School of Mechanical and Manufacturing Engineering

Faculty of Engineering

UNSW Sydney

Computational Methods for Unsteady Flow Structure Analysis

By

Jake Allan Edwards

Thesis submitted as a requirement for the degree of Bachelor of Engineering in Mechanical Engineering

Submitted: 10 August 2021 Student zID: z5165158
Supervisor: Sam Mallinson & Tracie Barber

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed _____

Date _____

Abstract

This report outlines the utility of computational methods such as autoregression (AR(p)), vector autoregression (VAR), Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD) in the context of the cylinder wake problem and for ink-jet print zone simulation. It was found that the Dynamic Mode Decompsition Method was severely limited by its inherent usage of one lag to reconstruct dynamics whereas methods such as VAR which use multiple lags can successfully reconstruct more complex dynamics associated with the high order modes. A generalized framework for identifying the bifurcation point is also proposed with respect to usage of these different algorithms.

Acknowledgments

I would like to especially thank Geordie D. McBain for all his thorough explanations with my never-ending questions and assistance with Python scripting and directing my research. You have been an immense help. Dr Sam Mallinson has been a huge help in helping me visualize fundamental flow phenomenon at a very detailed level and helped provide scope for the broader applications of the research. I would also like to thank Professor Tracie Barber & André Vilanova Aquino for their assistance in providing research tips and helping me structure my approach to research as well as developing the necessary geometry to analyse the tiger stripe error with modal analysis. I would also like to acknowledge the open-source community particularly the scientific computing section that has been so monumentally helpful to all of the work in this report.

Contents

Abstract iii
Acknowledgments iv
Table of Figures vi
Table of Tables viii
Table of Code Snippets viii
Nomenclature ix
Chapter 1 - Introduction 1
Chapter 2- Literature Review 4
2.1. POD Analysis of Fluid Interactions 4
2.2. POD Analysis of 3-D Complex Flow in Practical Applications 6
2.3. DMD Analysis of a Fluid Interactions 8
2.4. POD and DMD Analysis Comparison Studies 9
2.5. Machine Learning 12
2.6. Ink Jet Print Zone and Tiger Stripes 13
2.7. Supercritical Hopf Bifurcation on a Cylinder 14
Chapter 3 - Methodology 15
3.1. 2-D Cylinder Flow 15
3.1.1. Cylinder Wake Geometry Simulation 16
3.1.2. Selection of POD Algorithm 21
3.1.3. POD Validation Cylinder Wake 23
3.1.4. Selection of DMD Algorithm 25
3.1.5. DMD Validation Cylinder Wake 26
3.1.6. Autoregression of Cylinder Wake 28
3.1.7. Fast Fourier Transform of Cylinder Wake 30
3.1.8. High Order DMD, Prony Analysis and Vector Autoregression of Cylinder Wake 31
3.2. Ink-jet Printer Injection Zone 32
3.2.1. Ink-jet Printer Injection Zone Simulation 32
Chapter 4 – Results and Discussion 34
4.1. Confined Cylinder Wake Problem 34
4.2. Ink-jet Printer Injection Zone Preliminary Results 43
Chapter 5- Future Works and Current Work Limitations 46
Chapter 6 – Conclusions 48
References 49
Glossary 53
Appendices 56

Appendix A. Code Repository 56
Appendix B. Assortment of Flow Structure Figures 56
Appendix C. Assortment of Code Snippets 59

Table of Figures

Figure 1. Modal decomposition of 2-D flat-plate wing Re=100 α=30° [1] 1
Figure 2. Simplified diagrammatic overview of DMD on fluid flow [13] 4
Figure 3. Smoke-wire visualisation of the cylinder wake topology (a) bare cylinder; (b) porous cylinder [17] 5
Figure 4. Modal shapes of vortex rope (a) static pressure and (b) radial velocity mode 1 at 2% jet discharge value [24] 6
Figure 5. Modal shapes of vortex rope (a) static pressure and (b) radial velocity mode 1 at 5% jet discharge value [24] 7
Figure 6. Control valve extended-POD modes for pressure-velocity coupling field, XY plane [26] 7
Figure 7. DMD spatial patterns of mode 1 (a) square cylinder (b) wavy-square cylinder node plane (c) wavy-square cylinder middle plane (d) wavy-square cylinder saddle plane. [29] 8
Figure 8. Imaginary and real parts of spatial DMD mode (St = 0.173) for a diffuser [30] 9
Figure 9. Drag Coefficient as a Function of Time for a cylinder at Re=50 [32] 9
Figure 10. Shifted POD modes due to modified fuel injection spray angle [34] 10
Figure 11. Reduced sound pressure due to modified fuel injection spray angle [34] 10
Figure 12. POD and DMD modes for crevalle jack caudal fin FSI, Re=500 11
Figure 13. Leading travelling and stationary modes canonical two-dimensional cylinder at Re = 60 [38] 12
Figure 14. Reonstructed flow fields comparing POD and DMD for the tip leakage vortex in a multi-stage pump [39] 13
Figure 15. Solid block printing (left) and bar-code (right) showing the effect of airflow oscillations creating tiger stripes [43] 13
Figure 16. Bifurcation diagram - amplitude of oscillation squared (r^2) [mm^2/ms^2] vs. print density [46] .. 14
Figure 17. 2-D Cylinder flow geometry of a triangular mesh generated with Scikit FEM 16
Figure 18. Velocity probe at coordinate (1.759,0.114) of st08-NS simulation without restart facility for Re = 90 where time is in unit time, where dT =0.001 and 1 unit time is 1000 snapshots 18
Figure 19. Vorticity probe for fore and aft stagnation points with restart facility for Re = 90 19
Figure 20. Fore and aft stagnation point vorticity probes for Re = 74 19
Figure 21. Plot of snapshot 500 for Re = 150 using Matplotlib interval II, supercritical case generated using plot_snapshot.py 20
Figure 22. Plot of snapshot 11000 for Re = 150 using Matplotlib interval IV, supercritical Case generated using plot_snapshot.py 21
Figure 23. Plot of snapshot 18000 for Re = 70 using Matplotlib interval IV, subcritical Case generated using plot_snapshot.py 21
Figure 24. Spatial mode 1 from Re = 150, Interval IV, simulation generated by *st08_navier_stokes_cylinder.py* and spatial mode generated by POD_MODRED_Kutz.py ..... **Error! Bookmark not defined.**
Figure 25 Spatial mode 2 from Re = 150, Interval IV, st08-NS and spatial mode generated by POD_MODRED_Kutz.py 24
Figure 26. VORTALL Re=100 spatial mode 1, Interval IV, st08-NS and spatial mode generated by POD_Modred_Test.py 24
Figure 27. VORTALL Re=100 spatial mode 2, Interval IV, and spatial mode generated by POD_Modred_Test.py 24

Figure 28. Modal Energy Distribution Re = 150 25
Figure 29.DMD imaginary mode st08-NS Re = 150, Interval IV, st08-NS and spatial mode generated by DMD_Modred_Kutz.py 27
Figure 30.DMD real mode st08-NS Re = 150, Interval IV, st08-NS and spatial mode generated by DMD_Modred_Kutz.py 27
Figure 31. DMD imaginary mode VORTALL case, interval IV, and spatial mode generated by DMD_Test_Modred.py 27
Figure 32.DMD real mode VORTALL case , interval IV, and spatial mode generated by DMD_Test_Modred.py 28
Figure 33. DMD eigen values of first 21 modes from st08-NS simulation and VORTALL dataset generated by DMD_Modred_Kutz.py 28
Figure 34. First chronos mode reconstruction using AR(2) with a constant trend model with coefficient matrix and Vander modes for Re = 150 30
Figure 35. FFT of first chronos mode at Re = 150, considering Interval IV, snapshot window 10000-15000 31
Figure 36. Impinging jet printer zone vorticity contours in Paraview 33
Figure 37. Fore and Aft Stagnation Point Vorticity Probes for st08-NS simulation for subcritical cases, where Re = 70 (Left) and Re =75 (Right) 34
Figure 38 Fore and Aft Stagnation Point Vorticity Probes for st08-NS simulation for bifurcation transition between subcritical and supercritical cases, where Re = 76 (Left) and Re =77 (Right) 35
Figure 39. Fore and Aft Stagnation Point Vorticity Probes for st08-NS simulation for supercritical cases, where Re = 78 (Left) and Re =79 (Right) 35
Figure 40. Frequency Analysis of st08-NS fundamental frequency in Interval I, snapshot 3000-15000, using varying modal techniques 36
Figure 41.Growth Rate Analysis of st08-NS fundamental frequency in Interval I, snapshot 3000-15000, using varying modal techniques 36
Figure 42. Amplitude for first chronos mode in Interval IV, st08-NS supercritical cases 37
Figure 43. Squared singular values for Reynolds numbers 70-80 showcasing the bifurcation transition and bifurcation point Snapshot window 3000-30000 38
Figure 44. Squared singular values for Reynolds numbers 70-80 showcasing the bifurcation transition and bifurcation point snapshot window 3000-60000 38
Figure 45. Frequency Analysis of st08-NS fundamental frequency in Interval IV, snapshot 10000-15000, using varying modal techniques 39
Figure 46. Frequency Analysis of st08-NS fundamental frequency in Interval IV, snapshot 10000-15000, using varying modal techniques 39
Figure 47. Aft and fore stagnation point probes for Re = 150 40
Figure 48. FFT of aft and fore probes for Re = 150, st08-NS 40
Figure 49. Chronos mode plot for first 8 Modes and AR (2) reconstruction for Re = 150, st08-NS, Interval IV, down-sampling = 16 produce using AR_2 Reconstruction.py 41
Figure 50. POD modal energy distribution for Re = 150, from singular values 41
Figure 51. Chronos Mode Plot for first 8 Modes and VAR (2) Reonstruction for Re = 150, st08-NS, Interval IV, down-sampling = 16 producing using VAR_vorticity_prony_analysis.py 42
Figure 52. FFT Analysis of first 8 modes for Re = 150, st08-NS, Interval IV 42
Figure 53. Periodic 2-D flow over a cylinder, Prony roots μ (r = 8, p= 2, down-sampling = 16), horizontal real axis and imaginary vertical axis with unit circle produced using VAR_vorticity_prony_analysis.py 43
Figure 54. First chronos mode where print density = 0.085 for ink-jet print zone 44
Figure 55. Modal energy distribution at print density = 0.085 for ink-jet print zone simulation 44
Figure 56. Spatial modes 1-2 (from top to bottom) at print density = 0.085 for ink-jet print zone simulation 45
Figure 57 . Spatial Modes 3-4 (from top to bottom) at print density = 0.085 for ink-jet print zone simulation 45

Figure 58. Generalized Binary Search Hopf Bifurcation 47
Figure 59. Economy- Sized SVD diagram [1] 53
Figure 60. Limit cycle schematic diagrams [73] 53
Figure 61. Kelvin- Helmholtz instability observed in the upper cloud region [74] 54
Figure 62. Kármán vortex Street with cylinder cross flow @ R=140, V=1.4cm/s,Ø=1cm [75] 54
Figure 63. 2-D vortex pairing/merging of experiment dye visuals (a)-(c) & DNS (d)-(f). snapshots are taken (a),(d) before merging (b),(e) during merging & (c),(f) after merging [76] 55
Figure 64. Poiseuille velocity profile with Bottom axis as streamwise velocity and left axis as Y position perpendicular to direction of flow 56
Figure 65. Vorticity probe for Re = 40 condition for restart facility used as initial condition for supercritical cases 57
Figure 66. Subcritical Reynolds number case demonstrating volatile initial transients at time < 0.1 57
Figure 67. DMD time dynamics find peaks used to find frequency st08-NS case where one unit time is 1000 snapshots, since down sampling = 11 and dT = 0.001 from DMD_ Modred_Kutz.py 58
Figure 68. DMD time dynamics find peaks used to find frequency VORTALL case where one unit time is 50 snapshots since dT = 0.02 and down sampling = 1 from DMD_Test_Modred.py 58
Figure 69. Plot of vorticity probe for st08-NS case Re = 100 59

Table of Tables

Table 1. Cylinder wake simulation parameters for st08_navier_stokes_cylinder.py 16
Table 2. Script dependencies with Python 3.8.6 17
Table 3. Computation environment 17
Table 4. Ink-jet Printer Injection Zone OpenFOAM Simulation Parameters 33

Table of Code Snippets

Code Snippet 1. NumPy SVD implementation 21
Code Snippet 2. Dask Compressed SVD Implementation 22
Code Snippet 3. SciPy.sparse with the save_npz module to save the mass matrix for each simulation case in *st08_navier_stokes_cylinder.py.* 22
Code Snippet 4. SciPy.sparse.linalg library with eigsh module which allows snapshot method of POD to be employed 23
Code Snippet 5. Modred Library SVD Implementation POD_MODRED_Kutz.py 23
Code Snippet 6. Initial DMD algorithm implemented in DMD_MODRED_Kutz.py 26
Code Snippet 7. DMD Modred algorithm implemented in DMD_MODRED_Kutz.py 26
Code Snippet 8. AR(2) root calculation and reconstruction AR(P)_Complex_Frequency_Spectra.py 29
Code Snippet 9. FFT implemented in CHRONOS_FFT.py 31
Code Snippet 10. DMD_MODRED_Kutz.py line 123-139 demonstrating find_peaks for frequency of time dynamics 59
Code Snippet 11. VAR(p) and Prony reconstruction credit to G. D. McBain from VAR_vorticity.py 60

Nomenclature

α	= *Angle of Attack (°)*	*EVP*	= *Eigenvalue Problem*
D	= *Diameter (m)*	*HFM*	= *Hidden Fluid Mechanics*
f_B	= *Beat Frequency (Hz)*	*FFT*	= *Fast Fourier Transform*
f_N	= *Natural Frequency (Hz)*	*FSI*	= *Fluid Structure Interaction*
f_{VS}	= *Vortex Shedding Frequency (Hz)*	*HODMD*	= *High Order DMD*
L_0	= *Stroke Length (m)*	*IBPM*	= *Immersed Boundary Projection Method*
Re	= *Reynolds Number*	*KLD*	= *Karhunen-Loève decomposition*
Re	= *Reynolds Number Centerline*	*LDV*	= *Laser Doppler Velocimetry*
U_∞	= *Free Stream Velocity*	*LSTM*	= *Long short-term memory*
		PCA	= *Principal Component Analysis*
CFD	= *Computational Fluid Dynamics*	*PDEs*	= *Partial Differential Equations*
DES	= *Detached Eddy Simulation*	*PEP*	= *Polynomial Eigenvalue Problem*
DFG	= Deutsche Forschungsgemeinschaft	*PIV*	= *Particle Image Velocimetry*
DMD	= *Dynamic Mode Decomposition*	*POD*	= *Proper Orthogonal Decomposition*
DNS	= *Direct Numerical Simulation*	*RANS*	= *Reynolds-Averaged Navier Stokes*
EFM	= *Equation Free Method*	*ROM*	= *Reduced Order Model*
		SPOD	= *Spectral Proper Orthogonal Decomposition*
		SVD	= *Singular Value Decomposition*
		VAR	= *Vector Autoregression*

Chapter 1 - Introduction

Understanding the structures in unsteady flow conditions provides insights into their behaviour. This provides a pathway to controlling the flows in these conditions which has wide applications in many relevant systems and processes such as ink-jet printers. The computational methods discussed in this report are modal analysis algorithms applied to unsteady flows. There is significant interest in the snapshot-method variants of these techniques using discretised data. Problems with large spatio-temporal data sets in fluid dynamics are defined commonly through numerical or experimental data consisting of a set of flow-field values, such as velocity, pressure, and / or vorticity, mapped to a series of points over multiple time snapshots.

Measurements of velocity in highly complex fluid flows require experimental techniques such as Particle Image Velocimetry (PIV) and Laser Doppler Velocimetry (LDV). A key issue of analysing experimental data in comparison to direct numerical simulation (DNS) is that the system's matrix of partial differential equations (PDEs) is unknown for experimental data, thus a statistical technique is required to analyse stochastic fluid flow phenomenon to permit the determination of localised temporal coherencies. An example of such a statistical technique proper orthogonal decomposition (POD) is depicted visually in **Figure 1**.The identification of coherent flow structures allows a high-order data set to be represented by the superposition of a low-order dominant mode and other coherent features such as a Kármán vortex street. An unsteady flow can be separated into its mean and modal components whereby each modal component may be represented by an eigenfunction which denotes a coherent flow structure that is largely time invariant.

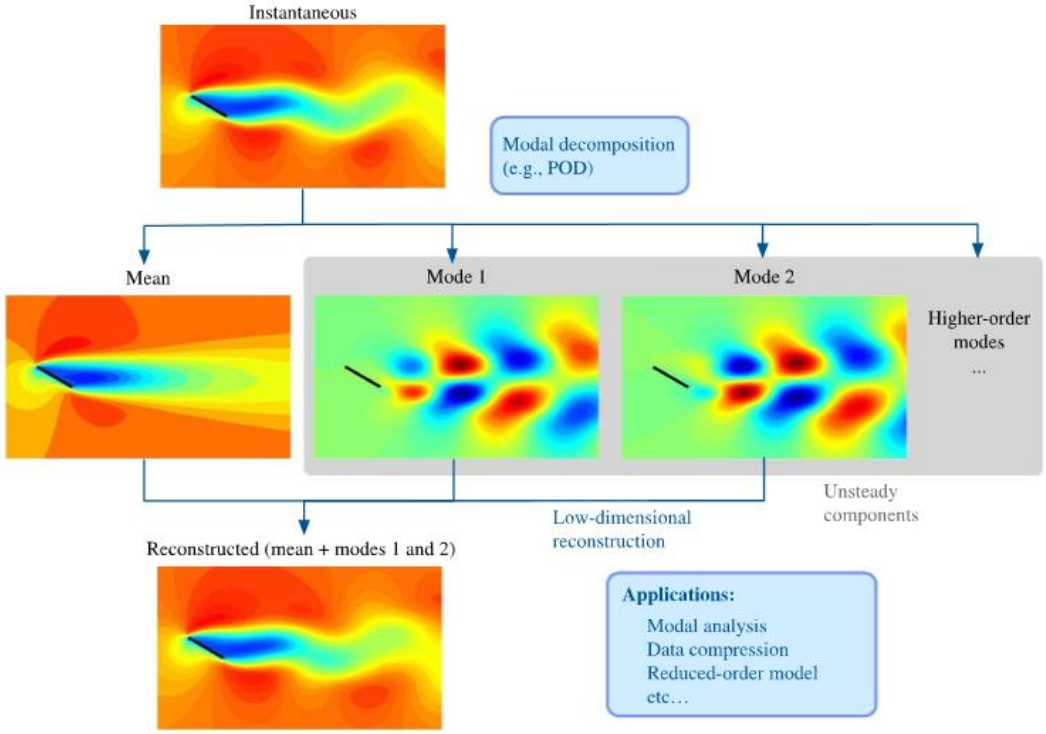

Figure 1.Modal decomposition of 2-D flat-plate wing Re=100 α=30° [1]

POD was established in the fluid dynamics community by Lumley [2]. He proposed a means of analysing turbulence through maximising a deterministic projection upon the velocity field. POD is an effective algorithm for obtaining the most energetic modes to identify dominant features and fluid events (coherent structures) for experimental data and numerical simulations.

POD is also known as Principal Component Analysis (PCA), singular system analysis, singular value decomposition (SVD), the Hotelling transform and was largely derived from Karhunen-Loève decomposition (KLD) [3]. POD analysis employs the use of the SVD, which was established by the Eckart-Young theorem to determine eigenvalues and eigenvectors or POD modes [4]. Velocity-state vectors are represented generally by an $n \times m$ matrix where n represents the degrees of freedom related to the dimension of the vector space and m represents the number of time snapshots, as shown in equation (2).

Lumley's definition of POD employs a model-based approach and considers an $n \times n$ eigenvalue problem (EVP); it is referred to as classical POD. Sirovich suggested a method, commonly called snapshot POD, where this problem can be reduced to an $m \times m$ eigenvalue problem, whereby $n >> m$ [5]. The snapshot method does not involve the covariance matrix and uses what is referred to as the economy-sized SVD (equivalent to an $n \times n$ EVP) instead of an $m \times m$ eigen-decomposition to achieve a matrix reduction. Coding Sirovich's snapshot method is simple and is less computationally costly than classical POD as it involves solving a singular value decomposition [6, 7, 4].

The SVD method to find the modes of the complete X matrix is shown in the following equations, noting that the economy SVD (using the truncated X matrix) method is faster than considering the full matrix. The inputs of each time snapshot components can be described by equation (1) where $q(\xi, t_i)$ can either represent a scalar (e.g. temperature or pressure) or vector field (e.g. velocity, vorticity) defined at a set of spatial points related to the term ξ, over one, two or three dimensional space at a discretized time, t_i . The temporal average of these scalars or vector is represented by $\bar{q}(\xi)$, thus by subtracting the mean (average) from the scalar or vector field the variation (perturbation), $x(\xi, t_i)$ can be resolved.

$$x(\xi, t_i) = q(\xi, t_i) - \bar{q}(\xi) \tag{1}$$

The perturbation term can be considered a stacked column vector, as the spatial points are synonymous to grid points which do not move but the scalar or vector values at the spatial points vary, hence the column vectors can be represented by the following matrix X.

$$X = [\, x(t_1)\, x(t_2) \ldots\ x(t_m)] \in \mathbb{R}^{n \times m} \tag{2}$$

The complete matrix can be decomposed using a singular value decomposition where $\mathrm{U} \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times m}$ and $\mathrm{V} \in \mathbb{R}^{m \times m}$ [1].

$$\mathrm{X} = U \Sigma V^{*} \tag{3}$$

The spatial modes can be represented by left-singular vectors in U and time coefficients by ΣV [8]. A more complete overview of POD analysis can be found in Taira et al [1]. It is also important to note that the snapshots do not necessarily need to be equidistant in time to calculate POD modes but are required to be equidistant in time to calculate DMD modes [4].

Dynamical Mode Decomposition (DMD) is a model-based approach which can be used for continuous dynamical systems in general. It was first introduced into fluid dynamics community by Schmid and Sesterhenn [9]. The method identifies coherent flow structures by determining the most energetic modes and correlative behaviors of the fluid flow for continuous dynamical systems. An example of this method is outlined in **Figure 2** below. Following the identified computational limitations of non-discretised analysis, a snap-shot method was introduced by Schmid using a data-based approach, with the resultant modes being equivalent to global linear stability analysis for linearized flows [10]. In contrast, DMD analysis of a nonlinear flow produces representations of a linear tangential approximation of the flow and

outlines dominant fluid features which can be identified as coherent flow structures. DMD is based on the Arnoldi algorithm [11], for a fixed sampling rate i.e. a single frequency [12]. The DMD snapshot-based method is intended as an equation free method (EFM) for use in large fluid flow datasets without fluid flow equations [13]. An EFM does not require prior understanding of the physics of the flow to utilise.

Koopman analysis is yet another computational method used for modal analysis with a great overview provided by Taira et al [1]. The DMD algorithm was redefined by Kutz et al. with a more rigorous representation of time series data which allowed the time series data to be non-sequential when the DMD method was applied to a dataset i.e. sequences could be missing [12]. This time order characteristic of DMD forms the basis of sparse DMD which is used for sensor & control applications for flow regimes [13]. DMD for a dataset can be broadly considered an eigendecomposition of an approximating linear operator whereby the independent time series representing the number of snapshots must be uniformly distributed [12].

Equation (2) and equation (3) used in POD are also used in DMD to develop the spatial modes. The matrix X' is defined as X shifted by a time-step.

$$X' = [\, x(t_2)\; x(t_3) \ldots\; x(t_{m+1})] \in \mathbb{R}^{n \times m} \tag{4}$$

The matrix, A can be formed by taking the product of the pseudo-inverse (denoted by $*$) of X.

$$A = X'X^* = X'V\Sigma^{-1}U^* \tag{5}$$

The matrix, $\tilde{A}$ serves as an approximation for the Koopman Operator for the relationship between X' and X .

$$\tilde{A} = U^*AU \tag{6}$$

The matrix is then subject to an eigendecomposition where columns of $W$are eigenvectors and Λ is the diagonal matrix of eigenvalues, λ_k.

$$\tilde{A}W = W\Lambda \tag{7}$$

DMD modes are found as the columns of Φ. A more complete overview of DMD can be found, described by Kutz et al. [13].

$$\Phi = X'V\Sigma^{-1}W \tag{8}$$

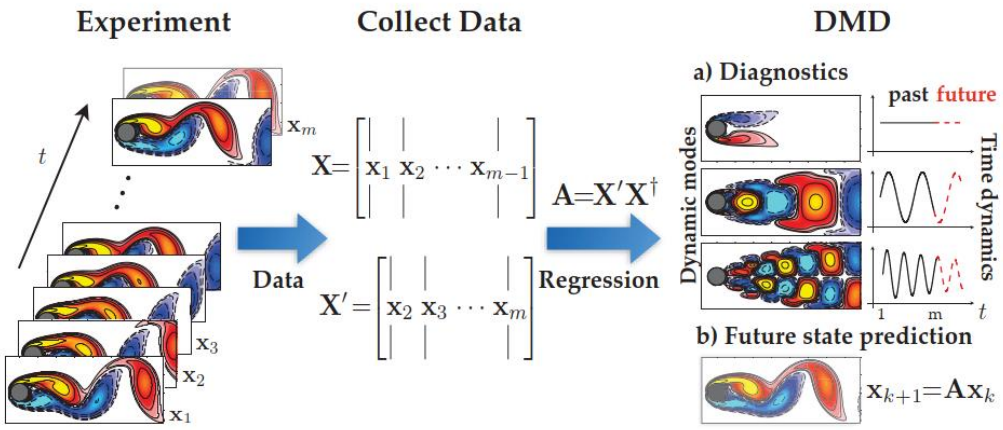

Figure 2. Simplified diagrammatic overview of DMD on fluid flow [13]

POD and DMD are both data driven methods for obtaining modal components of a fluid flow. Modal components are ordered and guaranteed to be orthogonal using the POD snapshot algorithm however it requires sequential time series data which DMD does not. The resultant modes found using DMD are, however unordered and not necessarily orthogonal. POD and DMD can be considered a form of unsupervised machine learning with respect to dimensionality reduction applications. Thus, once coherent structures are identified, these algorithms can be employed as a pre-processor for supervised learning algorithms useful for prediction and control of short-time future states [14].

Chapter 2- Literature Review

2.1. POD Analysis of Fluid Interactions

The fluid-bluff body interaction and subsequent vortex shedding is a well understood phenomenon. Study of bluff bodies allows for a clear and concise comparison between different computational methods for identifying coherent flow structures in an unsteady flow, as well as other flow features.

Usage of the POD-Galerkin projection method allows for comparison between computational times and accuracy of a reduced order model (ROM) and high order DNS model [1]. A comparison between the full order model using Navier-Stokes equations in a parametric study to determine the lift and drag coefficients and a reduced-order POD-Galerkin projection in a 2-D domain outlined that ability of POD to effectively compress spatio-temporal flow data for fluid-interaction DNS with $Re = 100$ and uniform horizontal flow [15]. For a modal decomposition of ten modal values, it was found that the lift and drag coefficients could be represented with an error of 1.89% and 6.43% respectively in comparison to a full order Navier-Stokes model with a comparative computation speed up factor of approximately 650 for a POD ROM [15]. POD is thus a significantly effective technique in reducing computational time for identification, calculation, and representation of features such as drag and lift coefficients compared to using Navier-Stokes equations. Similar parametric studies have been completed on bluff bodies such as on the DU91(2)250 aerofoils whereby extensive computational fluid modelling of the positioning of gurney flaps and microtabs with respect to its effect on the optimal lift-drag ratio was conducted [16]. Through POD-Galerkin projection of Reynolds averaged Navier-Stokes (RANS) model to a ROM the optimal lift-drag ratio can be recursively interpolated for control applications with respect to varying angle of attack, requiring a low number of computational resources to resolve, this also allows for prediction of a short-time future state within a selected domain. The study, however, is limited by assumed constant conditions of $Re = 106$ and $U\infty = 56.48 m/s$. Further data is likely required to consider control applications and variable environmental conditions for practical purposes.

POD beyond simply reducing orders of calculation allows a comprehensive analysis of coherent structures correlated with modal shapes of determined modal numbers to be conducted with low computational processing times. The characteristics of POD are particularly useful in assessing the effectiveness of varying flow control techniques such as piezoelectric actuators, wavy or rough surfaces and synthetic jet control, among others. A 2-D study was conducted to examine the influence of a surrounding porous material on the wake topology from a fluid-cylinder interaction, with the first four modes representing 46.3% and 27.0% of the total kinetic energy for the bare and porous cylinder, respectively [17]. By considering the first four POD modes of phase resolved PIV data it could be confirmed across several flow visualization techniques that the region of vortex shedding was less energetic and shifted further downstream due to the additional porous material in the porous cylinder case, as shown in **Figure 3** [17]. The porous material was cause of a wider wake topology with a correlated reduced vortex shedding frequency as corroborated by Klausmann and Ruck [18]. Further work may be completed to assess the impact of varying porous layer thickness or porosity of both open and closed cells on wake topology using POD analysis to improve understanding of a porous layer as a control flow technique.

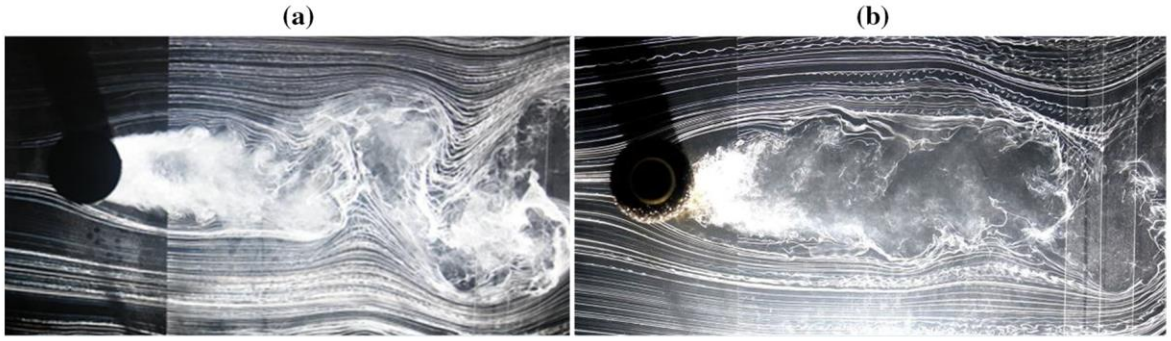

Figure 3. Smoke-wire visualisation of the cylinder wake topology (a) bare cylinder; (b) porous cylinder [17]

Active flow control techniques generally have a higher degree of dynamical control due to the ease with which flow parameters can be modified. Synthetic jet control is an active flow control technique. It normally involves oscillation of a flow due to actuation of a diaphragm, from an enclosure to an orifice. Synthetic jet control was assessed for a cylinder cross flow of $\mathrm{D} = 30\mathrm{mm}$ for $\mathrm{Re} = 950$ and 1800 of varying stroke lengths and excitation frequencies. Using POD it was found that the control effect of the synthetic jets could be represented by two dimensionless parameters, the momentum coefficient and stroke length [19]. The work is limited by insufficient comparable data for flow cases, for example there are only two unique instances of non-dimensional stroke length ratios considered within the study: $\mathrm{L_0/D} = 1.5$ and $\mathrm{L_0/D} = 3.3$.

The common use of averaging calculations across temporal and frequency domains for velocity fields hinders the improved understanding of non-linear dynamics. This is particularly true in the case of phase averaging to understand fluid-structure interactions. POD offers a means of considering dynamics due to perturbations where the mean component can be removed from the dynamics. A study of vortex induced vibrations with varying free stream velocity on a circular cylinder for $D = 25.4\mathrm{mm}$ and $2150 < \mathrm{Re} < 8530$ produced an approximate relationship between the vortex shedding, f_{VS} and beat frequency, f_B, with respect to the natural frequency, f_N and distinguished by the third mode pair [20].

$$\frac{f_B}{f_N} \approx 1 - \frac{f_{VS}}{f_N} \tag{9}$$

Analysis of the POD modes for vortex induced vibrations in terms of the beat frequency allows higher order modes to be considered. This study furthers understanding of non-linear dynamics due to the inclusion of the small vortices which cannot be examined with phase-averaging techniques due to this method's inherent characterization of dynamic frequency variation. It is hoped that a greater understanding of vortex induced vibrations in low order models for linearized flow through attribution of coherent structures to higher order modes may improve applications for energy harvesting [21]. Improved understanding of vortex induced vibrations through POD may be utilised in control flow techniques for suppression of oscillations in fluid-structure interactions as demonstrated by Chen et al. [22] .

2.2. POD Analysis of 3-D Complex Flow in Practical Applications

POD has seen further usage in 3-D flow analysis for comparing experimental data to numerical simulation to assess the effectiveness of flow control techniques. Axial water injection was investigated as a novel flow control mechanism to mitigate the flow instability associated with pressure pulsations in hydraulic turbines in an experimental setup [23]. The research was continued in a new study using the old experimental velocity datasets which were compared with an ANSYS Fluent numerical simulation using a zonal-DES turbulence model to investigate the influence of water jet injection. Individual POD modes reconstructed from the numerical simulations swirling flow were of particular interest: to minimize the radial velocity of the vortex rope at the critical threshold discharge value (11.5%). It was found that increasing the jet discharge value from 2% to 11.5% significantly reduced the dimensionless amplitude of the pulsations of the first mode. It was also observed that the mode shape for static pressure and radial velocity of the first mode shifted further down the diffuser, in the z-axis [24]. This shift confirmed the effectiveness of jet injection as a control flow technique in suppressing pulsations as seen when comparing **Figure 4** and **Figure 5**. Further study is needed to parametrically investigate an optimal value of jet discharge rate, while considering and minimizing volumetric losses to improve overall efficiencies of the system as well as develop dynamic control functionality [24].

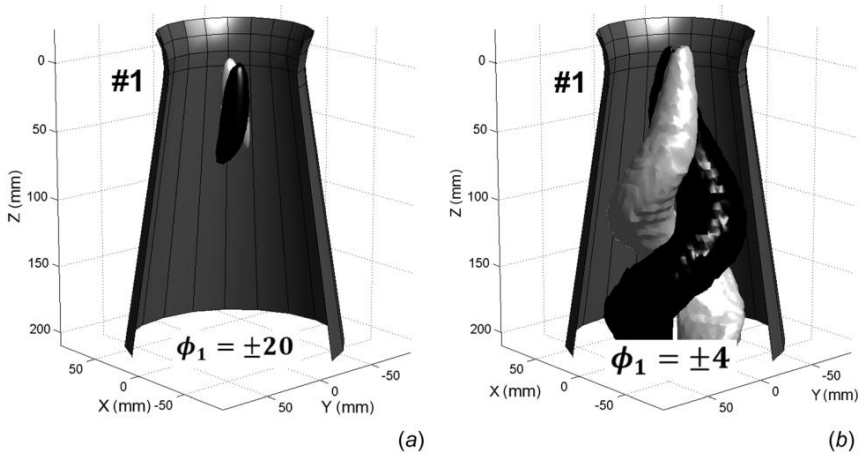

Figure 4. Modal shapes of vortex rope (a) static pressure and (b) radial velocity mode 1 at 2% jet discharge value [24]

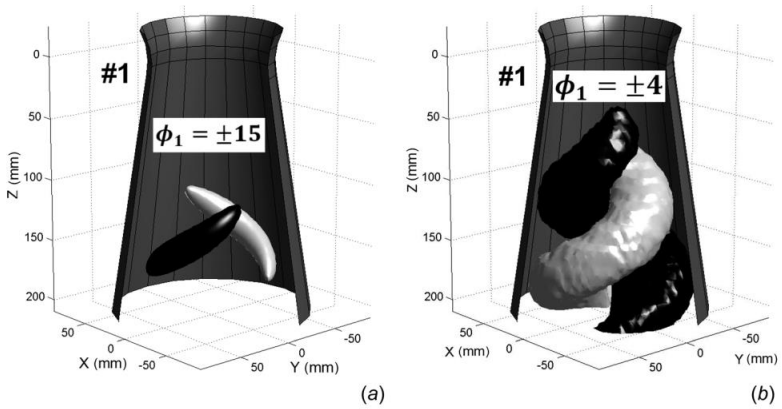

Figure 5. Modal shapes of vortex rope (a) static pressure and (b) radial velocity mode 1 at 5% jet discharge value [24]

POD although a useful tool for identifying coherent structures can be limited by the presence of turbulence in highly complex flow. The use of variations of the POD algorithm may help alleviate some of these limitations, such an example is the advent of extended-POD whereby pressure and velocity data can be coupled to uncover correlations between flow events and pressure variations as originally described by Maurel et al. [25]. The flow instabilities within a steam turbine control valve were investigated using a combination of POD and extended POD analysis by *Wang* et al to improve operational safety practices [26]. The first three modes of POD modes represented 9.6%, 5.7% and <5% of the total pressure fluctuation, respectively indicating the highly chaotic nature of the pressure variations, however insight was gained on the coherent structures [26]. A correlation was identified between positive pressure variations and synchronous oscillation from the impingement jet flow as exemplified in extended mode 1 in **Figure 6** . Implementation of control flow techniques may assist in reducing the chaos of the flow so a more thorough quantitative investigation can be conducted beyond qualitative relations between pressure and velocity fields. Furthermore, the assumption of circumferential symmetry for vortices may be invalid for the chaotic pressure variations and it is suggested the pressure fluctuations be studied on an additional orthogonal plane, so that the full 3-D turbulence effects may be studied [27].

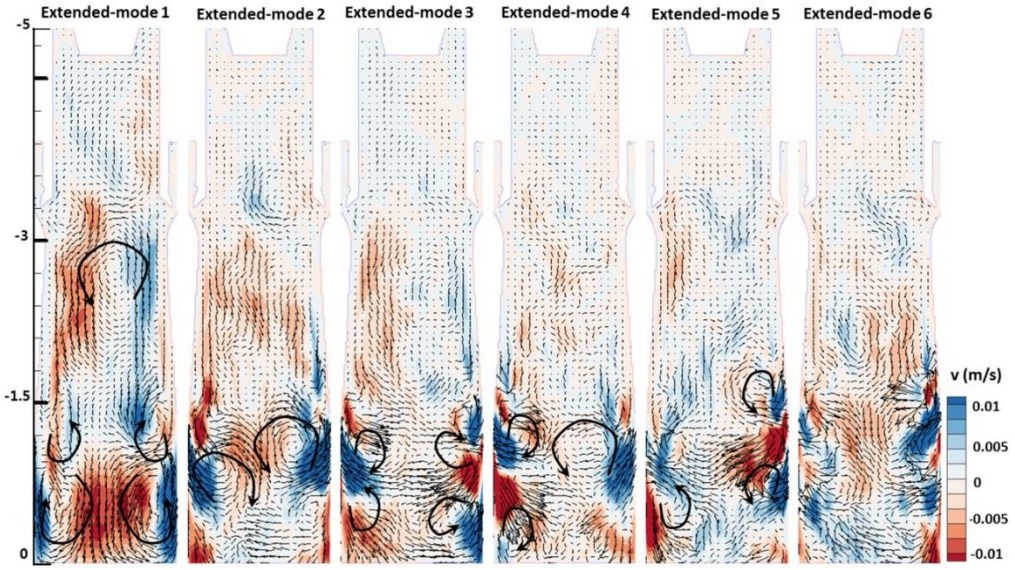

Figure 6. Control valve extended-POD modes for pressure-velocity coupling field, XY plane [26]

2.3. DMD Analysis of a Fluid Interactions

DMD analysis is a useful tool for determining specific dynamic structures of a fluid. However, it is difficult to determine the correct ranking of the modal decomposition and determine the relevancy of modes as they are not orthogonal, like in POD. Although, prior knowledge of the dynamical system is not required to determine modes, insight using DMD is lost if prior physical information about the system is unknown. A suggested method of easing the subjective relevancy of the modes involves the use of specific and well-defined criteria. A criterion I_j allows the influence of the temporal (time) coefficient $b_{ij}(t)$ on the entire time domain to be considered rather than for an instant, allowing for an effective method of ordering dominant DMD modes to determine the relevancy of coherent structures [28].

$$I_j = \int |b_j(t)|dt \approx \int_{i=1}^{N} |b_{ij}|dt \tag{10}$$

Similarly, to how POD has been applied to numerical simulations and experimental data, DMD can inform on the dominance of coherent structures for modes of single frequencies. The geometry of various differing bluff bodies can greatly impact the wake generated by a fluid flow. A comparison between the fluid interaction of a square cylinder and wavy square cylinder for Re= 4075 demonstrated the formation of secondary vortical structures associated with the leading wave edge as opposed to the square cylinder where no secondary vortical structures from Von Kármán vortex shedding could be identified [29]. The DMD method resolved spatial patterns, demonstrating that modification of surface topology by a wavy leading edge can provide passive flow control, increasing dissipation of vortical structures, as outlined in **Figure 7**. However, insight into the relevancy and ordering of each mode to the entire time domain could be improved by inclusion of the criterion discussed above in addition to the use of Fourier transforms to understand the time-frequency relationship.

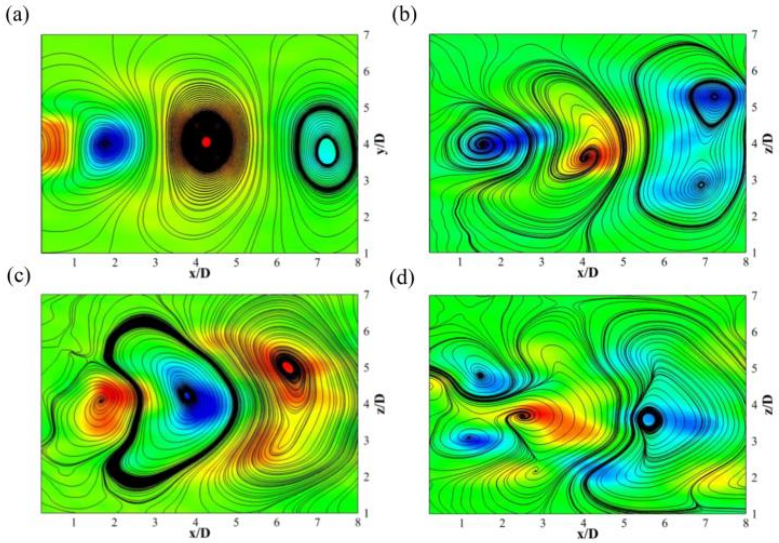

Figure 7. DMD spatial patterns of mode 1 (a) square cylinder (b) wavy-square cylinder node plane (c) wavy-square cylinder middle plane (d) wavy-square cylinder saddle plane. [29]

A major challenge in utilizing DMD modes for the identification of coherent structures is determining the contribution of the real and imaginary parts of the spatial mode. A study by Wang et al. suggests DMD spatial modes contain an imaginary and real part with both parts at different phases through mathematical manipulation and visualization, as depicted in **Figure 8** for a diffuser setup [30]. This visualization of coherent structures assists in the interpretation of the DMD modes with respect to the contribution of imaginary and real parts, as a spatial relation is apparent. As such DMD modes can be considered to contain

a constant, singular complex frequency for each mode, with a phase shift apparent between the real and imaginary parts of a mode. This relationship was earlier suggested by Goza et al. with modal analysis of flag flapping for a fluid structure interaction [31]. However, the mathematical relationship was not explicitly stated in the earlier work.

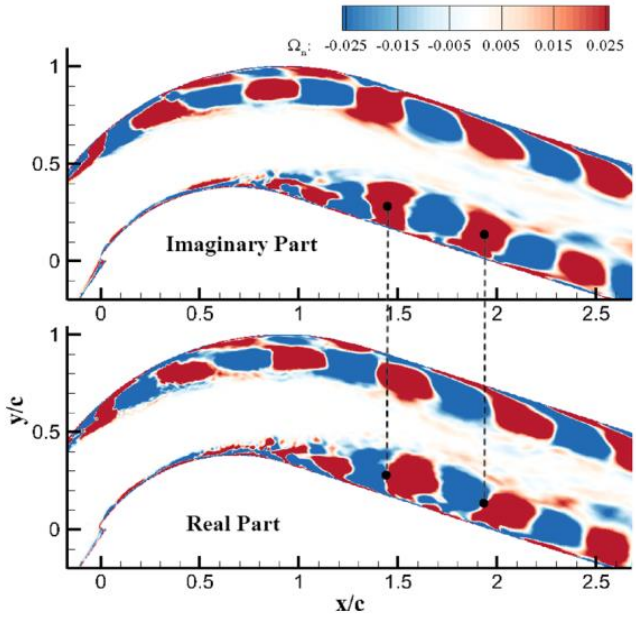

Figure 8. Imaginary and real parts of spatial DMD mode (St = 0.173) for a diffuser [30]

The DMD algorithm approximates the Koopman operator under certain conditions whereby Koopman modes can be harmonics of a fundamental real frequency as they are integer multiples of one another. Bagheri demonstrates that it is necessary to consider different regimes of flow with respect to time and use different analysis methods and some dynamics do not necessarily carry over into other intervals [32]. This is especially true with regards to the limitations of the FFT algorithm as the approach requires a predetermined set of frequencies and a periodic dataset which is only true for interval IV as shown in Figure 9 [32]. Other approaches are available that may overcome these limitations. Perhaps, autoregression and vector autoregression may provide further insights into these dynamics beyond those explored by Bagheri.

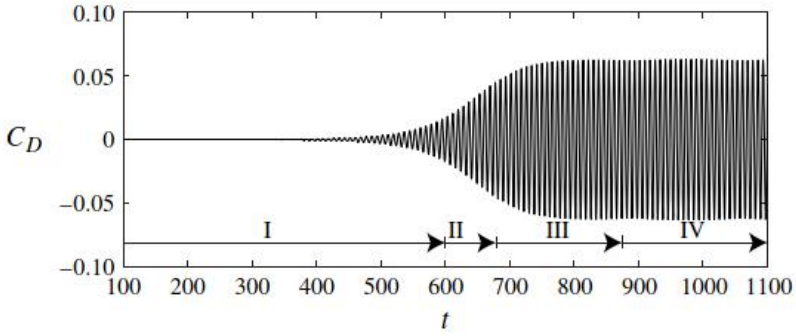

Figure 9. Drag coefficient as a function of time for a cylinder at Re=50 [32]

2.4. POD and DMD Analysis Comparison Studies

POD spatial modes generally contain a mix of frequencies although Spectral Proper Orthogonal Decomposition (SPOD) has been developed to alleviate this issue by resolving in a frequency rather than

temporal domain [1]. DMD can resolve spatial modes for singular frequencies thus it may be more useful than POD to significantly uncouple modes with more fragmented and detailed coherent structures [33]. As POD modes may share frequencies, it can be necessary to complete both a POD and DMD analysis on a fluid flow to identify coherent structures. Furthermore, DMD analysis can require a priori knowledge of modal energies in relation to the total kinetic energy to determine relevancy of modes especially for noisy data. This limitation gives rise to the in-tandem approach of applying both POD and DMD techniques to unsteady flow experiments.

POD modes can be separated into their spatial components independent of time and time component independent of space, commonly referred to as topos and chronos respectively. The chronos for a POD mode may contain a spectrum of frequencies. When detailed analysis of the frequency domain is required DMD can be preferred for further insight into these coherent structures as singular frequencies can be identified rather than a mix which POD chronos modes contain. This methodology has proven useful for analysis of combustion chambers; for example it was demonstrated that by modifying the angle of fuel injection from 150° to 90°, a squish dominated frequency spectrum (6-8.5khz) informed by DMD analysis could have its amplitude reduced (sound pressure level), which coincided with a shift of POD spatial modes as outlined in **Figure 10** and **Figure 11** respectively [34]. Although efficiencies of the modification were not considered, this control feedback loop of adjusting parameters informed by coherent structure identification stresses the ability of POD and DMD in informing or optimizing a control algorithm. It is also significant that although POD modes appear shifted, this is largely an observed approximation.

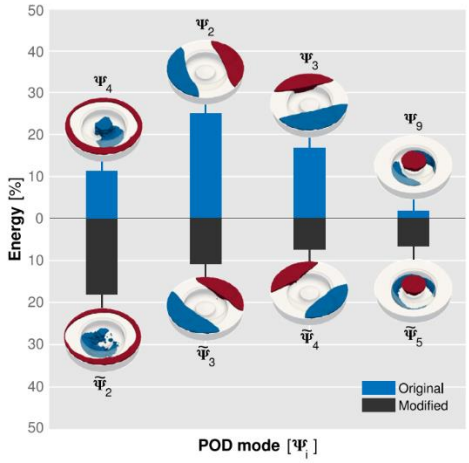

Figure 10. Shifted POD modes due to modified fuel injection spray angle [34]

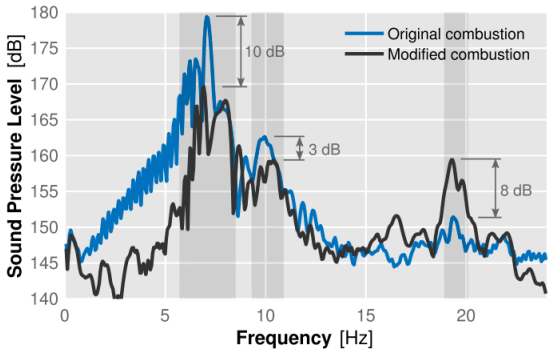

Figure 11. Reduced sound pressure due to modified fuel injection spray angle [34]

The applications of DMD and POD analysis go beyond the analysis of coherent structures in fluids, the insights provided on modal analysis also allow a combined analysis of a fluid-structure interaction. It is important to analyse these systems with respect to the coupled physics instead of independently to identify dominant features and improve physical understanding. A major limitation of a POD and DMD study completed by Muld et al. on a high-speed train [35] was the methodology in which the fluid and structure were not clearly coupled in the matrix formation, although this would have made the simulation quite expensive. It is also important to recognize that the model was analysed in 3-D which allowed a methodology with an improved analysis of turbulence effects compared to an equivalent 2-D study [27].

A method for analyzing a coupled fluid-structure system with respect to a weighting matrix was developed by Goza et al [31] for a flag flapping at Re = 500. The method found good agreement between the DMD and POD modes as it was determined there was a dominant frequency related to the limit cycle behaviour of the oscillating flag. The use of SPOD is suggested as an intermediary method between DMD and POD as it can isolate coherent structures at distinct frequencies in its modes. A key trait shared between DMD, and POD analysis is the similarity of spatial modes for some flows periodic in time [36]. This is confirmed by studies conducted on fluid-structure interactions of the Crevalle Jack's caudal fin for Re=500 and 4000 by Khalid et al [37]. Furthermore, the study identifies the occurrence of quadfurcations as coherent structures where the first mode accounts for the separation of each of the four vortex tubes and second mode relates to vortices pairing, as is visually depicted in **Figure 12.** The study successfully identifies coherent structure for a 3-D body using a modal analysis method for fluid-structure interactions inspired by Goza et al.

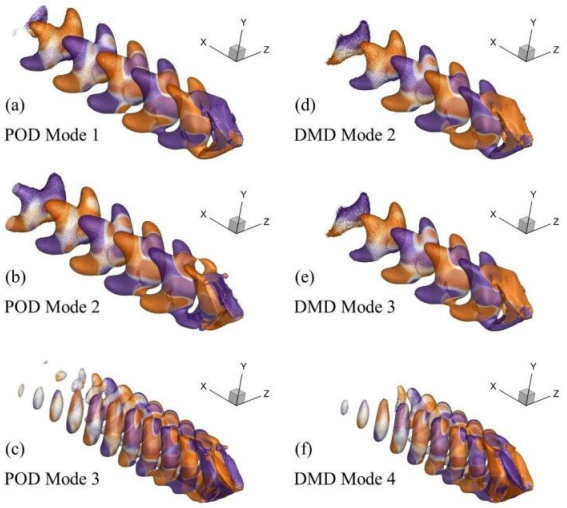

Figure 12. POD and DMD modes for crevalle jack caudal fin FSI, Re=500

A generally accepted limitation of POD is its inability to isolate individual frequencies within a mode. Rather, POD produces a spectrum of frequencies. However, in cases where there is no spectrum of frequencies or where spectrums can be broken down into individual frequencies through post processing methods, this limitation is less prevalent for the dominant or leading mode. Burtsev et al. have seemingly implied this important case exists for a cylinder flow of Re = 60 whereby the roots between POD and DMD in a linear regime match those of the Bi-Global EVP [38]. It is unclear which post processing steps if any were taken to arrive to these results with DMD such as down sampling or subsampling among other possible steps. Further clarification is required upon what factors impact on DMD and POD roots being equivalent allowing individual frequencies to be characterized.

	T1	S1
BiGlobal EVP (Nektar++)	$\pm 0.11926 + 0.04762i$	$0 - 0.05145i$
POD, DMD (linear)	$\pm 0.11926 + 0.04761i$	$0 - 0.05371i$
POD, DMD (nonlinear)	$0.13613 + 0i$	$0 - 0.05048i$

Figure 13. Leading travelling and stationary modes canonical two-dimensional cylinder at Re = 60 [38]

2.5. Machine Learning

DMD and POD can be thought of as linear encoder methods that are able to compress and decompress high dimensionality fluid data [39]. Such an example of this is viewed in research conducted on a mixed flow pump by Han and Tan, showcasing the ability to extract coherent structures using POD and DMD as well as reconstruct the flow field with a high degree of accuracy as seen in **Figure 14** [40].Such reconstruction of the flow field from dominant modes forms the basis for prediction modelling and prediction applications. A key limitation of POD and DMD is its ability to account for non-linearities, which auto-encoder networks have been found to alleviate as has been researched with the limitations of PCA, closely resembling the limitations of POD [41]. It is possible to use a neural network with encoders based on fundamental physics-informed knowledge such as energy conservation laws and Newton's equations to also improve the accuracy of the model. This deep learning framework has been referred to as hidden fluid mechanics (HFM). HFM has shown, for example, success in modelling flow fields for informing on hemodynamics of a 3D intracranial aneurysm [42].

Machine learning has seen great success in the use of unsupervised learning in neural networks which can be used for the prediction of fluid flow features. A method comparing linear DMD and POD with respect to auto-encoders for long short-term memory neural networks and a POD-LSTM network was studied by Eivazi et al [39]. The study outlined potential for improved prediction of non-linear flow fields using auto-encoder networks with respect to the reduction of the mean square error [39]. There are however large limitations associated with such techniques, particularly computational time with DMD being three orders of magnitude faster for prediction than a relevant neural network [39]. Furthermore, selection of the size of the bottle neck layer which dictates the dimensionality reduction before reconstruction is an a priori proposition that requires a posteriori validation. This means that poor selection of the size of the bottle neck layer can severely slow down calculation time and may require neural networks to have their training conducted again. This limitation is particularly risky for analysing large data sets. Further research is required to understand the risk profile of bottle neck layer selection in relation to the study of flow fields to encourage research in this area.

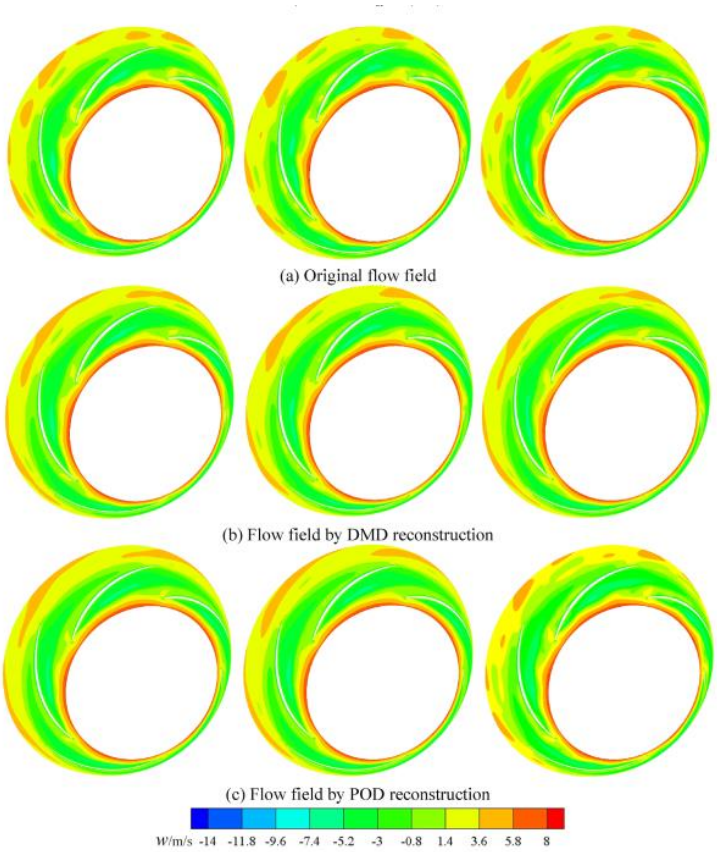

Figure 14. Reconstructed flow fields comparing POD and DMD for the tip leakage vortex in a multi-stage pump [40]

2.6. Ink Jet Print Zone and Tiger Stripes

It has been identified that minute changes in the stability of airflow dynamics within the print zone of inkjet printheads has a large impact on the output quality leading to issues such as the tiger stripe error [43].Tiger stripes occur as a result of the deposition of satellite droplets that disperse away from the main printhead flow which create an impinging jet flow effect with counter rotating vortices [44]. This issue becomes especially exacerbated when the distance is increased between the printhead and print material. Work has been conducted to significantly reduce the tiger stripe incidence in inkjet printing.However, these minimization strategies: blocking the flow by inducing a pressure gradient from the roller rotation and application of a pressure difference between the print zone ends. These strategies, however, do not completely eradicate the issue over all industrially useful configurations and operating conditions [45].

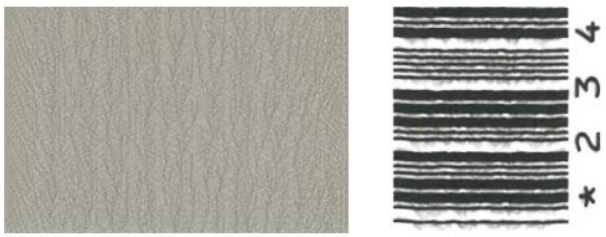

Figure 15. Solid block printing (left) and bar-code (right) showing the effect of airflow oscillations creating tiger stripes [43]

Aquino et. al. compared the amplitude of velocity oscillation at a probe point just downstream of the print zone as a function of the density of nozzles printing [46]. Figure 16 compares the square of the amplitude

of the oscillation with respect to print density. This shows that the onset of oscillations occurs approximately between print density values of 0.076 and 0.077, where print density is a measurement of the amount of ink measured normally with densitometer and pressmen [46]. The change in flow behaviour from steady to oscillatory is characteristic of a supercritical Hopf bifurcation [47]. The supercritical Hopf bifurcation marks the point of a transition from steady to unsteady flow as the Reynolds number or (in the case of inkjet printing) print density increases. It is expected that computational methods such as POD may provide a method to garner similar results which elucidate of the physics of the flow and inform on coherent features such as the critical point for other flow examples.

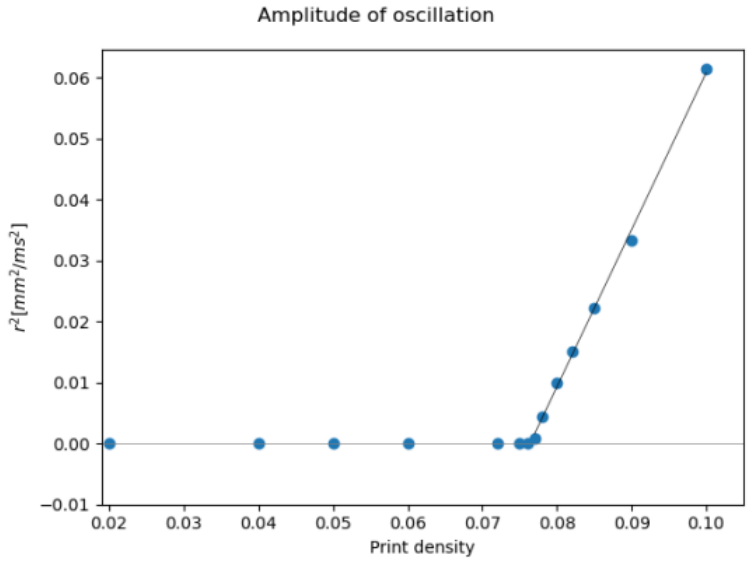

Figure 16. Bifurcation diagram - amplitude of oscillation squared (r^2) [mm^2/ms^2] vs. print density [46]

2.7. Supercritical Hopf Bifurcation on a Cylinder

A canonical bluff body for examining supercritical Hopf bifurcation is the circular cylinder [48, 49, 50, 51, 52]. Thus, it is a suitable problem for testing out the analyses developed herein. For the cylinder problem, a very large flow domain is required to remove any effect of the boundaries on the flow. Placing the cylinder on the centerline of the planar channel reduces the size of the domain, and thus computational cost for examining this problem. There are also numerical predictions of the critical Reynolds numbers in this configuration that are available for comparison [53, 54, 55].

The tiger-stripe problem in inkjet printers has barely been considered compared to the wide literature on flow over cylinders. There is a desire to understand the coherent structures with respect to bifurcation behavior in these flows [45]. This thesis aims to qualitatively identify coherent flow structures through visualizations using DMD and POD as well as other methods, hoping to explain the advent of supercritical Hopf bifurcations which have been observed in numerical simulations [46]. The methodology employed on the cylinder wake problem can then be applied in a generalized sense to the tiger-stripe problem.

Chapter 3 - Methodology

The 2-D cylinder wake flow example was first considered in the planning for this project to allow for a simple process of validation of the methodology presented against known and well researched results for a canonical problem. The 2-D cylinder despite being a relatively simple geometry, involves a rich level of detailed physics. Knowledge of applied mathematics is required to analyse the problem with a high degree of accuracy and insight. Traditional techniques such as bifurcation analysis can inform on the bifurcation point and coherent features of the fluid flow that are observed during this transition between steady and unsteady flow. Numerical simulations are sensitive to minute changes in initial conditions and equation solution methods ultimately have a large impact on the identification and accuracy of the determined bifurcation point. Theoretical mathematical analysis techniques can provide a benchmark for testing the numerical predictions and give insight into the problem over a wide range of parameter space for much less computational effort than simulations.

The aim of this methodology section is to propose, employ, validate, and verify statistical techniques using computational methods which may garner the same insights with a simpler methodology that is easier to repeat at scale. The cases where Reynolds number are near the observed bifurcation point are of particular interest in confirming previously observed results and providing further insight into supercritical Hopf bifurcation. The methodology presented will largely analyse results from a time window of flow whereby the flow is periodic and the oscillations in flow properties have reached a saturated amplitude i.e., Interval IV which is discussed later. The flow data simulation considered occurs where Re = 150 which is equivalent to Re = 100 for a circular cylinder wake flow, however the general process of methodology is applicable to other flow examples for a variety of Reynolds numbers. The impetus for using this time window is related to the later development of non-linearity and structure associated with a new attractor [3].

The methodology used to analyse the confined cylinder wake problem will then be applied to results from an ink-jet print zone simulation, to identify coherent features which corroborate with known printing dynamics and phenomena [43, 46, 44].

3.1. 2-D Cylinder Flow

The canonical confined cylinder wake problem will be examined in greater detail as it is a well-researched and well-defined simulation [32, 55]. It will form a testbed to develop modal analysis oriented computational techniques to identify coherent features which can be applied to more complex geometry. There is particular interest in the critical Reynolds number associated with the onset of the steady to unsteady flow transition in a fluid flow between supercritical and subcritical Reynolds numbers whereby the critical Reynolds number is identified by a zero-growth rate. The solution transitions from a breakdown of the steady flow to a periodic unsteady flow past this critical threshold. This zero-growth rate is representative of the fluid flow behaviors of the fundamental eigenmode. For a supercritical case some saturated amplitude behaviour would be expected whereas for subcritical cases a steady state would be expected. Various similar values have been observed for the critical Reynolds number, Gallaire et al. observed that the critical Reynolds number occurred at Re_{crit} = 46 [48]. Bagheri has provided a similar result for the critical Reynolds number with a more precise of value Re_{crit} = 46.6 [32]. Other research has indicated that the bifurcation point coincides with a critical Reynolds number of between 45-49 [49]. For the centerline Reynolds number this would be equivalent to approximately Re = 67.5 -73.5. Furthermore, other factors may impact the identified Reynolds number for the bifurcation point including:

- Width of the channel
- Inlet velocity profile
- Blockage Ratio

- Distance of cylinder from inlet and outlet
- Mesh settings (for a DNS)

3.1.1. Cylinder Wake Geometry Simulation

Several Python scripts were developed utilizing the scikit-fem library to create a geometrical cylinder model in a channel as described below in Table 1 and to solve the Navier-Stokes equations using a finite element method [56]. Numerical simulations of the canonical confined cylinder wake flow model were generated for multiple cases between Re = 70 and Re = 150 with adherence to the Deutsche Forschungsgemeinschaft (DFG) time periodic 2-D case [50]. This simulation will be referred to as the st08-NS case and is generated by the st08_navier_stokes_cylinder Python script or '.py' file. This simulation was generated to compare to existing studies and validate the implementation of POD, DMD and other Python scripts. These scripts were created to be later used for the more complex ink-jet printer injection zone geometric structure which was developed by Aquino et al.; using scikit-fem to solve the steady equations and to generate initial conditions and unsteady simulations were created using OpenFOAM [46].

Parameters for the geometry are given below in Table 1 and the mesh is shown in Figure 17.

Table 1. Cylinder wake simulation parameters for st08_navier_stokes_cylinder.py

Parameters	Value
Domain Height x Length	0.41(H) x 2.2 (L)
Cylinder Radius	0.05
Circle Centre	0.2,0.2
Edge Size	0.025
Blockage Ratio	0.244
Number of Cells per snapshot(elements)	3073
Number of Points per snapshot(nodes)	1651
Total Snapshots	15000-60000
Time Step	0.001

The following mesh was generated using the dmsh Python library [57] . The scikit-fem simulation was able to provide pressure, velocity, and vorticity data per each nodal value for a range of snapshots. The inlet is defined as the coordinates (x,y) = (0.0,0.0) to (x,y) = (0.0,0.4) and the outlet is defined as (x,y) = (2.2,0.0) to (x,y) = (2.2,0.4).

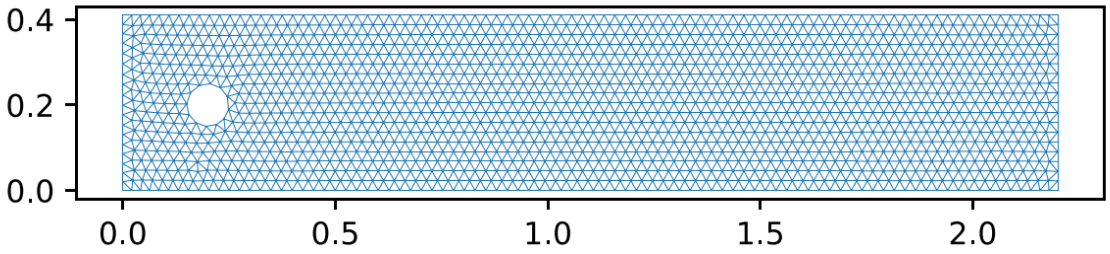

Figure 17. 2-D Cylinder flow geometry of a triangular mesh generated with scikit-fem

The Python simulation for a DNS solution to the confined cylinder wake problem and consequent post processing of matrices data for vorticity and other parameters was handled by the Python dependencies and their consequent child dependencies portrayed in Table 2. The libraries and software were selected for the project as they are open source and relatively portable which means that a non-commercial computing environment may handle the requirements of the simulation reducing the barrier to undertaking this study.

Table 2. Script dependencies with Python 3.8.6

Python Package	Version
scikit-fem [58]	2.5.0
dmsh [59]	0.2.16
h5py [60]	2.10
matplotlib [61]	3.4.2
modred [62]	2.1.0
pyamg [63]	4.1.0

The specifications of the computer and environment used for simulations and post processing are as follows in Table 3.

Table 3. Computation environment

Specification	Value
CPU	Intel(R) Core(TM) i5-9600KF CPU @ 3.70GHz (6 CPUs)
GPU	AMD RX 590 Series
RAM	16318MB
SSD	1TB
Operating System	Windows 10 Pro 64-bit (10.0, Build 19042)
IDE	Spyder 4.2.5
Python	3.8.6

The DNS was resolved by numerically resolving the incompressible Navier-Stokes equations using a finite difference method. The Navier-Stokes equations for mass conservation were considered with the assumption the fluid was incompressible and could be modelled as a continuous body.

In equation (11), L is the characteristic length in this case this is the size of the cylinder diameter D, and u is the flow speed. In this simulation u was considered as the maximum velocity flow vector of the parabolic fully developed Poiseuille flow which is imposed as an initial condition. Note that these dimensions are not indicative of a real-world simulation regarding scale.

$$Re = \frac{uL}{v} \tag{11}$$

It is important to note that discrepancies between the results achieved in the DNS may be different to other similar cylinder flow studies as they may use a plug flow where initial conditions of the flow are uniform parallel to the flow stream. Furthermore, mesh settings and finite difference methods may be different in approach. The canonical example of the cylinder wake flow for Re = 100 was examined by Kutz et al. for a plug flow with a dimensionless uniform initial velocity of one [13]. The equivalent case used to verify the simulation was examined for $Re_{centre\ line}$ = 150 whereby the max velocity was 1.5. It is important to

note how the velocity profile impacts on the Reynolds numbers which make these cases largely equivalent. As for an even Poiseuille velocity profile, the maximum velocity also known as the centerline velocity is 1.5 times that of the mean velocity as seen in equation (12), where u_m is the mean velocity, u_c is the maximum (centerline) velocity. This means that a critical Reynolds number of Re = 46.6 is equivalent to a $Re_{centre\ line}$ value of $Re_{centre\ line}$ = 69.9

$$Re_{centre\ line} = \frac{u_c L}{v} = \frac{1.5 u_m L}{v} \tag{12}$$

A restart facility was devised within the Python code motivated by the need to produce a linear stage of growing disturbance to verify Burtsev etl al's results as well as reduce initial transient features that are resultant of poorly defined initial conditions [38]. Different intervals are shown to exhibit different flow behaviour and have been defined by Bagheri as follows: interval I represents the exponential escape rate from the unstable equilibrium as the initial stage, interval II represents the growth from the interaction between the equilibrium and limit cycle and interval III represents the exponential relaxation rate due to the limit cycle and interval IV is the saturated growth stage marking a period of oscillations in the limit cycle [32]. For Figure 19 interval II – III is approximately represented by time 2 to time 8 from represented snapshots 2000-8000 and interval IV is represented by snapshots 8000-15000 i.e., time 8-15. It can be seen in Figure 19 that the restart facility produces the desired range of intervals from interval I to interval III thus it produces a linear stage of initial disturbance growth.

An example of a simulation with large initial transients can be seen in Figure 18. for velocities v and u (for the y and x axis, respectively), for time, t, less than 4. At t ~ 2.5 there is a large change in both the v and u values which does not reflect the expected shape of the function which should resemble that of Figure 9 for interval I to interval III. This would require an increased number of cycles for the equivalent growth stage in the cases being examined.

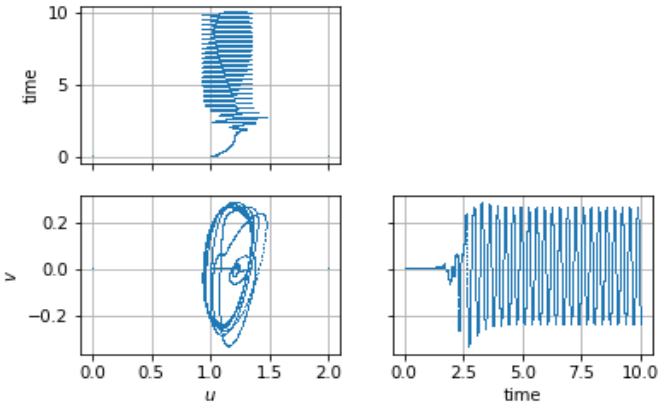

Figure 18. Velocity probe at coordinate (1.759,0.114) of st08-NS simulation without restart facility for Re = 90 where time is in unit time, where dT =0.001 and 1 unit time is 1000 snapshots

In contrast, the restart facility uses the solution for a subcritical Reynolds number in this case $Re_{centre\ line}$ = 40 as the initial condition which reduces the volatility of the initial transient dynamics. The restart facility implicitly assumes that the initial steady state dynamics for supercritical Reynolds numbers, $Re_{centre\ line} > Re_{crit}$ are approximately equal to the initial steady states of subcritical Reynolds numbers where, $Re_{centre\ line} < Re_{crit}$ and Re_{crit} is defined as the critical Reynolds number. Centre line Reynolds number from hereon will just be referred to as Re.

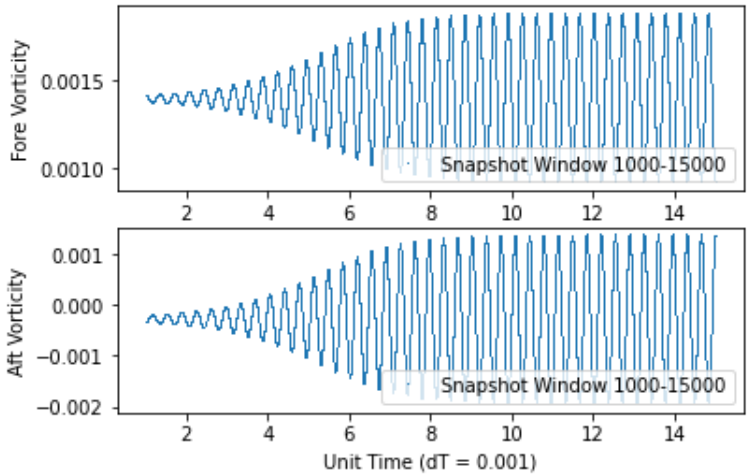

Figure 19. Vorticity probe for fore and aft stagnation points with restart facility for Re = 90

An example of a subcritical case is shown below in Figure 20. Interval I is comprised of initial transients which are largely superfluous and are resultant of stagnant initial conditions and other simulation settings. This stage is not present in Figure 20, as it has been cropped. An example of these transients can be found in the appendices in Figure 66. Interval II shows multimodal decay after the initial transients of interval I. In some cases, the results show several peaks representing the fundamental eigenmode and its harmonics which decay at different rates. This is not a very prominent stage in Figure 20 but likely explains the slightly reduced peak near snapshot 1100. Interval III is the linear decay of the fundamental mode whereby the signal can be largely interpreted to resemble that of trigonometric-exponential decay e.g., the product of an exponentially decaying curve and a sine wave. Interval III exists for snapshot 2000-45000 approximately. Past those snapshots, interval IV represents the stagnation where the amplitude of the decaying signal falls below the noise. This is shown from snapshot 45000-60000. It should be noted that the vorticity values for these perturbations are relatively quite small and Bagheri has not considered all the intervals for a subcritical Reynolds number case [32].

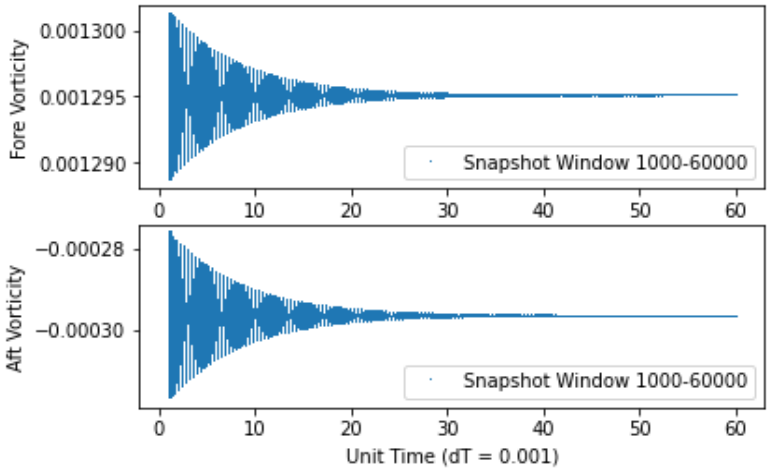

Figure 20. Fore and aft stagnation point vorticity probes for Re = 74

The simulation conducted produces an '.xdmf' file format which, for the purposes of analyzing the flow structures, can be read as a NumPy matrix of a desired shape with the number of rows equal to the number of nodal points and columns equal to the number of snapshots.

Reynolds numbers less than $Re = 80$ were run for 60000 snapshots with a time difference (time step value) of δt = 0.001 between each snapshot in relation to the Navier-Stokes numerical solver scripted in *st08_navier_stokes_cylinder.py*. Cases greater than $Re = 80$ in contrast were only run for 15000 snapshots also with a difference of δt = 0.001 between each snapshot. The cases near the bifurcation point and below the bifurcation point (critical Reynolds number) were run for a longer time window (i.e., more snapshots with same time step size) as this was necessary to reach the quasi-steady state for Interval IV for subcritical cases.

The results were first observed by reading the '.xdmf' files into the open source scientific visualization tool Paraview to observe the evolution of the dynamics for each case. Later necessary snapshots were plotted using the matplotlib library [61]. A script for plotting these snapshots can be found in the repository linked in the appendices, whereby the script *plot_snapshot.py* was used. For the given computational environment simulation generally took approximately 1hr and 6 minutes to complete for 60000 snapshots and approximately 17 minutes for 15000 snapshots. It should be noted that these simulations were performed in series when there was opportunity to process them in parallel. For example, a modified version of *st08_navier_stokes_cylinder.py* would be able to simulate multiple Reynolds number cases for approximately 1hr and 6 minutes. Optimization of this algorithm was not pursued due to time constraints of the project but would greatly improve simulation time and may be pursued in future works.

When comparing Figure 21 and Figure 23 it can be observed that there are similarities as the vorticity conditions for as a subcritical $Re = 40$ (similar to the plot observed in Figure 23) for the case in Interval IV is used as the in initial condition to create the simulation for Re = 150 as observed in Figure 21. This can be visually confirmed as there is a clear independent section between the red and blue vorticity stream in Figure 23. As expected in Figure 21 the vorticity streams are also independent in the wake of the cylinder, however as time evolves the growth rate increases leading to the vorticity dynamics observed in Figure 22.

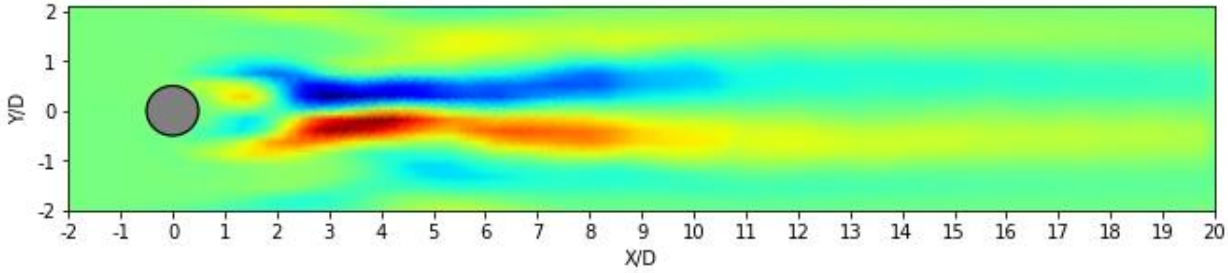

Figure 21. Plot of snapshot 500 for Re = 150 using matplotlib interval II, supercritical case generated using plot_snapshot.py

Figure 22 below is representative of a supercritical case for the confined cylinder wake problem, which, when viewed shows contours of y-velocity and it exhibits characteristics of a Von Kármán vortex street, whereby vortical structures exhibit oscillatory behaviour in vortex pairs. These simulations were then postprocessed using computational methods such as autoregression, POD and DMD among others to garner further insights into the fluid flow dynamics. Figure 22 corroborates with both Bagheri and Kutz et al regarding paired vortex shedding for supercritical cases past the bifurcation point which visually validates that the simulation has merit and matches the expected flow dynamics [13] [32].

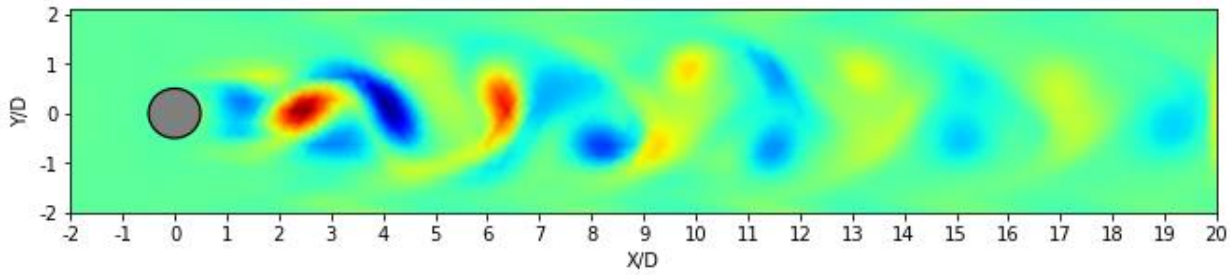

Figure 22. Plot of snapshot 11000 for Re = 150 using matplotlib interval IV, supercritical case generated using plot_snapshot.py

A subcritical case in contrast demonstrates Interval IV which decays to a steady laminar flow as time evolves hence achieving a flow where a probe response is below that of the noise. Such a snapshot of a subcritical case is demonstrated below in Figure 23.

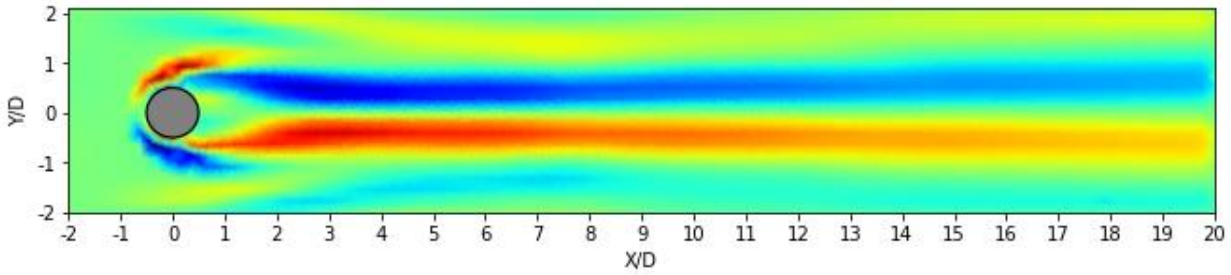

Figure 23. Plot of snapshot 18000 for Re = 70 using matplotlib interval IV, subcritical case generated using plot_snapshot.py

These simulations are postprocessed using computational methods such as autoregression, POD and DMD among others to garner further insights into the fluid flow dynamics.

3.1.2. Selection of POD Algorithm

Once the simulations were completed for the relevant Reynolds number cases POD modal analysis was performed. A key element in the methodology was determining which variant of the POD algorithm to use. Considerations of the applicable POD implementation included the speed of the algorithm, ease of use and adaptability. Initially the NumPy library as 'np' and the SVD module function was employed to perform the main part of the POD analysis whereby the variable `xT` represented a vorticity field consisting of 1651 rows for each nodal value within the mesh and either 15000 or 60000 columns representing the vorticity field values for each snapshot. The variable `qT` was assigned by reading the .xdmf files for the original vorticity field simulated by *st08_navier_stokes_cylinder.py*. The mean value of variable `qT` was subtracted from the vorticity field in line with POD theory as described in the Taira et al. overview paper [1]. This step allows the consideration of the perturbations rather than the entire field. The reduced SVD decomposition is used by setting `full_matrices` to false this decreases computation time by ignoring elements of the matrix that would have otherwise been multiplied by zero.

Code Snippet 1. NumPy SVD implementation

```
xT = qT - qT.mean(0)
v, s, uh = np.linalg.svd(xT, full_matrices = False)
```

Although the above methodology was effective in decomposing the vorticity field, it was limited in its speed. This led to the exploration of the dask Python library for parallel computing which was proposed to be faster for larger matrices due to the way it streams smaller decompositions of the matrix into blocks or

chunks which would assist in allowing parallelization for larger matrix multiplications [64]. In contrast to the NumPy SVD method, the dask SVD method also had the added benefit of being able to specify the rank reduction during the decomposition process which increased computational speed. Further to this dask has some ability to operate within a cloud computing environment for increased concurrent scheduling of processes. One of the other main drawbacks is that much of NumPy's linear algebra module has not been implemented in dask. This method did not prove native support for consideration of the mass matrix and there was difficulty in fully realizing the benefits of parallelization without serious development of the idea and related scripts.

Code Snippet 2. Dask compressed SVD implementation

```
v, s, uh = da.linalg.svd_compressed(xT, k = r)
```

The above implementations of SVD have not considered the mass matrix, which is required to determine the inner product. The incorporation of the mass matrix modifies the resultant matrices derived from SVD quantitatively. Without the mass matrix the energy of the modes would be representative of pseudo energy. For a mesh that is set with a uniform grid the mass matrix would be the identity matrix and would not be considered. However, since the mesh is not uniform for these simulations the mass matrix must be considered. The simulation code had to be modified to save the mass matrix once for each mesh.

Code Snippet 3. SciPy.sparse with the save_npz module to save the mass matrix for each simulation case in *st08_navier_stokes_cylinder.py*.

```
save_npz(
          Path(str(os.path.dirname(__file__)) + '\\' + 'SIM_XDMF\\' +
          os.path.splitext(str(os.path.basename(__file__)))[0] +
          "mass_Re_{}".format(str(Rey).replace('.', '-')) + ".npz"),
          skfem.asm(mass, basis["p"])

         )
```

The mass matrix was then employed for snapshot SVD [5]. This method allows for a different approach when including the mass matrix within the context of POD. Equation (13) represents the eigenvalue decomposition of the zero mean flow field with the inclusion of the mass matrix as an inner product. Equation (14) represents the eigen value problem for the covariance matrix of the zero mean flow field. The eigen values are effectively the square of the singular values whereby Ψ is a diagonal matrix of eigen values and Λ is a matrix of eigen vectors, whereby equation (15) highlights the relationship between the two eigen value decomposition methods.

$$X^T W X \Psi = \Psi \Lambda \tag{13}$$

$$X X^T W \Phi = \Phi \Lambda \tag{14}$$

$$\Phi = X \Psi \tag{15}$$

This method allows for incorporation of the mass matrix as well as the necessary include rank r truncation to select the number of modes. This is a large improvement of the functionality and output possible with the NumPy module and is much simpler to implement than in dask. Equation (13) is presented in its Python code form below in Code Snippet 4.

Code Snippet 4. SciPy.sparse.linalg library with eigsh module function which allows snapshot method of POD to be employed

```
xT = qT - qT.mean(0)
eiv, Psi = eigsh(xT @ W @ xT.T, k = r)  # Schmidt & Colonius (2020, eq. 17)
```

In line with existing literature, it was believed that a more adaptable approach would be appropriate for POD decomposition that was not limited by computation time. The direct method of POD as shown below in Code Snippet 5 does not square the singular vectors and singular values. It is more accurate than the eigen-decomposition as used in the snapshot method however it can be slower if there are more rows than columns i.e., $n > m$ [62].
Modred allows for the functionality to swap between the direct and snapshot methods for SVD with minimal effort. Modred is a separate library and is actively maintained and developed which is thus likely to be more suitable for future improvements especially with possibility of using Dask for larger matrices which can be collaborated on with Modred community.

Code Snippet 5. Modred Library SVD Implementation POD_MODRED_Kutz.py

```
xT = qT - qT.mean(0)
x  = xT.T
POD = mr.compute_POD_arrays_direct_method(x,
      inner_product_weights=W, mode_indices = range(r))
```

3.1.3. POD Validation Cylinder Wake

It was necessary to validate the current methodology against existing simulations to confirm that it is performing correctly and producing expected results both quantitatively and qualitatively. The fluid flow case for an unconfined cylinder wake data set named VORTALL was generated by Kutz et al. using an immersed boundary projection method (IBPM) fluid solver on a cylinder wake and sourced from an online repository [65] [13].This example data consisted of 150 snapshots with a 0.02 unit time step. Thus, there were five periods of vortex shedding sampled at a steady state of vortex shedding on the saturated limit cycle of the vorticity dynamics. Consequently, data was sampled at the same rate of approximately 30 snapshots per cycle in the saturated limit cycle region for data generated by the numerical simulation by the Python script *st08_navier_stokes_cylinder.py* by using subsampling. The resemblance between the topos or spatial modes of both simulations can be seen comparing **Error! Reference source not found.** and Figure 26 or Figure 25 and Figure 27. The Reynolds number for VORTALL data is equivalent to Re = 150.

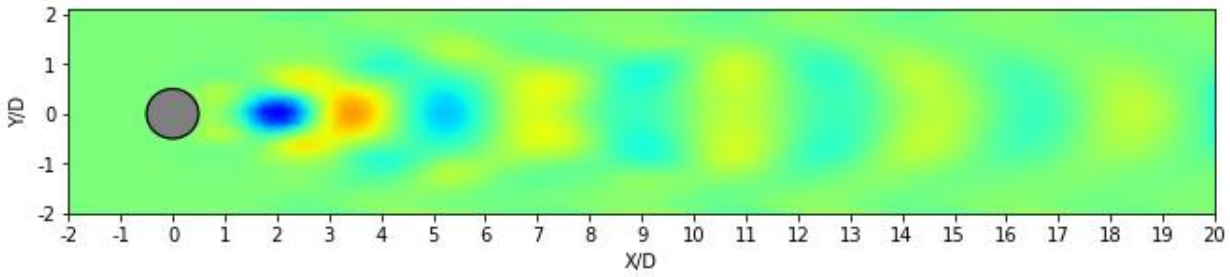

Figure 24. Spatial mode 1 from Re = 150, interval IV, simulation generated by st08_navier_stokes_cylinder.py and spatial mode generated by POD_MODRED_Kutz.py

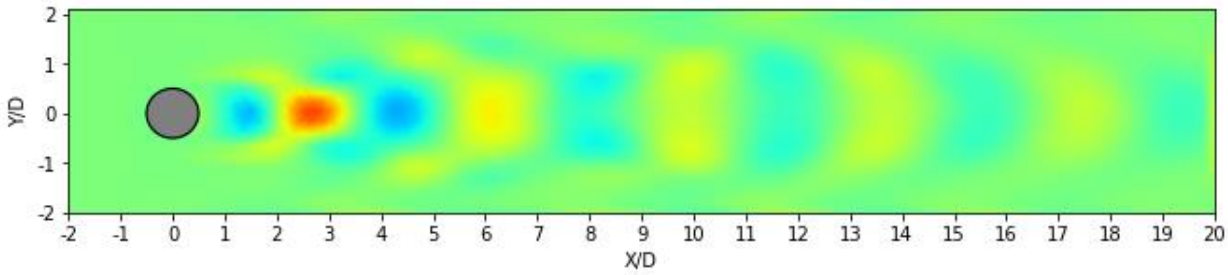

Figure 25. Spatial mode 2 from Re = 150, interval IV, st08-NS and spatial mode generated by POD_MODRED_Kutz.py

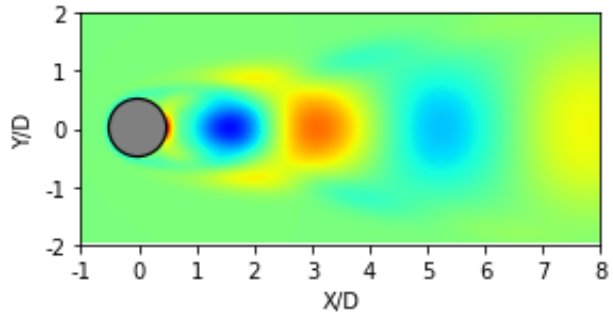

Figure 26. VORTALL Re=100 spatial mode 1, interval IV, st08-NS and spatial mode generated by POD_Modred_Test.py

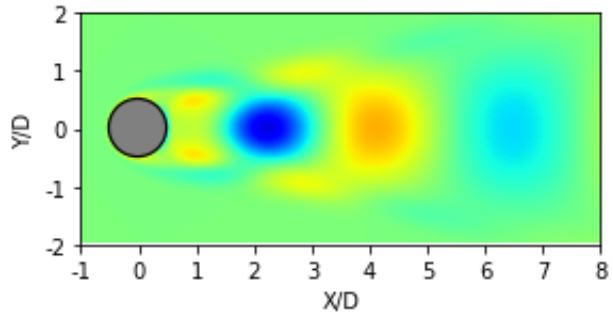

Figure 27. VORTALL Re=100 spatial mode 2, interval IV, and spatial mode generated by POD_Modred_Test.py

The colour alteration between red/orange and blue areas represents the alternating vortical structures which become smaller due to the generation of the harmonics of the unsteady laminar flow. The VORTALL data modes were generated using the Python library matplotlib. As the Navier-Stokes solvers used to generate both datasets are inherently different it is expected that there will be discrepancies between the non-dimensionalised spatial modes due to second order errors. First order parameter that may influence the results between the two datasets more significantly than second order errors may also include the blockage ratio and the inlet velocity profile, and geometry and boundary condition differences. For example, the scikit-fem solution has an inlet boundary condition for an even parabolic Poiseuille velocity profile which has an odd-linear vorticity profile, it has a mean velocity of 0.75 velocity units and a diameter of 0.1 length units. In contrast the VORTALL data consists of an inlet boundary condition with a uniform (plug) flow velocity of one velocity unit and a diameter of 0.5 length units [66]. The relationship between spatial mode 1 and spatial mode 2 for both datasets qualitatively and visually confirms a mode pairing, where there is a phase shift between each mode in the pairing for the respective datasets. These modes share an equal or near equal distribution of energy. Mode pairing in POD occurs for real valued (not complex valued modes)

which can represent a travelling structure, which is not possible with a single mode [1]. For example the energy in Figure 26 at (x,y) = (5.5,0) approximately corresponds to the phase shift of (x,y) = (6.5,0) in Figure 27. This is also true between **Error! Reference source not found.** and Figure 25 where (x,y) = (5.25,0) in spatial mode 1 corresponds to (x,y) = (4.25,0) in spatial mode 2. Note that the phase shift is not required to be positive but the absolute value is approximately the same from a qualitiative viewpoint.

Comparing quantitatively the modal energy distribution of both simulations for the same Reynolds number yields the results present in Figure 28. There is confirmed a high similarity between the modal energy distribution with a maximum discrepancy or difference of 1.74% and an average difference of ~ 0%.

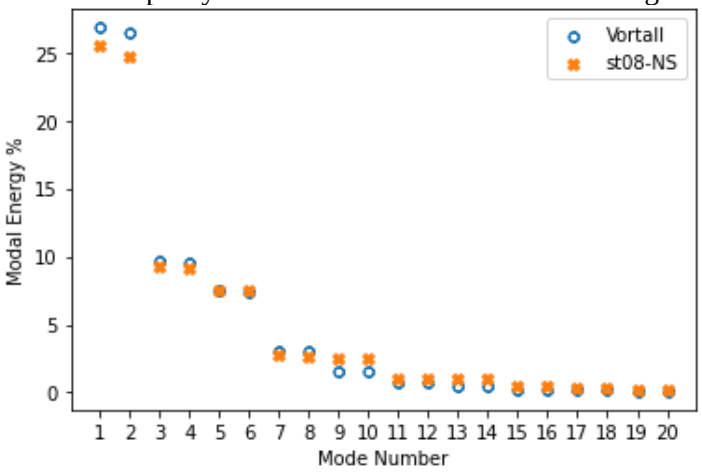

Figure 28. Modal energy distribution Re = 150

3.1.4. Selection of DMD Algorithm

The DMD algorithm was selected upon similar merits upon which the POD algorithm was selected not least of which is because POD or SVD is a major step within the DMD algorithm. It should be noted that unlike POD, the DMD algorithm does not take the zero mean of the vorticity field. Initially Kutz's algorithm was used and implemented within the Python environment as shown below in Code Snippet 6 following the notation present in Burtsev et al. [38]. Following similar code to that present in Code Snippet 1 `X1` and `X2` represent truncated matrices which then follows equations (4) to (8). The 0.001-time step was hard coded to extract the singular frequencies and growth rates using DMD of the st08-NS case which were assigned to the variables `f_j` and `g_j` respectively. The variable `b` represents an amplitude value for further reference consider Kutz's work on the topic [13]. Note down-sampling is the frequency of which the signal was taken e.g., if the down sampling frequency is 4 only every fourth snapshot is considered in chronological order of if the down sampling frequency is 10 only every tenth sample is considered and so on. When frequency values are generated, the negative conjugates of the conjugate pairs produce negative frequencies mirror the positive frequencies thus these modes can be disregarded using a condition referencing the index position of the mode and the corresponding frequency being less than zero, as seen in *DMD_MODRED_Kutz.py*, line 107-120. Much like how POD has mode pairs representing a phase shift the DMD modes have an imaginary and real part which functions as a pair whereby they can be processed separately when producing a spatial mode to see a phase shift, this can be seen when comparing Figure 29 and Figure 30.

Code Snippet 6. Initial DMD algorithm implemented in DMD_MODRED_Kutz.py

```
x = qT
X1 = x[:, :-1]                    #eq A1 (Burtsev et. al. 2021 )
X2 = x[:, 1:]                     #eq A2 (Burtsev et. al. 2021 )
'''
SVD and Rank-R Truncation to find first 'r' modes
'''
u, s, vh = np.linalg.svd(X1, full_matrices=False)
r = len(u[0])
Ur = u[:, :r]
Sr = s[:r]
Vr = vh.T[:, :r]

Atilde = Ur.T @ X2 @ Vr / Sr     #eq A4 (Burtsev et. al. 2021 )
D,W = np.linalg.eig(Atilde)      #eq A5 (Burtsev et. al. 2021 )
Phi =   X2 @ Vr / Sr @ W         #eq A6 (Burtsev et. al. 2021 )
x1 = x[:, 0]
b = np.linalg.lstsq(Phi, x1, rcond = None)[0]

dT = 0.001
f_j = np.angle(D) / (2 * np.pi * dT * downsampling)
#eq A9 (Burtsev et. al. 2021 )
g_j = np.log(np.abs(D)) / (dT * downsampling)

#eq A10 (Burtsev et. al. 2021 )
```

Upon review of the methods used to create the DMD algorithm in Code Snippet 6 it was decided that abstracting this methodology and improving the computational speed would be of significant benefit for future works and simplify the code greatly without impacting functionality or the ability to modify settings. The Modred library fulfilled this role of abstracting the algorithm while receiving all the desired inputs and returning all the desired outputs. Furthermore, Modred can include the mass matrix with native support which was a significant advantage over other possible implementations as seen in Code Snippet 7. The values for amplitude, frequency and growth rate however must still be calculated using the above methodology

Code Snippet 7. DMD Modred algorithm implemented in DMD_MODRED_Kutz.py

```
DMD = mr.compute_DMD_arrays_direct_method(x,
      mode_indices = range(r), max_num_eigvals = r)
x1 = x[:, 0]
dT = 0.001
b = np.linalg.lstsq(DMD.exact_modes, x1, rcond = None)[0]
f_j = np.angle(DMD.eigvals) / (2 * np.pi * dT * downsampling)
g_j = np.log(np.abs(DMD.eigvals)) / (dT * downsampling)
```

3.1.5. DMD Validation Cylinder Wake

The DMD validation between the VORTALL dataset and st08-NS dataset were created using the DMD Modred method. The imaginary and real spatial modes for the st08-NS dataset are shown below in Figure 29 and Figure 30,respectively. Similarly, the imaginary and real spatial modes for the VORTALL data set are shown below in Figure 31 and Figure 32. Much like POD mode pairs, DMD spatial mode pairs exhibit a phase shift relationship which can be observed visually for example the energy at (2,0) in Figure 29 approximately translates to (2.5,0) in Figure 30. However, this relationship should not be used as a rigorous

method to quantify phase shift as is also true for POD. DMD spatial modes unlike POD spatial modes do not have a natural ordering so it is easier to identify the modes by their frequency rather than the mode number as unlike POD, DMD does not contain any similar value to a normalized enstrophy value which is contained within the truncated singular values. The fundamental frequency is identified by a frequency which is an integer multiple of the remaining frequencies. The frequency for st08-NS is 2.90 and for the VORTALL data 1.65 cycles per unit time, these values were also corroborated by a peak fit of the time dynamics to also calculate the frequency as seen in *DMD_MODRED_Kutz.py*, line 123-139. Whereby, frequency would be represented by the number of snapshots per cycle. For the Strouhal frequency these values are equivalent to 0.4125 and 0.44 respectively, for the VORTALL and st08-NS case where U is the mean velocity and L is the length of the diameter. This discrepancy between fundamental frequencies for different simulations is largely explained by differences in initial conditions and mesh settings discussed previously for each respective simulation.

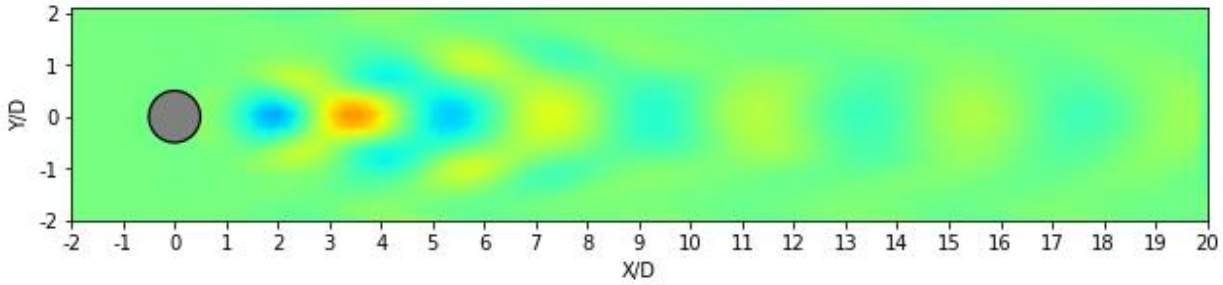

Figure 29.DMD imaginary mode st08-NS Re = 150, interval IV, st08-NS and spatial mode generated by DMD_Modred_Kutz.py

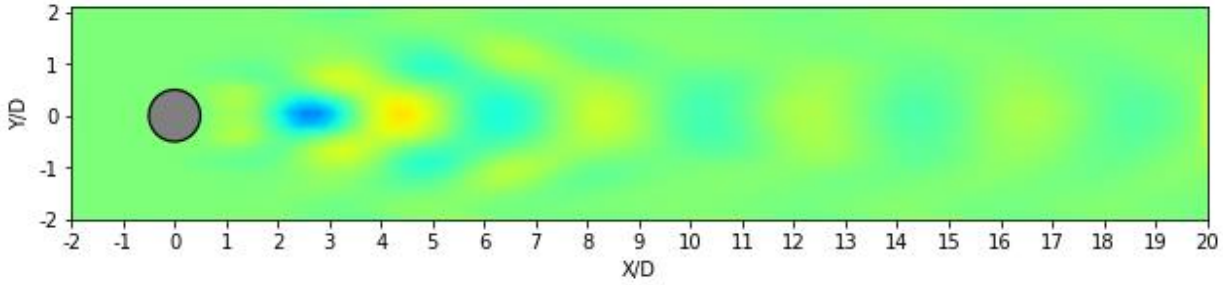

Figure 30.DMD real mode st08-NS Re = 150, interval IV, st08-NS and spatial mode generated by DMD_Modred_Kutz.py

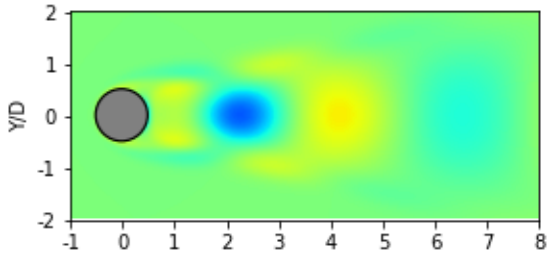

Figure 31. DMD imaginary mode VORTALL case, interval IV, and spatial mode generated by DMD_Test_Modred.py

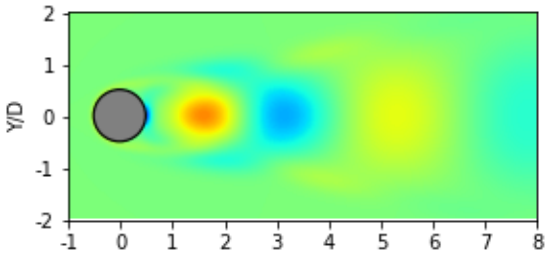

Figure 32.DMD real mode VORTALL case , interval IV, and spatial mode generated by DMD_Test_Modred.py

The near matching DMD eigen values for both the VORTALL and st08-NS simulation are exhibited below in Figure 33. The eigenvalues below $Im(\lambda) = 0$ are conjugate pairs of the eigen values above $Im(\lambda) = 0$ with the equivalent $Re(\lambda)$ value. The eigen value related to the fundamental frequency is denoted by the eigen value closest to the eigen value $Im(\lambda) = 0$ but where $Im(\lambda) > 0$. The eigen values being on the unit circle is indicative of a zero-growth rate this is exhibited as the snapshots taken are of a periodic limit cycle dynamics associated with the Von Kármán vortex Street of interval IV for the supercritical Reynolds number case. The eigen value similarity worsens around the cycle for high order modes due to sensitivity to noise. Thus, the methodology is validated as it is producing expected results in line with a similar simulation.

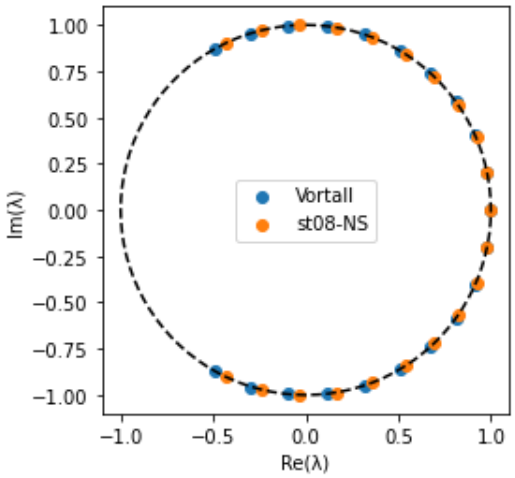

Figure 33. DMD eigen values of first 21 modes from st08-NS simulation and VORTALL dataset generated by DMD_Modred_Kutz.py

3.1.6. Autoregression of Cylinder Wake

Autoregression is a statistics-based model which can predict future values by using the values preceding it, it is useful in the context of time-series flow data to understand how various flows whether probes or some other time varying data changes. It is important to note that when using autoregression the variable which varies with time is analysed independently to other variables which may have an interdependent relationship with it. To consider these interdependent relationships HODMD or VAR are likely to be more appropriate but they are also more difficult to implement and computationally more costly.

An autoregressive model uses a linear combination of predictors which are past values of the variable being predicted. An autoregressive model with order p i.e. AR(p) is represented below in equation (16), where

y_t is the predicted value c is the constant value trend used y_t are the lagged values, Φ_p are the predictor parameters or coefficients and ε_t is the error or noise associated with the dataset.

$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \cdots + \Phi_p y_{t-p} + \varepsilon_t \tag{16}$$

DMD is able to determine the growth rate and frequency for a DMD mode directly from the eigen values it is believed that in some cases the same information can be determined from Autoregressive modelling of the chronos mode components of POD modes for dominant modes for high order modes this generally shouldn't be applicable as incoherent noise generally appears as higher-order POD modes [1]. If higher order POD modes are of significance DMD may be a more suitable method of analysis for obtaining frequency and growth rate information. A significant question brought up by the weaknesses of POD described in the overview paper by Taira et al. claims "spatial POD modes generally contain a mix of frequencies". It is then important to ask upon which circumstances can POD modes contain singular frequencies. Autoregression is a useful tool for independently analyzing individual chronos modes to determine growth rates or frequencies but is unable to show the relationship between chronos modes. Autoregression has been implemented in Python as per Code Snippet 8. The autoregression method is implemented using statsmodels AutoReg module [67]. Although not appropriate in all cases, AR(2) is used to find the frequency and growth rate of the first chronos mode of the st08-NS case at various Reynolds numbers as this first mode is approximately sinusoidal.

Code Snippet 8. AR(2) root calculation and reconstruction AR(P)_Complex_Frequency_Spectra.py

```
n = end_n - start_n  #number of snapshots
chronos_mod = POD.proj_coeffs  #chronos mode
zeta = chronos_mod[chronosi]

time = [*range(start_n, end_n)]
dt = 0.001                          #time step
n_lags = 2                  #number of lags
downsampling = 2**0
p = n_lags            #use p = 2 to find dominant frequency of each mode
zeta = zeta[::downsampling]
step = np.arange(zeta.size) * downsampling
res =  AutoReg(endog = zeta, lags = p).fit() #AR(p) model
asymptote = res.params[0] / (1 - res.params[1:].sum())
print("asymptote:",asymptote)
mu = 1 / res.roots
print("Spectrum:", mu)
s = np.log(mu) / (dt * downsampling)
print("s:", s)
modes = np.vander(mu, zeta.size, True).T
modes /= np.linalg.norm(modes, axis = 0)
coefficients = np.linalg.lstsq(modes, zeta - asymptote, rcond=None)[0]
fundamental = np.argmax(abs(coefficients))
print("frequencies",abs(s.imag) / (2 * np.pi))
reconstruction = asymptote + np.real_if_close(modes @ coefficients)
```

Note that the angular frequency in radians per unit time is represented by the following equations where angular frequency is in terms of radians per unit time and down sampling frequency is representative of the frequency unit times are considered. A unit time in the st08-NS case would be 1000 snapshots as dT is 0.001.

$$s = \frac{\log(\mu)}{dT \times downsampling} \tag{17}$$

$$Angular\ Frequency = \Im s \tag{18}$$

$$Cycles\ per\ unit\ time = \frac{\Im s}{2\Pi} \tag{19}$$

It is clear from Figure 34 that AR (2) is an appropriate method of modelling the first chronos mode (which is a multiplication of the singular values with the temporal modes), as it possible to create a near exact reconstruction of the original vorticity data with a reconstruction error less than 1%. This model approximates the frequency of 3.13 cycles per unit time for the st08-NS supercritical case in Interval IV as per equation (19) [68].

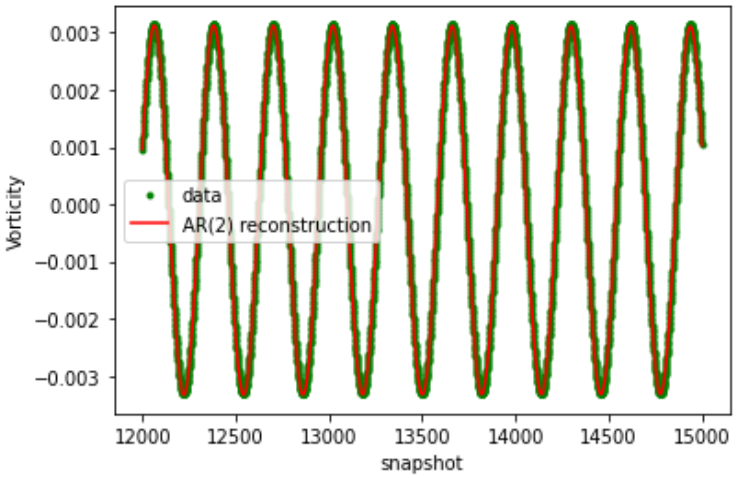

Figure 34. First chronos mode reconstruction using AR(2) with a constant trend model with coefficient matrix and Vander modes for Re = 150

3.1.7. Fast Fourier Transform of Cylinder Wake

Fast Fourier Transform (FFT) refers to a variant of the discrete Fourier Transform (DFT) algorithm whereby the number of computations require for N points is reduced from $2N^2$ to $2Nlog_2(N)$ computations [69]. The FFT algorithm like autoregressive models can extract frequency information from a signal however FFT provides a spectrum of frequencies where the peaks are the dominant frequencies whereas autoregressive models can present several frequencies depending on the number of lags. The FFT can estimate frequencies but only for a predetermined set of frequencies and periodic sampled data this means that it is appropriate for supercritical interval IV cases where the signal is periodic but not appropriate for other parts of the signal this is especially limiting as it is not able to capture the linear growth dynamics [32]. The FFT algorithm produced a frequency of 3.2 cycles per unit time for the st08-NS case at Re = 150 for the first chronos which correspond with the result found using autoregression for AR (2), as shown in Figure 35 by the stem plot peak.

Code Snippet 9. FFT implemented in CHRONOS_FFT.py

```
zeta = zeta - zeta.mean(0) #remove the zero frequency by demeaning
sr = 1000 #sampling rate -> 1/dT
FFT = fft(zeta) #FFT of signal zeta
FFT_FREQ = fftfreq(zeta.size, 1 / sr)
X_oneside = FFT
f_oneside = FFT_FREQ
```

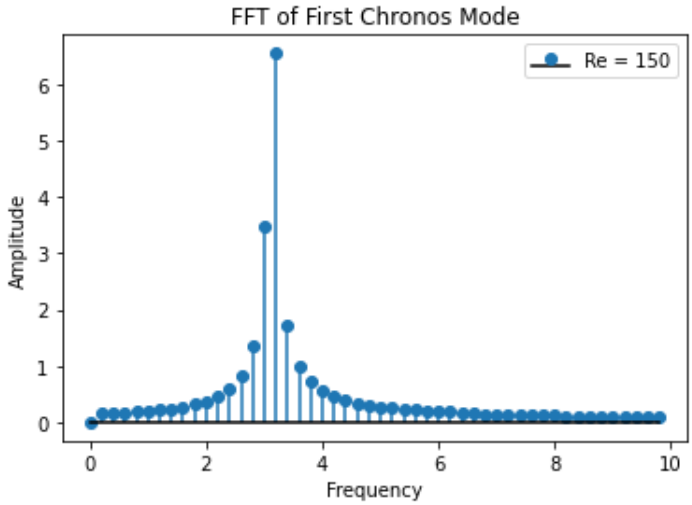

Figure 35. FFT of first chronos mode at Re = 150, considering interval IV, snapshot window 10000-15000

3.1.8. High Order DMD, Prony Analysis and Vector Autoregression of Cylinder Wake

Vector Autoregression (VAR) and Higher Order DMD (HODMD) also known as DMD-d (where d is the number of lags), share a large range of similarities and extend upon the central premise of DMD which uses a discrete linear Koopman operator, by considering an extended number of lags beyond the single lag value that DMD uses. That is HODMD replaces the Koopman assumption presented in equation (21) with the high order Koopman assumption in equation (22), where R is the Koopman matrix, d is the number of lags and m is the number of snapshots [4]. Equation (20) is derived from the temporal and singular value components that form a chronos which is derived in equation (3). Prony Analysis extends upon VAR to reconstruct the sinusoidal exponentials which is a major step in DMD to create a ROM.

$$\hat{X}_1^n = \Sigma V^* \tag{20}$$

$$\hat{X}_2^n \cong R\hat{X}_1^{n-1} \tag{21}$$

$$\hat{X}_{d+1}^n \cong R_1\hat{X}_1^{n-d} + R_2\hat{X}_2^{n-(d-1)} + \cdots + R_d\hat{X}_d^{n-1} \tag{22}$$

Similarly, VAR(p) extends upon autoregression by considering multiple variables and uses multiple lags like HODMD. Equation (23) provides a representation of the result of VAR(p) where Y is a matrix of n rows by m columns, where n is the number of variables, in this case n is the number of modes from the POD ROM considered and m is the number of snapshots considered with down sampling. Thus, each y_i element is a column vector of size n and p is the number of lags. In equation (25) c is a flattened matrix of intercepts or constants with zero mean for each time series of interest and A_i is a fixed $(n \times n)$ coefficient matrix for each lagged value and the term U is a matrix of e_i column vectors of size n, representing the

error or white noise terms. Thus, the lagged values are used to predict the future state contained in Y. For more information regarding the VAR(p) process see Lütkepohl's work on the subject [70]. In the implementation of VAR, U is assumed to be negligible furthermore it is unknown. In Code Snippet 11, c is $B[:,0]$ which is the second argument of np.linalg.solve which gives $uinf$.

$$Y = BZ + U \tag{23}$$

$$Y = [y_p\ y_{p+1} \dots y_T] \tag{24}$$

$$B = [c\ A_1\ A_2 \dots A_p] \tag{25}$$

$$Z = \begin{bmatrix} 1 & 1 & \dots & 1 \\ y_{p-1} & y_p & \dots & y_{T-1} \\ y_{p-2} & y_{p-1} & \cdots & y_{T-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_0 & y_1 & \cdots & y_{T-p} \end{bmatrix} \tag{26}$$

$$U = [e_p\ e_{p+1} \dots e_T] \tag{27}$$

Vector autoregression is then further extended by formulating a polynomial eigenvalue problem (PEP) of degree p which is converted to a regular eigen value problem of order $p \times r$, where r is the number of modes being considered, using the Frobenius form of the companion matrix. The Prony coefficients are then calculated using a least squares method. This allows for reconstruction of the signal as detailed in Code Snippet 11.

3.2. Ink-jet Printer Injection Zone

3.2.1. Ink-jet Printer Injection Zone Simulation

A pipeline for analyzing an ink-jet printer injection zone simulation was formulated with the assumption that the methods of analysis used on the cylinder flow example would naturally extend to this simulation for future use. OpenFOAM simulation files were provided with a '.VTK' file type for a single print density. The '.VTK' files are structured such that there exists one '.VTK' file for each timestep where the file would represent the cell-based values and mesh at that time step. It was thus necessary to parse the filenames of each of these files which displayed the respective time linked to that file. These files were opened individually in a loop using meshio [71] which was able to convert nodal values into a NumPy array, these values were then stacked into a matrix whereby the number of rows was representative of the number of nodal values and the number of columns was representative of the number of snapshots. This manipulation was conducted using the *vtk_import_test.py* script.

Following the formulation of the matrix various computational methods may be applied to the matrix to inform on the coherent features of the flow and create predictions. Only POD and autoregression has been applied within this report to this simulation due to the limited project time. In the future more cases at varying print densities should be considered for further analysis.

A brief and not at all extensive overview of the some of the main parameters use in the simulation of the inkjet print zone has been provided in Table 4.

Table 4. Ink-jet printer injection zone OpenFOAM simulation parameters

Parameters	Value
Retangle Grid for Fluid Space	3mm (H) x 60mm (L)
Start Time (ms)	370
End Time (ms)	400
Number of Points per snapshot(nodes)	30794
Total Snapshots	2832
Time Step	0.002
Write Interval	0.01
Print Density	0.085

A snapshot of the ink-jet printer injection zone simulation has been captured in Paraview as shown in Figure 36. The ink-jet stream impinges on the paper surface which forms a counter rotating vortex in the streamwise direction of the paper motion whereby the paper moves along the y axis. For more information on the topic see Aquino et al.'s work on the topic [43] [46].

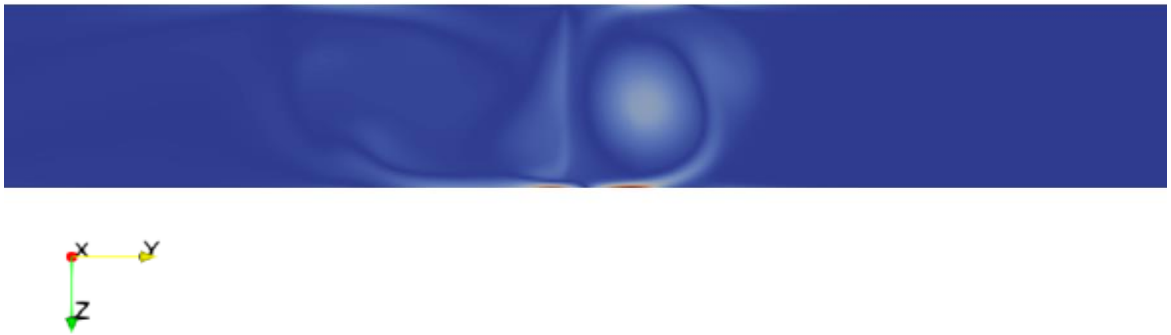

Figure 36. Impinging jet printer zone vorticity contours in Paraview

Chapter 4 – Results and Discussion

4.1. Confined Cylinder Wake Problem

The bifurcation in the context of the cylinder wake flow example represents the process under which the main flow transitions from the two vorticial structures lengthening with increasing Reynolds number and finally separating into two distinct vortical structures at the bifurcation point, as seen when comparing Figure 22 and Figure 23. A Hopf bifurcation is a critical point upon which the stability of the system changes and periodic solution arises in this context the Von Kármán Vortex Street structures. A vorticity probe can capture the change in dynamics with respect to the Reynolds number as the Reynolds number is increased through the bifurcation point whereby the sinusoidal exponential changes with respect to growth rate. These growth rate changes can be witnessed when probing the simulation at the same point for the same snapshot window for varying Reynolds Numbers. Since, the snapshot window is relatively large in Figure 37 it is hard to exactly distinguish the frequency impacts between Re = 70 and Re = 75. However, the apparent sinusoidal exponential increases in growth rate as Reynolds number increases, the asymptotic rate at which the steady state is approach is more rapid for Re = 75 compared to Re = 70.

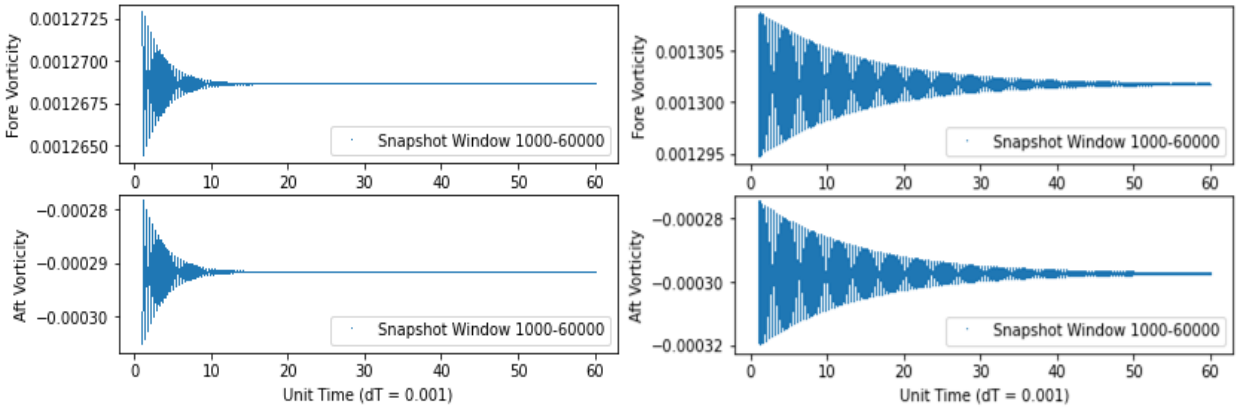

Figure 37. Fore and aft stagnation point vorticity probes for st08-NS simulation for subcritical cases, where Re = 70 (Left) and Re =75 (Right)

In Figure 38 the bifurcation point has been transitioned through as is indicated by the comparison between the first and last snapshot for the Re = 76 case and the Re = 77 case. As the eventual behavior changes from decreasing over time to increasing over time it can be said that the stability of the system has changed. A periodic solution is expected for Re =77 and a quasi-steady state solution is expected for Re = 76. Furthermore in Figure 39 the eventual periodic stable solution can be witnessed for the higher Reynolds numbers past the bifurcation point for Re = 78 and Re = 79. It is seen that the stable solution is reached at an earlier instant when increasing the Reynolds number too which provides further evidence on the relationship between growth rate increasing as Reynolds number increases.

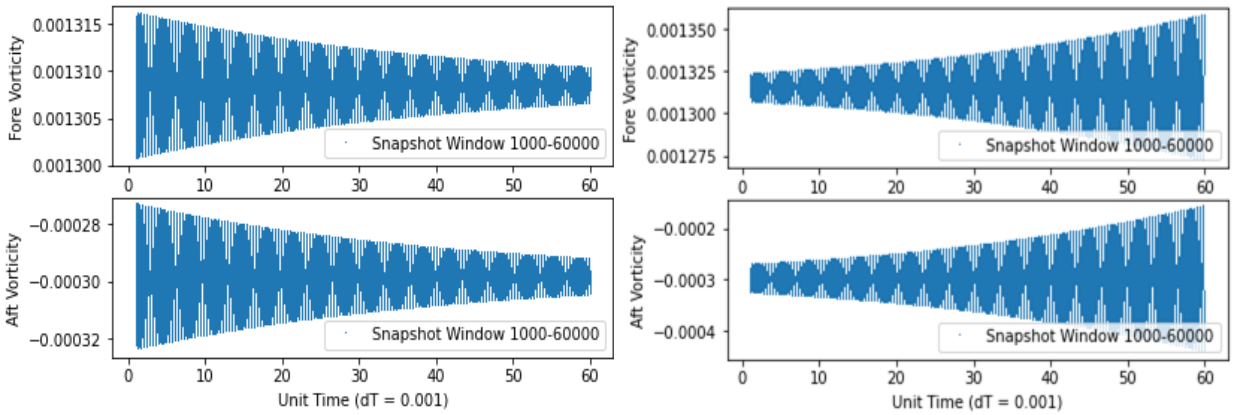

Figure 38 Fore and aft stagnation point vorticity probes for st08-NS simulation for bifurcation transition between subcritical and supercritical cases, where Re = 76 (Left) and Re =77 (Right)

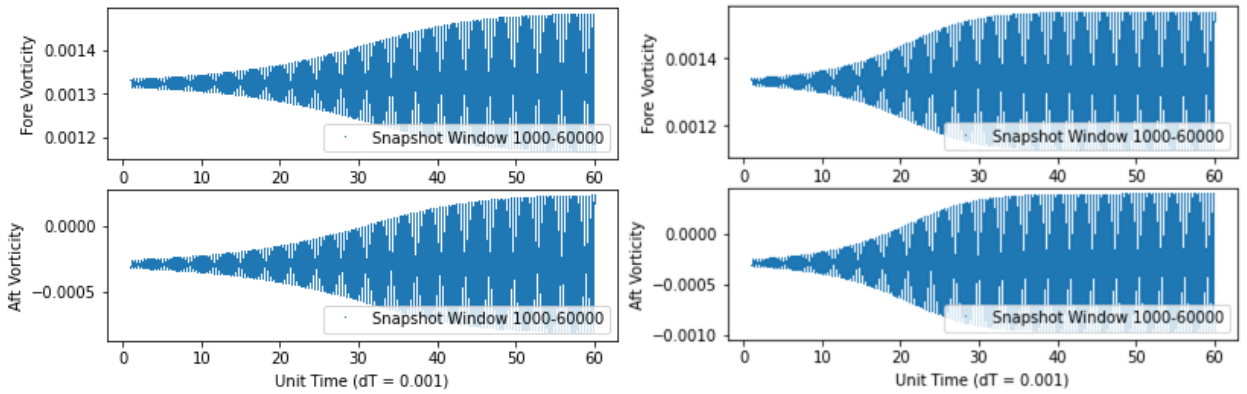

Figure 39. Fore and aft Stagnation point vorticity probes for st08-NS simulation for supercritical cases, where Re = 78 (Left) and Re =79 (Right)

Figure 40 and Figure 41 showcase the frequency and growth rate results obtained from using autoregression with two lags on the probes at the aft and fore stagnation point as well as autoregression on the chronos modes and finally the fundamental DMD mode for an interval I time window. The probe and chronos results with auto regression show great agreement in this domain with an average discrepancy less than 0.5%. However, when these results are compared to the DMD frequencies and growth rate linked to the fundamental frequency there are some significant outliers that are important to consider. One of the major drawbacks of DMD is it effectively equivalent to vector autoregression, but it uses one lags instead of two or more lags where the autoregression model in this case uses two lags. This makes it particularly sensitive to time dynamic changes. That is since interval I has been defined qualitatively and not rigorously mathematically DMD is sensitive to these interval changes, this is most apparent for the DMD values at Re = 80. Furthermore, it is quite difficult to apply a robust criterion to identify the fundamental frequency and corresponding growth rate. In this case from *INTERVAL_I_DMD.py*, lines 117-122 were used to separate negative frequencies from non-negative frequencies and lines 145-149 were used to find the fundamental frequency whereby this frequency would that which is closest to zero and not noise. It was particularly difficult to apply a robust criterion to define noise and this is one of the major shortfalls of using DMD which is apparent in Figure 40 and Figure 41. Autoregression of the chronos modes and the probes however promises an effective method of determining frequency and growth rate. In agreement with the information from the probes as Reynolds number increases both frequency and growth rate increase.

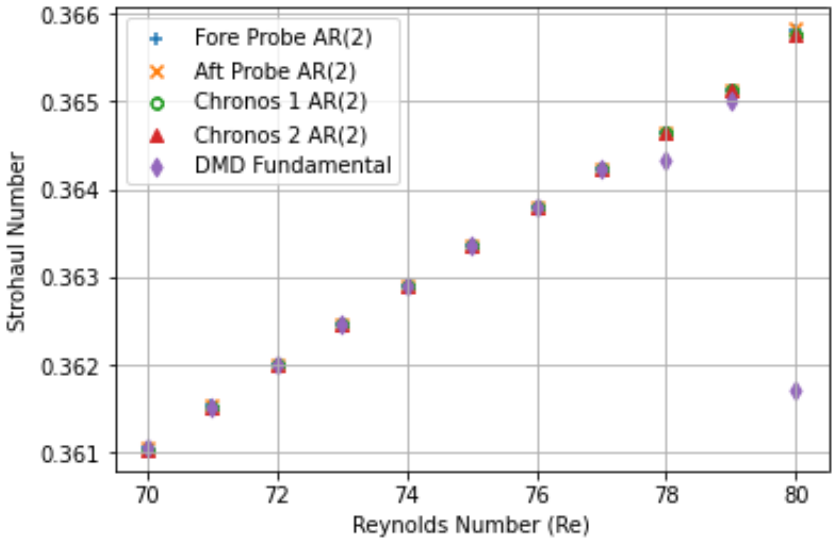

Figure 40. Frequency analysis of st08-NS fundamental frequency in interval I, snapshot 3000-15000, using varying modal techniques

The linear relationship between Reynolds number and growth rate allows the implementation of a linear interpolation function to calculate the approximate critical Reynolds number whereby the growth rate for interval I is zero. The critical Reynolds number was found to be Re = 76.44 which falls outside the expected range of values between 67.5 and 73.5. Given the simulation settings there are a multitude of factors that this 4% discrepancy may be attributed to including but not limited to the blockage ratio, mesh settings, boundary conditions and initial flow profile. The identified bifurcation point with the given percentage error however broadly supports the argument that there are a multitude of methods which may be used to identify the bifurcation point correctly or approximately for a given flow.

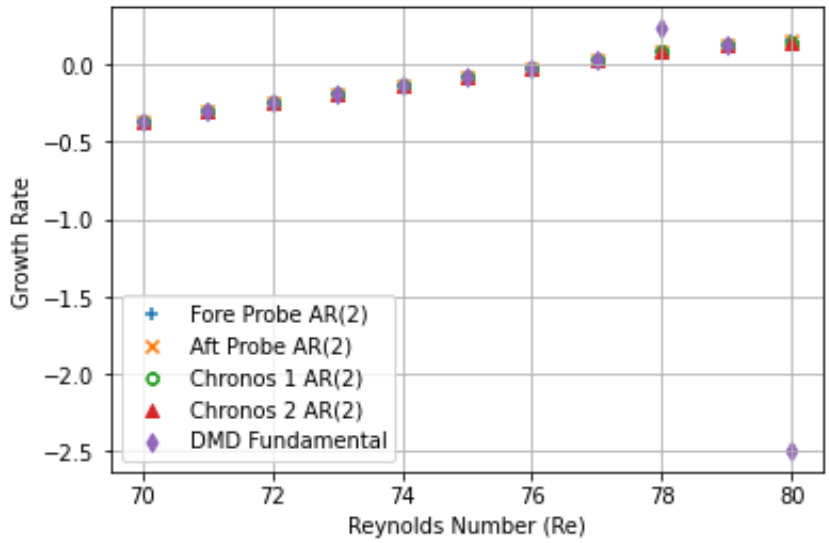

Figure 41.Growth rate analysis of st08-NS fundamental frequency in interval I, snapshot 3000-15000, using varying modal techniques

Figure 43 and Figure 44 showcase the results of another method which can approximately identify the bifurcation point. Driven by Aquino's bifurcation diagram in Figure 16 and Lumley's work on POD it is possible to identify the critical Reynolds number by identifying the amplitude connected with the eventual amplitude for a flow simulation at varying Reynolds numbers [3] [46]. This trend is mirrored in Figure 42

where a square root relationship is seen between the amplitude and Reynolds number, thus if the amplitude was squared there would be a linear relationship.

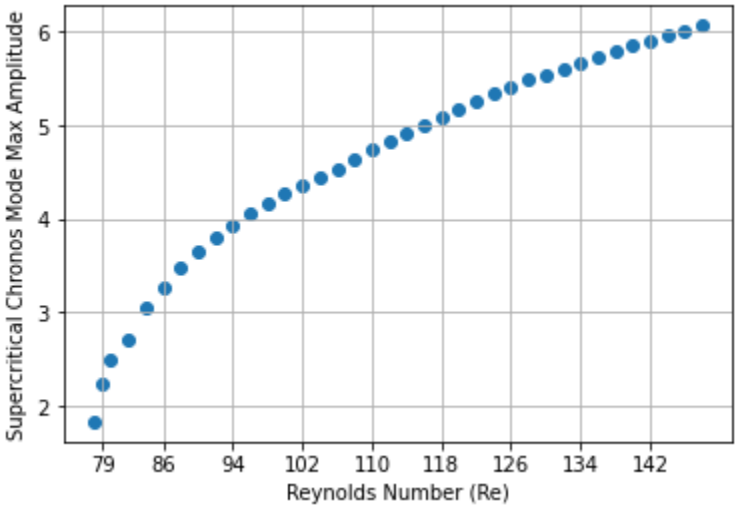

Figure 42. Amplitude for first chronos mode in interval IV, st08-NS supercritical cases

The POD singular values which are the square root of the eigen values are indicative of the eventual amplitude for each respective simulation for a subcritical case it is expected that this eventual amplitude will be zero whereas for the supercritical case it is expected that the eventual amplitude will be non-zero and increasing with higher Reynolds number. Figure 44 clearly shows this relationship whereby a linear relationship can be found for the supercritical cases when the entire snapshot window is considered. This linear relationship however does not hold when a partial time window is considered as shown in Figure 43. This can be a significant issue for computational resources as it suggests there is to some extent a requirement to reach the saturated amplitude state for a simulation to effectively use POD to interpolate the results to find the bifurcation point. The importance of this issue for applications will largely depend on the accuracy of the bifurcation point required as well as the complexity of the simulation.

One might consider using a combination of the autoregression with two lags and POD eigen value techniques to approximate the frequency and growth to assist in finding the bifurcation point. However, this will require the simulation to solve for the eventual state and to be in some form of steady or quasi steady state which may or may not be periodic. Other techniques such as VAR or HODMD with multiple lags show promise in being able to overcome this limitation of POD by being able to reconstruct and predict future dynamics, successfully and robustly developing a methodology which can do this should be the key next step in identifying the bifurcation point for increasingly more complex geometry. This may further assist in overcoming one of the central issues of these methodologies which require an understanding of the underlying physics before implementation.

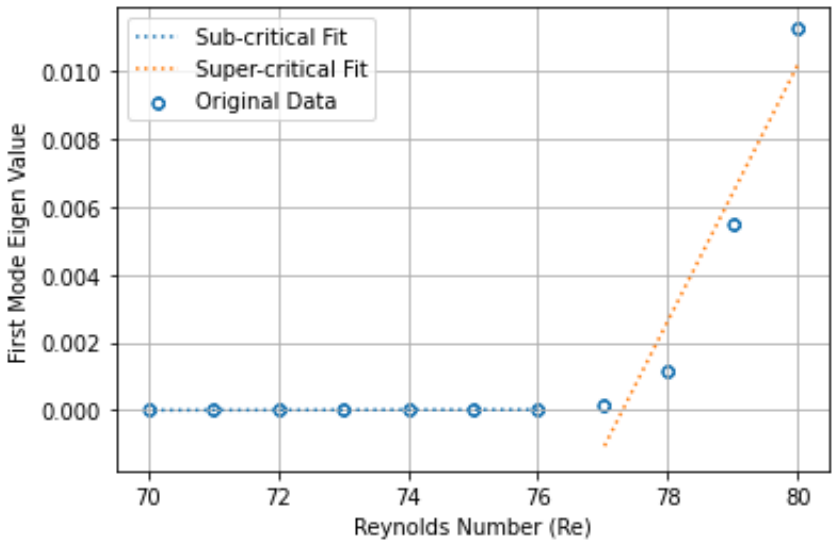

Figure 43. Squared singular values for Reynolds numbers 70-80 showcasing the bifurcation transition and bifurcation point Snapshot window 3000-30000

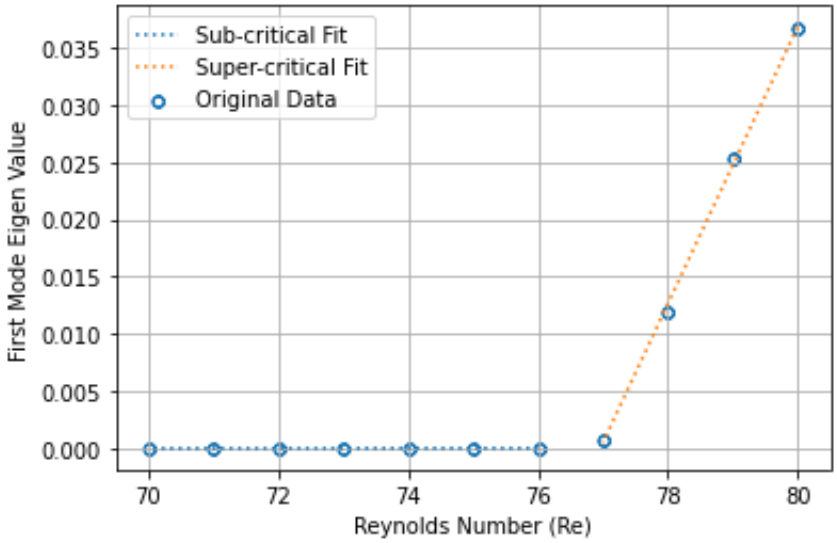

Figure 44. Squared singular values for Reynolds numbers 70-80 showcasing the bifurcation transition and bifurcation point snapshot window 3000-60000

Despite DMD's sensitivity to noise and its singular lag value used for decomposition and reconstruction, it is an effective means of accurately calculating the frequency and growth rate in Interval IV with where a saturated periodic amplitude is present. This is confirmed by Figure 45 and Figure 46 where the average discrepancy is less than 0.5% for frequency values and the zero-growth rate is correctly identified using the DMD method. It is expected in interval IV that the saturated amplitude is periodic but does not increase or decrease as time evolves. Significantly no other method besides DMD were able to identify the zero-growth rate for the fundamental frequency. This calls into question the superiority of autoregression for the interval I diagram but falls apart for interval IV analysis. Furthermore, the autoregression model is considering each chronos independently of one another which is not necessarily correct as it is expected modes might have some interdependence.

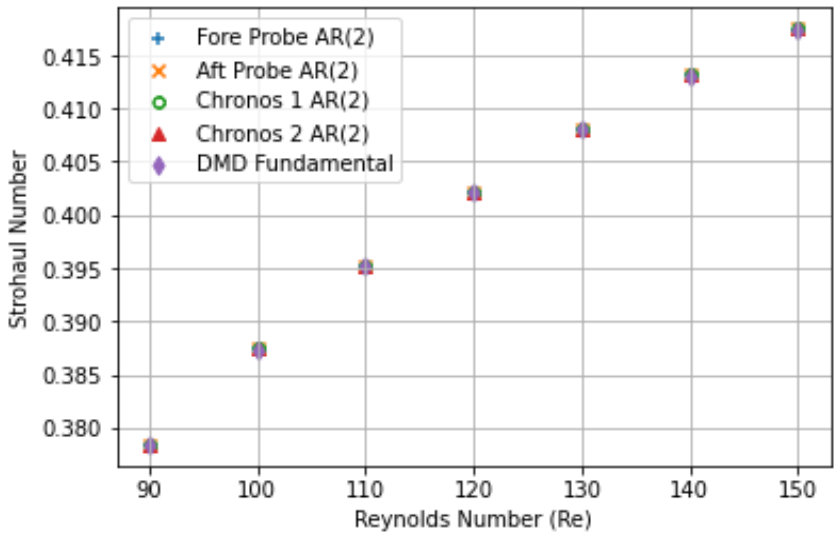

Figure 45. Frequency analysis of st08-NS fundamental frequency in interval IV, snapshot 10000-15000, using varying modal techniques

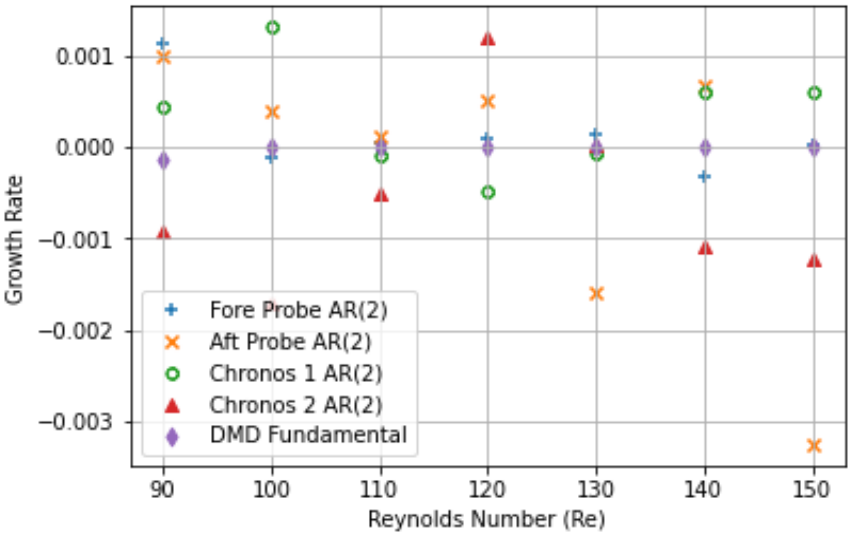

Figure 46. Frequency analysis of st08-NS fundamental frequency in interval IV, snapshot 10000-15000, using varying modal techniques

The need for Prony analysis arose from attempts to fit AR (2) to a chronos mode to identify its frequency and rate of growth. A significant issue in our analysis of individual chronos modes is the implicit assumption that chronos modes are scalar signals and not vector signal whereby each mode has some relation to other modes. It is important to consider not only scalar signals but vector too, since if POD does not separate nonlinearly generated harmonics, it might be necessary to auto-regress multiple chronos modes together, which techniques such as HODMD and vector Prony analysis are able to achieve [72]. This is clear when we considered a periodic case of the cylinder wake flow example. For a snapshot window in interval IV as seen in Figure 47. The probe data is seemingly periodic and may be able to be represented by a sine or cosine equation. However further analysis upon this snapshot windows is required to understand the high order dynamics that may be occurring.

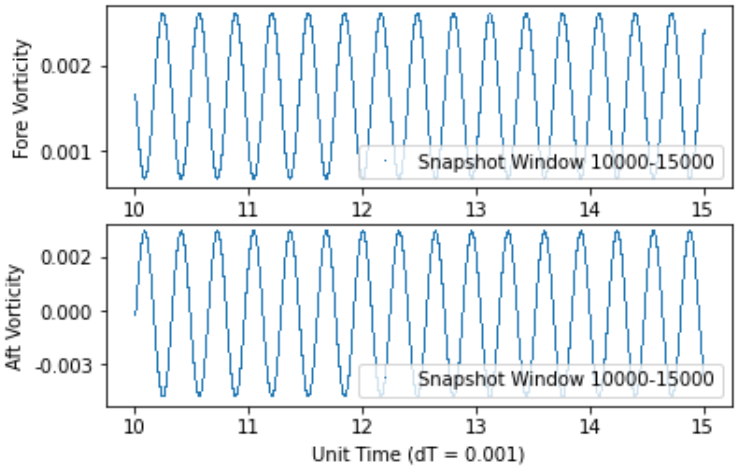

Figure 47. Aft and fore stagnation point probes for Re = 150

A FFT was applied to this periodic case of the cylinder wake flow for two probes where a fundamental frequency of 3.2 was determined as shown in Figure 48 which matches the same fundamental frequency found for the first chronos mode using AR (2). This however does not identify the high order dynamics that are present and their associated energies. Further it is important to take into consideration that probes do not necessarily represent the full dynamics that are occurring over the whole mesh but rather are samples of dynamics from two points.

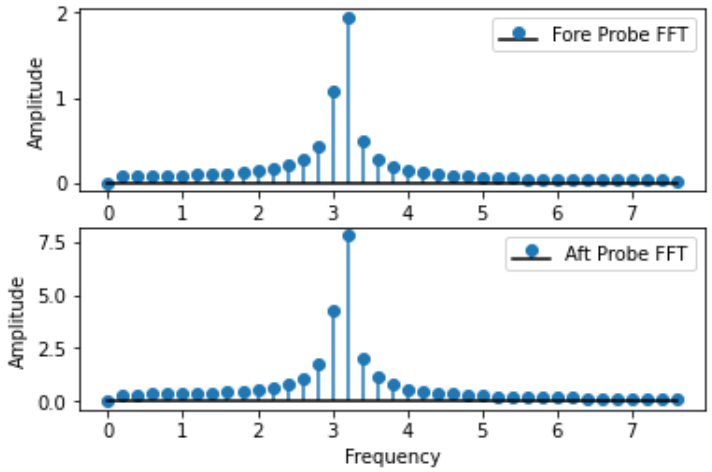

Figure 48. FFT of aft and fore probes for Re = 150, st08-NS

Prony analysis is a shown to be an effective tool in reconstructing the chronos modes as seen in Figure 51, whereby the ‘x’ markers represent the VAR (2) reconstruction. The orange and blue plots as seen in Figure 51 are of the first and second chronos in each chronos pair. This reconstruction performance is in direct contrast to the autoregression model, AR(2) ‘x’ marker reconstruction plot seen in Figure 49 which was unable to reconstruct the high-order POD chronos modes present in chronos pair 3 and chronos pair 4 i.e., mode 5,6,7,8. These high order modes collectively represent approximately 20% of the enstrophy present in this flow simulation as evidenced by the modal energy distribution in Figure 50. It is expected for even more complex flow problems the modal energy distribution will be distributed even less towards the first four modes thus autoregression is likely not an appropriate tool for many flow problems although it has shown great effectiveness for the cylinder wake problem and identifying the bifurcation point.

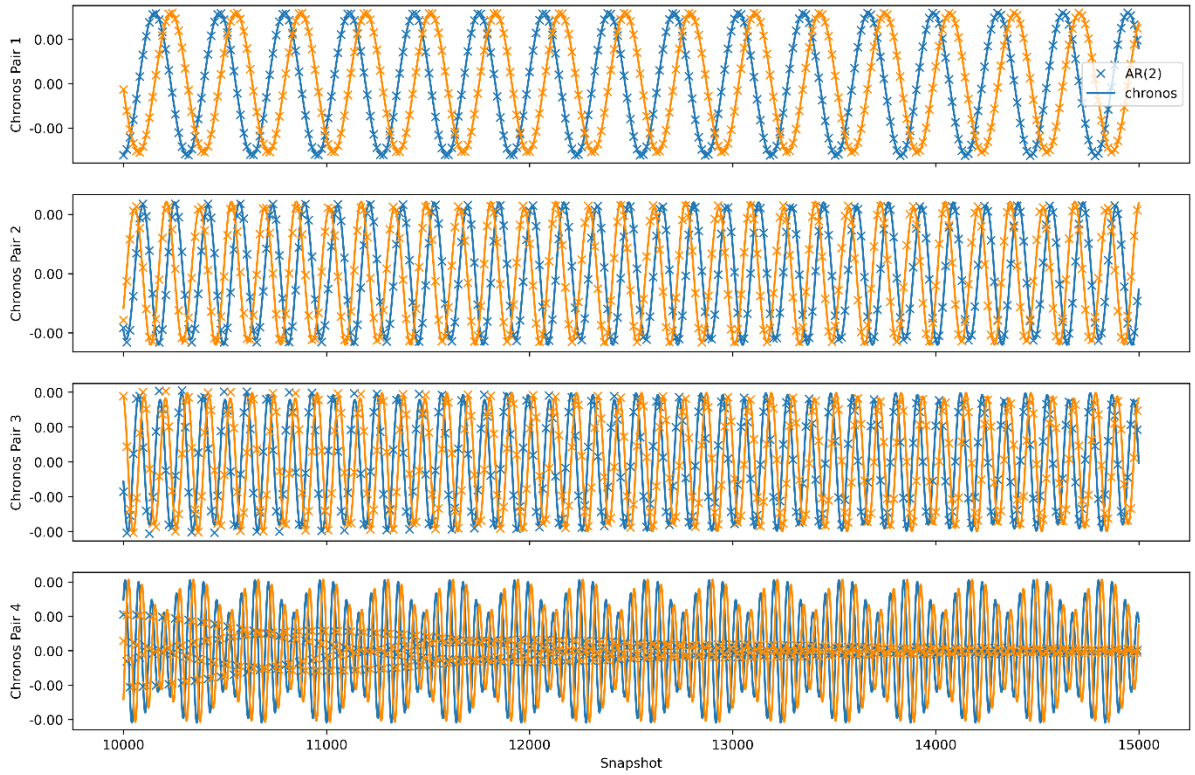

Figure 49. Chronos mode plot for first 8 Modes and AR (2) reconstruction for Re = 150, st08-NS, interval IV, down-sampling = 16 produced using AR_2 Reconstruction.py

An algorithm combining the concepts from VAR, Prony analysis and HODMD was used to create Figure 51 and Figure 53. In Figure 51 we can see from the reconstruction markers that the high-order modes are able to be extracted and reconstructed thus showing the higher harmonics, which are shown in Figure 52 as frequency peaks in the FFT analysis. It is also interesting to note the frequency relationships between certain modes for example chronos pair 2 seems to show a small relationship with chronos pair 3 where the frequency is approximately 9. Furthermore, an approximately diagonal relationship between the peaks and their stems in Figure 52 is presented which is further evidence of the modes representing higher harmonics from the fundamental shown in the first chronos pair as there is an integer multiple (linear) relationship between harmonics. In Figure 52 the blue plot represent the first mode in the chronos pair and the orange represents the second mode in the pair, it can be seen in the FFT plot that the frequencies are identical.

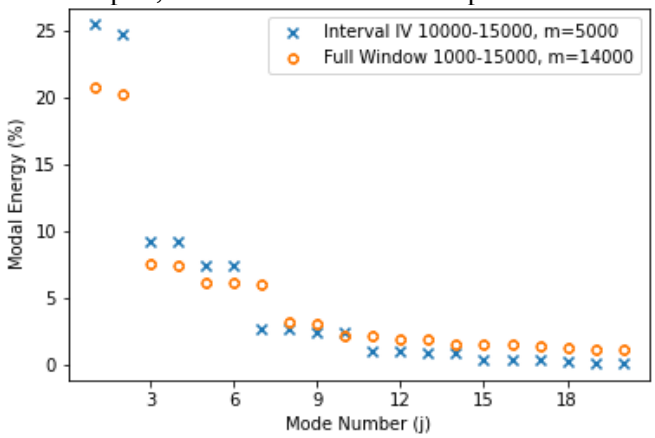

Figure 50. POD modal energy distribution for Re = 150, from singular values

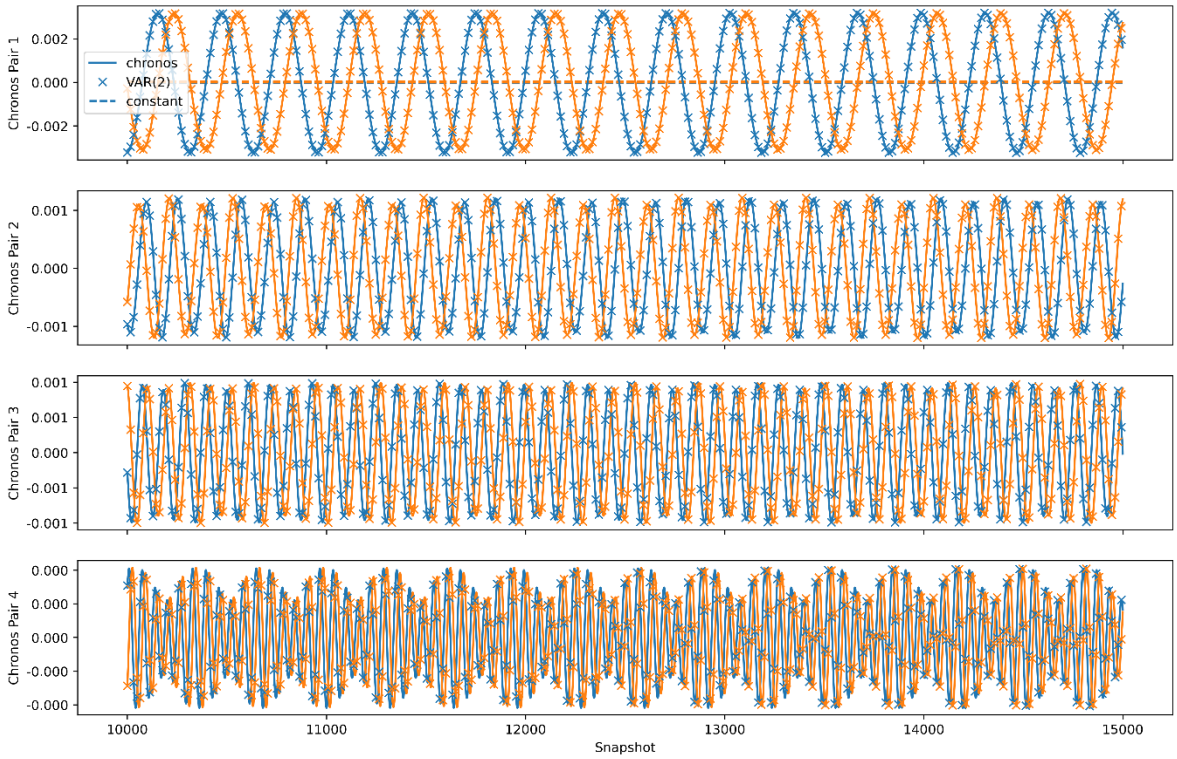

Figure 51. Chronos mode plot for first 8 Modes and VAR (2) reonstruction for Re = 150, st08-NS, interval IV, down-sampling = 16 produced using VAR_vorticity_prony_analysis.py

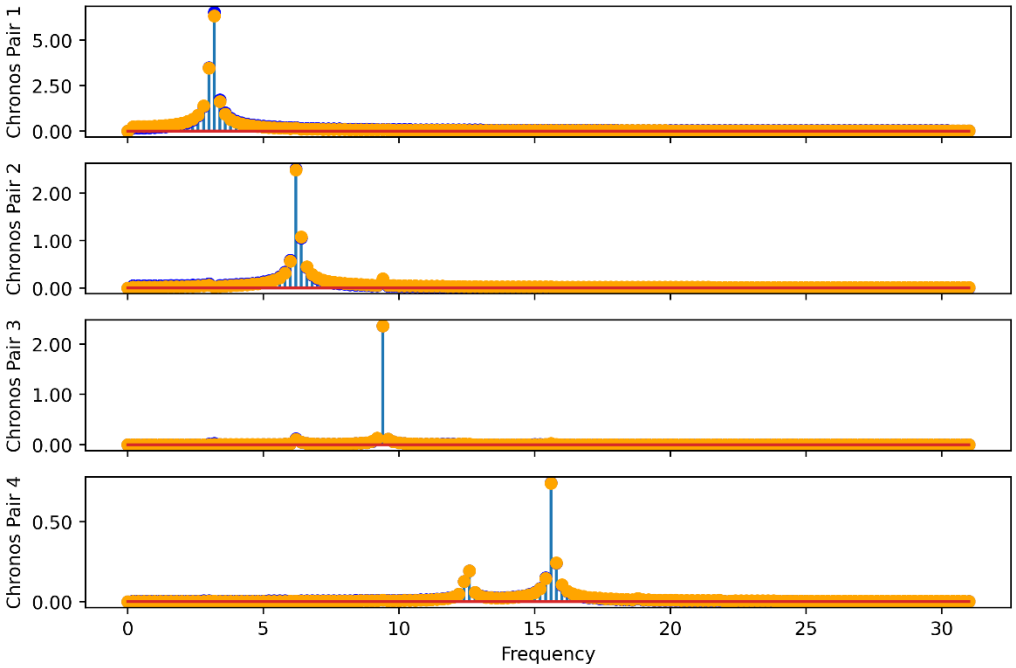

Figure 52. FFT Analysis of first 8 modes for Re = 150, st08-NS, interval IV

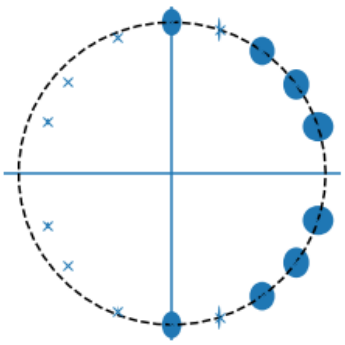

Figure 53. Periodic 2-D flow over a cylinder, Prony roots μ (r = 8, p= 2, down-sampling = 16), horizontal real axis and imaginary vertical axis with unit circle produced using VAR_vorticity_prony_analysis.py

Figure 53 is much like the DMD eigen value plot and is indicative of the stability of the system with respect to the time window and Reynolds number selected. The 'x' markers on this figure are representative of the roots and the size of the ellipses drawn on top of these markers are representative of the corresponding coefficients to the roots and their real and imaginary part weightings. Roots where the 'x' marker can be seen are largely irrelevant as their corresponding ellipse coefficient weighting is very small. As the roots come in complex conjugates it simpler to consider roots and their corresponding coefficients in the first quadrant as the conjugate pairs have mirror dynamics. These roots all lie on the unit circle, which is indicative of a stable system [13], this is the case as the periodic time window was selected for a supercritical case. Furthermore, the coefficients are logarithmically sized.

4.2. Ink-jet Printer Injection Zone Preliminary Results

Using methodology detailed in section 3 POD analysis was performed on the ink-jet printer injection zone simulation. AR(2) was used on the ink-jet printer injection zone simulation as most of the modal energy was distributed towards the first two modes as seen in Figure 55 thus the interdependence of modes was somewhat of less concern, however VAR(2) should be utilised in future works to confirm to what extent does the distribution of modal energy impact the usefulness of AR(2) versus VAR(2) reconstruction. Figure and Figure 55 indicate that the modal energy distribution lies largely with the first two chronos modes which form a chronos pair whereby the energy of each mode in this pair is nearly equivalent. This means that techniques such as autoregression with two lags can prove effective as a means of extracting information such as growth rate and frequency of the modes as high order more complex modes may be able to be ignored. The value of autoregression is highlighted especially in Figure 54 whereby the autoregression technique can perform a near perfect reconstruction with less than 0.1% error which can be likely attributed to floating point number issues. As over 95% of the energy is contained within the first four modes thus the other high order modes can be largely ignored.

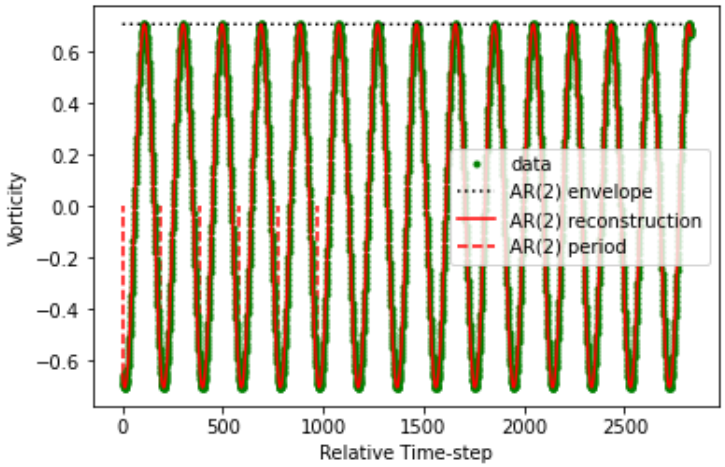

Figure 54. First chronos mode where print density = 0.085 for ink-jet print zone

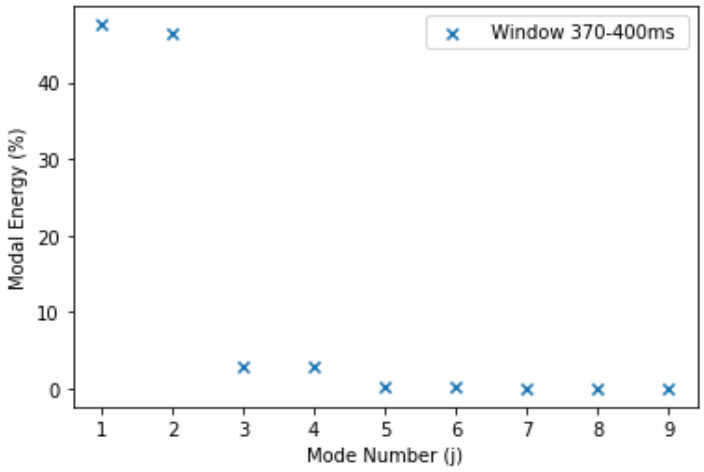

Figure 55. Modal energy distribution at print density = 0.085 for ink-jet print zone simulation

Spatial modes for the first four POD modes are presented in Figure 56 and Figure 57. It can be seen for all the spatial modes that a vortex exists whereby if we consider the centre of the vortex at approximately (Y/PPS, Z/PPS) = (0.2, -0.4) for the Y and Z coordinate respectively a phase shift occurs between modes, where the modes rotate in their respective chronos pairs. This is quite difficult to discern as the shapes seemingly coalesce and disperse irregularly unlike in the cylinder wake problem where mode shapes are somewhat more unidirectional. This phase shift is more apparent in the second pair of modes in Figure 57. It is important to note that in both the cylinder wake and ink-jet print zone simulation flow simulation the modal energy pairs are not exactly equal. It is known from Taira et. al. that POD modes must be paired to show a travelling structure, but they do not have to be paired in general thus the pairing allows a phase shift to be shown. In DMD phase shifts are shown between the real and imaginary components of a mode which in essence is the same as POD mode pairing not be confused with the conjugate pairs [1]. This does not account for the modal energies change although SPOD is suggested as an algorithm to further explore the phase shift. At this stage it is believed that since a true wave requires a domain that is translationally invariant in the direction of travel of the wave and both simulations are not perfectly translationally invariant with respect to the wave direction, that this could account for the discrepancies regarding some pairs in terms of modal energy.

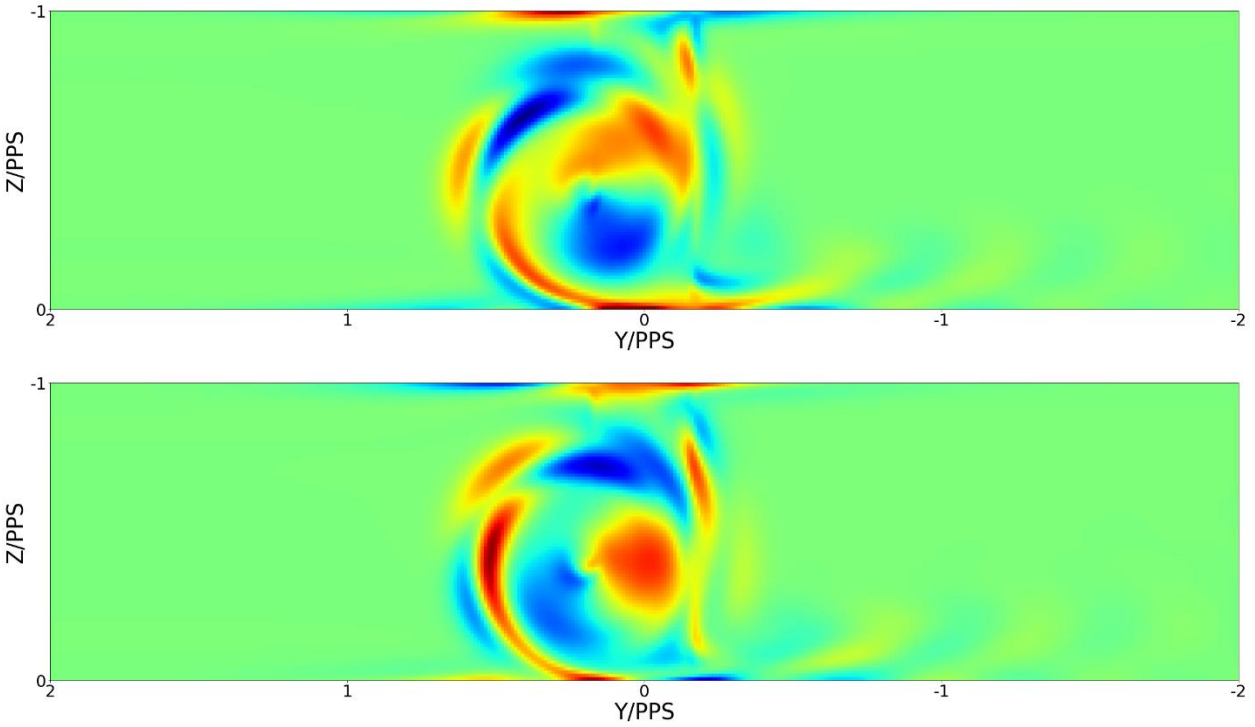

Figure 56. Spatial modes 1-2 (from top to bottom) at print density = 0.085 for ink-jet print zone simulation

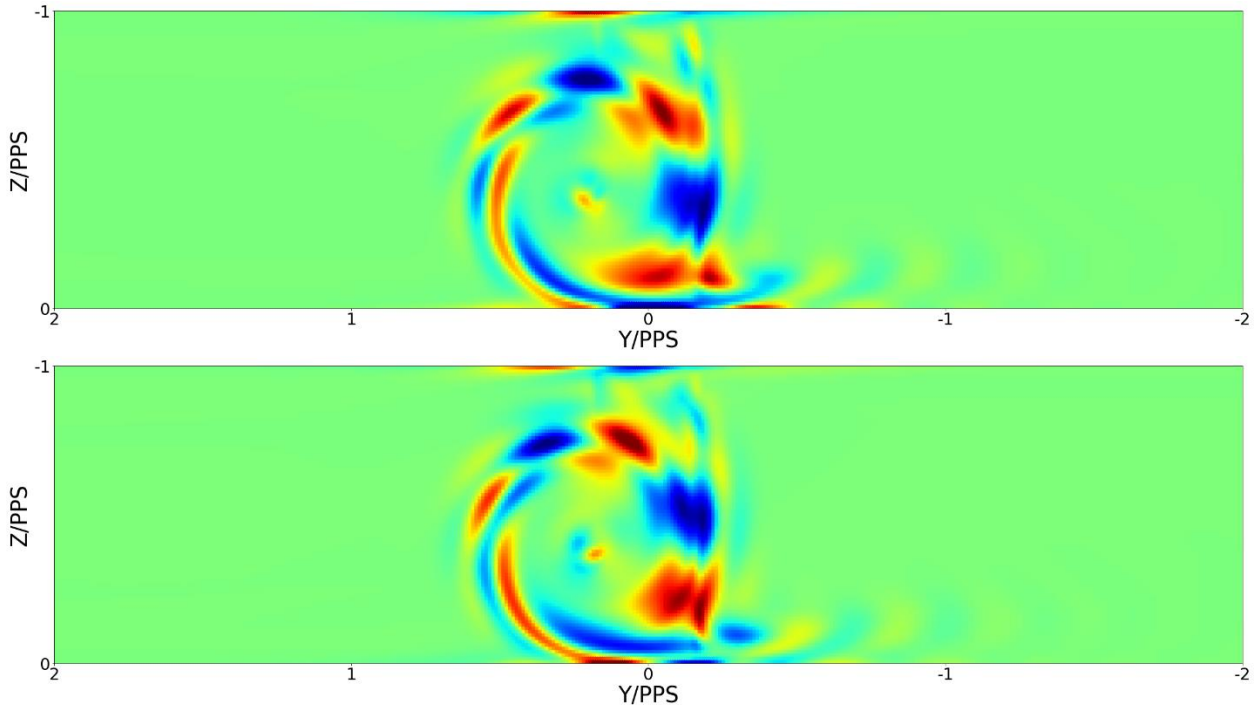

Figure 57 . Spatial Modes 3-4 (from top to bottom) at print density = 0.085 for ink-jet print zone simulation

Chapter 5- Future Works and Current Work Limitations

This study of computational methods for the analysis of fluid flow structures considers both the confined cylinder wake problem and the ink-jet print zone simulation. A variety of algorithms are proposed which can elucidate flow characteristics such as frequency, growth rate, bifurcation point, mode shape and some less important features. The implementation of a ROM that is able to be used for prediction, has somewhat been achieved with the reconstruction efforts shown with regard to VAR (2) and AR (2) reconstruction. POD is also easily able to be used for reconstruction whereby the significant modes with respect to modal energy are multiplied together with the chronos mode and spatial mode components. Albeit VAR (2) and AR(2) were used for reconstruction for chronos modes with the further ability for prediction however this prediction capability was not implemented as only time windows within the scope of the simulations were considered in their respective reconstruction expressions.

One of the most useful applications that VAR could achieve would be to determine the decay point or saturation point for subcritical or supercritical cases, respectively. This would allow for in effect a ROM as the whole simulation would not be required to run for the entire time window and the relationship between snapshot windows and their respective intervals would likely be easier to identify. This would significantly reduce necessary computation time as running simulations has accounted for a large part of the allocated projected resources specifically time in the early stages, this was especially problematic when simulations needed to be run again due to the noise created by the initial transients or when simulations were simply not run long enough.

Another avenue for in general improved computational time and scalability with respect to larger meshes and more complex flow problems would be the incorporation of parallel processing which utilizes the extra threads and cores present in modern CPUs. The dask Python library presented great promise within this area as it able to process significantly large matrices with the SVD method much quicker than library such as NumPy. Dask has further advantages as the community is building cloud computing capabilities whereby networked CPUs could be utilised for even large problems, in contrast the work here was limited to a local computer.

In general, the entire workflow with all related Python scripts was not optimized or designed with parallel computing in mind but rather each script was a proof of concept of mathematical algorithms working and producing some flow data result. This report generalizes three main methods of finding the bifurcation point for a given flow problem which has been studied before: POD, DMD and autoregression. It has become apparent though that the growth rate transition between negative and positive values at the zero-value growth rate evolves monotonically with respect to Reynolds number increasing, this linear relationship presents the opportunity to consider standard search related algorithms such as binary searches as a framework to find the bifurcation point. A significant hurdle would be setting the two limits points whereby one point should be a supercritical case and the other a subcritical case. This is largely a generalization of the work Fortin et. al. completed in finding the Hopf bifurcation point using a “guess and check method” in this case the guess and check would be automated and only be informed a priori with physical knowledge for notifying the algorithm of a supercritical case and subcritical case [51]. This generalized method is presented in Figure 58 as a binary search algorithm. It is hoped that this generalization for finding the bifurcation point might prove useful for simulation such as the ink-jet printer zone simulation and agree with existing literature [46].

Machine learning also has a variety of uses for prediction models for more complex flow whereby SVD normally is a precursor to a prediction model as it forms part of the unsupervised learning stage. This was largely beyond the scope of the project as machine learning models require many computational resources with the latest graphics cards to run them, normally in networked systems.

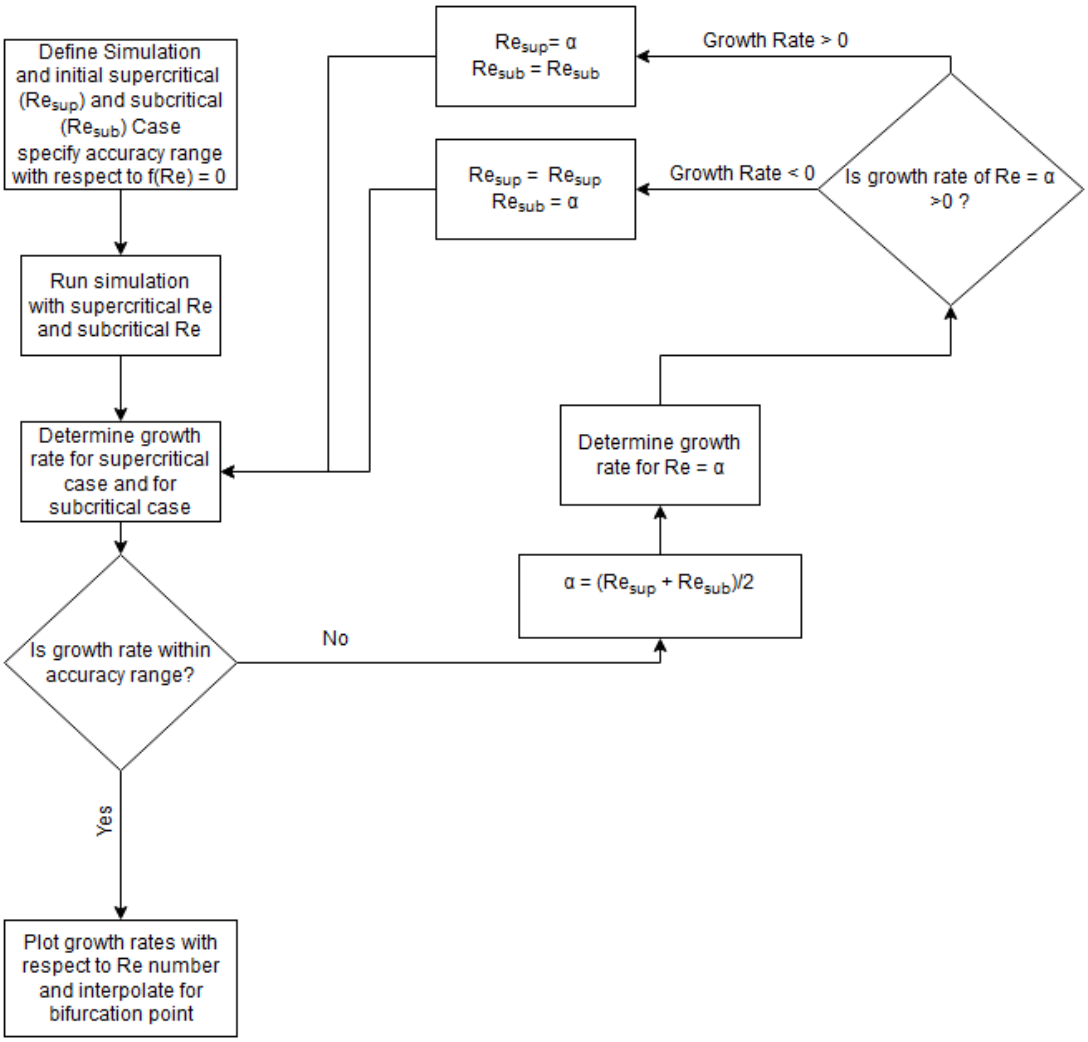

Figure 58. Generalized Binary Search Hopf Bifurcation

One of the central issues with this study is the approximation of the intervals for specific time windows used and that their features are not mathematically well defined. This is problematic for techniques such as DMD that are sensitive to these changes that occur over different intervals. From the reconstruction results obtained using VAR it is believed that techniques such as HODMD that use more lags to represent the time series data may be able to use a less strict interval or ignore interval entirely and be able to consider the entire time window. This however must be confirmed in another study as HODMD was a late addition to this project and was not researched in detail.

SPOD was also an algorithm that was not utilised for this project this was because when it was first tested it was discovered that due to the way windowing functions are implemented in SPOD a significant number of time cycles are necessary to provide useful frequency information otherwise SPOD suffers the sample problems as FFT with relation to the number of bins correlating to the number and accuracy of the frequencies able to be resolved. Such a simulation with enough time cycles would be very computationally costly.

In future works it may be considered that the steady state solution may be more appropriate to subtract rather than the vorticity field mean as this may be easier to analyse using statistical techniques as the mean of the periodic saturated oscillation in a supercritical Hopf bifurcation differs from the steady equilibrium by an amount that increases with the excess of the Reynolds number over the critical value.

Chapter 6 – Conclusions

This report details three major ways of identifying the bifurcation point through consideration of amplitudes, POD singular values and growth rates obtained from chronos modes using autoregressive techniques, as well as POD analysis of the singular values and DMD growth rates. An unfortunate finding of this work is the limitations of DMD for being able to reliably determine the bifurcation point. DMD was found to be very sensitive both to noise and the time window selected.

Late in the project, HODMD was identified to hopefully alleviate some of those issues present when using DMD. The full HODMD algorithm was not implemented due to time constraints however a VAR model with Prony Root reconstruction was and it was found that high order modes could be reconstructed which contained multiple frequencies, this was not possible with an autoregression model and further elucidated the interdependent relationship of modes whereas in contrast the autoregression models considered the chronos components separately.

As most of the modal energy for both the cylinder wake simulation and ink-jet print zone simulation was found within the first four modes and these modes were dynamically largely sinusoidal simple, autoregression models were found to be largely appropriate for reconstruction and identifying growth rate and frequency values. Furthermore, this energy distribution also allowed for the identification of the fundamental frequency for various Reynolds number. These fundamental frequencies were consequently confirmed by the identification of other integer multiple harmonic frequencies as well as the utilization of other frequency count methodologies such as FFT and find peaks from the SciPy Python library.

A generalized binary search algorithm to find the bifurcation point was presented which largely considered the chronos components of the simulation related to the chronos modes to recursively relate growth rates to Reynolds numbers. It is hoped that further works with the ink-jet printer zone simulation with varying print densities with the use of this framework will agree with previous work identifying the bifurcation point and confirm the algorithm's usefulness.

Of further value is developing an understanding of HODMD and implementing it both in relation to the cylinder wake example and inkjet print zone. This involves understanding how to appropriately select the number of lags for a given simulation allowing accurate prediction of future states. As well as obtain the standard frequency and growth rate information that DMD provides. It will be important to determine whether HODMD is noise sensitive as well or identify other significant parameters that may impact on its effectiveness.

HODMD may still need to be used in conjunction with POD to assist understanding as the orthogonal modes that arise out of the SVD are ordered helping identify important modes that can be used in a ROM. Other criteria may be available for exclusive use of POD in some flow problems, but this will need to be further explored in future research.

References

[1] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. Mckeon, O. T. Schmidt, S. Gordeyev, V. Theofilis and L. S. Ukeiley, "Modal analysis of fluid flows: an overview," *AIAA Journal*, vol. 55, no. 12, pp. 4013-4041, 2017.

[2] J. L. Lumley, The structure of inhomogeneous turbulent flows,atmospheric turbulence and radio wave propagation, Moscow: A. M. Yaglom and V. I. Tatarsky, 1967.

[3] P. Holmes, J. L. Lumley, G. Berkooz and C. W. Rowley, Turbulence, coherent structures, dynamical systems and symmetry, Cambridge: Cambridge University Press, 2012.

[4] S. L. Clainche, "An Introduction to Some Methods for Soft Computing in Fluid Dynamics," in *14th International Conference on Soft Computing Models in Industrial and Environmental Applications* , Seville Spain, 2019.

[5] L. Sirovich, "Turbulence and the dynamics of coherent structures parts I-III," *Quarterly of Applied Mathematics*, vol. XLV, no. 3, pp. 561-582, 1987.

[6] Mathworks, "svd," 2020. [Online]. Available: https://au.mathworks.com/help/matlab/ref/double.svd.html. [Accessed 18 November 2020].

[7] The SciPy community, "numpy.linalg.svd," 29 June 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html#numpy.linalg.svd. [Accessed 19 November 2020].

[8] J. Weiss, "A tutorial on the proper orthogonal decomposition," 15 June 2019. [Online]. Available: https://doi.org/10.2514/6.2019-3333. [Accessed 12 November 2020].

[9] J. S. Peter Schmid, "Dynamic mode decomposition of numerical and experimental data," *Bulletin of the American Physical Society*, vol. 61, p. MR.7, 2008.

[10] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5-28, 2010.

[11] W. E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem.," *Quarterly of applied mathematics*, vol. 9, no. 1, pp. 17-29, 1951.

[12] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton and J. N. Kutz, "On dynamic mode decomposition:theory and applications," *Journal of computational dynamics*, vol. 1, no. 2, pp. 391-421, 2014.

[13] J. N. Kutz, S. . L. Brunton, B. W. Brunton and J. L. Proctor, Dynamic mode decomposition data-driven modelling of complex systems, Siam, 2016.

[14] S. L. Brunton, B. . R. Noack and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual Review of Fluid Mechanics*, pp. 1-31, 2020.

[15] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi and G. Rozza, "POD-Galerkin reduced order methods for CFD using finite volume discretisation: vortex shedding around a circular cylinder," *Communications in Applied and Industrial Mathematics*, vol. 8, no. 1, p. 210–236, 2017.

[16] U. Fernandez-Gamiz, M. Gomez-Mármol and T. Chacón-Rebollo, "Computational modeling of gurney flaps and microtabs by POD method," *Energies*, vol. 11, no. 8, p. 2091, 2018.

[17] C. Xia, H. Yuan, Q. Li and Z. Yang, "POD analysis of the wake behind a circular cylinder coated with porous media," *Journal of visualization*, vol. 21, no. 6, pp. 965-985, 2018.

[18] K. Klausmann and B. Ruck, "Drag reduction of circular cylinders by porous coating on the leeward side," *Journal of fluid mechanics*, vol. 813, pp. 382-411, 2017.

[19] L.-H. Feng, J.-J. Wang and C. Pan, "Proper orthogonal decomposition analysis of vortex dynamics of a circular cylinder under synthetic jet control," *Physics of Fluids*, vol. 23, p. 014106, 2011.

[20] G. Riches, R. Martinuzzi and C. Morton, "Proper orthogonal decomposition analysis of a circular cylinder undergoing vortex-induced vibrations," *Physics of Fluids,* vol. 30, p. 105103, 2018.

[21] J. Wang, L. Tang, L. Zhao, G. Hu, R. Song and K. Xu, "Equivalent circuit representation of a vortex-induced vibration-based energy harvester using a semi-empirical lumped parameter approach," *International Journal of Energy Research,* vol. 44, no. 6, pp. 4516-4528, 2020.

[22] W.-L. Chen, G.-B. Chen, F. Xu, Y.-w. Huang, D.-L. Gao and H. Li, "Suppression of vortex-induced vibration of a circular cylinder by a passive-jet flow control," *Journal of Wind Engineering and Industrial Aerodynamics,* vol. 199, p. 104119, 2020.

[23] A. I. Bosioc, R. Susan-Resiga, S. Muntean and C. Tanasa, "Unsteady pressure analysis of a swirling flow with vortex rope and axial water injection in a discharge cone," *American Society of Mechanical Engineers,* vol. 134, no. 8, p. 81104, 2012.

[24] D. Stefan, P. Rudolf, S. Muntean and R. Susan-Resiga, "Proper orthogonal decomposition of self-induced instabilities in decelerated swirling flows and their mitigation through axial water injection," *Journal of Fluids Engineering,* vol. 139, no. 8, p. 081101, 2017.

[25] S. Maurel, J. Borée and J. L. Lumley, "Extended proper orthogonal decomposition: application to jet/vortex interaction," *Flow, Turbulence and Combustion,* vol. 67, no. 2, p. 125–136, 2001.

[26] P. Wang, H. Ma and Y. Liu, "Proper orthogonal decomposition and extended proper orthogonal decomposition analysis of pressure fluctuations and vortex structures inside a steam turbine control valve," *Journal of Engineering for Gas Turbines and Power,* vol. 141, no. 4, p. 041035, 2019.

[27] R. E. Ecke, "From 2D to 3D in fluid turbulence: unexpected critical transitions," *Journal of Fluid Mechanics,* vol. 828, pp. 1-4, 2017.

[28] J. Kou and W. Zhang, "An improved criterion to select dominant modes from dynamic mode decomposition," *European Journal of Mechanics B/Fluids,* vol. 62, no. 3, pp. 109-129, 2017.

[29] Y. Zheng, . H. Rinoshika, D. Zhang and A. Rino, "Analyses on flow structures behind a wavy square cylinder based on continuous wavelet transform and dynamic mode decomposition," *Ocean Engineering,* vol. 216, p. 108117, 2020.

[30] J. Wang, G. Huang, W. Lu and P. E. Sullivan, "Dynamic mode decomposition analysis of flow separation in a diffuser to inform flow control strategies," *Journal of Fluids Engineering,* vol. 142, pp. 021502-1, 2020.

[31] A. Goza and T. Colonius, "Modal decomposition of fluid–structure interaction with application to flag flapping," *Journal of Fluids and Structures,* vol. 81, pp. 728-737, 2018.

[32] S. Bagheri, "Koopman-mode decomposition of cylinder wake," *J. Fluid Mech.,* vol. 726, pp. 596-623, 2013.

[33] M. Liu, L. Tan and S. Cao, "Dynamic mode decomposition of cavitating flow around ALE 15 hydrofoil," *Renewable Energy,* vol. 139, pp. 214-227, 2019.

[34] A. J. Torregrosa, A. Broatch, J. García-Tíscar and J. Gomez-Soriano, "Modal decomposition of the unsteady flow field in compression-ignited combustion chambers," *Combustion and Flame,* vol. 188, pp. 469-482, 2018.

[35] T. W. Muld, G. Efraimsson and D. S. Henningson, "Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition," *Computers & Fluids,* vol. 57, pp. 87-97, 2012.

[36] K. Taira, M. S. Hemati, S. . L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson and C.-A. Yeh, "Modal analysis of fluid flows: applications and outlook," *AIAA Journal,* vol. 58, no. 3, pp. 998-1022, 2020.

[37] M. S. U. Khalid, J. Wang, I. Akhtar, H. Dong and M. Liu, "Modal decompositions of the kinematics of crevalle jack and the fluid-caudal fin interaction," *Bioinspiration & Biomimetics,* vol. In Press, no. https://doi.org/10.1088/1748-3190/abc294, 2020.

[38] A. Burtsev, W. He, V. Theofilis, M. Amitay and K. Taira, "Linear modal instabilities on swept finite-aspect ratio wings at low reynolds numbers and high angles of attack," American Physical Society, November 2020. [Online]. Available: https://meetings.aps.org/Meeting/DFD20/Session/H08.4. [Accessed 15 November 2020].

[39] H. Eivazi, H. Veisi, M. H. Naderi and V. Esfahanian, "Deep neural networks for nonlinear model order reduction of unsteady flows," *Physics of Fluids,* vol. 32, p. 105104, 2020.

[40] Y. Han and L. Tan, "Dynamic mode decomposition and reconstruction of tip leakage vortex in a mixed flow pump as turbine at pump mode," *Renewable Energy,* vol. 155, pp. 725-734, 2020.

[41] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science,* vol. 313, no. 5786, pp. 504-507, 2006.

[42] M. Raissi, A. Yazdani and G. . E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science,* vol. 367, pp. 1026-1030, 2020.

[43] A. F. V. d. A. Aquino, S. Mallinson, G. D. McBain, G. D. Horrocks, C. M. d. Silva1 and T. J. Barber, "Two-dimensional numerical simulation of inkjet print-zone flows," in *22nd Australasian Fluid Mechanics Conference AFMC2020* , Brisbane, 2020.

[44] S. Mallinson, G. McBain, G. Horrocks, A. North, A. O'Mahony, P. Reichl, S. Myers, J. Miller, B. Powell, J. Hess, D. Secker and P. Palma, "Suppressing tiger stripes:Taming flow oscillations to improve print quality," in *20th Australasian Fluid Mechanics Conference*, Perth, 2016.

[45] G. D. McBain, S. G. Mallinson, G. D. Horrocks, A. J. North, A. P. O'Mahony, P. J. Reichl, S. J. Myers, J. J. Miller, B. Powell, J. Hess, D. R. Secker and P. C. Palma, "Suppressing tiger stripes: taming flow oscillations to improve print quality," in *20th Australasian Fluid Mechanics Conference*, Perth, Australia, 5-8 December 2016.

[46] A. F. V. d. A. Aquino, S. G. Mallinson, G. D. McBain, G. D. Horrocks, C. M. d. Silva and T. J. Barber, "Investigation of the airflow instability in an inkjet print-zone using two-dimensional numerical analyses," School of Mechanical & Manufacturing Engineering University of New South Wales, Sydney NSW 2052, Australia and Memjet, Macquarie Park NSW 2113, Australia, Sydney, 2021.

[47] MIT, "Hopf Bifurcations.," 2004. [Online]. Available: https://ocw.mit.edu/courses/mathematics/18-385j-nonlinear-dynamics-and-chaos-fall-2004/lecture-notes/hopfbif.pdf. [Accessed 20 July 2021].

[48] F. Gallaire, E. Boujo, V. Mantic-Lugo, C. Arratia, B. Thiria and P. Meliga, "Pushing amplitude equations far from threshold: application to the supercritical Hopf bifurcation in the cylinder wake," *Fluid Dynamics Research,* vol. 48, no. 6, p. 061401, 2016.

[49] M. Brøns, B. Jakobsen, K. Niss and L. K. V. Anders V. Bisgaard, "Streamline topology in the near wake of a circular cylinder at moderate Reynolds numbers," *Journal Fluid Mechanical,* vol. 584, pp. 23-43, 2007.

[50] Technische Universität Dortmund, "DFG flow around cylinder benchmark 2D-2, time-periodic case Re=100," [Online]. Available: http://www.featflow.de/en/benchmarks/cfdbenchmarking/flow/dfg_benchmark2_re100.html. [Accessed 1 April 2021].

[51] A. Fortin, M. Jardak and J. J. Gervais, "Localisation of Hopf Bifurcations in Fluid Flow," *International Journal For Numerical Methods in Fluids,* vol. 24, p. 1185–1210, 1997.

[52] C. P. Jackson, "A finite-element study of the onset of vortex shedding in flow past variously shaped bodies," *Journal of Fluid Mechanics,* vol. 182, pp. 23 - 45, 2006.

[53] M. Sahin and R. G. Owens, "A numerical investigation of wall effects up to high blockage ratios on two-dimensional flow past a confined circular cylinder," *Physics of Fluids,* vol. 16, no. 1305, 2004.

[54] N. Kanarisa, D. Grigoriadis and a. S. Kassinos, "Three dimensional flow around a circular cylinder confined in a plane channel," *Physics of Fluids,* vol. 23, no. 6, 2011.

[55] J. Chen, W. G. Pritchard and S. J. Tavener, "Bifurcation for flow past a cylinder between parallel planes," *Journal of Fluid Mechanics,* vol. 284, pp. 23 - 41, 1995.

[56] G. D. McBain, "flostr private github repo," 2021.

[57] N. Schlömer, "dmsh 0.2.8," 7 November 2020. [Online]. Available: https://pypi.org/project/dmsh/. [Accessed 16 November 2020].

[58] T. Gustafsson and G. D. McBain, "scikit-fem: a Python package for finite element assembly," *The Open Journal,* vol. 5, no. 52, p. 2369, 2020.

[59] N. Schlömer, "dmsh 0.2.17," 24 June 2021. [Online]. Available: https://pypi.org/project/dmsh/. [Accessed 26 June 2021].

[60] A. Collette, "h5py," 22 June 2021. [Online]. Available: https://docs.h5py.org/en/stable/index.html. [Accessed 23 June 2020].

[61] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering,* vol. 9, no. 3, pp. 90-95, 2007.

[62] B. Belson, "Modred main repository," 2021. [Online]. Available: https://github.com/belson17/modred. [Accessed 2 April 2021].

[63] N. Bell, L. Olson and J. Schroder, "pyamg 4.1.0," 8 April 2021. [Online]. Available: https://pypi.org/project/pyamg/. [Accessed 9 April 2021].

[64] D. D. Team, "Dask: Library for dynamic task scheduling," 2016. [Online]. Available: https://docs.dask.org/en/latest/array-api.html?highlight=svd#dask.array.linalg.svd_compressed. [Accessed 23 January 2021].

[65] C. Rowley, "cwrowley/ibpm," 22 March 2020. [Online]. Available: https://github.com/cwrowley/ibpm. [Accessed 8 April 2021].

[66] C. Rowley, "ibpm," 7 November 2012. [Online]. Available: https://github.com/cwrowley/ibpm/blob/master/doc/examples/cylinder.cc#L24-L30 . [Accessed 14 April 2021].

[67] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python.," in *Proceedings of the 9th Python in Science Conference*, 2010.

[68] R. J. Hyndman, "Cyclic and seasonal time series," 14 December 2011. [Online]. Available: https://robjhyndman.com/hyndsight/cyclicts/. [Accessed 1 July 2021].

[69] E. W. Weisstein, "Fast Fourier Transform," MathWorld --A Wolfram Web Resource, 2021. [Online]. Available: https://mathworld.wolfram.com/FastFourierTransform.html. [Accessed 20 June 2021].

[70] H. Lütkepohl, New Introduction to Multiple Time Series Analysis, Springer-Verlag Berlin Heidelberg, 2005.

[71] N. Schlömer, G. D. McBain, K. L. christos, T. Li, M. Hochsteger, E. Keilegavlen, V. M. Ferrándiz, C. Barnes, V. Lukeš, L. Dalcin, M. Jansen, N. Wagner, A. Gupta, S. Müller and B. Woodsend, " nschloe/meshio: None (v4.4.6). Zenodo.," 4 June 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4900671. [Accessed 2 April 2021].

[72] A. Sidi, "Vector versions of Prony's algorithm and vector-valued rational approximations," *Advances in Computational Mathematics* , vol. 46, no. 30, 2020.

[73] X. Sun and J. Lei, "Limit cycle," Springer, New York, NY, 2013.

[74] K. Christakos, "Characterization of the coastal marine atmospheric boundary layer (MABL) for wind energy applications," University of Bergen, Bergen, 2013.

[75] M. V. Dyke, An album of fluid motion, Standford, California: The Parabolic Press, 1988.

[76] P. Meunier, S. L. Dizès and T. Leweke, "Physics of vortex merging," *C. R. Physique 6,* p. 431–450, 2005.

Glossary

Canonical
Simple and uniquely identifiable.

Coherent Structures
In turbulent flow a coherent structure refers to an observed or recorded flow structure which appears time invariant over a specified time domain whereby it can be analysed statistically despite the total flow having temporal variance.

Economy Sized SVD
The economy sized SVD is truncated to reduce computational costs as the last $m - n$ rows of Σ are all zero terms, thus the size of the U can also be truncated to reduce unnecessary multiplication of zero terms, as per **Figure 59**,where matrix A is equivalent to matrix X discussed in the introduction.

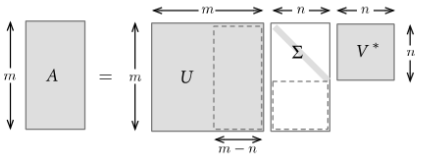

Figure 59. Economy- Sized SVD diagram [1]

Hopf Bifurcation
Is a mathematical theory which has applications in the representation of vortex shedding in fluid flow, further there are periodic solutions which exist at critical value synonymous with **Supercritical Hopf Bifurcation.** The bifurcation process furthermore represents the transition from laminar to turbulent flow [4].

Limit Cycle Behaviour
Bifurcation theory closely relates to limit cycle behaviour with supercritical Hopf bifurcation determining an occurrence of a stable limit cycle, whereby a trajectory is closed in phase space and other trajectories will spiral into the closed trajectory.

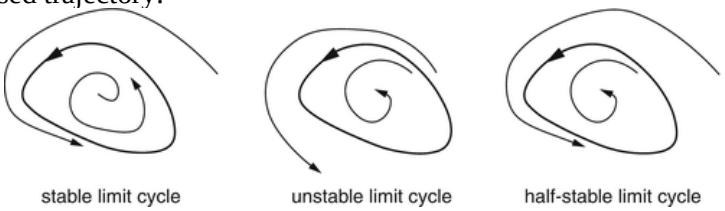

Figure 60. Limit cycle schematic diagrams [73]

Linearised Flow
"A flow of small perturbation about a steady base flow" [10].

Kelvin-Helmholtz Instability
Kelvin-Helmholtz instability refers to the phenomenon of instability which occurs when two adjacent, streams travel at different speeds.

Figure 61. Kelvin- Helmholtz instability observed in the upper cloud region [74]

Self - Oscillating
A signal that exponentially grows due to a negative damping term from small perturbations this especially shown in
Figure 9 from Interval I to Interval II.

Supervised Machine Learning
Supervised Machine Learning involves broadly classification, regression, optimisation and control, the objective of the learning is to find a function that maps input to output with a specified expected input-output pair.

Von Kármán vortex Shedding
Refers to the alternating dissipation of energy in the wake of a fluid flow as spiraling vortices divert from the mean flow normally due to interaction with a blunt surface e.g. a cylinder. A **Kármán Vortex Street** refers to an observed repeating pattern of vortex shedding.

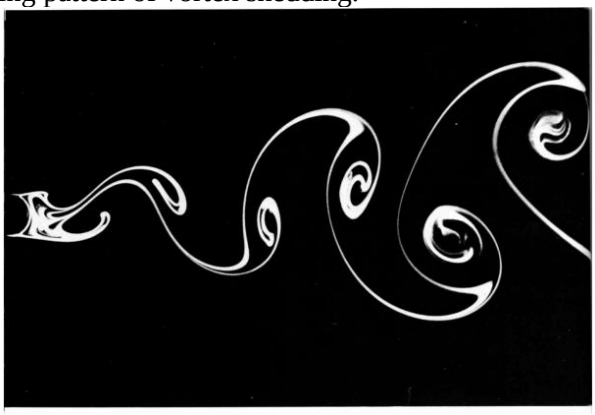

Figure 62. Kármán vortex Street with cylinder cross flow @ R=140, V=1.4cm/s,Ø=1cm [75]

Vortex Pairing/Merging

Vortex merging occurs when a pair of vortices of the same rotational direction with near parallel axes, interact at a critical distance, superposition of the velocity fields occur to create one vortex. [76]

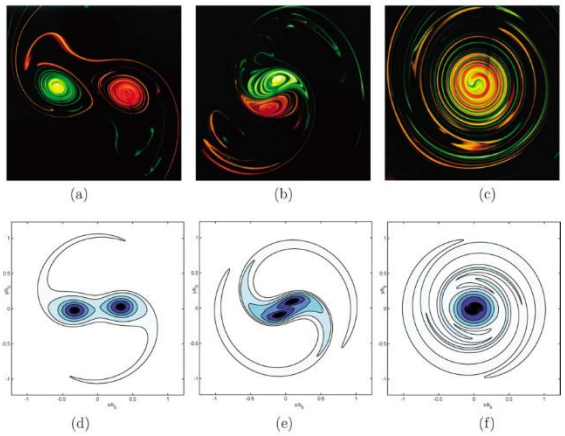

Figure 63. 2-D vortex pairing/merging of experiment dye visuals (a)-(c) & DNS (d)-(f). snapshots are taken (a),(d) before merging (b),(e) during merging & (c),(f) after merging [76]

Unsteady Flow

Flows that are dependent on a time variable and thus vary with respect to time such as cross flow of a cylinder.

Unsupervised Machine Learning

Unsupervised Machine Learning involves broadly clustering & dimensionality reduction of data, the objective is to identify patterns in data through minimisation or maximisation of an objective function.

Appendices

Appendix A. Code Repository

Code Repository
https://github.com/PythonpadawanEXE/flostr_report

Appendix B. Assortment of Flow Structure Figures

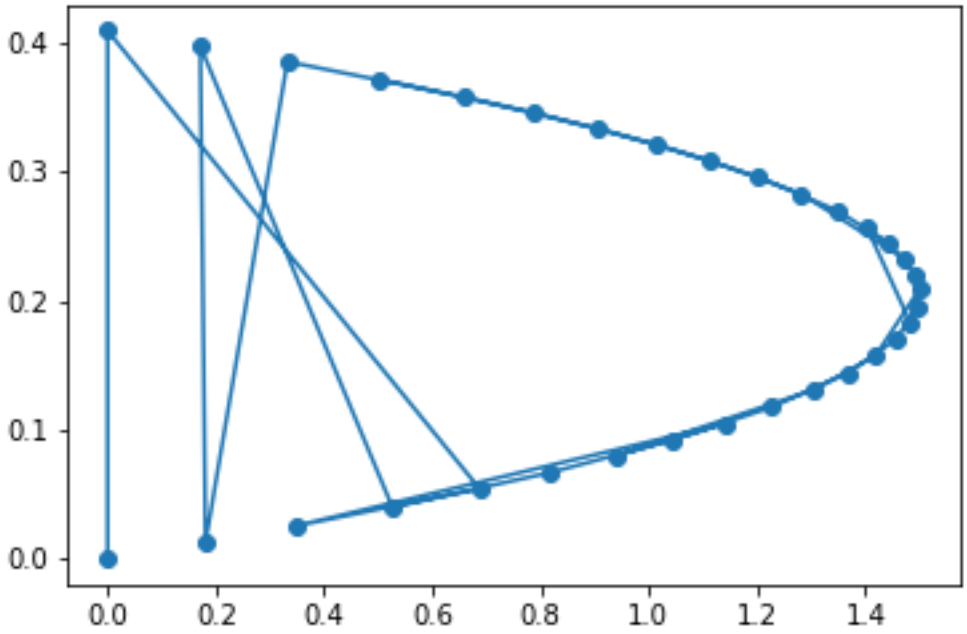

Figure 64. Poiseuille velocity profile with Bottom axis as streamwise velocity and left axis as Y position perpendicular to direction of flow

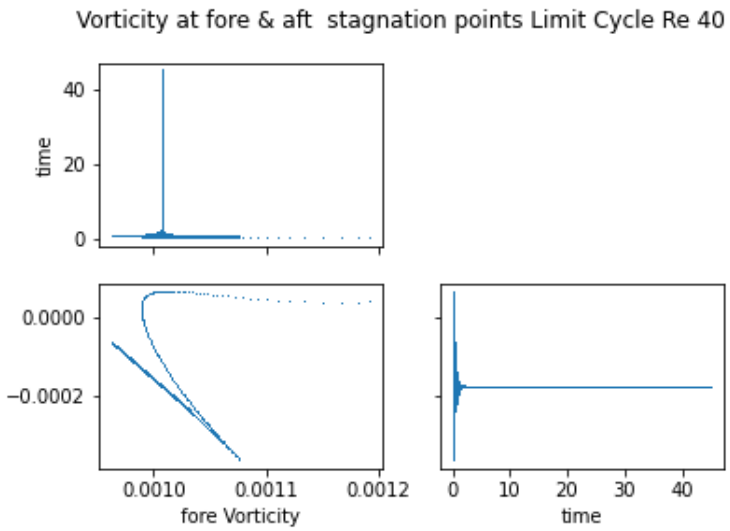

Figure 65. Vorticity probe for Re = 40 condition for restart facility used as initial condition for supercritical cases

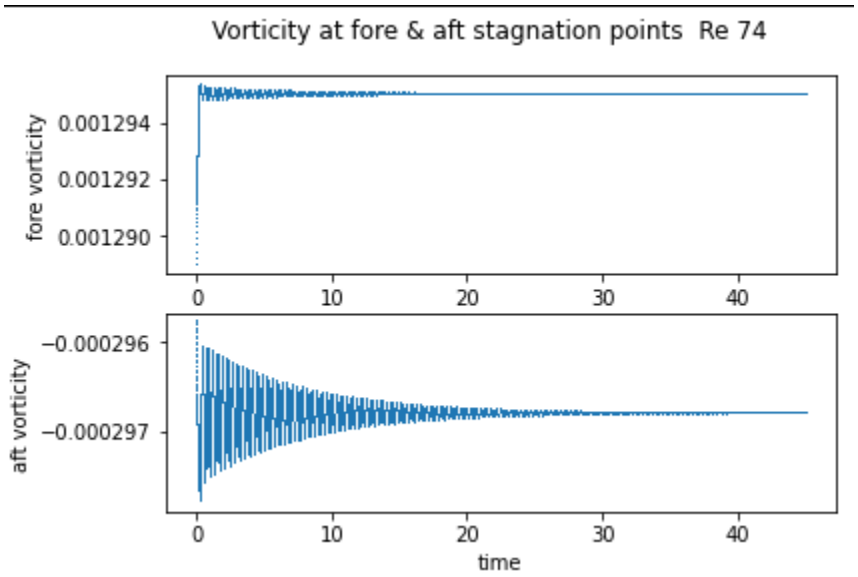

Figure 66. Subcritical Reynolds number case demonstrating volatile initial transients at time < 0.1

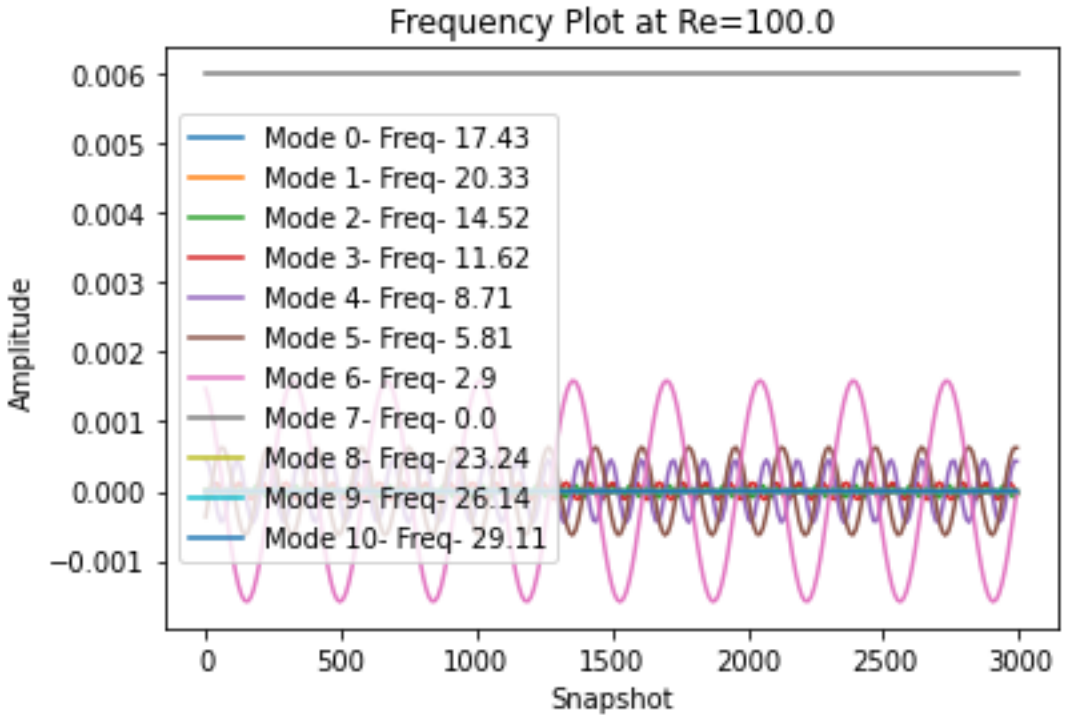

Figure 67. DMD time dynamics find peaks used to find frequency st08-NS case where one unit time is 1000 snapshots, since down sampling = 11 and dT = 0.001 from DMD_ Modred_Kutz.py

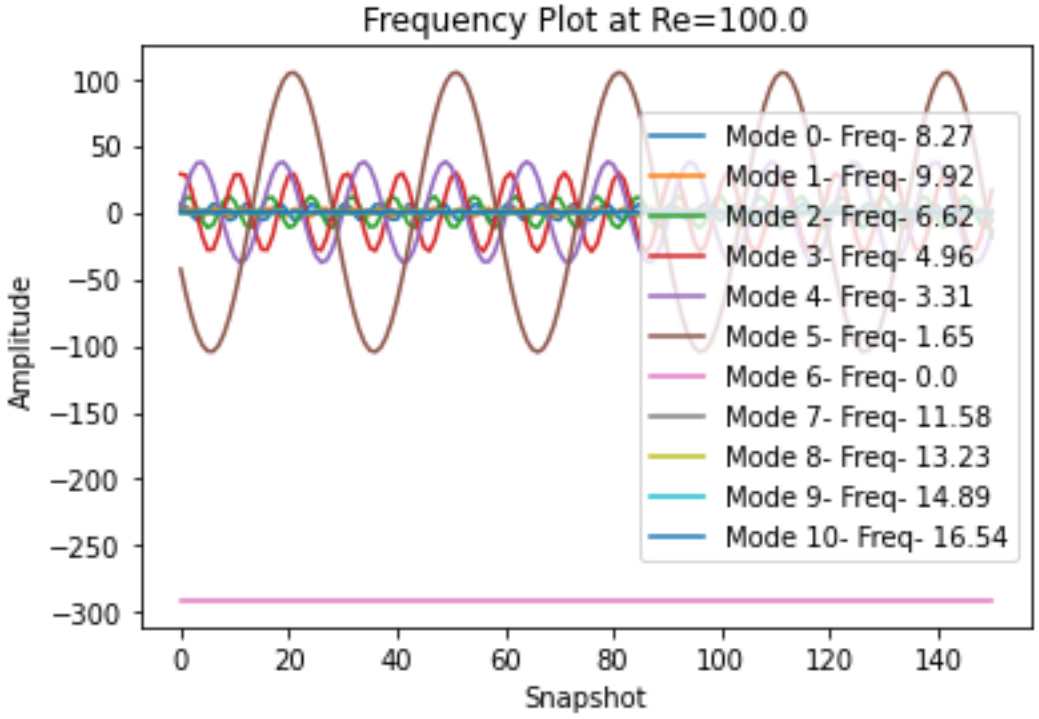

Figure 68. DMD time dynamics find peaks used to find frequency VORTALL case where one unit time is 50 snapshots since dT = 0.02 and down sampling = 1 from DMD_Test_Modred.py

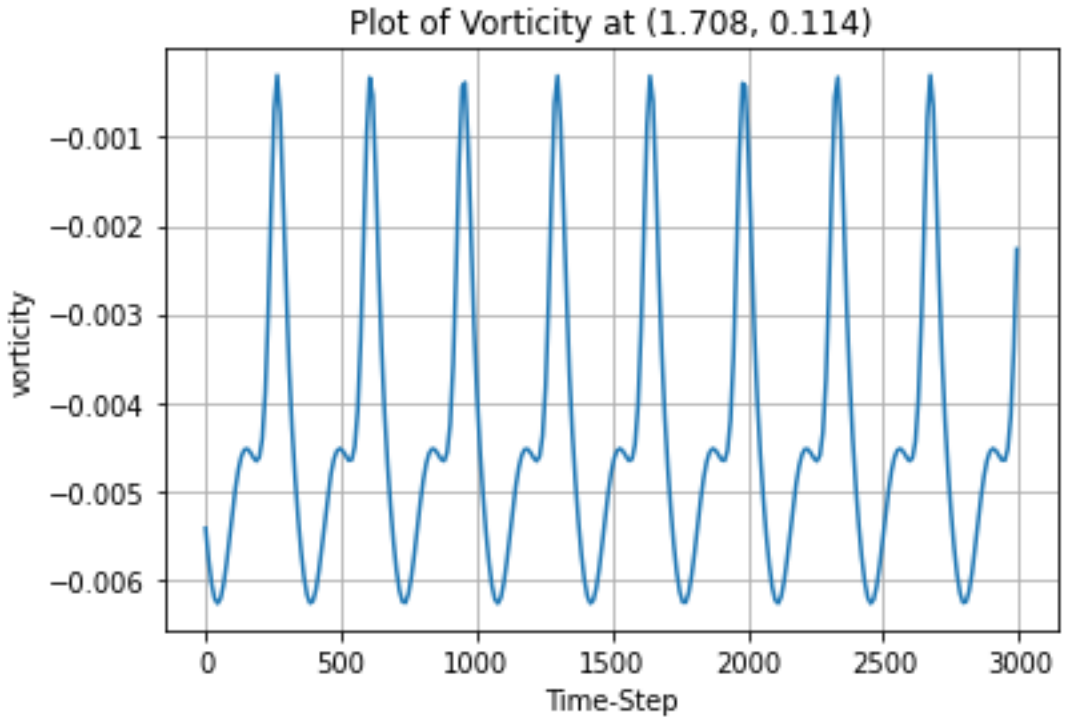

Figure 69. Plot of vorticity probe for st08-NS case Re = 100

Appendix C. Assortment of Code Snippets

Code Snippet 10. DMD_MODRED_Kutz.py line 123-139 demonstrating find_peaks for frequency of time dynamics

```
n= vortall.shape[1]
time_dynamics = np.zeros((MODES.shape[0],n),dtype=complex)
t = len(x[0])
ssf = 1
for i in range(t):
      time_dynamics[:,i] = (Bs*np.exp(OMEGAS*(i)*dT))



fig,ax = plt.subplots()
ax.set(xlabel = 'Snapshot', ylabel = 'Amplitude', title=str('Frequency Plot
at Re={}'.format(Rey)) )
for i in range(MODES.shape[0]):
      ax.plot(list(np.array(range(t))*ssf)[:],(time_dynamics[i].real)[:],
label="Mode {}- Freq- {}".format(i,round(FREQS[i],2)))
      peaks_A, _A = find_peaks(time_dynamics[i])
      troughs_A, t_A = find_peaks(-time_dynamics[i])
      cycles = round((len(peaks_A)+len(troughs_A))/2,0)
      print("For mode",i,"Freq is",cycles/(n/1000))
ax.legend()
plt.show()
```

Code Snippet 11. VAR(p) and Prony reconstruction credit to G. D. McBain from VAR_vorticity.py

```
def L(Q: np.ndarray, mu: np.ndarray, m: int) -> np.ndarray:
    M = diags(mu)
    return np.vstack([Q @ M ** k for k in range(m)])
def varp(
    snapshots: np.ndarray, p: int, trend: str = "c"
) -> Tuple[np.ndarray, np.ndarray, np.ndarray]:
    """Return matrix containing horizontally stacked VAR coefficients,
eigenvalues, and companion eigenvectors
    The first column is the constant inhomogeneous term, the succeeding
    squares multiply the sumccessive lags from 1 through p.
    The snapshots should also be horizontally stacked columns.
    """
    if trend != "c":
        raise NotImplementedError

    r, m = snapshots.shape

    # The notation Y = B Z is from
    # https://en.wikipedia.org/wiki/General_matrix_notation_of_a_VAR(p)

    Y = snapshots[:, p:]
    Z = np.vstack(
        [np.ones(m - p), *[snapshots[:, r : r + m - p] for r in range(p -
1, -1, -1)]]
    )
    B = np.linalg.lstsq(Z.T, Y.T, rcond=None)[0].T

    uinf = np.linalg.solve(
        np.eye(r) - sum(B[:, i : i + r] for i in range(1, p * r + 1, r)),
B[:, 0]
    ) #intercept values or constants

    # first Frobenius companion form (Mackey, Mackey, & Tisseur 2015,
Section 3)

    companion = np.vstack([B[:, 1:], np.eye((p - 1) * r, p * r)])
    mu, imbedded_eifs = np.linalg.eig(companion)
    eifs = imbedded_eifs[:r]

    # reconstruction (Le Clainche & Vega 2017 SIADS, Sections 2.1.3, 2.2.3)

    ell = L(eifs, mu, m)
    lstsq = np.linalg.lstsq(ell, (snapshots - uinf[:, None]).flatten("F"),
rcond=None)
    a = lstsq[0] #mode amplitudes or prony coefficents

    return B, mu, eifs[:r], uinf, a, ell

p = 2 #number of lags
r = 8 #number of modes
downsampling = 16
v = POD.proj_coeffs[:r, ::downsampling]
B, mu, eifs, uinf, a, ell = varp(v, p)
reconstruction = (ell @ a).reshape((r, m), order="F") + uinf[:, None]
```