ABSTRACT

VALIDATING STEADY TURBULENT FLOW SIMULATIONS USING STOCHASTIC MODELS

by John Chabot

Proper Orthogonal Decomposition was heralded as an objective means of extracting coherent structures from turbulent flows. Prior to its introduction, coherent structures were subjectively defined and discussed among experts. Since its introduction many reduced order models have been developed with inconsistent and often flow dependent validation procedures. This work sets up a framework for a data driven approach to validation of reduced order models derived from steady turbulent flows. Here it is proposed that the 'goodness' of a model can be scored by how similar experimental and simulated data move through the model space. This is achieved by generating a Markov model for both data sets, using clustering techniques and maximum likelihood estimates. Results show increasing scores correlate with improved turbulent kinetic energy and modal amplitude for 3 data sets and 14 models. Additionally the generation of a surrogate Markov model can be used to identify missing dynamics in a simulation.

VALIDATING STEADY TURBULENT FLOW SIMULATIONS USING STOCHASTIC MODELS

A Thesis

Submitted to the

Faculty of Miami University

in partial fulfillment of

the requirements for the degree of

Master of Science

by

John Chabot

Miami University

Oxford, Ohio

2015

Advisor _____
Dr. Edgar Caraballo

Reader _____
Dr. Mehdi Safari

Reader _____
Dr. Andrew Sommers

Contents

List of Tables

List of Figures

Dedication

I would like to dedicate this work to my loving and supportive parents, who instilled in me a sense of wonder in the world.

Acknowledgements

I would like to thank my first acknowledge my advisor, Dr. Edgar Caraballo, for providing guidance and support in the research conducted here. I've learned and accomplish far more than I thought possible at the beginning of graduate school, and I believe that is largely to his credit.

I would also like to thank Dr. Jesse Little, for supplying experimental data for two of the experimentally collected data sets, while studying at the Ohio State Universities Gas Dynamics and Turbulence Laboratory and at the University of Arizona where he currently resides. I would also like to acknowledge his collaboration with Dr. Caraballo and I on our submission to AIAA's Aviation conference in June 2015.

Finally, I would like to acknowledge, several friends in the mathematics and statistics graduate schools for their help with understanding concepts and brainstorming solutions. These Include Paul Kristofferson, Hannah Hoganson, and Anna Payne in the mathematics department and Diana Eid in the statistics department. There were roadblocks during this research that were only overcome with their help, and I am very grateful.

Chapter 1

Introduction

The ability to predict the behavior of a turbulent flow in real-time has the potential to increase the efficiency of fluid dynamic devices when implemented in closed-loop control systems. The Navier-Stokes equations (NSE) have been shown to accurately predict fluid flow behavior for almost any fluid in any flow geometry. However, there are only eleven known solutions to the NSE, all of which are either special cases, where non-linear terms disappear, or have special symmetry that simplifies them [28]. For practical applications, scientists and engineers have relied on techniques developed through computational fluid dynamics, statistical models of the flow, or order of magnitude analysis to get an understanding of the underlying dynamics. A more recent approach in reduced order models, seeks to address the computational speed and accuracy of the previous mentioned methods. Reduced order models (ROMs) lower computational time and cost by simplifying the NSE from a set of nonlinear partial differential equations to a set of ordinary differential equations that can be solved more rapidly using well established numerical methods for ODEs. Recently, ROMs have been gaining popularity due to advances in data recording, increased computing speeds, and their reduced computational cost compared to direct numerical simulation of the NSE [16, 25]. Despite these advantages, ROMs typically have a narrow range of validity around their derived system characteristics such as Mach number, dynamic pressure, and Reynolds number [58].

While ROMs have been shown to provide a number of benefits over competing modeling methods, validation of such models is inconsistent throughout the community. Here, a data driven validation method is proposed, relating the dynamics of the experimental data to data derived from simulation, in a probabilistic sense. The inspiration for this validation procedure comes from the work of Kaiser et al. [31] who proposed a novel modeling scheme using the probable evolution of clustered flow states. In this work

a similar procedure is followed, where the methods of clustering are used to produce flow states which represent 'typical' flow formations, with each state distilled directly from the empirical data. Simulated flow data is then grouped to these clusters by identifying which cluster they are most similar to. Validation comes from identifying models that move through the model space, in similar patterns as the original empirical data. In order to give credibility to this procedure, a range of models and data sets are tested.

For this work a collection of low dimensional models that captures the important dynamics of three sets of experimental data; a mixing layer, a cavity flow, and flow over an airfoil, as well as a numerical generated axisymmetric jet are produced. The development of these models builds off the ROMs produced by Caraballo [11], and Sullivan [62] for the cavity and airfoil flows respectively. Reduced order models were generated for these flow conditions using experimental data by way of Proper Orthogonal Decomposition (POD), which generates basis functions that represent the most energetic features of the flow. Using a finite truncation of this basis, the NSE is projected by Galerkin Method, leaving a low dimension approximation of the flow dynamics of the original data. POD-Galerkin models and its many variants and correctors appears to comprise a significant portion of the models developed in the subfield of reduced order models.

Prior to the introduction of POD, identification of recurring features in turbulent flow, known as coherent structures, were found through a myriad of flow visualization techniques or criterion invented by the experiment operator [28]. The introduction of POD into the flow community by Lumley in 1967 [37] provided a repeatable and objective means of identifying these flow structures. The POD formulation ensures that the residual of projection onto that subspace is minimized [16]. The success of POD in the field of turbulence has spun off alternative methods such as the POD method of snapshots, extended POD, biorthogonal decomposition, balanced POD, and dynamic mode decomposition to name a few [2, 23, 51, 54, 57]. At their roots these methods decompose the original model space, into a reduced model space on which the flow dynamics are projected. While coherent structure can now be identified objectively, by decomposition of experimental or simulated data by one of the many previously mentioned methods, validation of the dynamic evolution of these ROMs is now similarly ambiguous. Because of the complexity of the underlying problem of validation, categorization of model accuracy is more of a debate between experts than objective measurement just as coherent structures were 40 years ago.

This leads back to the work of Kaiser et al. [31], who's clustering procedure was not dependent on a given decomposition method, but was instead feasible for any decomposition method. While POD was not directly used,

it was used to perform the clustering. POD decomposes the data using the fewest orthogonal dimensions, simplifying the clustering problem, and reducing the time requirements. Because clustering can be performed on any state vector, it seemed a logical candidate for producing a validation method that can perform a more direct comparison between models than is performed today.

The work of this thesis is laid out as follows. First in Chapter 2, a brief review of literature is presented with additional insights into the need for such a validation procedure. Next Chapter 3 presents theory behind the test models as well as the validation procedure itself. Chapters 4 and 5 present insights into the code implementation challenges and the empirical data. While these chapters are less important to the derivation and results of this work, they do provide important background information for the experimental results. Finally chapters 6 and 7 provide the evidence found in favor of the validation procedure and possible conclusion and further research avenues.

Chapter 2

Study Rationale

The overarching goal of this work is to develop an objective measure for the validation of a ROM's dynamic evolution, based on the dynamics present in the original data. While these models produce deterministic results, ROMs reduce the dimensionality of the original empirical data potentially by several orders of magnitude. This drastic reduction can only approximately resolve the original model space, leading naturally to some loss of accuracy. At some point, the accumulations of errors introduced by this approximation almost certainly causes the solutions to diverge from the raw data, even if they qualitatively produce similar flow evolution. Because turbulent fluctuations were classically approached as a stochastic process, it seems reasonable to look at model accuracy from this prospective [28].

Previous studies have looked at a variety of metrics in order to argue the accuracy of one particular model over another. One common metric for determining the accuracy of a model is the comparison of the mean and variance of the turbulent kinetic energy predicted by the model compared to its derived data. This is a natural step for models based on POD-Galerkin methods and their derivatives, as POD produces basis functions that optimally capture the kinetic energy of the flow. Östh *et al.* [44] followed this approach when comparing several proposed corrective viscous dissipation methods that attempt to keep standard POD-Galerkin models, that are notorious for finite time blow up, bounded [41]. A different study, proposed using an 'optimal rotation' of the POD basis, based on the predicted energy as it's optimization objective function [5]. In a slightly different approach, Gross & Fasel [24] looked at the phase and amplitude of only the two most energetic POD modes as a basis of comparison for models. Other studies have instead focused on key frequencies of the resulting system as a method of gauging accuracy. Caraballo *et al.* [11] looked at models which produced frequency peaks that closely matched empirically derived Rossiter frequencies

of a cavity flow. Another emerging method, Dynamic Mode Decomposition based on a Koopman operator, seeks to produce modes containing only one frequency, where frequency matching is the primary accuracy criteria [50, 54] . Others still attempt to calculate absolute error between reconstructed velocity fields and original data sets using appropriate norms [6, 8, 9]. These methods are computationally expensive over large time scales and indicate huge errors if the system comes out of phase with the original data. Rowley *et al.* [51] used a H_2 norm giving an exact error for the produced truncated model compared to the full model but this approach is only applicable to linear systems.

The review of literature did not uncover a generally agreed upon method of approach to validating a ROM. Each of the previous methods attempts to validate a model with one or two specifics criteria, or features directly tied to the decomposition method or the specific flow configuration. Because of the complexity of turbulent systems, selecting one or a few features to use in validation simply provides too coarse of sieve for acceptance that a model, in fact, reflects reality. It is of interest then to develop a data driven metric that holistically measures the dynamics of the flow, that is independent of modeling method or flow configuration. This again leads back to the study by Kaiser *et al.* [31] who proposed a clustering approach to flow modeling. They note that in the clustering process "the POD coefficient vector is only used for logistical convenience and is not necessary for the clustering algorithm". POD is also known by statisticians by its alternative alias, principle component analysis (PCA). PCA can be used to simply increase the speed and help denoise clustering results which the group of Kaiser *et al.* claimed as well. [30, 31]. So while decomposing using POD provides some advantages over other decomposition methods when clustered, clusters could be produced off of any decomposed data. It is this universalism of clustered states that leads to the development of the surrogate model produced in this work. This simplified, but representative model, can then be used to quickly invalidate models that mimic specific features of the empirical data but retain little else in resemblance to the real system.

Chapter 3

Theory

In this chapter a review of the relevant mathematical theory in this thesis is covered, included are both the reduced order models, as well as, the proposed measures. Building from the work of Sullivan [62], we utilize the widely deployed POD-Galerkin methods as our reduced order modeling scheme of choice with additional augmenting corrective methods. After developing the models that will be used for validation, the theory behind the Surrogate Markov Model itself is discussed.

3.1 Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD) was first introduced into the flow community by Lumley [37] which provides an objective and optimal means of extracting the largest and dynamically most important structure in a flow. POD detects these structures by identifying the orthogonal directions in a Hilbert space of squared integrable functions, (L_2) that minimizes the projection residual of the flow onto the finite dimension subspace spanned by these directions. Formally this can be written as follow:

$$\max_{\varphi \in \mathcal{H}} \frac{\langle |(u, \varphi)| \rangle}{(\varphi, \varphi)} \tag{3.1}$$

Here, u is the velocity field, φ is the candidate basis, $(\cdot, \cdot)$ represents the inner product on the Hilbert space $\mathcal{H}$ defined in Eq. 3.2, and $\langle \cdot \rangle$ is a suitable averaging operator. In the case of this work the averaging operator will be the ensemble average. Additionally restricting the Hilbert space to L_2, simply restricts that the functional space remains to those that can carry kinetic energy [7]. The L_2 inner product is defined as:

$$(u, \varphi) = \int_{\Omega} u(x)\varphi^*(x)dx \tag{3.2}$$

Where Ω is the integration domain and $*$ is the complex conjugate. Following a derivation presented in Holmes *et al.* [28], Eq. 3.1 can be transformed into the following eigenvalue problem:

$$\mathcal{R}\varphi = \langle(\varphi, u)u\rangle \tag{3.3}$$

$$\mathcal{R}\varphi = \lambda\varphi \tag{3.4}$$

where $\mathcal{R}$ is a linear operator.

This original POD definition was later modified by Sirovich [57] introducing the POD method of snapshots. This method is better suited for the needs of data with high spatial resolution that was made possible by new measurement equipment such as PIV. Sirovich noted that with a large enough ensemble of flow images, additional images could be reconstructed as a linear combination of prior images, leading to a new means of obtaining a spanning basis.

$$D = [u^1 \cdots u^M] \tag{3.5}$$

$$\frac{1}{M}D^T Da = \lambda a,\ \varphi = Da \tag{3.6}$$

With Eq. 3.6 representing the eigenvalue decomposition of a covariance matrix defined by "stacking" all M snapshots in Eq. 3.5. Using either Eq. 3.4 or Eq. 3.6 the original velocity field can be reconstructed exactly using an infinite sum of POD basis functions φ_i of energy λ_i, and modal amplitudes a_i.

$$u_i(x) = \lim_{n\to\infty} \sum_{i=1}^{n} a_i\varphi_i \tag{3.7}$$

Because the amount of information captured by each POD basis function is optimal, a decomposition retaining N terms can be used to approximately reconstruct the turbulent fluctuations optimally for N spatial basis functions [25].

$$u_i(x) \approx \sum_{i=1}^{n} a_i\varphi_i \tag{3.8}$$

The following identities are presented now to be used in later derivations.

$$\langle a_i \rangle = 0 \tag{3.9}$$

$$\langle a_i a_j \rangle = \Lambda_{ij}, \quad \Lambda_{ij} = \begin{cases} i = j & \Lambda_{ij} = \lambda_i \\ i \neq j & \Lambda_{ij} = 0 \end{cases} \tag{3.10}$$

While there are many means by which to produce a reduced model space, the POD method of snapshots fits the application for this work. In addition to the optimal reconstruction, advantages include reduced computation cost, simplicity of implementation and its documented effectiveness for data of high spatial resolution [4, 11, 45, 56, 61].

3.2 Galerkin Projection

While POD provides an optimal spatial basis to represent its original data, it does not provide a means of predicting how this basis will evolve in time. This can be accomplished by means of projecting the Navier-Stokes equations onto this basis using the Galerkin method [28]. This transforms the infinite dimensional NSE into a coupled set of non-linear ordinary differential equations. The experimental data sets that will be used in this work are all incompressible flows ($a < 0.3$) so the incompressible Navier-Stokes equations will be used. The numerically generated jet data lies in the compressible range but will be modeled using the incompressible equations. The thought is to investigate models outside of their intended parameter set. This could provide simulated data that predict some properties of the empirical data accurately, such as the turbulent kinetic energy, but have dynamics notably differing from the data. Presented below are the incompressible NSE used for dynamic modeling.

$$\nabla \cdot u = 0 \tag{3.11}$$

$$\frac{\partial u}{\partial t} = \nu \Delta u - (u \cdot \nabla) u - \frac{\nabla p}{\rho} - g \tag{3.12}$$

Where ν is the kinematic viscosity, p is the pressure field, ρ is the density field, and g represents body forces per unit mass. Reynolds decomposition is performed on the original flow with POD used on the resulting fluctuating component. The fluctuating component u' in Eq. 3.13 is "stacked' into the data matrix of Eq. 3.5 where POD is performed.

$$u = U + u' \rightarrow u = U + \sum_{i=1}^{\infty} a_i \varphi_i \tag{3.13}$$

As a simplification, an additional mean flow mode is defined as $\varphi_0 \coloneqq U$, $a_0 \coloneqq 1$ [41]. The evolution of each individual modal amplitude is determined by the current state of all modal amplitudes, linked by a set of coefficients that define the coupled interactions. These coefficients are found by replacing the flow field in the NSE with the generated POD basis functions and then taking the inner product with each POD basis function.

$$\Delta u \rightarrow (\Delta \varphi_i, \varphi_j) = l_{ij} \tag{3.14}$$

$$(u \cdot \nabla) u \rightarrow ((\varphi_i \cdot \nabla)\varphi_j, \varphi_k) = q_{ijk} \tag{3.15}$$

The subscripts indicate basis functions i, j and k with equations written in tensor notations. Coefficients l and q are the low dimensional projections of the viscous and convective terms from the NSE. Body forces are typically assumed to be negligible, if only gravity is considered, while the pressure term vanishes identically if Dirichlet boundary conditions are present. For other boundary conditions the pressure term only provides small alterations to the behavior of the system. [41]. Together the evolution of the modal amplitude can be described as such:

$$\dot{a} = \nu \sum_{i=0}^{n} l_{ij} a_j + \sum_{i=0}^{n} q_{ijk} a_j a_k \tag{3.16}$$

Often this form is transformed in order to isolate, constant, linear and quadratic interactions; C_i, L_{ij}, and Q_{ijk} respectively. Here Eq. 3.16 is presented in this fashion:

$$\dot{a} = C_i + \sum_{j=1}^{n} L_{ij} a_j + \sum_{j,k=1}^{n} Q_{ijk} a_j a_k \tag{3.17}$$

$$C_i = \nu l_{i0} + q_{i00} \tag{3.18a}$$

$$L_{ij} = \nu l_{ij} + q_{ij0} + q_{i0k} \tag{3.18b}$$

$$Q_{ijk} = q_{ijk}, \quad i, j, k = 1, \ldots, n \tag{3.18c}$$

In addition to the standard Galerkin expansion, a weak formulation of the NSE is utilized by replacing the viscous term's Laplacian operator with a set of first order gradient terms by way of Green's Identity [12]. By substituting

a second order term with a set of first order terms, the solutions have the potential to be improved for the empirical data [41].

$$\int_{\Omega}(\phi\Delta\varphi + \nabla\varphi\cdot\nabla\phi)dV = \oint_{\omega}\phi\nabla\varphi\cdot dS \tag{3.19}$$

Rearranging:

$$\int_{\Omega}\phi\Delta\varphi dV = -\int_{\Omega}\nabla\varphi\cdot\nabla\phi dV + \oint_{\omega}\phi\nabla\varphi\cdot dS \tag{3.20}$$

Using Eq. 3.20 the viscous term can alternatively be calculated as:

$$l_{ij}^{w} = -(\nabla\varphi_k, \nabla\varphi_i) + \oint_{\omega}\varphi_i\nabla\varphi_k\cdot dS \tag{3.21}$$

Where ω is the domain of the free flow boundary for the surface integral. This formulation is the preferred means of determining the viscous term for a number of authors. For the remainder of this thesis l_{ij}^{w} can be substituted anywhere l_{ij} is used.

3.3 Corrector Methods

In turbulent flows, energy typically cascades from the mean flow down structures of decreasing length scales until it is eventually dissipated as heat at the smaller scales [43]. With the goal of reducing the model space to very few dimensions, the smallest scale structures are typically ignored. This truncates the energy cascade leading to over prediction of the model energy. In order to produce bounded models that at least qualitatively predict the dynamics of the flow, two classes of correction methods are utilized, sub-scale turbulent models in the form of eddy-viscosity and a optimized basis transformation method.

3.3.1 Viscous Dissipation

Eddy-viscosity models attempt to bound the solutions by adding additional viscosity to the POD-Galerkin model at the global or modal level. This viscosity attempts to mimic the total energy transfer to the truncated POD basis functions. Three eddy-viscosity models, in addition to a scaling factor are tested in this work. While each method differs in its approach, the solution forms follow that of Östh *et al.* [44] where the total turbulent kinetic energy is used to form a closure problem to solve for each term. Therefore it will be useful to define how energy will be calculated.

$$K(t) = \rho\frac{1}{2}\int_{\Omega}|u'(x,t)|^2dx \propto \frac{1}{2}\int_{\Omega}|u'(x,t)|^2dx \tag{3.22}$$

Assuming the POD basis functions have been normalized the modal energy contribution becomes:

$$K_i(t) = \rho\int_{\Omega}|a_i(t)\varphi_i(x)|^2dx \propto \frac{a_i^2(t)}{2} \tag{3.23}$$

One of the first corrective methods to the base POD-Galerkin model was introduced by Aubry *et al.* [3] who first proposed the inclusion of an eddy-viscosity term. Kraichnan and Chen noted that for a wide class of flow configurations that the quadratic terms are energy preserving [32]. Based on this Aubry *et al.* presented the following ansatz that can be readily transformed back into the form of Eq. 3.17.

$$\dot{a}_i = (\nu + \nu^T)\sum_{j=1}^{n} l_{ij}a_j + \sum_{j,k=1}^{n} q_{ijk}a_ja_k \tag{3.24}$$

The idea of an eddy-visocity was later extended by Rempfer and Fasel [46] by postulating that dissipation was a scale dependent phenomena.

$$\dot{a}_i = (\nu + \nu_i^T)\sum_{j=1}^{n} l_{ij}a_j + \sum_{j,k=1}^{n} q_{ijk}a_ja_k \tag{3.25}$$

Solutions for ν^T and ν_i^T can be found by solving for the energy balance averaged over the ensemble. The change of energy is found as:

$$\sum_{i=0}^{n}\frac{d}{dt}K_i(t) = \sum_{i=0}^{n}\frac{d}{dt}\frac{a_i^2}{2} = \sum_{i=0}^{n}a_i\dot{a}_i \tag{3.26}$$

averaged over the whole ensemble.

$$\sum_{i=0}^{n}\langle\frac{d}{dt}K_i(t)\rangle = \sum_{i=0}^{n}\langle\frac{d}{dt}\frac{a_i^2}{2}\rangle = \sum_{i=0}^{n}\langle a_i\dot{a}_i\rangle \tag{3.27}$$

Substituting Eq. 3.25 into Eq. 3.27, using the identity from Eq. 3.10 and rearranging terms, a solution to ν_i^T is found.

$$\nu_i^T = -\left(\nu + \frac{\sum_{j,k=0}^{n} q_{ijk}\langle a_ia_ja_k\rangle}{l_{ii}\lambda i}\right) \tag{3.28}$$

A globally averaged version of Eq. 3.28 can be found by simply summing the terms in the fraction for all basis functions included in the POD-Galerkin model.

In addition to the eddy-viscosity terms derived from an averaged energy balance, values are also found via a least squares solution proposed by Couplet [18].

$$\tilde{\nu}_i^T = \frac{\langle a_i^2 r^<(A) d^>(A) \rangle}{\langle a_i^2 (d^<(A))^2 \rangle} \tag{3.29a}$$

$$r^<(A) = \sum_{j=n+1}^{c} l_{ij} a_j + \sum_{k=n+1}^{c} \sum_{j=0}^{n} q_{ijk} a_j a_k \tag{3.29b}$$

$$d^>(A) = \sum_{j=0}^{n} l_{ij} a_j \tag{3.29c}$$

$$d^<(A) = \sum_{j=n+1}^{c} l_{ij} a_j \tag{3.29d}$$

Here terms with the superscript $>$ represents terms that are included in the model, while $<$ represents neglected terms up to c, the number of modes needed to account for 99% of the flow energy.

The three eddy-viscosity terms presented thus far are all linear correctors. Noack *et al.* [41] noted that the previously mentioned linear eddy viscosity models attempt to model neglected linear and *nonlinear* interactions. In a more recent effort to produce bounded solutions to a larger class of POD-Galerkin models, a non-linear scaling factor was proposed [17] and justified [44] to the previously discussed eddy-viscosity terms. To begin, a new state dependent eddy viscosity model is equated to the neglected higher order interactions in a similar fashion as Eq. 3.28 and 3.29a.

$$\nu^T(a) \sum_{j=1}^{n} l_{ij} a_j = \nu \sum_{j=n+1}^{\infty} l_{ij} a_j + \sum_{j,k=1}^{\infty} q_{ijk} a_j a_k, \quad max(j,k) > n \tag{3.30}$$

Again multiplying by a_i ensemble averaging $\langle \cdot \rangle$ and utilizing Identities 3.9 and 3.10 the follow equality is found.

$$\nu^T(a) l_{ii} \lambda_i = \sum_{j,k=0}^{\infty} q_{ijk} \langle a_i a_j a_k \rangle \tag{3.31}$$

Östh *et al.* [43, 44] utilized a statistical closure representing the intermodal energy transfer due to the convection term as a means of justifying the

scaling introduced by Cordier *et al.* [17]. Rearranging terms in this closure, an approximate scaling factor was produced based on the system's current turbulent kinetic energy.

$$K_\Sigma(t) = \sum_{i=1}^{n} a_i^2/2, \quad K_\Sigma = \langle K_\Sigma(t) \rangle \tag{3.32}$$

With the state dependent eddy viscosity taking the following form.

$$\nu^T(a) = \nu^T \sqrt{\frac{K_\Sigma(t)}{K_\Sigma}} \tag{3.33}$$

Justification for this square root relation are somewhat lengthy and therefore are omitted. The interested reader should refer to [44] and [41] for full details.

3.3.2 Basis Transformation

The final correction method utilized in this work is a basis transformation method, introduced by Balajewicz *et al.* [5]. The method generates a new optimal spanning basis from a larger spanning basis derived from a traditional POD-Galerkin model. Similar to the nonlinear eddy viscosity model, this method attempts to also address the neglected nonlinear inter-modal energy transfer. Instead of adding an additional term to the Galerkin system, to balance the model energy, this method attempts to produce a basis such that the total neglected inter-modal transfer is minimized towards zero.

In order to determine an optimal basis, average changes of energy must be determined in a similar fashion as Eq. 3.31 by multiplying the system by a_i and taking the ensemble average $\langle \cdot \rangle$. Here the system is now represented in the form of Eq. 3.17 resulting in:

$$2\dot{K}_\Sigma(t) = \sum_{i=1}^{n} C_i a_i + \sum_{i=1}^{n}\sum_{j=1}^{\infty} L_{ij} a_i a_j + \sum_{i=1}^{n}\sum_{j,k=1}^{\infty} Q_{ijk} a_i a_j a_k \tag{3.34}$$

Simplifying:

$$0 = \langle 2\dot{K}_\Sigma(t) \rangle = \sum_{i,j=1}^{n} L_{ij}\Lambda_{ii} + T^<, \quad T^< = \sum_{i=1}^{n}\sum_{j,k=1}^{\infty} Q_{ijk}\langle a_i a_j a_k \rangle \tag{3.35}$$

Using this equality a new spatial basis is generated such that the inter-modal energy transfer term $T^<$ vanishes. This new basis is 'minimally rotated' away from the original basis of size N by a transformation matrix $X \in \mathbb{R}^{N\times n}$ to a new basis of size n where $N > n$. This produces a new set of Galerkin coefficients and modal amplitudes.

$$\tilde{\varphi} = \sum_{j=1}^{N} X_{ij}\varphi_j, \quad \tilde{a}_i = \sum_{j=1}^{N} X_{ji}a_j \tag{3.36a}$$

$$\tilde{L} = X^T L X, \quad \tilde{C} = X^T C, \quad \tilde{\Lambda} = X^T \Lambda X, \tag{3.36b}$$

$$\tilde{Q}_i jk = \sum_{p,q,r=1}^{N} X_{pi} Q_{pqr} X_{qj} X_{rk} \quad i,j,k = 1,\cdots,n \tag{3.36c}$$

Balajewicz *et al.* [5] remarked that inter-modal energy flows predicted by Galerkin systems are often under-predicted . In response a free transfer term parameter ϵ was introduced to account for these discrepancies:

$$\epsilon = \sum_{ij}^{n} L_{ij}\Lambda_{ij} \tag{3.37}$$

$$r(\epsilon) := \sum_{i=1}^{n} \tilde{\Lambda}_{ii} - \langle \sum_{i=1}^{n} \tilde{a}_i^2(t) \rangle \tag{3.38}$$

A root finding procedure is implemented on a function of ϵ seen in Eq. 3.38 in order to determine the new basis. Full details of the algorithm have been omitted for brevity, see Balajewicz *et al.* [5] or Appendix A for more details.

Together these methods constitute a large spectrum of potential models to be tested. With the exception of the nonlinear eddy viscosity model and the basis transformation method, these methods are tried in combination, as well as isolation, totaling 28 models. A crucial assumption of the basis transformation method is that because these bases were minimally rotated, the properties of orthogonal basis were approximately preserved. This assumption, was not rigorously proven and because these bases were no longer orthogonal, the error in system energy was uncertain leading to separation of the two methods. Table 3.1 below provides the description of the models used.

Table 3.1: Listing of the models that are utilized in this thesis. Subscripts W, N, and T represent models also using weak formulation, nonlinear eddy viscosity, and the basis transformation

Method Abbreviation	Model Description
$GM_{(W,T)}$	Base POD-Galerkin Model
$GM1_{(W,N,T)}$	Averaged global eddy viscosity [44]
$GM2_{(W,N,T)}$	Averaged modal eddy viscosity [42]
$GM3_{(W,N,T)}$	Least squares modal eddy viscosity [18]

3.4 Surrogate Markov Models

The previously derived models, only represent a small sample of the proposed models for ROMs of turbulent flows. Each introduced method of modeling claims some aspect of improved agreement to the empirical data over competing models. The points of comparison in these studies is not consistent in the literature, making model selection for a practical application difficult. It was found using the model set shown in Table 3.1 that between a forced and unforced mixing layer, different model-corrector combinations performed better between the two flow configurations [14]. It therefore appears unlikely one modeling procedure, will always produce the best agreement for all flow configurations. For practical application, it is plausible that several methods should be tested to find the model with best agreement to the empirical data, requiring a universally applicable approach to validation.

The process of validating a ROM without introducing some form of subjectivity into the process is again the overarching goal of this work. Cluster-based reduced order modeling provides the initial framework for this procedure, which models a turbulent system using the most probable evolution of flow states. The model space is compressed from a POD decomposition, representing a finite set of dimensions, to a finite number of states via clustering [9, 31]. Once compressed, a Markov model is derived and statistics from the empirical system's evolution can be derived. This idea is extended for validation, by generating an additional Markov model representing the simulation. A simulated model's solution trajectory can be classified to one of the empirical data's clustered states at each time step, by finding which cluster it is most similar to. At this point, this new sequence of states can again be used to produce a Markov model. It is proposed that these simpler Markov models be used to compare empirical and simulated data indirectly. Therefore we call them surrogate Markov models (SMM). This section will provide some background on the basics of Markov models and the means by

which they shall be compared.

3.4.1 Markov Chains and Stochastic Matrices

While turbulence is deterministic; it was historically, and in many simplifying models treated as a stochastic process [28]. While modeling the system this way may not be applicable for prediction, it may prove useful for comparison. Even viewed in the context of a deterministic system, turbulence shows some connection to chaotic dynamics, characterized by sensitivity to initial conditions [60]. Ruelle and Takens [52] have also shown mathematically that for dissipative systems, which turbulence falls into, a transition from ordered to chaotic solutions can be dictated by a parameter μ, reflective of a flows transition based on its Reynolds number. Additionally the distinction between chaotic and stochastic systems in many ways is not as well defined as may first appear. Consider the classic real world analog for a random variable, the dice roll or coin flip. Given exact initial conditions and a controlled environment the outcome could be predicted prior to any flip or roll. The key to assumption for these simple systems is that exact conditions are not known and given the duration of the flip or roll, the sensitivity to initial conditions allows this uncertainty to be magnified. This same line of logic is applied to the turbulent system. During the clustering process exact initial conditions are forfeit, and data that is time uncorrelated in the case of the three experimental data sets allows this uncertainty to be amplified. With these arguments the author believes the assumption that a steady flow can be modeled as a stochastic process is justified. Following the lead of Kaiser et. al [31] this is taken a step further and it is assumed that a turbulent system can be well approximated as a Markov or memoryless process. Here the memoryless property is implied for systems with unique solutions, in which case only the current state of the system is needed to predict the future. For steady 2D systems it has been proven that for sufficiently smooth initial and boundary conditions the NSE produces unique solutions [34]. Kaiser *et al.* [31] suggest that 3D flows can generally be assumed unique in an adequate numerical Navier-Stokes discretization at least to a prediction horizon of interest. It was also pointed out by Kaiser *et al.* [31] that the discretization of the flow field by POD preserves unique solutions under Galerkin projection [22]. With evidence that a clustered turbulent system can at least be represented approximately as a Markov process, a formal definition of a Markov chain is given:

Markov Chain [55] 1. A stochastic process $X = \{X_t, \in \mathbb{N}\}$ on a state space S is a discrete-time Markov chain if:

-for all $t \geq 0$ $X_t \in S$
-for all $t \geq 1$ and for all $i_0, \ldots, i_{t-1}, i_t \in S$ we have:
$P\{X_t = i_t | X_{t-1} = i_{t-1}, \ldots, X_0 = i_0\} = P\{X_t = i_t | X_{t-1} = i_{t-1}\}$

The primary characteristics of a Markov chain is that the probability of transitioning to the next state i_t is only conditionally dependent on the current state of the system i_{t-1}, and not states prior. Also note that the sum of $P\{X_t = i_t | X_{t-1} = i_{t-1}\}$ for all i_t for a given i_{t-1} is identically equal to 1. With this definition in place it can be seen that only the current state i_{t-1} and the next state i_t are needed to describe the probability of all transitions possible in the system. This can neatly be represented by what's known as a stochastic matrix or a transition matrix.

Stochastic Matrix [15] 2. A stochastic matrix is a matrix containing elements p_{ij}, the transition probabilities, with $i, j \in S$ at times $t-1$ and t is:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix} \tag{3.39}$$

Typically the stochastic matrix will be unknown and will have to be estimated by observing one or many Markov chains. This can be accomplished by using a maximum likelihood estimate (MLE) which, as its name implies estimates a parameter or parameters θ by $\hat{\theta}$ based on likelihood of observing such a sequence given the parameter(s). Here, the equation for likelihood function and MLE of the stochastic matrix are presented , with a derivation of the MLE provided in Appendix B

$$L(p, \pi | X) = \prod_{i=1}^{n} \pi_i^{n_i^1} \prod_{i,j=1}^{n} p_{ij}^{n_{ij}} \tag{3.40}$$

$$\hat{p}_{ij} = \frac{n_{ij}}{\sum_{j=1}^{n} n_{ij}} \tag{3.41}$$

Here n_{ij} represents the observed number of transitions from state i into state j, π_i is known as the stationary distribution and n_i^1 is the number of starts in position i. Again full descriptions of these terms are provided in Appendix B. With this there is now a means of describing a discrete process only with probabilities. Some thought was given to attempt to represent the system in terms of a hidden Markov model, allowing estimation of a stochastic matrix from the state vector directly, but it is suspected that amount of data in each data set described in Chapter 5 is simply inadequate to estimate the

transition probabilities accurately. In addition it was unclear how another data set, in our case the simulated model, could be classified to the same set of states. Because of this, our ROM's state vector must be mapped from $\mathbb{R}^n :\rightarrow \mathbb{N}$ which will be described in the next section.

3.4.2 Clustering and Classification

The key challenge with producing a map from the system's state vector in $\mathbb{R}^n$ to a discrete point in $\mathbb{N}$ is to do so without prior knowledge of the system. For this problem the methods of machine learning or more specifically, unsupervised learning, a branch of machine learning, are used [26]. An important task in unsupervised learning is giving data, structure, without information outside of the data itself. This could broadly be used as a description for the clustering problem. While it seems obvious to us as humans on what is or isn't a cluster in a data set, we are pattern recognition machines and really carry many definitions on what constitutes a pattern. Likewise, in the clustering problem there are many definitions with pros and cons.

This work will focus on two methods of clustering, the k-means algorithm and clustering based on the posterior probability of a fitted Gaussian mixture model. The k-means algorithm was the method used by Kaiser et. al [31] to segment the data in k clusters [38]. This method produces 'hard' segmentation where each element is exclusively the member of one cluster. Clustering based on a Gaussian mixture model (GMM) was included because POD basis functions generated from Reynolds decomposed flow snapshots closely approximates a normal or Gaussian distribution where $a_i \approx N(0, \lambda_i)$. [28]. clustering from a GMM can be performed with 'hard' segmentation by assigning each data point to the Gaussian component contributing the largest posterior probability of all components. Here an informal description will be given for each.

K-means can be thought of as a heuristic for finding global minima to the minimization problem of the following form [30].

$$\min_{C_1,\ldots,C_k} \left\{ \sum_{k=1}^{K} W(C_k) \right\} \tag{3.42}$$

Where C_k are the K clusters and W is a measure of the difference between elements within the same cluster. The most common way to determine this difference is by Euclidean squared distance to the cluster centroid which this work will follow.

$$\min_{C_1,\ldots,C_k} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} (x_i - c_k)^2 \right\} \tag{3.43}$$

Here n_k is the number of elements and c_k is the centroid of cluster C_k. While k-means is only guaranteed to converge to local minima of Eq. 3.42 it is found in practice to often converge to global minima if the data is well spaced [29, 40]. A high level overview of the heuristic is outlined below in Algorithm 1.

Algorithm 1: k-means algorithm for finding cluster centroids

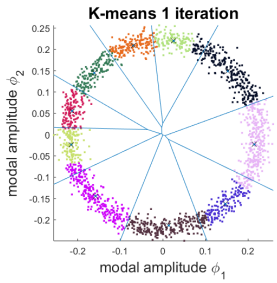
Input : array x on $\mathbb{R}^n$ and integer k
Output: integer membership array y and k centroids on $\mathbb{R}^n$

function: k-means (x, k) ;
$c_k \leftarrow \text{initialize}(k)$;
$y \leftarrow \text{assign}(x, c_k)$;
$w_{\text{new}} \leftarrow \text{within}(x, c_k)$;
$w_{\text{old}} \leftarrow \infty$;
while $(w_{old} - w_{new})/w_{old} \geq tolerance$ **do**
    $c_k \leftarrow \text{average}(y)$;
    $y \leftarrow \text{assign}(x, c_k)$;
    $w_{\text{old}} \leftarrow w_{\text{new}}$;
    $w_{\text{new}} \leftarrow \text{within}(x, c_k)$;
end

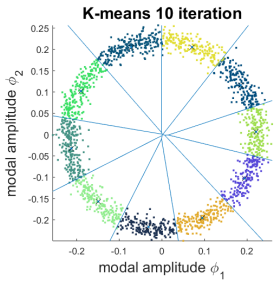
Figure 3.1 shows k-means after 1, 10 and 100 iterations, for a structured data set representing the forced mixing layer and a relatively unstructured data from the baseline mixing layer. This depicts how k-means converges to a local or global minimum for the clustering problem.

Once the data from the POD-Galerkin model has been grouped into k clusters, simulated data from one of the POD-Galerkin models in Table 3.1 can be classified to one of these clusters by finding its nearest neighbor centroid.

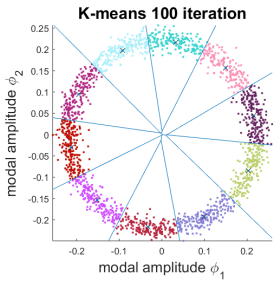
Clustering based on Gaussian mixture models is accomplished by the widely applicable EM algorithm. Similar to k-means it can be thought of as heuristic for solving for local optima of a complex likelihood function. Unlike the MLE of the stochastic matrix presented above in Eq. 3.41, where an analytical solution is available, closed-form solution are the exception for most MLE [35]. The EM algorithm is guaranteed to iteratively move towards a local or global maximum by repeatably applying an expectation step (E)



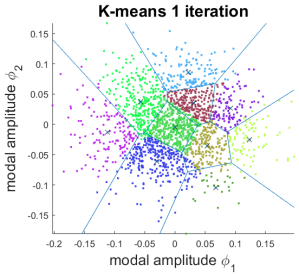
(a) 1 iteration of k-means: organized flow.



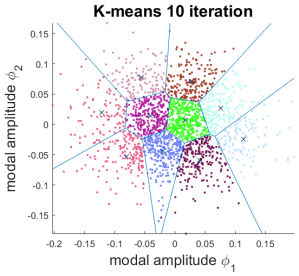
(b) 10 iteration of k-means: organized flow.



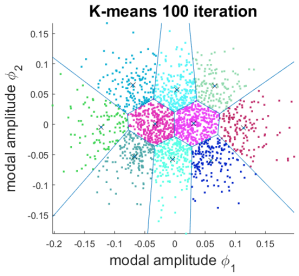
(c) 100 iteration of k-means: organized flow.



(d) 1 iteration of k-means: unorganized flow.



(e) 10 iteration of k-means: unorganized flow.



(f) 100 iteration of k-means: unorganized flow.

Figure 3.1: Different iterations of k-means for structured (forced) and unstructured (baseline) mixing layer data.

followed by a maximization step (M). The EM algorithm assumes there is some unknown complete data set X which must be estimated with observed data Y [35]. The case of a Gaussian mixture model it is assumed that X is derived from a mixture of K Gaussian components.

In the expectation (E) step, the conditional expected value is calculated for θ from the observed elements of Y and the current estimate of Gaussian components $\hat{\theta}$ as shown below:

$$Q(\theta|\hat{\theta}_i) = E\Big(\ln\Big(f(X|\theta)\Big) \;\Big|\; Y = y, \hat{\theta}_i\Big) \tag{3.44}$$

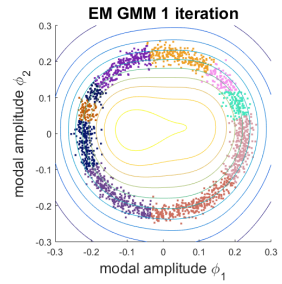
Where $f(X|\theta)$ is the conditional probability density function of the Gaussian components. Once the conditional expected value is calculated, the parameter θ is maximized with respect to the likelihood function in the maximization (M) step by either close form solution or using an iterative method such as gradient descent. This maximized value of θ becomes the new estimate $\hat{\theta}_{i+1}$ in the following expectation step [35]. Once the EM algorithm has converged to a local or global solution of the MLE, points can be assigned to the cluster based on which Gaussian component provided the largest posterior probability. As with k-means, a high-level overview of how clusters are determined is provided.

Algorithm 2: GMM clustering algorithm for finding Gaussian mixture components

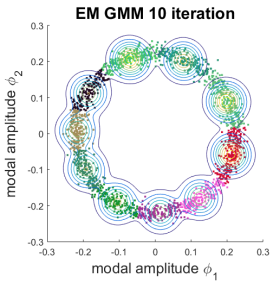
Input : array x on $\mathbb{R}^n$ and integer k
Output: integer membership array y and k Gaussian mixture components θ on $\mathbb{R}^n$

```
function: GMM cluster (x, k) ;
θ̂ ← ∞ ;
θ ← initialize(k);
θ ← gauss_maximize(x, θ);
while (θ − θ̂)/θ̂ ≥ tolerance do
    θ̂ ← θ;
    θ ← gauss_expectation(x, θ̂);
    θ ← gauss_maximize(x, θ̂);
end
y ←posterior_probability(x, θ)
```

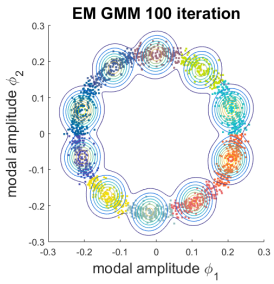
Figure 3.2 shows clustering based on GMM after 1, 10 and 100 iterations



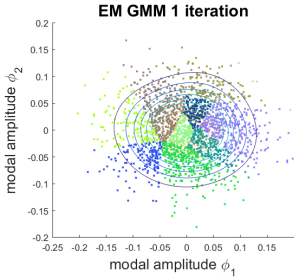
(a) 1 iteration of EM GMM: organized flow.



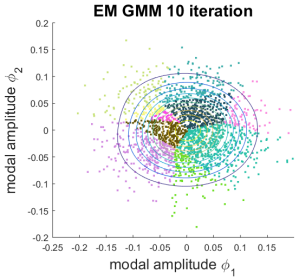
(b) 10 iteration of EM GMM: organized flow.



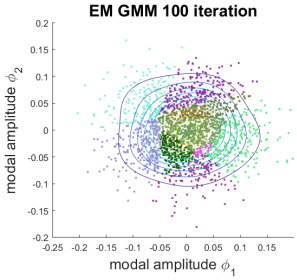
(c) 100 iteration of EM GMM: organized flow.



(d) 1 iteration of EM GMM: unorganized flow.



(e) 10 iteration of EM GMM: unorganized flow.



(f) 100 iteration of EM GMM: unorganized flow.

Figure 3.2: Different iterations of EM GMM for structured (forced) and unstructured (baseline) mixing layer data. Contour lines indicate the probability density function of the fitted Gaussian components.

of the EM algorithm. Data is again shown for a structured data set representing the forced mixing layer and a relatively unstructured data set from the baseline mixing layer. This depicts how the EM converged to a local or global maximum of the likelihood function.

In order to classify the simulated data to this fitted GMM, data is again assigned by finding which component has the highest posterior probability.

The last issue to deal with, when classifying the simulated data is what to do when the solution is very far from the region defined by the empirical data. If the simulated trajectory moves sufficiently far from all of the k-means centroids or the posterior probability, becomes sufficiently small when clustering with GMM, it no longer make sense to assign this predicted point to one of the valid clusters. Therefore, an additional outlier state is defined outside the region occupied by the empirical data. Models which predict solutions that move sufficiently far from the region of phase space occupied by the empirical data, are penalized by predicting some probability of transitioning to this outlier state. Because the probability of transition from the current state i

to another state j must sum to one, predicting any probability of transition to this erroneous state will scale down the probabilities of transitioning to a valid state. In order to reduce tuning of this value, the distance that defines the boundary between valid states and the outlier state, is scaled based on the properties of the generated clusters. For k-means this is a multiple of the largest distance between cluster centroid. For GMM clustering, this taken as a fixed Mahalanobis distance which is a generalization of standard deviation shown in Eq. 3.45.

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{3.45}$$

Here, x is a vector μ is the mean of a joint distribution, and S is the covariance matrix.

3.4.3 Measures

Now that methods have been established on how to estimate the stochastic matrix of both empirical and simulated SMMs, scoring method can now be developed. One approach will make a direct comparison of the estimated stochastic matrices between the empirical and simulated data. The other approach will look at the likelihood that the simulated data was derived from the stochastic matrix that describes the empirical data.

The first proposed approach generates an estimate for the transition matrices of both the empirical data $\hat{P}^e$, as well as the simulated data $\hat{P}^s$. From this, a direct comparison of the stochastic matrix structure can be performed by using the matrix analog to Euclidean distance, the Frobenius norm.

$$\sigma_d = \sqrt{\sum_{i,j=1}^{n} (\hat{p}^e_{ij} - \hat{p}^s_{ij})^2} \quad \hat{p}^e_{ij} \in \hat{P}^e, \quad \hat{p}^s_{ij} \in \hat{P}^s \tag{3.46}$$

Some basic insight into this shows that for $\sigma_d = 0 \rightarrow \hat{P}^e = \hat{P}^s$ and the worst possible score of $\sigma_d = 2n$ by noting each row must sum to 1 for each stochastic matrix.

The second proposed approach looks at the measure known as the relative likelihood function, to attempt to quantify the model accuracy. The relative likelihood function is defined as the ratio between the likelihood of an estimate $\tilde{\theta}$ and the MLE $\hat{\theta}$ for some random variable X [27]. Here the MLE of the simulated data is taken as the alternative model and then compared back to the MLE predicted by the empirical data. This measure for accuracy then becomes:

$$\sigma_l = \frac{L(P^s|X)}{L(P^e|X)} = \frac{P\{X_1 = j_1\}\prod_{i,j=1}^{n} {p^s}_{ij}^{n_{ij}}}{P\{X_1 = j_1\}\prod_{i,j=1}^{n} {p^e}_{ij}^{n_{ij}}} \quad \hat{p}^e_{ij} \in \hat{P}^e, \quad \hat{p}^s_{ij} \in \hat{P}^s \tag{3.47}$$

Repeated products of p_{ij} when $p_{ij} \leq 1$ in excess of 1000 times can quickly run into the finite precision of floating point arithmetic. In order to avoid this Eq. 3.47 is transformed into the log domain were floating point precision is again adequate. In addition, the negative of this value is taken so that higher scores indicate worse agreement for both σ_d and σ_l.

$$\sigma_l = -\left(\sum_{i,j=1}^{n} n_{ij} \ln(p^s_{ij}) - \sum_{i,j=1}^{n} n_{ij} \ln(p^g_{ij}) \right) \tag{3.48}$$

One caveat of this method is the possibility of the simulated model predicting a zero probability of a particular transition, which may occur in the empirical data. In order to get a score for models even when a given transition is not predicted, each zero probability element of the stochastic matrix of the simulated data is given a small probability (0.001), with the remaining elements in that row scaled proportionally to ensure that the probability of all transitions from a given state when summed, is still identically 1

In total this gives four means of measuring each model detailed in Table 3.1. Ideally, this will be reduced to the method found to universally be most effective, at measuring the 'goodness' of the tested models.

Chapter 4

Code Implementation

A secondary goal of this work was to develop a high performance, data abstract, computational pipeline that attempts to minimize additional development time for new data sets. In addition to the complexity of the modeling methods described in Chapter 3, some challenges emerged while implementing the theory. These were the result of attempts to keep the code abstract and minimize assumptions about the data. The two notable challenges were calculating an approximate volume at each mesh vertex, as well as, dealing with numerical integration without prior knowledge of boundaries present in the flow. Finally, a significant effort was made to optimize and parallelize the codebase in order to efficiently run through the large number of test cases, needed to validate the proposed measure.

First, a high level overview of the code is presented. Overall the code is divided into four primary segments, the preprocessing and POD basis function generation, calculation of the Galerkin coefficients and time integration, the basis transformation method, and the scoring method. A condensed summary of the preprocessing and POD generation code is given first.

Algorithm 3: POD_Gen (generate a POD basis given a set of raw data)

Input : A set of configurations $flags$ and a test $directory$
Output: A file containing information about the POD basis functions

```
1  function: POD_Gen (flags, directory) ;
2  [u, x] ← read_data(directory);
3  [u, x] ← non_dimensionalize(u, x, flags);
4  [u, x] ← image_flip(u, x, flags);
5  [u, U] ← reynold_decomposition(u);
6  bnds ← detect_bounds(U);
7  bnds ← gui_bounds(bnds, U);
8  vol ← calculate_volume(x, bnds);
9  cov ← calculate_covariance(u, vol);
10 [pod_u, λ, a] ← pod(cov, u);
11 foreach pod_u do
12 |  plot(pod_u);
13 end
14 save(u, U, x, bound, volume, pod_u, λ, a, clusters)
```

Here the POD_Gen code is a essentially a set of sequential steps to prepare the data for Proper Orthogonal Decomposition. The generated POD basis functions will then be used in the latter three stages of computation. Here, all the preprocessing such as correctly orienting and non-dimensionalizing the data is performed, based on flags passed at startup. Additional data is derived from the raw data, such as an approximation of volume represented by each mesh vertex and information about the location of solid and open boundaries. A nearly direct translation of the theory in Section 3.1, is used to generate the POD basis functions.

Next, a high level overview of the Galerkin coefficient generation and model simulation is given:

Algorithm 4: Galerkin_Project (generate and simulate ROMs given a set of POD basis functions)

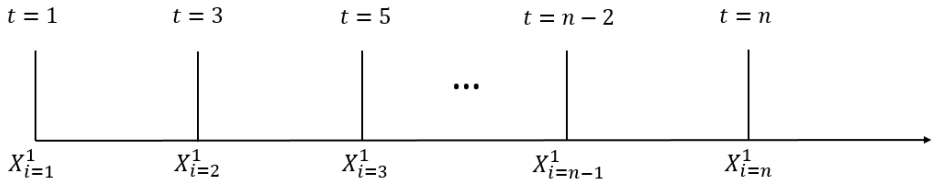
Input : A set of configurations $flags$ and a test $directory$
Output: A file containing each models' coefficents and simulation results

```
function: Galerkin_Project (flags, directory) ;
[x, bnds, vol, pod_u λ, a] ← load_pod(directory);
mode_set ← mode(flags);
foreach mode_set do
    [l, q] ← galerkin_coefficients(pod_u, vol, bnds);
    ν ← eddy_viscosity(l, q, a, λ);
    [a_s, t_s] ← ode_solve(l, q, ν);
    plot(a_s, t_s);
    save(a_s, t_s, l, q);
end
```

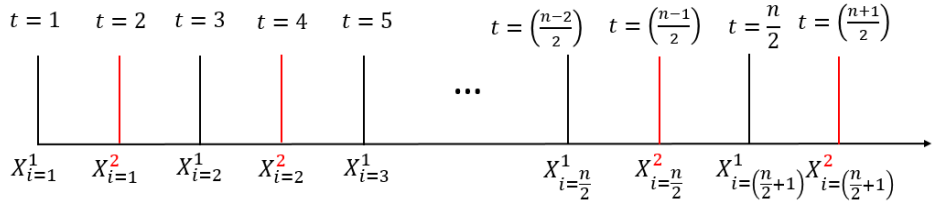
The Galerkin_Project code will load the results saved by POD_Gen and proceed to generate and simulate the produced models. In order to help minimize the amount of time loading data, sets of models based on various combinations of POD basis functions can be produced and simulated in series. As an example, the Galerkin_Project code could be requested to produce models for the following set of POD bases $\{\{1,2,3\},\{1,2,3,4\},\{1,3,5\}\}$. For this example, models would be generated using the first 3 basis functions, the first 4 basis functions, and the first 5 basis function excluding the 2nd and 4th basis functions. From this set of basis functions, the Galerkin coefficients and eddy viscosity terms would be calculated for each model described in Table 3.1, with the exception of those using the basis transformation method. With each model generated, simulation would be performed by integrating these models such that at least as many state transitions are observed as in the empirical data source. This will ensure that at the estimate of simulations stochastic matrix is at least as good as the estimate produced for the empirical data.

This estimate can be performed in a number of ways using different final times and sampling rates. The maximum likelihood estimate of the stochastic matrix in Eq. 3.41 has been shown in Appendix B to be valid for multiple Markov chains of the same length. Because the ratio between the ensemble sampling rate and the system's key frequency may be $f_{\text{ensemble}}/f_{\text{system}} \ll 1$, time integration in some scenarios proved to be excessively long. To overcome this when the frequency ratio follows the above inequality, tens or hundreds of Markov chains are interleaved such that the required integration time is

reduced signifcantly. Figure 4.1 below, presents a sample image showing how the integration time is reduce using many Markov chains compared to only one.



(a) Integration profile using one Markov chain.



(b) Integration profile using many Markov chains.

Figure 4.1: Comparison of how time integration is performed for one and two Markov chains for an example system with empirical sampling rate of 0.5Hz. In subfigure b, red and black represent samples used for two different Markov chains.

Notice in Figure 4.1b that the two sets of lines, black and red, both have spacing of 2 seconds between each sample. Notice that while the sampling rate of the integration is doubled, and time integrated is nearly halved, the spacing between elements of the same Markov chain are kept at the example empirical sampling rate. This process of interweaving chains can be extended for many additional chains in order to reduce integration time. Results are then saved to a file to be potentially be used as a basis for the basis transformation method explained next, or for scoring.

The basis transformation method can be seen as a post-processing method that potentially provides a further refinement to each model, and is thus invoked last, as Modified_Basis. Again, a top level overview is described:

Algorithm 5: Modified_Basis (generate and simulate a modified ROM given an existing ROM)

Input : A set of configurations $flags$ and a test $directory$
Output: A file containing modified model's coefficents and simulation results

function: Modified_Basis $(flags, directory)$;
$[x, pod_u, \lambda] \leftarrow$ load_pod($directory$);
$[l, q] \leftarrow$ load_galerkin($directory$);
foreach *model* **do**
    $[C, L, Q, \lambda] \leftarrow$ term2order(λ, l, q);
    $\epsilon \leftarrow$ coarse_search(C, L, Q, λ);
    $[\tilde{C}, \tilde{L}, \tilde{Q}] \leftarrow$ fine_search$(C, L, Q, \lambda, \epsilon)$;
    $[a_s, t_s] \leftarrow$ ode_solve$(\tilde{C}, \tilde{L}, \tilde{Q})$;
    plot(a_s, t_s);
    save$(a_s, t_s, l, q, sigma_d, sigma_l)$;
end

Modified_Basis departs slightly from the exact procedure set forth by Balajewicz et. al [5], by separating the root finding procedure into two steps. It was originally suggested that MATLAB's fzero command would be sufficient to find a root. This rarely proved to be fruitful, so a coarse sweep of values in the vicinity of the initial guess for ϵ is performed. If a change of sign is found, then a formal root finding procedure is performed. Otherwise, a minimization is performed at the lowest value found during the coarse sweep. Once the resulting model is simulated, data is again stored to be scored later.

Once all the models have been generated they now can be scored by the methods presented at the end of Chapter 3. Below is its operational description.

Algorithm 6: Score_Model (score model)

Input : A set of configurations $flags$ and a test $directory$
Output: A file containing model score to the data

function: Score_Model $(flags, directory)$;
$[a_e, f_s] \leftarrow$ load_pod($directory$);
$[a_s] \leftarrow$ load_galerkin($directory$);
$[clusters, X_i] \leftarrow$ cluster(a_e);
$P_e \leftarrow$ MLE($clusters, a_s$);
foreach *model* **do**
    $X_i \leftarrow$ classify($clusters, a_s$);
    $P_s \leftarrow$ MLE($clusters, a_e$);
    $[\sigma_d, \sigma_l] \leftarrow$ score(P_e, P_s, a_e);
    $save(\sigma_l, \sigma_d)$
end

Once models are generated, scoring takes relatively little time, on the order of a few seconds, with results saved to file.

4.1 Non-Orthogonal Vertex Volume

The first notable obstacle in implementing the theory came in the form of determining the volume or area represented at each vertex. Throughout this section the word volume will be used in a n-dimensional sense of a finite, closed, boundary in $\mathbb{R}^n$. The data used in this work is discrete, so the volume represented by each vertex is required for the many L_2 inner products taken. Data coming from experimental sources proved trivial as the mesh was uniform and orthogonal. On the other hand, data from the numerically generated axisymetric jet used a non-uniform, non-orthogonal mesh, as a means of balancing computation cost and accuracy. In addition, information about the boundary was included to further refine the estimate of the volume.

The first stage in calculating the volume, is determining which interstitial points to include in the computation. Below, in Figure 4.2, is an example of a two dimensional mesh with interstitial points included, but note this can be extended to three dimensional meshes as well.

From this mesh, any of the surrounding vertices could be in or on a boundary. Information about the boundary is determined by inspecting the velocity magnitude at each point. Points where 99% of the images show zero velocity are considered out of the flow. MATLAB's edge function, which

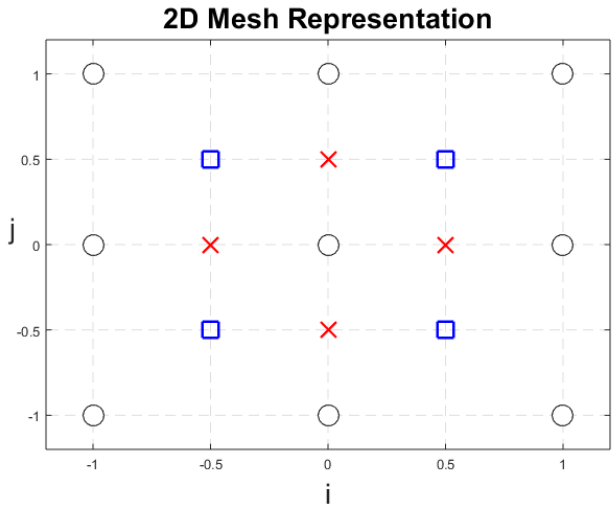


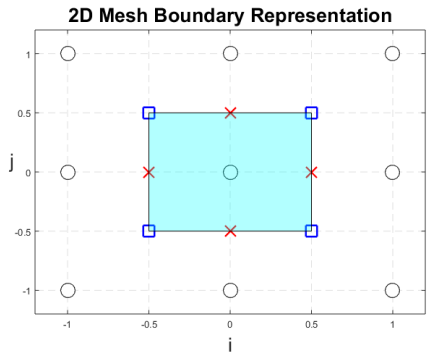
Figure 4.2: Generic two dimensional grid. Black circles represent vertices of the mesh, while red crosses and blue squares represent each interstitial points.

uses a Sobel operator and line thinning algorithm, is used to detect the boundaries between points in and out of the flow. Using this information the central vertex and all surrounding interstitial vertices are included that are either in the flow or on the boundary. Examples of this methodology are shown in Figure 4.3.

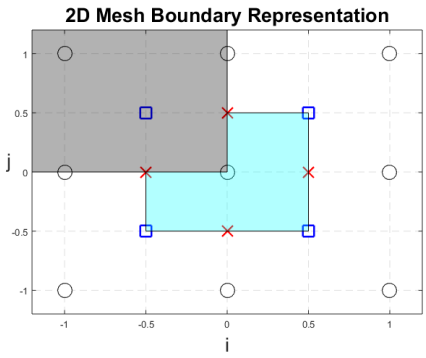
In Figure 4.3, the blue region shows the volume represented by the central vertex and the black region shows vertices on or in the boundary. Now that the interstitial vertices have been found that define the region represented by the central vertex, its volume can be determined. First, the blue region in Figure 4.3 needs to be subdivided such that each resulting subregion is convex. An example of this is shown in Figure 4.4a. Typically, breaking down a whole region into 'corners' such as Figure 4.4a will result in a convex region for affine or curvilinear coordinates. Once the subregion is known to be convex it and can be further segmented into a set of simplices on $\mathbb{R}^n$ with volume represented by Eq. 4.1 for arbitrary dimension. The final volume is found by summing these simplices for the entire region defined by central vertex.

$$\left| \frac{1}{n!} \det(v_1 - v_0, v_2 - v_0, \ldots, v_n - v_0) \right| \tag{4.1}$$

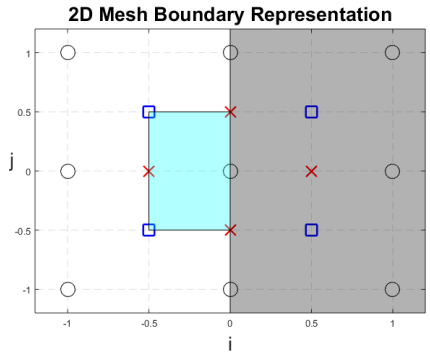
Finally, as an illustrative example, a surface plot of the area represented by meshes for the jet and airfoil at each vertex are shown in Figure 4.5. figures



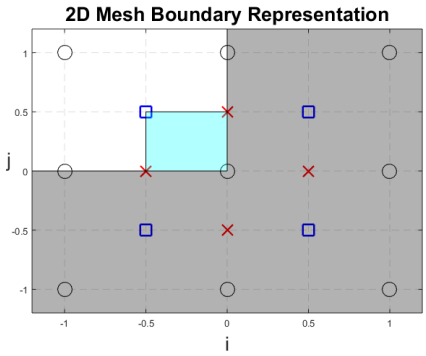
(a) No vertices in bounds



(b) Three vertices in bounds

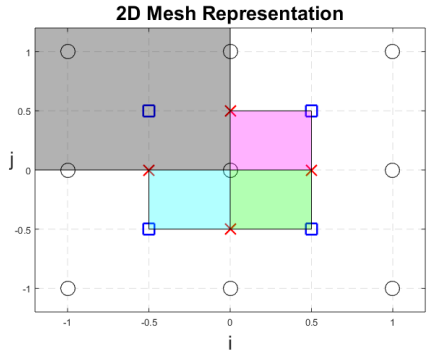


(c) Six vertices in bounds

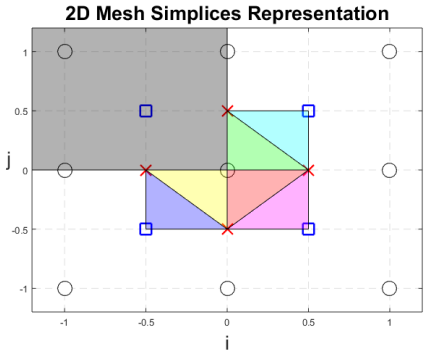


(d) Eight vertices in bounds

Figure 4.3: How volume is determined for different boundary configuration.



(a) Subdivided into convex subregion.



(b) Subdivided into simplices.

Figure 4.4: Subdivision of vertex volume into simplices.

4.5b and 4.5d shows how area represented by each vertex changes across the entire mesh for meshes shown in figures 4.5a and 4.5c. In Figure 4.5b the area of each vertex is quite small near the x and y axis, and again small near the origin of Figure 4.5d. Figure 4.5e shows a constant mesh spacing across the whole grid, which is reflected by the constant area captured at each vertex with the exception of those points not captured by the PIV system, and those on the boundary.

4.2 Adaptive Numerical Differentiation

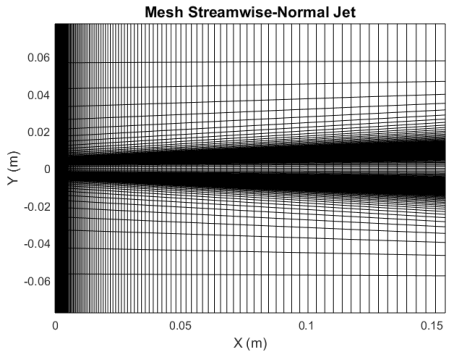
The second major challenge with implementing the theory of Chapter 3, was to calculate numerically the many required derivative terms of the vector fields with accuracy. The empirically and numerically generated data each presented their own challenges. As mentioned in the previous section, the numerical data does not use a uniformly spaced or orthogonal mesh. This requires the use of a coordinate transformation to a 'computational' domain; where the use of finite difference methods become straight forward. The empirical data presents an entirely different problem, the identification of open or closed boundaries. Regions of the grid indicating zero fluid movement may be a byproduct of the position of the PIV system relative to the test chamber, in which case the gradient near this region should be relatively low. On the other hand some regions of the grid are on physical boundaries such as the wall that defines the cavity or the airfoil. In these cases, because of the no-slip condition, the gradient should be quite steep. In both scenarios only the velocity is known to be zero at the real or imaginary boundary.

4.2.1 Coordinate Transform

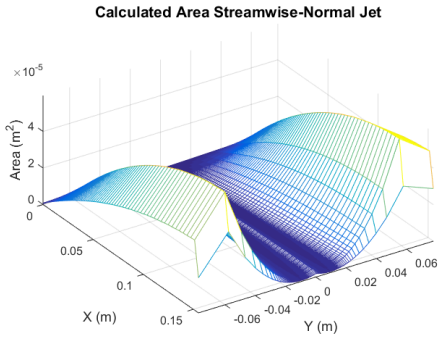
The coordinate transform, is effectively a multivariable application of the chain rule. Here, we want to represent the following derivative term as:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x} \tag{4.2}$$

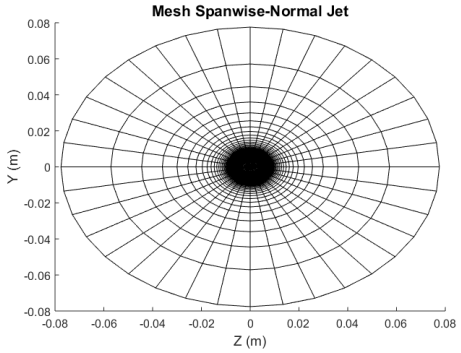
Where ξ is the 'computational' domain, a uniformly increasing field, which discreetly can be thought of as a matrix of indices in a given direction. This enables the use of the well described finite difference methods for $\partial u/\partial \xi$. Unfortunately, this still leaves the computation of $\partial \xi/\partial x$ dependent on the non-orthogonal basis of x. First, it is noted that $\partial \xi/\partial x$ at its core is a function $f(\cdot) = \partial(\cdot)/\partial x$. This function can be expressed with all derivative terms taken on ξ as follows:



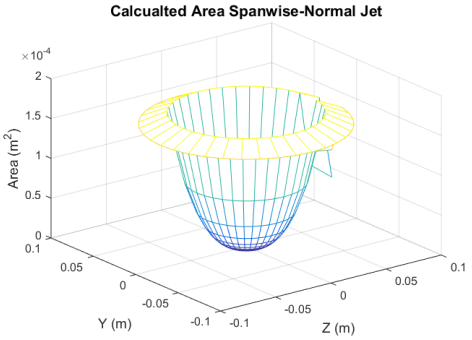
(a) Streamwise-normal plane mesh of the jet



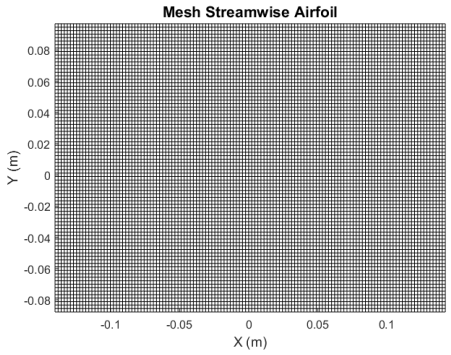
(b) Streamwise-normal plane area of the jet



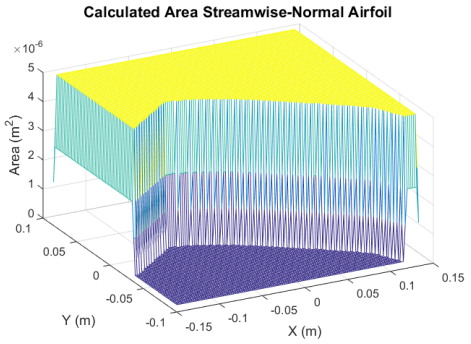
(c) Spanwise-normal plane mesh of the jet



(d) Spanwise-normal plane area of the jet



(e) Streamwise-normal plane mesh of the airfoil



(f) Streamwise-normal plane area of the airofil

Figure 4.5: Examples of the calculated area of uniform and nonuniform meshes.

$$\frac{\partial}{\partial \xi} = \frac{\partial x}{\partial \xi}\frac{\partial}{\partial x} \tag{4.3}$$

rearranging:

$$\frac{\partial}{\partial x} = \left(\frac{\partial x}{\partial \xi}\right)^{-1}\frac{\partial}{\partial \xi} \tag{4.4}$$

Where $(\partial x/\partial \xi)^{-1}$ generalizes to the inverse jacobian. The final representation of the derivative term is now:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi}\left(\frac{\partial x}{\partial \xi}\right)^{-1} \tag{4.5}$$

With Eq. 4.5 derivatives with respect to x are now possible by using finite difference methods with respect only the computational domain ξ

4.2.2 Method Selection

In order to maximize the accuracy of the many derivative terms needed generate the POD-Galerkin coefficients, finite difference methods are dynamically selected based on the boundaries present in the data. At a high level the procedure follows Algorithm 7 below.

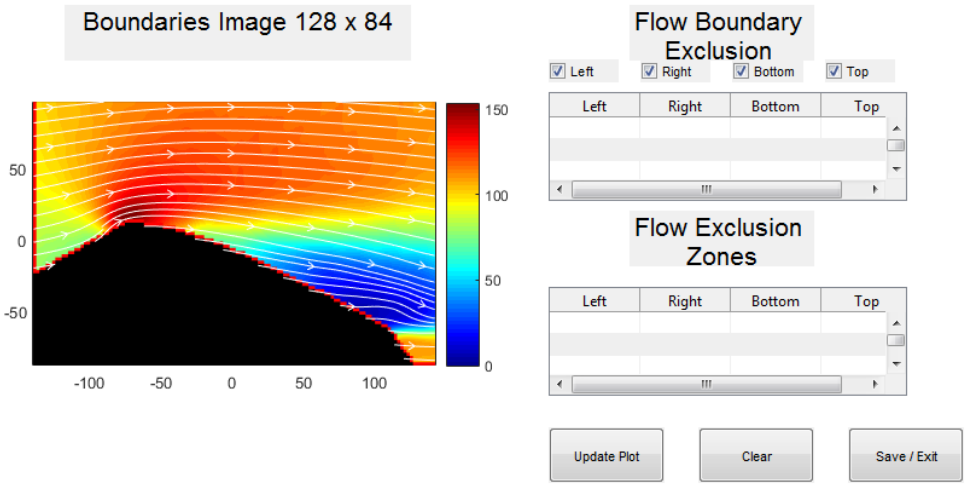
Algorithm 7: boundary_diff (used to calculate derivative based on boundaries)

Input : An array u and x representing the the velocity field and coordinate grid

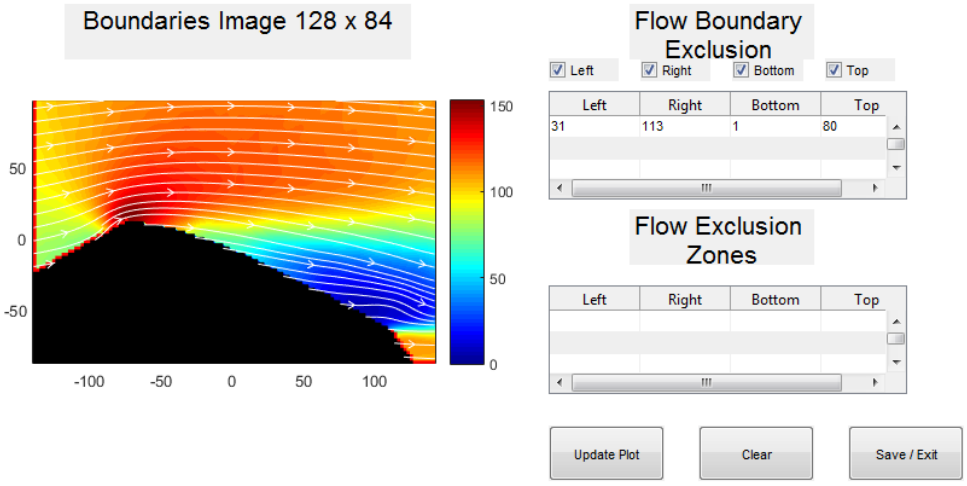
Output: An array udx representing the gradient

function: boundary_diff (u, x) ;
$[bnds, bnds_n] \leftarrow$ bounds(u);
$[bnds, bnds_n] \leftarrow$ gui_bounds$(u, bnds, bnds_n)$;
$uniform \leftarrow$ check_mesh(x);
if $uniform = false$ **then**
    $\xi \leftarrow$ make_mesh(x);
    $J \leftarrow$ jacobain(x, ξ);
    $transform \leftarrow$ coordinate_transform1(J);
    $x \leftarrow \xi$;
end
$method \leftarrow$ select_method$(bnds, bnds_n)$;
$\alpha \leftarrow$ get_coefficients$(method, x)$;
$udx \leftarrow$ finite_difference(α, u);
if $uniform = false$ **then**
    $udx \leftarrow$ coordinate_transform2$(udx, transform)$;
end

As previously described in Section 4.1, the boundaries and boundary normals are found by locating grid points with no velocity and MATLAB's edge function. In order to distinguish between physical and imaginary boundaries an interactive GUI is launched, to remove regions of the boundary surface normal that correspond to some physical structure, such as a wall. An example instance of the GUI is shown a airfoil data sets shown in Figure 4.6.



(a) Intial view GUI for an Airfoil



(b) GUI after a boundary normal region was removed from Airfoil

Figure 4.6: An example use of the GUI, to remove physical boundaries. Detected boundaries shown in red along perimeter

Once the modification have been made to the boundary normal matrix, the coordinate transform can be performed. In order to account for non-orthogonal meshes a matrix of indices ξ is produced representing the 'computational' domain. Using the index matrix ξ and the grid x representing the location of each grid point, a transformation matrix is calculated by taking the inverse jacobian of ξ and x over the whole grid.

Selection of an appropriate finite difference method proved crucial to producing sufficiently accurate differentiation for model generation. The method is selected based on the number of consecutive grid points within the boundaries. Along any dimension, if there is a at least 5 grid points that are consecutively within the boundary, that region of the grid will use $4th$ order finite difference methods. This proceeds down until only 2 points are available, where $1st$ order methods are used. In the base case, where only one point is present, it is assumed to have a gradient of 0 along that dimension. In order to account for locations that have been deemed to be physical boundaries, points within two indices of the boundary are changed to $4th$ order central to account for the hard gradient present in these locations. Below, Figure 4.7 shows an example of how finite difference methods would be selected for a hypothetical grid with boundaries. A legend of the finite difference methods is presented in Table 4.1 for reference.

Table 4.1: Legend for Figure 4.7, indicating the order and form of the finite difference methods used. Circles indicate the location of the grid point in question for each finite element stencil

	Name	Pattern		Name	Pattern
C_4	$4th$ order central	$\times \times \circ \times \times$	C_2	$2nd$ order central	$\times \circ \times$
F_4	$4th$ order forward	$\circ \times \times \times \times$	F_3	$3rd$ order forward	$\circ \times \times\times$
F_2	$2nd$ order forward	$\circ \times \times$	F_1	$1st$ order forward	$\circ\times$
B_4	$4th$ order backward	$\times \times \times \times \circ$	B_3	$3rd$ order backward	$\times \times \times\circ$
B_2	$2nd$ order backward	$\times \times \circ$	B_1	$1st$ order backward	$\times\circ$
FB_4	$4th$ order forward bias	$\times \circ \times \times \times$	FB_3	$3rd$ order forward bias	$\times \circ \times\times$
BB_4	$4th$ order backward bias	$\times \times \times \circ \times$	BB_3	$3rd$ order backward bias	$\times \times \circ\times$

4.3 Parallelization

Beyond the challenges of implementing methods to calculate the quantities required to generate each model a significant effort was placed in producing efficient and scalable code. This was achieved with the common practice in MATLAB and other scientific scripting languages of 'vectorizing' as well as explicit parallelization using MATLAB's parallel computing toolbox. A small set of tests were performed on quad-core workstation computer to investigate this effort.

Two common means of assessing the performance of parallel code are known as speedup and efficiency. The relationships are simple and shown below

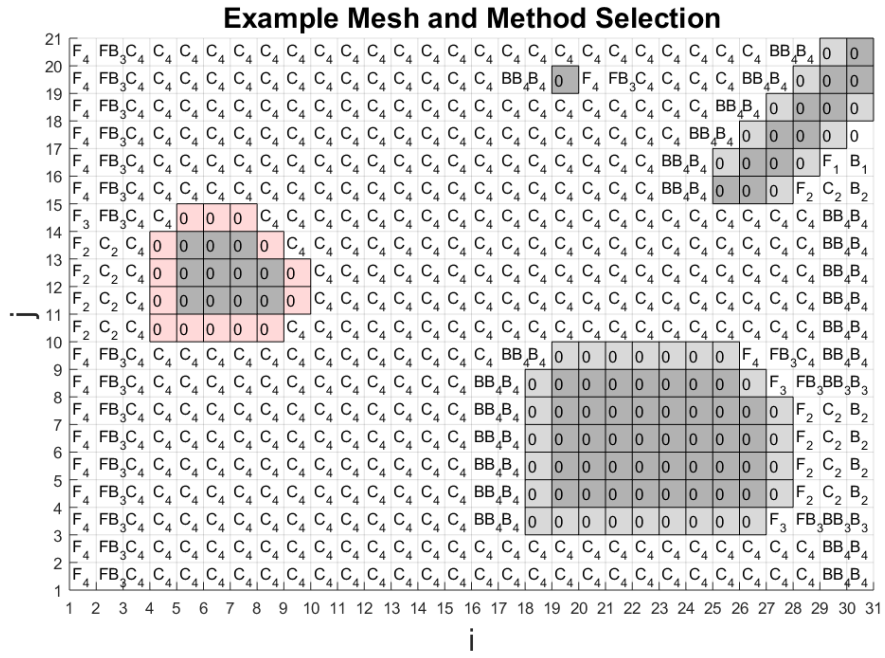
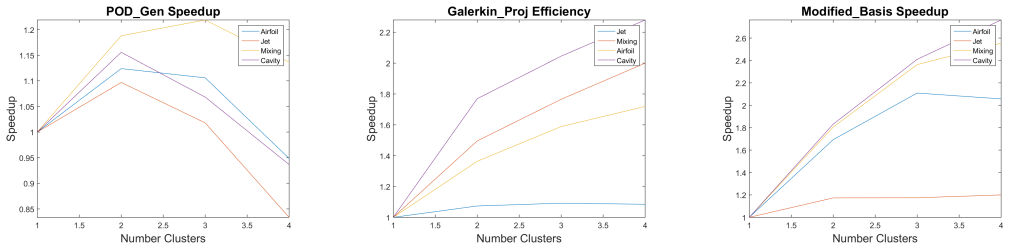


Figure 4.7: Example results of calling the select_method function on a grid with boundaries for the x direction. Black shows areas out of the flow, gray represents areas where boundary are due to some constraint on the PIV system, and red represents points on a physical boundary

$$S_n = \frac{T_n}{T_1} \tag{4.6}$$

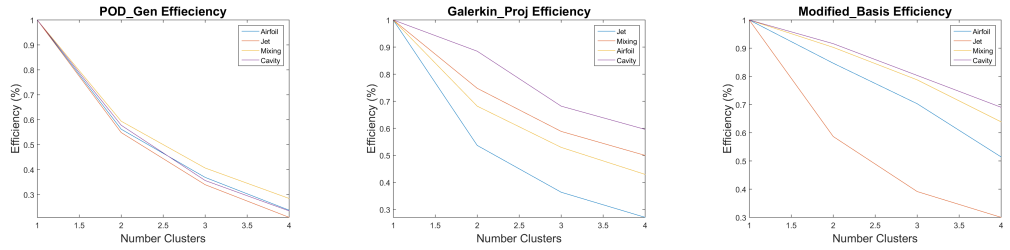
$$E_n = \frac{S_n}{n} \tag{4.7}$$

Where T is the run time, n is the number of cores, S is the speedup and E is the runtime efficiency. An ideal parallelization would result in runtime efficiency of 100%, indicting the use of n processors would result in a runtime speedup of n. Note that such efficiency is rarely achieved in practice. A small test was conducted on a local workstation for 1 to 4 cores with results shown below in figures 4.8 and 4.9. Included in the test is the POD_Gen, Galerkin_Proj and Modified_Basis functions. Because of the very low run time of Score_Model, it was not included.



(a) POD_Gen speedup (b) Galerkin_Proj speedup (c) Modified_Basis speedup

Figure 4.8: Speedup of POD_Gen, Galerkin_Proj and Modified_Basis for 1 to 4 cores.



(a) POD_Gen efficiency (b) Galerkin_Proj efficiency (c) Modified_Basis efficiency

Figure 4.9: Efficiency of POD_Gen, Galerkin_Proj and Modified_Basis for 1 to 4 cores.

From figures 4.8 and 4.9 it becomes obvious that both the Galerkin projection code and the basis transformation code scale much better than the

POD basis function generation code. This is likely due to the lack of highly CPU bound operations, with the exception of POD itself, which for these data sets does not take longer than 10 seconds. On the other hand, the Galerkin projection code and the basis transformation code scale because of the many required ODE integrations that can simply be run simultaneously.

Chapter 5

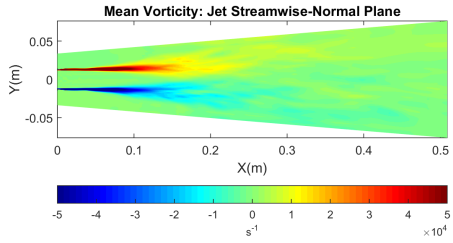
Experimental Data

In an attempt to get a good spread of possible flow configurations that the proposed validation procedure could be implemented for, four data sets are explored. Three of the data sets are gathered experimentally using a 2D PIV system with sampling rates that are low compared to the typical frequency ranges present in each system. The last data set is derived numerically and operates in the compressible flow regime. This last set was included for two reasons, first its sampling rate is significantly higher than the frequency content of the flow, which will help give some insights on the possible importance of the ratio between fundamental flow frequencies and sampling rate for the measures. In addition, the numerical data set will be modeled using the incompressible Navier-Stokes equations. Intuition says that these simulation results should perform poorly as an invalid assumption will be enforced. These models will ensure that trends in the measurement can be explored for model characteristics that are further from the actual system.

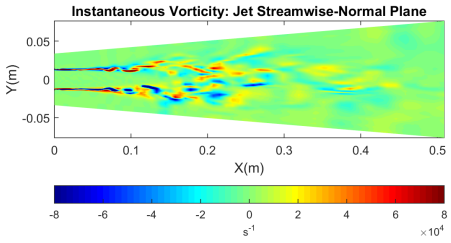
5.1 Axisymetric Jet

The oldest data set is for a 3D large eddy simulation (LES) generated axisymetric jet of air, developed by Dr. James DeBonis, of NASA's Glenn Research Center, using a compressible Smagorinsky sub-grid model [20]. The jet has an inlet velocity of Mach 1.4 for a 2.54 cm nozzle operating in the fully developed regime. The simulated region stretches 20 jet diameters in the streamwise direction and 10 jet diameters in the spanwise and normal directions. From the full 3D data, a selection of 2D slices are extracted following Caraballo [13]. Slices are located in the streamwise-normal plane, centered on the jet, as well as three spanwise-normal planes, with slices at 3,

6 and 9 jet diameters from the nozzle exit. The mesh is non-orthogonal in both the affine streamwise-normal plane and radial spanwise-normal planes, which requires the techinques for calculating volume described in Section 4.1. The flow's Reynolds number based on the diameter of nozzle and the free stream velocity at the nozzle exit is 1.2×10^6. Figures 5.1 and 5.2 show the averaged and instantaneous plots of the flows vorticity.

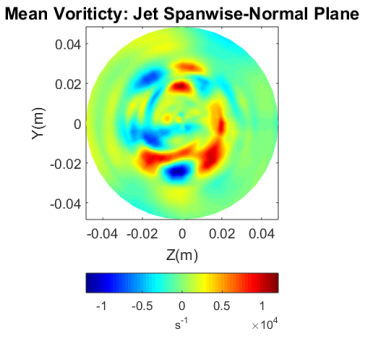


(a) Mean spanwise component vorticity field

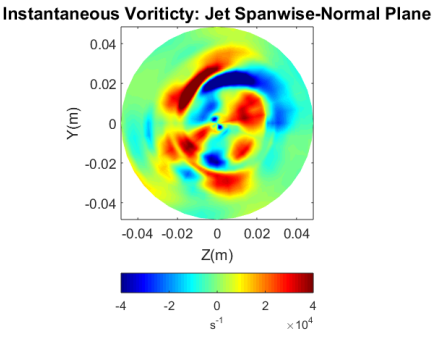


(b) Sample spanwise component Instantaneous vorticity field

Figure 5.1: PIV data for the axisymetric jet in streamwise and normal plane.



(a) Mean streamwise component vorticity field



(b) Sample streamwise component Instantaneous vorticity field

Figure 5.2: PIV data for the axisymetric jet in spanwise and normal plane at 9 jet diameters downstream.

Figure 5.1 shows that the jet remains very organized for roughly the first $1/4th$ of the viewing window then rapidly mixes with the surrounding low speed flow. Figure 5.2 shows the flow in the spanwise-normal plane

at 9 jet diameters downstream. Below in figures 5.3 and 5.4 are shown the dominate structures in the flow. Note that for Figure 5.3 the colormap scaled to highlight features further downstream.

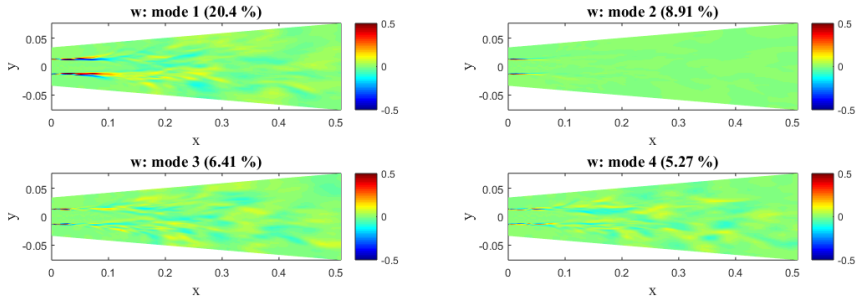


Figure 5.3: First 4 POD basis functions shown as vorticity for the streamwise and normal plane. (magnitude scaled by maximum absolute value)

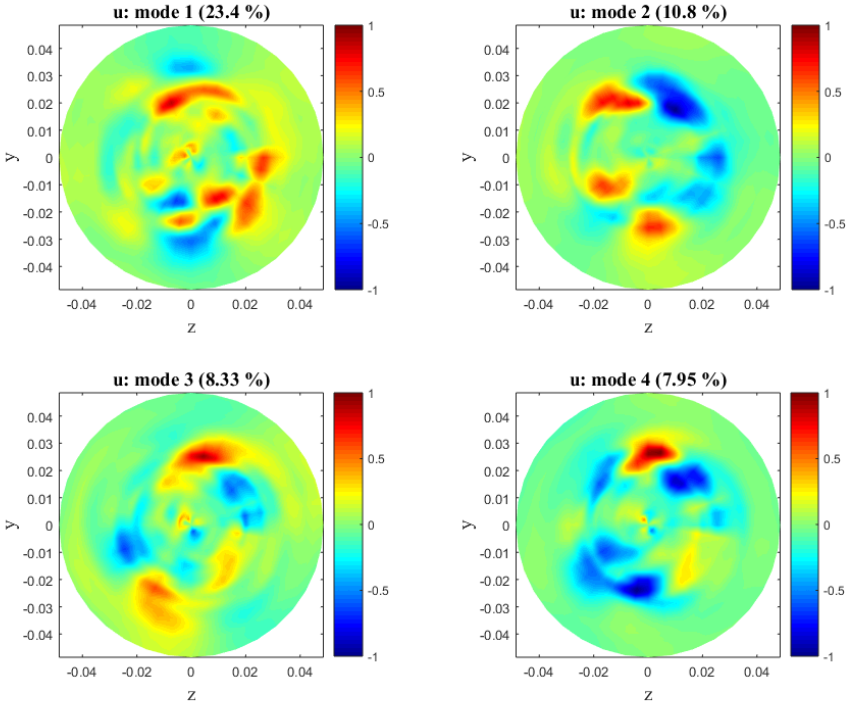
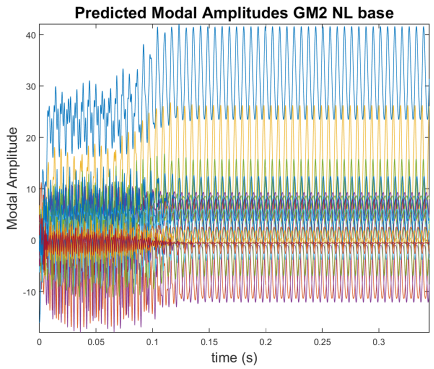
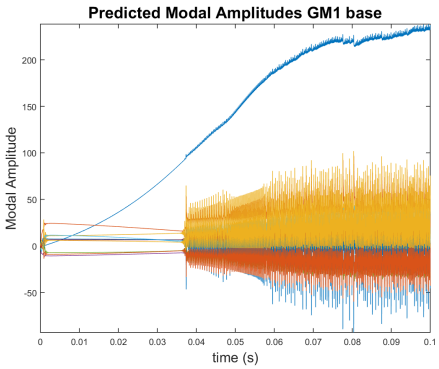


Figure 5.4: First 4 POD basis functions shown as vorticity for spanwise and normal plane. (magnitude scaled by maximum absolute value)

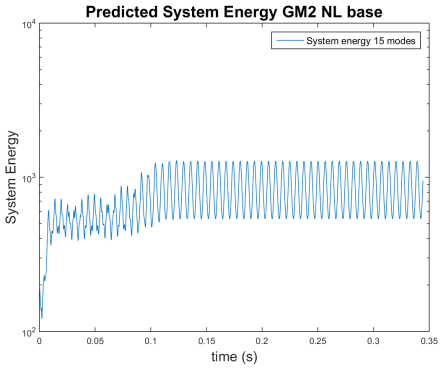
From the POD basis functions generated, models described in the previous chapter are produced. Here plots of the modal amplitude, system energy and frequency response are shown for a model that the author would consider good based on a few criteria. First, the system does not come to rest on a fix point or diverge in the long term. Next, a quick observation about the mean and spread of each modal amplitude is made, favoring models that maintain some level of oscillation with the mean of the oscillation centered around 0 on the y axis, because each POD basis function should be approximately normally distributed with a mean of zero [28]. Finally, models that maintain or minimally grow in energy from their initial conditions derived from an empirical flow snapshot are preferred. These attributes are favorable for any POD-Galerkin models because their predictions more closely follow the model amplitude distribution of the empirical data. Therefore these are the author's subjective 'at a glance' criteria for model selection. While frequency components carry meaningful information about the system, it requires some form of outside knowledge such as predicable key frequencies or some other data, such as pressure sensors or hot film measurements. Forcing frequency in the other test cases also provides a target for frequency comparison, but from the onset it is not clear what POD basis function or functions the forcing will manifest, so frequency was not considered in selecting the models shown in this chapter. Below, in Figure 5.5, candidate models for the jet are shown that follow these criteria. In these plots and those following, modal amplitude and frequency response plots show the value or frequency content of each of the modal amplitudes a_i from Eq. 3.8.



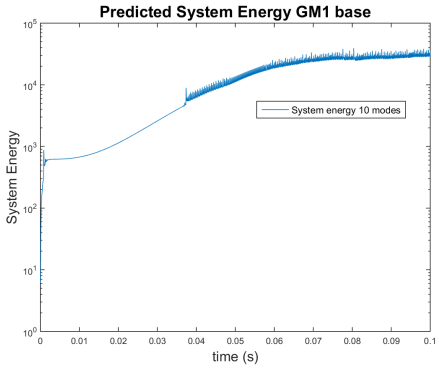
(a) Modal amplitude of streamwise-normal jet



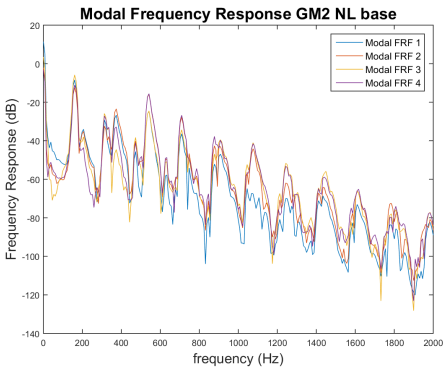
(b) Modal amplitude of spanwise-normal jet



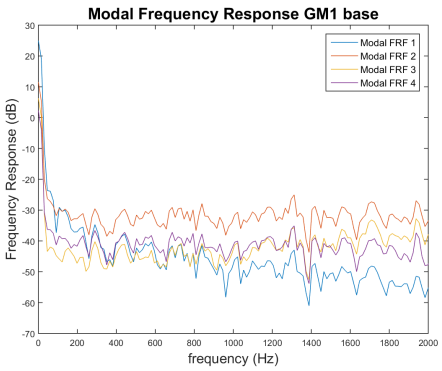
(c) System energy of streamwise-normal jet



(d) System energy of spanwise-normal jet



(e) Frequency response of streamwise-normal jet



(f) Frequency response of spanwise-normal jet

Figure 5.5: System characteristics for candidate models of the jet. Here a, c and e represent characteristics for the streamwise-normal plane, and $b, d,$ and f represent spanwise-normal plane model at 3 jet diameters from the orifice.

Both the streamwise-normal and spanwise-normal flows appears to move through at least 2 transitions in phase space. In both cases there appears to be a transition from the initial conditions to some new attracting region in the model space, with both appearing to reach a long term stability. In the spanwise-normal case, the predicted solution likely has little to do with the real dynamics of the jet. Not only is the flow modeled using the incompressible Navier-Stokes, but the oscillations in the modal amplitude abruptly appears after a relatively short period in the solution. The streamwise-normal plane produces a more sensible model, that may to some degree approximate the real system. This is because, a small region immediately following the jet orifice actually falls into the compressible range of Mach number, therefore dynamics outside of this small cone may be adequately modeled using the incompressible equations.

5.2 Cavity Flow

The oldest experimental data set is a cavity flow, collected at the Gas Dynamics and Turbulence Laboratory at the Ohio State University between 2005 and 2007. The test chamber features a shallow cavity of width $50.8mm$ depth of $12.7mm$ and length of $50.8mm$ [11]. Air for this data set was stored and dried in two high capacity tanks and is conditioned in a stagnation chamber before entering the test section through a smoothly contoured converging nozzle [19]. The test chamber is capable of producing flows up to Mach 0.7 but flows were recorded in the incompressible range of Mach 0.3 [11, 19]. Flow forcing is provided by a Selenium D3300Ti compression driver capable of producing oscillation of $1 - 20$ kHz and connected to the flow through a highly converging nozzle at a 30° angle to the flow. This nozzle exalted forcing frequencies in the range of $500 - 3000$ Hz, while in isolation, the compression driver had a relatively flat frequency response. Reynolds numbers for this cavity flow are 1×10^5 based on step height and 2×10^4 based on the initial shear momentum thickness.

Velocity data for this flow was captured using a 2D LaVision particle image velocimetry system. A series of baseline flow conditions, as well as, forced flows with driving actuation frequencies over a spectrum of $1610 - 3920$ Hz with driving voltages between 190 to 750 volts were investigated. Velocity fields were captured at 128×78 or 128×84 grid points, typical in a series of 1000 images. The size of the chamber and grid points collected resulted in a spatial resolution of approximately $0.4mm$. Several flush mounted Kulite pressure transducers were placed around the test chamber with a frequency response of up to 50 kHz. These sensors were originally used to generate

stochastic estimations of the flow state for online control purposes [12]. An image of the experimental setup shown in Figure 5.6 for reference [53]. For this study only [53] the velocity data will be used to generate the ROMs. An example forced flow using a driving frequency at 1830 Hz and driving voltage of 400, as well as, its accompanying baseline flow is shown in Figure 5.7 and Figure 5.8 and below.

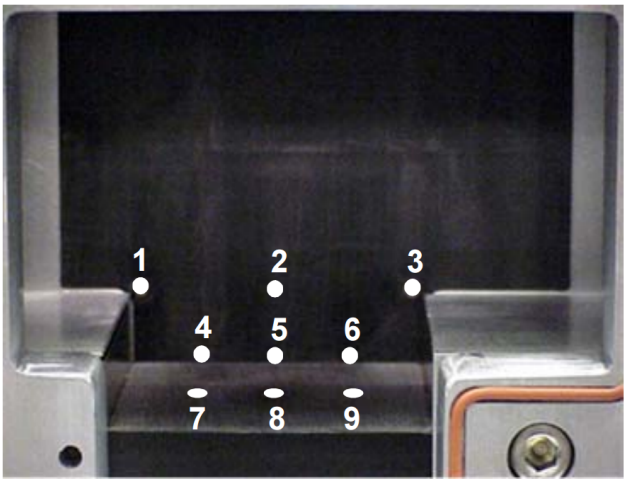
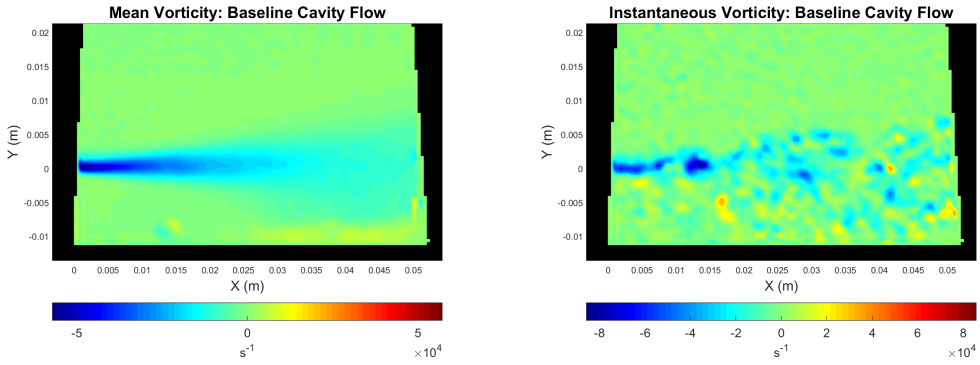


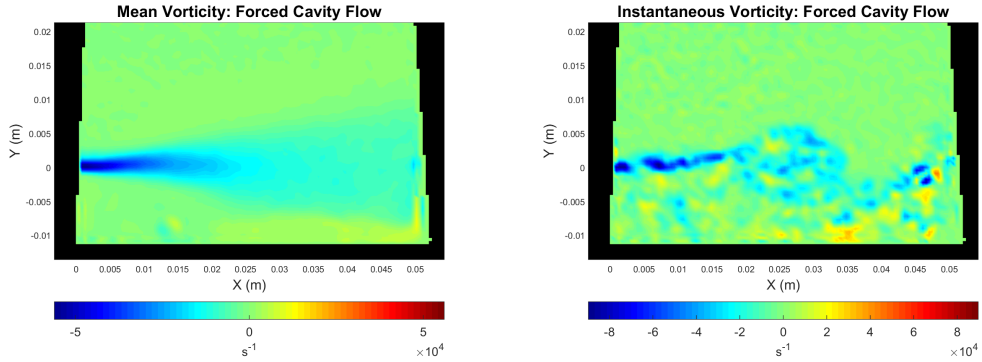
Figure 5.6: Image of the experimental test section of the cavity with flow inlet on the left [53]. Numbered locations indicate the position of pressure transducer in the original studies.



(a) Mean spanwise component vorticity field

(b) Sample spanwise component Instantaneous vorticity field

Figure 5.7: PIV data for cavity flow: baseline case.



(a) Mean spanwise component vorticity field

(b) Sample spanwise component Instantaneous vorticity field

Figure 5.8: PIV data for cavity flow: forced case.

The mean flow images in Figures 5.7a and 5.8a show strong clockwise rotation near the leading edge of the cavity indicating rapid diffusion of momentum at the boundary of the cavity and free stream regions. What can be seen in Figure 5.8b is the faint presence of a large roll absence in the unforced case of Figure 5.7b. The presence of forcing is again reflected in the POD modes shown below in figures 5.9 and 5.10. Here the most energetic basis functions for the baseline case show rotation throughout the free stream boundary. The forced case shows the strongest regions in the organized structures closer to the leading edge of the boundary. This reflects the intended goal of exciting the natural instabilities present at edge of the cavity.

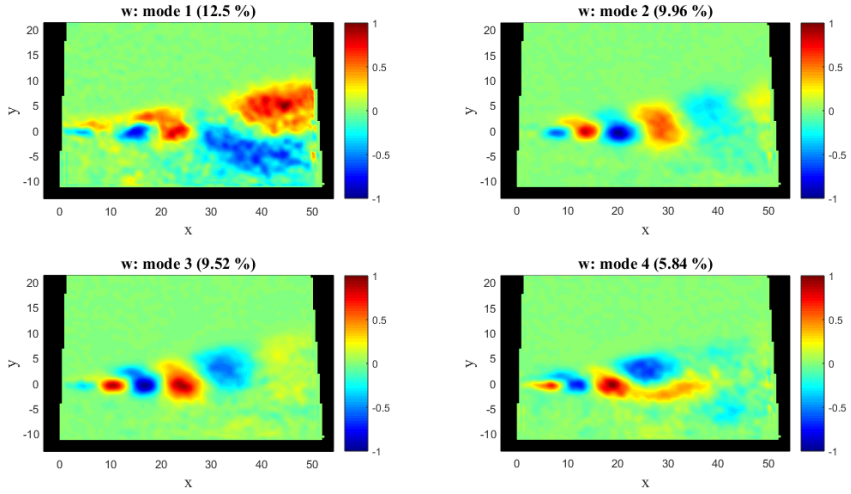


Figure 5.9: First 4 POD basis functions shown as vorticity for the cavity baseline case. (magnitude scaled by maximum absolute value)

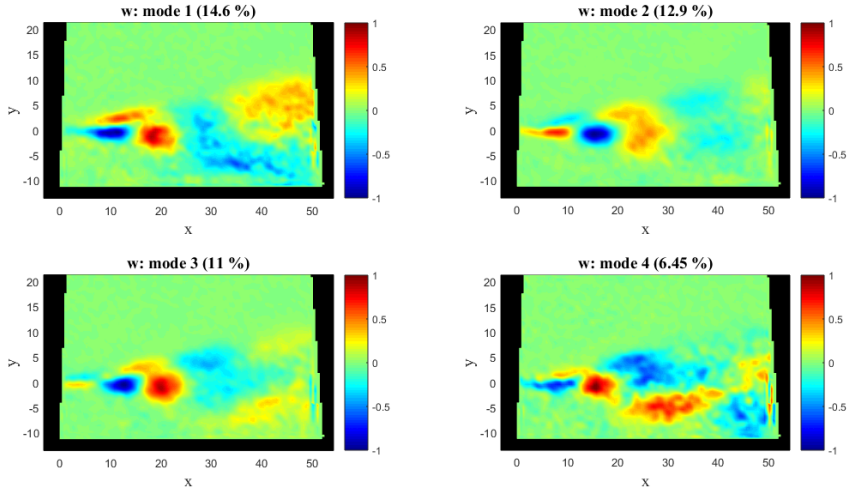
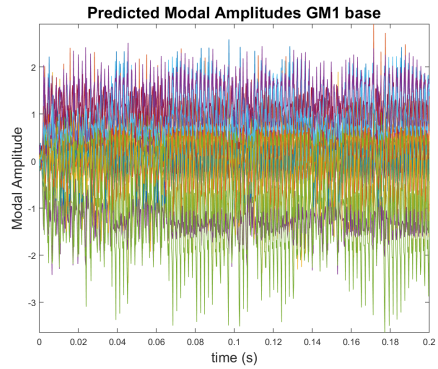
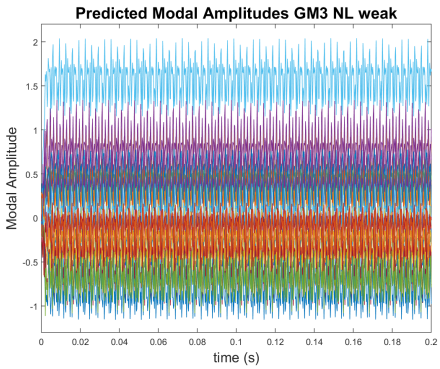


Figure 5.10: First 4 POD basis functions shown as vorticity for the cavity forced case. (magnitude scaled by maximum absolute value)

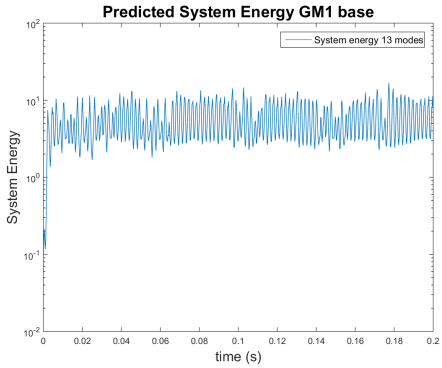
Here as with the jet data, sample candidate models are provided in Figure 5.11 to give an example of what is produced programmatically.



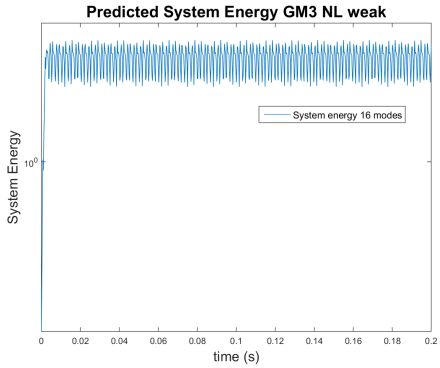
(a) Modal amplitude of a baseline cavity flow



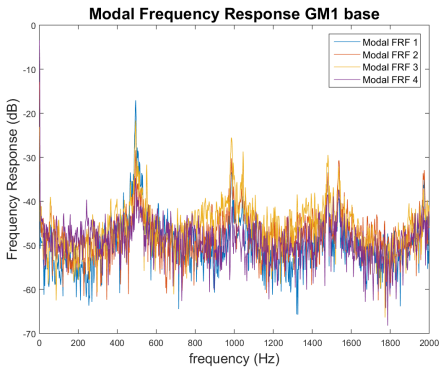
(b) Modal amplitude of a forced cavity flow



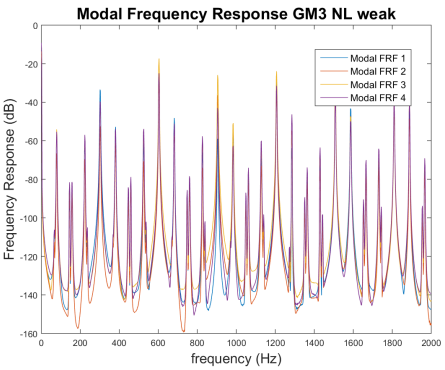
(c) System energy of a baseline cavity flow



(d) System energy of a forced cavity flow



(e) Frequency response of a baseline cavity flow



(f) Frequency response of forced cavity flow

Figure 5.11: System characteristics for candidate models of the cavity. Here a, c, and e represent model characteristics for a baseline cavity flow, and $b, d,$ and f represent a forced cavity flow with actuation provided 1830 Hz at 400 volts

Again, models are selected primarily based on the mean and variance of the modal amplitude as well as the predicted model energy. Observations of the cavity data with Rossiter frequencies will be left to Chapter 6, but the differences in the frequency response is interesting in of itself. In the forced case, the provided models appear to be attracted to a closed orbit in phase space. This is reflected by the unphysically smooth frequency response of this model. Contrast this to the selected baseline model, which like the forced model quickly moves from the initial conditions to a new attracting region in phase space. Unlike the baseline model, this model does not appear to fall on a closed orbit and produces a frequency response more similar to a real data set.

5.3 Airfoil Flow

The experimental data for the airfoil flow was also obtained at the Gas Dynamics and Turbulence Laboratory at the Ohio State University. The data corresponds to the subsonic flow over a NACA 0015 airfoil with a $203mm$ chord length and $609.6mm$ span. The recirculating wind tunnel is capable of producing velocities from 3 to $95m/s$ with free stream turbulence level on the order of 0.25%. The facility includes a heat exchanger to maintain temperature near ambient levels during prolonged testing. For the data used in this work, the Reynolds number based on the chord length and free stream velocity was held constant at 1.15×10^6 corresponding to Mach 0.26 ($U_\infty = 93m/s$). The data include test runs with angles of attack in post-stall configurations of 20°, 18° and 16° as well as pre-stall conditions at 14°, 12°, and 10°. Each inclination includes two runs with open loop forcing and two runs in a baseline configuration.

The velocity data for the flow field was again obtained using a 2D LaVision Particle Image Velocimetry (PIV) system. Nanosecond pulse driven Dielectric Barrier Discharge (ns-DBD) plasma actuators, installed at the leading edge ($x/c = 0.01$) were used to force the flow in an attempt to partially reattach the flow stream to airfoil surface. The forced instances for this data set consist of a forcing frequency of 1250 Hz, which corresponds to a Strouhal number of $F^+ = fc/U_\infty = 2.75$ based on the free stream velocity and the chord of the airfoil. The PIV images have a 128 × 84 grid with spatial resolution of approximately $2.4mm$ sampled at 10 Hz. While additional data was available in the form of pressure and hot film data, only the PIV collected velocity fields will be used to generate the ROMs described in Table 3.1. Additional information on the experimental setup and plasma actuators are given in Rethmel *et al.* [48] and Little *et al.* [36]. An image of

the experimental setup is provided in Figure 5.12 for reference [48]. A sample of a post stall flow at 20° is also provided for both forced and unforced cases in figures 5.13 and 5.14. Note in these images, the region below the airfoil is removed because of interference with the control cables visible in Figure 5.12.

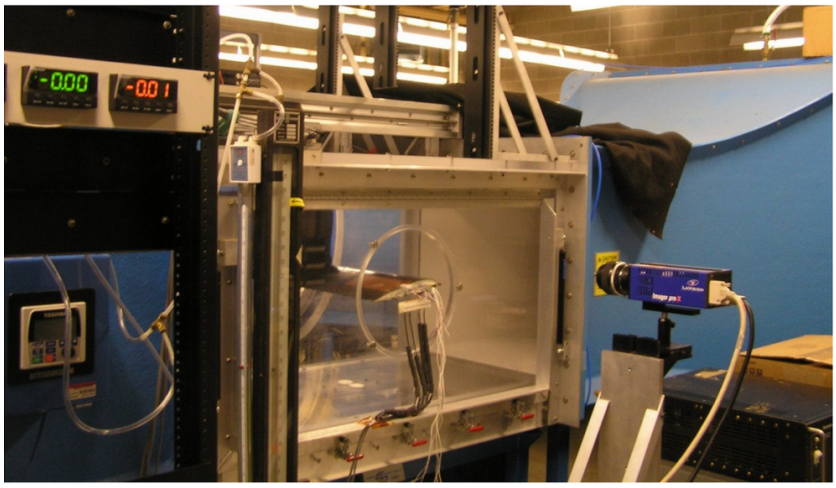
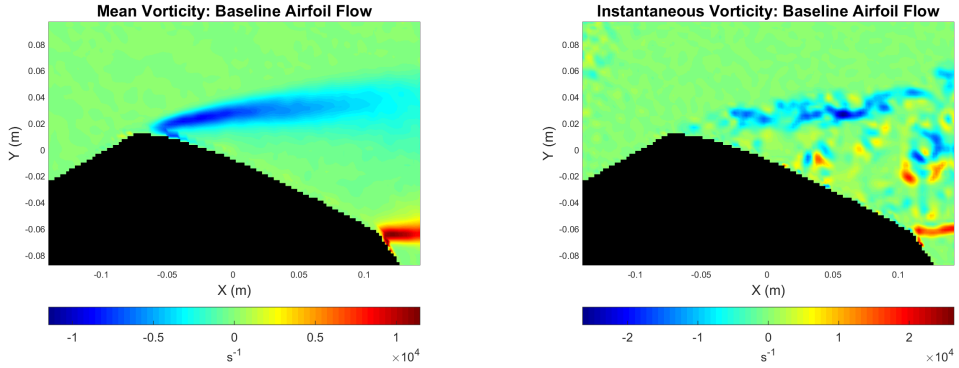


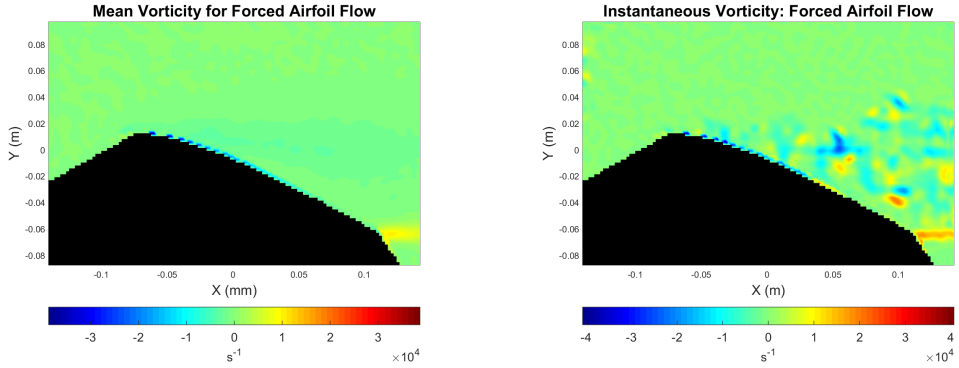
Figure 5.12: Image of the experimental airfoil test section with flow inlet on the right [48].



(a) Mean spanwise component vorticity field

(b) Sample spanwise component Instantaneous vorticity field

Figure 5.13: PIV data for a 20° post stall airfoil: baseline case.



(a) Mean spanwise component vorticity field

(b) Sample spanwise component Instantaneous vorticity field

Figure 5.14: PIV data for a 20° post stall airfoil: forced case.

The mean flows of figures 5.13a and 5.14a show how the flow is at least partially reattached by the forcing with strong vortical regions surrounding the low pressure zones behind the stalled baseline wing. The POD baseline reflect this phenomena, with baseline modes of Figure 5.15 indicating fluctuations in the low pressure zone and the free stream. The reattached flow in the forced case indicates the strongest vortical structures directly above the wing surface with weaker features in the wing's wake.

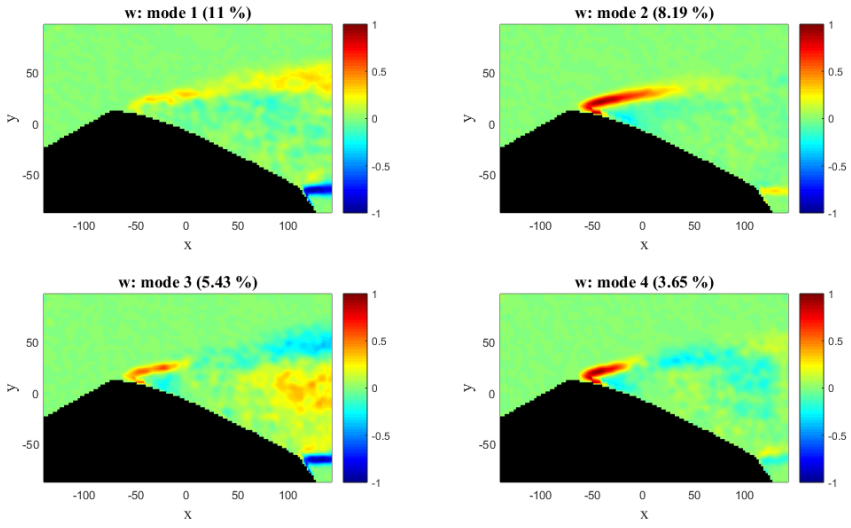


Figure 5.15: First 4 POD basis functions shown as vorticity for the airfoil flow: baseline case. (magnitude scaled by maximum absolute value)

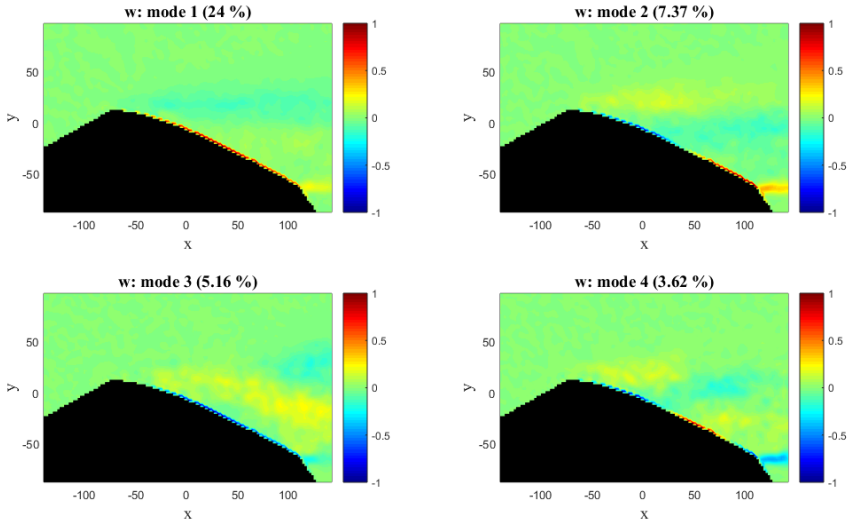
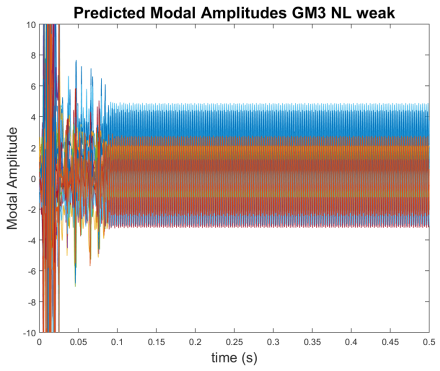
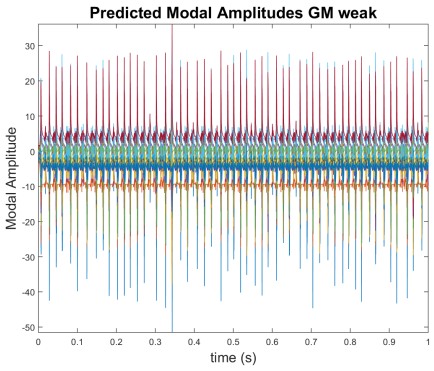


Figure 5.16: First 4 POD basis functions shown as vorticity for the airfoil flow: forced case. (magnitude scaled by maximum absolute value)

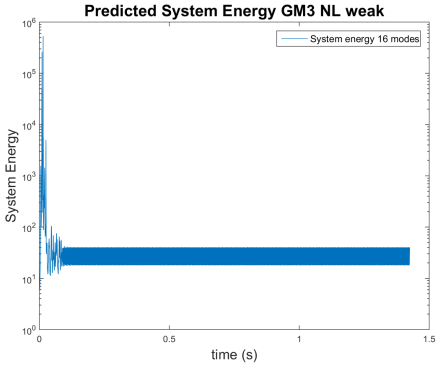
Following the trend of the last two data sets, sample models are again shown in the case of the airfoil.



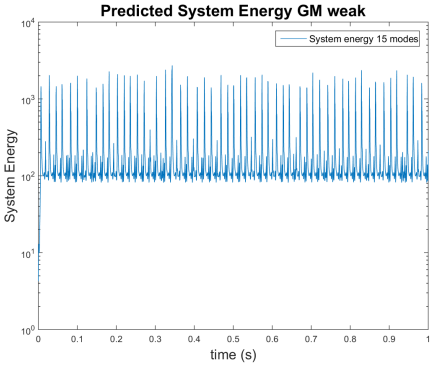
(a) Modal amplitude of baseline airfoil flow



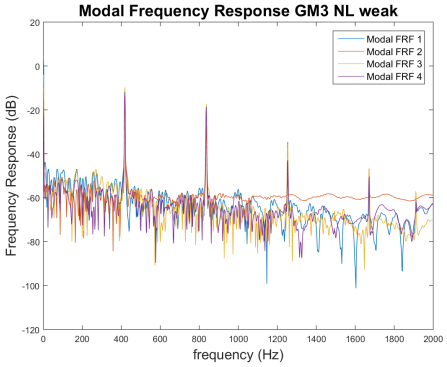
(b) Modal amplitude of forced airfoil flow



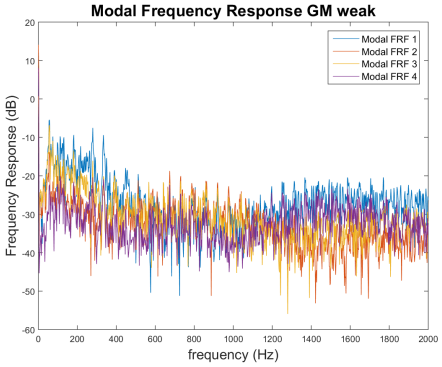
(c) System energy of baseline airfoil flow



(d) System energy of forced airfoil flow



(e) Frequency response of baseline airfoil flow



(f) Frequency response of forced airfoil flow

Figure 5.17: System characteristics for candidate models of the airfoil. Here a, c, and e represent model characteristics for a baseline airfoil flow at a 18° angle of attack, and $b, d,$ and f represent a forced airfoil flow at a 20° angle of attack

Both the baseline and forced models (shown in Figure 5.17), again quickly move from dynamics near the initial conditions to long run dynamics later in the simulation. This discrepancy between the dynamics near the initial conditions and in long run appears as a consistent behavior for the majority of models presented. Another point of interest is the bursting in the forced model. It may be possible that the model is predicting a temporary detachment of flow from the wing. This in itself highlights the difficulty of identifying valid and invalid behavior in ROMs. Even when using a relatively small number of basis functions which only captures $60\% - 70\%$ of the fluctuating energy, dynamics, such as flow detachment, may be represented by the coupled movement of several basis functions. This is illustrates how SMM could identify missing dynamics that may be difficult to detect using energy or frequency methods.

5.4 Mixing Layer Flow

The mixing layer flow's data was obtained at The University of Arizona in a $304.8mm \times 304.8mm \times 914.4mm$ closed test section of an open circuit wind tunnel [21]. The low and high speed streams were separated by a splitter plate of dimensions $304.8mm \times 304.8mm$ and a thickness of $34.92mm$. The splitter plate is tapered on the downstream side and a recess is present to accommodate the Dielectric Barrier Discharge (DBD) actuator flush to the splitter plate surface. The head loss is produced by way of two polyurethane filters and a honeycomb in order to decrease the velocity on the low speed side and provide flow conditioning. This configuration produces a velocity of $11.8m/s$ on the high speed side and a velocity ratio of $r \approx 0.28$. The Reynolds number for this flow configuration is approximately 0.28×10^6 based on the total splitter plate length and high speed velocity. Reynolds number defined by free stream velocity difference and downstream shear layer thickness is 1.53×10^3 [21].

The velocity data for the flow field was obtained using a 2D LaVision PIV system of higher resolution than the previous two empirical data sets. AC-DBD discharge plasma actuators were used to force the flow at 30 Hz and 60 Hz via modulation of a 12 kVpp and 15 kVpp, 3 kHz sinusoidal carrier frequency. In this work the focus will be on the 60 Hz 15 kVpp case. The experimental data sets are composed of 2000 instantaneous velocity fields for both baseline and forced cases. Resolution for both sets are comparable at $1.65mm$ and $1.6mm$ for the baseline and open forced cases respectively. An image of the experimental setup is provided for reference in Figure 5.18. Figures 5.19 and 5.20 show the mixing layer's mean and instantaneous vorticity

entering the chamber nearest the splitter plate on the left, and exiting the test chamber at outlet on the right.

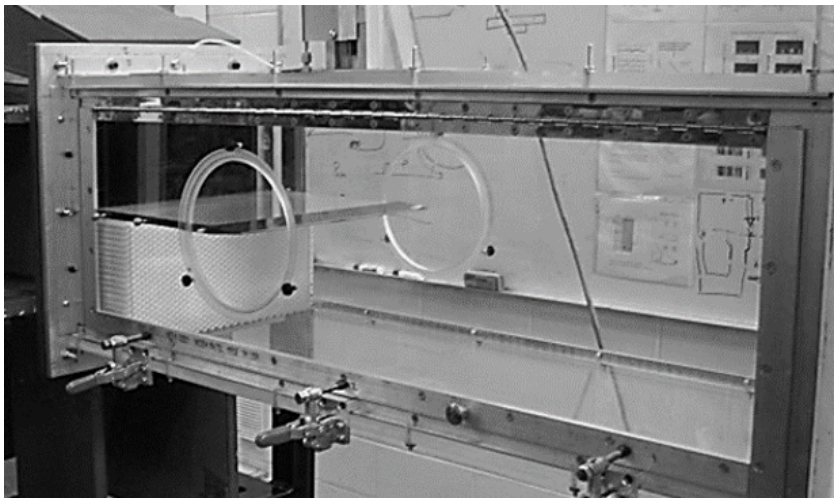
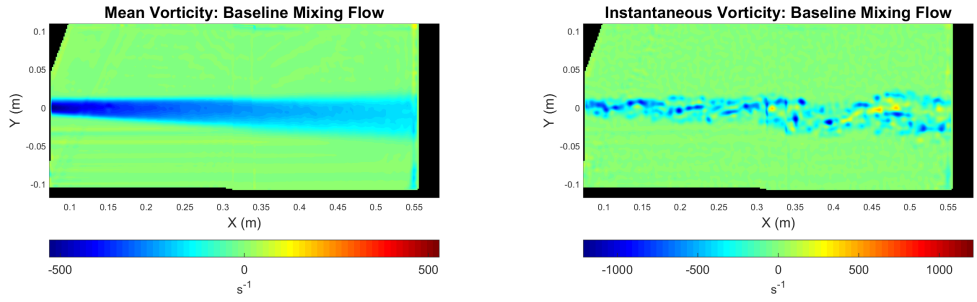


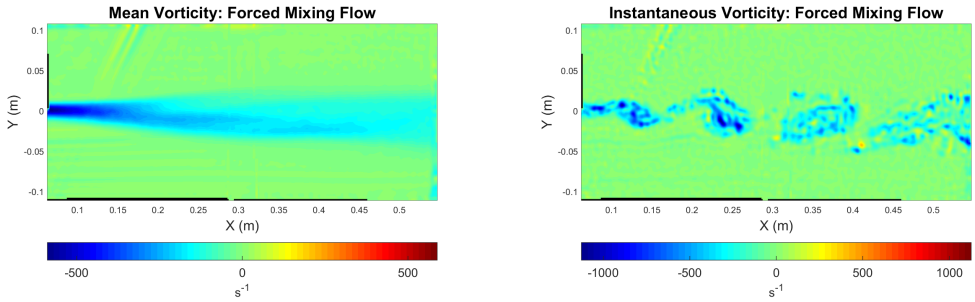
Figure 5.18: Image of the experimental mixing layer test section with flow inlet on the left courtesy of Dr. Little of the University of Arizona.



(a) Mean spanwise component vorticity field

(b) Sample spanwsie component Instantaneous vorticity field

Figure 5.19: PIV data for the mixing layer: baseline case.



(a) Mean spanwise component vorticity field

(b) Sample spanwise component Instantaneous vorticity field

Figure 5.20: PIV data for the mixing layer: forced case.

As with the airfoil and cavity flows, notable changes to the flow occur with the application of open loop forcing in both the mean and instantaneous images. The forced case shows a much wider mixing region in Figure 5.19a and distinct rolls in Figure 5.19b. The forced flow in Figure 5.20 has its first two basis functions shown in Figure 5.22 as nearly perfectly phase shifted copies of itself. In addition, forcing again moves the high energy structures to locations much sooner in the flow as compared to its baseline counterpart.

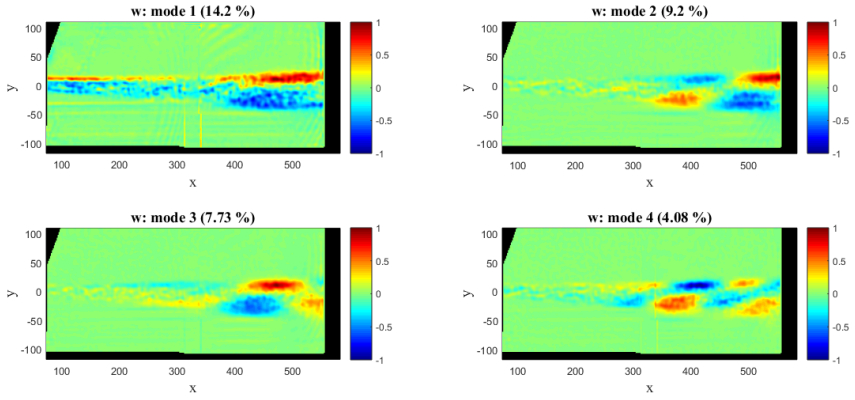


Figure 5.21: First 4 POD basis functions shown as vorticity for the mixing flow: baseline case. (magnitude scaled by maximum absolute value)

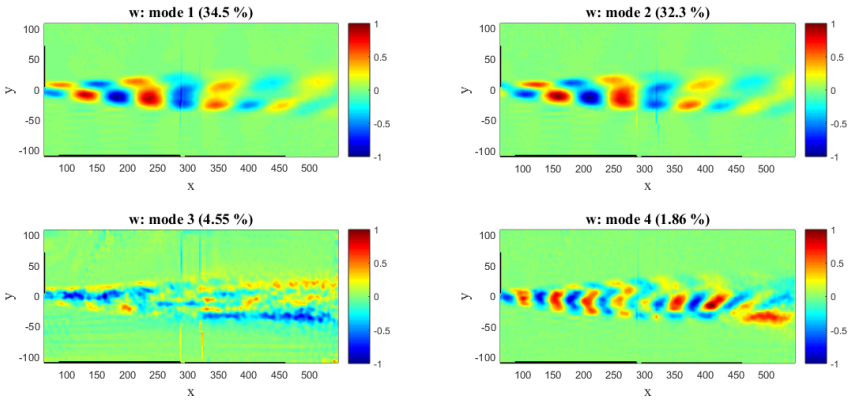
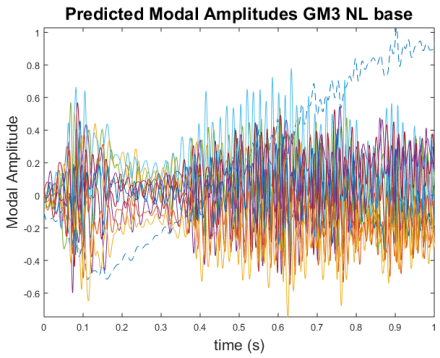
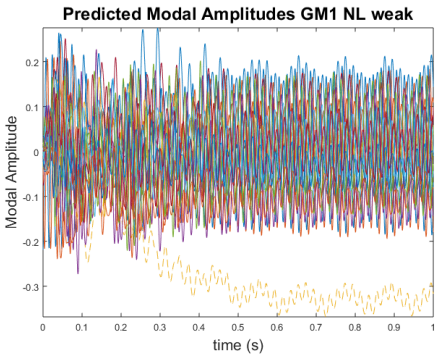


Figure 5.22: First 4 POD basis functions shown as vorticity for the mixing flow: forced case. (magnitude scaled by maximum absolute value)

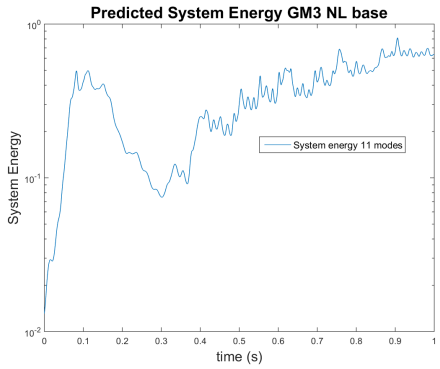
Of interest are the two vertical lines best seen in basis function 1 in Figure 5.21 of the baseline flow and basis function 3 in Figure 5.22 of the forced flow. It was found in a study by Chabot *et al.* [14] these lines in the POD modes correspond to an artifact in the collection of the PIV snapshots. Because the test chamber for this setup was fairly large, two separate cameras were used, with a composite image stitched together along the lines seen best in the indicated POD basis functions. Selectively removing these modes where the artifact was primarily captured generally improved the predicted model results [14]. Here, to keep a uniform procedure, the POD basis function where the artifacts manifested included in model derivations. The effects of these basis functions can be seen in the solutions of the selected models below in Figure 5.23. In the selected baseline flow model, the suspect basis function in Figure 5.23a is shown by the dashed light blue line. For basis function 3 of the forced case, this is seen as the dashed yellow line of Figure 5.23b.



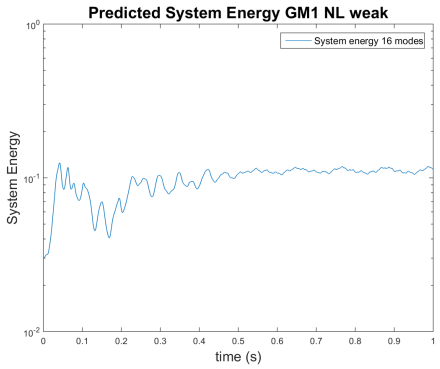
(a) Modal amplitude of baseline mixing flow



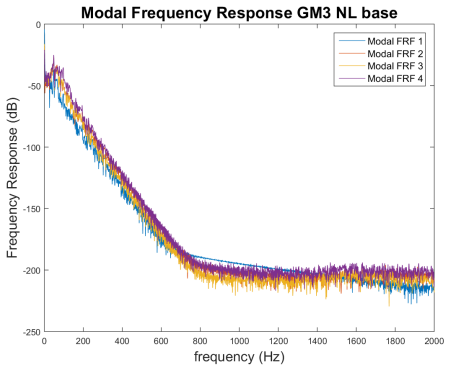
(b) Modal amplitude of forced mixing flow



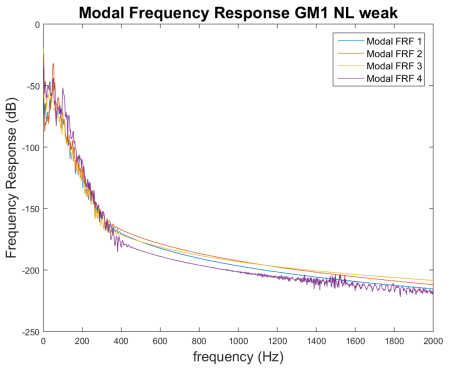
(c) System energy of baseline mixing flow



(d) System energy of forced mixing flow



(e) Frequency response of baseline mixing flow



(f) Frequency response of forced mixing flow

Figure 5.23: System characteristics for candidate models of the mixing layer. Here $a, c$and$e$ represent model characteristic for the baseline mixing layer flow, and $b, d,$ and f represent the forced mixing layer flow.

Here, the two phantom modes become apparent as they shift from the

origin in isolation from the other predicted modal amplitudes. Both models show a near logarithmic decrease frequency response across the spectrum. Also note that the forced model slightly under predicts the 60 Hz spike that should be present in the first two POD basis functions.

Chapter 6

Surrogate Markov Model Validation

In this chapter evidence is presented in an attempt to validate the process of scoring models via the SMM proposed in Chapter 3. First, observations and discussion are presented about dynamics features of the model captured by SMM and how it provides some insight into the underlying system. Next, a brief discussion of ideal cluster number selection is presented. Afterwards, the SMM is then compared to less universal validation measures, such as the distribution of modal amplitude or energy, as well as, frequency peaks. Scores for SMM will be calculated for each model as well as these three factors. It is expected there will be some correlation between better scores and better agreement between these factors, where, ideally there will be a strong correlation. Additionally, a coupled property is explored, namely the phase shift between two modal amplitudes of the mixing layer of known shift. Finally, some discussion is made of the jet data and the SMM potential time dependence.

Before diving into these aspects, a more intuitive explanation of clustering is presented. A potentially good analogy to the clustering of flow snapshots, by decomposing them using POD, would be to cluster an audio recording after decomposing it using Fourier analysis. Imagine a digital representation of a recording, for example the sound of three different instruments such as a trumpet, saxophone, and a piano each in turn playing the same A# note. Here, a Fourier fast transform could be repeatedly performed on small durations of the recording, this would break down each time segment by their frequency content. Clustering in this context, would ideally use this frequency content to identify that three distinguished clumps were present in the signal. These clumps, corresponding to each instrument, could then be labeled and identified, with the end result being the proper identification

of each instrument during the recording and the construction of the typical frequency content of each instrument. The same general concept is applied here. First, POD is used to generate a state vector of the system's flow snapshots. Then, the clustering algorithm will look for locations in the phase space, where collections of samples are naturally clumped together. These clumps, or clusters, would correspond to the timbre of the instruments in our analogy. The flow data could then be identified by which cluster it belonged to. From each cluster, the typical POD content could be determined that best represents the constituents snapshots of that cluster.

To give a sense of what a set of representative clusters could look like, the jet flow in the streamwise-normal plane is shown in Figure 6.1, with clusters based on a k-means clustering of 10 clusters. Here, the clusters represents 'typical' flow configurations in different regions of the phase space, where some level of grouping occurred. A few observations of the clustered results show that some clusters such as states $1, 5$ and 7 display a more extended plume. States 3 and 10 show a more condensed configuration. In fact, for this particular system moving from state $1 \to 2 \to ... \to 10$ would constitute the most probable path based on the estimated stochastic matrix.

While the surrogate Markov model will be tested against the traditional means of validating a ROM, as discussed above, here an effort is made to attempt to show that novel information is captured via the SMM. The first and most obvious use of the SMM is the detection of both outliers and 'holes' in phase space with respect to the original data. Detecting outliers in two dimensional data can be found in a fairly straight-forward manner using scatter plots, or for higher dimensional outliers using the Mahalanobis distance, shown in Eq. 3.45, if the data is fitted well by a joint distribution [39]. A quick example of when neither of these conditions are met can be found by using just 3 modal amplitudes of the forced mixing layer. Presented in Figure 3.1a, the forced mixing layer is shown in just 2D. This figure shows the first two modal amplitudes forming a ring, which is not well fit by a typical joint distribution and not suitable for detection via the Mahalanobis distance. Additionally, if the 3rd or 4th modal amplitudes were included, graphical methods for detecting outliers become impractical or nearly impossible for what would still be a very small model. In fact finding multivariate outliers is still an active area of research [33, 64, 65]. In GMM clustering, many multivariate normal distributions are fitted to the data. Here, the Mahalanobis distance can be used more effectively for finding outlines when calculated against each fitted component. This is due a to Gaussian mixture model collectively fitting the data set better than a single joint distribution. Outliers can be detected for k-means by defining a maximum distance from each of the cluster centroids, in effect defining a closed boundary in an $\mathbb{R}^n$ with

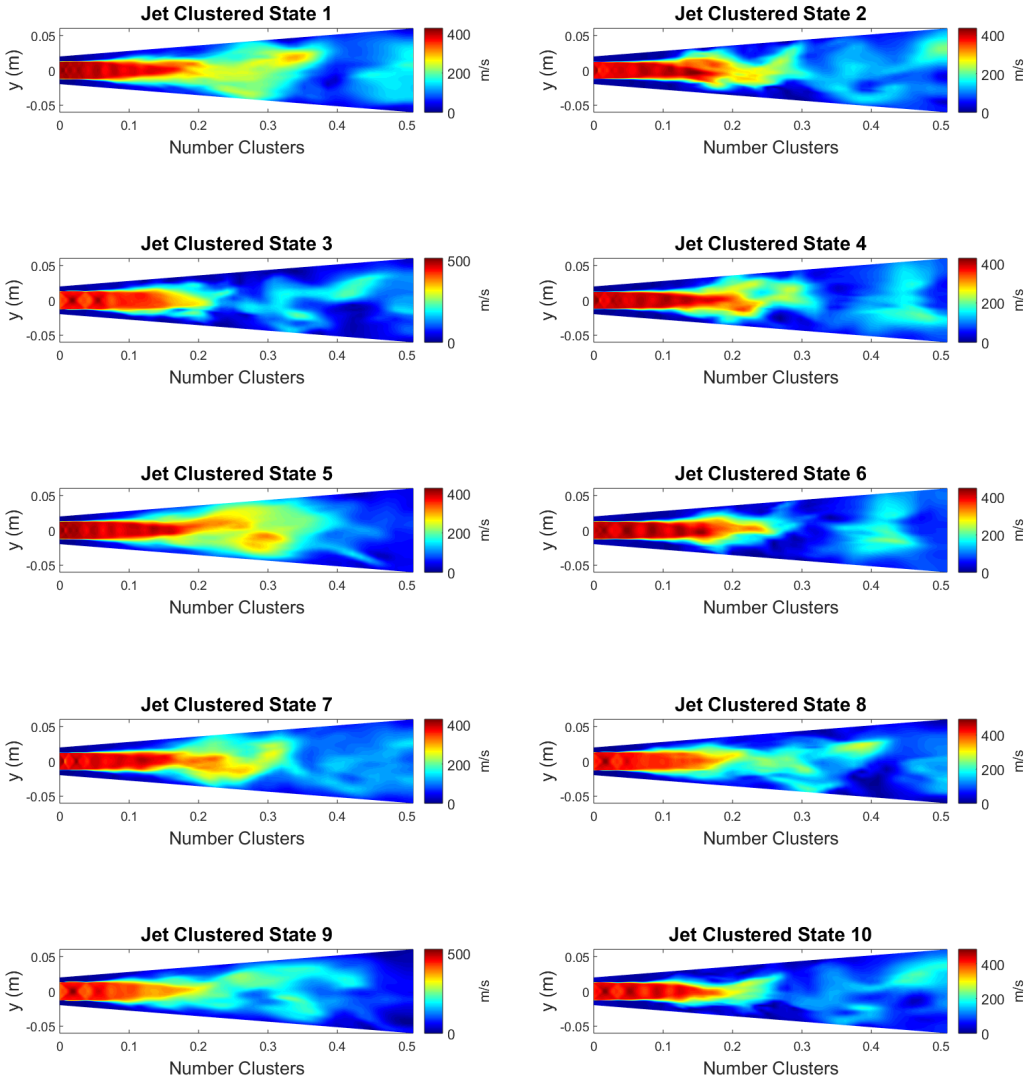


Figure 6.1: Sample clustered states for the jet in the streamwise-normal plane shown here as velocity magnitude. Clustering was performed using k-means with plot produced from defined cluster centroid of 10 clusters.

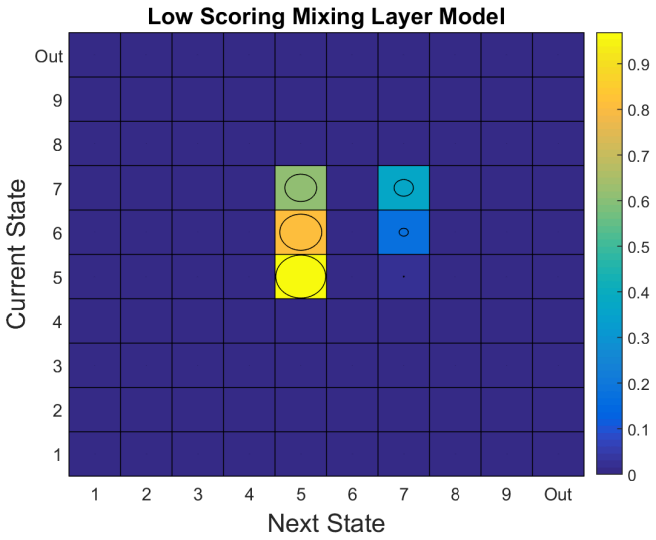


Figure 6.2: Sample stochastic matrix of the worst scoring baseline mixing layer model. This model represents a POD-Galerkin model with linear averaged modal eddy viscosity corrector, using the weak formulation of the NSE. Transition probability are shown in the colorbar with circles draw for visual aid.

points within this boundary valid and those outside, invalid.

Finding regions in phase space that are notably absent in the model's simulation, can be easily identified as well. Here these 'holes' are found by observing how the simulation is classified into the empirical clusters. If the simulation is never classified to a given cluster, this identifies flow phenomena that are not captured by the simulation. Remember that clusters are generated for the empirical data, at locations in phase space where the POD modal amplitudes clump. Below, in figures 6.2, 6.4, and 6.5, are three models shown as their estimated stochastic matrix for the baseline mixing layer flow using 11 POD basis functions scored with 9 k-means clusters. Here it will be shown, that the scoring methods capture aspects of the empirical data not depicted by energy comparison.

First, the worst scoring model by both scoring methods in Eq. 3.46 and Eq. 3.48 when clustered for both k-means and GMM is presented in Figure 6.2. This figure provides a graphical representation of Eq. 3.39, where color and the size of the circle indicate transition probabilities.

Here the system never enters the outlier mode, described in Chapter 3, but at the same time never transitions to states $1-4$, 6 8, or 9, which can be identified by the zero probabilities in those columns. Also, note that when

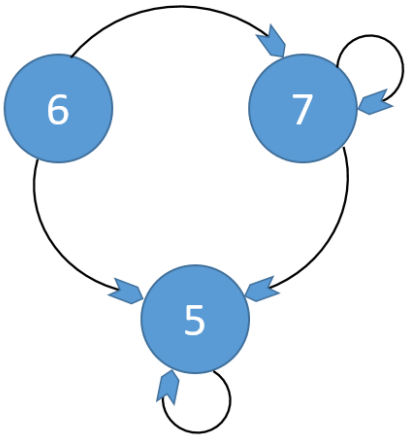


Figure 6.3: Representation of the transition matrix in Figure 6.2. States are shown as nodes and probable transitions shown as arrows.

the system enters state 5, it almost certainly stays in state 5. To help further visualize this, Figure 6.3 is provided, showcasing potential state transitions for this model.

Next, an intermediate scoring model is presented in Figure 6.4 with its description in the caption. This model shows a better spread than the worst scoring model. Here, at least initially, there is full coverage of all the states by the presence of a transition probability in all 9 valid rows. After some initial period, states 1, 5, and $7 - 9$ are never visited again indicated by the lack of any transition probabilities in these columns. The system then typically follows the chains $2 \rightarrow 3 \rightarrow 4 \rightarrow 2$ with occasional visits to state 6 which then almost certainly returns to states 2 or state 4. Finally the best scoring model is shown in Figure 6.5

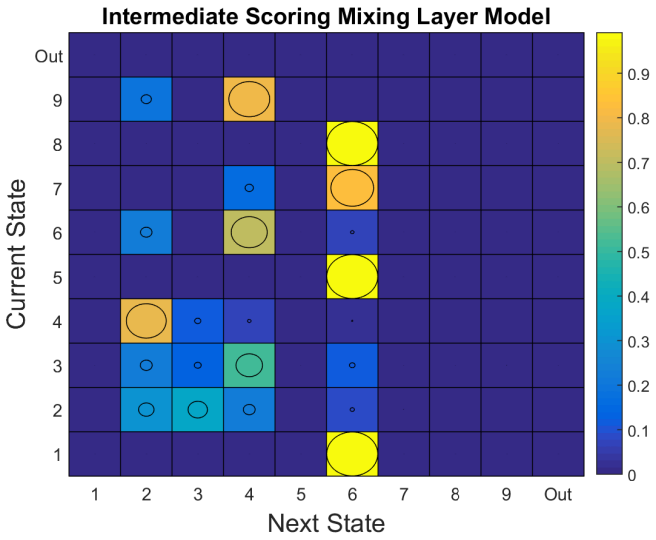


Figure 6.4: Sample stochastic matrix of a intermediate scoring baseline mixing layer model. This model represents a POD-Galerkin model with non-linear least squared modal eddy viscosity corrector, using a weak formation of the NSE. Transition probabilities are shown in the colorbar with circles draw for visual aid.

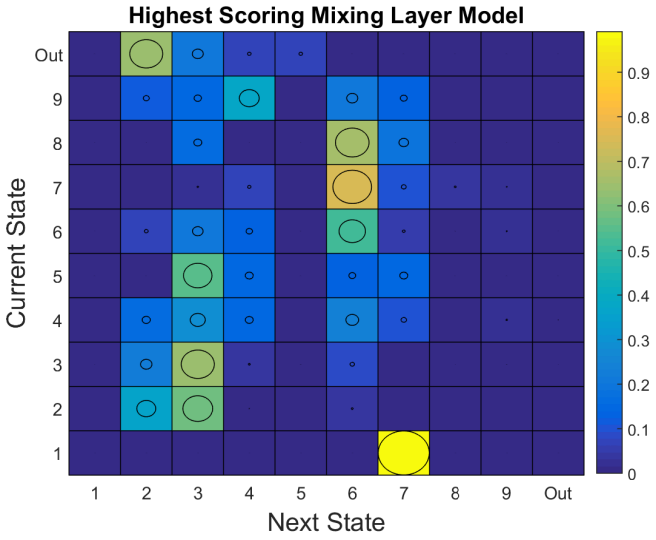


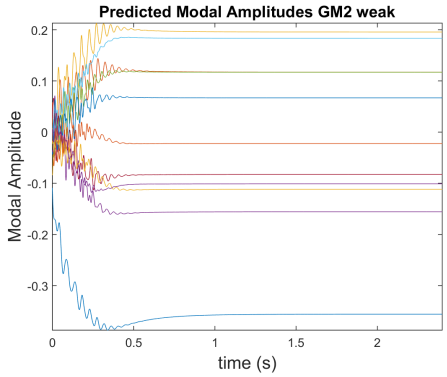
Figure 6.5: Example stochastic matrix of the best scoring baseline mixing layer model. This model represents a POD-Galerkin model with non-linear least squared modal eddy viscosity corrector using the standard formation of the NSE. Transition probabilities are shown in the colorbar with circles draw for visual aid.

This final model again predicts the solution moves in phase space away from modes 1, 5 and the outliers mode. Here, while it is less probable the system visits states 8 and 9 they do occur. Next, plots of the modal amplitudes and frequency response are shown for the three models shown above in Figure 6.6. In this case, both the worst and intermediate scoring models showed lower predicted energy compared to the best scoring model. The worst scoring model quickly finds and comes to rest on a fixed point, while the intermediate model appears to fall into a beating pattern. The best scoring model maintains a more realistic prediction of the system.

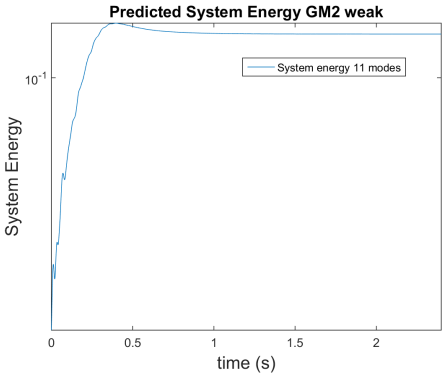
6.1 Ideal Cluster Number

In both of the clustering methods presented previously, an important unanswered question is how to select the appropriate number of clusters. In k-means and GMM clustering, data is grouped based on a local solutions of their objective functions, but neither specify how many clusters should actually be present in the data. In addition to determining an optimal number of clusters, solely in the context of clustering, there needs to be some consideration of the end goal of scoring the underlying models. Selecting too few clusters may cast too wide a net, in which case too much of the phase space will be defined as one state. This may falsely validate the underlying model when the dynamics of the model are truly significantly different. On the other end of the spectrum, selecting too many clusters runs into issues with the experimentally collected data. When the number of clusters is large there are simply not enough data points to achieve a good estimate of all the transition probabilities. Additionally, as the number of clusters increases, the argument that the Markov model represents an approximate stochastic process breaks down, as the region represented by a single cluster is constricted becomes smaller and smaller.

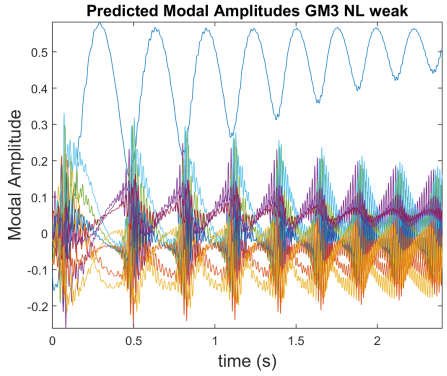
To begin to develop a guideline for the number of clusters to use, validation methods specific to clustering are performed. While some metrics of cluster selection rely on outside information, internal validation metrics, based solely on the clusters themselves, are utilized to remain as a priori as possible. A study in 2011 by Rendon *et al.* [47] investigated the relative accuracy of many proposed measures for cluster evaluation and found that for a collection of artificial data sets, internal validation metrics tended to perform better for data clustered using k-means. For this trial, two metric from the Rendon *et al.* study, the silhouette index and Calinski-Harabaz index, as well as the gap statistic, which is a formalization of the 'elbow criteria' discussed in Kaiser *et al.* [31], are performed. Each method defines a means of



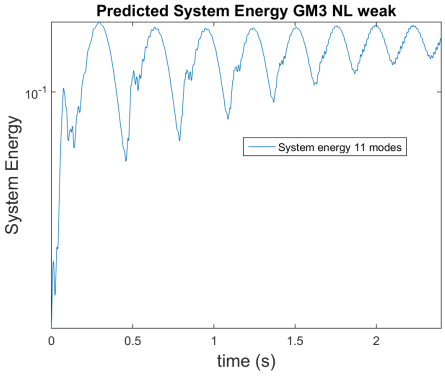
(a) Low scoring model : modal amplitude



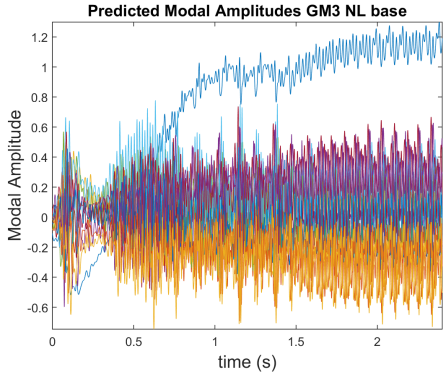
(b) Low scoring model : TKE



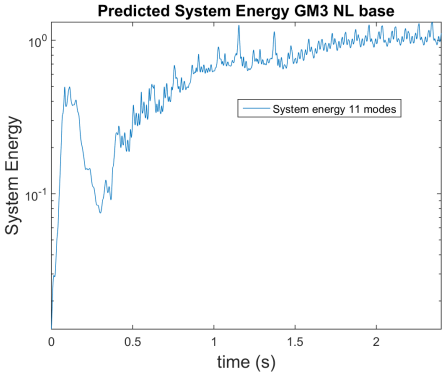
(c) Intermediate scoring model : modal amplitude



(d) Intermediate scoring model : TKE



(e) Highest scoring model : modal amplitude



(f) Highest scoring model : TKE

Figure 6.6: Comparison of modal amplitudes to system TKE for the low, intermediate and high scoring models.

identifying how many clusters best fit the data. These methods identify the best number of clusters to use by finding the maximum of some 'goodness' criteria. Detail descriptions are omitted and instead the formulating papers are provided for reference [10, 49, 63].

To investigate the optimal cluster number, a data set was taken for each of the 8 high level test conditions such as the baseline airfoil, streamwise-normal plane jet, forced mixing, etc. described in Chapter 5, on which each of the three metrics were performed. For this investigation the range of clusters investigated are restricted in line with the arguments made earlier in this section, therefore the tested range of clusters is set between 4 and 16 clusters. Based on the success of Kaiser *et al.* [31] 10 clusters was the median, based on the success of using 10 clusters for their models. For the cavity and the airfoil where many test cases exist, one data set was selected for a baseline and a forced case. Here, tables 6.1 and 6.2 show each measures predicted ideal cluster number. Plots of each of the three criteria are provided in Appendix C.

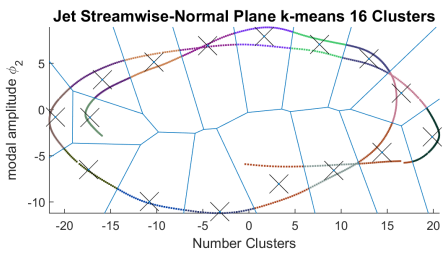
Table 6.1: Selected ideal cluster number for silhouette index, Calinski-Harabaz index, the gap statistic for each of the high level tested conditions. These evaluations are performed using k-means clustering

	Silh	Gap	CH
Jet stream-norm	15	15	16
Jet span-norm	6	6	4
Cavity base	5	6	4
Cavity forced	4	10	4
Airfoil base	4	12	4
Airfoil forced	4	10	4
Mixing base	4	4	4
Mixing forced	4	8	4

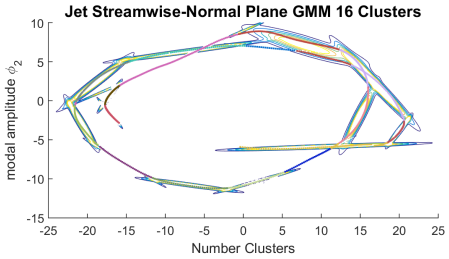
Table 6.2: Selected ideal cluster number for silhouette index, Calinski-Harabaz index, the gap statistic for each of the high level tested conditions. These evaluations are performed using GMM clustering.

	Silh	Gap	CH
Jet stream-norm	16	14	16
Jet span-norm	7	8	15
Cavity base	7	7	4
Cavity forced	4	4	4
Airfoil base	4	4	4
Airfoil forced	4	5	4
Mixing base	6	4	5
Mixing forced	4	8	4

The predicted ideal cluster number tends towards the lower end of the investigation range with the exception of the jet data. Here, the spike in values can partially be explained by the limited time integration of the jet data. Because of the small simulation time it is unlikely that the jet's snapshots would have fully converged to the long run statistical quantities. Additionally, while the empirical data inherently contains some noise related to PIV itself, which is typically compensated with the application of a Gaussian filter to the raw data, the clean numeric data tends to capture more length scales in the more energetic POD basis functions. To show why clustering of the jet data produces such high predictions of optimal clusters, the first two POD basis functions are shown clustered in Figure 6.7. It can be seen that clustering via GMM produces Gaussian components that effectively sketch an outline of the solution trajectory. Predictions for the ideal number of clusters such as this may be ideal in terms of grouping the data, but such a division would not be very useful for its application here. For the jet's clustering, solutions that didn't fall almost exactly on numerical solution would be classified as an outlier. While the ideal cluster number has been found in the context of clustering alone, further investigation will be made to see if cluster number has a notable effect on how well scores for SMM correlate with the classical validation methods presented in the next section. To test if these correlations are affected by cluster number, each model on the proceeding sections will be scored when using 4 different clusters numbers. First they will be scored using 8, 10 and 12 clusters because of the success of the group of Kaiser *et al.* when using 10 cluster. Additionally, the values found in tables 6.1 and 6.2 will be tested as well.



(a) 2D k-means clustering of Streamwise-normal jet using 16 clusters



(b) 2D GMM clustering of Streamwise-normal jet using 16 clusters

Figure 6.7: Clustering of the Jet data using 16 clusters recommended by the 3 cluster criteria.

6.2 Cross Validation

While the surrogate Markov Model provides useful information about how empirical and simulated data occupies and moves through phase space, there is a desire to know how traditional validation methods trend with model scores. Ideally, producing and scoring a surrogate Markov Model will allow a large number of candidate models to immediately be excluded from further consideration and therefore further validation. In order to use this score as a sufficient condition to eliminate candidate models from a pool of models, the scores should tend to positively correlate with improved accuracy of other classic validation measures such as energy or modal amplitude. To show this, scores are compared to the discrepancy between the mean, median, and standard deviation of turbulent kinetic energy, as well as, the first POD basis function's modal amplitude. Additionally, for a select number of data sets, where strong frequency peaks are known to occur, the predicted response will be compared for the three most energetic POD basis functions. Finally, the mixing layer has been found to have a near constant phase shift of $\pi/2$ for the first two POD basis functions [14]. This shift which can be seen in Figure 3.1 will also be investigated.

In order to explore all the factors that have been presented thus far, as potential contributors to the resulting scores, a large number of models have been generated. In total, approximately 6000 models were generated between the 53 data sets tested. Models were generated utilizing Miami's Redhawk Cluster as well as two workstation computers. Each data set produced models

for the base POD-Galerkin methods, linear and non-linear eddy viscosity correctors, and basis transformation method using GM and GM1 models as generating bases. From these model templates, specific instances of these models were created for $4-6$, $9-11$ and $13-16$ POD basis functions. Each model derived from experimental data was simulated at 400 times the experimental sampling frequency of $10Hz$. This resulted in the interleaving of 400 Markov chains for approximately ~ 2.5 seconds, cumulatively giving four times as many cluster transitions as were present in the experimental data. Because of the very high sampling rate (250kHz) of the jet, simulations were simply performed for, four times as long as the numerical data, to again give one chain of four times as many transitions. Models that were not able to produce bounded solutions for the requested integration period were eliminated from consideration. All bounded models were collected, with each model scored using the two clustering methods, k-means and GMM clustering, as well as the two scoring method, σ_l and σ_d, described in Chapter 3. For the presented models, correlations are only shown for the optimally predicted number of clusters shown in tables 6.1 and 6.2. For the quantity of interest, the absolute difference between the value predicted by the model and the empirical data is calculated and shown. Finally, if a given data set was able to produce at least 10 bounded models, the scores were correlated and a linear regression line fitted. Ideally, to at least partially confirm that in addition to the information provided about where and how each model moves through phase space, positive correlations will be present indicting a general increase in model 'goodness'.

In order to keep the figures more readable, legends have been omitted from individual plots in Subsection 6.2.1 and instead will have readers refer to Table 6.3 as the legend for these plots.

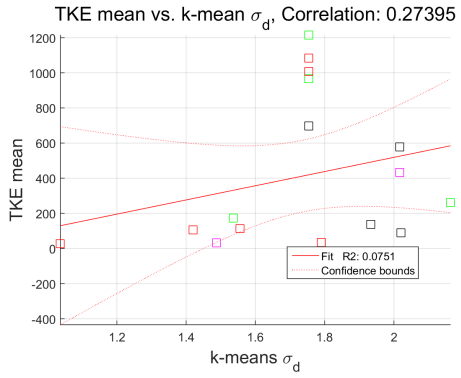
Table 6.3: Legend for figures in Subsection 6.2.1

	GM	GM1	GM2	GM3
Base model	□	□	□	□
Nonlinear Eddy Visocity	-	×	×	×
Basis Transformation	-	○	○	○

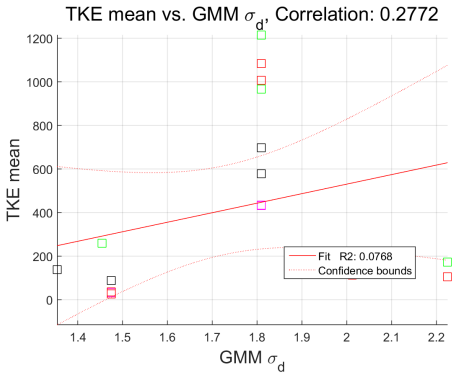
6.2.1 Individual Comparisons

First, the most ubiquitous validation method for ROMs in the authors view, the turbulent kinetic energy, is compared to the model scores. Because the number of comparisons that can be made is still extremely large, one data

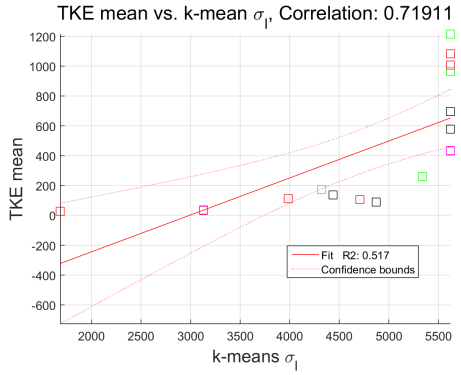
set is shown for the four scoring methods. The data set presented here is for a baseline 18° airfoil flow. Figure 6.8 shows scatter plots for the scores vs. the mean TKE with calculated correlations shown in the title of each graph. The same procedure was performed for median and standard deviation with those figures to be found in Appendix D.



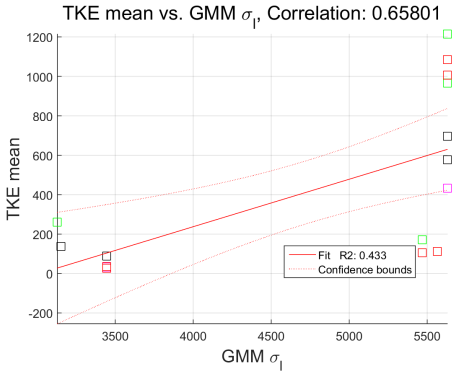
(a) Airfoil mean TKE : k-means σ_d.



(b) Airfoil mean TKE : GMM σ_d.



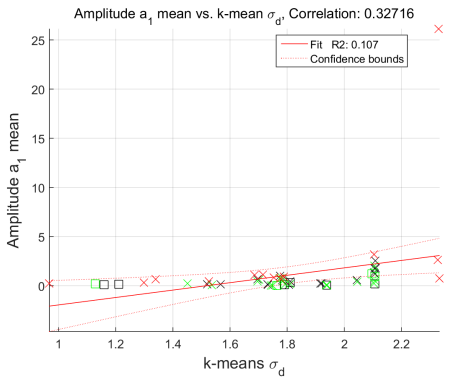
(c) Airfoil mean TKE : k-means σ_l.



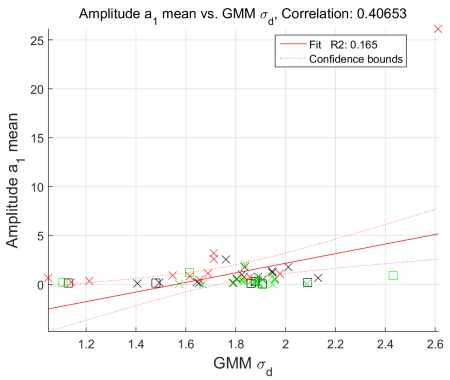
(d) Airfoil mean TKE : GMM σ_l.

Figure 6.8: Scatter plots of the four score producing methods for the system's mean turbulent kinetic energy for a 18° baseline airfoil flow.

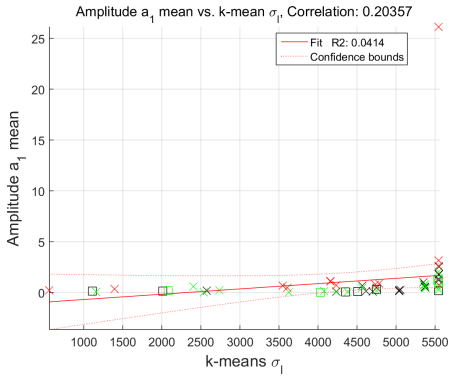
Looking at the calculated correlations a general trend is established for this particular test case, with σ_l correlating more strongly to the system's TKE. Next, a comparison of just the first modal amplitude is made to the scoring methods. For this comparison, a baseline cavity flow is selected, with correlation to the mean again presented here with median and standard deviations located in Appendix D.



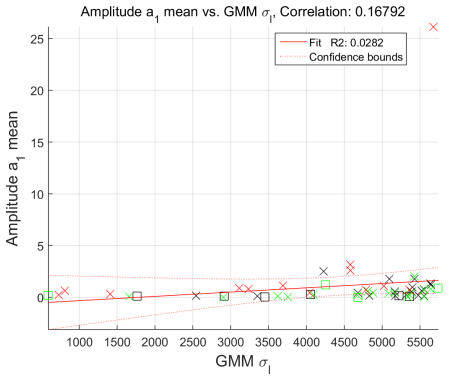
(a) Cavity mean modal amplitude a_1 : k-means σ_d.



(b) Cavity mean modal amplitude a_1 : GMM σ_d.



(c) Cavity mean modal amplitude a_1 : k-means σ_l.

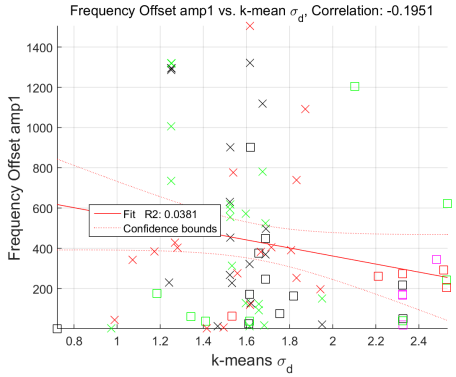


(d) Cavity mean modal amplitude a_1 : GMM σ_l.

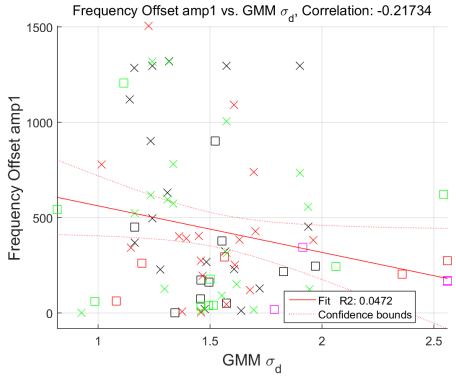
Figure 6.9: Scatter plots of the four score producing methods for the system's first modal amplitude of a baseline cavity flow.

Again a small positive correlation is established, with σ_d now showing better correlation with these results. Note that one model that scored poorly in all measures, remained bounded at an usually high energy. While it clearly does not follow the linear regression line, it would be part of the pool of models that will identified as invalid and would be eliminated from further consideration. Continuing with the comparisons, a forced cavity flow will now be shown using forcing at $1830Hz$ at 400 volts. In order to perform frequency comparisons, the 10 largest peaks were found in the data for the first 3 POD basis functions. Here, it is assumed that the system's prominent frequency components will manifest in the largest basis functions. Peaks were detected for each of the three POD basis functions, with each peak given a 25Hz buffer on either side to avoid repeatedly detecting the same peak. From this pool of detected peaks, the peak found to be the closest to the target

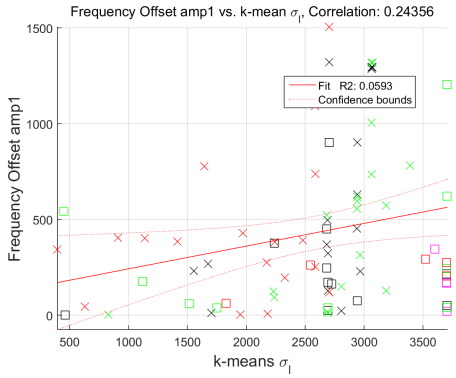
peak of interest, was selected as representing the closest peak. In this case the forcing frequency of 1830 Hz was selected as the target frequency. These scatter plots are shown here in Figure 6.10.



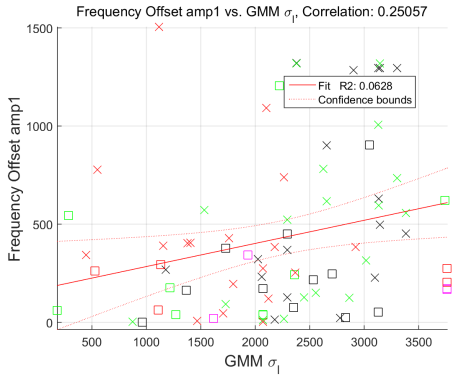
(a) Cavity detected frequency peak difference : k-means σ_d.



(b) Cavity detected frequency peak difference : GMM σ_d.



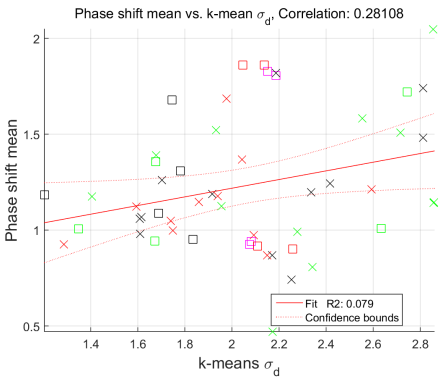
(c) Cavity detected frequency peak difference : k-means σ_l.



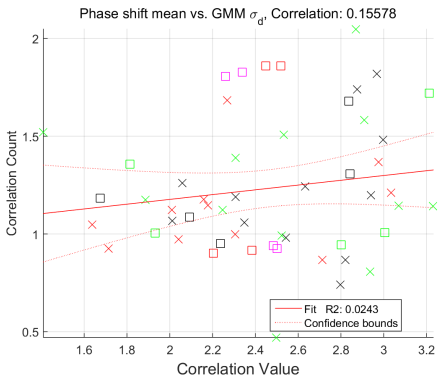
(d) Cavity detected frequency peak difference : GMM σ_l.

Figure 6.10: Scatter plots of the four score producing methods for the system's frequency response discrepancy for a forced cavity flow.

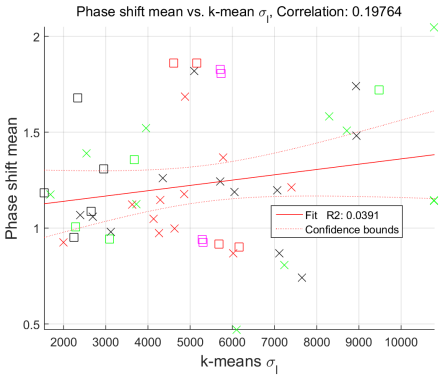
Here, there is no longer consistency between the scores, it appears that overall that SMM does not correlate with frequency response. Lastly, the phase shift described at the beginning of the section for the mixing layer is explored. In these plots, the mean phase shift for each model is compared against the target value $\pi/2$ with the absolute difference taken as the discrepancy.



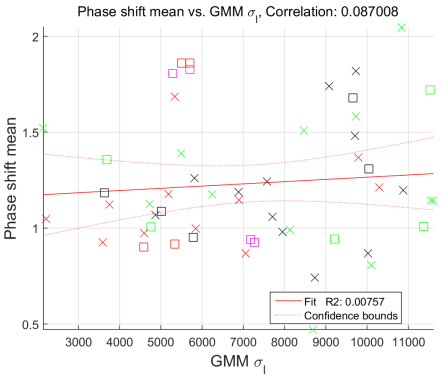
(a) Mixing layer mean phase discrepancy : k-means σ_d.



(b) Mixing layer mean phase discrepancy : GMM σ_d.



(c) Mixing layer mean phase discrepancy : k-means σ_l.



(d) Mixing layer mean phase discrepancy : GMM σ_l.

Figure 6.11: Scatter plots of the four score producing methods for the mixing layers mean phase shift discrepancy.

Here, there is a very weak correlation towards the expected phase angle with decreasing score. While this can not be taken by itself as much evidence for the relations it is promising.

6.2.2 Bulk Comparisons

In order to give real credibility that scoring via a surrogate Markov model is in fact reflective of the underlying system, care must be taken to remove any bias in the selection of the shown results. In the previous subsection weak correlations for the selected data sets were shown with the exception of the frequency peak agreement. While these results consistently showed the presence of a positive correlation, the individual correlations were not strong. In order to show that the trend presented here, exist outside the

Table 6.4: Averaged Correlations for TKE vs. Scoring method

	k-means σ_d	k-means σ_l	GMM σ_d	GMM σ_l
TKE mean	0.158	0.184	0.287	0.159
TKE median	0.176	0.219	0.304	0.182
TKE std	0.121	0.051	0.182	0.031

Table 6.5: Averaged Correlations for a_1 vs. Scoring method

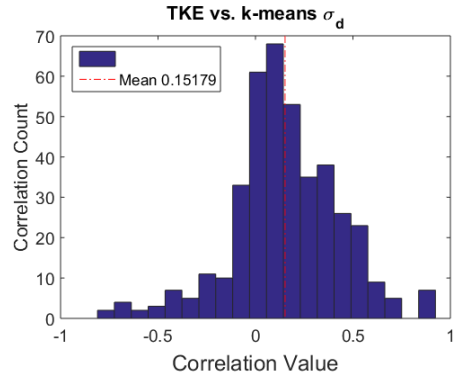
	k-means σ_d	k-means σ_l	GMM σ_d	GMM σ_l
a_1 mean	0.169	0.219	0.304	0.241
a_1 median	0.164	0.218	0.286	0.236
a_1 std	-0.041	-0.013	0.045	0.006

selected data sets, results were aggregated across all bounded models produced. As well as providing additional evidence to the trends established in the last section, individual factors such as scoring method, or cluster number can be compared against the global trends. This information can be used to find which parameters and methods work best to produce the strongest relations. Here, results for the turbulent kinetic energy, modal amplitudes, and frequency peaks will be revisited.

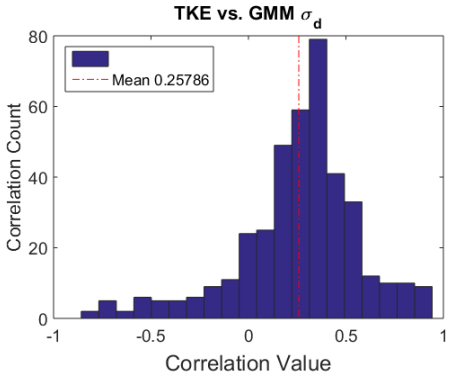
Following the same pattern as with the previous section, the turbulent kinetic energy is presented first, with correlations for the mean, median and standard deviation lumped together for all cluster numbers tested. Here, each data set producing at least 10 bounded models, was included as a valid correlation and plotted on a histogram comparing the correlation value to the number of observed correlations of that value shown in Figure 6.12. Afterwards, the lumped scores are presented in Table 6.4 showing the mean value of the individual statistical components.

Figure 6.12 shows a peak in the number of correlations found moving just into the positive range for each of the 4 scores. In addition to showing the strongest composite mean, GMM σ_d shows the strongest component correlations as well. In all four cases, standard deviation shows the weakest overall relation as seen in Table 6.4. Moving to the next comparison point, histograms in Figure 6.13 and component breakdowns in Table 6.5 for the modal amplitude relations to the first POD basis function are presented.

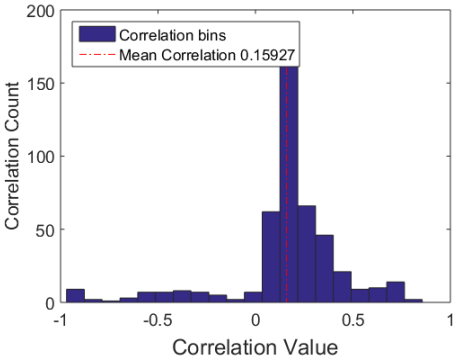
A similar but weaker trend is again observed for the first POD basis function, with GMM σ_d showing the strongest correlation. One possible explanation for the overall stronger correlations for the TKE and a_1 is related



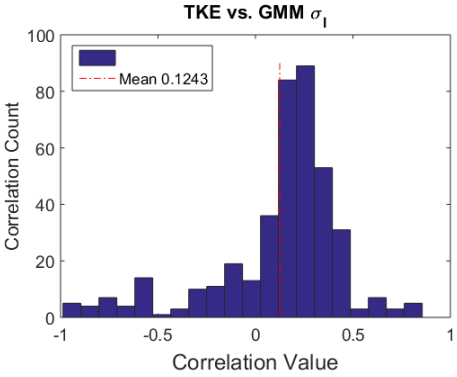
(a) TKE statistics for k-means σ_d.



(b) TKE statistics for GMM σ_d.

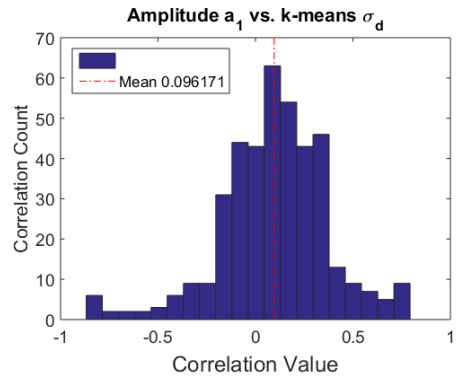


(c) TKE statistics for k-means σ_l.

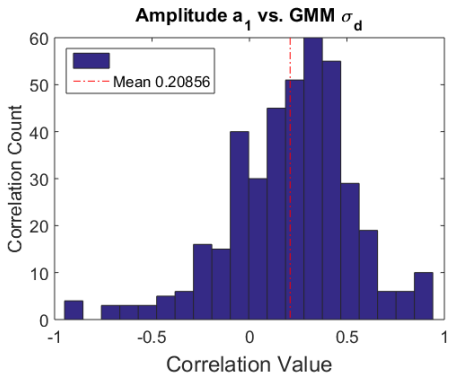


(d) TKE statistics for GMM σ_l.

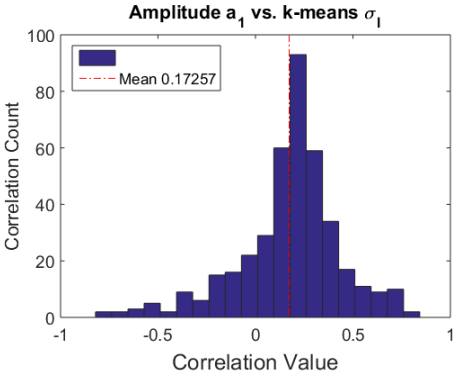
Figure 6.12: Histogram of the occurrence correlations of a given value for each score compared to the TKE measures



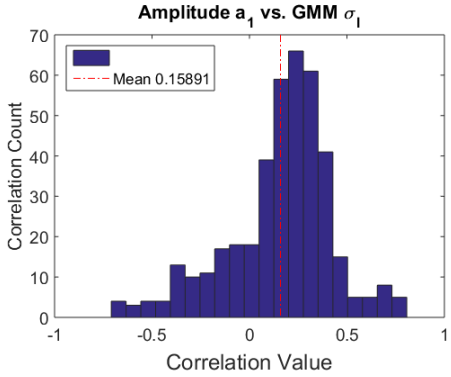
(a) Modal amplitude statistics for k-means σ_d.



(b) Modal amplitude statistics for GMM σ_d.



(c) Modal amplitude statistics for k-means σ_l.



(d) Modal amplitude statistics for GMM σ_l.

Figure 6.13: Histogram of the occurrence correlations of a given value for each score compared to the modal amplitude measures.

to how the clusters are generated. Because POD decomposes the flow into a set of principle flow features, each feature is centered about the origin. This would naturally lead to clusters grouped in the vicinity of the origin. A shift in the solution trajectory away from the origin will be repeatedly penalized during the scoring, for predicting low probability of improbable transitions. Such a shift naturally raises the systems TKE. Similarly, the modal amplitude by definition should be centered on or near the origin. On the other hand, models predicting solutions closely centered on the origin, inherently allow classification of a range of points to the same cluster. This would allow models to range to some degree in amplitude, and be viewed effectively as the same to the SMM. While changes to the standard deviation of the TKE more broadly shift solution trajectory, a small change in one component will likely have little effect on how the model is classified. This is one possible explanation for why there is some correlation for the TKE standard deviation compared to the negligible trend for a_1 standard deviation.

Lastly, the frequency peak prediction will again be revisited with the bulk results shown as a histogram in Figure 6.14. This figure shows quite definitively, that for the methodology of peak detection used in this work, there is virtually no relation with SMM scoring.

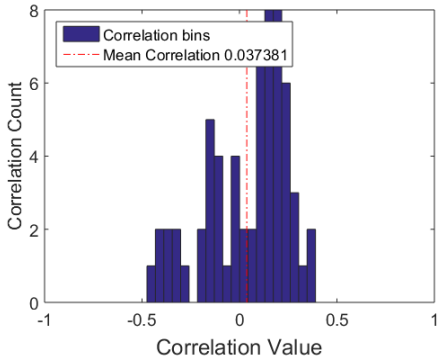
From the bulk data, results were filtered to look at how the optimally selected number of clusters fared compared to the full aggregate. Here simple percentage changes, are shown in Table 6.6

Table 6.6: Change in correlations using optimal cluster compared to aggregate

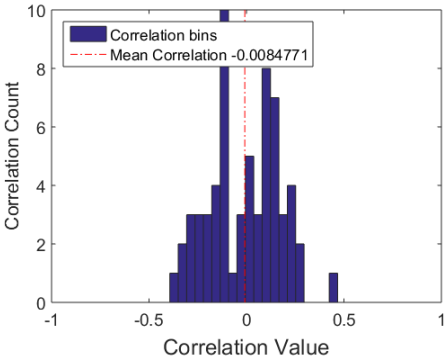
	k-means σ_d	k-means σ_l	GMM σ_d	GMM σ_l
TKE	54.3%	15.5%	30.4%	24.1%
a_1	16.4%	21.8%	28.6%	23.6%
frequency	−4.1%	−1.3	4.5%	0.6%

For TKE and the modal amplitude of a_1 using the optimally selected cluster in tables 6.1 and 6.2 it appears to have a notable improvement in relations. On the other hand the frequency peak agreement was not particularly affected by using the optimal cluster number.

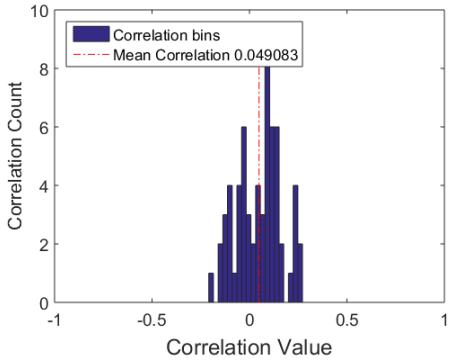
Throughout this section, it has been shown that when looked at collectively, better scores produced from σ_d or σ_l and clustering from k-means or GMM show either increased agreement or no relation to the three alternative validation methods. To reiterate, the goal of SMM is not to try predict these three validation methods with one alternative validation method, instead it



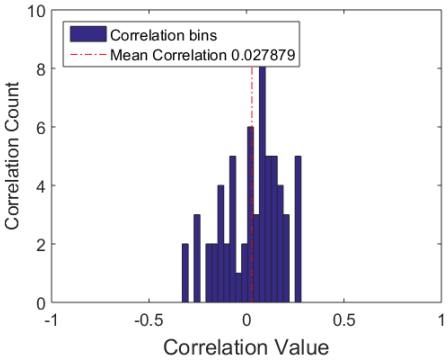
(a) Peak detection for k-means σ_d.



(b) Peak detection for GMM σ_d.



(c) Peak detection for k-means σ_l.



(d) Peak detection for GMM σ_l.

Figure 6.14: Histogram of the occurrence correlations of a given value for each score compared to the detected peak

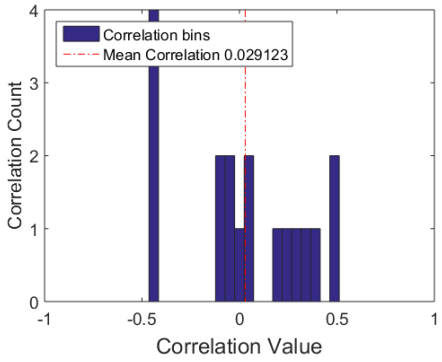
is meant to indicate how closely a simulation's dynamics replicate the original data's dynamics, as outlined in the beginning of this chapter. Here, this cross-validation is provided to show that in addition identifying missing or incorrectly predicted dynamics, that better scores tend to indirectly indicate improved agreement with validation methods currently used in the field.

6.3 Time Step Dependence

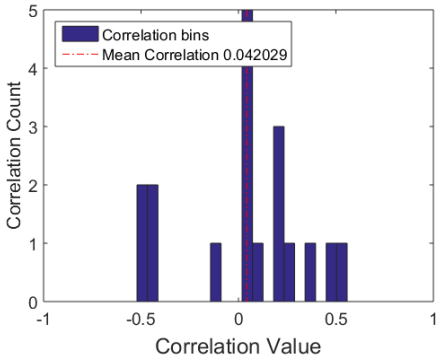
The last point of investigation in this work, probes the argument made in Chapter 3 for approximating the system as a stochastic process. This is the underlying assumption that makes this approach viable. Here, it is known that the three experimentally collected data sets have sampling rates that make each image in the ensemble time uncorrelated. On the other hand, the numerically generated jet data has time steps that makes a single image time correlated to many images following and preceding the image of interest. While modeling the jet with the incompressible Navier-Stokes equations will introduce some inaccuracies, the correlations scores indicate if the characteristics in question simply trend together. Here similar to the previous section, histograms aggregating the correlations are presented for the jet and the cavity for comparison. The TKE of the jet is shown in Figure 6.15 with the cavity's TKE shown in Figure 6.16, immediately followed by the modal amplitude of the first POD basis function jet seen in Figure 6.17 and cavity in Figure 6.18. Because the original jet data did not look at predicted frequencies no comparison will be made there.

While there are not as many correlations to draw from, there does not appear to be any trend present for the jet data. Compare this to the cavity data, which shows essentially the same trends as the aggregate over all the data sets shown previously in Subsection 6.2.2. Here, the estimated stochastic matrix of a sample baseline cavity data set and the streamwise-normal plane jet are provided in figures 6.19 and 6.20, to give further insight into the histograms presented. For these two data sets, 10 clusters were generated by k-means to estimate the stochastic matrix.

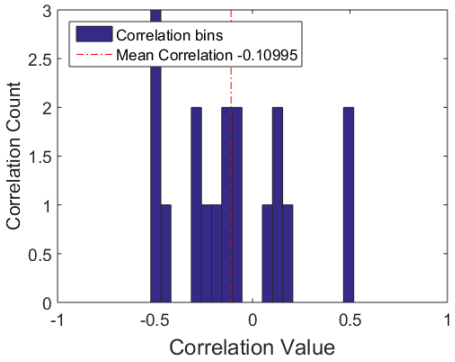
The cavity's stochastic matrix shows that from a given state there is some possibility of transitioning to any other state. While every transition is possible, some transitions are more probable with some examples including $6 \rightarrow 2$, $4 \rightarrow 3$, or $9 \rightarrow 10$. From an intuitive standpoint, such a stochastic matrix should be expected. Collecting a velocity field sample and then taking another sample such that the two samples are time uncorrelated should be unpredictable without additional knowledge. Because each time step is classified to a cluster, some knowledge is retained about the models current



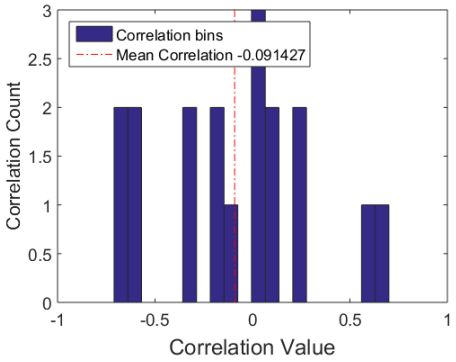
(a) TKE statistic for the jet data using k-means σ_d.



(b) TKE statistic for the jet data using GMM σ_d.

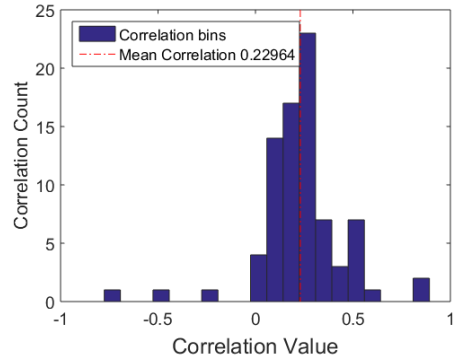


(c) TKE statistic for the jet data using k-means σ_l.

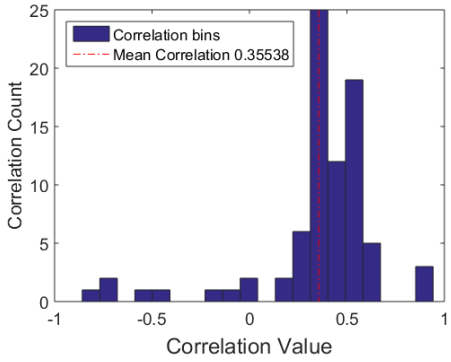


(d) TKE statistic for the jet data using GMM σ_l.

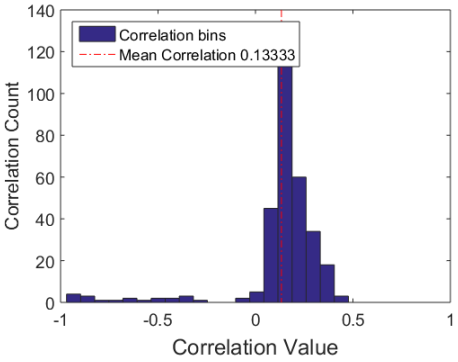
Figure 6.15: Histogram of the occurrence of correlations of a given value for each score compared to the TKE measures of the jet data.



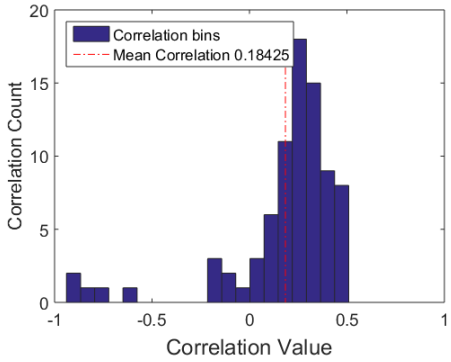
(a) TKE statistic for the cavity data using k-means σ_d.



(b) TKE statistic for the cavity data using GMM σ_d.

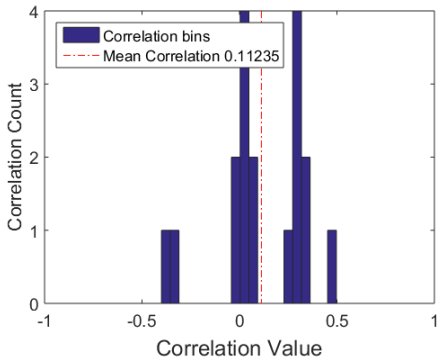


(c) TKE statistic for the cavity data using k-means σ_l.

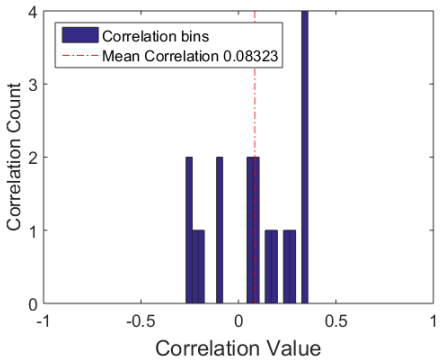


(d) TKE statistic for the cavity data using GMM σ_l.

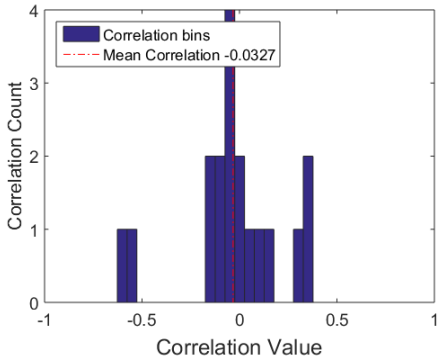
Figure 6.16: Histogram of the occurrence of correlations of a given value for each score compared to the TKE measures of the cavity data.



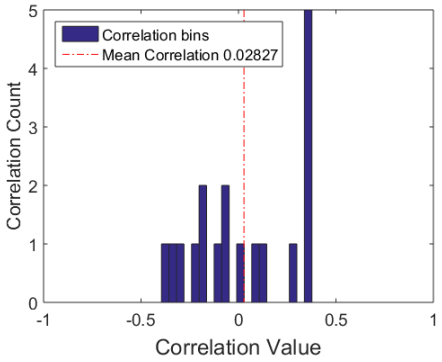
(a) Modal amplitude $a1$ statistic for the jet data using k-means σ_d.



(b) Modal amplitude $a1$ statistic for the jet data using GMM σ_d.

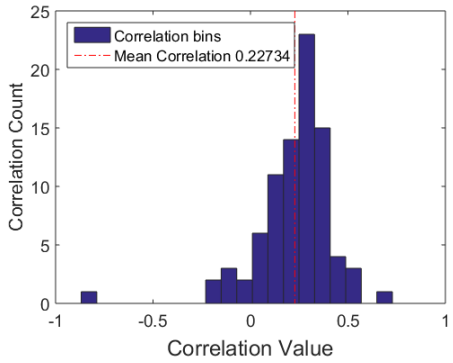


(c) Modal amplitude $a1$ statistic for the jet data using k-means σ_l.

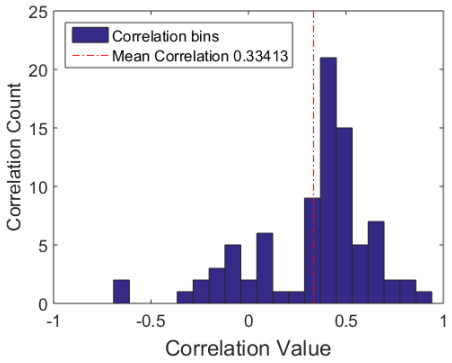


(d) Modal amplitude $a1$ statistic for the jet data using GMM σ_l.

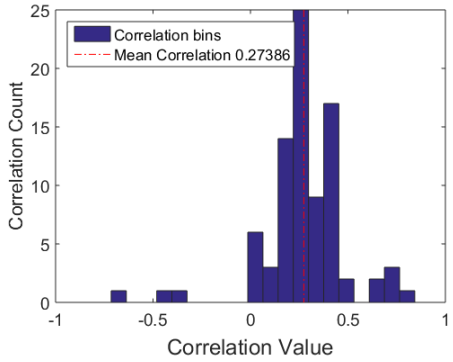
Figure 6.17: Histogram of the occurrence of correlations of a given value for each score compared to the modal amplitude $a1$ measures of the jet data.



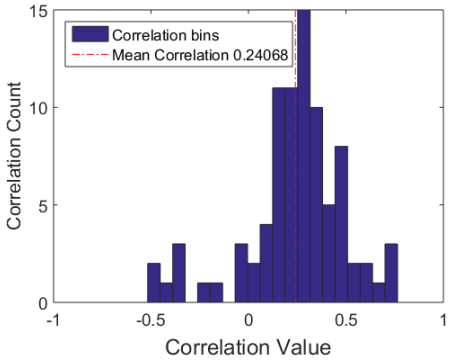
(a) Modal amplitude $a1$ statistic for the cavity data using k-means σ_d.



(b) Modal amplitude $a1$ statistic for the cavity data using GMM σ_d.



(c) Modal amplitude $a1$ statistic for the cavity data using k-means σ_l.



(d) Modal amplitude $a1$ statistic for the cavity data using GMM σ_l.

Figure 6.18: Histogram of the occurrence of correlations of a given value for each score compared to the modal amplitude $a1$ measures of the cavity data.

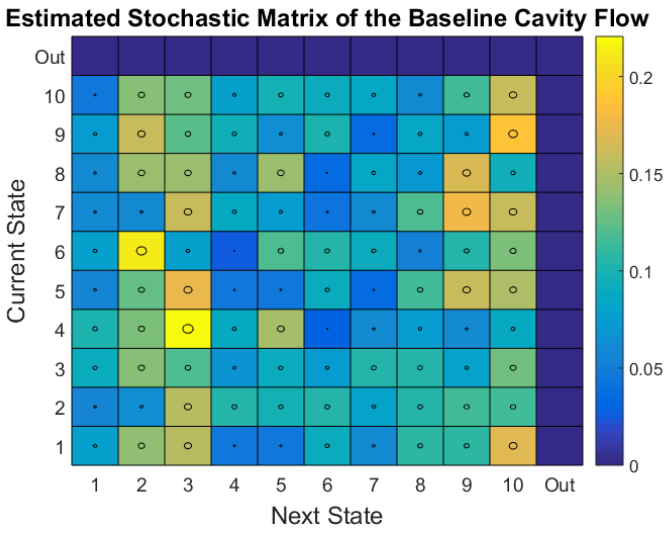


Figure 6.19: Estimated stochastic matrix from a baseline cavity data set using k-means for 10 clusters

location in the model or phase space. Imagine now, simulating a model from every possible initial condition within that cluster. After integrating each initial condition for one time step, individual solutions may land anywhere in the phase space, but because initial conditions were clumped, it would be expected that some clumping would occur in the results of the integration. Here, the variability of the transition probabilities shown in Figure 6.19 are the realization of this clumping.

In the case of the jet, the correlated data quickly stands out as compared to the cavity data. Looking back at Figure 6.7, each time step is typically so short that it remains within a single cluster. In this case the argument that this system can be approximated as a stochastic process breaks down. Given knowledge that the system is in a particular state dictates that it will almost certainly remain in that state, or move to a state that it borders. Because of this, SMM does not appear effective for time correlated data.

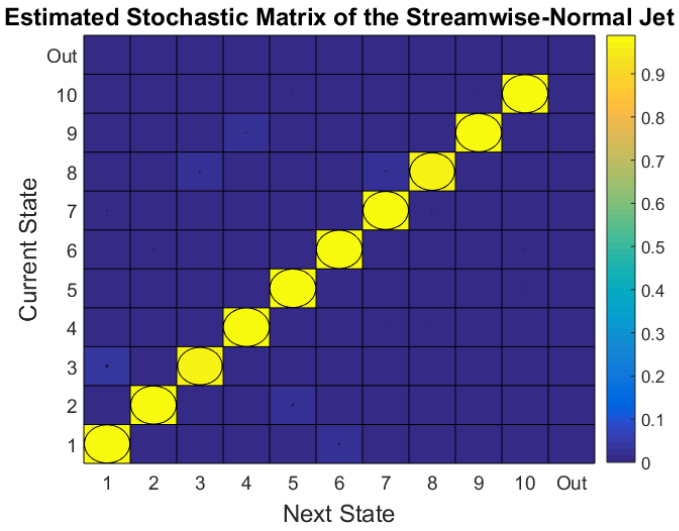


Figure 6.20: Estimated stochastic matrix from a streamwise-normal plane jet data set using k-means for 10 clusters

Chapter 7

Conclusions and Future Work

This work presents the framework for a data driven, ROM validation procedure for steady flow by use of the surrogate Markov model. In principle, validation is performed by generating a simplified, but representative model from both the empirical and simulated data. The empirical data source is first clustered to attempt to find the most typical flow formations. Next, simulated data is assigned to a clustered or outlier state, based on which state it is most similar to. Given that the data is time uncorrelated, arguments are made that the system can be approximately modeled as a Markov process. Matrices representing the probability of moving between states are then estimated, for both empirical and simulated data. These matrices are then used to compare the underlying data sets.

Using the machine learning tools of clustering and classification, affords a holistic view of the system without being hung up on whether the simulation produced an identical reconstruction or not. Instead of attempting to compare the whole system at each time step, or simply looking at a small set of value such as energy or frequency content that summarized some aspect of the flow, this procedure looks to identify the models that are most similar. Estimation of the systems stochastic matrix can quickly identify if the model is missing entire regions of phase space occupied by the empirical data.

It has been shown that lower scores for both clustering schemes and scoring methods indicate increasing agreement in both center and spread of the system's TKE and individual modal amplitudes. Key frequency peaks were shown to be uncorrelated to the SMM scorings for the data sets and models tested in this work. It was also found that using a smaller set of clusters predicted by the optimal cluster criteria, strengthen the relation between scores predicted for the SMMs and TKE and modal amplitudes. Finally, out of the clustering procedures, k-means and GMM, and the two scoring procedures, σ_l and σ_d it was found that scores for σ_d produced from GMM clusters were

most strongly correlated to the TKE and modal amplitude. Given information about the phase space and it's transitions, and that, lower scores for the SMM indicate that model aspects are either improving or were unrelated, it is believed that lower scores for the SMM generally indicates better models. It is believed that the SMM provides sufficient information for a selection of models, those models scoring the worst can be excluded from further validation consideration.

7.1 Future Work

There are a few interesting questions that can be explored based off this work. The most obvious continuation of this work would be to simply investigate additional models or test cases. While POD-Galerkin models were used and provided the smallest possible state vector to cluster, it would be worth an investigation into the growingly popular Dynamics Mode Decomposition [50, 54], to identify if this framework can be applied there. More broadly this scheme may be applicable to validation of models outside of reduced order models. For example, LES or DNS simulations with experimentally gathered PIV data could potentially be compared in this framework. First, the PIV data could be decomposed using POD, with the LES or DNS data projected onto this basis. Given that a small projection residual was found, transition matrices could again be estimated and the described score methodologies applied.

Beyond further testing, one area of ambiguity existing in this work is the best procedure for defining an outliers state. In this thesis, a critical distance was used to define the maximum distance any data point could be from a cluster, before it was assigned to the outlier cluster. This distance was found by taking an arbitrary multiple of the largest distance between any two cluster centroids when k-means clustering was used. Similarly, an arbitrary Mahalanobis distance was used as critical distance in GMM clustering. While these definitions worked well enough for this work a more logically deduced procedure could be defined. One procedure that was researched but not implemented for GMM clustering, would give a value to each simulated data point analogous to a z-score for standard deviation of normal distribution [1]. The key to any outlier definition in this context would be to define the boundary based on properties of generated clusters, while also remaining computationally cheap.

Finally while it was shown that highly time correlated data such as the jet, completely breaks down the arguments for a stochastic process, the time scales that it does hold for, are not well defined. On one end, there is data

that is effectively deterministic in this framework such as the jet data, where many time steps occur within the same cluster. At the other end of the spectrum, given very large time steps; knowledge about the current state and its approximate location in phase space become meaningless. As was described in Section 6.3, because some information is retained about where in phase space a state of the system is currently located, the next transition will most likely have one or two transitions that are much more probable than the others. As the time step between samples increases eventually each states transition probabilities will converge to those of the stationary distribution described in Appendix B. Here, a well defined set of bounds would be valuable to decide, before scoring, if the assumptions made in this work will remain valid.

Appendices

Appendix A

Basis Transformation Algorithm

The numerical procedure proposed by Balajewicz et al. [5] can be thought of as a post-processing technique to produce a basis that has natural energy balancing properties. As shown in Chapter 3 this energy balance is found by a root finding procedure based on a critical transfer term ϵ. Presenting the equation for this balance is shown below:

$$r(\epsilon) := \sum_{i=1}^{n} \tilde{\Lambda}_{ii} - \langle \sum_{i=1}^{n} \tilde{a}_i^2(t) \rangle \tag{A.1}$$

$$\epsilon = \sum_{ij}^{n} L_{ij}\Lambda_{ij} \tag{A.2}$$

Where both $\tilde{\Lambda}$ and $\tilde{a}$ are found through numerical processes. First, the basis is 'minimally rotated' using the following optimization formulation.

$$\begin{aligned} \underset{X\in\mathbb{R}^{N\times n}}{argmin} \quad & \sum_{i=1}^{n} \left(\lambda_i - \left(X^T \langle a_i a_j \rangle X \right)_{ii} \right) \\ s.t \quad & X^T X = I_{n\times n} \\ & \sum_{i,j=1}^{n} \left(X^T l X \right)_{ij} \left(X^T \Lambda X \right)_{ij} = \epsilon \end{aligned} \tag{A.3}$$

Here the objective function seen in Eq. A.3 first constrains the transformation to rotations only, while the second constraint is intended to produce a transformation that conforms to a target free transfer term. Using the produced transformation matrix, a new set of system coefficients shown in

Chapter 3 Eq. 3.36 are used to produce a time integration of the transformed model. All three numerical processes are unfortunately effectively unconstrained. Following the recommendation of Balajewicz et al. [5] the dimension of the optimization problem scales by $2n^4$ with no additional structure on X. Likewise the root finding procedure is unbounded.

Appendix B

Maximum Likelihood Estimate

This appendix provides a derivation of the analytical maximum likelihood estimate of a stochastic matrix for an observed sequence of one, or many Markov chains of the same length. This derivation should prove what appears intuitively in Eq. 3.41. First the probability of observing a Markov chain is shown.

$$P\{X_T = i_T\} = P\{X_1 = i_1\} \prod_{t=2}^{n} P\{X_t = i_t \mid X_{t-1} = i_{t-1}\} \tag{B.1}$$

Next, replacing the conditional probability with the elements of the estimated transition matrix p_{ij}, the likelihood function L is:

$$L(p|X) = P\{X_1 = i_1\} \prod_{t=2}^{T} p_{ij}(X_t) \tag{B.2}$$

The initial probability will be easier to represent for multiple Markov chains if the quantity known as the stationary distribution π is introduced.

Definition [15] 1. A vector $\pi = (\pi_1, \pi_2, \ldots, \pi_k)^T$ is said to be a stationary distribution of a finite Markov chain if it satisfies:

$$\pi_i \geq 0 \text{ and } \sum_{i=1}^{k} \pi_i = 1 \tag{B.3a}$$

$$P\pi = \pi \tag{B.3b}$$

Here π represent the long run proportion of time X spends in state i. In this case $\pi(X) = P\{X_1 = i_1\}$ leading to:

$$L(p, \pi|X) = \pi(X) \prod_{t=2}^{T} p_{ij}(X) \tag{B.4}$$

Two useful quantities to define are the number of transitions from $i \to j$ and the number starts in state i shown below:

$$n_{ij} \coloneqq \sum_{t=2}^{T} I(X_t = i, X_{t-1} = j) \tag{B.5a}$$

$$n_i^1 \coloneqq I(X_1 = i) \tag{B.5b}$$

Where I is the indicator function. Equation B.4 can now be represented as:

$$L(p, \pi|X) = \prod_{i=1}^{n} \pi_i^{n_i^1} \prod_{i,j=1}^{n} p_{ij}^{n_{ij}} \tag{B.6}$$

This expression can now be generalized for set of M Markov chains $\mathcal{X} = \{X_1, \ldots, X_M\}$, by taking the product of all the likelihoods.

$$L(p, \pi, |\mathcal{X}) = \prod_{m=1}^{M} \left(\prod_{i=1}^{n} \pi_i^{I(X_{m,1}=i)} \prod_{i,j=1}^{n} p_{ij}^{I(X_m, t=i, X_{m,t-1}=j)} \right) \tag{B.7a}$$

$$L(p, \pi|\mathcal{X}) = \prod_{i=1}^{n} \pi_i^{n_i^1} \prod_{i,j=1}^{n} p_{ij}^{n_{ij}} \tag{B.7b}$$

$$n_{ij} \coloneqq \sum_{m=1}^{M} \sum_{t=2}^{T} I(X_t = i, X_{t-1} = j) \tag{B.7c}$$

$$n_i^1 \coloneqq \sum_{m=1}^{M} I(X_1 = i) \tag{B.7d}$$

With an equation for the likelihood of a set of Markov chains in place, the MLE of the transition matrix p_{ij} can now be determined. First, Eq. B.7b is transformed into the log domain to simplify the calculation.

$$\mathcal{L}(p, \pi|\mathcal{X}) = \sum_{i=1}^{n} n_i^1 ln(\pi_i) + \sum_{i,j=1}^{n} n_{ij} ln(p_{ij}) \tag{B.8}$$

The application of the MLE in this work has a known initial state exactly, as it will be determined by the initial conditions set for the ROM. In this

case the summation of the stationary distribution of Eq. B.8 will reduce to a constant and won't affect the location of the maximum reducing Eq. B.8 to:

$$\mathcal{L}(p|\mathcal{X}) = \sum_{i,j=1}^{n} n_{ij} ln(p_{ij}) \tag{B.9}$$

At this point Lagrange multipliers, an optimization technique for enforcing constraints, are introduce to enforce the constraints of Eq. B.10. For a quick introduction to Lagrange multipliers the reader can refer to almost any introductory optimization text or the cited text [59]. Referring back to Subsection 3.4.1 we can rewrite a property of Markov chains in terms of the stochastic matrix's as:

$$\sum_{j=1}^{n} p_{ij} = 1 \tag{B.10}$$

Instead of simply looking for the maximum of $\mathcal{L}(p)$, the incorporation n Lagrange multiplier, $\lambda_1, \lambda_2 \ldots \lambda_n$, constrains the objective function.

$$\mathcal{L}(p|X) - \sum_{i=1}^{j} \lambda_i \Bigg(\sum_{j=1}^{n} p_{ij} - 1 \Bigg) \tag{B.11}$$

A key property of Lagrange multipliers, is that when you take the derivative with respect to one of the multipliers you recover the constraint that it is enforcing. This simply increases the dimension of the optimization problem from n^2 to $n^2 + n$. Now taking the derivative of Eq. B.11 with respect to p_{ij}:

$$0 = \frac{n_{ij}}{p_{ij}} - \lambda_i \tag{B.12}$$

$$p_{ij} = \frac{n_{ij}}{\lambda_i} \tag{B.13}$$

Using Eq. B.10 and plugging back into Eq. B.13 we now have.

$$\sum_{j=1}^{n} \frac{n_{ij}}{\lambda_i} = 1 = \sum_{j=1}^{n} p_{ij} \tag{B.14}$$

$$\sum_{j=1}^{n} n_{ij} = \lambda_i \tag{B.15}$$

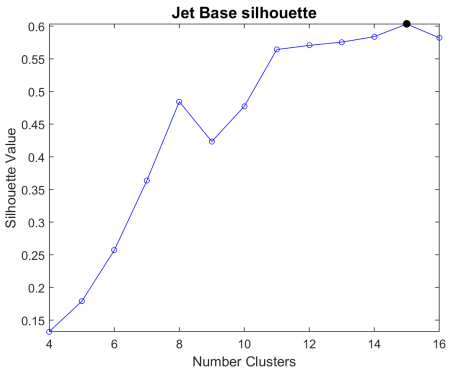
Plugging this results for λ_i back into Eq. B.13 and rearranging, gives the results presented in Chapter 3 Eq. 3.41.

$$\hat{p}_{ij} = \frac{n_{ij}}{\sum_{j=1}^{n} n_{ij}} \tag{B.16}$$

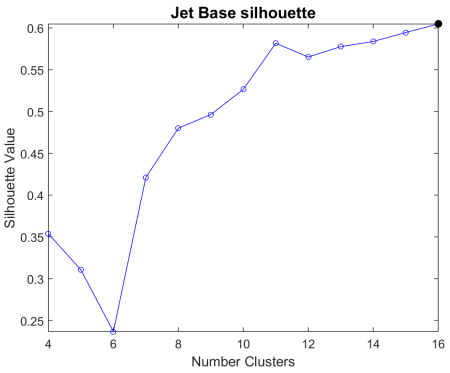
Appendix C

Cluster Evaluation

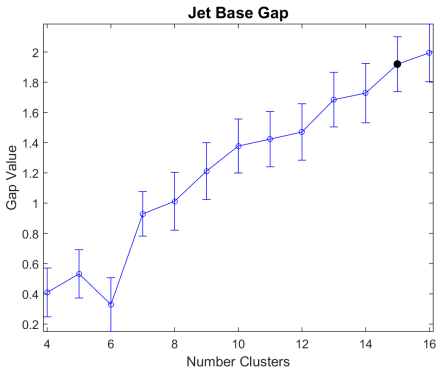
Here, the results of cluster number evaluation are presented. These are from a sample data set for each test conditions reported in Chapter 5. Here plots are presented in the order they appear in tables 6.1 and 6.2.



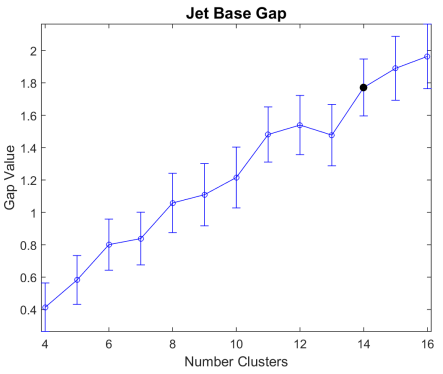
(a) silhouette index : k-means



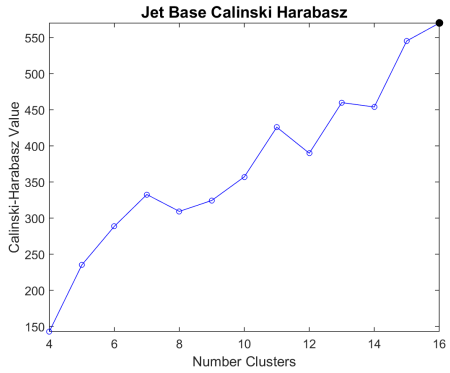
(b) silhouette index : gmm



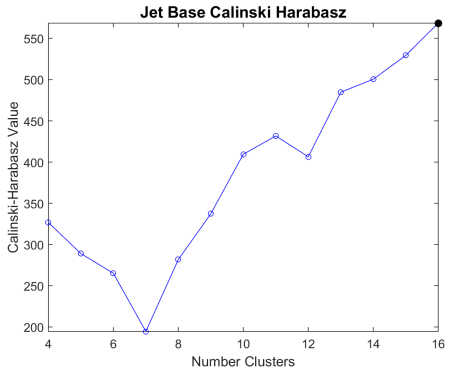
(c) gap statistic : k-means



(d) gap statistic : gmm

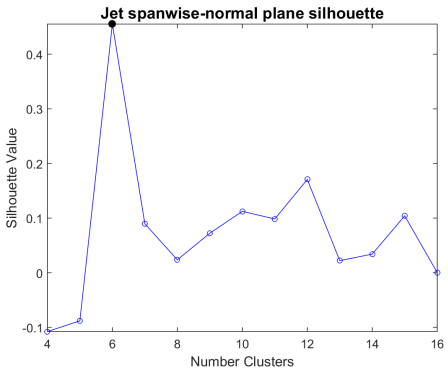


(e) calinski-harabasz index : k-means

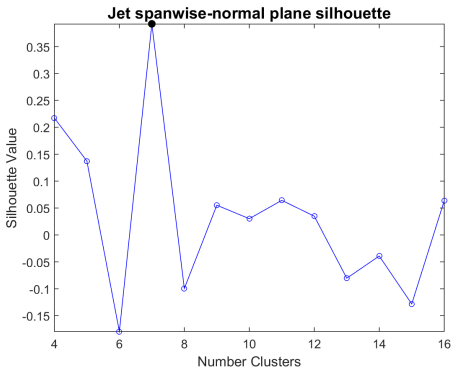


(f) calinski-harabasz index : gmm

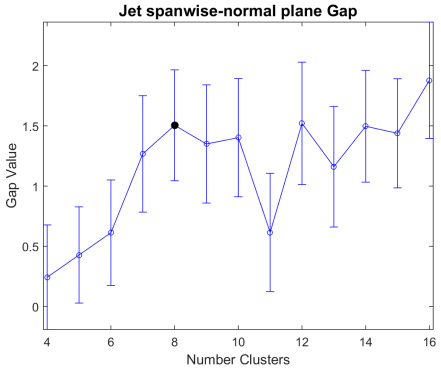
Figure C.1: Cluster evaluation for the jet in the streamwise-normal plane.



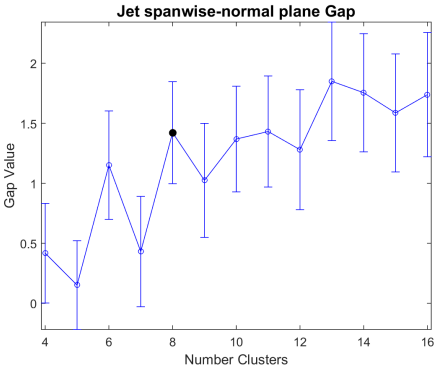
(a) silhouette index : k-means



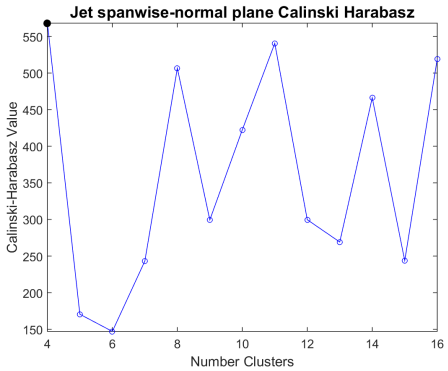
(b) silhouette index : gmm



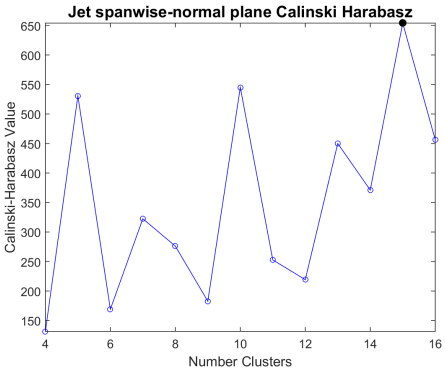
(c) gap statistic : k-means



(d) gap statistic : gmm

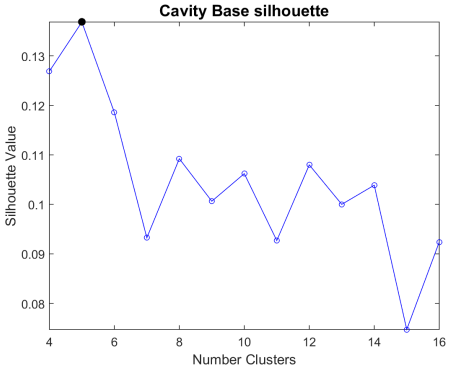


(e) calinski-harabasz index : k-means

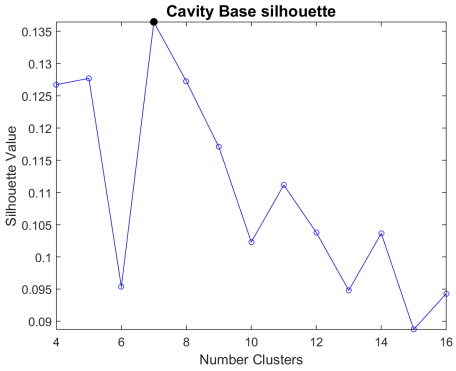


(f) calinski-harabasz index : gmm

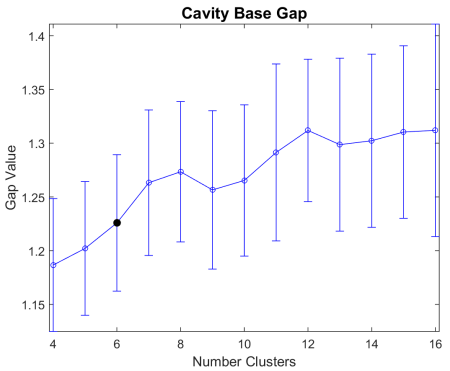
Figure C.2: Cluster evaluation for the jet in the spanwise-normal plane



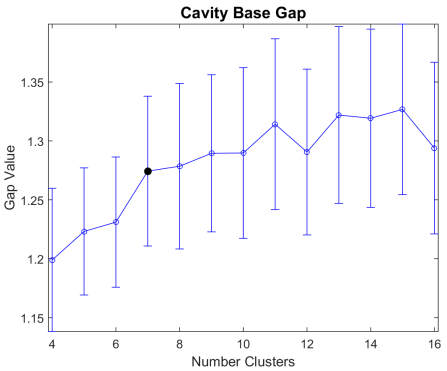
(a) silhouette index : k-means



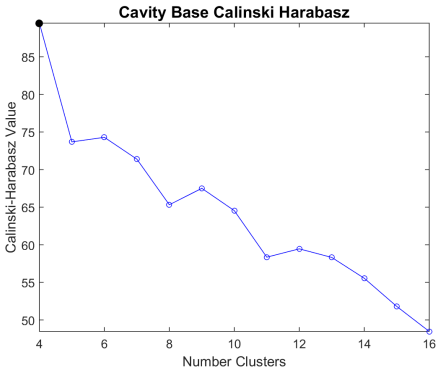
(b) silhouette index : gmm



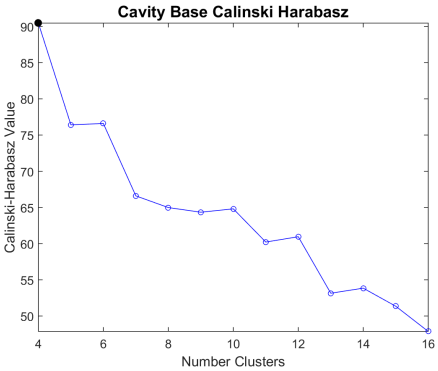
(c) gap statistic : k-means



(d) gap statistic : gmm

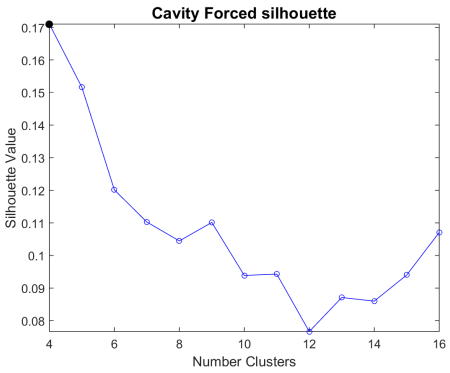


(e) calinski-harabasz index : k-means

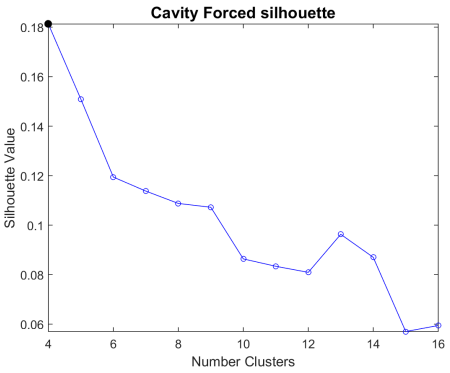


(f) calinski-harabasz index : gmm

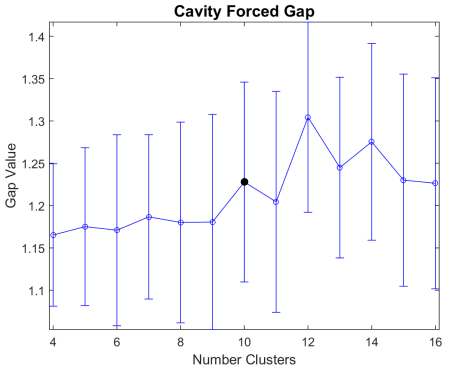
Figure C.3: Cluster evaluation for a baseline cavity flow.



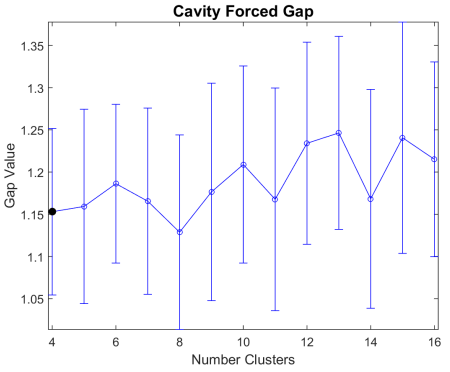
(a) silhouette index : k-means



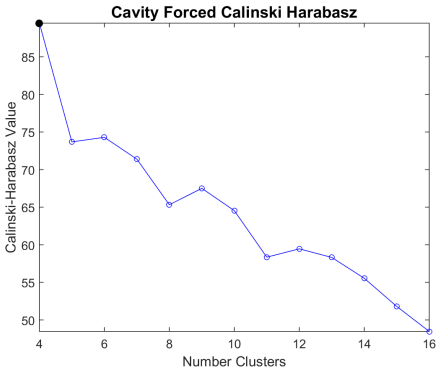
(b) silhouette index : gmm



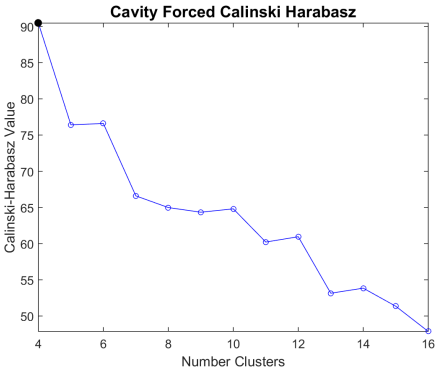
(c) gap statistic : k-means



(d) gap statistic : gmm

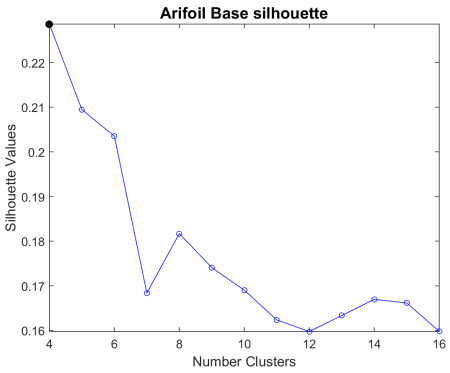


(e) calinski-harabasz index : k-means

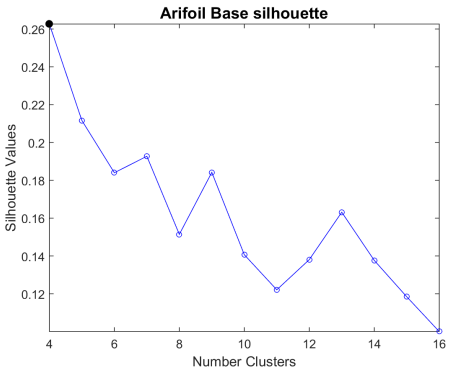


(f) calinski-harabasz index : gmm

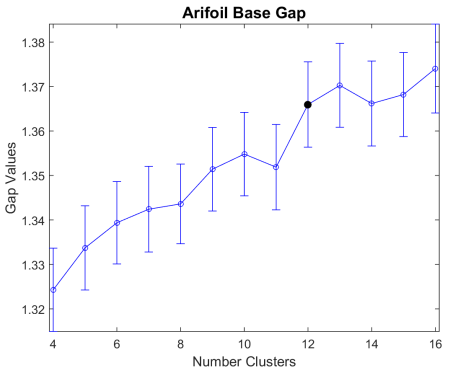
Figure C.4: Cluster evaluation for a force cavity flow.



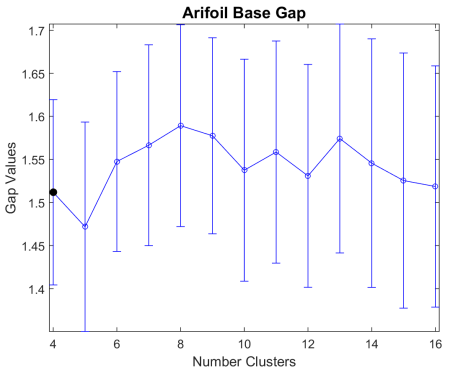
(a) silhouette index : k-means



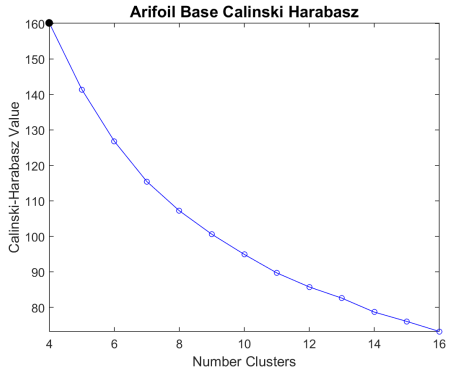
(b) silhouette index : gmm



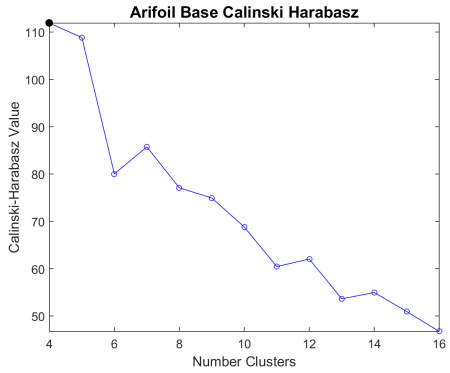
(c) gap statistic : k-means



(d) gap statistic : gmm

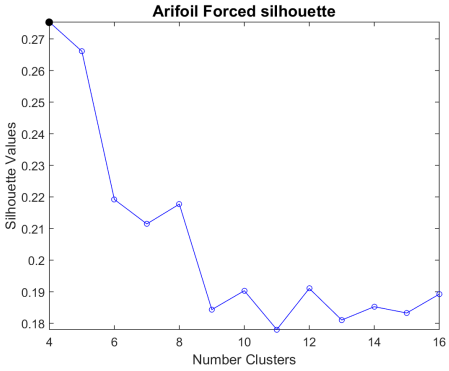


(e) calinski-harabasz index : k-means

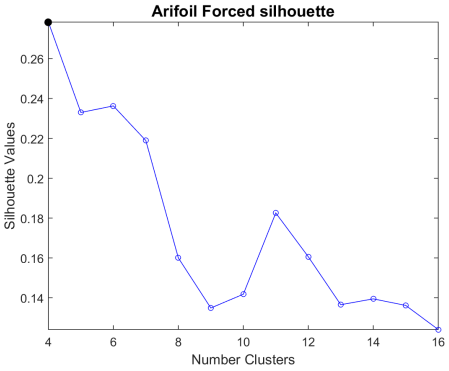


(f) calinski-harabasz index : gmm

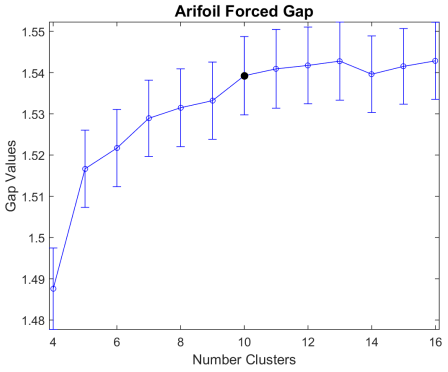
Figure C.5: Cluster evaluation for a baseline airfoil flow.



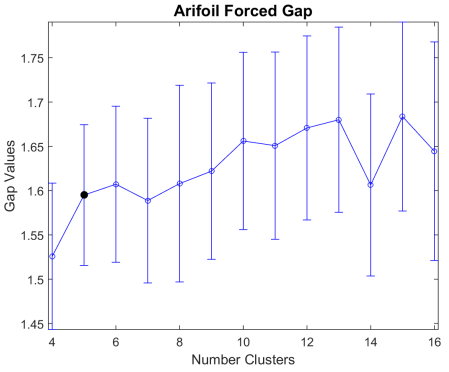
(a) silhouette index : k-means



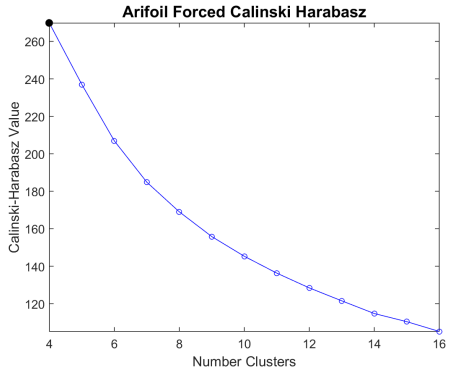
(b) silhouette index : gmm



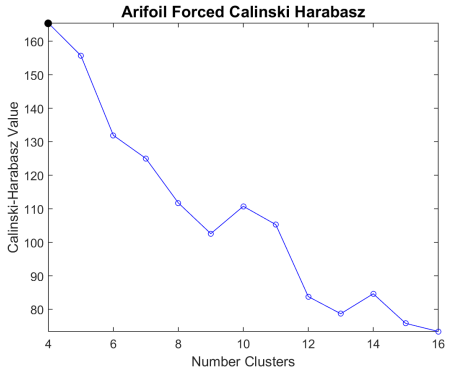
(c) gap statistic : k-means



(d) gap statistic : gmm

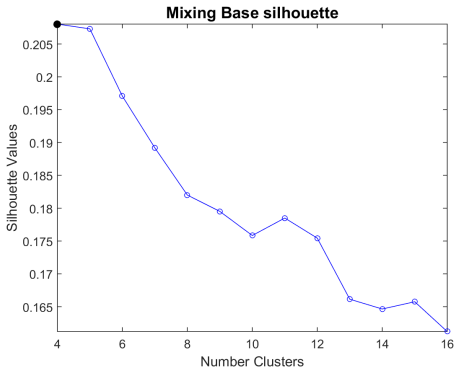


(e) calinski-harabasz index : k-means

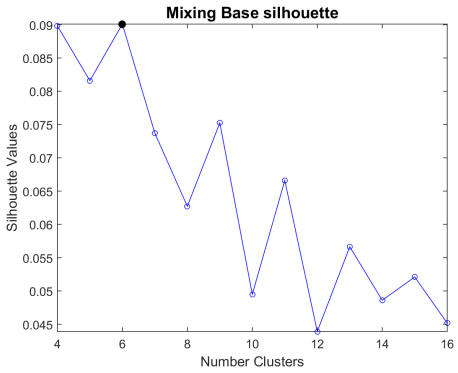


(f) calinski-harabasz index : gmm

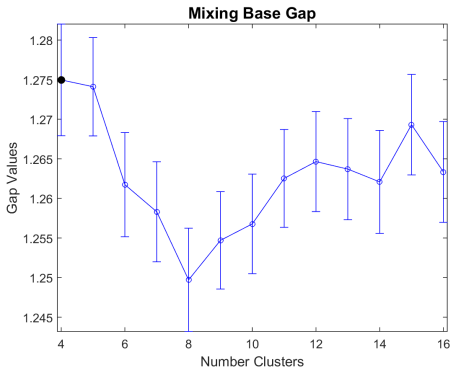
Figure C.6: Cluster evaluation for a forced airfoil flow.



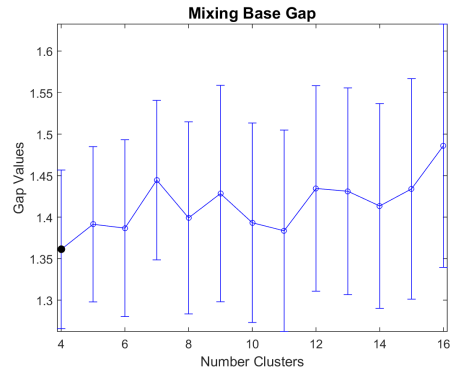
(a) silhouette index : k-means



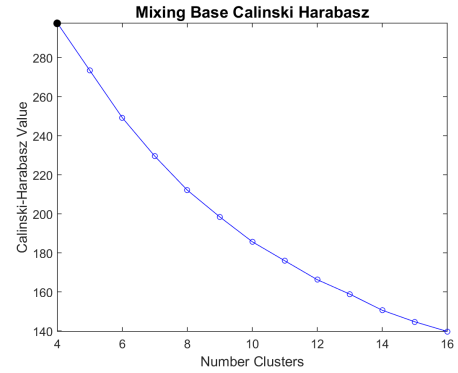
(b) silhouette index : gmm



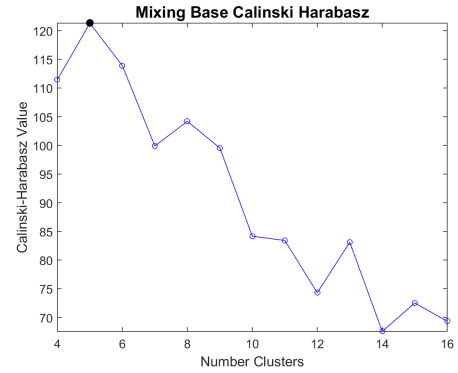
(c) gap statistic : k-means



(d) gap statistic : gmm

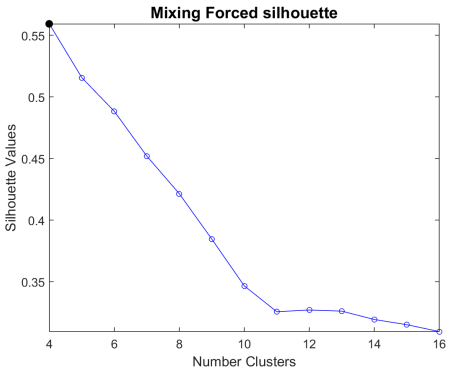


(e) calinski-harabasz index : k-means

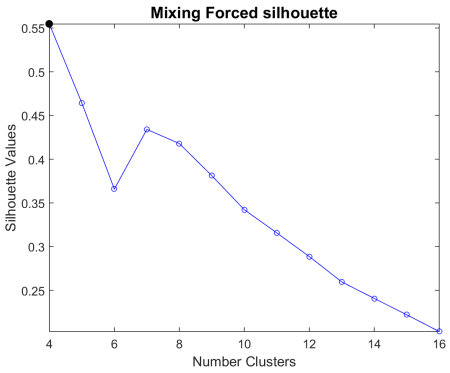


(f) calinski-harabasz index : gmm

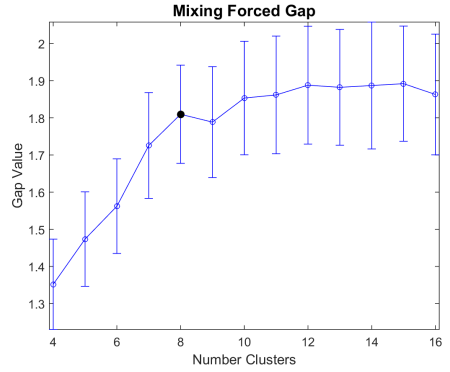
Figure C.7: Cluster evaluation for the baseline mixing layer flow.



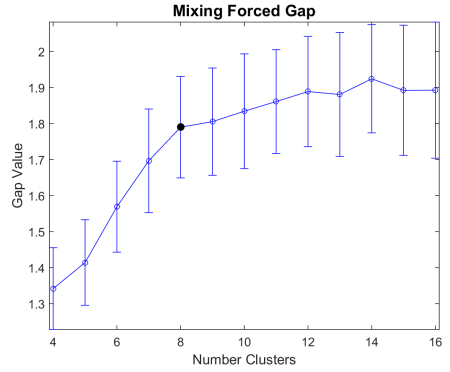
(a) silhouette index : k-means



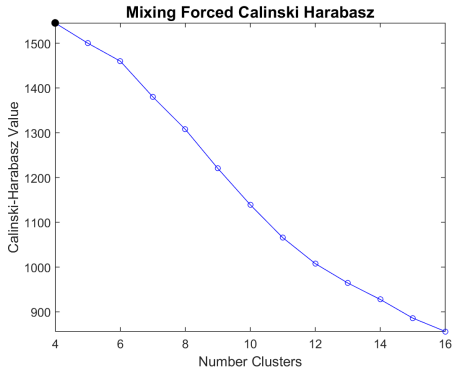
(b) silhouette index : gmm



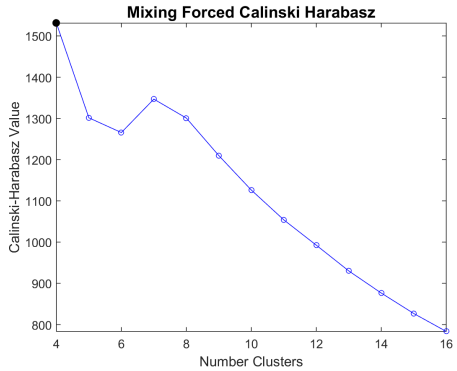
(c) gap statistic : k-means



(d) gap statistic : gmm



(e) calinski-harabasz index : k-means



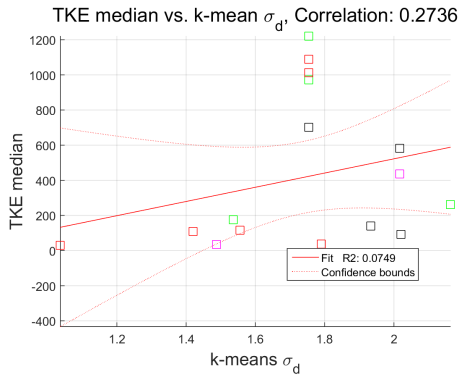
(f) calinski-harabasz index : gmm

Figure C.8: Cluster evaluation for the forced mixing layer flow.

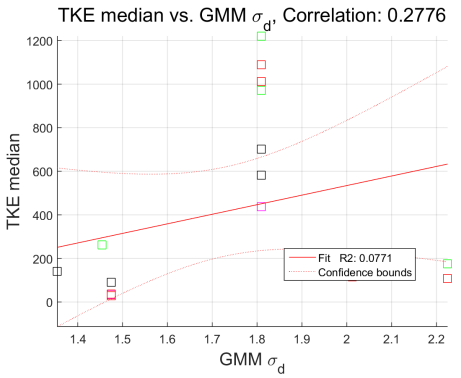
Appendix D

SMM Correlations

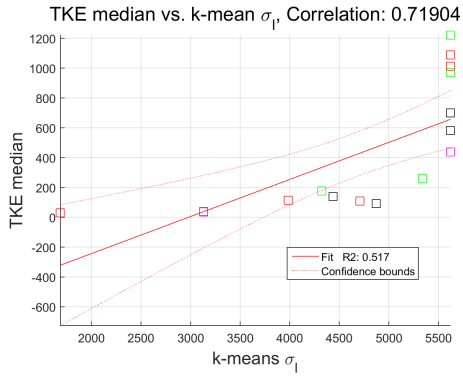
Plots that were not included in Chapter 6 Section 6.2 are presented. First the correlations between the 4 scores for the 18° baseline airfoil flow. Here the discrepancy between the empirical median and std deviation are shown. Next, a baseline cavity flow is presented where the first POD basis function's modal amplitude is the characteristic of choice. Following the last set of correlations observed, standard deviation of phase shift is shown. The system would show no oscillation and have a deviation close to zero.



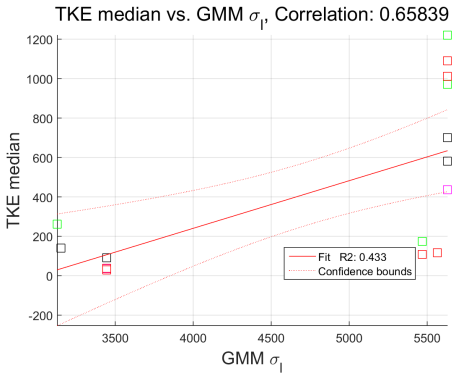
(a) Baseline 18° Airfoil flow : median TKE : k-means σ_d.



(b) Baseline 18° Airfoil flow : median TKE : GMM σ_d.

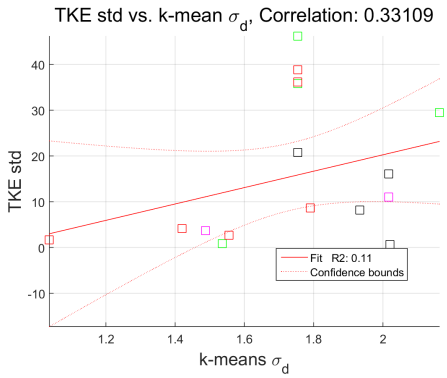


(c) Baseline 18° Airfoil flow : median TKE : k-means σ_l.

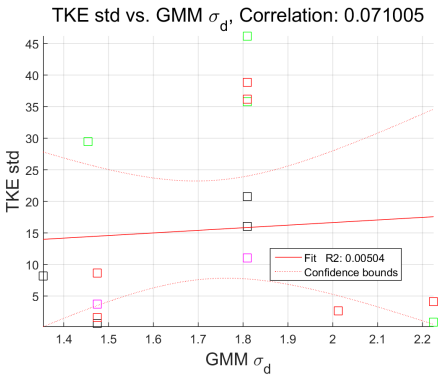


(d) Baseline 18° Airfoil flow : medain TKE : GMM σ_l.

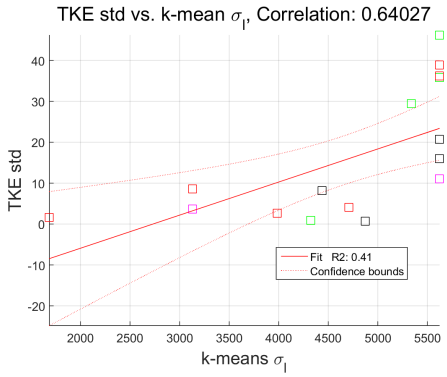
Figure D.1: Scatter plots of the four scoring methods for a baseline Airfoil flow at 18° for median TKE.



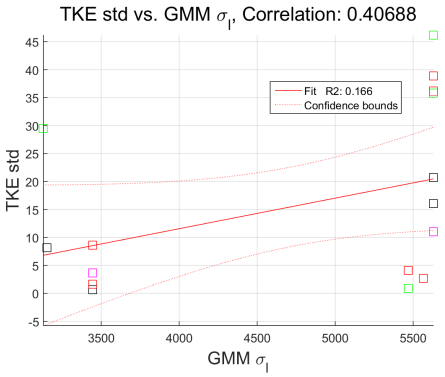
(a) Baseline 18° Airfoil flow : TKE standard deviation : k-means σ_d.



(b) Baseline 18° Airfoil flow : TKE standard deviation : GMM σ_d.

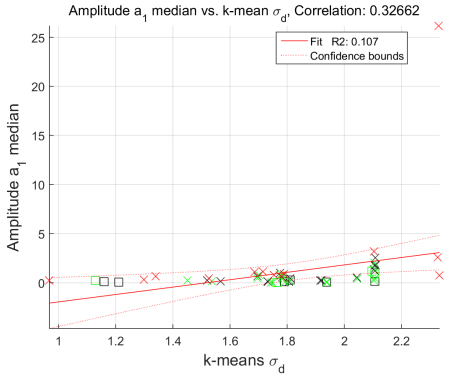


(c) Baseline 18° Airfoil flow : TKE standard deviation : k-means σ_l.

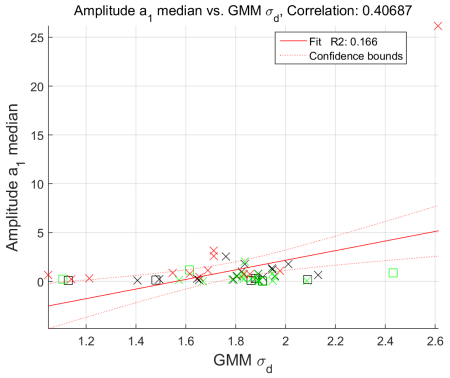


(d) Baseline 18° Airfoil flow : TKE standard deviation : GMM σ_l.

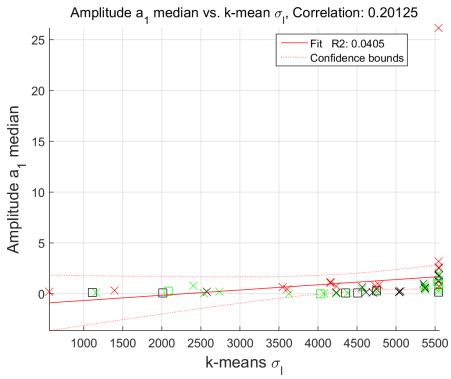
Figure D.2: Scatter plots of the four scoring methods for a baseline Airfoil flow at 18° for standard deviation TKE.



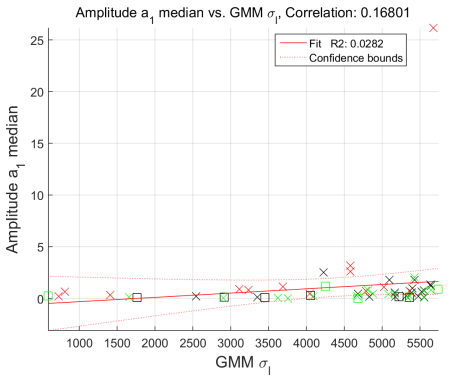
(a) Baseline Cavity flow : median modal amplitude a_1 : k-means σ_d.



(b) Baseline Cavity flow : median modal amplitude a_1 : GMM σ_d.

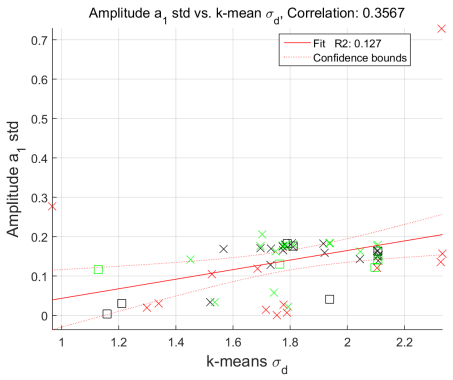


(c) Baseline Cavity flow : median modal amplitude a_1 : k-means σ_l.

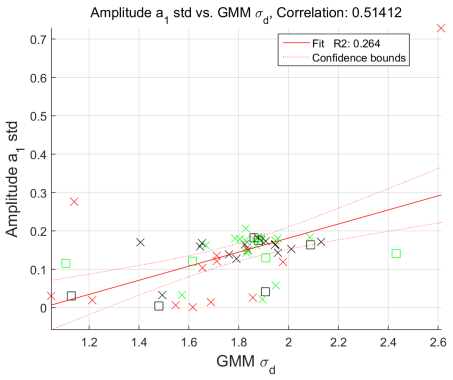


(d) Baseline Cavity flow : median modal amplitude a_1 : GMM σ_l.

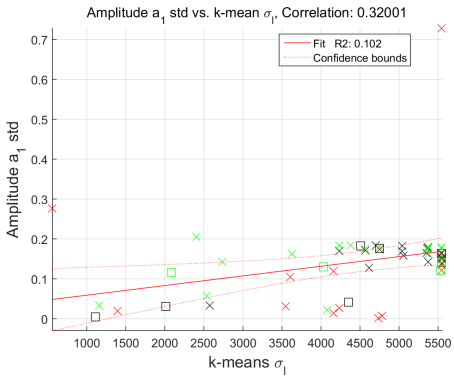
Figure D.3: Scatter plots of the four scoring methods for a baseline airfoil flow at 18° for modal amplitude a_1.



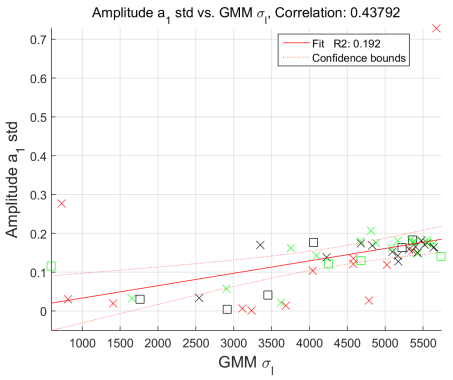
(a) Baseline Cavity flow : modal amplitude a_1 standard deviation : k-means σ_d.



(b) Baseline Cavity flow : modal amplitude a_1 standard deviation : GMM σ_d.

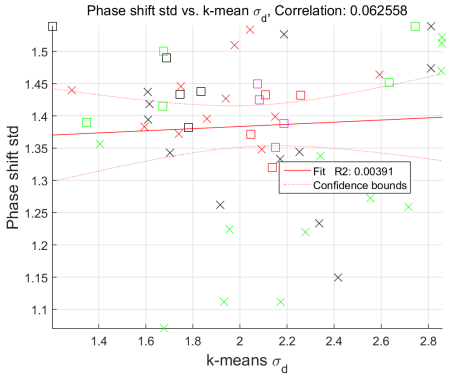


(c) Baseline Cavity flow : modal amplitude a_1 standard deviation : k-means σ_l.

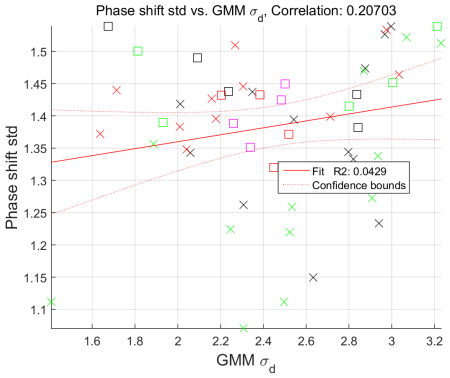


(d) Baseline Cavity flow : modal amplitude a_1 standard deviation : GMM σ_l.

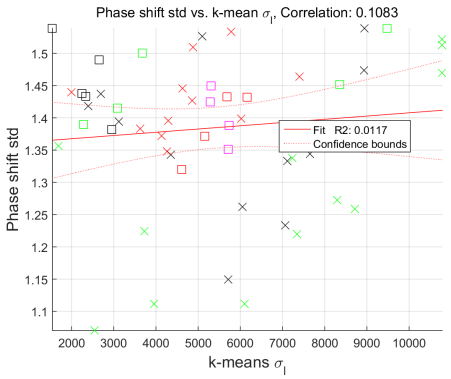
Figure D.4: Scatter plots of the four scoring methods for a baseline cavity flow at 18° for modal amplitude a_1 standard deviation.



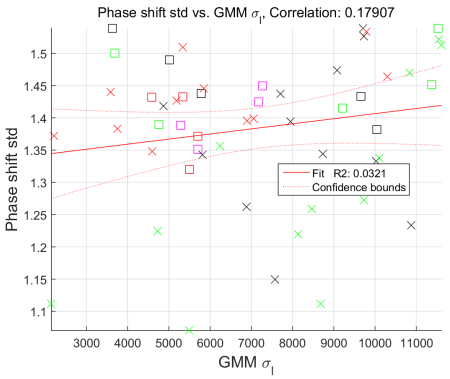
(a) Forced Mixing Layer : phase descrepancy standard deviation : k-means σ_d.



(b) Forced Mixing Layer : phase descrepancy standard deviation : GMM σ_d.



(c) Forced Mixing Layer : phase descrepancy standard deviation : k-means σ_l.



(d) Forced Mixing Layer : phase descrepancy standard deviation : GMM σ_l.

Figure D.5: Scatter plots of the four scoring methods for the forced mixing layer of the phase discrepancy standard deviation.

Bibliography

[1] *Applied Multivariate Analysis.*

[2] Nadine Aubry, Régis Guyonnet, and Ricardo Lima. Spatiotemporal analysis of complex signals: theory and applications. *Journal of Statistical Physics*, 64(3-4):683–739, 1991.

[3] Nadine Aubry, Philip Holmes, John L Lumley, and Emily Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics*, 192:115–173, 1988.

[4] Julie M Ausseur, Jeremy T Pinier, Mark N Glauser, Hiroshi Higuchi, and Henry Carlson. Experimental development of a reduced-order model for flow separation control. *AIAA paper*, 1251:2006, 2006.

[5] Maciej J Balajewicz, Earl H Dowell, and Bernd R Noack. Low-dimensional modelling of high-reynolds-number shear flows incorporating constraints from the navier–stokes equation. *Journal of Fluid Mechanics*, 729:285–308, 2013.

[6] Michel Bergmann and Laurent Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *Journal of Computational Physics*, 227(16):7813–7840, 2008.

[7] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[8] John Burkardt, Max Gunzburger, and Hyung-Chun Lee. Centroidal voronoi tessellation-based reduced-order modeling of complex systems. *SIAM Journal on Scientific Computing*, 28(2):459–484, 2006.

[9] John Burkardt, Max Gunzburger, and Hyung-Chun Lee. Pod and cvt-based reduced-order modeling of navier–stokes flows. *Computer Methods in Applied Mechanics and Engineering*, 196(1):337–355, 2006.

[10] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[11] Edgar Caraballo, Cosku Kasnakoglu, Andrea Serrani, and Mo Samimy. Control input separation methods for reduced-order model-based feedback flow control. *AIAA journal*, 46(9):2306–2322, 2008.

[12] Edgar J Caraballo. *Reduced Order Model Development for Feedback Control of Cavity Flows*. PhD thesis, The Ohio State University, 2008.

[13] Edgar Javier Caraballo. *An application of the proper orthogonal decomposition to an axisymmetric supersonic jet*. PhD thesis, The Ohio State University, 2001.

[14] John Chabot, Edgar Caraballo, and Jesse Little. Reduced order modeling of a dielectric barrier discharge controlled shear layer using minimum basis rotations. In *Proceeings of the 45TH AIAA Fluid Dynamic Conference*, June 2015.

[15] Wai-Ki Ching and Micheal Ng. *Markov Chains: Models, Algorithms and Applications*. Springer, 2006.

[16] Laurent Cordier, El Majd, B Abou, and J Favier. Calibration of pod reduced-order models using tikhonov regularization. *International Journal for Numerical Methods in Fluids*, 63(2):269–296, 2010.

[17] Laurent Cordier, Bernd R Noack, Gilles Tissot, Guillaume Lehnasch, Joel Delville, Maciej Balajewicz, Guillaume Daviller, and Robert K Niven. Identification strategies for model-based control. *Experiments in fluids*, 54(8):1–21, 2013.

[18] M Couplet, P Sagaut, and C Basdevant. Intermodal energy transfers in a proper orthogonal decomposition–galerkin representation of a turbulent separated flow. *Journal of Fluid Mechanics*, 491:275–284, 2003.

[19] Marco Debiasi and Mo Samimy. Logic-based active control of subsonic cavity flow resonance. *AIAA journal*, 42(9):1901–1909, 2004.

[20] James R DeBonis. *The numerical analysis of a turbulent compressible jet*. PhD thesis, Ohio State University, 2001.

[21] Richard Ely and Jesse Little. The mixing layer perturbed by dielectric barrier discharge. In *In proceedings of the 43rd AIAA Flow Control Conference, San Diego, CA, USA*. AIAA, 2013.

[22] Clive AJ Fletcher. *Computational galerkin methods*. Springer, 1984.

[23] Ari Glezer, Zafer Kadioglu, and Arne J Pearlstein. Development of an extended proper orthogonal decomposition and its application to a time periodically forced plane mixing layer. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 1(8):1363–1373, 1989.

[24] Andreas Gross and Hermann F Fasel. Control-oriented proper orthogonal decomposition models for unsteady flows. *AIAA journal*, 45(4):814–827, 2007.

[25] Hasan Gunes and Ulrich Rist. Proper orthogonal decomposition reconstruction of a transitional boundary layer with and without control. *Physics of Fluids (1994-present)*, 16(8):2763–2784, 2004.

[26] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[27] Leonhard Held and Daniel Sabanés Bové. *Applied Statistical Inference: Likelihood and Bayes*. Springer Science & Business Media, 2013.

[28] Philip Holmes. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

[29] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[30] Gareth James, Daniela Witten, and Trevor Hastie. *An Introduction to Statistical Learning: With Applications in R*. Taylor & Francis, 2014.

[31] Eurika Kaiser, Bernd R Noack, Laurent Cordier, Andreas Spohn, Marc Segond, Markus Abel, Guillaume Daviller, Jan Östh, Siniša Krajnović, and Robert K Niven. Cluster-based reduced-order modelling of a mixing layer. *Journal of Fluid Mechanics*, 754:365–414, 2014.

[32] Robert H Kraichnan and Shiyi Chen. Is there a statistical mechanics of turbulence? *Physica D: Nonlinear Phenomena*, 37(1):160–172, 1989.

[33] H Kriegel, Peer Kroger, Eugen Schubert, and Arthur Zimek. Outlier detection in arbitrarily oriented subspaces. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 379–388. IEEE, 2012.

[34] Olga A Ladyzhenskaya and Richard A Silverman. *The mathematical theory of viscous incompressible flow*, volume 76. Gordon and Breach New York, 1969.

[35] Kenneth Lange. *Optimization*. Springer, 2013.

[36] Jesse Little, Keisuke Takashima, Munetake Nishihara, Igor Adamovich, and Mo Samimy. Separation control with nanosecond-pulse-driven dielectric barrier discharge plasma actuators. *AIAA journal*, 50(2):350–365, 2012.

[37] John Leask Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, pages 166–178, 1967.

[38] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[39] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[40] Marina Meilă. The uniqueness of a good optimum for k-means. In *Proceedings of the 23rd international conference on Machine learning*, pages 625–632. ACM, 2006.

[41] Bernd R Noack, Marek Morzynski, and Gilead Tadmor. *Reduced-order modelling for flow control*, volume 528. Springer Science & Business Media, 2011.

[42] Bernd R Noack, Paul Papas, and Peter A Monkewitz. The need for a pressure-term representation in empirical galerkin models of incompressible shear flows. *Journal of Fluid Mechanics*, 523:339–365, 2005.

[43] Bernd R Noack, Michael Schlegel, Boye Ahlborn, Gerd Mutschke, Marek Morzyński, and Pierre Comte. A finite-time thermodynamics of unsteady fluid flows. *Journal of Non-Equilibrium Thermodynamics*, 33(2):103–148, 2008.

[44] Jan Östh, Bernd R Noack, Siniša Krajnović, Diogo Barros, and Jacques Borée. On the need for a nonlinear subscale turbulence term in pod models as exemplified for a high-reynolds-number flow over an ahmed body. *Journal of Fluid Mechanics*, 747:518–544, 2014.

[45] Mojtaba Rajaee, Sture KF Karlsson, and Lawrence Sirovich. Low-dimensional description of free-shear-flow coherent structures and their dynamical behaviour. *Journal of Fluid Mechanics*, 258:1–29, 1994.

[46] Dietmar Rempfer and Hermann F Fasel. Evolution of three-dimensional coherent structures in a flat-plate boundary layer. *Journal of Fluid Mechanics*, 260:351–375, 1994.

[47] Eréndira Rendón, Itzel Abundez, Alejandra Arizmendi, and ElviaM Quiroz. Internal versus external cluster validation indexes. *International Journal of computers and communications*, 5(1):27–34, 2011.

[48] Christopher C Rethmel. *Airfoil leading edge flow separation control using nanosecond pulse DBD plasma actuators*. PhD thesis, The Ohio State University, 2011.

[49] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[50] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.

[51] CW Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.

[52] David Ruelle, Floris Takens, et al. On the nature of turbulence. *Commun. math. phys*, 20(3):167–192, 1971.

[53] M Samimy, M Debiasi, E Caraballo, A Serrani, X Yuan, J Little, and JH Myatt. Feedback control of subsonic cavity flows using reduced-order models. *Journal of Fluid Mechanics*, 579:315–346, 2007.

[54] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[55] Bruno Sericola. *Markov Chains: Theory, Algorithms and Applications*. Wiley, 2013.

[56] WL Siauw, JP Bonnet, J Tensi, and LN Cattafesta III. Physics of separated flow over a naca 0015 airfoil and detection of flow separation.

In *47Th AIAA Aerospace Sciences Meeting Including The New Horizons Forum And Aerospace Exposition,(47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition)*, 2009.

[57] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. part i: Coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[58] Torstens Skujins and Carlos ES Cesnik. Reduced-order modeling of hypersonic vehicle unsteady aerodynamics. In *Proceedings of the 2010 AIAA Guidance, Navigation, and Control Conference*, 2010.

[59] Jan Snyman. *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, volume 97. Springer Science & Business Media, 2005.

[60] Katepalli R Sreenivasan. Fluid turbulence. *Reviews of Modern Physics*, 71(2):S383, 1999.

[61] W Stankiewicz, M Morzyński, BR Noack, and G Tadmor. Reduced order galerkin models of flow around naca-0012 airfoil. *Mathematical Modelling and Analysis*, 13(1):113–122, 2008.

[62] Taylor D Sullivan. Reduced order modeling of flow over a naca 0015 airfoil for future control application. Master's thesis, Miami University, 2014.

[63] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[64] Yang Zhang, Nicholas AS Hamm, Nirvana Meratnia, Alfred Stein, M van de Voort, and Paul JM Havinga. Statistics-based outlier detection for wireless sensor networks. *International Journal of Geographical Information Science*, 26(8):1373–1392, 2012.

[65] Arthur Zimek, Matthew Gaudet, Ricardo JGB Campello, and Jörg Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 428–436. ACM, 2013.