AN ABSTRACT OF THE THESIS OF

Keith L. Pearen for the degree of Master of Science in Mechanical Engineering presented on September 26, 2005.

Title: Development of a Reduced-Order Dynamic Model of a Jet-in-Crossflow from Experimental Data.

Abstract Approved:

Redacted for privacy

Belinda A. Batten

A reduced-order dynamic model is derived that captures the primary dynamics of a jet-in-crossflow. This is significant progress toward the end goal of reduced order model based closed-loop flow control. The reduced order model is a set of differential equations obtained through the use of snapshot based Proper Orthogonal Decomposition (POD) of experimental Particle Image Velocimetry (PIV) data in conjunction with a Galerkin Projection of the Navier-Stokes equations onto the POD basis functions. Such models are ultimately useful in optimization calculations, control derivation and sensitivity analysis, when the size of a model from computational fluid dynamics is too large to be tractable. The goal of this thesis is to derive a reduced-order model and to investigate, via simulation, the type of dynamic behaviors exhibited with a variety of initial conditions. The results obtained are promising and show that the model is capable of reproducing the dynamics of the experimental system in an open-loop computer simulation.

Development of a Reduced-Order Dynamic Model of a Jet-in-Crossflow from Experimental Data

by
Keith L. Pearen

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 26, 2005
Commencement June 2006

Master of Science thesis of Keith L. Pearen
presented on September 26, 2005.

APPROVED:

Redacted for privacy

Major Professor, representing Mechanical Engineering

Redacted for privacy

Head of the Department of Mechanical Engineering

Redacted for privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of the Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Keith L. Pearen, Author

ACKNOWLEDGEMENTS

I would first and foremost like to recognize the amazing contribution made by Bertrand Dano and Jim Liburdy. They were extremely generous to let me use their experimental data as a foundation of my work. Without their support this would not have been possible. Second, I would like to thank Belinda Batten for always making time for me in her extremely busy schedule. Those sit down tutoring sessions got me up to speed on everything from approximating functions with finite sums to the finer points of discretizing PDEs; they were enjoyable, informative and extremely necessary. Thanks also to my other three committee members, Mark Costello, Timothy Kennedy and Frederick Kamke, for taking time out of your busy schedules to serve on my committee. Next, I would like to thank Melanie Love for just being Melanie. And finally, I would like to thank all my other friends and family who have supported me throughout my education both formal and informal.

TABLE OF CONTENTS

LIST OF FIGURES

Development of a Reduced-Order Dynamic Model of a Jet-in-Crossflow from Experimental Data

By

Keith L. Pearen

CHAPTER 1

INTRODUCTION

1.1 Overview

The creation of reduced-order models for systems described by partial differential equations is key to design of real-time control systems. There has been much recent literature on this topic, with many different approaches for deriving models of reduced-order. The methodology proposed here uses a proper orthogonal decomposition (POD) of experimental data and subsequent Galerkin projection to derive a dynamic model of a jet-in-crossflow. The reduced-order models can then be used for a variety of purposes: calculation of control laws, optimization problems, and sensitivity analysis.

The conventional approach to discretization of the Navier-Stokes equations and other nonlinear partial differential equations (PDEs) is by use of a finite-difference, finite-element, or spectral method involving the use of basis functions. A Fourier basis is generally used in spectral methods just as piecewise polynomials are used with finite-element method. These basis functions are mathematically convenient but have very little connection with either the underlying physics of a specific case or the corresponding partial differential equations. In contrast, POD involves the use of basis functions generated from numerical solutions of the partial differential equations or from experimental data. In the case of this study, the basis functions are generated directly from experimental data. More specifically, POD involves the extraction of an optimal set of basis functions from a computational or experimental data set, by use of an eigenvalue analysis or singular value decomposition. Then by means of Galerkin projection, an approximation is calculated as a combination of basis functions from the optimal set. This solution is the desired reduced-order model solution.

An infinite number of basis functions could, in theory, yield a "perfect" model while a relative few basis functions might yield a reasonably good model. Perfect is a misnomer in the sense that the model can only be as good as the data from which the POD basis vectors are generated. It is important to have a data set that captures the complete dynamics of the system, including actuator affects of the control, if the eventual intent of having a reduced order model is for control design (Atwell and King 1999, Prabhu *et al.* 2001). The control actuator in the case of this research will presumably be pulsing of the jet when future research considers developing a controller for the model derived here.

The remainder of this thesis will be an in-depth look at the process necessary to develop a reduced-order dynamic model of a jet-in-crossflow from experimental data using POD and Galerkin projection. While the methods used here are applied to a very specific problem they are widely applicable to many instances involving the discretization of PDEs where experimental data is available. The remainder of the first chapter talks about the reasons why reduced-order modeling is necessary and discusses the history of reduced-order modeling specifically related to fluid flow control and jet-in-crossflow research. The second chapter outlines the general concepts of PIV, POD and Galerkin projection before becoming more specific when these concepts are applied to the problem studied in this thesis. The third chapter discusses results of the numerical methods used and analysis of those results. Finally, the fourth chapter draws some conclusions about the method used here discussing both the strong points and limitations.

1.2 Motivation

The need for reduced-order modeling arises because of the inherent magnitude and complexity of the computations for solving PDEs such as the full Navier-Stokes equations. The time and cost of computation can be reduced through the use of reduced-order models. Although POD based models have often been used for analysis of system dynamics and simulation, it is of interest to derive suitable reduced-order models that approximate the essential dynamics of the full order system well enough for purposes such as control and optimization. The POD-based methodology presented in Chapter 2 provides a means to create a state space form of the dynamic equations.

Closed-loop control of highly non-linear systems has been another topic of extensive research in recent years.* So far there have been many successes in enhancing the performance of complex systems and some attempts have been made to use it in flow control (*e.g.*, Caraballo and Samimy 2005, Bewley and Liu 1998, Ravindran 1999). Open-loop flow control has proven particularly useful in the design and development phase of aerodynamic bodies (*e.g.*, Brooks and Powers 2002) but lacks the flexibility and robustness needed to control an actuator in a dynamic flight environment with unmodeled dynamics and disturbances. Closed-loop flow control, on the other hand, appears to be a good candidate technique for successful flow

* While development of a control is not within the scope of this thesis, much of the previous relevant work performed on this topic has been in the context of fluid control. Thus, much of the research referred to subsequently is control research, but it all includes some component of model development which is pertinent to this thesis.

management in many applications (Bewley 2001). However, the tools of classical control system theory are not directly applicable to the fluid systems governed by the non-linear Navier-Stokes equations which pose formidable modeling challenges due to their infinite dimensionality and their complexity. Therefore, in order to design and successfully implement a closed-loop control strategy, it is necessary to obtain simpler models of the flow systems, which capture their important dynamic characteristics as well as the effect of the actuation. The development of such reduced-order models is the motivation behind the research presented in this thesis.

1.3 Turbulent Flow Modeling

During the last decade, many advances have been made in the area of characterizing and modeling turbulent flow. The work has largely been done by two separate communities working independently. POD has been used by the experimental fluid dynamics community to characterize the large scale structures in the flow. Most of these experiments have been conducted in wind tunnels and utilize Particle Image Velocimetry (PIV) to capture velocity data for the entire flow field at many discrete times or "snapshots." As will be made clear in Chapter 2, PIV data is particularly well suited for POD. Some examples of the PIV method as used to characterize turbulent structures include the wake of a circular cylinder (Cohen *et al.* 2003), cavity flow (Rowley *et al.* 2004) and jet-in-crossflow (Bernero and Fielder 2000).

Meanwhile, the controls community primarily utilizes computational fluid dynamics (CFD) code based on finite element, finite-difference or spectral methods to generate "data" for the flow field. Most research then creates a POD basis from these CFD generated snapshots and projects the Navier-Stokes equations in some form onto the POD basis via Galerkin projection. The solution to this problem is a dynamic model in state space form which a controller is built around. In older papers, the controller is often of the open-loop variety. A more recent example of this type of work is given by Brooks and Powers (2002). However, within the past five years many controllers have begun to "close the loop." Closed-loop control has been shown to be quite effective in many of these studies, but often the control input is not computed in real time (*e.g.* Kim 2003). This is possible because the control is applied to a computer simulation which can be slowed to accommodate the increased time of computing the actuator input.

Bewley and Liu (1998) created a closed-loop controller for plane channel flow and were able to modify eigenvalues and wall shear in their CFD environment. Wall aspiration, unsteady blowing and sucking, was used as the control actuator. Their CFD results were very encouraging, but not replicable in a real time experimental setting. Similar research on channel flow with a backward facing step with similar results was conducted by Ravindran (1999).

Narayanan *et al.* (2003) created an open-loop control law for a jet-in-crossflow based on experimental data. The experimental data in this study was not PIV data, but instead they used a single sensor hot film probe in the jet flow to capture flow field velocities at one point and many discrete time instants. These data along with photos of the flow field enabled the researchers to create a model and write an open-loop control that, "was recognized and demonstrated to be effective in organizing unsteadyness and enhancing mixing and entrainment in the flowfield" (Narayanan *et al.* 2003). The experimental setup was further used to validate the control response.

The only research group to test a closed-loop controller in an experimental setting is the Collaborative Center of Control Science at Ohio State University (Caraballo *et al.* 2005). This is also the only paper which uses POD of PIV data to create a spatial basis which the Navier-Stokes equations are projected onto via Galerkin projection. This technique appears to be similar to the one used in this thesis, however details on the exact methodologies used have yet to be published. This work at Ohio State is occurring simultaneously to this research and is applied to the cavity flow problem. This team has also generated POD basis of cavity flow from CFD models (e.g. Caraballo et al. 2003, Caraballo and Samimy 2003b). The results given in Caraballo et al. (2005) are preliminary but, "are very promising and show that control is capable of reducing the cavity flow resonance not only at the Mach 0.3 flow, for which the reduced order model was specifically derived, but also at other flows with some variation in Mach number."

CHAPTER 2

APPROACH

As previously mentioned, experimental data will be used to produce a POD basis which will subsequently be used to create a system of reduced-order differential equations. In this chapter, details of the various methods necessary to this approach are provided.

2.1 Particle Image Velocimetry

Particle image velocimetry (PIV) is a reliable experimental method for determining the velocities of many discrete points simultaneously in a two dimensional flow field and can also be adapted to 3-D. It is based entirely on imaging and is non-disruptive to the flow to the extent that small “seed” particles do not affect the flow. PIV uses a planar laser light sheet which is pulsed twice, and images of fine particles lying in the light sheet are recorded with a camera. The displacement of the particles between the two images is measured and used to determine the velocity of the particles in the flow simply by dividing the displacement of the particles by the time between the two laser pulses. The most common way of measuring displacement is to divide the image plane into small discrete areas and cross correlate the images from the two time exposures. The spatial displacement that produces the maximum cross-correlation statistically approximates the average displacement of the particles in the interrogation cell.

The experimental data used in this study were collected in the low-speed recirculating wind tunnel at Oregon State University by Bertrand Dano. For a complete explanation of the experimental setup see Dano and Liburdy (2004). The PIV data of the jet-in-crossflow originally contained data for full three dimensional flow, however only streamwise, x, and wall-normal, y, components are used in this work. Only the x and y components from the PIV slice right through the centerline of the jet in the x direction are used here, as this is the plane of symmetry for the flow. All results are for a steady jet inclined at 45° in the streamwise direction. Similarly, all experimental data is for a free-stream crossflow velocity of 2 m/s and a jet velocity of 6.8 m/s. For an example of a single flow visualization image and sample jet behavior see Figure 1.

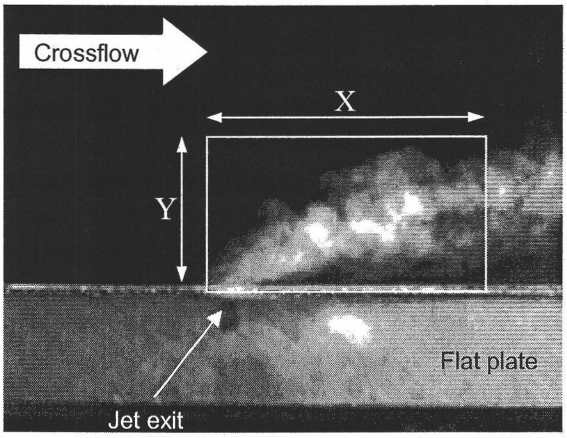


Figure 1: Typical 2D flow visualization of the steady jet in crossflow (Dano and Liburdy 2004).

Figure 2 shows a schematic of the 3D PIV system used by Dano and Liburdy (2004) in their experiment. While the data used in this thesis were obtained using this setup, the third velocity component was not used in this study.

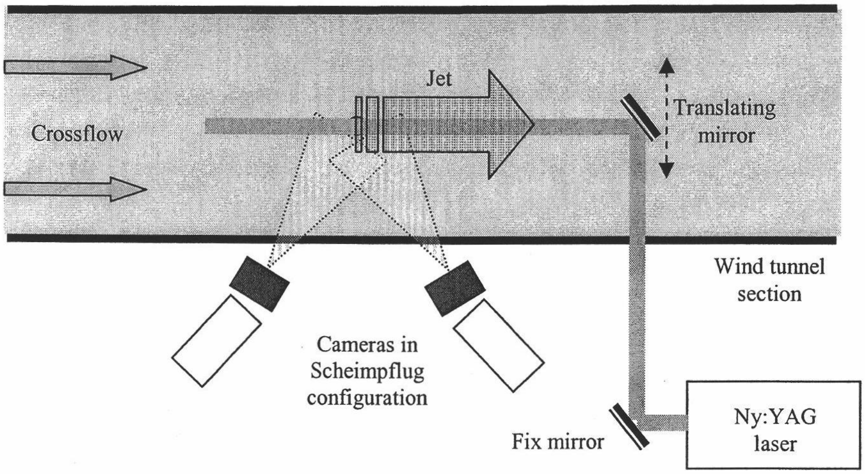


Figure 2: Schematic of the 3D PIV system (Dano and Liburdy 2004).

The PIV method produces measurements of m state variables, in the case of this work these are the fluid velocities u and v at each grid point in the flow field. The flow field for the problem addressed in this thesis is discretized into a 58 by 39 grid creating 2262 discrete points, as illustrated in Figure 3. Each of these grid-points has a two velocities associated with it, u and v, thus $m = 2262 \times 2 = 4524$. These state variables are recorded at N instants in time. There are 200 "time snapshots" available in the PIV data set provided by Dano thus $N = 200$. These N sets of m simultaneous measurements that are arranged in a $N \times m$ matrix D. Each row of D represents a "time snapshot" of the system and each column contains the flow velocity u or v at a point in the flow field. The manner in which these discrete flow areas are numbered does not matter, but consistency is required. For the purposes of this research they are numbered starting with one at the upstream flat plate surface and moving across the flow in the wall normal, y, direction. Once point 39 is reached the next consecutive number (point 40) is back at the wall immediately downstream of point one. See Figure 3 to get a better sense of the numbering scheme.

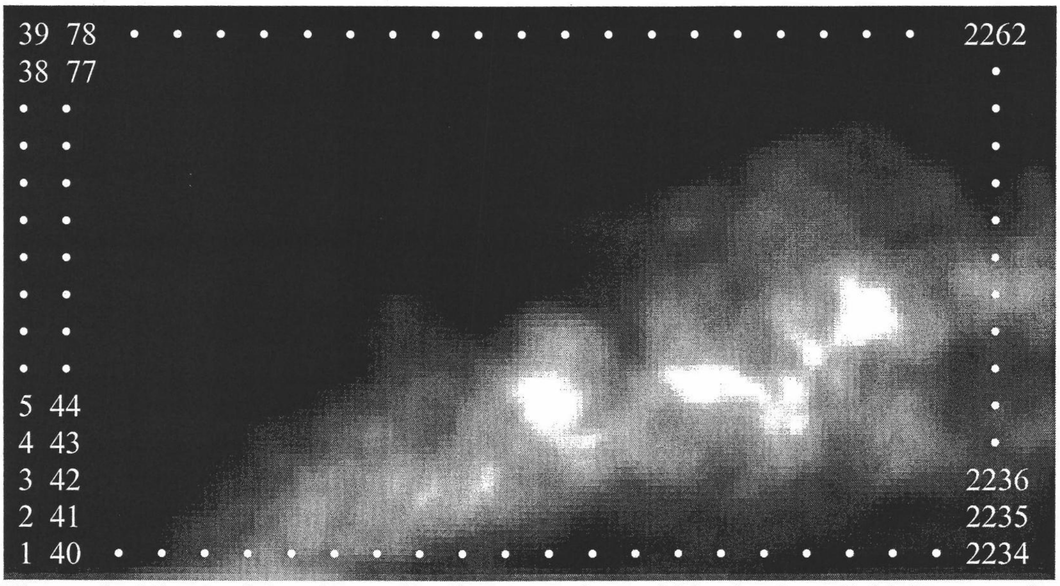


Figure 3: Numbering scheme for discrete points within the flow field (Dano and Liburdy 2004).

2.2 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) provides a powerful method for deriving low order models of complex dynamic systems. The underlying problem is to identify coherent structures in a seemingly random vector field. The development of POD is credited independently to several people, including Kosambi (1946), Karhunen in 1946 and Loéve in 1955, and is also commonly referred to as Principal Component or Hotelling Analysis (Hotelling 1933), the Karhunen- Loéve Decomposition, Empirical Eigenfunction Decomposition and Singular Value Decomposition. POD is commonly used in a variety of applications from our approximation of highly non-linear turbulent flows, to structural vibrations, damage detection in structures, as well as signal analysis, image processing, and file compression.

The POD method was introduced to the fluid dynamics community by Lumley (1967) as an objective way to extract the large-scale structures from a turbulent flow. Details on the fundamentals of the POD method can be found in Berkooz *et al.* (1993) and Holmes *et al.* (1996), while an easy practical guide can be found in Chatterjee (2000). The general idea is to decompose the flow field into a set of orthogonal bases that contains the most dominant characteristics of the flow.

The process is started by approximating some function $u(x,t)$ over a domain of interest as a finite sum

$$u(x,t) \approx \sum_{k=1}^{M} \alpha_k(t)\phi_k(x), \tag{2.1}$$

where $\{\phi_k(x)\}_{k=1}^{M}$ is a set of basis functions. This approximation is expected to become exact as M approaches infinity. The variable x is generally referred to as the spatial variable and t as the temporal variable. The basis functions $\phi_k(x)$ can be chosen as any number of things, including Fourier series, Chebyshev polynomials or piecewise polynomials. Each choice of basis for the function $u(x,t)$ results in a different sequence of time functions $\alpha_k(t)$. Naturally, trigonometric functions will require one set of functions, $\alpha_k(t)$, while polynomials of different types require a different sets. The POD basis arises as follows.

There are two criteria used in selecting the POD basis functions, $\phi_k(x)$. First, it is necessary to have an orthonormal basis. An orthonormal set of basis functions has the property:

$$\int_x \phi_{k_1}(x)\phi_{k_2}(x)dx = \begin{cases} 1 \text{ if } k_1 = k_2 \\ 0 \text{ otherwise} \end{cases} \tag{2.2}$$

Using the definition of orthonormality and substituting in eq.(2.1), it can be shown that each series of coefficient functions $\alpha_k(t)$ depends only on the corresponding $\phi_k(x)$ and not on other ϕ_is where $(i \neq k)$

$$\alpha_k(t) = \int_x u(x,t)\phi_k(x)dx. \tag{2.3}$$

Second, while it is always possible to approximate the original function $u(x,t)$ with an infinite number of basis functions, the most accurate approximation with the least number of basis functions is desired. It is desirable to choose $\phi_1(x)$ so that it provides the best one term approximation of $u(x,t)$. $\phi_1(x)$ and $\phi_2(x)$ should provide the best two term approximation, and so on. The mathematical explanation of this lies in the error

$$E = \int_0^T \left\| u(x,t) - P_r u(x,t) \right\|^2 dt. \tag{2.4}$$

Where P_r is the projection given by

$$P_r = \sum_{k=1}^{r} \phi_k(x)\phi_k(x)^T. \tag{2.5}$$

These ordered, orthogonal vectors ϕ_k are the proper orthogonal nodes of $u(x,t)$ when the error, E, is minimized for a given rank, r (Rowley *et al.* 2004, Berkooz *et al.* 1993). With these special vectors eq. (2.1) becomes the POD of $u(x,t)$.

The experimental data from the jet-in-crossflow covers a finite spatial dimension and is recorded in discrete time. Since the 2-D case is currently under consideration, each point in the flow field has two velocities, u and v, associated with it. These will be inserted in matrix D so that the first $m/2$ columns (1-2262) contain u velocities for each grid point and the second $m/2$ columns (2263-4524) contain v data. More information on the organization of the experimental data can be found in the previous section, *2.1 Particle Image Velocimetry*. It is also common practice to subtract the mean velocity at each point from each column of D. The reason for this is because the mean flow is so dominant that it can conceal any coherent structures in the flow. This practice will receive further treatment in Chapter 3.. The mean velocity of each column will create a vector $\bar{u}(x)$ and eq. (2.1) becomes:

$$u(x,t) \approx \bar{u}(x) + \sum_{k=1}^{m} \alpha_k(t)\phi_k(x). \tag{2.6}$$

The next step in obtaining the POD of the experimental data is to perform a singular value decomposition or SVD of matrix D. The SVD of D less the mean of each column takes the form

$$D - \bar{u} = U\Sigma V^T. \tag{2.7}$$

Where, U is an $N \times N$ orthogonal matrix, V is an $m \times m$ orthogonal matrix and Σ is a $N \times m$ matrix of zeros, except on the main diagonal which contains the singular values of D. The singular values σ_i are all positive real numbers arranged in decreasing order, i.e. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$. The number of nonzero singular values is equal to the rank r of matrix D.

In eq.(2.7), let $U\Sigma = Q$. If Q_k is the kth column of Q and V_k is the kth column of V, then:

$$D=\bar{u}+QV^T=\bar{u}+\sum_{k=1}^{m}Q_kV_k^T. \tag{2.8}$$

This is the discrete form of eq. (2.6). The function $u(x,t)$ is represented here by the matrix D as described above. The function $\alpha_k(t)$ is analogous to the vector Q_k as the function $\phi_k(x)$ is to vector V_k^T.

A key issue in numerical approximation schemes for systems of partial differential equations is that of convergence. For finite element, finite difference and spectral methods, there are theorems that can be applied to ascertain if a numerical approximation method is convergent for a particular problem. The notion of convergence can be expressed in the following sense: that as the grid determined by Δx and Δy is refined (that is, as $\Delta x \to 0$ and $\Delta y \to 0$) and as the timesteps are refined, that solution of the approximating system converges to the solution of the partial differential equation. If one uses a numerical method for which convergence for a particular problem is not known, one runs the risk of obtaining results that bear little relevance to the true PDE solution.

This issue is tricky in the case of POD. When error bounds and convergence are considered, one has the following:

$$\|D-D_R\|_F^2=\sigma_{R+1}^2+\cdots+\sigma_n^2, \tag{2.9}$$

where D is the entire data set and D_R is the data set reconstructed with a reduced number of basis vectors. This formula implies that as the number of basis vectors used in the approximation increases, the approximation converges to the original data set. This has no implications for convergence of the approximate system obtained by

Galerkin projection onto the POD basis to the solution of the PDE. It is expected that if the data set were rich enough to capture the dynamics of the original system, then some sort of convergence properties would exist. To date, there has not been a convergence theory developed for POD that addresses convergence of the finite dimensional approximation to the infinite dimensional solution of the partial differential equation, although the method often works in practice (Atwell and King 1999, Caraballo *et al.* 2003 and Ravindran 1999).

2.3 Galerkin Projection

The Galerkin projection is based upon the idea that the dynamics of a system can be approximated by the dynamics based upon a subspace of the original system. Consider a dynamic system which evolves in a Hilbert space H. This can be shown as, $u(t) \in H$, and $u(t)$ satisfies

$$\dot{u}(t) = X(u(t)), \tag{2.10}$$

where $\dot{u}$ denotes a time derivative $\frac{du}{dt}$ and X is an operator on H. For instance, in a partial differential equation governing a variable $u(x,t)$, defined on some spatial domain $x \in \Omega$, H is a space of functions defined on Ω, and X is a spatial differential operator. Given S, a finite-dimensional subspace of H, Galerkin projection specifies a dynamic system which evolves on S and approximates eq. (2.10) in some sense. This approximate dynamic system is obtained by orthogonal projection of the vector field X onto the subspace, and is denoted

$$\dot{r}(t) = P_S X(r(t)) \tag{2.11}$$

where $r(t) \in S$ just as the original system $u(t) \in H$, and $P_S : H \to S$ is the orthogonal projection map.

To apply this method to the required task, eq. (2.11) needs to be written in terms of a sum of basis functions. Let $\left\{\phi_k\right\}_{k=1}^{N}$ be an orthonormal basis for the subspace S. This is the very same basis discussed in the previous section obtained from POD of the data set. Writing $r(t)$ as a finite sum of time functions $\alpha_k(t)$ with respect to this basis, yields

$$r(t) = \sum_{k=1}^{N} \alpha_k(t)\phi_k(x). \tag{2.12}$$

Rearranging eq. (2.11) gives $P_S(\dot{r} - X(r)) = 0,$ and the equations of motion become

$$\dot{\alpha}_k(t) = \left\langle X(r(t)), \phi_k \right\rangle, \qquad k = 1, \ldots, n, \tag{2.13}$$

based on the orthonormality of the ϕ_k.

For many types of equations, the ODEs given by eq. (2.13) may be determined analytically, in terms of the time function α_k. This is particularly useful computationally, as the inner product in eq. (2.13) does not need to be computed at every timestep. For instance, if $X(u)$ is quadratic, given by

$$X(u) = A(u), \tag{2.14}$$

where A is a linear operator, the projected ODEs (2.13) become

$$\dot{\alpha}_k(t) = A\sum_{i} \left\langle \phi_i, \phi_k \right\rangle \alpha_k(t), \tag{2.15}$$

where the inner products are independent of time and can be determined before numerically solving the ODEs in eq. (2.15) with some initial condition or boundary value.

2.4 Governing Equations

The governing equations for the jet-in-crossflow problem at relatively low Mach number are the incompressible Navier-Stokes equations on a domain Ω, they are often written in the following form,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} \tag{2.16}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.17}$$

Where in 2D $\mathbf{u} = (u, v)$ is the velocity vector, p is the pressure and $\nu = \mu / \rho$ is the kinematic viscosity. If all quantities are made non-dimensional by normalizing velocities by some velocity scale U, lengths by a length scale L, time by U / L, pressure by ρU^2 where ρ is the density, and viscosity by ρUL, then ν is the reciprocal of the Reynolds number Re. Thus, the Navier-Stokes equation (2.16) can be written in terms of Reynolds number for this problem as

$$\frac{\partial \mathbf{u}^*}{\partial t} + (\mathbf{u}^* \cdot \nabla)\mathbf{u}^* - \frac{1}{\mathrm{Re}} \Delta \mathbf{u}^* + \nabla p^* = 0. \tag{2.18}$$

Where, the * operator denotes a non-dimensional quantity. The mathematical definition of incompressibility as shown in eq. (2.17) has been omitted because it is automatically satisfied by the POD basis. This makes physical sense as well, the air in the wind tunnel is clearly compressible, but it is reasonable approximation to model it as an incompressible flow at the low Mach number operating conditions examined in this thesis.

The following boundary conditions are imposed on the problem at hand,

$$\begin{aligned} \mathbf{u} &= (u_{in}, 0) = \mathbf{u}_{\text{in}} && \text{on} \quad \Gamma_{\text{in}} \times [0, T] \\ p\mathbf{n} - \frac{1}{\text{Re}} \frac{\partial \mathbf{u}}{\partial \mathbf{n}} &= (0, 0) && \text{on} \quad \Gamma_{\text{out}} \times [0, T] \\ \mathbf{u} &= (0, 0) && \text{on} \quad \Gamma_{\text{bottom}} \times [0, T] \\ \mathbf{u} &= (u_{\infty}, 0) && \text{on} \quad \Gamma_{\text{top}} \times [0, T] \end{aligned} \tag{2.19}$$

The boundary condition for Γ_{out} is a method of dealing with flow in an unbounded region. See Heywood *et al.* (1996) for a full explanation of the Γ_{out} boundary condition. The values for u_{in} are simply taken to be the mean values at each discrete point on the leading edge of the flow field. This corresponds to points 1 thru 39 inclusive in this numbering scheme.

A weak form of the Navier-Stokes equations (2.16) and (2.17) is necessary for a discrete projection and subsequent reduced-order approximation. For an identical formulation of the weak Navier-Stokes see Ravindran (1999),

$$\begin{aligned} \left\langle \dot{\mathbf{u}} + (\mathbf{u} \cdot \nabla)\mathbf{u}, \Phi \right\rangle + \frac{1}{\text{Re}} \left\langle \nabla \mathbf{u}, \nabla \Phi \right\rangle - \left\langle p, \nabla \cdot \Phi \right\rangle &= 0 \\ \left\langle \nabla \cdot \mathbf{u}, \Psi \right\rangle &= 0 \end{aligned} \tag{2.20}$$

for all test functions Φ and Ψ belonging to Hilbert space $H^1(\Omega)$. The state variables $\mathbf{u}$ and p lay in the $L^2(\Omega)$ space in time between 0 and T. If the velocity $\mathbf{u}$ is written as an expansion in POD modes $\phi(x)$, defined on a spatial domain Ω in which the fluid evolves

$$u(x,t) \approx \overline{u}(x) + \sum_{k=1}^{M} \alpha_k(t)\phi_k(x), \tag{2.21}$$

then the Hilbert space H is just the space of smooth, divergence-free, vector-valued functions on Ω, with the standard inner product

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int_{\Omega} \mathbf{u}(x) \cdot \mathbf{v}(x) dx, \tag{2.22}$$

and norm

$$|\mathbf{u}| = \langle \mathbf{u}, \mathbf{u} \rangle^{1/2}. \tag{2.23}$$

Inserting the expansion (2.21) into the weak form of the Navier-Stokes equations (2.20) gives the Galerkin projection of the weak form

$$\langle \dot{\mathbf{u}} + (\mathbf{u} \cdot \nabla)\mathbf{u}, \phi_i \rangle + \frac{1}{\mathrm{Re}} \langle \nabla \mathbf{u}, \nabla \phi_i \rangle - \langle p, \nabla \cdot \phi_i \rangle + \left\langle p\mathbf{n} - \frac{1}{\mathrm{Re}} \frac{\partial \mathbf{u}}{\partial \mathbf{n}}, \phi_i \right\rangle_{\Gamma_{\mathrm{out}}} = 0 \tag{2.24}$$

Now it is important to note that the basis vectors ϕ_i are naturally divergence free due to the fact that the flow in the experimental data is incompressible. This makes the pressure term vanish. Also, the flow satisfies zero boundary conditions on Γ_{out} so that the boundary term vanishes; see Ravindran (1999) for a similar problem. Now eq. (2.24) becomes

$$\langle \dot{\mathbf{u}} + (\mathbf{u} \cdot \nabla)\mathbf{u}, \phi_i \rangle + \frac{1}{\mathrm{Re}} \langle \nabla \mathbf{u}, \nabla \phi_i \rangle = 0 \tag{2.25}$$

For all $\phi_i \in span\{\phi_1, \phi_2, \ldots, \phi_N\}$. When eq. (2.21) is substituted in to eq. (2.25) the following nonlinear equation for the coefficients $\alpha_i(t)$ is obtained.

$$\begin{aligned} \dot{\alpha}(t) &= \mathbf{A}\alpha(t) + \alpha^T(t)\mathbf{N}\alpha(t) + \mathbf{e}, \\ \alpha(0) &= \boldsymbol{\alpha}_0 \end{aligned} \tag{2.26}$$

where

$$\begin{aligned} \mathbf{A}_{ij} &= -\left\langle \phi_j \cdot \nabla \bar{\mathbf{u}}, \phi_i \right\rangle - \left\langle \bar{\mathbf{u}} \cdot \nabla \phi_j, \phi_i \right\rangle - \frac{1}{\mathrm{Re}} \left\langle \nabla \phi_j, \nabla \phi_i \right\rangle, \\ \mathbf{N}_{ikl} &= -\left\langle \phi_k \cdot \nabla \phi_l, \phi_i \right\rangle, \\ \mathbf{e}_i &= -\left\langle \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}}, \phi_i \right\rangle - \frac{1}{\mathrm{Re}} \left\langle \nabla \bar{\mathbf{u}}, \nabla \phi_i \right\rangle, \\ \boldsymbol{\alpha}_{0i} &= \left\langle u_0, \phi_i \right\rangle. \end{aligned} \tag{2.27}$$

2.5 Implementation

Application of the Galerkin method to the nonlinear convective term **N** produces three products of nodal values. This feature makes eq. (2.26) far more costly to solve, because the nonlinear term must be computed at each time step as a subroutine of the ODE solver. However, this unnecessary computation can be avoided by introducing the group finite element formulation (Fletcher 1988). First, rewrite the 2-D Incompressible Navier-Stokes Equations (2.16) in conservation form as

$$\begin{aligned} u_t(x,y,t) + \left(u^2(x,y,t)\right)_x + \left(uv(x,y,t)\right)_y &= -p_x(x,y,t) + \frac{1}{\mathrm{Re}}\left(u_{xx}(x,y,t) + u_{yy}(x,y,t)\right), \\ v_t(x,y,t) + \left(uv(x,y,t)\right)_x + \left(v^2(x,y,t)\right)_y &= -p_y(x,y,t) + \frac{1}{\mathrm{Re}}\left(v_{xx}(x,y,t) + v_{yy}(x,y,t)\right), \end{aligned}$$

and introduce the weak form

$$\begin{aligned} \frac{\partial}{\partial t}\left\langle u, \phi \right\rangle &= -\left\langle u^2{}_x, \phi \right\rangle - \left\langle uv_y, \phi \right\rangle - \left\langle p_x, \phi \right\rangle + \frac{1}{\mathrm{Re}}\left(\left\langle u_x, \phi_x \right\rangle + \left\langle u_y, \phi_y \right\rangle\right), \\ \frac{\partial}{\partial t}\left\langle v, \phi \right\rangle &= -\left\langle uv_x, \phi \right\rangle - \left\langle v^2{}_y, \phi \right\rangle - \left\langle p_y, \phi \right\rangle + \frac{1}{\mathrm{Re}}\left(\left\langle v_x, \phi_x \right\rangle + \left\langle v_y, \phi_y \right\rangle\right). \end{aligned} \tag{2.28}$$

As before, the linear terms $u(x,t)$ are approximated by a finite sum as in eq. (2.1), except now the nonlinear terms $u^2(x,t)$ are approximated by the group formulation

$$\begin{aligned} u^2(x,t) &= \sum_{k=1}^{N} u_k^2(t)\phi_k(x), \\ v^2(x,t) &= \sum_{k=1}^{N} v_k^2(t)\phi_k(x), \\ uv(x,t) &= \sum_{k=1}^{N} uv_k(t)\phi_k(x). \end{aligned} \tag{2.29}$$

Substituting these group formulations and the standard finite sum into the weak form and eliminating the pressure term as before gives the group finite element version of the ordinary differential equations

$$\sum_i \langle \phi_j, \phi_i \rangle \dot{\alpha}_i(t) = \sum_i \frac{1}{\mathrm{Re}} \langle \phi'_j, \phi'_i \rangle \alpha_i(t) - \sum_i \langle \phi'_j, \phi_i \rangle \alpha_i^2(t). \tag{2.30}$$

Thus the ODE is obtained

$$\mathbf{M}\dot{\alpha}(t) = \mathbf{A}\alpha(t) + \alpha^T(t)\mathbf{N}\alpha(t). \tag{2.31}$$

Where again $\alpha(t) = [\alpha_1(t) \ldots \alpha_N(t)]^T$. $\mathbf{M}$ is still the identity matrix due to the orthonormality of the basis vectors, $\mathbf{A}$ is the same as given in eq. (2.27), and $\mathbf{N}$ is governed by

$$\mathbf{N}_{ij} = -\langle \nabla \phi_j, \phi_i \rangle. \tag{2.32}$$

In order to solve for the time coefficients $\alpha_k(t)$ spatial derivatives of the basis vectors $\phi_k(x)$ are necessary. These could be solved for directly through a finite difference method; however this method is extremely sensitive to noise in the experimental data. This problem can be negotiated by casting each basis vector $\phi_k(x)$ as a finite sum of linear spline functions

$$\phi_k(x) = \sum_{j=1}^{N} \delta_j^k \beta_j \tag{2.33}$$

where β_j is the linear spline functions and δ_j^k is the coefficient for each spline. It is a very nice property of these functions that the coefficiens δ_j^k are equal to the values contained in the original basis vectors ϕ_k. Using eq. (2.33) it is possible to rewrite the inner product of two basis vectors $\langle \phi_i, \phi_j \rangle$

$$\begin{aligned} \langle \phi_i, \phi_j \rangle &= \int_\Omega \phi_i(x) \cdot \phi_j(x) dx \\ &= \int_\Omega \sum_{k=1}^{N} \phi_k^i \beta_k \sum_{l=1}^{N} \phi_l^j \beta_l dx \\ &= \left[\phi_1^i, \phi_2^i, \ldots, \phi_N^i \right] \int_\Omega \beta_k \beta_l dx \begin{bmatrix} \phi_1^j \\ \phi_2^j \\ \vdots \\ \phi_N^j \end{bmatrix} \end{aligned} \tag{2.34}$$

Similarly,

$$\langle \nabla\phi_i, \nabla\phi_j \rangle = \int_\Omega \nabla\phi_i(x) \cdot \nabla\phi_j(x) dx = [\phi_1^i, \phi_2^i, \ldots, \phi_N^i] \int_\Omega \nabla\beta_k \cdot \nabla\beta_l dx \begin{bmatrix} \phi_1^j \\ \phi_2^j \\ \vdots \\ \phi_N^j \end{bmatrix} \tag{2.35}$$

Thus, it is possible to avoid using a finite difference method for differentiating the basis vectors: instead spatial derivatives of the spline functions are used and simply multiplied by the basis vectors to evaluate the inner products involving spatial derivatives.

CHAPTER 3

RESULTS AND ANALYSIS

The major result of this thesis is the production of a dynamic model for the uncontrolled Navier-Stokes equations of the form

$$\dot{u}(t) = \mathbf{A}u(t) + \mathbf{N}\left(u(t)\right). \tag{3.1}$$

To produce a model that can be used for the purpose of control, an actuator needs to be included. This dynamic system takes the form

$$\dot{u}(t) = \mathbf{A}u(t) + \mathbf{N}\left(u(t)\right) + \mathbf{B}g(t). \tag{3.2}$$

In this system, $g(t)$ is the actuator command. For instance, if the jet were to be used as an actuator to control the flow in some way, then $g(t)$ would be the signal that commands the jet action and the $\mathbf{B}$ operator would represent the effect of that signal on the flow. In order to obtain such a model using POD, the data set for the actuator must contain input signals and output effects. Since no such data set is available at this writing, instead this thesis focuses on producing a dynamic model of the jet-in-crossflow as a first step toward obtaining dynamic models from experimental data.

3.1 POD from Data

Prior to performing a Galerkin projection it is possible to plot the POD modes and visually confirm that they capture the dynamic structures of the system. Figures 4 and 5 show the first six POD modes for the u and v components of velocity respectively with the time averaged mean flow removed from the data set.

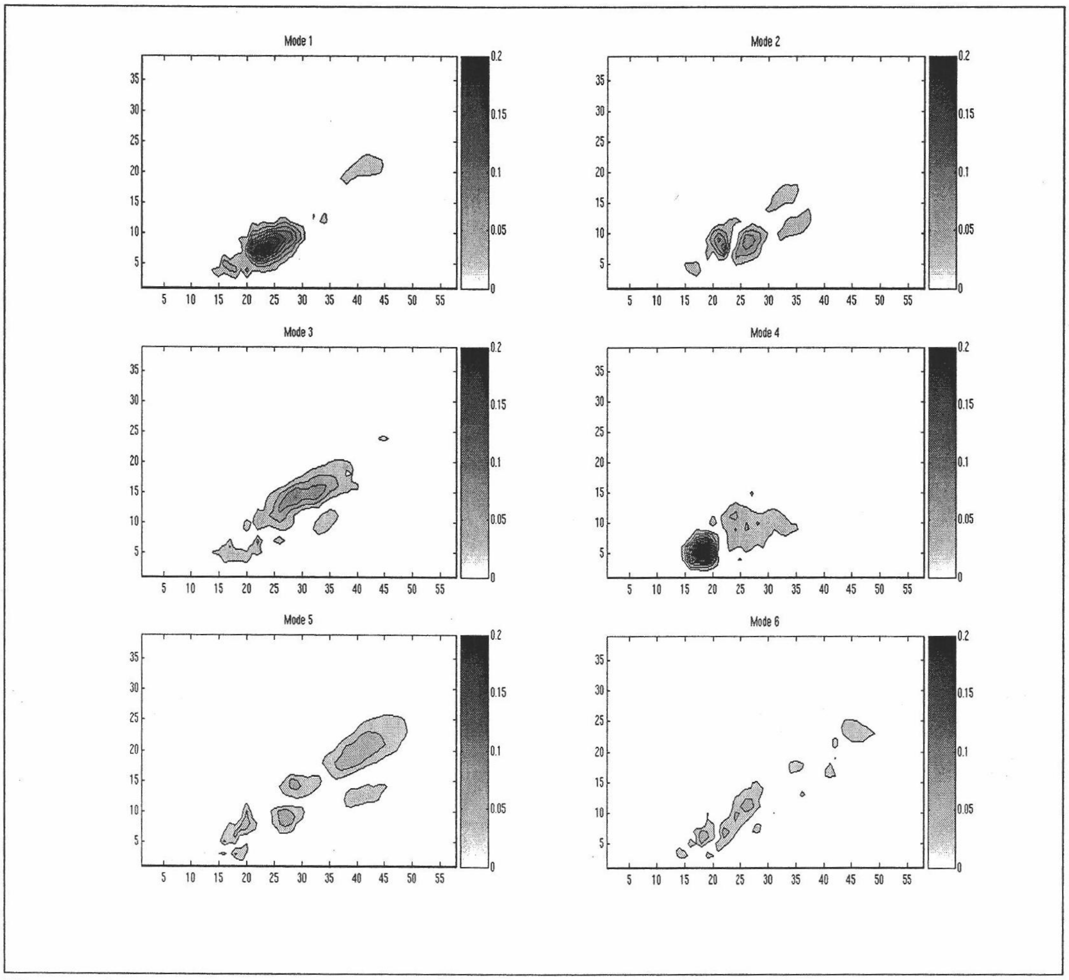


Figure 4: *u* component eigenfunctions of the first six POD modes based on wind tunnel data with a steady jet. The time averaged *u* component of velocity was removed from the data set.

These POD modes represent the primary dynamics of the original data set and a summation of all 200 modes will reconstruct the original data in full. Similarly, a summation of a reduced number of the POD modes will capture a certain percentage of the original dynamics. The squares of the singular values represent the kinetic energy captured by each mode and a percentage of energy captured by each mode can be calculated; this is shown in Figure 6. The mean flow by itself captures 90% of the energy of the data set, and the first ten POD modes capture approximately 30% of the

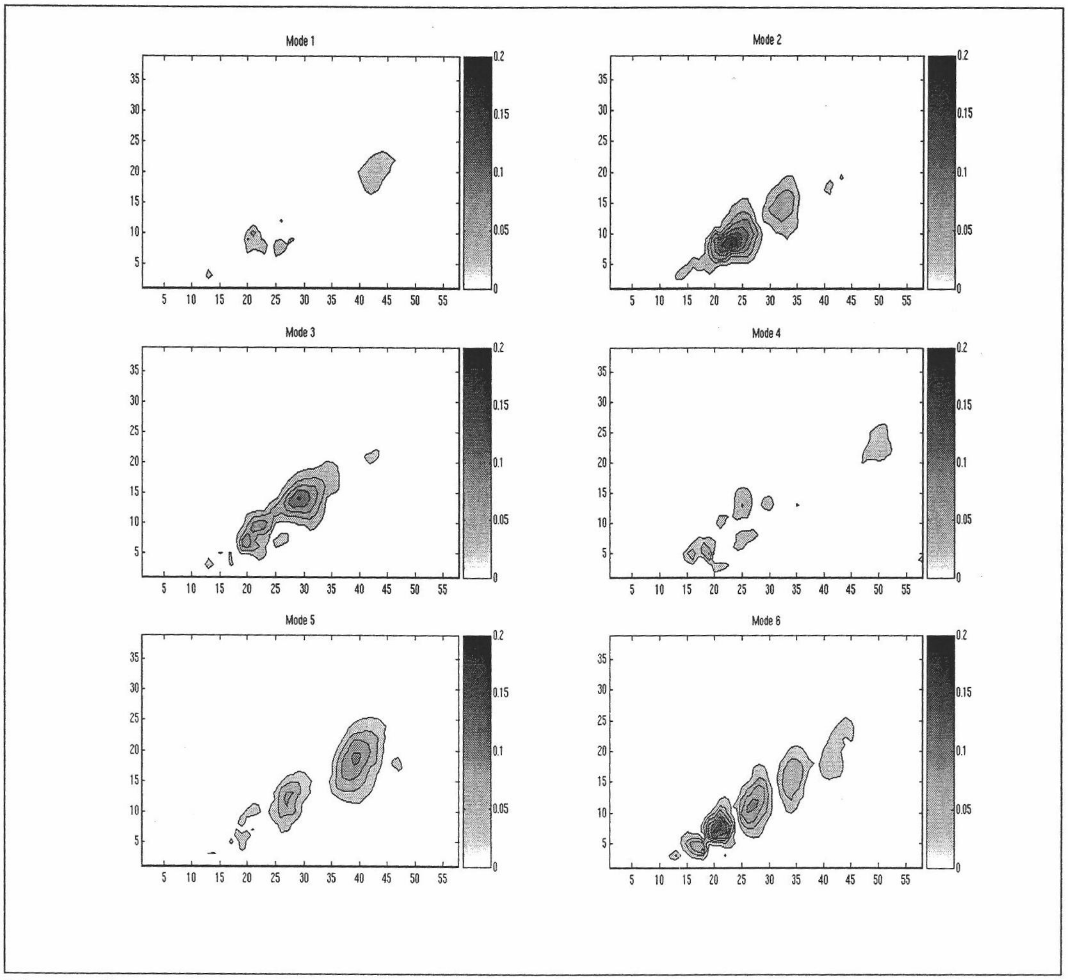


Figure 5: *v* component eigenfunctions of the first six POD modes based on wind tunnel data with a steady jet. The time averaged *v* component of velocity was removed from the data set.

remaining energy of the data. It is important to note that the percent energy trend line in Figure 6 is a cumulative sum of the energies and will always reach 100% when the number of POD modes equals the number of time snapshots in the experimental data set. This is because the summation of energies is divided by the total energy in the data set which does not always capture all of the energy of the physical system. The number of POD modes used to generate the dynamic model can be varied, but the increased accuracy associated with more POD modes comes at the cost of computing efficiency. This will be discussed more fully in Section 3.3.2.

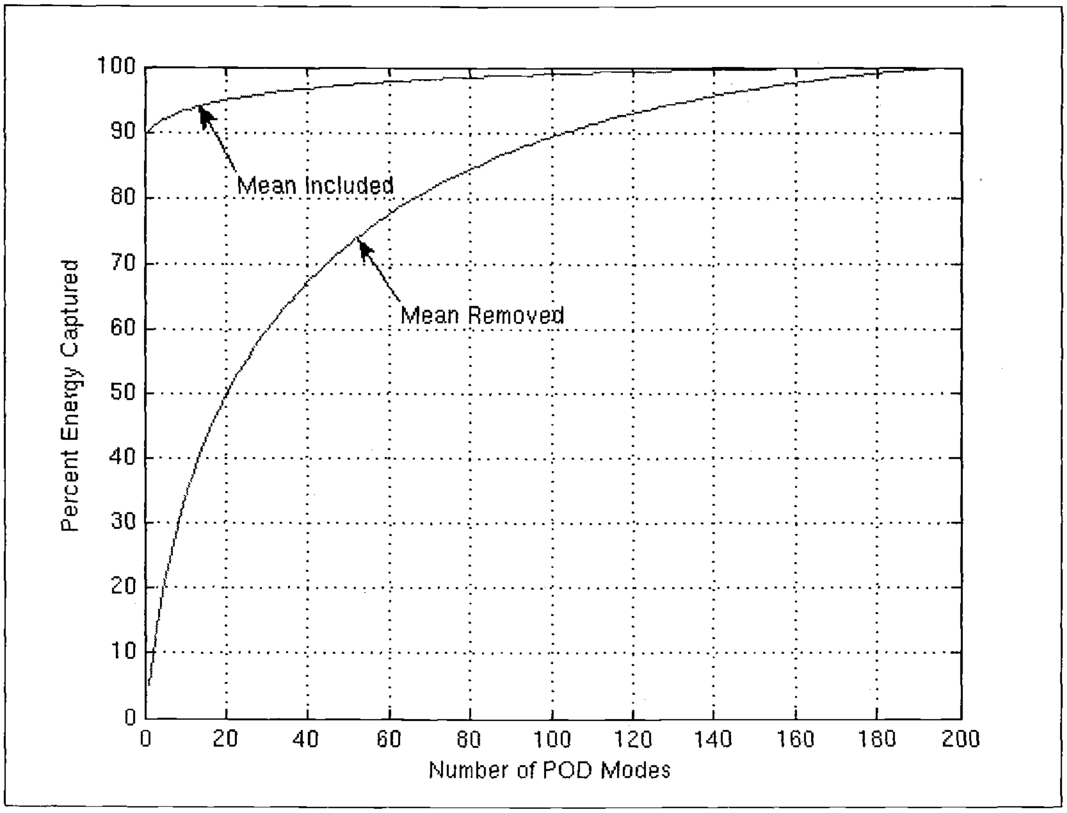


Figure 6: Summation of percent energy captured by each POD mode.

Since 90% of the energy captured by the data is contained in the mean flow it is important to show exactly what this flow looks like. Figure 7 shows the mean flow in the *u* and *v* directions taken directly from experimental data.

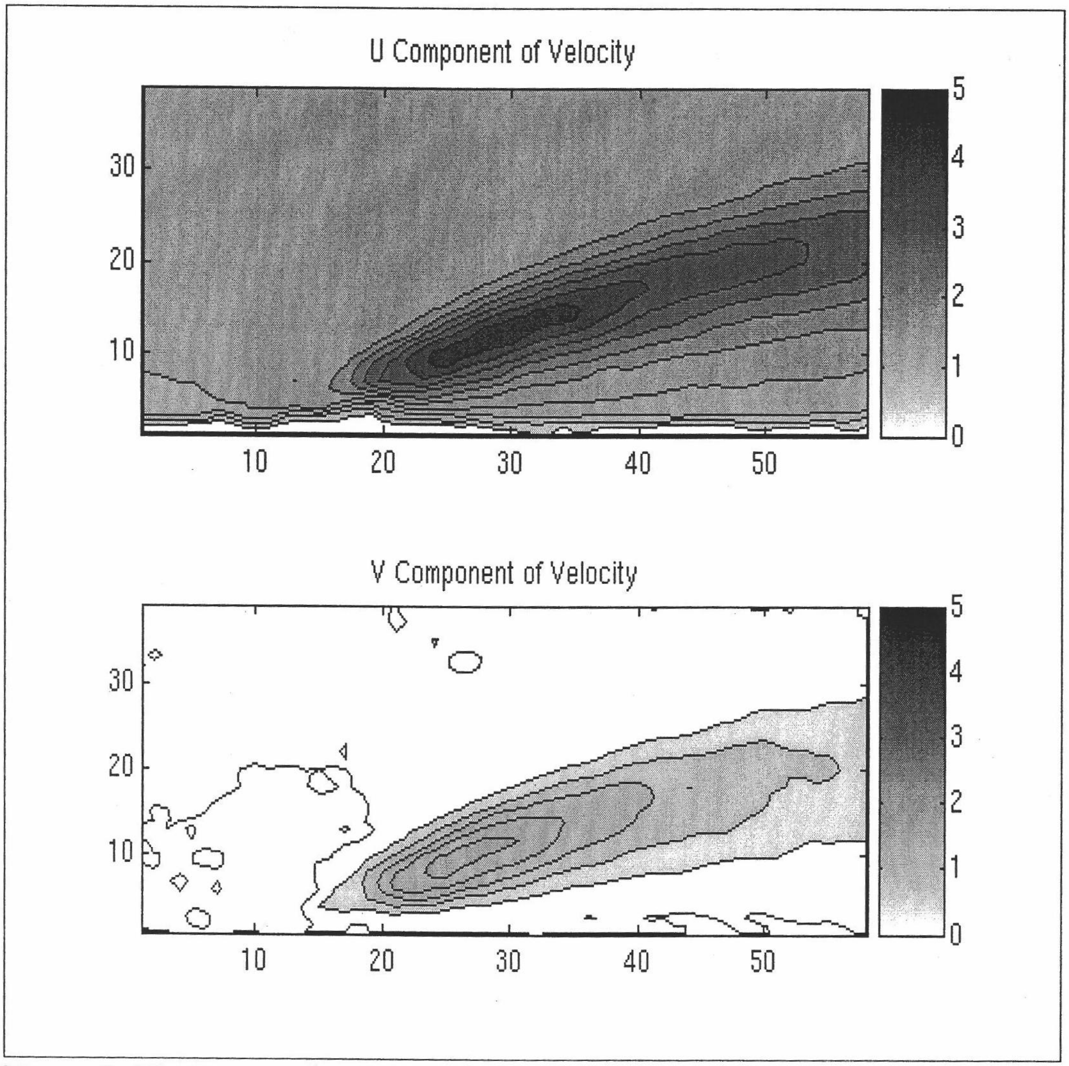
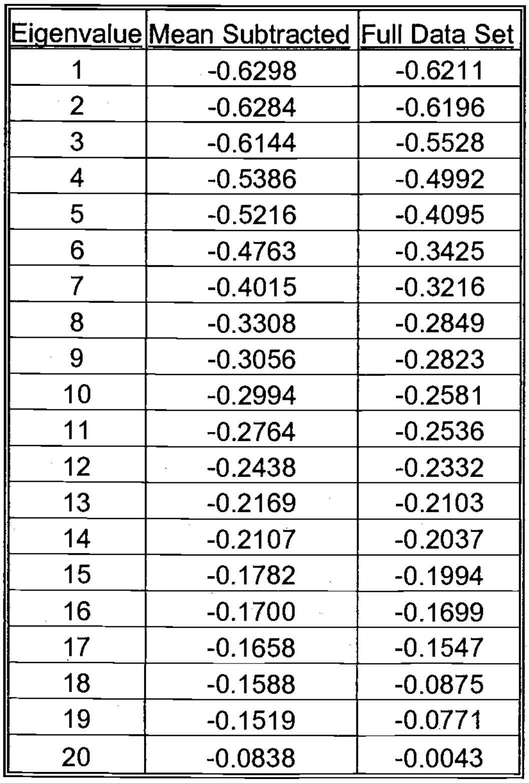


Figure 7: Time averaged mean velocity components in the u and v directions.

3.2 Model Stability

In two cases, where the mean flow is subtracted before the POD basis is calculated and no mean is subtracted, the systems of ODEs obtained via Galerkin projection as described in Section 2.3 are asymptotically stable. This is to say that all of the eigenvalues are negative. This is expected and further validates the model. The sorted negative eigenvalues of the $\mathbf{A}$ matrix are shown in Figure 8. These are generated for a system based on 10 POD modes. The system is always twice the size of the number of basis functions used due to the two components of velocity, thus the 20 eigenvalues. While the figure is shown for a system based on 10 POD modes, both larger and smaller systems of ODEs have negative eigenvalues of the $\mathbf{A}$ matrix.



Eigenvalue	Mean Subtracted	Full Data Set
1	-0.6298	-0.6211
2	-0.6284	-0.6196
3	-0.6144	-0.5528
4	-0.5386	-0.4992
5	-0.5216	-0.4095
6	-0.4763	-0.3425
7	-0.4015	-0.3216
8	-0.3308	-0.2849
9	-0.3056	-0.2823
10	-0.2994	-0.2581
11	-0.2764	-0.2536
12	-0.2438	-0.2332
13	-0.2169	-0.2103
14	-0.2107	-0.2037
15	-0.1782	-0.1994
16	-0.1700	-0.1699
17	-0.1658	-0.1547
18	-0.1588	-0.0875
19	-0.1519	-0.0771
20	-0.0838	-0.0043

Figure 8: Eigenvalues of $\mathbf{A}$ $(1*10^{-4})$.

3.3 Model Simulation

This portion of the results section presents findings that contribute new knowledge to the engineering community. The solution to the initial value problem

$$\mathbf{M}\dot{\alpha}(t) = \mathbf{A}\alpha(t) + \alpha^{T}(t)\mathbf{N}\alpha(t) + \mathbf{e} \tag{3.3}$$

was obtained using MatLab's ODE45 solver which utilizes a 4/5 Runge-Kutta method for the coefficients α of the POD approximation. Where $\mathbf{M}$ is the identity matrix, $\mathbf{A}$ and $\mathbf{e}$ are given in eq. (2.27), and $\mathbf{N}$ is given in eq. (2.32). Once the coefficients α are obtained they are multiplied by their respective basis vectors and a summation of the weighted basis vectors simulates the flow.

Three different initial conditions were evaluated. This results section is organized into three separate sub-sections based on the initial condition used. The in initial condition $\boldsymbol{\alpha}_0$ is formed as follows

$$\boldsymbol{\alpha}_{0i} = \langle u_0, \phi_i \rangle, \tag{3.4}$$

where u_0 is a function, and $\boldsymbol{\alpha}_0$ is the projection onto the set of POD basis. Here the subsections to follow are described; the first initial condition (Section 3.3.1) examined is simply the first POD mode. In this section the results show differences between subtracting the mean flow from the data and raw data with no mean flow removed. The second initial condition (Section 3.3.2) is Couette flow. In this section the number of POD modes used to generate the model was varied. The third and final section (3.3.3) looks at what happens with an initial condition very dissimilar to any of the POD modes. To achieve this, a uniform flow of high velocity is used as an initial condition. In this section the timestep used by the ODE solver is varied to assess convergence of the model.

It is important to emphasize the degree to which this dynamic model is verifiable within the scope of a thesis. Ideally a couple of methods of verification could be used, but both would be extensive projects in and of themselves. The first option for testing this model would be to apply initial conditions that can be verified experimentally. That is similar to what is done in section 3.3.1, however it would be beneficial to have an initial condition different from that at which the data were collected. To do that a whole new experimental data set would be necessary to verify the results of the model created here. That data set is not yet available. A second method of verifying the results of this model would be to create CFD code for this jet-in-crossflow problem. The CFD code could then be used to verify this model created from experimental data. Creation of a control based on this model and subsequent verification of that control in an experimental environment is another possible method of model verification. Again, this is beyond the scope of this thesis. For this reason, eigenvalue analysis (Section 3.2), simulation of the model with different initial conditions (Sections 3.3.1-3.3.3), variation in the number of modes used in the dynamic model (Section 3.3.2), and convergence analysis (section 3.3.3) will suffice as a workable substitute in the area of model validation.

3.3.1 1st POD mode initial condition

If the first POD mode is used as an initial condition in the simulation, then presumably the experimental data should be reproduced with some degree of accuracy. This initial condition is implemented by setting

$$\boldsymbol{\alpha}_0 = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T . \tag{3.5}$$

This is for a system with the number of basis functions $n = 10$. Two cases were tested with this initial condition, one with the mean flow removed from the data and the other with no mean flow removed.

It is standard practice when using POD of CFD simulations to subtract the mean from the computational fluid dynamic data set before computing the POD basis vectors. It makes sense that this step would also be necessary with experimental data. This is done because the mean flow dynamics dominate all lesser dynamics captured in subsequent POD modes, and Figure 9 depicts this visually. It shows the u component of velocity at three time steps during the simulation with little difference. The simulation is completely dominated by the first POD mode which is very similar to the mean flow shown in Figure 7. This first POD mode has far stronger dynamics than any of the subsequent POD modes, thus changes in the subsequent modes are covered up and the simulation just appears as a steady mean flow; recall that 90% of the energy is captured in this mode. This can be contrasted with Figure 10, which shows and identical simulation of u velocities, except the mean flow was subtracted from the experimental data prior to computing the POD basis.

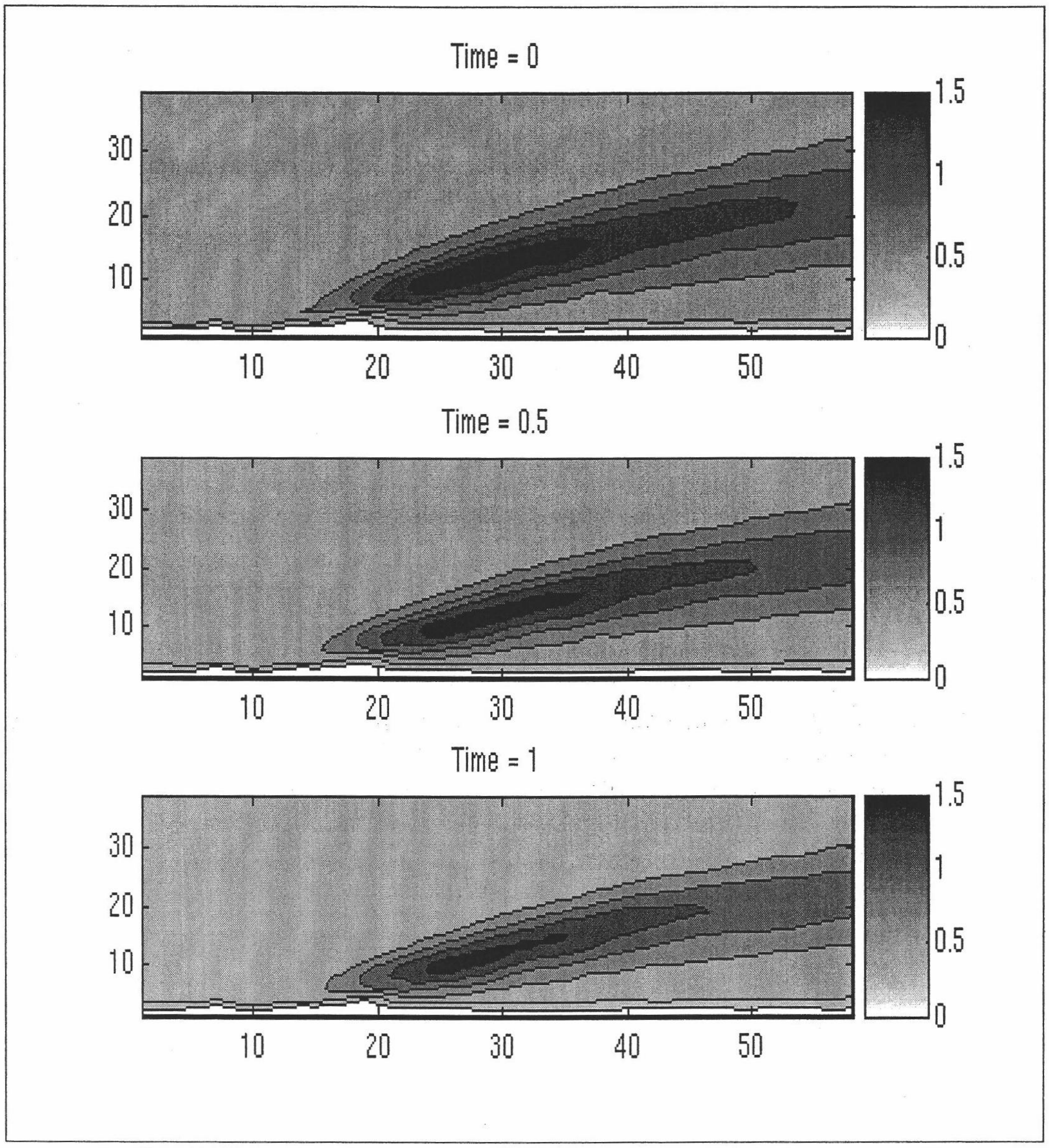


Figure 9: Ten POD mode simulation of the *u* component without the mean flow removed and first POD mode initial condition.

It is repetitive to show the v component counterpart of Figure 9, as it shows a steady state very similar to the mean v component in Figure 7. Even if initial conditions other than the first POD mode are used, the first mode quickly becomes the dominant characteristic of the flow.

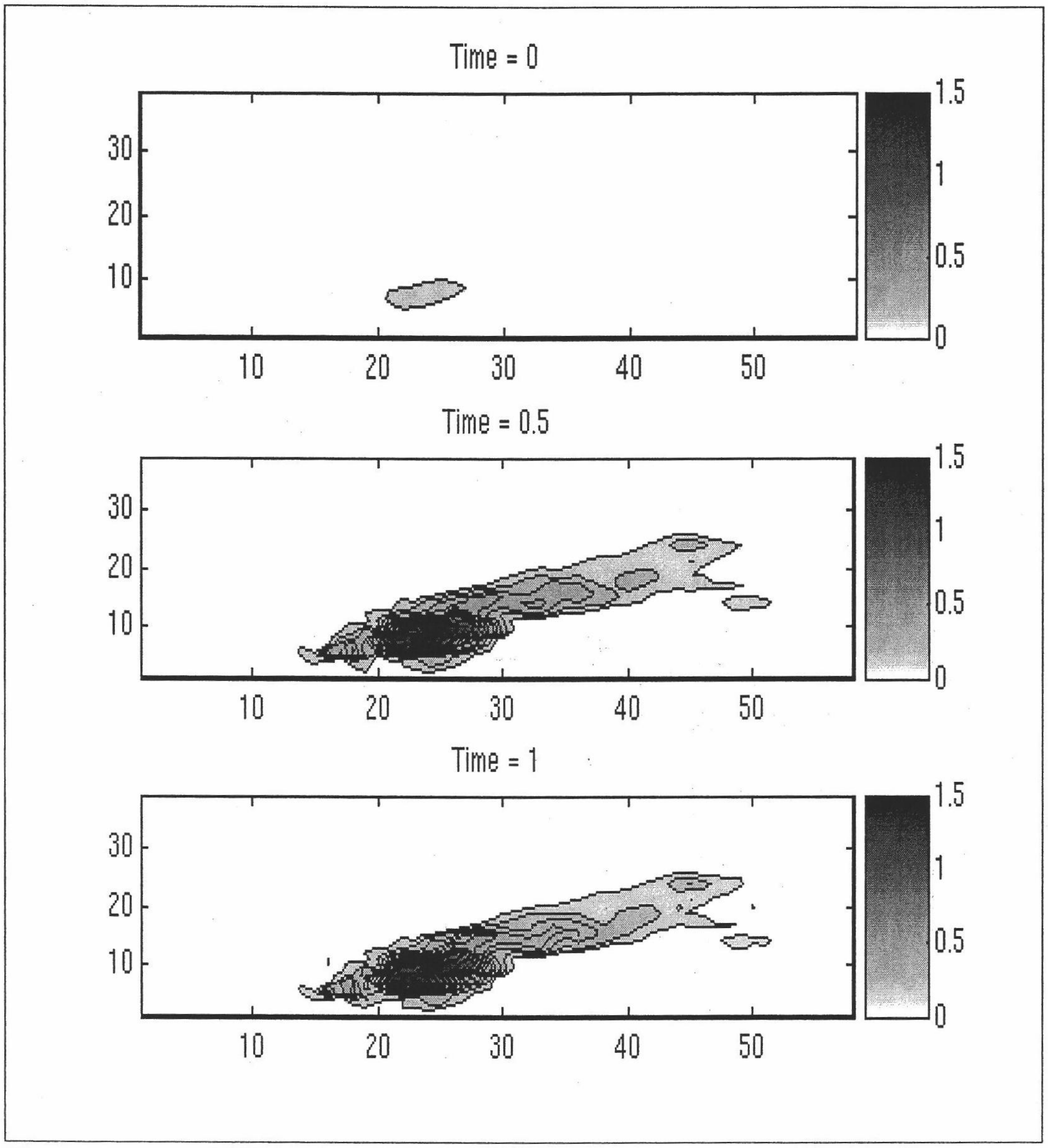


Figure 10: Ten POD mode simulation of the u component with the mean flow removed and first POD mode initial condition.

3.3.2 Couette flow initial condition

This initial condition varies the flow linearly in the wall normal direction from $u = 0$ at the wall boundary (bottom) to $u = 1$ at the top of the flow. The initial condition $v = 0$ was used for the whole domain. In this section it is also assessed how the number of POD modes used to generate the model affects the solution. Three, ten and 100 POD mode models were all created and given the same couette flow initial condition. This initial condition did result in a different $\boldsymbol{\alpha}_0$ vector for each model due to the different sizes of the systems of ODEs. For example, the three POD basis model used

$$\boldsymbol{\alpha}_0 = \left[\text{-2.68 -0.50 13.57 0 0 0}\right]^T,$$

while the 10 POD basis model used

$$\boldsymbol{\alpha}_0 = \left[\text{-2.68 -0.50 13.57 6.40 -12.79 7.42 -2.47 -7.52 7.96 -6.34 0 0 0 0 0 0 0 0 0 0}\right]^T.$$

More POD functions should in theory provide greater accuracy in the model, but the cost of computing and running simulations increases. At some point it is no longer feasible to compute in real time. This may not be an issue for "off-line" computations, however the goal is to provide the needed accuracy for the final application of the model, be it controller design, stability analysis or optimization, with as few POD modes as possible. Figures 11-13 all show the same simulation, except the number of POD modes utilized in constructing the model were varied.

As expected, the more POD modes that are used in construction of the model the more detailed the results of the simulation. A greater overall non-dimensional velocity is seen in the 100 POD mode simulation simply because the simulated flow is a summation of each coefficient multiplied by each POD mode. Because there are more modes there are more terms in the summation and a greater overall velocity. Also, dynamics can be included in the larger system of ODEs that there is no possible way to include in the more reduced models just because those dynamics are not captured by the higher energy or lower numbered POD modes.

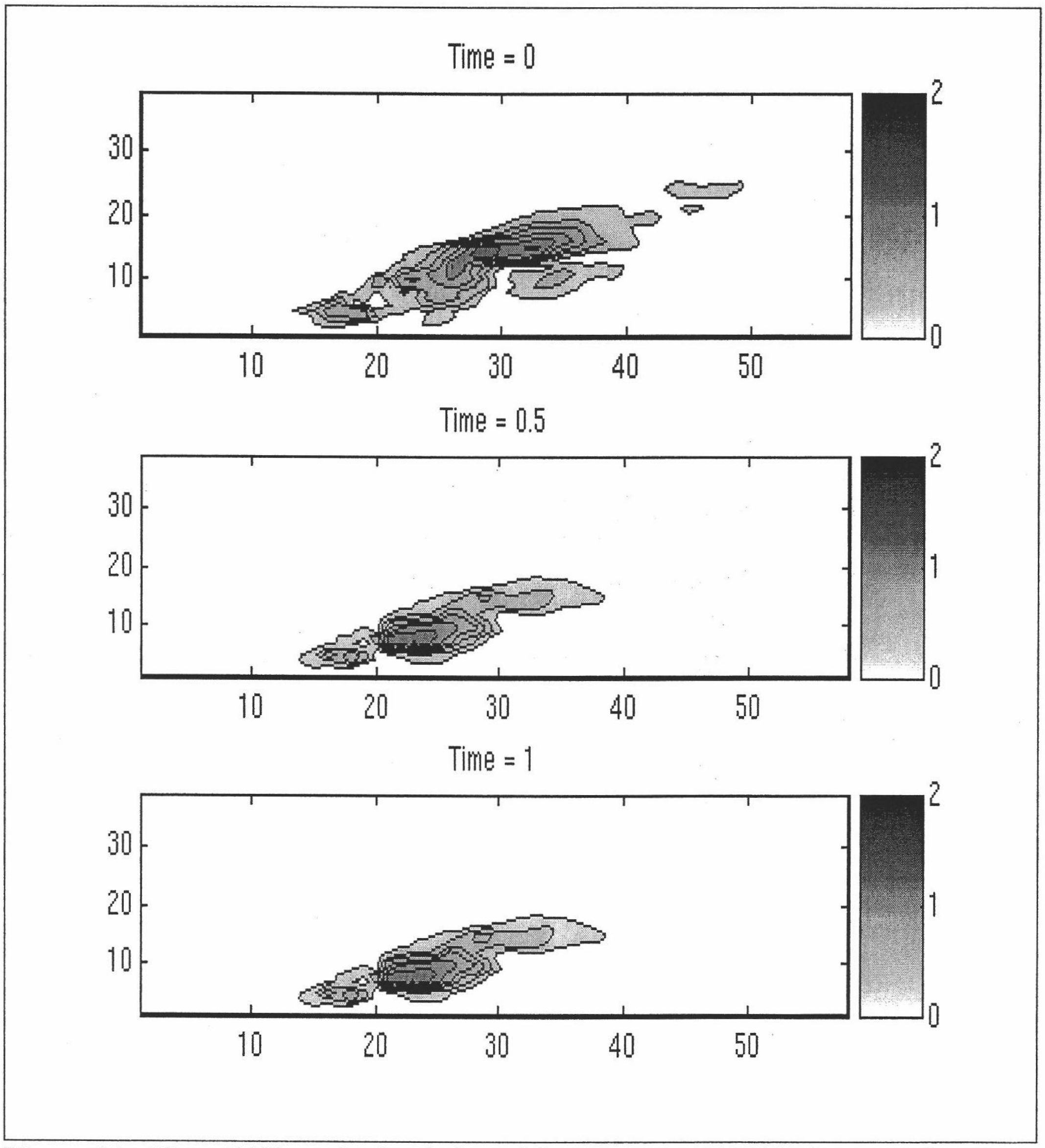


Figure 11: Three POD mode simulation of the *u* component with the mean flow removed and a couette flow initial condition.

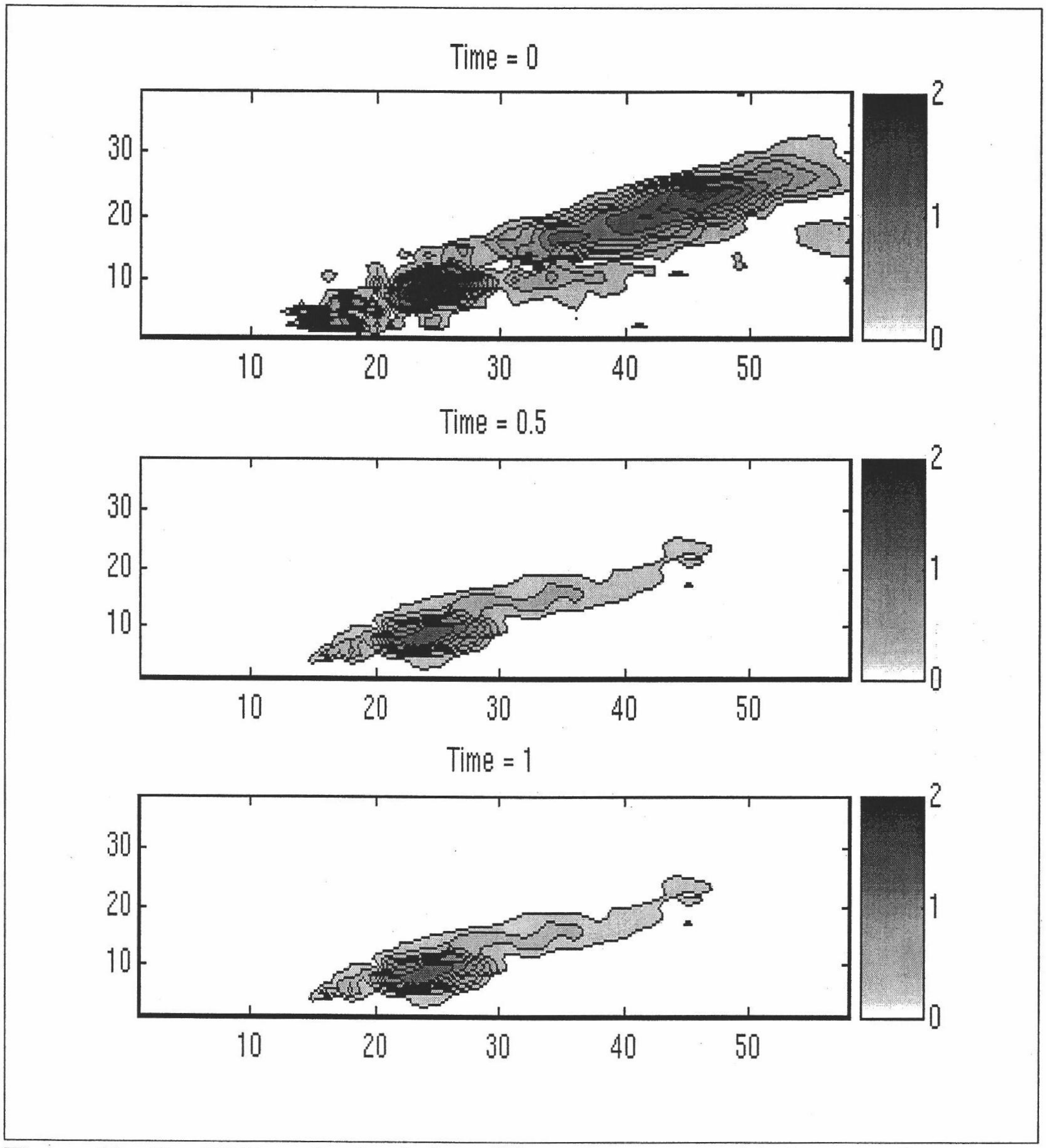


Figure 12: Ten POD mode simulation of the *u* component with the mean flow removed and a couette flow initial condition.

Figure 13 shows the same simulation as Figs. 11 and 12, but with 100 POD functions. The resolution of the model is noticeably increased, however the extra computing time required for a model of this size is significantly greater, approximately 10 times. The results for the v component are very similar in all cases and inclusion of those figures would be repetitive.

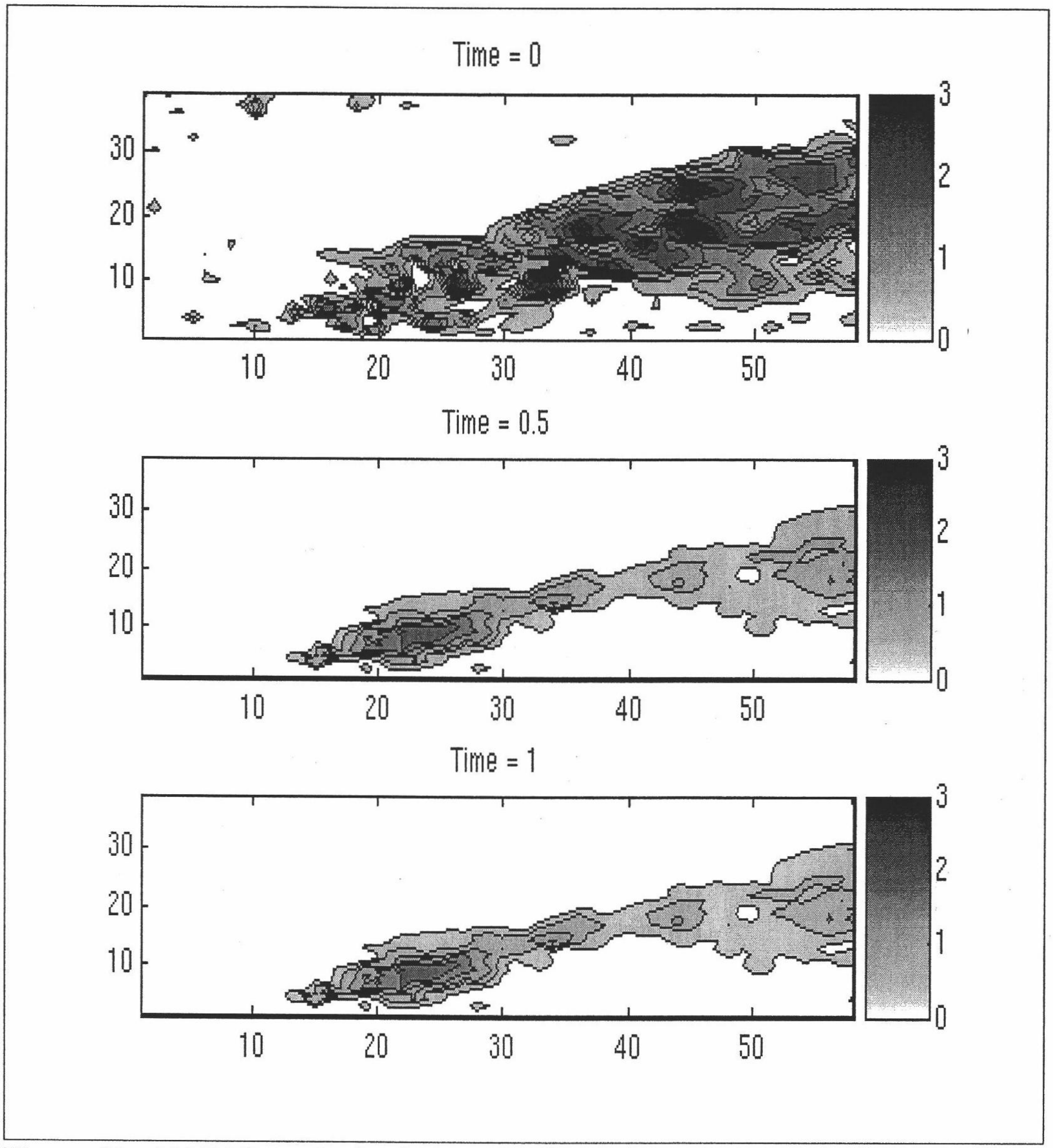


Figure 13: 100 POD mode simulation of the *u* component with the mean flow removed and a couette flow initial condition.

3.3.3 Constant high velocity initial condition

The inclusion of this initial condition has the purpose of evaluating how the model responds to initial conditions far away from the information contained in the data set. The initial condition was implemented with a constant non-dimensional velocity in the x direction of 1000 and no velocity in the y direction. This generates the following initial condition:

$$\boldsymbol{\alpha}_0 = 1000 * [\text{-3.61 0.07 3.19 0.71 -2.32 1.91 0.52 -1.52 0.72 -2.29 0 0 0 0 0 0 0 0 0 0}]^T.$$

To attempt to assess convergence, various timesteps within the simulation are used. All previous simulations have been run with MatLab's adaptive time stepper which decreases Δt as the system changes more rapidly. In this section the time step is refined from $\Delta t = 0.01$ to $\Delta t = 0.1*10^{-3}$ to $\Delta t = 0.1*10^{-6}$ in order to see if the same or a similar solution is generated. With each time step size the Frobenius norm of the non-dimensional velocities is evaluated once the system has reached steady state, which it does by $t = 1$ in the simulation. With all timesteps the Frobenius norm of the final steady state response was identical which indicates that the model converges. Figure 14 shows how the model responds to the large steady flow initial condition.

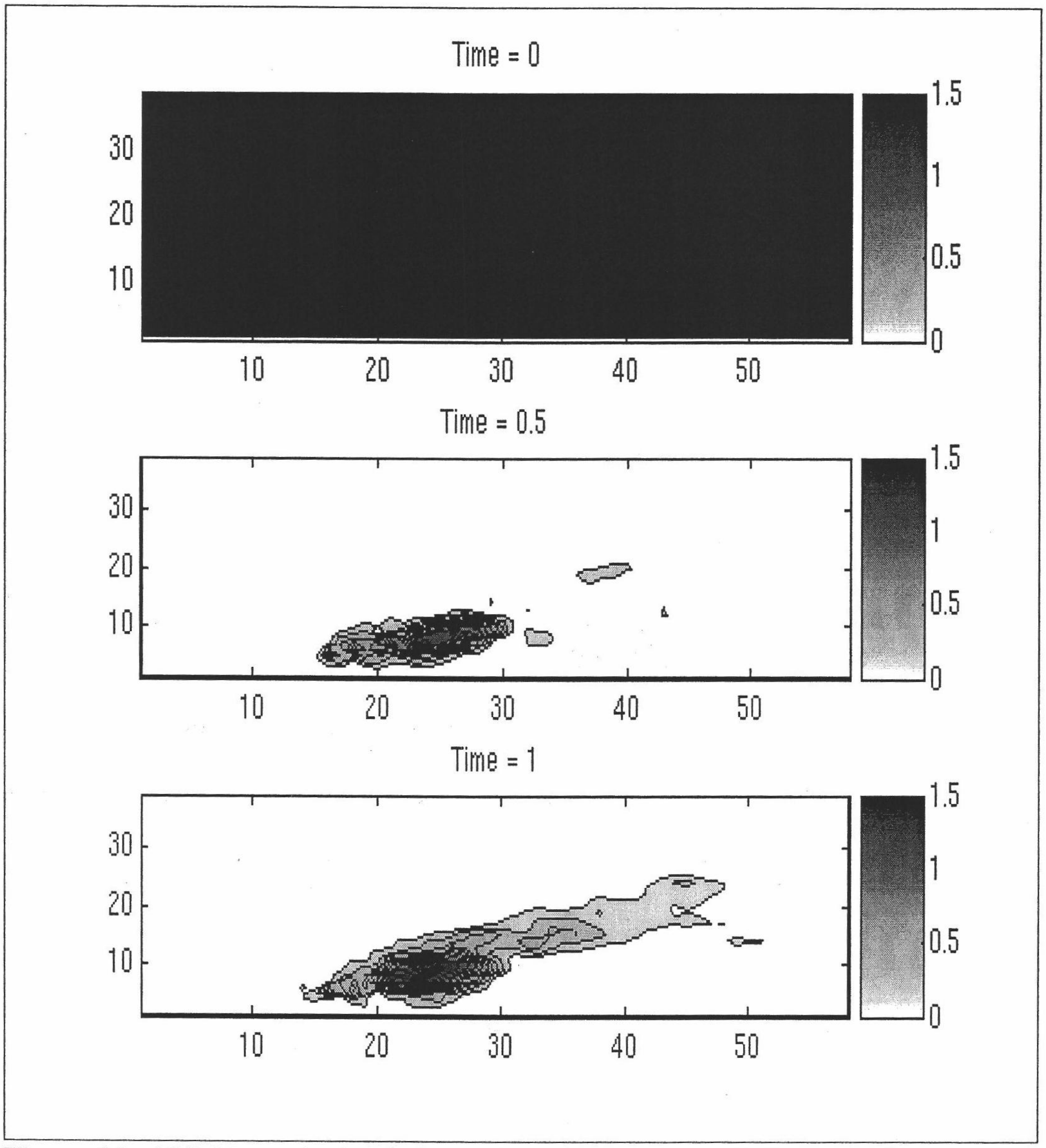


Figure 14: 10 POD mode simulation of the u component with the mean flow removed and a steady flow initial condition $u = 1000$.

CHAPTER 4

CONCLUSION

The present reduced-order model does capture the dynamics of the jet-in-crossflow system. This is significant progress toward the end goal of reduced order model based closed-loop flow control. The reduced order model is obtained through the use of snapshot based Proper Orthogonal Decomposition of PIV data in conjunction with a Galerkin Projection of the Navier-Stokes equations onto the POD basis functions. This method appears to achieve the desired goal, however a working controller based on the reduced order model must be created to verify this result.

Since the end purpose of this model is ultimately closed-loop control design which is outside the scope of this work it is difficult to assess the overall success of this model. Even if an adequate controller can be created based on this model there are still inherent limitations to deriving a model from POD of experimental data. The primary limitation of POD is that the POD basis functions are intrinsic to the particular flow they were created with. As the flow is modified by the action of the control or disturbances in the flow the POD basis will change. It follows that, a reduced order model built around a set of POD basis functions for one flow would certainly not be optimal and may not even represent the dynamics of the controlled flow under some conditions. This dynamic model is based on one specific set of steady operating conditions for the flow, thus it only captures those dynamics. Likely, this particular model is not very suitable for many applications, but the methodology used to create the model and corresponding MatLab code can now be used to generate a similar model given a richer data set.

Future work is clearly needed in a number of areas. Development and verification of a working control system in the laboratory environment is the next obvious step. Also, due to the impracticalities of having full PIV data outside the laboratory environment, a stochastic method must be used for real-time estimation of the time coefficients. Input to this model would presumably come from pressure measurements on the plate surface downstream of the jet. It could also prove useful to adapt the present model to three dimensions by adding additional "slices" of PIV data. Addressing all three of these areas would provide a fine PhD dissertation for an interested individual.

BIBLIOGRAPHY

Atwell, J. A. and B. B. King, “Reduced Order Controllers for Spatially Distributed Systems via Proper Orthogonal Decomposition,” *ICAM Report* 99-07-01, 1999.

Berkooz, G., P. Holmes, and J. L. Lumley, “The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows,” *Annual Review of Fluid Mechanics,* Vol. 25, pp. 539-575, 1993.

Bernero, S., and H. E. Fiedler, “Application of Particle Image Velocimetry and Proper Orthogonal Decomposition to the Study of a Jet in a Counterflow,” *Experiments in Fluids*, Vol. 29, No. 7, pp. S274 - S281, Dec. 2000.

Bewley, T. R., “Flow Control: New Challenges for a New Renaissance,” *Progress in Aerospace Sciences,* Vol. 37, No. 21, 2001.

Bewley, T. R. and S. Liu, “Optimal and Robust Control and Estimation of Linear Paths to Transition,” *J. Fluid Mech.*, Vol. 365, No. 305, 1998.

Brooks, G. P., and J. M. Powers, *A Karhunen-Loeve Galerkin Technique with Shock Fitting for Optimization of a Blunt Body Geometry,* AIAA-2002-3861, 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Indianpolis, Indiana, July 2002.

Canuto, C., Y. Hussaini, A. Quarteroni and T. Zang, *Spectral Methods in Fluid Dynamics*, Springer Verlag, New York, 1988.

Caraballo, E., M. Samimy, J. Scott, S. Narayanan, and J. DeBonis, “Application of Proper Orthogonal Decomposition to a Supersonic Axisymmetric Jet,” *AIAA Journal*, Vol. 41, No. 5, pp. 866-877, May 2003.

Caraballo, E., and Samimy M., “Low Dimensional Modeling of Flow for Closed-Loop Flow Control,” *AIAA-2003-0059,* 2003b.

Caraballo, E., X. Yuan, P. Yan, M. Debiasi, A. Serrani, M. Samimy, and J. Myatt, “Feedback Control of Cavity Flow Using Experimental Based Reduced Order Model,” *AIAA Paper 2005-5269*, 35th AIAA Fluid Dynamics Conference and Exhibit, Toronto, ON, June 2005.

Chatterjee, A., “An Introduction to the Proper Orthogonal Decomposition,” *Current Science*, Vol. 78, No. 7, pp. 808-817, 2000.

Dano, B.P.E and J. A. Liburdy, *Three Dimensional Stereoscopic Imaging of a Pulsed Jet in Crossflow*, 11th International Symposium on Flow Visualization, Notre Dame, Indiana, August, 2004.

Fletcher, C. A. J., *Computational Techniques for Fluid Dynamics, Volume I,* Springer series in computational physics, Springer-Verlag, Heidelberg, 1988.

Heywood, J. G., R. Rannacher, and S. Turek, "Artificial Boundaries and Flux and Pressure Conditions for the Incompressible Navier–Stokes Equations," *Int. J. for Numerical Methods in Fluids,* Vol. 22, pp. 325–352, 1996.

Högberg, M., T. R. Bewley, and D. S. Henningson, "Linear Feedback Control and Estimation of Transition in Plane Channel Flow," *J. Fluid Mech.*, Vol. 481, No. 149, 2003.

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

Hotelling H., "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology,* Vol. 24, pp. 417-441, 1933.

Kim, J., "Control of turbulent boundary layers," *Phys. Fluids*, Vol. 15, No. (5), 2003.

Kosambi, D. D., "Statistics in function space," *J. Indian Math. Soc.*, Vol. 7, pp.76–88, 1946.

Kunisch, K. and S. Volkwein, "Control of Burger's equation by a reduced order approach using proper orthogonal decomposition," *J. Optim. Theory Appl.* Vol. 102, pp. 345-356, 1999.

Lumley, J., and P. Blossey, "Control of turbulence," *Annu. Rev. Fluid Mech.*, Vol 30, pp. 311-327, 1998.

Lumley, J. L., *The Structure of Inhomogeneous Turbulent Flows*, In: Yaglom AM, Tatarsky VI (eds) Proceedings of the international colloquium on the fine scale structure of the atmosphere and its influence on radio wave propagation, Moscow, Nauka, pp 166–178, 1967.

Narayanan, S., Barooah, P., and Cohen, JM, "Dynamics and Control of an Isolated Jet in Crossflow," AIAA. Journal, Vol. 41, No. 12, pp. 2316-2330, 2003.

Prabhu, R. D., S. S. Collis, and Y. Chang, "The Influence of Control on Proper Orthogonal Decomposition of Wall-Bounded Flows," *Physics of Fluids*, Vol. 13, pp. 520-537, 2001.

Ravindran, S.S, "Proper Orthogonal Decomposition in Optimal Control of Fluids," *NASA Technical Memorandum*, NASA TM 1999-209113, 1999.

Rowley, C.W., T. Colonius, and R.M. Murray, "Model Reduction for Compressible Flows using POD and Galerkin Projection," *Physica D*, Vol. 189, No.1-2, pp.115-129, Feb 2004.

Sirovich, L., "Turbulence and the Dynamics of Coherent Structure," *I. Coherent structures, Quart. Appl. Math.,* Vol. 45, No. 3, pp. 561–571, 1987.

Temam, R., *Navier-Stokes Equations and Nonlinear Functional Analysis*, CBMS Regional Conference Series, No. 41, SIAM, Philadelphia, 1983.