singular vector, grows or decays according to whether the singular values, σ_i, are above or below unity. The left singular vector is the corresponding deviation in the solutions. For a more detailed description, see [Lor85], [LP08], [Cac1a], [Cac1b] , [Err97]. Note that

$$\lambda_i = \lim_{t \to \infty} \frac{1}{t - t_0} \log ||\sigma_i||_{2,\infty}$$

are the Lyapunov exponents [Lor85].

To estimate the sensitivity of systems due to variations in problem parameters, there are two approaches one can take. The first consists of constructing and analyzing a linear equation for the difference of the perturbed an unperturbed solution

$$\frac{dz}{dt} = Jf_y(z) + Jf_p(\delta p), \qquad z(t_0) = y_0(p)(\delta p) \tag{4.3}$$

where Jf_y and Jf_p are the jacobians of f with respect to y and p respectively and are evaluated at $(y(t), p, t)$. $z(t_0) = y_0(p)(\delta p)$ is the first order approximation of $y(p + \delta p, t_0) - y(p, t_0)$.

The other approach consists of transforming the problem to one where the initial conditions are perturbed by enlarging the dimension of 1.1, so the system would then become

$$\frac{dy}{dt} = f(y, p, t) \qquad y(t_0) = y_0$$

$$\frac{dp}{dt} = 0 \qquad p(t_0) = p$$

This is the technique used in the numerical examples considered below.

Instead of calculating the tangent linear operator, $L(t_0, t)$, one can numerically compute the derivatives of the solution at time $t_0 + \delta t$ with respect to the perturbations. For example, in the case where one perturbs initial conditions, we do this by first evolving $y(t)$ for a short period of time from t_0 to $t_0 + \delta t$. We then perturb y_0 with n vectors $\varepsilon z_1(t_0), ..., \varepsilon z_n(t_0)$ where $z_{ij}(t_0) = \delta_{ij}$ and ε is small. By integrating 4.1 numerically n times from t_0 to $t_0 + \delta t$, we can approximate the differences, $z_i(t_0 + \delta t)$. The deviations of the solution data with respect to the perturbation of the ith initial coordinate are then given by

$$z_i(t_0 + \delta t) = \frac{S_{\delta t}(\varepsilon z_i(t_0) + y(t_0)) - S_{\delta t}(y(t_0))}{\varepsilon} + O(k^r) \tag{4.4}$$

Here, k^r is the error associated with the numerical computation of $y(t)$ using stepsize k.

Finding techniques to calculate the singular values of the solution operator is an active area of research [LH09], [HL12]. One common method is the use of Lanczos-type algorithms which rely on only the matrix vector products [GL96]. Each iteration requires the forward integration of a linearized model followed by a backward in time solution of the corresponding adjoint model. Then, a generalized Davidson method can be used to iteratively find the leading eigenvectors ([LP08]).

Many systems where sensitivity analysis would be an important tool are simulated using high-dimensional equations. For example, in numerical weather prediction done on large scales of hundreds of kilometers, the dimension of the state vector is of order 10^5-10^8 (see the review paper [LP08]). Regardless of whether the sensitivites are determined by calculating the solution operator, $L(t_0, t)$, or using finite difference calculations, it can be computationally unfeasible [LH09], [HL12]. For this reason, it is important to explore alternative methods for calculating sensitivities.

4.2 Limitations of the Linear Error Equation

As discussed in the previous section, one can carry out sensitivity analysis using local linearization. Depending on how the solution operator is calculated, however, this can be problematic. Specifically, we show that in commonly used reaction-diffusion equations, a sensitivity analysis based upon the linearized error equations leads to approximations that are severely limited.

A generic reaction-diffusion equation has the form

$$\frac{dy}{dt} = R(y) + D\nabla^2 y$$

When the reaction term is linearized, its Jacobian is evaluated at a specific time, $\tilde{t}$. The reaction then becomes "fixed" at whatever time the Jacobian is evaluated at. In particular, in the absence of "spontaneous reaction", the reaction term where no reaction is occuring will be identically zero. For example, the Jacobian of 4.5 after it has been partially discretized using method of lines, is

$$\begin{pmatrix} 4y_1 - 3{y_1}^2 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 4y_n - 3{y_n}^2 \end{pmatrix} + \frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix}$$

The Jacobian of the reaction term evaluated at a fixed time, $\tilde{t}$, will be nonzero where $y_i(\tilde{t})$ is nonzero, and zero where $y_i(\tilde{t})$ is zero. This means that the Jacobian of the reaction term will be nonzero only where the front has reached by $\tilde{t}$. As the solution moves, only diffusion will act on the new nonzero locations. This suggests that the solution of the linear error equation will not be able to model the error that develops between perturbed trajectories and the original solution correctly unless δt is extremely small.

4.2.1 Numerical Examples

The consequences of this property of the Jacobian associated with linearizing the reation term can be clearly observed in the following two cases of reaction-diffusion equations. In these cases, we compute the relative difference between the "true" variations in the solution and the variation in the solution obtained using a local linearization:

$$E = \frac{\|e^{Jf(\delta t)}\varepsilon z_i(t_0) - (S_{\delta t}(\varepsilon z_i(t_0) + y(t_0)) - S_{\delta t}(y(t_0)))\|_2}{\|S_{\delta t}(\varepsilon z_i(t_0) + y(t_0)) - S_{\delta t}(y(t_0))\|_2}$$

1) The first example is the reaction-diffusion equation

$$\frac{dy}{dt} = ay^2(2-y) + by_{xx} \tag{4.5}$$

$$y(0,t) = 2,\ y(2,t) = 0,\ y(x,0) = 0$$

with reaction coefficient, $a = 20.0$ and diffusion coefficient, $b = .025$. The partially discretized ODE has dimension $n = 300$. We evolve the system until $\tilde{t} = .2$ using 1200 time steps, and then form the linear error equation around $y(.2)$. The number of timesteps was chosen so that the numerical solution was accurately computed..

We introduce perturbations onto the front at $\tilde{t}$ (the portion of the solution between 2.0 and 0.0 with the restriction that the perturbations can not exceed two or be less than zero). Specifically, we generate 28 orthonormal perturbation vectors with random entries (from the interval $[0,1]$) that are nonzero only on the front and then scale them to have a magnitude of $\frac{1}{100}$.

If the system is evolved for all the perturbations for $\delta t = .05$ (discretized with 300 time steps), we find that the relative variation between the linearized variation and the true variation (calculated numerically) for the perturbations varies between $E = 25.8\%$ and $E = 242\%$, with a mean relative difference of $E = 75.8\%$.

2) The second example is the Fitz-Hugh Nagumo reaction-diffusion equation.

$$\varepsilon\frac{du}{dt} = \varepsilon^2 u_{xx} + u(u-p_1)(p_2-u) - y \ \ on\ [0,1.3] \tag{4.6}$$

$$\frac{dy}{dt} = bu - \lambda y$$

$$u(x,0) = 0 \quad y(x,0) = 0$$

$$u_x(l,t)) = 0 \quad u_x(0,t) = -i_0(t)$$

with the scaling of Keener and Sneyd [KS98] that

$$\varepsilon = 0.015, \quad b = 0.5, \quad and \quad \lambda = 2$$

The stimulus is $i_0 = mt^3e^{-15t}$ where $m = 15000$. By using method of lines, the PDE is converted to an ODE using standard centered difference in space to approximate the spatial derivatives on a grid of size 250. Time is discretized with 4000 timesteps for $t \in [0, 4]$. We evolve the system until $\tilde{t} = 1.0$ with 1000 steps using 3rd order, explicit Runge-Kutta so that the traveling wave solution has stabilized to its final form. We then generate the linear error equation around $\tilde{t} = 1.0$. For this test case, we perturb the problem parameters $p_1 = .1$ and $p_2 = 1.0$ by .01 and evolve the solution for $\delta t = .15$.

When we calculate the variations between the solutions determined numerically versus using the linear error equation, we get very different answers. Varying p_1 by .01 gives a relative difference of $E = 296\%$. Varying p_2 by .01 generates a relative difference of $E = 127\%$. The variations at time $t = 1.15$ calculated using the two methods caused by the perturbation of p_2 are shown in Fig. 4.1.

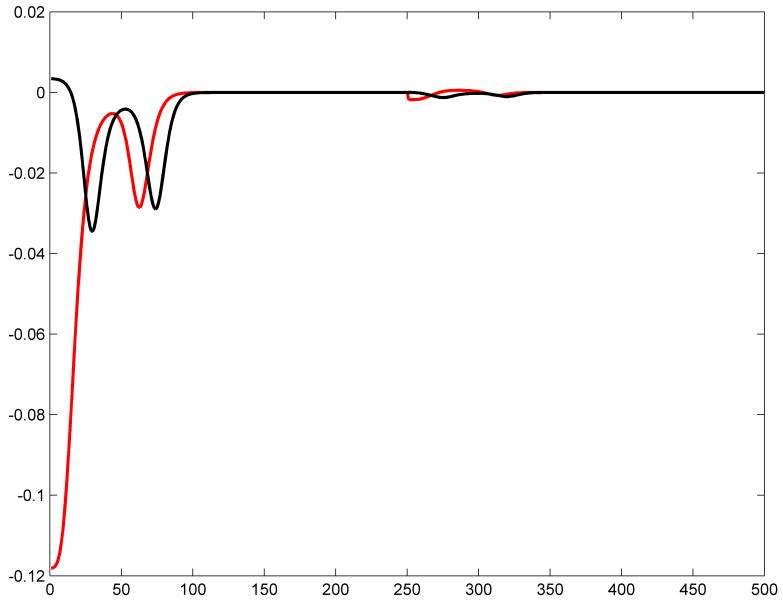

Figure 4.1: Variations between the solutions at $t = 1.15$ caused by perturbing p_2 at time $t = 1.0$. Black is the "true" variation calculated numerically, red is the variation based upon a linearization of the solution operator at time $t = 1.0$.

4.3 Sensitivity Analysis and POD reduced order models

The examples above indicate that the linear error equation is not a reliable method for determining the sensitivities of reaction-diffusion equations. Perturbations to the front of the 1-D Reaction-Diffusion equation produced differences in the variation of $E = 75.8\%$. The linearization was much less accurate for the Fitz-Hugh Nagumo equation where the relative differences in the variations were $E = 127\%$ and $E = 296\%$.

For these particular examples of reaction-diffusion equations, and presumably many others, local linearization is problematic. They suggest that it is either necessary to calculate a highly accurate value of the solution operator, $L(t_0, t)$ (4.1), or it would be better to determine the sensitivities by evolving the solution operator.

If there are not very many parameters to perturb and the systems are not too large, it is not a problem to calculate the effects of perturbations with these method for short periods of

time. For problems involving many parameter alterations or when the system is large, this is not possible. An alternative is to do sensitivity analysis directly on a reduced order model such as POD.

To demonstrate this approach and illustrate the computational improvements one can obtain, we again consider the examples above. Instead of calculating the linear error equation, however, we calculate the difference between the errors caused by perturbing problem parameters calculated numerically in the full system versus the POD system. The sensitivities calculated using the POD basis are significantly more accurate than those calculated using the linear error equation in all the cases we look at below.

4.3.1 Numerical Examples

In the first test case, we consider the 1D reaction-diffusion equation (with the same parameters and discretization as above 5.4). The POD model is constructed from 2400 equally-spaced snapshots collected over the time interval $[0, 1.1]$. The basis contains 28 vectors. The relative error between the full and reduced models on the interval $[0, 1]$ is

$$\frac{||y - \tilde{y}||}{||y||} < .0013$$

which indicates that the ROM of dimension 28 represents the full order model very well.

The relative $\|.\|_2$ differences between the solutions caused by perturbing the front vary from 1.30% to 6.74%. The average of all 28 possible perturbations is 3.82%. This is significantly better than the difference in the solution caused by using the linear error equation which had a mean relative error of $E = 75.8\%$.

In the next test case, we considered the same reaction-diffusion system as above except the diffusion parameter, b, is perturbed instead of the front. Specifically, the diffusion parameter is perturbed by $-.02$, and the system is evolved for time equal to .05 and .1 (discretized by 80 and 160 steps respectively) starting at $t = .2$.

When we carry out sensitivity analysis on both the full dimensional and the POD reduced order model, we find that there is a 98% agreement between the variation calculated numerically

in the full system and the reduced (see Table 4.1).

For both of the cases involving reaction-diffusion, the dimension of the POD reduced order model was only 28 as opposed to 300 in the full system. Despite this fact, we still acheived good accuracy in the sensitivities.

Table 4.1: The relative differences between the variations caused by perturbing diffusion by $-.02$ for time evolved $t = .05$ and $t = .1$

Relative Differences in the Variations	
Time Evolved	Relative Difference
.05	0.0153
0.1	0.0104

Lastly, for the Fitz-Hugh Nagumo equation 4.6 (with the same parameters and discretization as above), the system is constructed on a grid of size 250. The POD model is constructed with 4000 snapshots taken over the interval $[0, 4]$ with dimension 130. The relative errors between the full and POD reduced models sampled over the domain are $< .005$ for both u and y so this is a good POD model. From Table 4.2, it is clear that the POD model is quite successful at reproducing the variations caused by perturbations. The largest relative difference is 6%. This contrasts strongly with the linear error equation which generated relative differences of 296% and 127% for perturbations in p_1 and p_2 respectively. The dimension of the POD model was almost 50% less than the original system.

Table 4.2: Listed below are the exact and relative $\|.\|_2$ differences between the variations caused by perturbing p_1 and p_2 by .01 between the full and reduced order models.

Exact and Relative Differences of the Variations		
Perturbed Parameter	p_1	p_2
Exact	0.0095	0.062
Relative	0.061	0.035

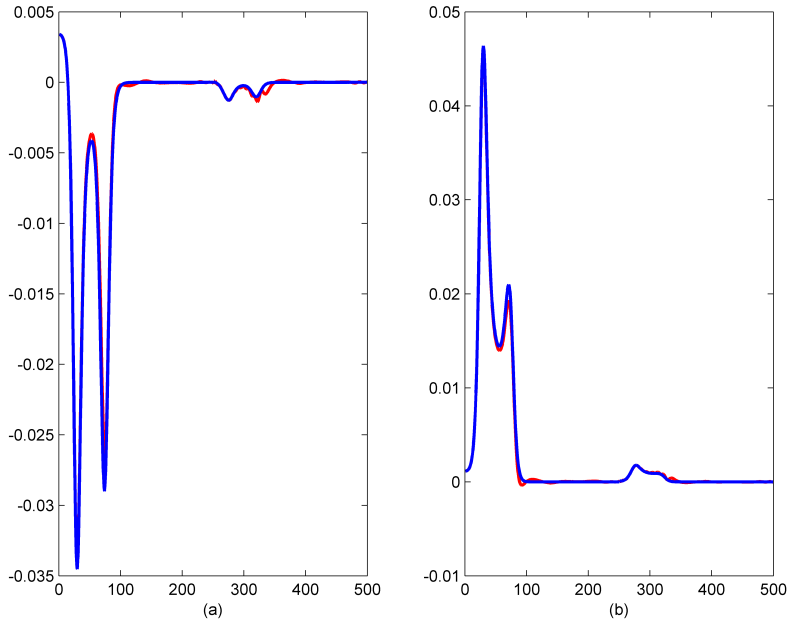

Figure 4.2: Variations in the solution at time $t = 2.15$ caused by perturbing p_1 (a)) and p_2 (in (b)) at time $t = 2.0$. Blue is the "true" variation calculated numerically, red is the variation calculated by evolving the solution of the POD system.

4.4 Conclusion

On two common types of reaction-diffusion type equations, we showed that there is a large difference between the variations, $z_i(t)$, caused by perturbations (both of initial conditions and model parameters) calculated by using the linear error equation, and the true variation calculated numerically. The relative differences varied between $E = 75.8\%$ and $E = 296\%$. The results indicate that sensitivity analysis carried out using the linear error equation for reaction-diffusion type equations cannot be relied upon to produce the correct singular vectors and values indicating the sensitivities. They also suggest that generally, a lot of care must be taken with the calculation of sensitivities for these types of equations.

In cases when there are relatively few parameters to perturb and the system is not too big, it is likely better to do sensitivity analysis by evolving the solution numerically or by calculating a highly accurate value of the solution operator, $L(t_0, t)$ (4.1). When this is prohibitive

computationally, sensitivity analysis could be done instead on a reduced order model such as POD.

In the examples we looked at, we found that the POD models produce dramatically better estimations of the errors caused by perturbations than the linear error equation. These systems were significantly reduced in dimension with a 90% and 50% reduction in size for reaction-diffusion and Fitz-Hugh Nagumo respectively while retaining good accuracy in the sensitivity calculations. Furthermore, as discussed in 6 using a POD reduced order model, there is the additional advantage that significantly larger time steps can be taken due to the relaxation of the stability requirement in the reduced order model.

Doing sensitivity analysis on POD reduced order models is limited by the ability of the model to handle perturbations. There are certainly some cases where the POD reduced order model could not model the perturbed directions even for a very short period of time. In models such as 2D Brusselator PDE (see [HPS05] for a more detailed discussion) in which the direction of maximum error growth for the full system is almost orthogonal to the POD subspace, sensitivity analysis will not give accurate results on the POD model.

By combining basis augmentation with error detection (in Chapter 5), this could hopefully be corrected for. At the very least, if the POD basis can not handle the perturbed direction, the error detection method discussed in Chapter 3 should provide feedback if the POD basis is not able to manage the perturbation. The examples where we apply the error detection method demonstrate that it is particularly sensitive to the solutions trying to leave the POD subspace. So, for examples such as the 2D Brusselator, the user would be aware when to distrust POD results.

CHAPTER 5

POD Basis Augmentation

This chapter investigates the effects of using a variety of strategies for basis augmentation, in particular, how to incorporate components of perturbed systems to improve the ROM basis. We present three techniques for augmenting a POD basis such that its corresponding ROM is significantly more adaptable to perturbations. The first two basis augmentation methods are introduced in Section 5.2. The third method is a modified form of POD for perturbed systems which offers reductions in the computational cost over the standard POD method and is also discussed in Section 5.2. Section 5.3 contains numerical experiments on two types of reaction-diffusion systems: the 1-D reaction-diffusion equation and the 1-D Fitz-Hugh Nagumo model.

In 5.3.1- 5.3.3, we demonstrate both how to use these methods in practice, and compare them to the alternative of applying the non-augmented POD basis to the perturbed systems. The first two subsections involve varying one parameter. Section 5.3.3 deals with the more complex situation of varying three parameters in the Fitz-Hugh Nagumo equation. In all three experiments, we find that the augmented bases work extremely well. In 5.3.5, we compare two of our methods for augmenting the POD basis to a method presented in [SDW12]. We demonstrate that on the problems considered, the methods outlined in 5.2 are more successful.

5.1 Background

Reaction-diffusion equations are an important component of mathematical models for a wide variety of phenomenon, and there is great interest in the construction of POD based ROM models for such equations. A particular challenge in developing ROM models is creating models

that are capable of accurately representing solutions to reaction-diffusion type equations as model parameters or initial conditions are varied.

Although there is a lot of interest in augmenting the POD basis in order to improve its ability to predict behavior away from the data used to create it (for example, see [SPH07]), there has, so far, only been a few results on specific problems. The majority have been focused on fluid flows with moving discontinuities.

An example of earlier work is that due to Lucia et al. who augmented a POD model with a local full resolution solution to get significant computational improvements on a problem of moving shocks in a nozzle [LKO01]. Specifically, they decomposed the domain such that the problem was solved with full dimension around the shock, and with a reduced, POD basis elsewhere in the domain. LeGresley and Alonso used this technique of enhancing the POD basis to the full dimension only in specific portions of the domain for 2-D tranasonic flow (which can generate significant shock movement), a drag minimization problem, and an aero-structural problem [LJ03], [LJ04]. More recently, in [BFC10], a more efficient POD method involving modes being appended to the POD basis at each step was developed to deal with a moving discontinuity in a simulation of a bubbling, fluidized bed. In these cases, the general idea is to use a POD model in conjuction with another model inorder to improve accuracy, rather than to formulate a better global POD model.

Recent work that describes procedures for improving the global accuracy of a POD model was carried out in the context of laser physics and optics on two different mode locking systems. Here, modes derived using POD for different values of a parameter are combined in an attempt to obtain as much information as possible in the augmented POD basis so that it can be used to study the bifurcations and dynamics of the particular system. Using a POD augmented basis, Ding et al. was able to capture the entire bifurcation sequence from the single mode-locked pulse to the double-pulse solution observed in a ring cavity laser mode locked by a saturable absorber as a single variable, pumping strength, is varied. They accomplished this by combining m modes from a single pulse solution with n modes from a double pulse solution, and then re-orthogonalizing to ensure orthogonality [DSK10]. This is in contrast to [SDW12], where a set of solutions for a waveguide array mode-locking model were calculated as a single parameter

was varied. They were then grouped together into a single data set, and a subset of their SVD was used to create a POD based model which was capable of capturing the dynamics of the system for the perturbed values of the parameter.

5.2 POD Basis Augmentation

Here, we describe three methods which combine POD bases in order to make an augmented basis that is more robust to perturbations for reaction-diffusion type equations. We also provide guidelines for constructing augmented bases that are not restricted to a specific problem (in contrast to much of the earlier work). The use of the methods are demonstrated on model problems of reaction-diffusion type equations. On these representative problems, we demonstrate the importance of augmenting the basis using the methods outlined, and not one of the alternatives used in the literature [SDW12].

All three methods are based on the simple idea of including components of the perturbed solution in the model. The distinction between the methods is the manner in which the additional solution components are constructed from the perturbed solutions. Consider the generic equation

$$\frac{dy}{dt} = f(y, p, t), \qquad y(t_0) = y_0(p) \tag{5.1}$$

If we have a POD basis for the system with parameter, $p = \tilde{p}$, (such that the error between the full and projected systems is less than some tolerance), we should be able to improve the ROM's ability to deal with a range of parameters, $[p_{min}, p_{max}]$, which contains $\tilde{p}$ by incorporating perturbed solution components into the original POD basis.

We assume that $\tilde{p}$ has been chosen such that the corresponding solution is representative of the range of solutions (for example, it should be a traveling wave solution rather than a decaying solution in the case of the Fitz-Hugh Nagumo equation). If the basis requires s elements to model the unperturbed system accurately, and a maximum size, $r > s$, for the basis size of the augmented system has been pre-determined (based on computing resources), then the remaining $k = r - s$ spots in the basis can be filled with other basis vectors. The question then becomes how to determine these vectors.

These additional vectors could be obtained by selecting a collection of parameters from the interval $[p_{min}, p_{max}]$ (for example, using random sampling), running the full dimensional models, and then doing SVD on the snapshots for each individual parameter. The k singular vectors corresponding to the largest singular values would then be combined with the original POD basis. We would then orthonormalize again, and the resulting singular vectors would be used to form the new POD basis. Depending on the number of parameters selected, however, this could be computationally prohibitive.

Rather than using a random sampling procedure, we have found that by using knowledge about the dynamics of the equations to select only a few parameters, p_i, with which to run the full system and thereby obtain candidates for the basis augmentation vectors we can create a augmented basis that works extremely well. In fact, we found that we were able to augment the basis with very few vectors and still obtain a relative error $< 1\%$ for a wide range of perturbed system solutions. In the examples we look at, we select the p_i's based on the extent to which the dynamics differ. Below, we outline four methods for basis augmentation.

5.2.1 Method 0

The first method is the procedure described in [SDW12]. This method consists of combining together snapshots from an unperturbed solution with those from the selected perturbed solutions and selecting a basis from the dominant singular vectors of the combined set.

The motivation for this procedure is clear, one wishes to use a basis set that captures the dominant features of the solution of both the perturbed and unperturbed problems. There is the risk, however, that the dominant singular vectors of the combined set will correspond only to a specific parameter value, or a small range of parameter values (due to the weighting of the SVD vectors). Intuitively one might expect this, as the components of the perturbed system may get "washed" out by the components of the base solution. In particular, there is no guarantee that the largest singular vectors from the combined set will approximate solutions of both the perturbed and unperturbed systems equally well. In section 5.3.5, the limitations of using this method are demonstrated on a number of PDE's.

5.2.2 Method 1

POD's success is predicated on the assumption that the dominant singular vectors correspond to important dynamics. This is particularly questionable, however, when modes for different parameter values are grouped together. In doing this, we risk ignoring the behavior of solutions in certain parameter ranges. Therefore, in Method 1 the SVD of each system is computed, and then a portion of the resulting singular vectors are combined to form a basis. This is analogous to the method used by Ding et al. ([DSK10]) on a mode locking system. By selecting a sufficient number from each set, one is guaranteed to obtain a basis that will work well for the specific perturbed and unperturbed solutions.

Specifically, we first select a number of parameters, p_i's, with which to run the full dimensional system and generate solutions, y_{p_i}. The p_i's are selected such that their solutions model the extreme behavior of the system. The snapshots are then collected and SVD is performed on each collection of snapshots corresponding to the p_i's. The k vectors corresponding to the largest singular values from each set are then combined with the s basis elements from the original POD basis to form a set of size, r. The new collection of vectors, however, are not necessarily orthogonal so they need to be re-orthonormalized.

A question naturally arises as to how many singular vectors from each set should be chosen to ensure that the basis is robust over a range of parameters. This is somewhat problematic because the dominant singular vectors for the separate systems may have significant intersection (large inner products). One can err on the side of caution and always choose the number of dominant singular vectors in each set that are necessary to reproduce the solution that was used to create the set, but this may lead to unnecessarily large bases. In the case when the perturbed solution and the unperturbed solution are significantly different this is not a concern and we find that the method works well on the numerical example considered in Section 5.3.5. This method has the advantage that the computations and construction of the singular vectors for each system are completely independent, thus enabling one to take advantage of multi-processor computational resources.

5.2.3 Method 2

In contrast to calculating the SVD of each set of snapshots from the perturbed solutions, and then selecting a few POD basis vectors to be adjoined to the original basis, this method consists of carrying out POD directly on the systems associated with the perturbations. Specifically, as with Method 1, the parameters, p_i, are selected to generate solutions exhibiting the systems most extreme behavior. Instead of carrying out SVD analysis on $y_{p_i}(t)$ directly, however, we create snapshots of

$$w_{p_i}(t) = y_{p_i}(t) - Py_{p_i}(t)$$

e.g. the components of the snapshot that are not represented by the current POD basis. Alternatively, in order to speed up computations, if $y_{p_i}(t)$ is close enough to $y_p(t)$, $w_{p_i}(t)$ can be estimated by

$$w_{p_i}(t) \cong y_{p_i}(t) - y(t)$$

where $y(t)$ is the base (or unperturbed) solution.

Once $w_{p_i}(t)$ is determined, an SVD anaylsis can be done on its time slices. The vectors corresponding to the largest singular values are then combined with the basis elements from the original POD basis (derived from the $y(t)$). To form the new basis, the vectors just need to be re-orthonormalized.

This method has the advantage that by applying SVD to $w_{p_i}(t)$ as opposed to $y_{p_i}(t)$, we avoid some of the redundancy caused by performing SVD on the time slices corresponding to different parameter values separately before combining the selected singular vectors. The singular values inform us exactly what vectors need to be attached to the original basis in order to get good accuracy in the perturbed system. If $w_{p_i}(t)$ is modeled well, than $y_{p_i}(t)$ will necessarily be modeled well.

We find that in the numerical example considered below, far fewer POD basis elements are necessary to obtain a given accuracy using this method (see Section 5.3.2). This is in contrast to using the original basis derived from $y(t)$, or Method 1 outlined above (see 5.2.2). Doing SVD on $w(t)$, in contrast to the solutions of the system with varying parameter values (such as in [SDW12], [DSK10]), can greatly increase the adaptability of the augmented POD basis.

5.2.4 Method 3

This method is similar to Method 2, in which the POD basis utilizes vectors constructed from the POD basis for the differences between the base (unperturbed) and perturbed solutions. However, we obtain the POD basis by using snapshots of the solution of the equation for the difference between the perturbed solution and a base solution interpolant. This means that a full set of basis solution snapshots or a base solution ROM basis is not required – the required base solution components necessary to form the differences is obtained by interpolation. Additionally, one can form a reduced order model that combines the interpolant with a POD model for the perturbation components alone.

The first step in the method consists of creating a base solution, $y_b(t)$, to be an approximation of the solution $y(t)$ of the 5.1 for $p = \tilde{p}$. We then save

$$y(t_k) \quad and \quad \frac{dy}{dt}|t_k = f(y(t_k)),\ where\ y(t_0) = y_0$$

at a collection of points

$$t_k\ for\ k = 1, ...m$$

in order to construct a high order Hermite interpolant, $y_b(t)$, such that it is a good approximation to the solution, $y(t)$ and $\frac{dy}{dt}$, of the full system where $p = \tilde{p}$. Next, we solve the perturbation equation for the variation of the solution about the interpolated solution (as opposed to the variation about the POD ROM)

$$\frac{dw_{p_i}}{dt} = f(w_{p_i} + y_b) - \frac{dy_b}{dt}$$

and carry out SVD on its time slices. Once the POD basis for w_{p_i} is made, then we use it to construct the reduced order model

$$\frac{dv_{p_i}}{dt} = P^T f(Pv_{p_i} + y_b) - P^T \frac{dy_b}{dt} \tag{5.2}$$

where P is the truncated POD basis for w_{p_i}, $w_{p_i} \cong Pv_{p_i}$, and $y_b(t)$ and $\frac{dy_b}{dt}$ are the interpolants of $y(t)$ and $\frac{dy}{dt}$ respectively. Thus, if we have an approximate model for w_{p_i}, then we have an approximate model for $y_{p_i}(t)$. Note that any errors in $y_b(t)$ will be corrected by w_{p_i} so that the true solution, y_{p_i}, will still be accurate.

There are two aspects of this procedure that can lead to significant computational improvement over Method 2. First, instead of evolving a ROM which consists of the original solution augmented with a basis constructed from the w_{p_i}, one can evolve 5.2 and obtain an approximation to y_{p_i} by adding the base interpolant to v_{p_i}. If the number of vectors required to form an accuracte interpolant is significantly less than the number of basis elements in the base solution POD model, then the per timestep cost of evolving the POD model can be greatly reduced. This follows because the computational cost per time step for evolving a POD solution using a POD basis is proportional to the number of POD basis elements. The cost is also proportional to the dimension of the vectors, and can be quite large for higher dimensional problems. The evaluation of the interpolant at each time step using Hermite interpolation only requires 6 vectors, and hence if the POD basis has more than 6 vectors, we reduce the cost of going back and forth between the POD basis and the standard basis required for evolving the ROM. Second, the interpolant of the base solution can be determined adaptively and produced incrementally as the base solution evolves, and so there is no need to compute and store a large numer of solution snapshots as is reqired when creating a POD based ROM for the base solution.

Assuming that both the number of POD vectors necessary to capture w_{p_i} and the number required to form the interpolant are small, the computational savings over Method 2 may be substantial. The ability to create the basis without having to store and work with all the snapshots will be a big improvement, especially for higher dimensional problems.

In the numerical example considered below, far fewer POD basis elements are necessary to obtain a given accuracy using this method (see Section 5.3.4) than solving directly for the solution of the perturbed system.

5.3 Application of Basis Augmentation

Below, we implement and investigate the ability of augmented basis reduced order models to accurately capture the solution behavior as problem parameters are varied. We consider two PDE's in 1-D: the Fitz-Hugh Nagumo equation and the 1-D reaction-diffusion equation introduced in Chapter 2.1. In Section 5.3.1, we apply Method 1 while varying one parameter,

while Section 5.3.3 deals with the more complicated situation of varying 3 problem parameters in the Fitz-Hugh Nagumo equation. In Section 5.3.2, we apply Method 2 to the 1-D reaction-diffusion equation. In Section 5.3.4, we apply Method 3 to the 1-D reaction-diffusion equation. Lastly, we compare the results of using augmented bases constructed using Method 1 and Method 2 with Method 0, the latter method being the one in which a single SVD analysis is done on a combined set of snapshots 5.3.5.

5.3.1 FitzHugh-Nagumo equation using Method 1

First, we consider the nonlinear reaction-diffusion arising in neuron modeling: the FitzHugh-Nagumo model. This is a two-dimensional simplification of the Hodgkin-Huxley model of spike generation in squid giant axons and has been used to model a wide range of phenomenon from slime mold amoeba to cardiac tissue. Locating bifurcations in the FHN model is particularly interesting because it is often difficult to find bifurcations via simulation [DSK10]. The FHN model has the form

$$\varepsilon \frac{du}{dt} = \varepsilon^2 u_{xx} + u(u - .1)(1 - u) - y \ \ on\ [0, 1.3] \tag{5.3}$$

$$\frac{dy}{dt} = bu - \lambda y$$

$$u(x, 0) = 0 \quad y(x, 0) = 0$$

$$u_x(l, t)) = 0 \quad u_x(0, t) = -i_0(t)$$

We will use the scaling of Keener and Sneyd [KS98] that

$$\varepsilon = 0.015, \quad b = 0.5, \quad and \quad \lambda = 2$$

u is the voltage and y is the feedback. The stimulus is $i_0 = mt^3e^{-15t}$ where m is a constant. The size of the stimulus determines whether the system will produce a stable traveling wave or will go to a rest state. For large i_0, the solution will be a traveling wave shown in Figs. 5.1 and 5.2. For small i_0, the solution will tend to $u = y = 0$, illustrated in Figs. 5.3 and 5.4.

For m in the interval $[12000, 15000]$, a bifurcation occurs. Our goal is to determine an augmented basis that can accurately capture the solution over this range, including the bifurcation. As a base solution, we use the solution where $m = 13500$, a value that corresponds to

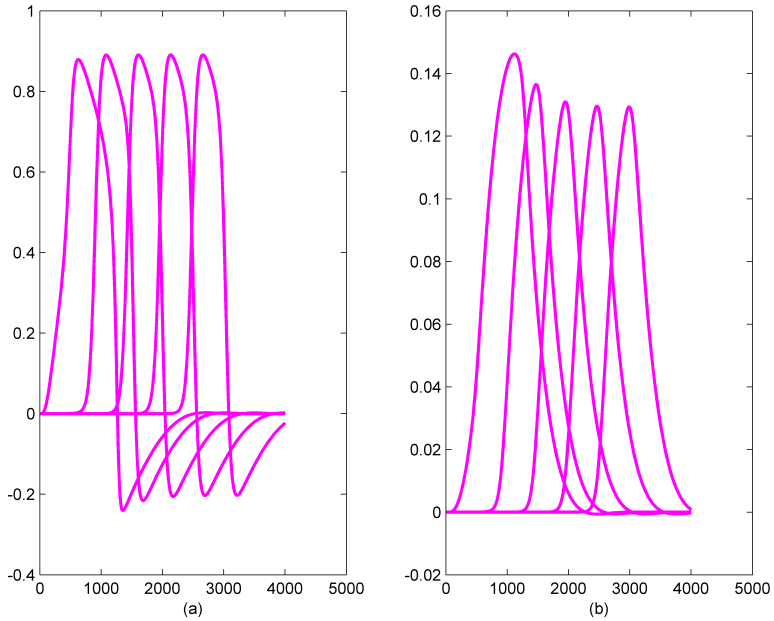

Figure 5.1: Traveling wave solutions for $m = 13500$ shown at different spatial points, x. (a) u(x,t) (b) y(x,t).

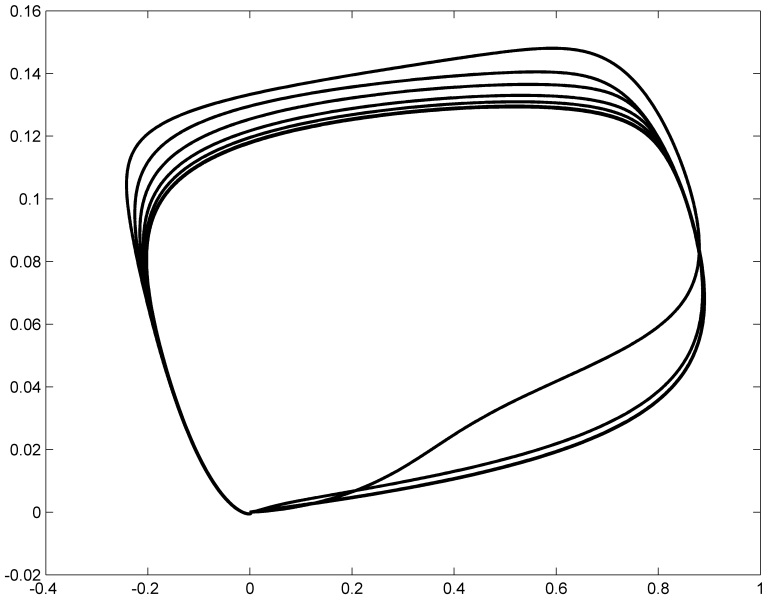

Figure 5.2: Phase-space diagram of u and y at different spatial points, x, for $m = 13500$.

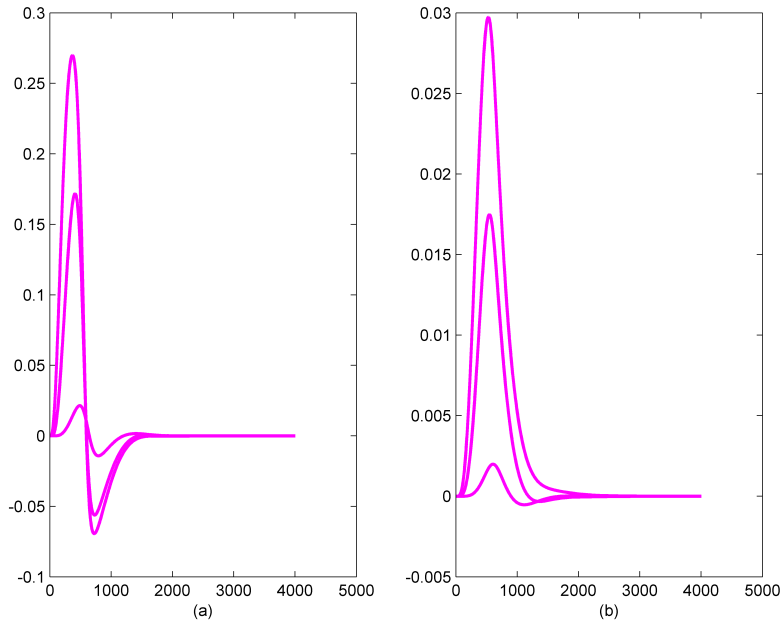

Figure 5.3: Decaying solutions for $m = 12000$ shown at different spatial points, x. (a) u(x,t) (b) y(x,t).

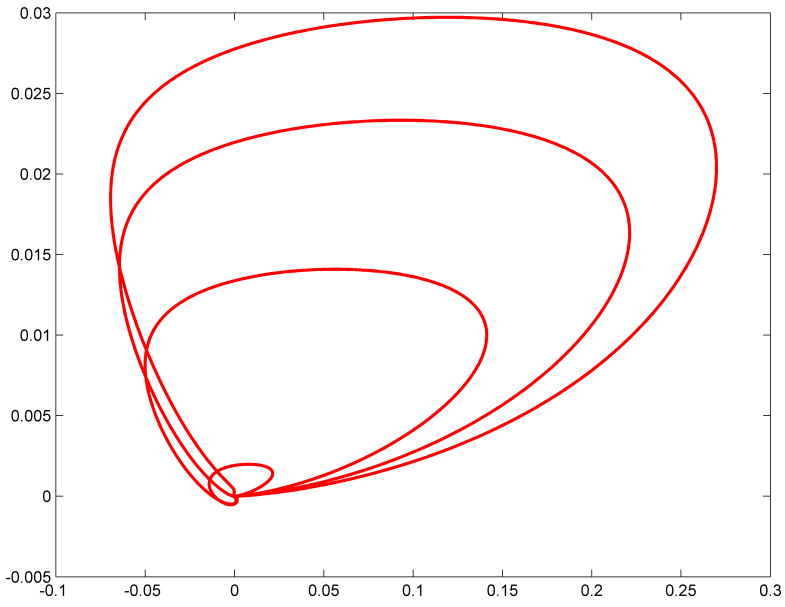

Figure 5.4: Phase-space diagram of u and y at different spatial points, x, for $m = 12000$.

a traveling wave. We form a POD basis for $m = 13500$ such that a tolerance, $TOL = 1.1\%$, is met, and then augment this basis with elements selected from the systems which exhibit the most extreme behavior. For this interval, these elements will correspond to the systems' where $m = 12000$ and $m = 15000$.

By using method of lines, we convert the PDE to an ODE using standard centered difference in space to approximate the spatial derivatives on a grid of size 250. We then use 3rd order Runge Kutta time integration with 4000 time steps. The POD model is constructed from the 4000 equally-spaced snapshots collected over the time interval $[0, 4]$. The projection matrix is constructed from 130 modes. The relative errors between the full and reduced models sampled over the domain are

$$\frac{||u - \tilde{u}||}{||u||} < .011$$
$$\frac{||y - \tilde{y}||}{||y||} < .010$$

which indicates that the ROM of dimension 130 represents the full order model very well for $m = 13500$.

In order to make a basis robust over the range of m, in addition to the original basis, we include singular vectors from the POD bases derived from the full system run at $m = 15000$, and at $m = 12000$. This brings up the question of whether to allocate more vectors from the system at $m = 15000$, or $m = 12000$. We base our decision on selecting singular vectors associated with the extreme parameter solutions that are not well captured by the ROM using the original basis. The relative errors of y and u between the full dimensional system with $m = 15000$, and its projected system sampled over the domain and constructed using the original POD basis derived from the time slices for $m = 13500$, are less than .73%. This is a smaller relative error than in the original system with $m = 13500$. In contrast, the relative errors sampled over the domain between the full system with $m = 12000$, and its projected system using the original POD basis are around 2.5% (only the relative errors where the true solution is greater than .01 are considered). The solution at $m = 12000$ decays to zero which means that a bifurcation occurs between 12000 and 13500. Because the errors are very small at $m = 15000$ and there are no major dynamical changes for $m \in [13500, 15000]$, it seems

reasonable to assume that the original POD basis will work well on this upper range. Since there is a bifurcation for $m \in [12000, 13500]$ and the relative errors are larger at $m = 12000$, the basis should be augmented with singular vectors derived from the system with the smaller stimulus ($m = 12000$).

Using Method 1, we form two augmented bases, $B1$ and $B2$, by taking 4 and 8 singular vectors respectively corresponding to the largest singular values for the system at $m = 12000$. The singular vectors are then combined with the original POD basis of rank 130 and re-orthogonalized. The basis $B1$ has rank 134 and $B2$ has rank 138.

The original basis containing 130 modes generated relative errors for both u and y of less than 1.1% for m sampled in $[13500, 15000]$. Thus, the use of an augmented basis was not necessary. For $m \in [12666, 13500]$, the original basis preserves the qualitative behavior of the solution and maintains a relative error of $< 2\%$ for $m > 13200$. As m approaches 12666, the error becomes as large as 130%. Although the system goes through a bifurcation between $m = 12665$, and $m = 12666$, the ROM is unable to capture its behavior and generates a traveling wave instead of decaying wave solution as shown in Fig. 5.5. It only indicates the existence of the bifurcation at $m = 12631$. It is necessary to include 158 basis elements in the original basis for the ROM to show the bifurcation at its correct location.

The new bases, $B1$ and $B2$, perform significantly better for $m \in [12000, 13500]$ compared to the POD bases of the same rank formed from the original system with $m = 13500$. For m greater than 13200, the relative errors of u and y for both the original and the augmented bases of the same rank were less than 1%. In the Tables below (5.1), we compare the relative errors generated by the ROM's formed using $B1$ and $B2$ as opposed to bases of the same ranks formed from the original system. The recorded values are an average of four relative errors sampled over the domain where the true solution is $> .01$ and $m \in [12666, 13200]$.

Next, in Tables 5.2, we compare the relative errors generated by the ROM's formed using $B1$ and $B2$ to the errors generated using bases of the same ranks formed from the original system. The recorded values are an average of three relative errors sampled over the domain where the true solution is $> .01$ and $m \in [12000, 12625]$.

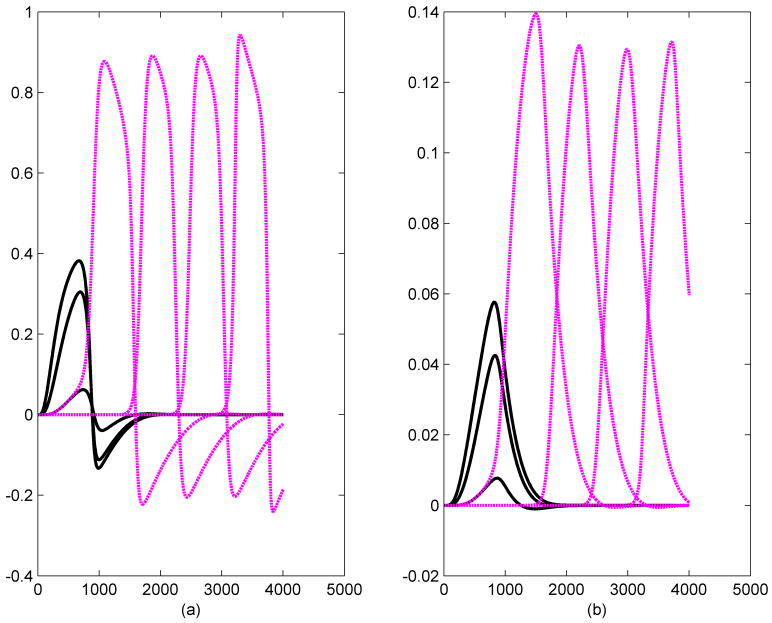

Figure 5.5: $m = 12660$, black is the true solution and purple is the solution generated by the ROM using the original basis with rank 134 (a) u(x,t) (b) y(x,t).

The results indicate that the basis augmented using Method 1 consistently improves the ROM, particularly close to the bifurcation. For $m \in [12631, 12665]$, the results are striking. $B1$ captures the correct qualitative behavior with relative errors sampled over the domain under 4% as opposed to the original basis which shows the bifurcation happening between 12631 and 12632 (the "true" interval being for $m \in [12655, 12666]$). With only four more basis elements, $B2$ produces highly accurate solutions with relative errors under 1%.

For $m < 12625$, there is an order of magnitude ten improvement between the relative errors in the solutions generated using the original basis and $B1$ (both with the same rank) while for $m \in [12666, 13200]$, there is an order of magnitude five improvement. For $m \in [13200, 15000]$, the original and the augmented bases both work well.

Table 5.1: Listed below are the average relative $\|.\|_2$ errors in u and y produced by using the new bases, $B1$ and $B2$ of rank 134 and 138 respectively in comparison with the errors using the original basis of corresponding rank. In $B1$, four vectors corresponding to the smallest singular values in the original POD basis have been replaced by singular vectors derived from the system at $m = 12000$. In $B2$, eight vectors corresponding to the smallest singular values in the original POD basis have been replaced by singular vectors derived from the system at $m = 12000$. The errors are listed according to the size of m, $m \in [12666, 13200]$, and values $< .1\%$ are reported as 0.

Average Relative Errors of u, rank=134						
m	13200	13000	12800	12700	12680	12666
Original Basis	0.009	0.016	0.039	0.232	0.296	0.826
B1	0.001	0.001	0.002	0.007	0.013	0.240

Average Relative Errors of y, rank=134						
m	13200	13000	12800	12700	12680	12666
Original Basis	0.006	0.009	0.022	0.123	0.217	0.525
B1	0.000	0.001	0.001	0.003	0.007	0.131

Average Relative Errors of u, rank=138						
m	13200	13000	12800	12700	12680	12666
Original Basis	0.006	0.028	0.038	0.185	0.187	0.781
B2	0.000	0.000	0.000	0.001	0.002	0.052

Average Relative Errors of y, rank=138						
m	13200	13000	12800	12700	12680	12666
Original Basis	0.004	0.016	0.032	0.102	0.104	0.490
B2	0.000	0.000	0.000	0.001	0.001	0 .028

Table 5.2: Listed below are the average relative $\|.\|_2$ errors in u and y produced by using the new bases, $B1$ and $B2$, and the original POD bases, both rank 134 and 138 respectively. In $B1$, four vectors corresponding to the smallest singular values in the original POD basis have been replaced by singular vectors derived from the system at $m = 12000$. In $B2$, eight vectors corresponding to the smallest singular values in the original POD basis have been replaced by singular vectors derived from the system at $m = 12000$. The errors are listed according to the size of m, $m \in [12000, 12625]$, and values $< .1\%$ are reported as 0.

Average Relative Errors of u, rank=134					
m	12000	12300	12500	12600	12625
Original Basis	0.013	0.025	0.055	0.154	0.296
B1	0.000	0.001	0.001	0.003	0.005

Average Relative Errors of y, rank=134					
m	12000	12300	12500	12600	12625
Original Basis	0.014	0.022	0.045	0.116	0.217
B1	0.000	0.001	0.001	0.002	0.003

Average Relative Errors of u, rank=138					
m	12000	12300	12500	12600	12625
Original Basis	0.013	0.018	0.038	0.103	0.184
B2	0.000	0.000	0.000	0.000	0.000

Average Relative Errors of y, rank=138					
m	12000	12300	12500	12600	12625
Original Basis	0.014	0.016	0.032	0.078	0.135
B2	0.000	0.000	0.000	0.000	0.000

5.3.2 Basis augmentation for the 1-D Reaction-Diffusion Equation using Method 2

Next, we look at the 1-D reaction-diffusion model

$$\frac{dy}{dt} = ay^2(2-y) + by_{xx} \tag{5.4}$$

$$y(0,t) = 2,\ y(2,t) = 0,\ y(x,0) = 0$$

where a and b are constants. The PDE is defined on the interval $[0, 2.0]$ and for time, on $[0, 1.0]$. The reaction coefficient, $a = 20.0$ and the diffusion coefficient, $b = 0.025$. Using method of lines, we convert the PDE to an ODE using standard centered difference in space to approximate the spatial derivatives on a grid of 300 points. The resulting ODE has dimension $n = 300$. The POD model is constructed from 3100 equally-spaced snapshots collected over the time interval $[0, 1.45]$ (this is to ensure that perturbed solutions do not overreach the basis). 3rd order Runge Kutta is used for the time discretization. The POD projection matrix is constructed from 33 modes. The relative error between the full and reduced models at time $t = 1.0$ is

$$\frac{||y - \tilde{y}||}{||y||} = .011$$

indicating a ROM of dimension 33 is a good model for the full dimensional case.

We choose for our base solution to be that associated with the value of the diffusion constant, $b = .025$, and select a number of basis elements so that an error less than a tolerance of 1.1% is achieved. We seek to augment the POD basis so that solutions of the system can be accurately computed for b in the range $b_{min} = .005$ to $b_{max} = .045$. The solutions to the reaction-diffusion equation at the final time, $t = 1.0$, as b varies over this interval are shown in Fig. 5.6. Both the speed and the shape of the fronts change dramatically as b changes.

The allocation of the additional singular vectors depends on how well the original basis models the perturbed systems for b_{min} and b_{max} since the solutions for these values exhibit the most extreme behavior. Because $y_{b_{min}}(t)$ and $y_{b_{max}}(t)$ are sufficently close to $y(t)$ over the time interval considered, the calculation of, $w_{b_{min}}(t) = y_{b_{min}}(t) - Py_{b_{min}}(t)$ can be estimated by $w_{b_{min}}(t) \cong y_{b_{min}}(t) - y(t)$ (similarly, $w_{b_{max}} \cong y_{b_{max}}(t) - y(t)$). We start by recording $w_{b_{min}}(t_k) = y_{b_{min}}(t_k) - y(t_k)$ and $w_{b_{max}}(t_k) = y_{b_{max}}(t_k) - y(t_k)$ at each time step.

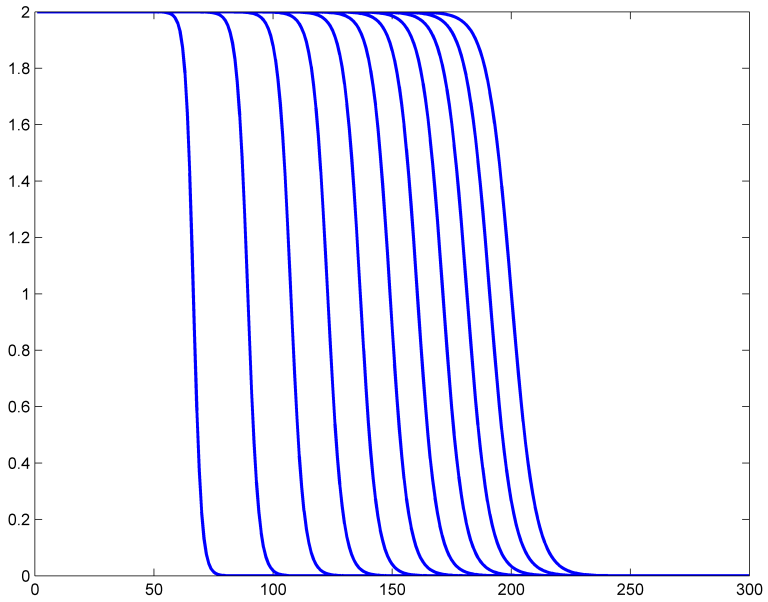

Figure 5.6: The solutions of the system, 5.4 at t=1.0 for 20 equally spaced diffusion constants starting with $b = .005$ and ending with $b = .045$.

Next, the original POD basis (derived from the time slices for $b = .025$) is applied to the two perturbed systems. The relative error of $y_{b_{max}}$ between the full system with $b_{max} = .045$, and its projected system (using the original POD basis) at the final time is order 10^{-3}. This is a smaller relative error than in the original system. When $b_{min} = .005$, however, the relative error for $y_{b_{min}}$ at the final time is 32%. This indicates that the additional singular vectors should be selected from the system with the least diffusion.

We examine the ability of the two ways of augmenting the basis (Method 1 and Method 2) and the original, non-augmented basis (where $b = .025$) to model perturbed systems as diffusion is varied. The vectors used to augment the original basis are taken from 2200 equally-spaced snapshots collected over the time interval, $[0, 1.0]$, for the system with diffusion constant, $b = .005$. Using Method 2, the singular vectors to be combined with the original basis are selected from the SVD of the vectors $w_{b_{min}}$ as opposed to $y_{b_{min}}$. We included all $w_{b_{min}}$'s singular vectors so that the energies sum to approximately 99% of the total energy. The singular vectors corresponding to the largest singular values were then appended to the original basis with 33

Table 5.3: Listed below are the relative $\|.\|_2$ errors produced by using the original system's ($b = .025$) POD basis, the augmented basis using Method 1, and the augmented basis using Method 2, all with equal rank. The singular vectors corresponding to the largest singular values in the POD basis for $b_{min} = .005$ have been used for augmentation. The relative errors are listed for diffusion coefficients equally spaced between.005 and .025. a), b), and c) show the results for different dimensions of the bases.

a) Bases have rank 41. 8 singular vectors used for augmentation											
b	.005	.007	.009	.011	.013	.015	.017	.019	.021	.023	.025
Original Basis	.1188	.0554	.0274	.0145	.0079	.0046	.0027	.0017	.0011	.0008	.0005
Method 1	.0794	.0317	.0163	.0101	.0068	.0051	.0037	.0027	.0020	.0015	.0012
Method 2	.0574	.0252	.0167	.0116	.0085	.0061	.0043	.0032	.0023	.0018	.0014

b) Bases have rank 43. 10 singular vectors were used for augmentation											
b	.005	.007	.009	.011	.013	.015	.017	.019	.021	.023	.025
Original Basis	.0968	.0427	.0204	.0104	.0055	.0031	.0018	.0011	.0007	.0005	.0003
Method 1	.0743	.0290	.0144	.0081	.0055	.0036	.0027	.0020	.0016	.0012	.0009
Method 2	.0349	.0162	.0125	.0091	.0072	.0053	.0039	.0029	.0021	.0016	.0013

c) Bases have rank 45. 12 singular vectors were used for augmentation											
b	.005	.007	.009	.011	.013	.015	.017	.019	.021	.023	.025
Original Basis	.0791	.0331	.0151	.0074	.0038	.0021	.0012	.0008	.0005	0.0003	.0002
Method 1	.0388	.0143	.0082	.0052	.0041	.0029	.0022	.0018	.0014	.0011	.0009
Method 2	.0197	.0098	.0084	.0070	.0054	.0045	.0034	.0025	.0019	.0015	.0012

elements and re-orthonormalized. An augmented basis using Method 1 was also constructed. In this case, the vectors were selected from the SVD of snapshots of $y_{b_{min}}$. The vectors corresponding to the largest singular values were then combined with the original basis with 33 elements and re-orthonormalized.

In Tables 5.3 above, the relative $\|.\|_2$ errors at time $t = 1.0$ are compared for the three different basis types as diffusion ranges from $b_{min} = .005$ to $b = .025$. The bases all have equal

rank, and the augmented bases have an equal number of singular vectors taken from the SVD's of $w_{b_{min}}$ and $y_{b_{min}}$ (for Method 1 and Method 2 respectively). In a), 8 singular vectors were combined with the original basis of 33 elements while 10 were used in b) and 12 in c). For the singular vectors taken from the SVD of $w_{b_{min}}$, the corresponding energies of the retained modes sum to 99.3% of the total energy for 8 singular vectors, and 99.9% for 12. The results for $.025 < b \leq .045$ are not included because the relative errors are $< .001$ for all three basis types for the dimensions considered (≥ 41).

The relative improvements of using Method 1 and Method 2 compared to the non-augmented basis (for values of b where the relative $\|.\|_2$ errors $> 1\%$) can be seen in Fig 5.7.

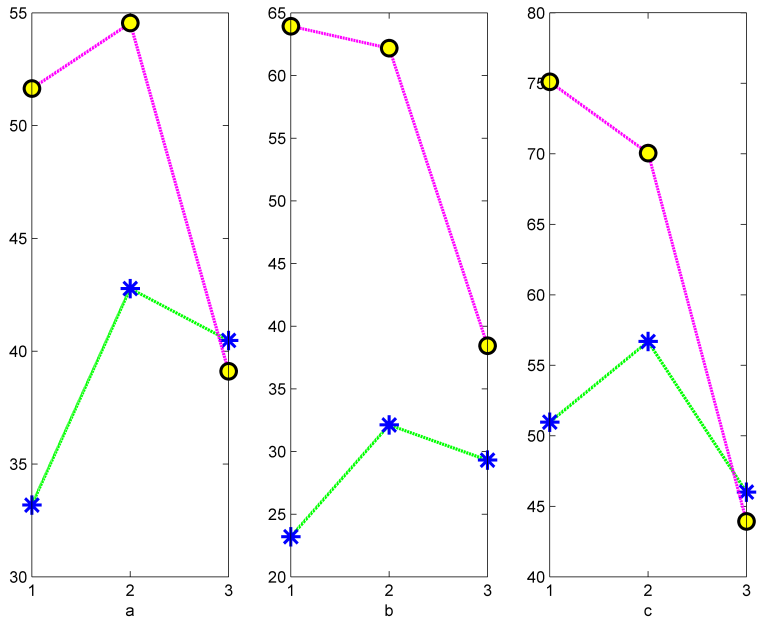

Figure 5.7: Displayed above are the relative improvements in the errors caused by using the augmented basis in comparison to the POD basis derived from the base solution. Circles correspond to the augmented basis made using Method 2 and stars to Method 1. a) Bases have rank 41. 8 singular vectors were used for augmentation b) Bases have rank 43. 10 singular vectors were used for augmentation c) Bases have rank 45. 12 singular vectors were used for augmentation

By adjoining vectors from the perturbed system to the original basis with 33 elements, we

are able to greatly improve the ability of the ROM to model perturbations in the diffusion parameter for a given basis dimension. As diffusion decreases, the relative errors from using the augmented bases are one third to one quarter the size of those from using the basis from the original system (of equivalent rank). When the augmented bases are used to solve problems close to the base solution (in this case, where the diffusion parameter has the value $b = .025$), it is expected that they will not do as well since the augmented vectors are being used to replace base solution singular vectors that have been optimally selected to reproduce the base solution. Despite this, both augmented bases still have small errors when modeling nearby to the base solution, and they work much better than the original basis on the systems where it functions poorly.

As shown in Fig. 5.7, the augmented bases designed using both Method 1 and Method 2 generated improvements in the quality of the reduced order models in comparison to the use of the original basis when applied to the perturbed systems that were poorly modeled by original basis. Method 2 in particular generated improvements in the error of 45% to 75%.

5.3.3 FitzHugh-Nagumo with 3 Perturbed Parameters using Method 1

Next, we return to the FitzHugh-Nagumo equation (see Section FHBaug for more details on the FHN system).

$$\varepsilon \frac{du}{dt} = \varepsilon^2 u_{xx} + u(u-r)(s-u) - y \;\; on\; [0, 1.3]$$

$$\frac{dy}{dt} = bu - \lambda y$$

$$u(x,0) = 0 \quad y(x,0) = 0$$

$$u_x(l,t)) = 0 \quad u_x(0,t) = -i_0(t)$$

$$\varepsilon = 0.015, \quad b = 0.5, \quad \lambda = 2, \quad and \quad i_0 = mt^3 e^{-15t}$$

Instead of forming an augmented basis for the situation when only one parameter is perturbed, we investigate the behavior of the ROM models as three parameters are varied. These values vary over the following values

$$r \in [.07, .13], \;\; s \in [.97, 1.3], \;\; and \;\; m \in [10000, 16000] \tag{5.5}$$

For these sets of parameters, solutions are either traveling waves or decay to zero depending on the size of the impulse. A bifurcation occurs for $m \in [10000, 16000]$ for each (r, s) pair $\in [.07, .13] \times [.97, 1.3]$. The range of m was selected so that the percentage of traveling and decaying solutions generated are roughly equal for all sets (r, s, m) in 5.5.

Perturbing r and s dramatically changes both the value of m where the bifurcation occurs, and the shape, height, and speed of the waves. In order to evaluate the quality of the basis, we will consider nine sets of parameters, $p_j = (r_j, s_j)$ sampled from the interval $[.07, .13] \times [.97, 1.3]$ for the values of r and s in the polynomial term $u(u-r)(s-u)$ in the FHN equation. The values are listed in Table 5.4 along with the corresponding value of $m = m_i$ where the bifurcation occurs (rounded down to the nearest integer).

Table 5.4: Listed below are the nine sets of parameters for the polynomial in the Fitz-Hugh Nagumo equation and the corresponding value of $m = m_i$ where the bifurcation occurs. The values of the m_i's have all been rounded down to the nearest integer.

The value of m where the bifurcation occurs for the specific sets of parameters		
p_i	(r_i, s_i)	m_i
p_1	$(.07, .97)$	10604
p_2	$(.07, 1.0)$	10300
p_3	$(.07, 1.03)$	10034
p_4	$(.1, .97)$	13015
p_5	$(.1, 1.0)$	12665
p_6	$(.1, 1.03)$	12368
p_7	$(.13, .97)$	15650
p_8	$(.13, 1.0)$	15220
p_9	$(.13, 1.03)$	14875

Some examples of the the system's reactions to perturbations in the parameters of the polynomial can be seen in Figs. 5.8 and 5.9. The solutions are displayed for four different sets of $(p_i, m_i + 100)'s$ and are all traveling waves. For each p_i, the variable m is selected to equal

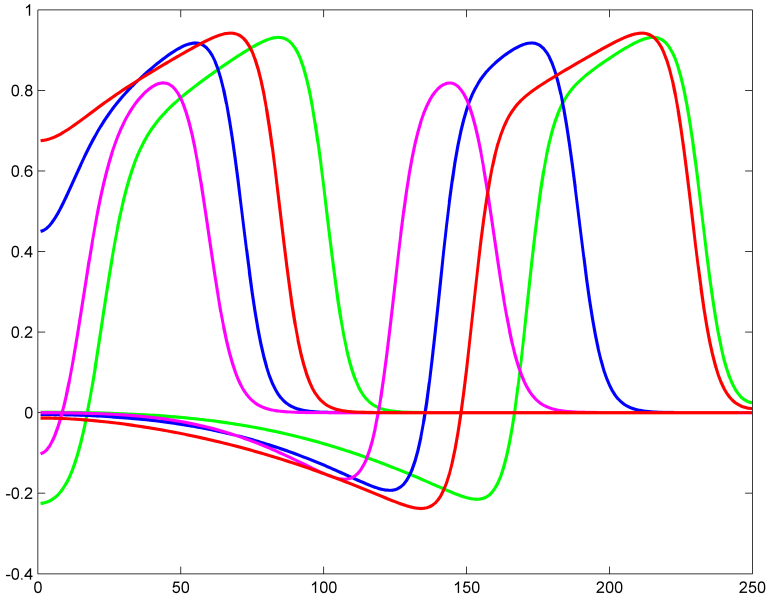

Figure 5.8: u for four different sets of $p_i's$ and $m's$. The value of m is selected to be equal $m_i + 100$. For each set of parameters, u is shown for $t = 1.3$ and $t = 2.6$. Different colors are used for different values of the parameters.

$m_i + 100$.

It is clear from the figures and table that both the size of the impulse necessary to cause the bifurcation, and the type of solutions generated are very sensitive to slight changes in the values of the parameters r and s. We apply Method 1 to this system. The POD bases derived from traveling wave solutions for these parameters performed well for the entire range of parameters (see 5.5) as long as they were used to create solutions that were not too close to a bifurcation. Specifically, in order for the ROM to generate good solutions for each set of roots, m needed to be 100-200 away from the value $m = m_i$ that causes the bifurcation. Because of this, we focus on improving the basis in order to capture accurately solutions near the bifurcation points. The quality of the augmented POD basis will be evaluated based on its ability to both correctly track the bifurcation for each pair of parameters, (r, s) and produce accurate solutions.

We set up the full dimensional system as in Section 5.3. The grid has size 250 and we use forward Euler time integration with 4000 time steps. We form the augmented basis by

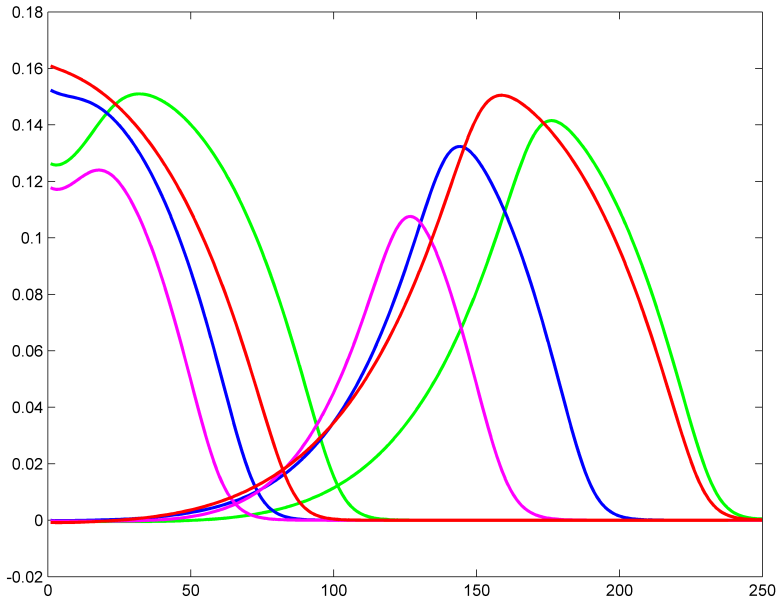

Figure 5.9: y for four different sets of $p_i's$ and $m's$. The value of m is selected to be equal $m_i + 100$. For each set of parameters, y is shown for $t = 1.3$ and $t = 2.6$. Different colors are used for different values of the parameters.

combining basis elements from a traveling wave and a decaying solution. To find a solution of each type, we randomly selected sets of parameters, $(r_i, s_i, m_i) \in [.07, .13] \times [.97, 1.3] \times [10000, 16000]$, solved the resulting system, and observed the solutions.

A suitable traveling wave solution and a decaying solution were found with parameters (r, s, m) equal to $(0.0876, 0.9873, 12579)$ and $(0.1275, 0.9874, 14338)$ respectively. A POD basis was constructed from the 4000 equally-spaced snapshots collected over the time interval $[0, 4]$ for each set of parameters.

As before, we form the augmented basis by first creating a POD basis the generates an accurate reduced order model for the system that produces the traveling wave . Using a POD basis of dimension 135, the relative errors between the full and reduced models sampled over the domain are

$$\frac{||u - \tilde{u}||}{||u||} < .019$$
$$\frac{||y - \tilde{y}||}{||y||} < .011$$

for $(r, s, m) = (0.0876, 0.9873, 12579)$ (a traveling wave solution). In order to make this basis more robust, we combine it with the four singular vectors corresponding to the largest singular values from the decaying solution where $(r, s, m) = (0.1275, 0.9874, 14338)$ and re-orthonormalize.

Throughout the discussion of the results, the system with $(r, s) = (0.0876, 0.9873)$ will be referred to as the original system, and the POD basis derived from its snapshots as the original basis. When the results of using the original basis is compared to those derived using the augmented basis, the bases have identical dimension. In the interest of brevity, we have only included the relative errors in y because in every test, they were close to the relative errors in u. The errors are averages calculated depending on whether the solution is a traveling wave or decaying. For propagating wave solutions, the errors recorded below are the average of 4 relative errors sampled over the domain. In the case of a decaying solution, the errors are the average of 3 relative errors sampled over the domain where the true solution is $> .01$.

We first investigate if the augmented basis is at least capable of correctly capturing the solution behavior around the bifurcation with the parameter values, $(r, s) = (0.0876, 0.9873)$,

associated with the base solution used to create the augmented basis. In this case, the bifurcation occurs at $m_i = 11798$. The augmented basis correctly generates a decaying wave solution with a relative error of 5.6%. In contrast, the original basis does not detect the location of the bifurcation and continues to produce a traveling wave until $m = 11749$.

Table 5.5: Listed below are the average relative $\|.\|_2$ errors in y produced by using the augmented basis and the original POD basis, both with rank 139 where $(r, s) = (0.0876, 0.9873)$. The bifurcation occurs at $m = 11798$ (rounded down to the nearest integer), and values $< .1\%$ are reported as 0.

Average Relative Errors of y, rank=139						
m	11624	11724	11749	11799	11824	11924
Original Basis	0.068	0.176	0.3328	0.643	0.124	0.038
Augmented Basis	0.000	0.001	0.001	0.095	0.003	0.001

For $m \in [11624, 11924]$, the augmented basis preserves the qualitative behavior of the solution. For decaying solutions, even when the original basis produces the correct type of solution, there is a dramatic difference in accuracy between ROM's. The augmented basis produces errors $\leqslant .1\%$ while the original basis generates errors ranging from 6.8% to 33%. Directly above the bifurcation, there is a difference of 55% between the relative errors of the two bases with the augmented basis working significantly better. The errors for the augmented basis rapidly drop off as m increases, and for $m > m_i + 25$, they are around .1% as opposed to 12% and 3.8% for the original basis.

In the second test, we allowed the (r, s, m) parameters to range over the intervals in 5.5. In order to evaluate the accuracy with which solutions can be captured for these parameters, we consider the nine parameter values $p_i = (r_i, s_i)$ listed in Table 5.4 (sampled from $[.07, .13] \times [.97, 1.3]$). We compare the augmented basis to the original basis at each p_i and see how it performs as m varies in the corresponding interval $[m_i - (125\ to\ 200), m_i + 125]$ around the bifurcation. As before, the value of m where the bifurcation occurs is rounded down to the nearest integer.

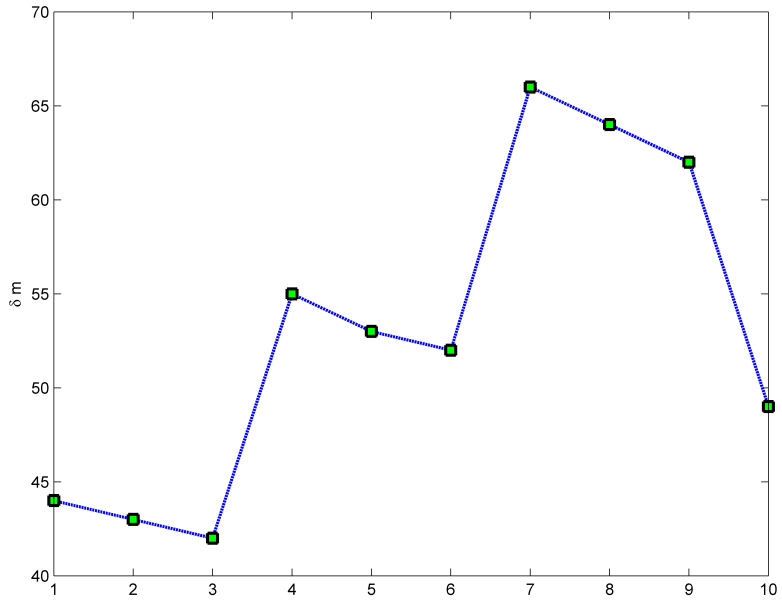

Figure 5.10: The difference, Δm_i, between the true location of the bifurcation, m_i, and the m value where it is detected by the original basis for each p_i.

We first compare the ability of the augmented basis versus the original basis of equal dimension ($\cong 139$) to track the location of the bifurcation at each of the nine test points, p_i. Fig. 5.10 shows the difference, Δm_i, between the true location of the bifurcation, m_i, and the m value where it is detected by the original basis for each corresponding p_i. In all the test cases, the ROM using the original basis generates a traveling wave instead of a decaying solution until m gets well below the value where the bifurcation is supposed to occur. The Δm_i's varies between 64 and 42.

In contrast, the augmented basis switches from producing a traveling wave to a decaying solution within one of the true value of the bifurcation. The relative errors caused by using the augmented basis at m_i are included in Table 5.6 (because m_i is rounded down to the nearest integer, these are all decaying solutions). With the exception of the parameters equal to p_4, the relative errors are all $< 13\%$ and half are $< 4\%$. Even for p_4, however, the relative error drops off immediately to 5% at $m_i - 1$. These results are quite remarkable when compared to the original basis which was unable to detect the bifurcation at all for $\Delta m_i \leq 42$, much less

with this degree of accuracy.

Table 5.6: Listed below are the average, relative $\|.\|_2$ errors in y produced by using the augmented basis at m_i (or $m_i - 1$ for p_5 and p_7).

The relative errors in y									
p_i	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
Relative Error	0.070	0.032	0.031	0.469	0.036	0.081	0.031	0.126	0.090

Next, we investigate the accuracy of the solutions generated by the original versus the augmented basis for each p_i. At each set of parameter values, $p_i = (r_i, s_i)$, we compare the relative errors produced by the two different bases for a variety of m values around the corresponding location of the bifurcation, m_i. Specifically, we record the errors of y for m values of $m_i + 1$, m_i+26, and m_i+126 (which all produce traveling wave solutions), and $m_i-\Delta m_i$, $m_i-\Delta m_i-25$, and $m_i-\Delta m_i-125$ below the bifurcation (which are all decaying solutions). Recall from above that $m_i-\Delta m_i$ is the first value of m for which the original basis produces the correct qualitative behavior. Specifically, it correctly generates a decaying solution instead of a traveling wave. The exact value depends on the roots selected.

Table 5.7 summarizes the results. The same information, in a slightly different form, is presented in Figs. 5.11- 5.13.

There is a striking difference in accuracy between original and augmented reduced order models. We first consider p_1, p_2, and p_3 (see Fig. 5.11 for a graphical depiction). Below the bifurcation, the augmented basis generates solutions with accuracy of about .1%. The original basis, in contrast, produces errors from 5.6% to 33.4%. Just above the bifurcation at $m = m_i+1$, the augmented system has errors of only 10% to 50% of the original. For the largest two m values, the augmented ROM consistently produces solutions either as good, or much better solutions than its counterpart.

Similarly, for p_4, p_5, and p_6 (see Fig. 5.12), the augmented basis generates solutions with accuracy of .1% below the bifurcation. The ROM based on the original basis has errors ranging

Table 5.7: Listed below are the average, relative $\|.\|_2$ errors in y for the original and augmented bases calculated at the 9 sets of roots $p_i = (r_i, s_i)$. Note that the location of the bifurcation, m_i, and the $m = m_i - \Delta m_i$ value sufficiently small for the original basis to produce the correct qualitative behavior are specific for each p_i.

The average, relative $\|.\|_2$ errors in y.							
p_i	Basis	$m_i - \Delta m_i - 125$	$m_i - \Delta m_i - 25$	$m_i - \Delta m_i$	$m_i + 1$	$m_i + 26$	$m_i + 126$
p_1	Original	0.063	0.168	0.334	0.567	0.117	0.034
	Augmented	0.000	0.001	0.001	0.055	0.009	0.008
p_2	Original	0.059	0.156	0.306	0.843	0.106	0.037
	Augmented	0.009	0.009	0.001	0.372	0.029	0.037
p_3	Original	0.056	0.148	0.293	0.654	0.1109	0.0544
	Augmented	0.001	0.001	0.001	0.185	0.053	0.054
p_4	Original	0.200	0.369	1.796	0.548	0.159	0.065
	Augmented	0.001	0.001	0.001	.048	0.030	0.029
p_5	Original	0.0719	0.185	0.332	0.581	0.129	0.041
	Augmented	0.001	0.001	0.001	0.062	0.007	0.006
p_6	Original	0.067	0.171	0.315	0.431	0.1077	0.032
	Augmented	.001	.001	.001	0.0320	0.013	0.013
p_7	Original	0.105	0.284	0.535	0.507	0.237	0.210
	Augmented	0.001	0.001	0.001	0.105	0.092	0.092
p_8	Original	0.089	0.220	0.382	0.642	0.1829	0.085
	Augmented	0.001	0.001	0.001	0.091	0.048	0.047
p_9	Original	0.081	0.198	0.341	0.610	0.143	0.052
	Augmented	0.001	0.001	0.001	0.082	0.020	0.020

from from 6.7% to 179.6%. Immediately above the bifurcation, the original system has errors of only $\geqslant 43\%$ while the augmented errors $\leqslant 6.2\%$. At $m = m_i + 26$, the original solution has errors from 10.7% to 16% as opposed to the augmented basis whose errors vary from 0.7% to 3%. For the largest m value, the augmented varies between 0.6% and 1.3% and the original

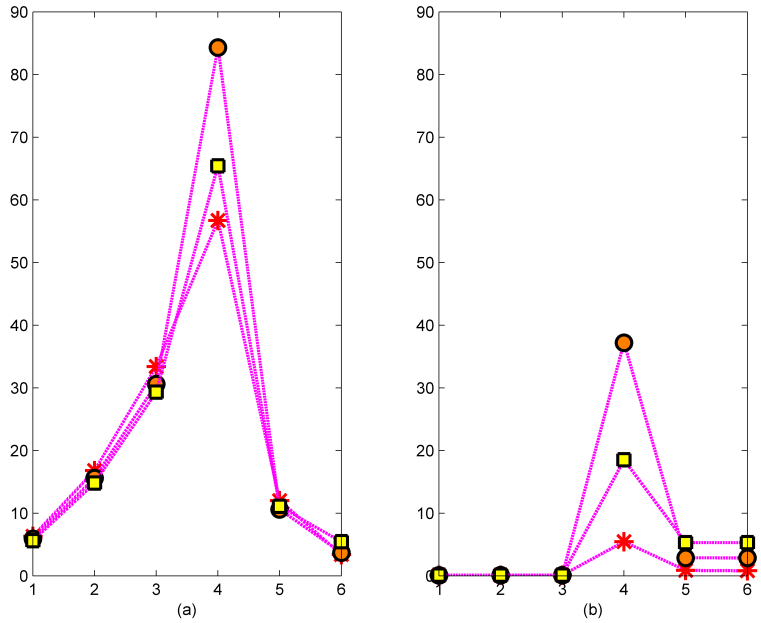

Figure 5.11: The average, relative $\|.\|_2$ errors in y recorded in percentages for p_1 p_2, and p_3. a) Errors from using the original basis b) Errors from using the augmented basis

from 6.5% to 3.2%.

Finally, we consider p_7, p_8, and p_9 (see Fig. 5.12). Below the bifurcation, the original basis produces errors ranging from 8.1% to 53% as opposed to the augmented basis whose relative errors are around .1%. Right above the bifurcation at $m = m_i + 1$, the original basis has errors around 60% while the errors for the augmented ROM hover around 9%. For the biggest two m values, the augmented system's largest errors are just 50% of the original system's.

The results indicate that the augmented basis constructed using Method 1 is able to both correctly track the bifurcation, and produce very accurate answers over a wide range of parameters. This is especially remarkable considering that perturbing the three parameters dramatically changes both the value of m where the bifurcation occurs, and the shape, height, and speed of the waves. By combining just two solution types, we dramatically increased the ability of the POD reduced order model to model perturbations in its parameters.

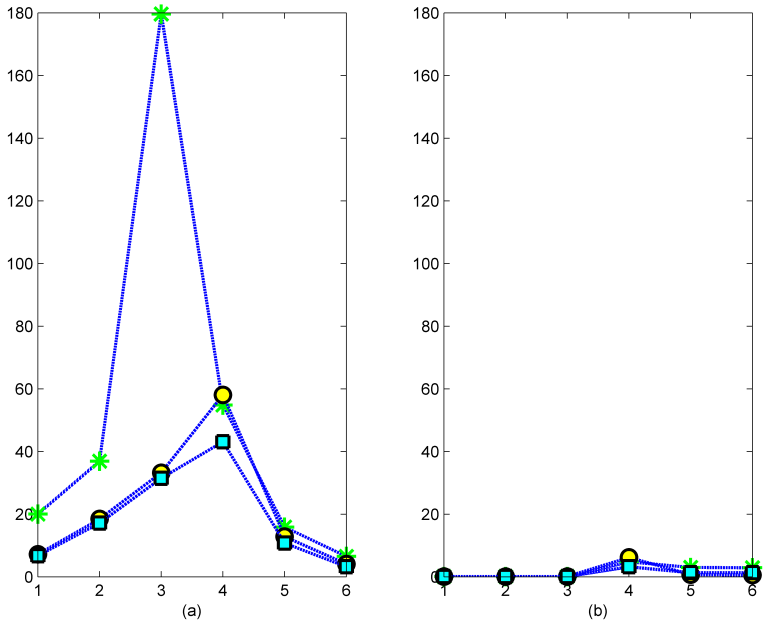

Figure 5.12: The average, relative $\|.\|_2$ errors in y recorded in percentages for $p_4 - p_6$. a) Errors from using the original basis b) Errors from using the augmented basis

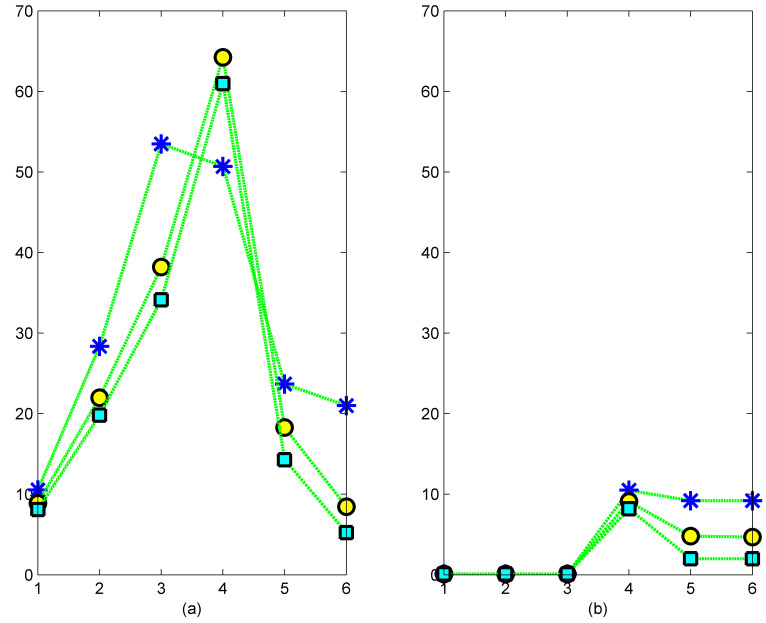

Figure 5.13: The average, relative $\|.\|_2$ errors in y recorded in percentages for $p_7 - p_9$. a) Errors from using the original basis b) Errors from using the augmented basis

5.3.4 1D Reaction-Diffusion using Method 3

Finally, we return to the 1-D reaction-diffusion model 5.4. The PDE is defined on the interval $[0, 2.0]$ and for time, on $[0, 1.0]$. In Section 5.3.2, we found that Method 2, with the approximation that $w_{p_i}(t) \cong y_{p_i}(t) - y(t)$, worked well on this problem. Here, the goal is to use its variation, Method 3 (5.2.4), in order to make a more efficient POD model for the perturbed system. For the base solution, we use the solution computed with a reaction coefficient, $a = 20.0$, and a diffusion coefficient, $b = 0.025$. For the perturbed system, the reaction coefficient is increased to $a = 180.0$ while diffusion is reduced to $b = 0.003$.

As before, the original system and the perturbed system are both solved using method of lines. The PDE's are converted to ODE's using standard centered differencing in space to approximate the spatial derivatives on a grid of 300 points. The resulting ODE's have dimension $n = 300$. 1800 equally-spaced snapshots are collected over the time interval $[.1, 1.0]$ for each equation. 3rd order Runge Kutta is used for the time discretization.

Once the time slices for y have been constructed, Hermite interpolation is used to form an approximation to the base solution, y_b. By using 17 vectors for the interpolant, there is a maximal, relative error of .005 between y and y_b, and .03 between $\frac{dy}{dt}$ and $\frac{dy_b}{dt}$ evaluated over the interval.

In order to make a POD model for the perturbation equation:

$$\frac{dw_p}{dt} = f(w_p + y_b) - \frac{dy_b}{dt}$$

we carry out SVD on the time slices of $w_p = y_p - y_b$ to form the POD basis for w_p. Alternatively, one could just solve the w_p equation. Once the POD basis for w_p is made, it is used to construct the reduced order model

$$\frac{dv_p}{dt} = P^T f(Pv_p + y_b) - P^T \frac{dy_b}{dt} \tag{5.6}$$

where P is the truncated POD basis for w_p so that $w_p \cong Pv_p$.

The POD reduced order model for 5.6 is more efficient than the standard POD model for y_p. In order to get a relative $\|.\|_2$ error at time $t = 1.0$ of 1% for y_p, it is necessary to use a POD basis of size 130. In contrast, 5.6 only requires 95 POD vectors to get a relative $\|.\|_2$

error $< 1\%$. Furthermore, the evaluation of the interpolant at each time step using Hermite interpolation only requires 6 vectors while just 17 are needed for storage. Since the POD basis is much larger than this, the interpolant contributes minimally to the total computational cost.

The size of the POD basis was reduced by nearly 30% using this modified POD method. This suggests that by applying POD to the perturbation equation around a base solution instead of the the perturbed system directly, problems which might otherwise not be approachable due to the size of the POD basis might be manageable using this method instead. The method would be particularly advantageous for systems in which the perturbed system relaxes back to the original after a period of time.

5.3.5 Basis Selection

To form the new bases in the numerical examples above, we performed SVD on the snapshots from the perturbed systems either on y_{p_i} directly (using Method 1 introduced in 5.2.2) or on the system associated with the perturbation about a base solution w_{p_i} (using Method 2, see 5.2.3, and Method 3, see 5.2.4). The singular vectors corresponding to the largest singular values were then picked and combined with the vectors from the original basis, and re-orthonormalized. By doing this with Method 1, the additional dominant singular vectors may have a large inner product with the original basis and thus lead to an unnecessarily large basis set. The same problem occurs with Method 2 when $w_{p_i}(t)$ is estimated with $y_{p_i}(t) - y(t)$

This brings up the question, why not simplify the process and use Method 0 5.2.1 by combining together the snapshots obtained from running the original and the perturbed systems in one large data set before doing SVD such as in Shlizerman et al. [SDW12]? A basis could then be selected from the vectors which correspond to the largest singular values, thus avoiding any loss of information. There is no guarantee, however, that this alternative basis will give the desired accuracy on the original system, or instead, that it will capture the important behavior in the perturbed systems.

To demonstrate the problems that can arise with Method 0, we compare alternate ways of selecting the basis in the examples below. Method 0 is compared to Method 1 on the FHN

system, and to Method 2 on the 1-D reaction-diffusion equation.

5.3.5.1 FitzHugh-Nagumo

We first look at the FitzHugh-Nagumo system from Section 5.3.1. Recall that m is allowed to vary between $[12000, 15000]$ and a bifurcation occurs for m in this interval. We use the original and $B1$ bases both with dimension 134 constructed in 5.3.1. The original basis is constructed for $m = 13500$ (which generates a traveling wave solution) with rank 134. Both u and y have relative errors of $< 1\%$ sampled over the domain. $B1$ is made by combining the modes with the largest singular values from two different POD bases using Method 1. Specifically, the first 130 modes corresponding to the traveling wave solution where $m = 13500$ (which produced relative errors of 1.1% and 1% for u and y respectively sampled over the domain) are combined with the first four modes derived from the decaying solution where $m = 12000$. The vectors are then orthonormalized to form the augmented basis, $B1$.

To evaluate Method 0, we create another basis $B0$, obtained by combining all the snapshots together before doing SVD. In this case, we find that the traveling wave solutions are favored over the decaying modes. The ROM which uses $B0$, like the original system, is unable to capture the bifurcation. Even when it models the correct qualitative behavior for $m \in [12500, 12700]$, the relative errors of u and y are quite large compared to the ROM, which uses $B1$. As can be seen in Table 5.8, the average relative errors in u sampled over the interval are at least order of magnitude 10 worse than those generated using $B1$ for $m < 12625$, and order of magnitude 7 worse for $m > 12666$ (the bifurcation occurs between $m = 12665$ and 12666). For y, they are more than order of magnitude 10 worse for $m \in [12500, 12700]$. Carrying out SVD individually on collections of snapshots taken from perturbed systems before combining subsets of the singular vectors in a basis (and re-orthonormalizing) performs significantly better than combining all the runs together prior to doing SVD.

Table 5.8: Listed below are the average relative $\|.\|_2$ errors in u produced by using $B1$, $B0$, and the original POD bases all with rank 134. In $B1$, the four vectors corresponding to the smallest singular values in the original POD basis have been replaced by singular vectors derived from the system at $m = 12000$. $B0$ is formed by doing SVD after combining the runs taken when $m = 13500$ and $m = 12000$. The errors are listed according to the size of $m \in [12500, 12700]$.

Average Relative Errors of u, rank=134						
m	12500	12600	12625	12666	12680	12700
Original Basis	0.055	0.154	0.296	0.826	0.296	0.232
$B0$	0.012	0.029	0.048	0.605	0.077	0.019
$B1$	0.001	0.003	0.005	0.240	0.013	0.007

5.3.5.2 1-D Reaction-Diffusion

Next, we consider the 1-D reaction-diffusion model. Recall that the augmented basis (constructed in Section 5.3.2) was created in order to better model perturbations in the reaction coefficient, b, as it varied in $[.005, .045]$. Method 2 was used in this case. Since the POD basis from the original system (where $b = .025$, with rank 33) modeled the solution poorly for $b_{min} = .005$, the singular vectors for augmentation were selected from the system with the least diffusion. Specifically, the 8 largest singular vectors from the SVD of the time slices of $y - y_{b_{min}}$ (recall that y was similar enough to $y_{b_{min}}$ that w_{min} could be estimated with $y - y_{b_{min}}$) were combined with the 33 elements in the original basis. The corresponding energy sumed up to 99% of the total energy of w_{min}. This guaranteed a relative error $\leq .01$ for the original system. The augmented basis is denoted by $B2$ in Table 5.9.

The results of using Method 2 are compared to the ROM generated by Method 0 in Table 5.9 for values of the diffusion parameter $b \in [.005, .025]$. The basis constructed using Method 0 is notated by $B0$ in Table 5.9. The results for $b \in (.025, .045]$ are not included because the relative errors in the solutions using all of the basis types are less than 1%. The two bases both have ranks = 41. Although the perturbed system had not diverged significantly from the

original system in this case, we would expect that the quality of the ROM produced to not differ significantly between the different bases. However, $B0$ still performed slightly worse for all the values of the diffusion coefficient tested except where $b_{min} = .005$ thereby emphasizing the least diffuse system. This suggests that combining all the snapshots together before doing SVD should only be done if there is no computational benefit to Method 2 (ie the construction of the interpolant, y_b, is prohibitively expensive) and there is little difference between the perturbed and original system.

Table 5.9: Listed below are the relative $\|.\|_2$ errors produced by using $B0$ and $B2$ both with equal rank. The relative errors are listed for diffusion coefficients equally spaced between .005 and .025.

a) Bases have rank 41. 8 singular vectors were used for augmentation											
b	.005	.007	.009	.011	.013	.015	.017	.019	.021	.023	.025
$B0$	0.0097	0.0255	0.0297	0.0245	0.0186	0.0139	0.0103	0.0077	0.0058	0.0044	0.0034
$B2$	0.0574	0.0252	0.0167	0.0116	0.0085	0.0061	0.0043	0.0032	0.0023	0.0018	0.0014

5.4 Conclusion

We introduced three POD augmentation methods which explore different ways of incorporating components of perturbed systems POD bases in order to make ROM's that are more robust to perturbations. Method 1 involves computing the SVD of each system and then combining the results. By selecting a sufficient number from each set of singular vectors, one creates a basis that will work equally well for the specific perturbed and unperturbed solutions. This is a very simple method that has the advantage that the computations and construction of the singular vectors for each system can be accomplished independently, and thus efficiently use multi-processor computational resources.

The second method (Method 2) is a variant of Method 1 that allows one to avoid the consequences of basis set overlap. The dominant singular vectors for the difference between the perturbed solution and the projection of that solution on the base solution basis set are used

to augment the base solution basis. In addition to creating singular vectors that have reduced redundancy, the singular values computed can provide useful information about the magnitude of the solution changes with respect to changes in solution parameters.

In the above examples, we were able to significantly improve the ability of a POD basis to handle a large range perturbations by running only a few simulations in the full dimension. We got substantial improvements in the errors generated by the ROM when compared to the standard POD basis. In the case of the FitzHugh-Nagumo equation with one parameter varied, the augmented basis constructed using Method 1 dramatically improved the ROM as the stimulus ranged in size, particularly close to the bifurcation where the original POD basis was unable to capture the bifurcation. When 3 parameters were varied, the results were even better. By adding in only 4 basis elements, the augmented basis was able to not only correctly track the bifurcation, but also to produce amazingly accurate answers over a wide range of parameters. This is a particularly exciting result because it is often difficult to find bifurcations via simulation [DSK10]. This is in contrast to the original basis which, even when it produced the correct qualitative behavior, performed very poorly around the bifurcation. We also considered the 1-D reaction-diffusion equation with the diffusion coefficient varied. Here, the augmented basis designed using Method 2 produced errors roughly one third to one half the size as those from using the basis from the original system (of equivalent rank) on the systems where the original basis functioned poorly. When the augmented basis was applied to systems close to the base solution (where the original basis worked well), it still had small errors.

In addition, a third modified POD method was presented which improves the computational efficiency of the ROM to model perturbed systems. This can be expected to work as well as Method 2 at determining the vectors for augmented basis sets. By applying POD to the perturbation equation around a base solution interpolant instead of to the perturbed system directly, we were able to reduce the cost of using POD. In the example considered, the method enabled us to maintain the same level of accuracy as with normal POD while reducing the POD basis size by roughly 30%. This method has potential for large scale problems which might otherwise not be approachable due to the size of the POD basis.

Additionally, we showed that the basis made from combining the snapshots together from

selected systems before performing SVD (Method 0) performs significantly worse than the augmented bases. In this case, there is no guarantee that the largest singular vectors form the combined set will approximate solutions of both the perturbed and unperturbed systems equally well. When we compared Method 0 to Method 1 on the FitzHugh-Nagumo equation, Method 0 was unable to locate the bifurcation remotely close to its true value. Even when the qualitative behavior was correct, the relative errors were order of magnitude 7 worse or more than the augmented basis for for $m \in [12500, 12700]$. For the 1-D reaction-diffusion system, the perturbed solutions were sufficiently similar to the base solution that we only saw a slight improvement in Method 2 over Method 0. As shown in Section, Method 3 can be used instead of Method 2 on this problem and its computation is much less expensive than Method 0's.

CHAPTER 6

Timestep Stability Restrictions for POD Models

In the process of carrying out the numerical experiments for Chapters 2-5, we observed that the timestep required for evolving a POD based reduced order model using an explicit ODE method could be taken to be much larger than the timestep required for the same method applied to the full system. In this section, we report on computational experiments that further substantiate the conclusion that POD based ROM's will generally have reduced stability timestep restrictions. The consequence of this observation is that the ability to use a larger timestep will typically lead to greater computational efficiency, and thus help to offset the additional cost per timestep that evolving a POD based ROM requires for problems with nonlinear terms that cannot be precomputed. The general experimental design is presented in Section 6.1. This is followed by the numerical details in 6.1.1.

6.1 Reducing POD Computational Cost by Larger Timestep Selection

One limitation of POD reduced order models is their increased cost per timestep to advance the solution of problems with non-linear terms. The representation of the evolution operator in the ROM basis, $\tilde{V}^T f(\tilde{V}z, t)$ (see 3.3 for more detail) is not, or cannot be precomputed except for operators with a special structure (such as being linear or bilinear). As a result of needing the application of the projection operator, the computing cost is greater than the full order system ([NP08]). There has been a lot of effort to address the complexity issue of the nonlinear terms in POD reduced order models such as in [Ast04], [GFW10], [NP08], [Cha11].

An observation made while carrying out experiments for Chapters 2-5 is that when using

explicit methods for the POD based reduced order models, one could use much larger timesteps than those required for stably computing the solution of the full order system. The fact that larger timesteps can generally be taken makes intuitive sense. Specifically, as suggested by the sensitivity analysis done in Chapter 4, the POD basis automatically filters solution components that are rapidly decaying, e.g. have evolution behavior that requires small timesteps to capture. This suggests that one should be able to use larger timesteps because the components that lead to numerical instability are not present in the evolution operator.

To demonstrate the validity of the observation, we again look at the equations introduced in Chapter 2 (see Section 2.1 for the numerical details). Recall that we consider three different 1-D PDE's: hyperbolic conservation law (shock), reaction-diffusion, and transport with essentially the same solution (2.2, 2.3, and 2.1). This insures that their respective singular value decompositions, and thus, their POD bases, will be nearly identical. We consider the problems with solutions that consist of a single front moving left to right. For each equation, a numerical method and problem parameters are selected such that the solution has a leading edge that we classify as: steep, medium diffuse, and most diffuse. For the hyperbolic conservation law, a diffusion term was added to the system in order to obtain the medium diffuse and most diffuse fronts. With these equations, we explore the effect that modifying the size of the POD basis has on the stability timestep restriction of the ODE method used to evolve the reduced order model.

6.1.1 Numerical Results

For each of the different types of equations and fronts, we record a number of pieces of data which are listed in the Tables 6.1 - 6.3 below. First, the dimension of the reduced POD space necessary to have a relative, $\|.\|_2$ error of $< 1\%$ (rounded to the nearest percentage point) when compared to the original system. Second, the number of time steps necessary in the reduced, POD system to get a relative, $\|.\|_2$ error of $< 5\%$ (rounded to the nearest percentage point) from the original system. Lastly, the number of time steps necessary in the full dimensional system to get a relative, $\|.\|_2$ error of $< 5\%$ (rounded to the nearest percentage point) from the

original system. Note that for some of the equations considered, the solution became unstable

Table 6.1: Listed below for the steepest fronts are the dimensions of the reduced POD spaces necessary to have a relative, $\|.\|_2$ error of $< 1\%$, the number of time steps necessary in the reduced, POD systems to get a relative, $\|.\|_2$ error of $< 5\%$, and the number of time steps necessary in the full dimensional systems to get a relative, $\|.\|_2$ error of $< 5\%$.

Steepest fronts			
Equation Type	Reaction-Diffusion	Shock	Transport
Dimension of POD System	147	112	105
POD Time Steps	384	253	224
Time Steps Full Dimension	392	318	512

for a relative error $< 5\%$. In this case, the smallest number of time steps necessary to preserve stability are recorded instead. For the 1-D advection equation and the hyperbolic conservation law, the relative errors are calculated at the final time step because this corresponded to the largest error. For the reaction-diffusion equation, the error is computed as the average of four relative $\|.\|_2$ errors sampled at times $\{0.25, 0.50, 0.75, 1.0\}$.

Table 6.2: Listed below for the medium diffuse fronts are the dimensions of the reduced POD spaces necessary to have a relative, $\|.\|_2$ error of $< 1\%$, the number of time steps necessary in the reduced, POD systems to get a relative, $\|.\|_2$ error of $< 5\%$, and the number of time steps necessary in the full dimensional systems to get a relative, $\|.\|_2$ error of $< 5\%$.

Medium Diffuse Fronts			
Equation Type	Reaction-Diffusion	Shock	Transport
Dimension of POD System	48	35	46
POD Time Steps	240	124	75
Time Steps Full Dimension	436	676	180

Table 6.3: Listed below for the steepest fronts are the dimensiosn of the reduced POD spaces necessary to have a relative, $\|.\|_2$ error of $< 1\%$, the number of time steps necessary in the reduced, POD systems to get a relative, $\|.\|_2$ error $< 5\%$, and the number of time steps necessary in the full dimensional systems to get a relative, $\|.\|_2$ error of $< 5\%$.

Most Diffuse Fronts			
Equation Type	Reaction-Diffusion	Shock	Transport
Dimension of POD System	19	20	17
POD Time Steps	252	240	40
Time Steps Full Dimension	928	1080	72

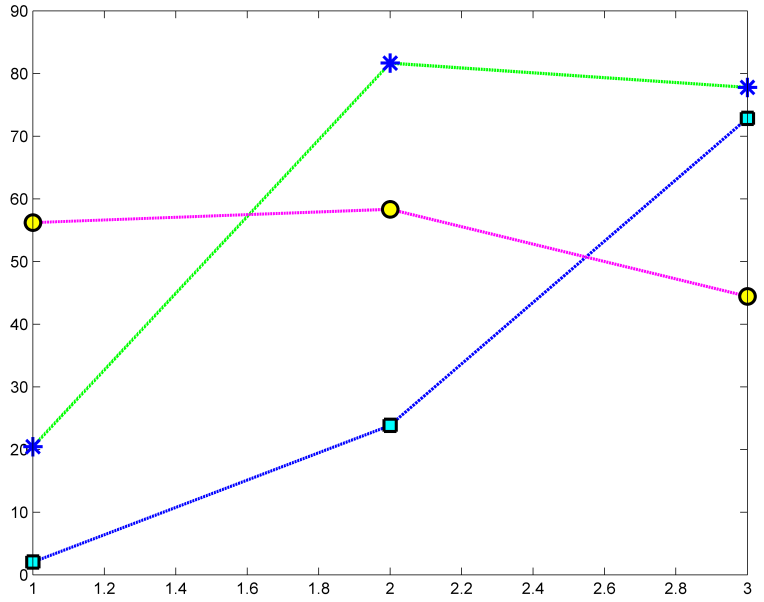

Figure 6.1: The percentage reduction of the number of time steps between the reduced system and the full dimensional system recorded in percentages. The following markers correspond to the different equations: stars for hyperbolic conservation law, squares for reaction-diffusion, and circles for transport. The markers correspond to the slopes of the fronts and are listed left to right from steepest to most diffuse.

The percentage improvement for the number of time steps necessary to get an error of $< 5\%$ in the reduced order system are shown in Fig 6.1. Specifically, this is calculated as the difference between the number of time steps, in the full and reduced systems, to get an error $< 5\%$ divided by the number of time steps necessary in the full system to get an error $< 5\%$.

There was the least improvement with the steepest fronts for reaction-diffusion which only had an improvement of 2% and shock which improved by 20%. In contrast, transport improved by 56%. For the medium and most diffuses cases, transport improved by 58% and 45% respectively. For the other two types of fronts, the hyperbolic conservation law had improvements of about 80% while reaction-diffusion improved by 24% and 73%.

6.2 Discussion and Conclusion

For the transport equation, a linear equation which is not stiff, the dominant factor determining the timestep is the accuracy, so that as the profile becomes more diffuse, and hence smooth in both space and time, the size of the timestep required to obtain a specified accuracy will be reduced. The number of timesteps required by the POD model for the accurate computation of the transport equation was about 50% that of the full dimensional solution for all three types of fronts. In this situation, accuracy was the determining factor, and as the solution became smoother in both space and time, a larger timestep could be taken. The improvement that POD provides over the full system is the fact that the implicit filtering associated with the POD method provides a smoother solution.

For both the reaction-diffusion and hyperbolic conservation law systems, the growth in the number of timesteps required as the diffusion parameter in the problem is increased is consistent with a local linear analysis, e.g. the number of timesteps should grow as the diffusion parameter is increased. The beneficial effects of the implicit filtering of high frequency components of the solution associated with the POD method are clearly seen in the case of the reaction-diffusion and hyperbolic conservation law equations for the medium diffuse and most diffuse cases. For example, though the diffusion parameter was 25 times increased for the reaction-diffusion problem between the medium and most diffuse cases, there was essentially

no change in the number of timesteps required in the reduced order system while in full dimension, the number of timesteps increased by about 50%. The hyperbolic conservation law had improvements of about 80% over the full dimensional system for both the medium and most diffuse cases. For the steepest reaction-diffusion case, the near equivalence of the number of timesteps for the POD system indicates that accuracy is the primary factor in determining the timestep. Similarly, for the steepest fronts generated by the hyperbolic conservation law, there was only a 20% reduction in the number of timesteps for the ROM as compared to the full dimensional model. The timestep size required to resolve the dynamics associated with the POD basis in the steepest case for both these systems was close to the same size as the full dimensional system.

The results demonstrate that for problems that are stiff, the implicit filtering associated with the POD methods can lead to a system that is not stiff, and hence enable one to use explict timestepping methods with large timesteps to obtain accurate solutions. Also, for the transport equation (which is not stiff), applying POD resulted in smoother solutions which enabled the use of larger timesteps. For systems without the complexity issue of specific nonlinear terms in POD reduced order models, the ability to use larger timesteps in the reduced order model makes the use of POD more attractive. In the case of systems with nonlinear terms where the cost per timestep of a POD method is greater than that of the full system, this extra cost can be somewhat offset by the use of simpler ODE methods and larger timesteps.

CHAPTER 7

Conclusion

We first presented results in Chapter 2 comparing the outcomes of using essentially the same POD basis for different types of PDE's that demonstrate the accuracy of the POD model depends not only on the ability of the POD basis to represent the solution as set by the tolerance used in the POD selection procedure, but also on the type of equation. Additionally, the results showed that the way in which inaccuracies arise in POD reduced order models are dependent on the type of equation being modeled. Reaction-diffusion's poor performance with a POD model when compared to hyperbolic conservation law and transport equations emphasizes the importance of equation specific corrections for POD reduced order models.

In the absence of theoretical a-priori estimates of the error associated with a POD model solution, it is very useful to have a computational procedure that indicates when the solution is likely to have a significant error. In 3, we introduced an error indicator designed to detect when a POD basis stops performing well on a perturbed system as the numerical solution evolves. In all the numerical examples considered, the behavior of error indicator mimicked the behavior of the true relative error.

In Chapter 4, we investigated how to carry out sensitivity analysis efficiently by applying it to POD reduced order models on reaction-diffusion type equations while in Chapter 5, we introduced a variety of strategies for basis augmentation. We investigated both how best to pick components of perturbed systems in order to improve the POD basis, and how to improve the computational efficiency of the calculations. In all the systems tested, we were able to significantly improve the ability of a POD basis to handle a large range perturbations using augmented bases that were obtained from only a few simulations in the full dimension of the perturbed systems. We got substantial reduction in the errors generated by the ROM when

compared to the standard POD basis. In particular, we formed a ROM using an augmented basis on the FitzHugh-Nagumo system which was able to closely indicate the location of bifurcations while providing a significant reduction in the dimension when compared to the full order solution. This is an exciting result because it is often difficult to find bifurcations via simulation [DSK10].

Inorder to consider using a POD method on perturbed systems, a few criteria should be met. Clearly, if solution trajectories deviate from the paths of the solutions used to create the augmented basis, the POD basis will no longer be relevant. This highlights the importance of simultaneously carrying out a computation of the error indicator of the type introduced in Chapter 3. Additionally, if the singular values associated with the snapshots of the perturbed system do not decay sufficiently rapidly, and the majority of singular vectors are not spanned by the original POD basis, then a large basis will be required which diminishes the attractiveness. Notwithstanding these general concerns about applicability, the numerical results presented in Chapter 5 demonstrate that using augmented basis constructed with these methods can substantially improve the accuracy of POD reduced order models on perturbed systems.

Finally, we considered how to improve the computational efficiency of POD. We proposed that the implicit filtering associated with POD enables the use of larger timesteps in the POD model. Our computational experiments supported this idea that POD based ROM's will generally have reduced stability timestep restrictions. The consequence of this observation is that the ability to use a larger timestep will typically lead to greater computational efficiency, and thus help to offset the additional cost per timestep that evolving a POD based ROM requires for problems with nonlinear terms that cannot be precomputed.

In the future, we plan to focus on two main areas of research: basis augmentation and error detection. We would like to determine a criteria for choosing the type of basis augmentation method best suited for a given PDE. We would also like to work on developing a more efficient, adaptive framework for the application of the augmentated bases. Additionally, we plan to apply Method 3 (5.2.4) to a broader class of PDE's. For the error indicator, we intend to theoretically and computationally investigate its general applicability. Furthermore, we would like to investigate additional error estimates that can be done a-priori on POD type systems.

Lastly, we intend to work on the use of a combination of basis augmentation and error estimation in order to improve the accuracy and efficiency of POD on perturbed systems.

References

[Ast04] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach*. PhD thesis, Eindhoven University of Technology, 2004.

[Atk88] K. E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons, Inc., second edition, 1988.

[BFC10] T. Brenner, R. Fontenot, P. Cizmas, T. O'Brien, and R. Breault. "Augmented proper orthogonal decomposition for problems with moving discontinuities." *Powder Technology*, **203**(1):78–85, 2010.

[BJS02] M. S. Bartlett, J. R. Javier, and T. J. Sejnowski. "Face recognition by independent component analysis." *IEEE Transactions on Neural Networks*, **13**(6):1450–1464, 2002.

[BJW00] H. Banks, M. Joyner, B. Wincheski, and W. Winfree. "Nondestructive evaluation using a reduced-order computational methodology." *Inverse Problems*, **16**(4):929–945, 2000.

[Cac1a] D. G. Cacuci. "Sensitivity theory for nonlinear systems. I. Nonlinear functional analysis approach." *J. Math. Phys.*, **22**(12):2794–2802, 1981a.

[Cac1b] D. G. Cacuci. "Sensitivity theory for nonlinear systems. II. Extension to additional classes of responses." *J. Math. Phys.*, **22**(12):2803–2812, 1981b.

[Cha05] J. Chan. "The Physics of Tropical Cyclone Motion." *Annual Review of Fluid Mechanics*, **37**:99–128, 2005.

[Cha11] S. Chaturantabut. *Nonlinear Model Reduction via Discrete Empirical Interpolation*. PhD thesis, Rice University, 2011.

[CW87] J. Chan and R. Williams. "Analytical and Numerical Studies of the Beta-Effect in Tropical Cyclone Mottion PartI: Zero Mean Flow." *Journal of the Atmospheric Sciences*, **44**(9):1257–1265, 1987.

[DSK10] E. Ding, E. Shlizerman, and J. Kutz. "Modeling multipulsing transition in ring cavity lasers with proper orthogonal decomposition." *Phys. Rev. A*, **82**, 2010.

[ECS97] R. Everson, P. Cornillon, L. Sirovich, and A. Webber. "An Empirical Eigenfunction Analysis of Sea Surface Temperatures in the Western North Atlantic." *J. Phys. Oceanogr.*, **27**(3):468–479, 1997.

[Err97] R. Errico. "What is an adjoint model?" *Bull. Am. Meteor. Soc.*, **78**:25772591, 1997.

[FE89] M. Fiorino and R. Elsberry. "Some aspects of vortex structure related to tropical cyclone motion." *Journal of the Atmospheric Sciences*, **46**:975–990, 1989.

[Fed96] R. P. Fedkiw. *A Survey of Chemically Reacting, Compressible Flows*. PhD thesis, UCLA, 1996.

[GFW10] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. "Non-linear model reduction for uncertainty quantification in large-scale inverse problems." *International Journal for Numerical Methods in Engineering*, **81**(12):1581–1608, 2010.

[GL96] B. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Md, third edition, 1996.

[HJB98] P. Holmes, J.Lumley, and G. Berkooz. *Turbulence, coherent structures, dynamical systems*. Cambridge University Press, 1998.

[HL12] P. Heimbach and M. Losch. "Adjoint sensitivities of sub-ice shelf melt rates to ocean circulation under Pine Island Ice Shelf, West Antarctica." *Annals of Glaciology*, **53**(60):59–69, 2012.

[HPS05] C. Homescu, L. R. Petzold, and R. Serban. "Error estimation for reduced-order models of dynamical systems." *SIAM J. Numerical Analysis*, **43**:1693–1714, 2005.

[KS98] J.P. Keener and J. Sneyd. *Mathematical Physiology*. Springer-Verlag, New York, 1998.

[KV99] K. Kunisch and S. Volkwein. "Control of the Burgers equation by a reduced-order approach proper orthogonal decomposition." *J. Optim. Theory Appl.*, **102**(2):345–371, 1999.

[KV01] K. Kunisch and S. Volkwein. "Galerkin proper orthogonal decomposition methods for parabolic problems." *Numer. Math*, **90**:117–148, 2001.

[KV02] K. Kunisch and S. Volkwein. "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics." *Siam Journal On Numerical Analysis*, **40**(2):492–515, 2002.

[LH09] Y. Lee and Z. Hung. "Alternative Approaches to Estimating the Linear Propagator and Finite-Time Growth Rates from Data." *Terrestrial, Atmospheric and Oceanic Sciences*, **20**(2):365–375, 2009.

[LJ03] P. LeGresley and J.Alonso. "Dynamic domain decomposition and error correction for reduced order models." In *Proc. AIAA 41st Aerospace Sciences Meeting and Exhibit*, volume 250, 2003.

[LJ04] P. LeGresley and J.Alonso. "Improving the performance of design decomposition methods with POD." In *Proc. AIAA/ISSMO 10th Multidisciplinary Analysis and Optimization Conference*, volume 4465, 2004.

[LKO01] D. Lucia, P. King, M. Oxley, and P. Beran. "Reduced order modeling for a one-dimensional nozzle flow with moving shocks." In *Proc. AIAA 15th Computational Fluid Dynamics Conference*, volume 2602, 2001.

[LMM09] F. Leite, R. Montagne, C. Montagne, and L. Lucena. "KarhunenLove spectral analysis in multiresolution decomposition." *Comput. Geosciences*, **13**(2):165–170, 2009.

[Lor65] E. Lorenz. "A study of the predictability of a 28 variable atmospheric model." *Tellus*, **17**:321–333, 1965.

[Lor85] E. Lorenz. *The growth of errors in prediction.* In M. Ghil, (Ed.), Turbulence and Predictability in Geophysical Fluid Dynamics and Climate Dynamics, Italian Physical Society, North-Holland, 1985.

[LP08] M. Leutbecher and T. Palmer. "Ensemble forecasting." *J. Comput. Phys.*, **227**:3515–3539, 2008.

[MM03] M. Meyer and H. Matthies. "Efficient model reduction in non-linear dynamics using the Karhunen-Loeve expansion and dual-weighted-residual methods." *Comput. Mech.*, **31**:179–191, 2003.

[NP08] N. C. Nguyen and J. Peraire. "An efficient reduced-order modeling approach for non-linear parametrized partial differential equations." *International Journal for Numerical Methods in Engineering*, **76**(1):27–55, 2008.

[Pea01] K. Pearson. "On lines and planes of closest to points in space." *Philosophical Magazine*, **2**, 1901.

[Pin08] R. Pinnau. "Model reduction via proper orthogonal decomposition, in Model Order Reduction: Theory, Research Aspects and Applications." *In W. Schilders and H. Van der Vorst and J. Rommes (Eds.), Springer Series Mathematics in Industry*, **13**, 2008.

[Rav02] S. Ravindran. "Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition." *SIAM J. Sci. Comput.*, **23**(6):1924–1942, 2002.

[SDW12] E. Shlizerman, E. Ding, M. Williams, and J. Kutz. "The Proper Orthogonal Decomposition for Dimensionality Reduction in Mode-Locked Lasers and Optical Systems." *International Journal of Optics*, **2012**, 2012.

[Sel97] F. Selten. "Baroclinic empirical orthogonal functions as basis functions in an atmospheric model." *J. Atmos. Sci.*, **54**:2100–2114, 1997.

[Sir87] L. Sirovich. "Turbulence and the dynamics of coherent structures." *I–III. Quart. Appl. Math*, **45**(3):561–590, 1987.

[SO88] C. W. Shu and S. Osher. "Efficient implementation of essentially non-oscillatory shock-capturing schemes." *Journal of Computational Physics*, **77**(2):439–471, 1988.

[SPH07] R. Serban, L. R. Petzold, and C. Homescu. "The effect of problem perturbations on nonlinear dynamical systems and their reduced order models." *SIAM Review*, **49**:277–299, 2007.

[SS75] P. Swarztrauber and R. Sweet. "Efficient fortran subprograms for the solution of elliptic equations." *ncar tn/ia-109*, p. 138, 1975.

[UCS85] S. Utku, J. Clemente, and M Salama. "Errors in reduction methods." *Comput. Struct.*, **21**:1153–1157, 1985.

[Vol08] S. Volkwein. "Model Reduction using Proper Orthogonal Decomposition.", 2008.