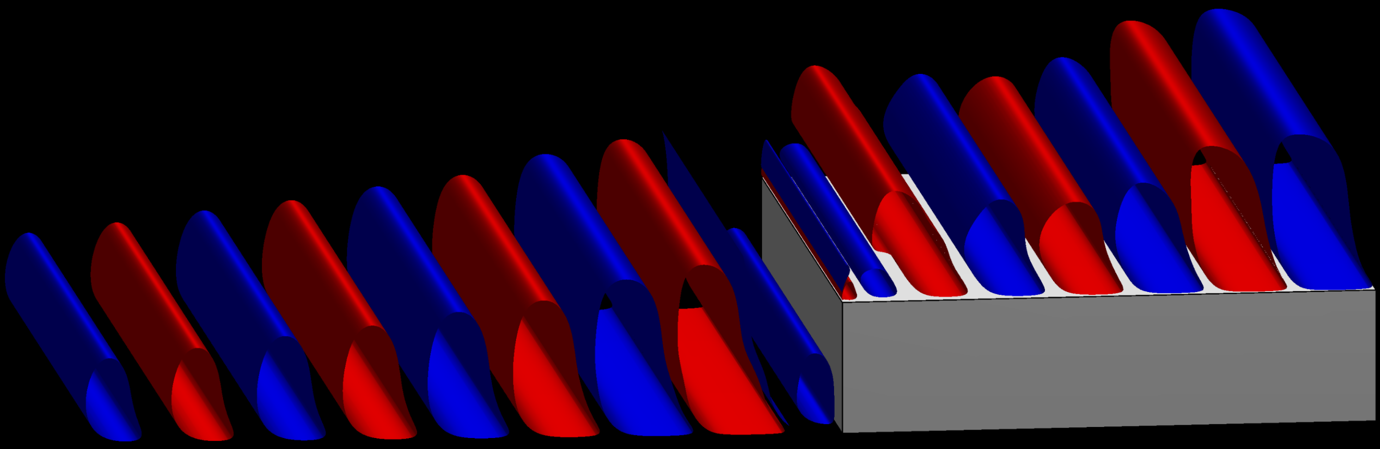
Goal-Oriented Reduced-Order Modeling of the Incompressible Navier-Stokes Equations

Application to a 2D transitional boundary layer



Technische Universiteit Delft

Peter Blom

Goal-Oriented Reduced-Order Modeling of the Incompressible Navier-Stokes Equations

Application to a 2D transitional boundary layer

by

Peter Blom

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday November 5th, 2021 at 1:30 PM.

Student number:	4475011	
Project duration:	September 6th, 2020 – November 5th, 2021	
Thesis committee:	Dr. M. Kotsonis,	Aerodynamics
	Dr. S. J. Hulshoff,	Aerodynamics
	Dr. S. R. Turteltaub,	Aerospace Structures and Computational Mechanics
	Ir. J. Casacuberta Puig,	Aerodynamics

Acknowledgements

When originally starting my BSc degree in Aerospace Engineering back in 2015, I knew that I would at some point be able to call myself an engineer. This idea was a relatively abstract concept in my mind and too far away to really realize what that means. Six years later, becoming an engineer is no longer too far away and I have no doubt in my mind I will see it as one of my most proud achievements. This achievement is by no means my own and thanks to many great people.

The first person I should mention is my supervisor Dr. Steven Hulshoff. His advice, positive attitude and help with the implementation have been of critical importance to the completion of this work. I can only express my sincerest gratitude for the time he has spend on helping me finishing this project. Moreover, I should thank Jordi Casacuberta Puig and Dr. Marios Kotsonis for their guidance and insights into the flow physics of the stability part of this thesis. Furthermore, I would like to thank Dr. Sergio Turteltaub for agreeing to be a part of the graduation committee.

The DNS to obtain the data used in this work was not performed by me. Instead, my work relied extensively on previous work done by Haris Shahzad and I would like to express my gratitude for supplying the data and taking the time to help me with any questions I had. Furthermore, the work previously done by Dr. Lei Cheng and Tariq Shajahan has been very useful for my understanding of the subject. And although I have never met both authors in person, I would like to thank them for the influence their work has had on completing this thesis. I also would like to thank Oscar for our fruitful discussions on the GOROM technique and his willingness to provide feedback on a draft of this report.

Furthermore, my parents have supported me throughout my study and I can say with confidence that I would not have made it this far without them. I am very grateful for their mental support throughout the years. Another person that has had a major influence on my mental well being, which was especially appreciated during the COVID pandemic, is my girlfriend Carlijn. Her support has always helped me to keep going when things did not seem to go to plan. Finally, I would like to thank my friends, some of whom I have know since I was three years old. Hanging out with them over the past years has made my time as a student so much more enjoyable. In particular, I should thank Pascal, Simone and Roel for also agreeing to check the quality of my sometimes lackluster writing.

Peter Blom
Naaldwijk, October 2021

Abstract

Research into constructing reduced-order models (ROM) to reduce computational cost or to interpret complex datasets from numerical or experimental sources has steadily been gaining popularity over the past few years. Many methods currently exist to construct ROMs. The proper orthogonal decomposition (POD) is one of the most widely adopted methods used for these purposes in the field of fluid dynamics. The POD modes favor high-energy structures due to the formulation of the POD however. This may result in an inaccurate ROM, as it might be possible that dynamically relevant, yet low-energy structures are truncated in favor of higher-energy structures. In recent years, an alternative goal-oriented method has been proposed. This goal-oriented method not only has less limits on the goal function that is optimally represented relative to the POD, the resulting reconstruction of the flow field is model constrained as well. Both these properties are conjectured to lead to more dynamically relevant modes. This goal-oriented, reduced-order modeling (GOROM) technique has been formalized for the advection, Burger's and incompressible Navier-Stokes equations in 1D and the Stokes equations in 2D, all with homogeneous boundary conditions. This work extends the formulation to the 3D incompressible Navier-Stokes equations with inhomogeneous boundary conditions and provides a parallel implementation. The method is applied to a DNS dataset of a 2D transitional boundary layer with and without a forward-facing step present and compared to the performance of the POD. It is found that the GOROM modes provide a basis which is numerically more accurate than the POD modes for most cases, yet do not provide additional insight into the flow physics. The results presented nevertheless show promise for further applications to construct more accurate and goal-oriented ROMs with the GOROM technique, especially for problems with nonlinear interactions and nonlinear goal functions.

Contents

List of Figures **viii**
List of Tables **x**
1 Introduction, Literature Review and Objectives **1**
1.1 Projection-Based Reduced-Order Modeling 2
1.1.1 Classical POD 2
1.1.2 Dynamic Mode Decomposition 4
1.1.3 Spectral POD 4
1.1.4 Goal-Oriented Reduced-Order Modeling 4
1.1.5 Projection-based Reduced-Order Modeling Comparison 6
1.2 Boundary layer stability 6
1.2.1 Stability Theory 7
1.2.2 Tollmien-Schlichting Problem 8
1.2.3 Boundary layers with steps 9
1.3 Research Objectives and Questions 10
2 Semi-Continuous Formulation of Goal-Oriented Reduced-Order Modeling **11**
2.1 Model Equations 11
2.2 Formulation of the Optimization Problem 12
2.3 Gradient Calculation 12
2.4 Optimization Technique 13
2.5 Secondary Basis Function 14
3 Goal-Oriented Reduced-Order Modeling of 3D Incompressible Navier-Stokes **16**
3.1 Choice of Basis 16
3.1.1 Separation of Variables 16
3.1.2 Secondary Basis 17
3.2 Model Constraint 17
3.2.1 Sub-Grid Scale Model 18
3.2.2 Boundary Conditions 18
3.2.3 Final Model Constraint 19
3.3 Lagrangian Formulation 20
3.4 State Equations 20
3.5 Adjoint Equations 21
3.5.1 Adjoint Boundary Conditions 21
3.5.2 Stabilization 22
3.5.3 Final Adjoint Equations 23
3.6 Gradient Equations 24
3.7 Possible Goal Functions 26
3.8 Gradient Verification 29
3.8.1 1D Case 30
3.8.2 2D Case 32
3.8.3 3D Case 34
4 Tollmien-Schlichting Instability on a Flat Plate **36**
4.1 Numerical Setup 36
4.2 POD Reduced-Order Model 37
4.2.1 Verification 38
4.2.2 Validation of Modal Solver 39
4.2.3 ROM Performance 39
4.3 Goal-Oriented Reduced-Order Model 40
4.4 Concluding Remarks 41
5 Tollmien-Schlichting Instability with Step **42**
5.1 Cases and Numerical Setup 42
5.2 POD Reduced-Order Model 45
5.2.1 Verification 47

5.2.2 ROM Performance . . . 48
5.3 Goal-Oriented Reduced-Order Model . . . 49
5.3.1 Optimizations . . . 49
5.3.2 Mode Shapes and Amplitudes . . . 50
5.4 Concluding Remarks . . . 53
6 Computational Cost and Scaling **55**
6.1 Parallel Scaling . . . 55
6.2 Influence of the Number of Basis Functions . . . 56
6.2.1 Influence of the number of ϕ . . . 56
6.2.2 Influence of the number of ψ . . . 58
7 Conclusion and Recommendations **60**
7.1 Conclusions . . . 60
7.2 Discussion . . . 61
7.3 Recommendations & Future Work . . . 61
A Additional Considerations Stability Theory **63**
A.1 Orr-Sommerfeld Equations . . . 63
A.2 Solving the Orr-Sommerfeld Equations . . . 64
A.2.1 Spatial & Temporal Approach . . . 64
A.2.2 Inviscid & Viscous Instabilities . . . 65
A.2.3 e^N Method . . . 65
A.3 (Nonlinear) Parabolized Stability Equations . . . 65
A.4 Additional Considerations Tollmien-Schlichting Problem . . . 66
A.4.1 Squire Theorem . . . 66
A.4.2 Neutral Stability Curve . . . 66
A.4.3 Secondary Instabilities and Breakdown . . . 67
B Derivations of the Adjoint and Gradient for the General Formulation **69**
B.1 Derivation of the Adjoint . . . 69
B.2 Derivation of the Gradient . . . 71
C Algorithms **73**
C.1 Trust-Region Method . . . 73
C.2 Conjugate-Gradient Method . . . 73
D Derivation of the Model Constraint **75**
D.1 Projection of Governing Equations . . . 75
D.1.1 Momentum equation . . . 75
D.1.2 Continuity Equation . . . 76
D.2 Varitiational-Multiscale Method . . . 76
D.3 Final Model Constraint . . . 78
E Shape Functions **79**
E.1 Secondary Shapes Flat Plate . . . 79
E.2 Secondary Shapes Step Case . . . 82
E.2.1 Secondary Shapes Low Boundary Case . . . 82
E.2.2 Secondary Shapes High Boundary Case . . . 85
E.3 Primary Shapes Step Case . . . 88
E.3.1 Case B . . . 88
E.3.2 Case C . . . 90
E.3.3 Case H . . . 93
Bibliography **98**

List of Figures

1.1 Simplified visual representation of the processes that a perturbation can go through to cause transition in a boundary layer, redrawn from Saric et al. (2002). 7
1.2 Typical shape of the v component (a) and u component (b) of a TS wave. The thin lines are the imaginary and real parts of the shape function. Taken from Schmid and Henningson (2001). . . 8
1.3 Streamwise velocity of the base flow of the highest step considered from Shahzad (2020) along with isolines of the streamfunction. 9

2.1 Basic Flow chart of the procedure followed to solve the optimization procedure. 13
2.2 Visual representation of the choice of types of basis functions. 15

3.1 Objective functional history for several starting points 31
3.2 Response surface of the Lagrangian for the 1D MFS case along with the optimization history. . . 31
3.3 History of the value of the objective functional in the 2D manufactured solution case for a linear and non-linear goal. 32
3.4 Optimization history of the coefficients of the x-velocity components with a linear and non-linear goal. 33
3.5 Optimization history of the coefficients of the y-velocity components with a linear and non-linear goal. 33
3.6 Optimization history of the coefficients of the pressure with a linear and non-linear goal. 33
3.7 History of the value of the objective functional in the 3D manufactured solution case for a linear and non-linear goal. 35

4.1 Wireframe of part of the DNS mesh used in the simulations of the flat plate. Only every 8th and 16th point are plotted in the wall-normal and streamwise direction respectively. 37
4.2 Normalized singular values of the reference data for the flat plate case. 37
4.3 First 2 POD modes of the x-velocity for the flat plate case. 38
4.4 Amplitudes of the u(1,0) and u(2,0) component for a flat plate case with different temporal refinements. 39
4.5 Resulting GOROM shapes for the x-velocity with a perturbed starting condition and the global u goal function. 41
4.6 Amplitudes of the streamwise GOROM modes of case C from table 4.4. 41

5.1 Plot of the mean-flow distortion around the step of the two datasets considered in this work. The x-velocity is plotted in these figures. 43
5.2 Wireframe of part of the DNS mesh used in the simulations of the step case with the lower top boundary. Only every 8th and 16th point are plotted in the wall-normal and streamwise direction respectively 43
5.3 Cell location in the wall-normal direction of the two step datasets compared. 44
5.4 Normalized singular values of both the datasets considered in this work. 45
5.5 Plot of the first two POD modes used in cases D-H. The modes originate from the x-velocity. . . 46
5.6 Non-normalized, projected amplitudes of the first four POD modes from case D-H. 46
5.7 Shape function of the third POD mode from the dataset of case A-C. 47
5.8 Non-normalized, projected amplitudes of the first three POD modes from case A-C. 47
5.9 Value of the objective functional plotted against the number of POD modes retained in the basis. The goal function is the global u goal function from table 3.1. 48
5.10 Plot of the GOROM modes of case B and their associated amplitudes. The modes originate from the x-velocity. 51
5.11 Amplitudes of the modes plotted in figure 5.12. 52
5.12 Plot of the GOROM modes of case C. The modes originate from the x-velocity. 52

5.13 Plot of the GOROM modes of case H. The modes originate from the x-velocity. 53
5.14 Amplitudes of the modes plotted in figure 5.13. 54
5.15 Amplitudes of the mode 3 and 4 from case H decomposed into their harmonic components. . . 54

6.1 Evaluation time of the total gradient plotted against the number of cores used for several numbers of primary basis functions. Note that the problem does not change as the number of cores increases, thus strong scaling is investigated here. 56
6.2 Speedup of the gradient, including the individual components that make up the total gradient. This investigation uses *strong* scaling. 57
6.3 Slowdown ratio plotted against the number of primary basis functions for several choices of the number of secondary basis functions. 58
6.4 Relative computational cost of the different components making up the gradient integral 58
6.5 Slowdown ration plotted against the number of secondary basis functions. 59

A.1 Graphical representation of an absolute and convective instability, from Boiko et al. (2012) . . . 65
A.2 Neutral stability curve of the Blasius boundary layer with contours of the growth rate. The shaded area has positive growth rates and are thus unstable. Taken from Schmid and Henningson (2001). 67

E.1 First 4 POD modes of the x-velocity for the flat plate case. 79
E.2 First 4 POD modes of the y-velocity for the flat plate case. 80
E.3 First 4 POD modes of the pressure for the flat plate case. 81
E.4 First 6 POD modes of the x-velocity for the step case with a low top boundary. 82
E.5 First 6 POD modes of the y-velocity for the step case with a low top boundary. 83
E.6 First 6 POD modes of the pressure for the step case with a low top boundary. 84
E.7 First 6 POD modes of the x-velocity for the step case with a high top boundary. 85
E.8 First 6 POD modes of the y-velocity for the step case with a high top boundary. 86
E.9 First 6 POD modes of the pressure for the step case with a high top boundary. 87
E.10 GOROM modes of the velocities for case B. 88
E.11 GOROM modes of the pressure for case B. 89
E.12 GOROM modes of the x-velocity for case C. 90
E.13 GOROM modes of the y-velocity for case C. 91
E.14 GOROM modes of the pressure for case C. 92
E.15 GOROM modes of the x-velocity for case H. 93
E.16 GOROM modes of the y-velocity for case H. 94
E.17 GOROM modes of the pressure for case H. 95

List of Tables

1.1 Overview of the Pros and cons for different projection based methods. 6

3.1 Goal functions used throughout this work. 27
3.2 Resulting values for the optimization variables and objective functional for a 1D manufactured solution case with different starting conditions. 31
3.3 Resulting values for the optimization variables and objective functional for a 2D manufactured solution case with different goal functions. 34
3.4 Resulting values for the optimization variables and objective functional for a 2D manufactured solution case with different goal functions. 35

4.1 Values of the OF for several mesh refinements and number of timesteps. The columns x & y denote the downsampling factor used. Each simulation is ran for 80 timesteps. 38
4.2 Values of the OF for several number of timesteps. The difference column is with respect to case with the largest number of timesteps. 38
4.3 Values of the OF for several goals from table 3.1. 40
4.4 Optimization parameters and results. 40

5.1 The number of streamwise elements in the mesh per wavelength at the step location and at the outflow of the domain. The wavelengths are inferred from the flat plate case. 43
5.2 Cases used in the optimizations. A description of the goal functions can be found in table 3.1. . 44
5.3 Values of the OF for several number of timesteps. The difference columns are with respect to case with the largest number of timesteps. 48
5.4 Values of the OF for several mesh refinements and number of timesteps. The columns x & y denote the downsampling factor used. 48
5.5 Values of the OF for several choices of goal function. The number of POD modes retained in the basis corresponds to the column *N* from table 5.2. 49
5.6 Results of the optimizations defined in table 5.2. Some of the relevant parameters are present here as well for clarity. The goal function numbers correspond to the numbers found in table 3.1. 50

Nomenclature

The text in the following report contains an explanation of each variable used in the equations. Therefore, the following list is incomplete and only contains the most important symbols in the document that are repeated several times in the report.

Latin Symbols

$\hat{u}$	Resolved scales of the reduced-order model
f	Source term
g	Goal function
p	Pressure
Re	Reynolds number
U_b, V_b, W_b	Base flow velocity component in the directions x, y, z respectively
x, y, z	Cartesian coordinates in the streamwise, wall-normal and spanwise direction respectively

Greek Symbols

α	Streamwise wavenumber of a perturbation
α_i	Temporal amplitude of a spatial mode
β	Spanwise wavenumber of a perturbation or regularization parameter in the context of GOROM
η	Variation applied to optimal value of optimization variable
Γ	Boundary of the integration domain
λ, μ	Adjoint variables
ν	Kinematic viscosity of air
Ω	Integration Domain
ω	Temporal frequency of a perturbation
ϕ	Primary basis function
ψ	Secondary basis function

Miscellaneous

$(a, b)_\Omega$	Inner product: $\int_\Omega ab d\Omega$
$\langle a \rangle$	Time averaging operation of signal a
$\mathbb{R}$	Real numbers
∇	Nabla operator: $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]^T$

Abbreviations

(N)PSE	(Nonlinear) parabolized stability equations
2D	2-dimensional
3D	3-dimensional
DNS	Direct numerical simulation
GOROM	Goal-oriented reduced-order modeling
OS	Orr-Sommerfeld
POD	Proper orthogonal decomposition
ROM	Reduced-order model
SCF	Semi-continuous formulation
SGS	Sub-grid scale
TS	Tollmien-Schlichting
VMM	Variational-multiscale method

1

Introduction, Literature Review and Objectives

Reduced-order models have widely been utilized to interpret high-dimensional datasets produced by numerical or experimental simulations of fluid flows. Most notably, proper orthogonal decomposition (POD) has been used with success (Lumley, 1967). The POD provides spatial structures of the flowfield that optimally represent the L_2 norm of the flow variables. This optimality makes the POD biased towards high energy structures. A more advanced technique developed by Bui-Thanh et al. (2007) and extended by Cheng et al. (2015), allows for a chosen goal function to be taken into account while also model constraining the resulting reconstruction of the flow field. These two properties of the method aim to provide a more general and dynamically relevant set of modes which could lead to a more suitable basis for interpretation or reduced-order modeling.

One topic where this technique could prove to be useful is the investigation into transitional flow. Generally speaking, transitional flow describes an area of the flow field in which the flow does remain laminar, yet small perturbations are able to grow in size, eventually leading to turbulent breakdown. Ever since the discovery of laminar-to-turbulent transition by Reynolds (1883), researchers have tried understanding the physical processes leading to this phenomenon. This has lead to the emergence of a field called *hydrodynamic stability* (Schmid & Henningson, 2001).

The investigation of transitional boundary layers is a particularly relevant application of the field of hydrodynamic stability. Usually the boundary layer over a surface is initially laminar and at some point downstream transitions to turbulent. It is generally advantageous for slender bodies to maintain laminar flow over as much of its surface as possible due to the increased skin-friction produced by a turbulent boundary layer (see e.g. Anderson (2017, ch.15)). The wing of an aircraft is one such surface on which reducing the skin-friction could increase the energy efficiency of an aircraft substantially. According to Torenbeek (2013, p.110), it is theoretically possible to achieve a 30% reduction in fuel usage if the transition of the boundary layer from laminar to turbulent on both sides of the wing and tail surfaces is delayed sufficiently. Although the practical implementation to achieve such a delay in transition still remains a challenge, there is a considerable potential for fuel savings. With the goal of the aviation industry to reach a reduction in net CO_2 emissions of 50% by 2050 relative to emissions produced in 2005[1], the large potential of increasing fuel efficiency by reducing the extent of turbulent flow over an aircraft motivates further research into methods to achieve this potential.

Although recent research into reducing the skin-friction of a turbulent boundary layer by means of small dimples on the surface is gaining popularity (e.g. Nesselrooij et al., 2016), the techniques Torenbeek (2013) refers to aim to limit the extent to which small perturbations in the transitional region of the boundary layer grow and are called *laminar-flow control*. In order to devise successful techniques and active flow control schemes, a thorough understanding and accurate reduced-order models of the transitional region in boundary layers are required. On smooth surfaces the transition process is understood quite well. In practice however, wings of an aircraft show steps and gaps introduced during the manufacturing process, which affect the transition behaviour of the boundary layer (see e.g. Drake et al., 2010).

---

[1] See: https://www.iata.org/en/programs/environment/climate-change/, *date accessed: 10th of December 2020*

To investigate the effect surface imperfections have on growth of perturbations in a boundary layer, a research group at the TU Delft is performing direct numerical simulations (DNS) of transitional flow over a forward-facing step. The conjecture is that applying the advanced dimension reduction technique described by Cheng et al. (2015) to the problem of a transitional boundary layer with a step can aid in interpreting the data produced by the DNS. Additionally, recent advancements in machine learning can also reduce the dimensions of the flow problem (see e.g. Brunton et al. (2020) for a review). The physical interpretation using these techniques will be lost, however. Therefore, this work will only focus on so-called *projection-based* dimension reduction techniques that provide structures to project the governing equations on.

The remaining part of this chapter is structured as follows. First, some popular projection-based techniques are reviewed. The second section reviews the standard methods used in boundary-layer stability theory, the review is closed by considering some of the challenges faced by traditional modeling techniques when including a step in the boundary layer. This lays the connection between both sections and provides the motivation for this thesis. The third section then introduces the research objectives and questions for this work and finally, the chapter is closed by introducing the structure of the remaining report. [2]

1.1. Projection-Based Reduced-Order Modeling

This section reviews the literature on projection-based reduced-order modeling. Many methods exist in this field, thus only a few key methods that relate to the flow problem of interest are discussed here. For exactly that reason the popular method Balanced POD (Rowley, 2005; Willcox & Peraire, 2002) is excluded in the upcoming detailed discussions. This method is particularly aimed at control problems and thus requires a set of input variables, which are not readily available in the context of this problem.

The first section deals with classical POD, after which another popular decomposition technique called the Dynamic Mode Decomposition is treated. The following section discusses Spectral POD, which is followed by a thorough investigation in a more advanced goal-oriented reduced-order modeling (GOROM) technique. Finally, this section is concluded with a table providing a very broad summary of the pros and cons of using these methods.

1.1.1. Classical POD

The (POD) is in fluid dynamics the projection-based technique most often used to reduce the degrees of freedom of a particular flow problem. The POD finds a set of orthonormal modes that optimally represent the flowfield with the least amount of modes possible. Each mode has a corresponding weight that quantifies the importance of the mode.

Although usually referred to as the POD in fluid dynamics, the method is known by many names in other fields like: principle component analysis or the Karhunen–Loève decomposition (Berkooz et al., 2003). Lumley (1967) first introduced this method to a fluid-dynamics problem. From there, the method has was successfully applied to a variety of problems, including: aeroelastic, design optimization and control problems. See the review article by Lu et al. (2019) for a recent and extensive overview of the many applications of POD.

The POD aims to find spatial modes ($\boldsymbol{\phi}(\mathbf{x})$) that maximize:

$$\max_{\boldsymbol{\phi}} \langle || \left(\mathbf{u}, \boldsymbol{\phi}\right)_{\Omega} || \rangle \tag{1.1}$$

where $\mathbf{u}$ represents the flowfield in time, $\langle . \rangle$ is an averaging operator in time and $(a,b)_{\Omega} = \int_{\Omega} ab d\Omega$ is the inner product. The POD modes are constraint to be orthonormal, that is:

$$\left(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j\right)_{\Omega} \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{1.2}$$

Of these spatial modes, m modes can be chosen such that the lower order representation of the system can be reconstructed as in equation 1.3 (Berkooz et al., 2003).

$$\hat{\mathbf{u}} = \sum_{j=1}^{m} \alpha_j(t) \boldsymbol{\phi}_j(\mathbf{x}) \tag{1.3}$$

---

[2] Do note, some of this chapter reprints some sections from an assignment performed by this author as part of this project. The reader might find some similarities from a document handed in on the 29th of January, 2021.

where the temporal coefficients can be found using the inner product, as in equation 1.4, or through the eigenanalysis used to construct the modes. Alternatively, the governing equations can be projected onto the POD basis functions to obtain a ROM that can be evaluated for a reduced cost. When this is done for a nonlinear system like the Navier-Stokes equations, truncated smaller scales might result in an unstable model. Modeling their effect on the resolved scales through a sub-grid scale (SGS) model might be required to obtain a robust ROM (see e.g. Bergmann et al., 2009)

$$\alpha_j(t) = \left(\mathbf{u}(t), \boldsymbol{\phi}_j\right)_\Omega \tag{1.4}$$

The method initially introduced by Lumley (1967) is a method which later is referred to as the *spatial POD*. To perform the spatial POD, the formulation from the review paper by Taira et al. (2017) is briefly described here. First a snapshot matrix is attained through a numerical simulation or experimental measurements. Usually the mean value of the velocity is subtracted from the velocity field such that the decomposition is performed on a zero-mean flowfield. This snapshot matrix is a nxm matrix $X \in \mathbb{R}^{n\text{x}m}$ in which each column corresponds to a 'snapshot' of the flow stacked in 1 *n-dimensional* vector.

When this snapshot matrix is multiplied by its transpose, the spatial covariance matrix is obtained as a nxn matrix, i.e. $R = XX^T \in \mathbb{R}^{nxn}$.[3] Performing the eigenvalue decomposition of this correlation matrix yields n eigenvectors and eigenvalues. The eigenvectors describe the spatial modes and the corresponding eigenvalue the mean energy of the mode. To achieve order reduction, the first r modes can be chosen based on their energy content.

Although this method of computing the POD modes is relatively straightforward, problems arise when considering very large datasets. The eigenvalue decomposition is performed on the matrix $R \in \mathbb{R}^{nxn}$, where n is the number of degrees of freedom in the dataset. Since CFD computations are usually performed on large spatial grids with several flow variables, this decomposition becomes computationally expensive to perform.

Sirovich (1987) introduced the *method of snapshots*, which greatly increased the applicability of the method to large datasets. The method of snapshots is applied by performing the eigenvalue decomposition on the temporal covariance of the snapshot matrix as opposed to the spatial covariance (i.e. $R = X^T X \in \mathbb{R}^{mxm}$,[4] where m is the number of snapshots). The spatial POD modes can be recovered with the eigenvalues, eigenvectors and the original snapshot matrix (see Taira et al. (2017) for the details). Since the number of snapshots is usually much smaller than the number of spatial degrees of freedom, the resulting eigenvalue problem is less computationally expensive and (when neglecting round-off errors) returns exactly the same set of eigenvectors and values for a reduced cost.

The aforementioned methods to calculate the POD are also connected to a method called singular value decomposition (SVD), which can be viewed as a more general eigenvalue decomposition also applicable to rectangular matrices (Taira et al., 2017). The SVD of the snapshot matrix is defined as in equation 1.5.

$$X = \Phi \Sigma \Psi^T \tag{1.5}$$

where the matrices Φ and Ψ correspond to the left and right singular vectors respectively and the diagonal matrix Σ contains the so-called singular values. The left singular vectors are analogous to the eigenvectors of the spatial correlation matrix and the right singular vectors are analogous to the eigenvectors of the temporal correlation matrix previously discussed. The squares of the singular values correspond to the eigenvalues found using the eigenvalue decomposition of both the correlation matrices. This makes it possible to directly perform the SVD on the snapshot matrix to find the POD modes and their associated temporal behaviour.

Although these modes can be found at a reduced cost, one disadvantage is still present when the number of snapshots is high. Part of the problem stems from the inability of the SVD to be computed in parallel. This disadvantage can be reduced by using incremental SVD (Sarwar et al., 2002). Incremental SVD allows for a calculation of the POD modes in part of the domain, which can then be used to calculate the rest of these modes in chunks in the rest of the domain by a *fold-in* technique. These calculations can be spread out over multiple processors, allowing for shorter computation times.

It should be noted that the POD requires two modes with similar energy and a phase shift of 180° to capture convecting structures fully due to the orthonormal spatial constraint. This can, for example, be seen when Serpieri and Kotsonis (2016) use the POD to identify convecting structures in a transitional boundary

---

[3] The authors of the review mention that this formulation assumes an equally spaced grid. When the velocity is defined on a non-uniform grid, the formulation should be augmented with volumetric weights of the grid points to simulate the integration in the inner product that stands at the base of this correlation matrix.

[4] The footnote that was placed by the spatial correlation matrix holds here too, i.e. to simulate the inner product, this formulation should be augmented with volumetric weights on a non-uniform grid.

layer. Thus, when interpreting data using the POD, this should be taken into account.

Finally, through the maximization goal discussed in the beginning of this section, POD modes are optimal in representing the time-averaged square of the velocity field. This implies that the modes found optimally represent the mean energy of the flowfield, which is disadvantageous if the phenomenon under investigation is only present on a small part of the domain, or is dominated by low energy structures.

1.1.2. Dynamic Mode Decomposition

Dynamic Mode Decomposition (DMD) is a technique introduced by Schmid (2010). DMD modes are eigenmodes with their corresponding eigenvalues of the best-fit linear operator that advance the snapshots one step in time (Taira et al., 2017). Several formulations to find the DMD modes are available. See Tu et al. (2014) for a comparison of several algorithms and the resulting DMD modes for a few sample problems. Regardless of the algorithm chosen to compute the modes, the result is a set of spatial modes with corresponding (possibly complex) eigenvalues. These eigenvalues can be related to the dynamic behaviour of the spatial mode, e.g. whether the mode is growing or decaying in time.

This final property makes the technique advantageous in the context of boundary layer stability problems. DMD results in modes oscillating at one frequency with a growth rate. As seen in section 1.2, this is similar to the general ansatz used in the temporal approach of stability theory.

Furthermore, as Rowley et al. (2009) showed, the DMD can provide an approximation of the behaviour of the Koopman operator. The Koopman operator is an infinite dimensional linear operator that represents the nonlinear dynamics of a system. Therefore, investigating the eigenmodes and eigenvalues of the leading terms in this linear operator can give insight into the dynamic behaviour of even a nonlinear system.

These properties make DMD a good candidate for investigating the behaviour of perturbations in a boundary layer. However, several disadvantages still exist. Nonlinear behaviour can only be captured if care is taken in the snapshot selection. Furthermore, the modes found do not provide any selection criterion as with POD. This is disadvantageous, especially when considering interpretation of large datasets (Taira et al., 2017).

1.1.3. Spectral POD

Spectral POD (SPOD) can refer to two methods. The method recently introduced by Sieber et al. (2016) uses a band-pass filter on the temporal covariance of the snapshot matrix before performing the POD. By changing the width of the filter, the method shifts between being analogous to a regular POD and a Discrete Fourier Transform (DFT). This way, the most advantageous properties of both methods can be combined.

Another method bearing the name Spectral POD was already introduced in the previously discussed paper by Lumley (1967). According to Towne et al. (2018) this method has been largely overlooked due to the popularity of the spatial POD and the extra computational expense that was not available in the early days of the POD. One of the disadvantages of the spatial POD is that it finds modes optimal in representing the spatial correlation matrix. This has as a consequence that the resulting modes might not evolve coherently in time. In other words, a ‘coherent’ flow structure might be divided over several POD modes. SPOD finds modes that optimally represent the two-point correlation tensor in both space and time, thus alleviating (part of) this problem. The method performs the POD on snapshot matrices that are constructed from the Fourier components at one particular frequency. At each frequency, the result is a set of orthonormal modes.

Towne et al. (2018) connected DMD with SPOD. The authors conclude that SPOD modes are DMD modes that optimally represent the average flow statistics. This connection seems to be most relevant in turbulent flows, which will not be the focus of this work. Since there will also not be stochastic noise due to measurement equipment, SPOD and DMD modes will result in very similar structures and therefore are treated as if they share the same advantages and disadvantages.

1.1.4. Goal-Oriented Reduced-Order Modeling

As mentioned before, one disadvantage of the POD is that the average energy content might not be the best norm to use to describe a certain phenomenon of interest (for example, growth of an instability). Therefore, finding modes that minimize an error in some quantity of interest (QoI) could provide results that better represent the dynamics of a particular phenomenon.

One of the techniques developed using this line of reasoning can be found in Bui-Thanh et al. (2007). Instead of solving the optimization problem directly (as is done with the POD), the authors first define a minimization problem of the error in the QoI with two constraints. The first constraint forces the resulting modes to be orthonormal, similar to the POD modes. The second is a constraint referred to as a ‘model constraint’. This constraint projects the governing equations on the resulting modes and forces them to satisfy the obtained reduced-order representation of the governing equations.

The above formulation also has some issues as pointed out by Cheng et al. (2015). First of all, the governing equations used for the model constraint are defined for large linear time-invariant systems of the form: $M\dot{u} + Ku = F$, where $u(t) \in \mathbb{R}^n$ is the state vector and $M \in \mathbb{R}^{nxn}$, $K \in \mathbb{R}^{nxn}$ and $F \in \mathbb{R}^n$ are discrete representations of the model. This representation requires these matrices to be able to be constructed, which is difficult when considering experimental data or when the user is interested in using alternative boundary conditions for a ROM. Finally, extending this formulations to nonlinear PDEs and possibly a nonlinear goal function is not possible.

For this reason, Cheng et al. (2015) introduced the Semi-Continuous Formulation (SCF) of the GOROM approach. The dataset is still defined at discrete points, but the ROM and the optimization problem are defined within a continuous framework.

A deficiency in both goal-oriented formulations is that the optimality of a set of goal-oriented modes is found by performing numerical optimization. As Bui-Thanh et al. (2007) mentions, the objective functional surface is not necessarily convex and, depending on the initial guess, the numerical optimization might converge to a local minimum. Therefore, choosing a suitable initial guess is important. Assuming that the resulting GOROM modes are close to the POD modes, Bui-Thanh et al. (2007) propose two strategies:

1. Use POD modes as the initial guess.
2. Find the GOROM modes for the problem with $m-1$ modes and add the m^{th} POD mode to this set of modes for the initial guess of the minimization for m GOROM modes. This approach can then be iterated until required number of modes m is large enough.

These two approaches are applied by Cheng et al. (2015) for both linear and nonlinear models and goal functions. Depending on the goal function and the model used, the authors indeed find different optimal solutions underlining the need for a suitable initial guess.

The major advantage of the goal-oriented framework is connected to the model constraint. The resulting modes are better at representing the QoI, which can be some quantity in a small part of the domain, but also are forced to adhere to the reduced governing equations in the rest of the domain. This is shown by Bui-Thanh et al. (2007) to result in larger errors in representing the flowfield outside of the domain of interest compared to POD, but smaller errors within the domain of interest. The addition of this model constraint allows focusing the attention on this small and more interesting part of the domain without completely ignoring the behaviour of the modes in the rest of the domain.

The addition of the model constraint does increase the computational cost, however. At each step in the minimization problem, the reduced model constructed using the current modes needs to be integrated over the domain. Fortunately, this process can quite easily be computed in parallel, similar to how regular finite-element models are parallelized.

The final advantage for this formulation stems from the flexibility of the minimization formulation. For example, the model constraint can be removed completely or only applied partially to reduce computational cost. By selecting the norm of the entire velocity field as the goal function and removing the model constraint, the POD minimization can also be recovered. If desired, adding additional dynamical constraints is also a possibility.

Finally, Borggaard et al. (2016) altered the formulation by Bui-Thanh et al. (2007) by introducing goal-oriented POD (GOPOD). The authors note that the optimization is performed on the goal function integrated over the full time interval, even though some of the relevant dynamics might only be present on a small time interval. Finding so-called *global modes* can result in smoothing out these short-term important structures.

With this in mind, the authors propose performing POD on the dataset and fixing the resulting modes as global modes. The residual of the flow field in time can be calculated using the temporal evolution of the POD modes. This temporal residual field is then divided into smaller time intervals and a minimization problem for finding goal-oriented modes is solved in each time interval on the residual field. The resulting flow field can be reconstructed using the global POD modes and the short-term goal-oriented structures.

The GOPOD formulation does have some other distinct advantages over classical GOROM. For example, the method doesn't require the integration of the projected model at each iteration step, greatly reducing the cost of constructing the model. A downside however is that this GOPOD formulation can not be directly extrapolated to the SCF as proposed by Cheng et al. (2015) as estimating the effect of the residual on the representation of the QoI assumes a linear goal function. Nevertheless, the general principle of defining global modes before performing the optimization on a smaller time interval is at least a possible extension.

1.1.5. Projection-based Reduced-Order Modeling Comparison

A very broad comparison and summary between the different techniques can be found in table 1.1.

Table 1.1: Overview of the Pros and cons for different projection based methods.

Method	**Pros**	**Cons**
Classical POD (spatial, snapshots and SVD based)	POD basis vectors are orthonormal. Tractable to compute, even for a very high-dimensional dataset.	Temporal correlation and higher order spatial correlations are ignored. POD modes are selected and ordered based on the energy of the mode.
DMD/SPOD	DMD gives specific dynamic structures with just one frequency and one growth rate. If the data is chosen properly, DMD can give an approximation of the Koopman operator.	The dynamic relevance of an individual mode is more difficult to determine. The modes found by DMD are not necessarily orthogonal, which complicates constructing a ROM.
Goal-Oriented ROM	Modes can be found to represent a quantity of interest, which could lead to modes that are dynamically more important. Formulation of the minimization problem is flexible to adding or removing constraints	Constructing the optimised modes is computationally expensive. Dynamically important structures that appear for short periods of time might get smoothed out due to time integration.

1.2. Boundary layer stability

Reynolds (1883) performed an experiment with a dye streak introduced in a glass tube with flowing water. The author found that the streak remained laminar when the velocity was sufficiently low. Somewhere in the pipe the streak started to break down and mix the dye with the remaining water in the cross section once the velocity reached a critical value however. This phenomenon is called laminar-to-turbulent transition and has remained a subject of investigation ever since those early experiments.

The result has been the emergence of a field called *hydrodynamic stability*. The methods applied in this field model the effect of small disturbances on the stability of the laminar flow solution to the Navier-Stokes equations (Schmid & Henningson, 2001). The different processes that a small perturbation can go through to cause transition is presented in a graphical (yet simplified) manner in figure 1.1 .

As seen from figure 1.1, all external disturbances first go through a receptivity process. Generally speaking, the receptivity process describes excitation of boundary layer instabilities through external perturbations (e.g. sound waves or freestream turbulence), see Saric et al. (2002) for a review. Although the eventual transition behaviour is dependent on this phenomenon, for the remainder of this work, it is assumed that all perturbations have gone through this process and thus that the relevant boundary layer instabilities are already present.

After the external environmental perturbations have been introduced into the boundary layer through receptivity, several paths can be followed leading to turbulent breakdown. In general, the path the perturbation follows moves further to the right in figure 1.1 with higher amplitudes of the freestream perturbations. During cruise flight of an aircraft, the freestream turbulence is generally low, which can be used to justify limiting the research to path A in figure 1.1 (Reed et al., 1996). This has an advantage in that the reasonably low-cost and simple linear stability theory (LST) can in some cases predict the local growth rate of the primary modes accurately. This theory is further discussed in section 1.2.1. However, as Saric et al. (2002) pointed out: when not much is known about the receptivity process leading up to the specific disturbances investigated, care should be taken in deriving causal relations between two seemingly similar environments.

Following path A, the block called ‘Primary Modes’ is encountered first. These primary modes grow until they saturate and a new base-flow profile is reached. With this new profile, additional perturbations can

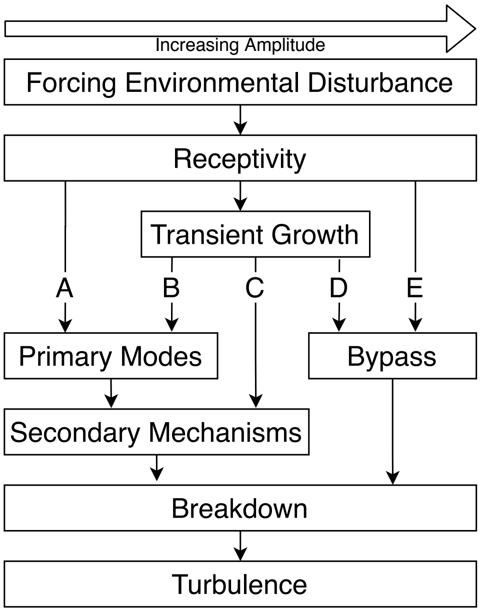


Figure 1.1: Simplified visual representation of the processes that a perturbation can go through to cause transition in a boundary layer, redrawn from Saric et al. (2002).

become unstable and grow (Herbert, 1988). These perturbations eventually lead to turbulent breakdown and are called *secondary instabilities* which close path A in figure 1.1.

Alternatively, some perturbations might first go through *transient growth* (path B through D in figure 1.1). This is a mechanism in which some perturbations might decay when $t \to \infty$, but temporarily show growth on shorter time intervals. Transient growth mechanisms might lead to faster growth of primary instabilities, appearance of secondary instabilities or something called *bypass transition*. Bypass transition is a general term for the observation that, for some cases, the boundary layer transitions without first passing through a growth mechanism. This historical view of bypass transition is under pressure of more research in transient growth mechanisms (see Reshotko (2001) for more details). The main focus of this work is on path A and therefore, these mechanisms are not treated in more detail.

The remainder of this section is structures as follows: firstly, subsection 1.2.1 treats the general approach to model problems constrained to path A in figure 1.1. The next subsection then discusses a particular type of instability that can be found from applying these models to a 2D boundary layer. This section is then concluded with a brief discussion of a field of research that has gained recent interest. This research is focused on including a step in the boundary layer to better understand what happens to the stability of the boundary layer when a manufacturing imperfection is introduced. In what follows, unless otherwise specified, the procedure for deriving the stability equations follows Schmid and Henningson (2001).

1.2.1. Stability Theory

The general approach used by stability theory is to divide the flow into a base flow and a total flow. The base flow is a laminar flow solution to the Navier-Stokes equations of which the components are denoted as follows: (U_b^i, P_b). The total flow is equally a solution to the Navier-Stokes equations, however, it is assumed a perturbation is added to the state. This total flowfield is then assumed a linear combination of the base flow and a perturbation, i.e. $(U_b^i + u^i, P_b + p)$, where a lowercase letter denotes the perturbation. Note that the base and total flow are both a solution to the Navier-Stokes equations (albeit with slightly different initial and/or boundary conditions). However, the perturbation flow is not necessarily a solution of the Navier-Stokes equation.

Using this notation, the total flow can be substituted into the incompressible Navier-Stokes equations. When just the base flow is substituted into the same Navier-Stokes equations and subtracted from the substitution of the total flow, the nonlinear disturbance equations are obtained as seen in equation 1.6 and equation 1.7.

$$\frac{\partial u^i}{\partial t} = -U_b^j \frac{\partial u^i}{\partial x^j} - u^j \frac{\partial U_b^i}{\partial x^j} - \frac{\partial p}{\partial x^i} + \frac{1}{Re}\nabla^2 u^i - u^j \frac{\partial u^i}{\partial x^j} \tag{1.6}$$

$$\frac{\partial u^i}{\partial x^i} = 0 \tag{1.7}$$

where the Einstein summation convention over repeated indices is used.[5] These equations can then further be simplified based on the case under consideration. When assuming a parallel base flow (that is: $\mathbf{U_b} = \left[U_b(y), 0, 0\right]$) and neglecting the $u^j \frac{\partial u^i}{\partial x^j}$ term, the Orr-Sommerfeld (OS) equations can be recovered. These can be solved by inserting the following ansatz in the equations:

$$q(x, y, z, t) = \hat{q}(y)e^{i(\alpha x + \beta z - \omega t)} + c.c. \tag{1.8}$$

where q denotes a component of the perturbation field, α the spatial wavelength, β the spanwise wavelength, ω the temporal frequency and $\hat{q}$ the shape function of the particular perturbation. As the equations are not as relevant for this work, the interested reader can find a more thorough discussion in appendix A.

Alternatively it is possible to account for a slowly varying base flow, increasing the fidelity of the model. This results in the parabolized stability equations (PSE), which can be found in appendix A.3. A discussion of a nonlinear version of the PSE equations is also present in appendix A.3. This nonlinear PSE (NPSE) also accounts for the nonlinear $u^j \frac{\partial u^i}{\partial x^j}$ term.

1.2.2. Tollmien-Schlichting Problem

Tollmien (1929) and Schlichting (1933) first managed to solve the OS equations for a boundary layer on a flat plate. A typical shape function that results from following their analysis can be found in figure 1.2. In honor of the work done by Tollmien and Schlichting, these structures are called *Tollmien-Schlichting Waves* and are characterized by two maxima of the streamwise perturbation with a phase shift at the location of the maximum perturbation in y. When inserting the ansatz defined in equation 1.8 into the continuity equation for u and v, it can be concluded that the streamwise perturbation shape function $\hat{u}$ needs to go to zero at the location of the maximum value for $\hat{v}$. The sign of the real part of the shape function above this y-coordinate is flipped with respect to below this y-coordinate through the same line of reasoning. Because of the wave-like nature of the perturbation, this is treated as a 180° shift in phase. A few extra notes related to TS waves that are not as relevant for the remainder of this work can be found appendix A.4.

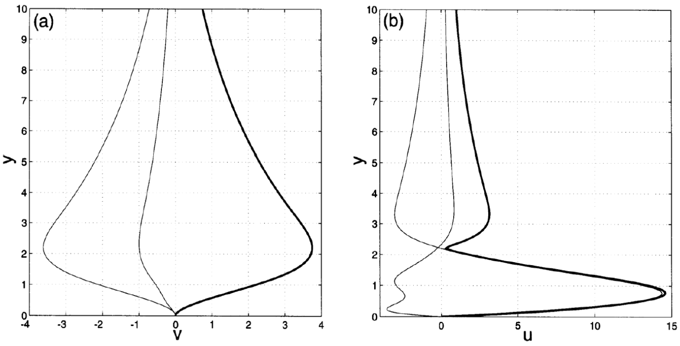


Figure 1.2: Typical shape of the v component (a) and u component (b) of a TS wave. The thin lines are the imaginary and real parts of the shape function. Taken from Schmid and Henningson (2001).

---

[5] That is for example: $u^j \frac{\partial u^i}{\partial x^j} = \sum_{i=1}^{3} u^j \frac{\partial u^i}{\partial x^j}$.

1.2.3. Boundary layers with steps

The above mentioned research has (mostly) been focused on boundary layers with smooth surfaces. On a wing, manufacturing imperfections can lead to small gaps or steps on the surface of the wing. The aerodynamics group at Delft University of Technology is actively researching the effect of a forward facing step on the growth of perturbations. This subsection does not give an overview of the current findings in literature, but merely show some of the challenges the established methods have in analysing these complex flow problems.

Figure 1.3 shows the streamwise component of the base flow over a step (Shahzad, 2020) for a 2D boundary layer. Not surprisingly, the base flow exerts a considerably larger wall-normal component around the step than in the rest of the domain, as the flow has to convect over the step. This in itself already raises concerns in trying to apply the OS equations, PSE and NPSE for analysing the flow near the step, as the OS equations assume no wall-normal component and PSE and NPSE both assume a slowly-varying wall-normal component. Furthermore, as seen from figure 1.3, the base flow adjusts to the step already before the step is actually reached, insinuating the existense of elliptic effects. This begs the question of whether a parabolized assumption for determining the perturbations is valid, or if these perturbations themselves also show signs of ellipticity around the step.

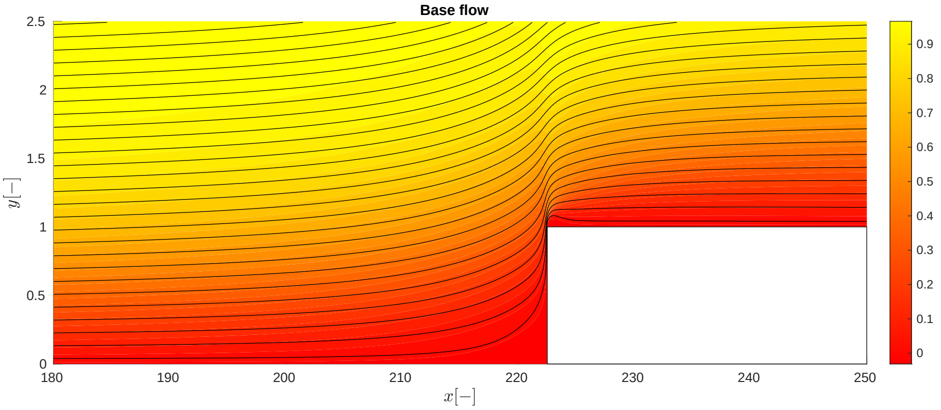


Figure 1.3: Streamwise velocity of the base flow of the highest step considered from Shahzad (2020) along with isolines of the streamfunction.

With these notes in mind, it is highly likely that only DNS can be used (from a computational modeling perspective) to be able to accurately investigate the physics around the step, although current research is being conducted to see if NPSE also provides accurate results. DNS data provides an additional challenge however, as it is very high-dimensional and complex.

As discussed in section 1.1, the GOROM approach is specifically designed to find a low-order representation of the system that focuses on representing a certain goal function as best as possible. If this physical interaction of the step can be translated to a goal function, the GOROM approach might be able to find a few modes representing the physics using the DNS data. These optimal modes can then be used when trying to interpret the physical interaction the step has with the perturbation field. Seeing as GOROM has not been widely applied, at least not in the field of boundary-layer stability theory with a forward facing step, this method might prove useful in interpreting this complex, nonlinear and possibly elliptic behaviour of stability problems. The work done in this thesis will therefore bridge some of the gap between these complex stability problems and this advanced mode identification technique. Currently however, the GOROM technique has only been formalized for the 1D advection and Burger's equation Cheng et al. (2015), 1D incompressible Navier-Stokes equations Shajahan (2016) and the 2D Stokes equations Cheng (2017), all with homogeneous boundary conditions. The formulation should thus first be extended to the multi-dimensional, incompressible Navier-Stokes equations with inhomogeneous boundary conditions.

1.3. Research Objectives and Questions

With the above discussion in mind, the research objectives can be formalized. The research objectives of this work are:

1. Verify the applicability of the goal-oriented, reduced-order modeling technique to the multidimensional, incompressible Navier-Stokes equations by solving the optimization problem for cases using manufactured solutions.
2. Demonstrate the applicability of the goal-oriented, reduced-order modeling technique for practical problems by solving the optimization problem for a 2D transitional boundary layer with and without a forward-facing step.
3. Investigate the computational cost of solving the goal-oriented optimization problem.
   (a) Investigate the parallel scaling of the goal-oriented, reduced-order modeling technique.
   (b) Investigate the effect of the number of basis functions on the computational cost.

The questions this work aims to answer are:

1. How do the modes produced by the goal-oriented, reduced-order modeling technique compare to the modes produced by the proper orthogonal decomposition for a 2D transitional boundary layer on a flat plate?
   (a) Does the GOROM technique result in a basis that is numerically more accurate in a ROM than a basis constructed using the POD?
   (b) Can the GOROM modes help in interpreting the underlying physics of the flow problem?
2. How do the modes produced by the goal-oriented, reduced-order modeling technique compare to the modes produced by the proper orthogonal decomposition for a 2D transitional boundary layer on a flat plate with a forward-facing step?
   (a) Does the GOROM technique result in a basis that is numerically more accurate in a ROM than a basis constructed using the POD?
   (b) Can the GOROM modes help in interpreting the underlying physics of the flow problem?
3. How does the parallel implementation of the goal-oriented, reduced-order modeling technique scale against the number of processors used?
4. How is the computational cost affected by the choice of the number of basis functions used?

The remainder of this work is structured as follows. Chapter 2 treats the general formulation of the GOROM technique as specified in literature (Cheng et al., 2015). Chapter 3 applies the GOROM technique to the general-dimensional, incompressible Navier-Stokes equations with inhomogeneous boundary conditions. Chapter 4 and chapter 5 apply the GOROM technique to a 2D, transitional boundary layer in the absence (chapter 4) and presence (chapter 5) of a forward-facing step. Chapter 6 then discusses the stong parallel scaling and computational cost of the problem with the step present. And finally, the report is concluded in chapter 7 and recommendations for future work are presented.

2

Semi-Continuous Formulation of Goal-Oriented Reduced-Order Modeling

This chapter will treat the semi-continuous formulation (SCF) of the goal-oriented reduced-order modeling (GOROM) technique as specified in literature (Cheng, 2017; Cheng et al., 2015). The derivation of the terms as seen here is more extensively treated in appendix B and as such, only the results and motivation behind the method are discussed in this chapter. The chapter is structured as follows: firstly, the problem the GOROM technique solves is introduced by means of the model equations. Following the model equations, the optimization problem will be formulated, after which the approach to solving the optimization problem is discussed. The succeeding section treats the numerical optimization technique used in this work and finally, the concept of secondary basis functions will be introduced.

2.1. Model Equations

The GOROM technique assumes that the reference data satisfies the following equations:

$$\mathcal{L}(u) = f \tag{2.1}$$

$$u(0) = u_0 \tag{2.2}$$

where $\mathcal{L}$ is a (possibly nonlinear) differential operator, f is a source term and u_0 denotes the initial conditions. Appropriate boundary conditions should also be specified to close the model. To construct a ROM of the above differential equations, the following separation of variables is utilized:

$$\hat{u} = \sum_{i=1}^{m} \alpha_i(t)\phi_i(\mathbf{x}) \tag{2.3}$$

where ϕ denotes a mode, α a temporal amplitude, m the number of modes employed to solve the system of equations and $\hat{u}$ the approximation of the solution. Galerkin projection can then be used to create a weak form of the differential equations (equations 2.1 and 2.2), leading to:

$$\int_\Omega \phi_k \left(\mathcal{L}(\hat{u}) - f\right) d\Omega = 0 \tag{2.4}$$

$$\int_\Omega \phi_k \left(\hat{u}_0 - u_0\right) d\Omega = 0 \tag{2.5}$$

When the separation of variables employed in equation 2.3 is substituted in equation 2.4, a system of m ordinary differential equations in time emerge for the amplitudes α. These can then consequently be discretized further and solved by marching in time. The resulting ROM solution can be reconstructed by recombining the modes and amplitudes using equation 2.3. This reconstruction can be used to estimate the fitness of the set of modes for a given goal function, which can consequently be optimized. This is the subject of the next section.

2.2. Formulation of the Optimization Problem

The goal of the GOROM technique is to represent the L_2 norm of some (possibly nonlinear) goal function $g(\hat{u})$ optimally. Furthermore, the technique penalizes deviations from an orthonormal set of basis functions, that is: [1]

$$\min_{\phi,\alpha} \mathscr{G} = \int_\Omega \left(\frac{1}{2}\int_0^{t_f} \left(g-\hat{g}\right)^2 dt\right) d\Omega + \frac{\beta}{2}\sum_{i,j=1}^{m} (\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega)^2 \tag{2.6}$$

The choice of α in equation 2.6 is not independent of the choice of ϕ. That is, the optimization problem is constrained to solely use values for α that satisfy the model equations in equations 2.4 and 2.5. Solving minimization problems with such external constraints can be problematic for standard numerical optimization techniques. Therefore, the constrained minimization problem specified by equation 2.4 can be converted to an unconstrained one by using the ideas of calculus of variations (see e.g. the textbook by Lanczos (1952)).

The optimal functions α and ϕ that minimize equation 2.6 *and* satisfy equations 2.4 & 2.5 are also the functions that minimize the following functional:

$$\min_{\phi,\alpha} \mathscr{L} = \int_\Omega \left(\frac{1}{2}\int_0^{t_f} E dt + \sum_{k=1}^{m}\int_0^{t_f} \lambda_k G_2^k dt + \sum_{k=1}^{m} \mu_k G_3^k\right) d\Omega + \frac{\beta}{2} G_1 \tag{2.7}$$

with:

$$E = \left(g-\hat{g}\right)^2 \tag{2.8}$$

$$G_1 = \sum_{i,j=1}^{m}\left(\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega\right)^2 \tag{2.9}$$

$$G_2^k = \phi_k\left(\mathcal{L}\left(\hat{u}\right) - f\right) \tag{2.10}$$

$$G_3^k = \phi_k\left(\hat{u}_0 - u_0\right) \tag{2.11}$$

where the Lagrange multipliers are defined by $\lambda(t) \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^m$ and β is a regularization parameter. Solving this optimization problem is the subject of the next section.

2.3. Gradient Calculation

A given set of modes satisfies the optimality of equation 2.7 if the following three conditions are met:

1. $\delta\mathscr{L}_{\lambda_q} = 0$ & $\delta\mathscr{L}_{\mu_q} = 0$
2. $\delta\mathscr{L}_{\alpha_q} = 0$
3. $\delta\mathscr{L}_{\phi_q} = 0$

The first of these three conditions recovers a set of equations referred to as the *state equations*. When performing the corresponding functional derivatives, the model equations (equation 2.10) are recovered for $\delta\mathscr{L}_{\lambda_q} = 0$ and the initial conditions (equation 2.11) for $\delta\mathscr{L}_{\mu_q} = 0$. When evaluating $\delta\mathscr{L}_{\alpha_q} = 0$, the adjoint equations are recovered along with final conditions on λ and μ, which can be found in equations 2.12-2.14.

$$\int_\Omega \left[\frac{1}{2}\frac{\partial E}{\partial \alpha_q} + \sum_{k=1}^{m} \lambda_k \frac{\partial G_2^k}{\partial \alpha_q} - \frac{d}{dt}\left(\sum_{k=1}^{m} \lambda_k \frac{\partial G_2^k}{\partial \dot{\alpha}_k}\right)\right] d\Omega = 0 \tag{2.12}$$

$$\lambda(t_f) = 0 \tag{2.13}$$

$$\mu = \lambda(0) \tag{2.14}$$

Finally, when evaluating $\delta\mathscr{L}_{\phi_q} = 0$, the gradient expression can be found to be:

$$\int_0^{t_f} \left[\frac{1}{2}\frac{\partial E}{\partial \phi_q} + \sum_{i=1}^{m} \lambda_i \frac{\partial G_2^i}{\partial \phi_q} - \frac{d}{dx}\left(\sum_{i=1}^{m} \lambda_i \frac{\partial G_2^i}{\partial \phi_{q_x}}\right)\right] dt + \sum_{i=1}^{m} \mu_i \frac{\partial G_3^i}{\partial \phi_q} + 2\beta \sum_{i=1}^{m}\left(\int_\Omega \phi_i\phi_q d\Omega - \delta_{iq}\right)\phi_i = 0 \tag{2.15}$$

---

[1] Note that $\hat{g} = g(\hat{u})$ is the approximation of the goal using equation 2.3 and $g = g(u)$ is the goal evaluated using reference data. In the following, this shortened notation for the goal will be used as well.

In practise, these equations are resolved using a segregated approach. This approach is summarized in figure 2.1. That is, initially a guess is made on the optimal modes. The state equation (equation 2.10 and 2.11) can be resolved for the temporal amplitudes α with this first guess of modes. Using both the modes and state amplitudes, the adjoint equations (equation 2.12-2.14) can be resolved for the adjoint amplitudes λ and μ. With the values for α, λ and μ found, the gradient can consequently be calculated. This gradient is not necessarily zero for a given guess of optimal modes, however, it can be used to update the current guess of the modes with a numerical optimization technique. This procedure can then be iterated until a set of modes is found that does satisfy all the required conditions.

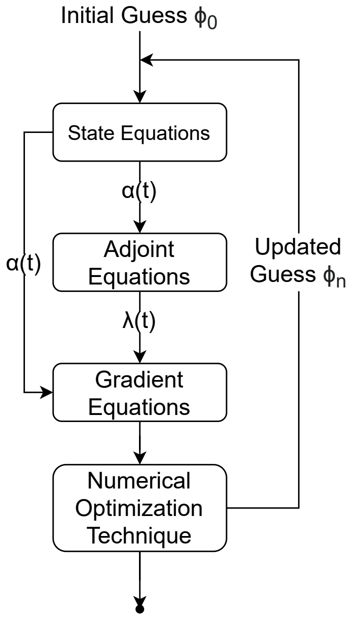


Figure 2.1: Basic Flow chart of the procedure followed to solve the optimization procedure.

2.4. Optimization Technique

An extensive amount of literature is available on solving unconstrained minimization problems like equation 2.7. The optimization technique used in previous applications is a trust-region method (Bui-Thanh et al., 2007; Cheng, 2017; Cheng et al., 2015). According to Cheng (2017) a trust-region method is chosen over a line-search algorithm since trust-region algorithms are more efficient with a close-to-singular Hessian matrix. For this reason this work will use this optimization technique as well.

The trust-region algorithm assumes that the shape of the objective function can be approximated using a quadratic model (referred to here as m_k) within a certain circular trust region around the current guess of the optimal coefficients. The guess of the optimal coefficients will then be updated in the next iteration by setting a step toward the combination of coefficients that minimize m_k within the trust region. As it is unknown if the quadratic model is a good approximation of the objective function within the trust region, this step is first referred to as the *candidate* step p_k. To optimize the value of the objective function, a subproblem now first needs to be solved. That is, the minimum of m_k needs to be found within the trust region. In other words:

$$\min_{p_k} m_k(p_k) = f(x_k) + p_k^T \nabla f_k + \frac{1}{2} p_k^T \nabla^2 f(x_k) p_k \quad \text{with } ||p_k|| < \Delta \tag{2.16}$$

where Δ refers to the trust-region size, f refers to the objective function and x_k to the current guess of the optimization coefficients. Finding the expression for p_k that minimizes this problem is rather straightforward. That is, the candidate step that minimizes m_k can be found by solving the Newton equations:

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k) \tag{2.17}$$

Although solving this equation exactly is possible through inverting the Hessian, the process can be quite costly for larger optimization problems. For this reason, *inexact-Newton methods* are used to resolve the

Newton equation approximately. That is, the candidate step p_k is found by iteratively minimizing the residual until the residual is below a certain tolerance, that is:

$$||-\nabla f(x_k) - \nabla^2 f(x_k) p_k|| < \theta_k \tag{2.18}$$

where θ_k is a small tolerance. To achieve this, the conjugate-gradient method by Steihaug (1983) is used. The algorithm can be found in appendix C. Once the candidate step p_k is found for a given trust region, it is still unknown whether m_k is a good representation of f_k at the step p_k. By defining the ratio ρ_k to be:

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} \tag{2.19}$$

the approximation of the objective function made by m_k can be tested. That is, if the approximation made by m_k is good, the value for ρ_k should be close to 1. Similarly, if the approximation is poor, the value of ρ_k is close to, or even below 0.

If the trust region is small enough to provide a good approximation of the objective function, it might be desirable to increase the size of the trust region. After all, the larger the trust region, the larger the step size can be and consequently the more progress is made within one iteration. On the other hand, if the approximation is poor, the trust region should reduce in size to increase the accuracy of m_k. Throughout this work, the following strategy is employed:

$$\Delta_{k+1} = \begin{cases} \frac{1}{4}\Delta_k, & \text{if } \rho_k < 0.25 \\ 2\Delta_k, & \text{if } \rho_k > 0.75 \\ \Delta_k, & \text{else} \end{cases} \tag{2.20}$$

The algorithm summarizing these steps can be found in appendix C. Do note that maximum and minimum trust-region sizes are also defined in the algorithms in appendix C. A convergence criteria is connected to the minimum trust-region size as well. That is, if the algorithm is not making any notable progress on the coefficients, the optimization is terminated.

It is undesirable and not necessary to compute the Hessian matrix directly in the case of the conjugate-gradient method. Instead the product of the Hessian and the candidate step that arises in equation 2.18 is approximated using:

$$\nabla^2 f(x_k) p_k \approx \frac{\nabla f(x_k + \epsilon p_k) - \nabla f(x_k)}{\epsilon} \tag{2.21}$$

where ϵ is a small value. With the approximation of the Hessian defined it is only required to be able to calculate the value of the objective function and its gradient given a set of optimization coefficients.

2.5. Secondary Basis Function

Throughout this work, the primary basis ϕ will be constructed by using a linear combination of so-called secondary bases ψ. This will take form of:

$$\phi_i(x) = \sum_{j=1}^{M} c_{ij} \psi_j(x) \tag{2.22}$$

The secondary basis ψ will be functions known prior to the optimization. This shifts the overall formulation of the optimization from finding the optimal functions ϕ to finding the optimal coefficients c_{ij} that minimize the norm of the specific goal function.

Some flexibility is also present in the choice of these secondary basis functions. That is, these basis functions can be chosen *globally* or *locally*. Global secondary basis functions are, as the name suggest, functions defined on the entire domain Ω. Local secondary basis functions, on the other hand, are defined only within one element in a mesh, similar to how finite-element models (FEM) use local shape functions with corresponding degrees of freedom (DOF) to numerically approximate differential equations.

In the analogy between local secondary basis functions and a FEM, the DOFs from the FEM correspond to the optimization coefficients of GOROM. This highlights one of the problems with the local secondary basis functions. Although it provides more freedom with regards to shape of the modes, the number of optimization coefficients does increase considerably.

Finally, do note that Cheng (2017) defines a third way to construct the primary basis functions. This is framed within a discussion of what the author calls *P-type* and *F-type* primary basis functions. The *F-type* primary basis functions use a linear combination of known secondary basis functions as discussed within this section. These thus include both the local and global secondary basis functions, which will be referred to in this work as *local F-type* basis functions and *global F-type* basis functions respectively.

The *P-type* primary basis functions define the primary basis functions as a vector, where each entry denotes the value of the function at a spatial location. The *P-type* primary basis function can also be seen as continuous function multiplied by a series of delta functions centred at the spatial locations. The three methods are summarized in figure 2.2.

The *P-type* primary basis functions are clearly very similar to the *local F-type* basis functions and share the same problems related to the size of the optimization vector. The *P-type* primary basis functions do allow a comparison with the initial discrete formulation of GOROM (Bui-Thanh et al., 2007), see Cheng et al. (2015) and Cheng (2017) for this comparison. This work will however exclusively use the *global F-type* basis functions.

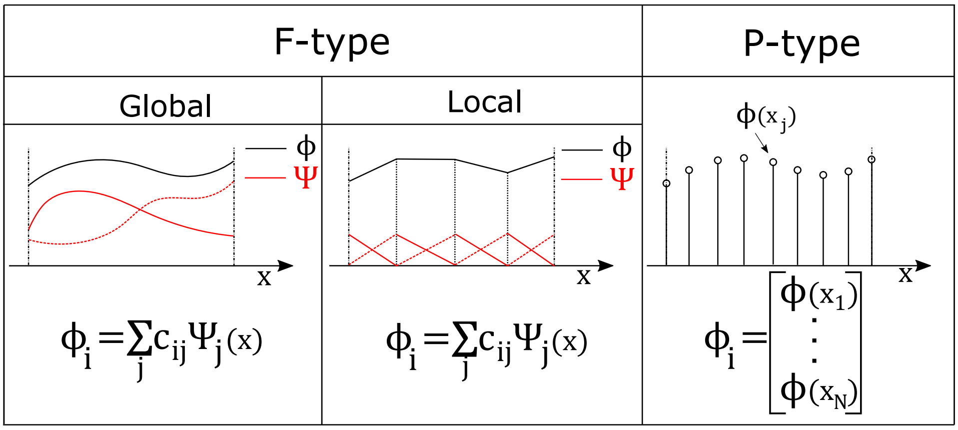


Figure 2.2: Visual representation of the choice of types of basis functions.

3

Goal-Oriented Reduced-Order Modeling of 3D Incompressible Navier-Stokes

This chapter will formalize the SCF of the GOROM technique for the general-dimensional, incompressible Navier-Stokes equations with inhomogeneous boundary conditions. This chapter is structured as follows: firstly the choice of basis for the multi-variable formulation of GOROM is discussed. The subsequent section discusses the model constraint with the choice of basis present. The section after then formalizes the optimization problem, after which the state, adjoint and gradient expressions are constructed. The section afterwards then discusses possible goal functions to use and finally, the verification of the gradient expression and numerical techniques is presented.

3.1. Choice of Basis

In multivariable problems for the GOROM there is a choice in the approach to the separation of variables and the secondary basis functions. These discussed in this section starting with the separation of variables and then treating the secondary basis.

3.1.1. Separation of Variables

As in the general formulation specified in chapter 2, the approximation of the solution u to the model is represented using the following separation of variables:

$$u = \sum_{i}^{N} \alpha_i(t)\phi_i(\mathbf{x}) \tag{3.1}$$

The 3D incompressible Navier-Stokes equations contain four free variables, i.e. 3 velocity components and pressure. In this multi-variable setting, several choices to achieve the decomposition in equation 3.1 are available. In the context of GOROM, Cheng (2017) identified the *scalar-valued* and *vector-valued* approaches. The scalar-valued approach allows each variable to have different temporal behaviour. That is, each variable follows the same separation of variables seen previously in equation 3.1, fully written out in equations 3.2 and 3.3. Note that the superscripts here denote indices rather than powers. This notation will be utilized throughout this work to be able generalize the equations in arbitrary dimensions.

$$u^r = \sum_{i}^{N} \alpha_i^r(t)\phi_i^r(\mathbf{x}) \tag{3.2}$$

$$p = \sum_{i}^{N} \alpha_i^p(t)\phi_i^p(\mathbf{x}) \tag{3.3}$$

where $r = [1,2,3]$ denotes the direction of the velocity component.

Alternatively, the *vector-valued* approach assumes the same temporal behaviour of one mode for all variables. That is, the separation of variables is achieved by means of equation 3.4.

$$\begin{bmatrix} u \\ v \\ w \\ p \end{bmatrix} = \sum_{i}^{N} \alpha_i(t) \begin{bmatrix} \phi_i^u \\ \phi_i^v \\ \phi_i^w \\ \phi_i^p \end{bmatrix} \tag{3.4}$$

This work will utilize the *scalar-valued* primary basis functions. Using this approach does increase the size of the optimization problem. However, choosing the vector-valued approach does lose flexibility, as modes for all variables are forced to have the same temporal behaviour, which might not be the case in practical situations.

3.1.2. Secondary Basis

In multiple dimensions, Cheng (2017) again identifies two possible approaches very similar to the separation of variables discussed in the previous section to express the primary basis functions in terms of known secondary basis functions.

The *scalar-valued* secondary basis functions can be seen in equations 3.5 and 3.6.

$$\phi_i^r = \sum_{j=1}^{M} c_{ij}^r \psi_j^r \tag{3.5}$$

$$\phi_i^p = \sum_{j=1}^{M} c_{ij}^p \psi_j^p \tag{3.6}$$

where $r = [1,2,3]$ denotes the spatial direction again. The alternative *vector-valued* secondary basis functions can be found in equation 3.7.

$$\begin{bmatrix} \phi_i^u \\ \phi_i^v \\ \phi_i^w \\ \phi_i^p \end{bmatrix} = \sum_{j=1}^{M} c_{ij} \begin{bmatrix} \psi_j^u \\ \psi_j^v \\ \psi_j^w \\ \psi_j^p \end{bmatrix} \tag{3.7}$$

Clearly, the size of the optimization problem is reduced when the vector-valued approach is chosen. However, some flexibility will be lost. This work will utilize the scalar-valued approach for the secondary basis functions. The added flexibility of representing the primary modes with different coefficients is deemed more relevant than reducing the computational cost.

3.2. Model Constraint

As seen in chapter 2, the resulting modes will be constrained to adhere to a ROM of the governing equations. In the context of the problems in this work, the governing equations are the non-dimensionalized, incompressible Navier-Stokes equations in conservation form. That is:

$$\dot{u} + \nabla \cdot \mathcal{F}_{inv}^x(\boldsymbol{u}) - \frac{1}{Re} \nabla \cdot \mathcal{F}_{visc}^x(\boldsymbol{u}) + \frac{\partial p}{\partial x} = f^x \tag{3.8}$$

$$\dot{v} + \nabla \cdot \mathcal{F}_{inv}^y(\boldsymbol{u}) - \frac{1}{Re} \nabla \cdot \mathcal{F}_{visc}^y(\boldsymbol{u}) + \frac{\partial p}{\partial y} = f^y \tag{3.9}$$

$$\dot{w} + \nabla \cdot \mathcal{F}_{inv}^z(\boldsymbol{u}) - \frac{1}{Re} \nabla \cdot \mathcal{F}_{visc}^z(\boldsymbol{u}) + \frac{\partial p}{\partial z} = f^z \tag{3.10}$$

$$\nabla \cdot \boldsymbol{u} = f^c \tag{3.11}$$

with initial conditions: $\boldsymbol{u}_0 = [u_0, v_0, w_0]^T$ and p_0. A dot above the variable denotes a derivative with respect to time, $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]^T$, $\boldsymbol{u} = [u, v, w]^T$, p the pressure and the superscript x, y, z of the denotes the spatial direction of the momentum equation, $\mathcal{F}_{inv}^x(\boldsymbol{u}) = u\boldsymbol{u}$, $\mathcal{F}_{inv}^y(\boldsymbol{u}) = v\boldsymbol{u}$ and $\mathcal{F}_{inv}^z(\boldsymbol{u}) = w\boldsymbol{u}$ and $\mathcal{F}_{visc} = \nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T$.

To construct a ROM of the governing equations, a Bubnov-Galerkin method is used to project the equations onto weighting functions. Furthermore, integration by parts is used to lower the order of the derivative. The resulting weak form of the governing equations can be solved for the amplitudes α. Before the resulting expressions are shown, both the sub-grid scale model to stabilize the equations and the boundary conditions to close the problem are discussed.

3.2.1. Sub-Grid Scale Model

A stabilization method is required to be able to solve the ROM using the resulting GOROM modes. This work will use the Variational-Multiscale Method (VMM), introduced by Hughes et al. (2001). The authors divide the solution to the governing equations up into the resolved and unresolved scales by means of equation 3.12 and 3.13.

$$u^r = \hat{u}^r + u'^r \tag{3.12}$$

$$p = \hat{p} + p' \tag{3.13}$$

where $\hat{u}$ and $\hat{p}$ are the resolved scales, and u' and p' are the unresolved scales in this context. The unresolved scales are usually modelled through a Sub-Grid Scale (SGS) model. The model used in this work makes use of the estimations for the unresolved velocity and pressure as seen in equations 3.14 and 3.15.

$$u'^r = -\tau_m R^r_m \tag{3.14}$$

$$p' = -\tau_c R_c \tag{3.15}$$

where τ_m and τ_c are two stabilization parameters and:

$$R^r_m = \dot{\hat{u}}^r + \sum_{j=1}^{N_{dim}} \frac{\partial(\hat{u}^r \hat{u}^j)}{\partial x^j} - \frac{1}{Re} \sum_{j=1}^{N_{dim}} \frac{\partial}{\partial x^j} \left(\frac{\partial \hat{u}^r}{\partial x^j} + \frac{\partial \hat{u}^j}{\partial x^r} \right) + \frac{\partial \hat{p}}{\partial x^r} - f^r \tag{3.16}$$

$$R_c = \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^j} - f^c \tag{3.17}$$

are the residuals of the momentum equations and continuity equation. To determine τ_m and τ_c, expressions based on Taylor et al. (1997) will be used. These can be found in equations 3.18 and 3.19 in this work.

$$\tau_m = \frac{C}{\sqrt{c_1 \hat{u}^i G^{ij} \hat{u}^j + c_2 \nu^2 G^{ij} G^{ij} + \frac{c_3}{\Delta t^2}}} \tag{3.18}$$

$$\tau_c = \frac{\sqrt{\hat{u}^i G^{ij} \hat{u}^j}}{G^{ii}} \tag{3.19}$$

where the summation over repeated indeces is implied, ν represents the viscosity and G is the metric covariant tensor. The values used for the coefficients are: $C = 1$, $c_1 = 1$, $c_2 = 36$ and $c_3 = 4$. Finally, the unresolved viscous terms, the unresolved time derivative terms and all unresolved scales on the boundary will be assumed zero from here on out, greatly simplifying the final expression.

3.2.2. Boundary Conditions

Although it would be possible to attempt to reproduce the boundary conditions applied in the production of the reference dataset in a form suitable for the ROM, instead a much simpler option is opted for. For all applications in this work, high-resolution reference data will be available through DNS data or a manufactured solution. For this reason, it is possible to apply Dirichlet-type boundary conditions for the velocity on all boundaries, as the exact solution is known.

One way of achieving this goal is by enforcing these conditions strongly. That is, replacing equations for certain DOFs in the overall system of equations for equations that force the sum of all modes to be equal to the values specified on the boundaries, similar to how strong Dirichlet boundary conditions are applied in a regular FEM. Enforcing these conditions strongly does create a problem that the model will lose accuracy in the domain in favour of accuracy on the boundary.

The alternative is to enforce these conditions using a weak boundary condition. That is, the difference between the solution to the ROM and the reference data on the boundary is not enforced to be zero, but is penalized for being larger than zero. The added integral present on the boundary can be found in equation 3.20.

$$B_4^r = \int_\Gamma \sigma^r (\hat{u}^r - u_{bc}^r) d\Gamma \tag{3.20}$$

where σ^r is a controllable penalty parameter and u_{bc}^r is the boundary condition in direction r and Γ denotes the boundary. Since the weak method allows for much more flexibility in trading-off accuracy of the solution at the boundary with accuracy of the solution in the interior of the domain through the penalty parameter σ, it is chosen to apply the boundary conditions through adding equation 3.20.

3.2.3. Final Model Constraint

For the sake of brevity of the main body of work, the above steps are performed in appendix D for the interested reader. The final model constraint can be summarized in the following equations:

$$(\phi_k^r, \dot{\hat{u}}^r)_\Omega + B_1^r - \frac{1}{Re} B_2^r + B_3^r + B_4^r - D_1^r + \frac{1}{Re} D_2^r - D_3^r - (\phi_k^r, f^r)_\Omega = 0 \tag{3.21}$$

$$D_4 - \left(\phi_k^p, f^c\right)_\Omega = 0 \tag{3.22}$$

$$(\phi_k^r, \hat{u}^r - u_0^r)_\Omega = 0 \tag{3.23}$$

$$(\phi_k^p, \hat{p}_0 - p_0)_\Omega = 0 \tag{3.24}$$

where $r = [1,2,3]$ denotes the direction of the momentum equation[1], equations 3.23 and 3.24 are the Galerkin-projected initial conditions, the boundary terms B can be found in equations 3.25-3.28

$$B_1^r = \int_\Gamma \phi_k^r \hat{u}^r \sum_{j=1}^{N_{dim}} \hat{u}^j n^j d\Gamma \tag{3.25}$$

$$B_2^r = \int_\Gamma \phi_k^r \sum_{j=1}^{N_{dim}} \left(\frac{\partial \hat{u}^r}{\partial x^j} + \frac{\partial \hat{u}^j}{\partial x^r} \right) n^j d\Gamma \tag{3.26}$$

$$B_3^r = \int_\Gamma \phi_k^r \hat{p} n^r d\Gamma \tag{3.27}$$

$$B_4^r = \int_\Gamma \phi_k^r \sigma^r (\hat{u}^r - u_{bc}^r) d\Gamma \tag{3.28}$$

and the domain integrals D can be found in equations 3.29-3.32.

$$D_1^r = \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} (\hat{u}^r - \tau_m R_m^r)(\hat{u}^j - \tau_m R_m^j) d\Omega \tag{3.29}$$

$$D_2^r = \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} \left(\frac{\partial \hat{u}^r}{\partial x^j} + \frac{\partial \hat{u}^j}{\partial x^r} \right) d\Omega \tag{3.30}$$

$$D_3^r = \int_\Omega \frac{\partial \phi_k^r}{\partial x^r} \hat{p} d\Omega - \int_\Omega \frac{\partial \phi_k^r}{\partial x^r} \tau_c R_c d\Omega \tag{3.31}$$

$$D_4 = \int_\Omega \phi_k^p \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^j} d\Omega + \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^p}{\partial x^j} \tau_m R_m^j d\Omega \tag{3.32}$$

---

[1] i.e. 1 denotes the x-direction, 2 the y-direction and 3 the z-direction

3.3. Lagrangian Formulation

Using the definitions of the model constraint in section 3.2.3, the optimization problem can be constructed:

$$\min_{\phi^r,\phi^p,\alpha^r,\alpha^p} \mathscr{G} = \frac{1}{2}\sum_{r=1}^{N_{dim}} \int_\Omega \int_0^{t_f} \left(g^r - \hat{g}^r\right)^2 dt d\Omega + \frac{1}{2}\int_\Omega \int_0^{t_f} \left(g^p - \hat{g}^p\right)^2 dt d\Omega \tag{3.33}$$

$$+\frac{\beta}{2}\left[\sum_{r=1}^{N_{dim}} \sum_{i,j=1}^{m} (\delta_{ij} - \int_\Omega \phi_i^r \phi_j^r d\Omega)^2 + \sum_{i,j=1}^{m} (\delta_{ij} - \int_\Omega \phi_i^p \phi_j^p d\Omega)^2\right] \tag{3.34}$$

where β is a regularization parameter and g is the goal function. This minimization problem is subject to:

$$M_k^r = (\phi_k^r, \dot{\hat{u}}^r)_\Omega + B_1^r - \frac{1}{Re}B_2^r + B_3^r + B_4^r - D_1^r + \frac{1}{Re}D_2^r - D_3^r - (\phi_k^r, f^r)_\Omega = 0 \tag{3.35}$$

$$C_k = D_4 - \left(\phi_k^p, f^c\right)_\Omega = 0 \tag{3.36}$$

$$G_2^r = (\phi_k^r, \hat{u}^r - u_0^r)_\Omega = 0 \tag{3.37}$$

$$G_2^p = (\phi_k^p, \hat{p}_0 - p_0)_\Omega = 0 \tag{3.38}$$

where in practise N_{dim} is either 1,2 or 3. The definition for R_m^r and R_c can be found in equations 3.16 and 3.17 respectively. The constrained minimization problem is converted to an unconstrained minimization problem through the use of calculus of variations to construct the Lagrangian seen in equation 3.39.

$$\mathscr{L} = \frac{1}{2}E + \frac{\beta}{2}G_1 + \int_0^{t_f} \sum_{k=1}^{m} \left(\lambda_k^p C_k + \sum_{r=1}^{N_{dim}} \lambda_k^r M_k^r\right) dt + \sum_{k=1}^{m} \left(\mu_k^p G_2^p + \sum_{r=1}^{N_{dim}} \mu_k^r G_2^r\right) \tag{3.39}$$

with:

$$E = \sum_{r=1}^{N_{dim}} \int_\Omega \int_0^{t_f} \left(g^r - \hat{g}^r\right)^2 dt d\Omega + \int_\Omega \int_0^{t_f} \left(g^p - \hat{g}^p\right)^2 dt d\Omega \tag{3.40}$$

$$G_1 = \sum_{r=1}^{N_{dim}} \sum_{i,j=1}^{m} (\delta_{ij} - \int_\Omega \phi_i^r \phi_j^r d\Omega)^2 + \sum_{i,j=1}^{m} (\delta_{ij} - \int_\Omega \phi_i^p \phi_j^p d\Omega)^2 \tag{3.41}$$

Do note that the goal function can be different for each variable when using this formulation. That is, the goal function can be written as a vector:

$$\hat{g} = [g(\hat{u}), g(\hat{v}), g(\hat{w}), g(\hat{p})]^T \tag{3.42}$$

Alternatively, the formulation can be simplified by considering only one function (e.g. $\hat{g} = \hat{u}$). This loses flexibility, as no trade-off can be made between representing different variables. On top of the loss of flexibility, it is possible to reproduce the one function version with the vector-valued version of the goal by zeroing out all other components of the goal vector. Thus, the vector-valued version can be seen as a more general approach.

3.4. State Equations

To recover the state equation, a small variation in the adjoint variable $\lambda_q^r(t)$ is applied:

$$\lambda_q^r = \lambda_q^{r*} + \epsilon\eta(t) \tag{3.43}$$

where λ_q^{r*} denotes the value for λ_q^r that minimizes equation 3.39. Calculus of variations teaches that in order for λ_q^{r*} to be an optimum, the Lagrangian (equation 3.39) evaluated at $\epsilon = 0$ needs to be a stationary point, which means:

$$\left.\frac{d\mathscr{L}}{d\epsilon}\right|_{\epsilon=0} = 0 \tag{3.44}$$

When filling the variation in equation 3.43 into equation 3.39, differentiating according to equation 3.44 and forcing the variation defined by $\eta(t)$ to be an arbitrary function, the momentum part of the ROM from equation 3.35 is found. Similarly, when a small variation is added to λ_q^p, μ_q^r and μ_q^p and the same line of reasoning is applied, the continuity ROM (equation 3.36), the initial conditions on the velocity (equation 3.37) and the initial conditions on the pressure (equation 3.38) are the result respectively. Hence, the state equations are equivalent to solving the ROM for the state variables α^r and α^p. To march the state equations in time, use will be made of the second-order generalized-α time-marching method introduced by Jansen et al. (2000).

3.5. Adjoint Equations

The expression in equation 3.39 includes terms for the unresolved velocity and pressure of the state equations. It was found that these are not required in the adjoint to produce an accurate gradient. Therefore, the terms related to the unresolved velocity and pressure will not be taken into account in what follows.

A similar process used to recover the state equations is used to find the adjoint equations by applying a variation on one of the temporal coefficients α^d or α^p. Doing so results in the adjoint equations, the final conditions on λ and the conditions on μ. To shorten the notation of the expression, the adjoint variables are defined in the following two equations:

$$\hat{v}^r = \sum_{k=1}^{N} \lambda_k^r \phi_k^r \tag{3.45}$$

$$\hat{v}^p = \sum_{k=1}^{N} \lambda_k^p \phi_k^p \tag{3.46}$$

Note that the solution to the state equations is denoted with a u and the solution to the adjoint equations is denoted with a v, not to be confused with the y-component of the state velocity. When the procedure of applying a variation on the temporal amplitudes is followed, the following expressions are recovered:

$$\begin{aligned}
-\int_\Omega \phi_q^d \dot{\hat{v}}^d \, d\Omega - \int_\Omega \phi_q^d \sum_{r=1}^{N_{dim}} \hat{u}^r \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) d\Omega + \frac{1}{Re} \int_\Omega \sum_{r=1}^{N_{dim}} \frac{\partial \phi_q^d}{\partial x^r} \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) d\Omega + \int_\Omega \frac{\partial \phi_q^d}{\partial x^d} \hat{v}^p \, d\Omega \\
+ \int_\Gamma \phi_q^d \sum_{r=1}^{N_{dim}} \left(\hat{v}^d \hat{u}^r n^r + n^d \hat{v}^r \hat{u}^r \right) d\Gamma - \frac{1}{Re} \int_\Gamma \sum_{r=1}^{N_{dim}} \frac{\partial \phi_q^d}{\partial x^r} \left(\hat{v}^d n^r + \hat{v}^r n^d \right) d\Gamma + \int_\Gamma \phi_q^d \sigma^d \hat{v}^d \, d\Gamma \\
= \int_\Omega \phi_q^d (g^d - \hat{g}^d) \frac{\partial \hat{g}^d}{\partial \hat{u}^d} d\Omega
\end{aligned} \tag{3.47}$$

$$\int_\Gamma \phi_q^p \sum_{r=1}^{N_{dim}} \hat{v}^r n^r \, d\Gamma - \int_\Omega \phi_q^p \sum_{r=1}^{N_{dim}} \frac{\partial \hat{v}^r}{\partial x^r} d\Omega = \int_\Omega \phi_q^p (g^p - \hat{g}^p) \frac{\partial \hat{g}^p}{\partial \hat{p}} d\Omega \tag{3.48}$$

3.5.1. Adjoint Boundary Conditions

The expressions in equations 3.47 and 3.48 represent the integrated-by-parts weak form of the adjoint equations. To define the boundary conditions and the strong residuals for the stabilization method, the original strong form of the equations needs to be recovered. To do so, the viscous and pressure domain integrals from equation 3.47 need to be integrated by parts again to swap the derivative of the test functions to the adjoint variables. The following expression can then be recovered:

$$\begin{aligned}
\int_\Omega \phi_q^d \left[-\dot{\hat{v}}^d - \sum_{r=1}^{N_{dim}} \frac{\partial \hat{v}^d}{\partial x^r} \hat{u}^r - \sum_{r=1}^{N_{dim}} \frac{\partial \hat{v}^r}{\partial x^d} \hat{u}^r - \frac{1}{Re} \sum_{r=1}^{N_{dim}} \frac{\partial}{\partial x^r} \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) - \frac{\partial \hat{v}^p}{\partial x^d} \right] d\Omega \\
+ \int_\Gamma \phi_q^d \left[\sum_{r=1}^{N_{dim}} \hat{v}^d \hat{u}^r n^r + \sum_{r=1}^{N_{dim}} \hat{v}^r \hat{u}^r n^d + \frac{1}{Re} \sum_{r=1}^{N_{dim}} \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) n^r + \hat{v}^p n^d + \sigma^d \hat{v}^d \right] d\Gamma \\
- \sum_{r=1}^{N_{dim}} \int_\Gamma \frac{\partial \phi_q^d}{\partial x^r} \frac{1}{Re} \left[\hat{v}^d n^r + \hat{v}^r n^d \right] d\Gamma = \int_\Omega \phi_q^d (g^d - \hat{g}^d) \frac{\partial \hat{g}^d}{\partial \hat{u}^d} d\Omega
\end{aligned} \tag{3.49}$$

The strong form of the adjoint PDE is then recovered by arguing that the weight functions (in this case ϕ_q^d) are arbitrary in the domain. Thus leading to the conclusion that, for the interior solution, the following needs to hold:

$$-\dot{\hat{v}}^d - \sum_{r=1}^{N_{dim}} \frac{\partial \hat{v}^d}{\partial x^r} \hat{u}^r - \sum_{r=1}^{N_{dim}} \frac{\partial \hat{v}^r}{\partial x^d} \hat{u}^r - \frac{1}{Re} \sum_{r=1}^{N_{dim}} \frac{\partial}{\partial x^r} \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) - \frac{\partial \hat{v}^p}{\partial x^d} = (g^d - \hat{g}^d) \frac{\partial \hat{g}^d}{\partial \hat{u}^d} \tag{3.50}$$

$$-\sum_{r=1}^{N_{dim}} \frac{\partial \hat{v}^r}{\partial x^r} = (g^p - \hat{g}^p) \frac{\partial \hat{g}^p}{\partial \hat{p}} \tag{3.51}$$

This set of PDEs requires appropriate boundary conditions such that the boundary integrals in equations 3.49 and 3.48 amass to zero. Each boundary requires four conditions (in the case of 3D flow). The first can be recovered from equation 3.48. Similar to the value in the domain, the weighting function (ϕ_q^d) on the boundary is arbitrary. It can thus be concluded that, on the boundary, the following needs to hold:

$$\sum_{r=1}^{N_{dim}} \hat{v}^r n^r = 0 \tag{3.52}$$

This condition on adjoint variables is applied by adding the contributions of the boundary integral as specified in equation 3.48 to the system of equations. Alternatively, it is also possible to integrate the domain term of the weak form of equation 3.51 by parts. This results in a natural boundary condition in which the value from equation 3.52 can be substituted.

The three other conditions that close the adjoint problem are specified by the boundary integrals from the momentum equations in equation 3.49. It is clear that, similar to the previously discussed condition, the boundary integrals need to evaluate to zero. That is:

$$\begin{aligned} &\int_\Gamma \phi_q^d \left[\sum_{r=1}^{N_{dim}} \hat{v}^d \hat{u}^r n^r + \sum_{r=1}^{N_{dim}} \hat{v}^r \hat{u}^r n^d + \frac{1}{Re} \sum_{r=1}^{N_{dim}} \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) n^r + \hat{v}^p n^d + \sigma^d \hat{v}^d \right] d\Gamma \\ &\qquad - \sum_{r=1}^{N_{dim}} \int_\Gamma \frac{\partial \phi_q^d}{\partial x^r} \frac{1}{Re} \left[\hat{v}^d n^r + \hat{v}^r n^d \right] d\Gamma = 0 \end{aligned} \tag{3.53}$$

An equivalent 'strong' boundary conditions as in equation 3.52 can not be recovered due to the presence of both the weighting function and its derivative in the boundary integral. However, the condition in equation 3.53 can be rewritten to:

$$\begin{aligned} &\int_\Gamma \phi_q^d \left[\sum_{r=1}^{N_{dim}} \hat{v}^d \hat{u}^r n^r + \sum_{r=1}^{N_{dim}} \hat{v}^r \hat{u}^r n^d + \hat{v}^p n^d + \sigma^d \hat{v}^d \right] d\Gamma - \sum_{r=1}^{N_{dim}} \int_\Gamma \frac{\partial \phi_q^d}{\partial x^r} \frac{1}{Re} \left[\hat{v}^d n^r + \hat{v}^r n^d \right] d\Gamma \\ &\qquad = - \int_\Gamma \phi_q^d \left[\frac{1}{Re} \sum_{r=1}^{N_{dim}} \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) n^r \right] d\Gamma \end{aligned} \tag{3.54}$$

When deriving the weak form of equation 3.50, the viscous term can be integrated by parts. The resulting weak form of the equations is:

$$\text{[Domain Integrals]} - \int_\Gamma \phi_q^d \left[\frac{1}{Re} \sum_{r=1}^{N_{dim}} \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d} \right) n^r \right] d\Gamma = 0 \tag{3.55}$$

Where, for the sake of brevity, the domain integrals are not completely written out. Finally, the boundary conditions on the adjoint problem can be applied by realizing that the boundary integral in equation 3.55 can be replaced by the left-hand side of equation 3.54.

3.5.2. Stabilization

Similar to the primal problem, the adjoint requires stabilization as well. The method to achieve this stabilization is the same as used by the primal problem, i.e. the variational-multiscale method. That is, the adjoint variables consists out of a resolved and unresolved scales:

$$v^d = \hat{v}^d + v'^d \tag{3.56}$$

$$v^p = \hat{v}^p + v'^p \tag{3.57}$$

The unresolved scales are modelled using the strong residual again, i.e.:

$$v'^d = -\tau_m \mathcal{R}_m^d \tag{3.58}$$

$$v'^p = -\tau_c \mathcal{R}_c \tag{3.59}$$

The definitions of τ_m and τ_c can be found in equations 3.18 and 3.19. The strong residuals can be derived from equations 3.50 and 3.51 to be:

$$\mathcal{R}_m^d = -\dot{\hat{v}}^d - \sum_{j=1}^{N_{dim}} \frac{\partial \hat{v}^d}{\partial x^j}\hat{u}^j - \sum_{j=1}^{N_{dim}} \frac{\partial \hat{v}^j}{\partial x^d}\hat{u}^j - \frac{1}{Re}\sum_{j=1}^{N_{dim}} \frac{\partial}{\partial x^j}\left(\frac{\partial \hat{v}^d}{\partial x^j} + \frac{\partial \hat{v}^j}{\partial x^d}\right) - \frac{\partial \hat{v}^p}{\partial x^d} - (g^d - \hat{g}^d)\frac{\partial \hat{g}^d}{\partial \hat{u}^d} \tag{3.60}$$

$$\mathcal{R}_c = -\sum_{j=1}^{N_{dim}} \frac{\partial \hat{v}^j}{\partial x^j} - (g^p - \hat{g}^p)\frac{\partial \hat{g}^p}{\partial \hat{p}} \tag{3.61}$$

Do note that the unresolved scales of the time derivative and viscous terms are omitted again.

3.5.3. Final Adjoint Equations

Applying the operations discussed in the previous two subsections, results in the final adjoint equations in 3.62 and 3.63.

$$-\int_\Omega \phi_q^d \dot{\hat{v}}^d d\Omega - \mathcal{D}_1^d + \frac{1}{Re}\mathcal{D}_2^d - \mathcal{D}_3^d + \mathcal{B}_1^r - \frac{1}{Re}\mathcal{B}_2^r + \mathcal{B}_3^d + \mathcal{B}_4^d = \int_\Omega \phi_q^d (g^d - \hat{g}^d)\frac{\partial \hat{g}^d}{\partial \hat{u}^d} d\Omega \tag{3.62}$$

$$\mathcal{B}_5 - \mathcal{D}_4 = \int_\Omega \phi_q^p (g^p - \hat{g}^p)\frac{\partial \hat{g}^p}{\partial \hat{p}} d\Omega \tag{3.63}$$

where the definitions of the domain integrals $\mathcal{D}$ can be found in equations 3.64-3.67 and the definitions of the boundary integrals $\mathcal{B}$ can be found in equations 3.68-3.72.

$$\begin{aligned}\mathcal{D}_1^d = &\int_\Omega \phi_q^d \sum_{r=1}^{N_{dim}} \hat{u}^r \left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d}\right) d\Omega + \int_\Omega \sum_{r=1}^{N_{dim}} \frac{\partial \phi_q^d}{\partial x^r}\hat{u}^r \tau_m \mathcal{R}_m^d d\Omega + \int_\Omega \frac{\partial \phi_q^d}{\partial x^d}\sum_{r=1}^{N_{dim}} \hat{u}^r \tau_m \mathcal{R}_m^r d\Omega \\ &+ \int_\Omega \phi_q^d \sum_{r=1}^{N_{dim}} \tau_m \mathcal{R}_m^d \frac{\partial \hat{u}^r}{\partial x^r} d\Omega + \int_\Omega \phi_q^d \sum_{r=1}^{N_{dim}} \tau_m \mathcal{R}_m^r \frac{\partial \hat{u}^r}{\partial x^d} d\Omega\end{aligned} \tag{3.64}$$

$$\mathcal{D}_2^d = \int_\Omega \sum_{r=1}^{N_{dim}} \frac{\partial \phi_q^d}{\partial x^r}\left(\frac{\partial \hat{v}^d}{\partial x^r} + \frac{\partial \hat{v}^r}{\partial x^d}\right) d\Omega \tag{3.65}$$

$$\mathcal{D}_3^d = \int_\Omega \phi_q^d \frac{\partial \hat{v}^p}{\partial x^d} d\Omega + \int_\Omega \frac{\partial \phi_q^d}{\partial x^d}\tau_c \mathcal{R}_c d\Omega \tag{3.66}$$

$$\mathcal{D}_4 = \int_\Omega \phi_q^p \sum_{r=1}^{N_{dim}} \frac{\partial \hat{v}^r}{\partial x^r} d\Omega + \int_\Omega \sum_{r=1}^{N_{dim}} \frac{\partial \phi_q^p}{\partial x^r}\tau_m \mathcal{R}_m^r d\Omega \tag{3.67}$$

$$\mathcal{B}_1^d = \int_\Gamma \phi_q^d \sum_{r=1}^{N_{dim}} \left(\hat{v}^d \hat{u}^r n^r + n^d \hat{v}^r \hat{u}^r\right) d\Gamma \tag{3.68}$$

$$\mathcal{B}_2^d = \int_\Gamma \sum_{r=1}^{N_{dim}} \frac{\partial \phi_q^d}{\partial x^r}\left(\hat{v}^d n^r + \hat{v}^r n^d\right) d\Gamma \tag{3.69}$$

$$\mathcal{B}_3^d = \int_\Gamma \phi_q^d \hat{v}^p n^d d\Gamma \tag{3.70}$$

$$\mathcal{B}_4^d = \int_\Gamma \phi_q^d \sigma^d \hat{v}^d d\Gamma \tag{3.71}$$

$$\mathcal{B}_5 = \int_\Gamma \phi_q^p \sum_{r=1}^{N_{dim}} \hat{v}^r n^r d\Gamma \tag{3.72}$$

The final conditions on the adjoint variables are:

$$\lambda^d(t_f) = 0 \tag{3.73}$$

$$\lambda^p(t_f) = 0 \tag{3.74}$$

$$\mu_q^d = \lambda^d(0) \tag{3.75}$$

$$\mu_q^p = \lambda^p(0) \tag{3.76}$$

As final conditions are specified rather than initial conditions, the set of PDEs is marched backward in time. The marching technique used is the generalized-α timemarch, which is the same marching technique used as is used in the state equation.

3.6. Gradient Equations

With the definitions of the state and adjoint treated, the formulation can be finalized by considering the gradient. As the primary modes will be formulated through a linear combination of secondary modes, the gradient will be determined with respect to the velocity coefficients c^d_{qs} and c^p_{qs}.

In the case of the adjoint, it is sufficient to neglect the effect of the stabilization terms in the primal problem, simplifying the derivation of the final expression of the adjoint equations. Unfortunately this is not the case when considering the gradient. The derivatives related to $\frac{\partial \tau_m}{\partial c^d_{qs}}$ are still neglected however. This results in the rather lengthy expression seen in equation 3.77.

$$
\begin{aligned}
\frac{\partial \mathscr{L}}{\partial c^d_{qs}} = {} & [\text{Goal Function}] + [\text{Continuity Equation}] + [\text{Time Derivative}] + [\text{Convective Domain Terms}] \\
& + [\text{Viscous Domain Terms}] + [\text{Pressure Domain Terms}] + [\text{Momentum Source}] \\
& + [\text{Convective Boundary Terms}] + [\text{Viscous Boundary Terms}] + [\text{Pressure Boundary Terms}] \\
& + [\text{Initial Conditions}] + [\text{Orthonormality Constraints}] + [\text{Stabilization Terms}] \\
& + [\text{Stabilization Derivative Terms}]
\end{aligned} \tag{3.77}
$$

with:

$$[\text{Goal Function}] = -\int_0^{t_f} \int_\Omega \alpha^d_q \psi^d_s (\mathrm{g}^d - \hat{\mathrm{g}}^d) \frac{\partial \hat{\mathrm{g}}^d}{\partial \hat{u}^d} d\Omega dt \tag{3.78}$$

$$[\text{Continuity Equation}] = \int_0^{t_f} \int_\Omega \alpha^d_q \frac{\partial \Psi^d_s}{\partial x^d} \sum_{k=1}^{N} \lambda^p_k \phi^p_k d\Omega dt \tag{3.79}$$

$$[\text{Time Derivative}] = \int_0^{t_f} \int_\Omega \left[\lambda^d_q \Psi^d_s \dot{\hat{u}}^d + \dot{\alpha}^d_q \Psi^d_s \sum_{k=1}^{N} \lambda^d_k \phi^d_k \right] d\Omega dt \tag{3.80}$$

$$
\begin{aligned}
[\text{Convective Domain Terms}] = -\int_0^{t_f} \int_\Omega \Bigg[& \lambda^d_q \sum_{j=1}^{N_{dim}} \frac{\partial \Psi^d_s}{\partial x^j} \hat{u}^d \hat{u}^j + \alpha^d_q \Psi^d_s \sum_{j=1}^{N_{dim}} \hat{u}^j \sum_{k=1}^{N} \lambda^d_k \frac{\partial \phi^d_k}{\partial x^j} \\
& + \alpha^d_q \Psi^d_s \sum_{j=1}^{N_{dim}} \hat{u}^j \sum_{k=1}^{N} \lambda^j_k \frac{\partial \phi^j_k}{\partial x^d} \Bigg] d\Omega dt
\end{aligned} \tag{3.81}
$$

$$
\begin{aligned}
[\text{Viscous Domain Terms}] = \frac{1}{Re} \int_0^{t_f} \int_\Omega \Bigg[& \lambda^d_q \sum_{j=1}^{N_{dim}} \frac{\partial \Psi^d_s}{\partial x^j} \frac{\partial \hat{u}^d}{\partial x^j} + \alpha^d_q \sum_{j=1}^{N_{dim}} \frac{\partial \Psi^d_s}{\partial x^j} \sum_{k=1}^{N} \lambda^d_k \frac{\partial \phi^d_k}{\partial x^j} \\
& + \lambda^d_q \sum_{j=1}^{N_{dim}} \frac{\partial \Psi^d_s}{\partial x^j} \frac{\partial \hat{u}^j}{\partial x^d} + \alpha^d_q \sum_{j=1}^{N_{dim}} \frac{\partial \Psi^d_s}{\partial x^j} \sum_{k=1}^{N} \lambda^j_k \frac{\partial \phi^j_k}{\partial x^d} \Bigg] d\Omega dt
\end{aligned} \tag{3.82}
$$

$$[\text{Pressure Domain Term}] = -\int_0^{t_f} \lambda^d_q \int_\Omega \frac{\partial \Psi^d_s}{\partial x^d} \hat{p} d\Omega dt \tag{3.83}$$

$$[\text{Momentum Source}] = -\int_0^{t_f} \lambda^d_q \int_\Omega \Psi^d_s f^d d\Omega dt \tag{3.84}$$

$$
\begin{aligned}
[\text{Convective Boundary Terms}] = \int_0^{t_f} \int_\Gamma \Bigg[& \lambda^d_q \Psi^d_s \hat{u}^d \sum_{j=1}^{N_{dim}} \hat{u}^j n^j + \alpha^d_q \Psi^d_s \sum_{j=1}^{N_{dim}} \hat{u}^j n^j \sum_{k=1}^{N} \lambda^d_k \phi^d_k \\
& + \alpha^d_q \Psi^d_s n^d \sum_{j=1}^{N_{dim}} \hat{u}^j \sum_{k=1}^{N} \lambda^j_k \phi^j_k \Bigg] d\Gamma dt
\end{aligned} \tag{3.85}
$$

$$\begin{aligned}\text{[Viscous Boundary Terms]} = -\frac{1}{Re}\int_0^{t_f}\int_\Gamma \Bigg[& \lambda_q^d \Psi_s^d \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^d}{\partial x^j} n^j + \alpha_q^d \sum_{j=1}^{N_{dim}} \frac{\partial \Psi_s^d}{\partial x^j} n^j \sum_{k=1}^{N} \lambda_k^d \phi_k^d \\ & + \lambda_q^d \Psi_s^d \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^d} n^j + \alpha_q^d \sum_{j=1}^{N_{dim}} \frac{\partial \Psi_s^d}{\partial x^j} n^d \sum_{k=1}^{N} \lambda_k^j \phi_k^j \Bigg] d\Gamma dt \end{aligned} \tag{3.86}$$

$$\text{[Pressure Boundary Terms]} = \int_0^{t_f} \lambda_q^d \int_\Gamma \Psi_s^d \hat{p} n^d d\Gamma dt \tag{3.87}$$

$$\begin{aligned}\text{[Stabilization Terms]} = \int_0^{t_f}\int_\Omega \Bigg[& \lambda_q^d \tau_m R_m^d \sum_{j=1}^{N_{dim}} \frac{\partial \Psi_s^d}{\partial x^j} \hat{u}^j + \alpha_q^d \Psi_s^d \sum_{j=1}^{N_{dim}} \tau_m R_m^j \sum_{k=1}^{N} \lambda_k^j \frac{\partial \phi_k^j}{\partial x^d} \\ & + \alpha_q^d \Psi_s^d \sum_{j=1}^{N_{dim}} \tau_m R_m^j \sum_{k=1}^{N} \lambda_k^d \frac{\partial \phi_k^d}{\partial x^j} + \lambda_q^d \frac{\partial \Psi_s^d}{\partial x^d} \tau_c R_c \\ & + \lambda_q^d \sum_{j=1}^{N_{dim}} \frac{\partial \Psi_s^d}{\partial x^j} (\hat{u}^d - \tau_m R_m^d) \tau_m R_m^j \Bigg] d\Omega dt \end{aligned} \tag{3.88}$$

$$\begin{aligned}\text{[Stabilization Derivative Terms]} = \int_0^{t_f}\int_\Omega \Bigg[& \sum_{j=1}^{N_{dim}} \tau_m \frac{\partial R_m^j}{\partial c_{qs}^d} \sum_{k=1}^{N} \lambda_k^p \frac{\partial \phi_k^p}{\partial x^j} + \sum_{j,r=1}^{N_{dim}} \tau_m \frac{\partial R_m^r}{\partial c_{qs}^d} (\hat{u}^j - \tau_m R_m^j) \sum_{k=1}^{N} \lambda_k^r \frac{\partial \phi_k^r}{\partial x^j} \\ & + \sum_{j,r=1}^{N_{dim}} (\hat{u}^r - \tau_m R_m^r) \tau_m \frac{\partial R_m^j}{\partial c_{qs}^d} \sum_{k=1}^{N} \lambda_k^r \frac{\partial \phi_k^r}{\partial x^j} + \tau_c \frac{\partial R_c}{\partial c_{qs}^d} \sum_{j=1}^{N_{dim}} \sum_{k=1}^{N} \lambda_k^j \frac{\partial \phi_k^j}{\partial x^j} \\ & + \frac{\partial \tau_c}{\partial c_{qs}^d} R_c \sum_{j=1}^{N_{dim}} \sum_{k=1}^{N} \lambda_k^j \frac{\partial \phi_k^j}{\partial x^j} \Bigg] d\Omega dt \end{aligned} \tag{3.89}$$

$$\text{[Initial Condition]} = \int_\Omega \Bigg[\mu_q^d \Psi_s^d (\hat{u}_0^r - u_0^r) + \alpha_q^d(0) \Psi_s^d \sum_{k=1}^{N} \mu_k^d \phi_k^d \Bigg] d\Omega \tag{3.90}$$

$$\text{[Orthonormality Constraints]} = 4 \sum_{k=1}^{m} \int_\Omega \left(\phi_k^d \Psi_s^d \int_\Omega \phi_k^d \phi_q^d d\Omega - \delta_{kq} \right) d\Omega \tag{3.91}$$

The terms related to $\frac{\partial R_m^r}{\partial c_{qs}^d}$, $\frac{\partial R_c}{\partial c_{qs}^d}$ and $\frac{\partial R_m^r}{\partial c_{qs}^d}$ can be found in equations 3.92-3.94.

$$\frac{\partial R_m^r}{\partial c_{qs}^d} = \dot{\alpha}_q^d \psi_s^d + \alpha_q^d \psi_s^d \frac{\partial \hat{u}^r}{\partial x^d} + \sum_h \alpha_q^d \frac{\psi_s^d}{\partial x^h} \hat{u}^h + \sum_h \alpha_q^d \psi_s^d \frac{\partial \hat{u}^h}{\partial x^h} + \alpha_q^d \frac{\partial \psi_s^d}{\partial x^d} \hat{u}^r - \frac{1}{Re} \left(\alpha_q^d \frac{\partial \psi_s^d}{\partial x^r \partial x^d} + \sum_h \alpha_q^d \frac{\partial \psi_s^d}{\partial x^d \partial x^h} \right) \tag{3.92}$$

$$\frac{\partial R_c}{\partial c_{qs}^d} = \alpha_q^d \frac{\partial \psi_s^d}{\partial x^d} \tag{3.93}$$

$$\frac{\partial \tau_c}{\partial c_{qs}^d} = \frac{1}{\sum_i G^{ii}} \frac{1}{2\sqrt{\sum_i \sum_h \hat{u}^i G^{ih} \hat{u}^h}} \left(\alpha_q^d \psi_s^d \sum_h \left(G^{dh} + G^{hd} \right) \hat{u}^h \right) \tag{3.94}$$

Repeating the above process for the coefficients of the pressure, the following expression is recovered:

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial c^p_{qs}} = &\text{[Goal Term]} + \text{[Continuity Domain Terms]} + \text{[Continuity Source Term]} + \text{[Pressure Boundary Integral]}\\ &+\text{[Pressure Domain Term]} + \text{[Initial Condition Terms]} + \text{[Orthonormality Term]}\\ &+\text{[Stability Terms]} + \text{[Stability Derivative Terms]}\end{aligned} \tag{3.95}$$

with:

$$\text{[Goal Term]} = -\int_0^{t_f} \alpha^p_q \int_\Omega \psi^p_s (g^p - \hat{g}^p)\frac{\partial \hat{g}^p}{\partial \hat{p}} d\Omega dt \tag{3.96}$$

$$\text{[Continuity Domain Terms]} = \int_0^{t_f} \lambda^p_q \int_\Omega \Psi^p_s \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^j} d\Omega dt \tag{3.97}$$

$$\text{[Continuity Source Term]} = -\int_0^{t_f} \lambda^p_q \int_\Omega \Psi^p_s f^c d\Omega dt \tag{3.98}$$

$$\text{[Pressure Boundary Integral]} = \sum_{k=1}^{N}\sum_{r=1}^{N_{dim}} \int_0^{t_f} \lambda^r_k \int_\Gamma \phi^r_k \alpha^p_q \Psi^p_s n^r d\Gamma dt \tag{3.99}$$

$$\text{[Pressure Domain Term]} = -\sum_{k=1}^{N}\sum_{r=1}^{N_{dim}} \int_0^{t_f} \lambda^r_k \int_\Omega \frac{\partial \phi^r_k}{\partial x^r} \alpha^p_q \Psi^p_s d\Omega dt \tag{3.100}$$

$$\text{[Stability Terms]} = \int_0^{t_f} \lambda^p_q \sum_{j=1}^{N_{dim}} \int_\Omega \frac{\partial \Psi^p_s}{\partial x^j} \tau_m R^j_m d\Omega dt \tag{3.101}$$

$$\begin{aligned}\text{[Stability Derivatives Terms]} = \int_0^{t_f} \int_\Omega \Bigg[&\tau_m \alpha^p_q \sum_{j=1}^{N_{dim}} \frac{\partial \psi^d_s}{\partial x^j} \sum_{k=1}^{N} \lambda^p_k \frac{\partial \phi^p_k}{\partial x^j} + \tau_m \alpha^p_q \sum_{j,r=1}^{N_{dim}} \frac{\partial \psi^d_s}{\partial x^r}(\hat{u}_j - \tau_m R^j_m) \sum_{k=1}^{N} \lambda^r_k \frac{\partial \phi^r_k}{\partial x^j}\\ &+\tau_m \alpha^p_q \sum_{j,r=1}^{N_{dim}} (\hat{u}^r - \tau_m R^r_m) \frac{\partial \psi^d_s}{\partial x^j} \sum_{k=1}^{N} \lambda^r_k \frac{\partial \phi^r_k}{\partial x^j}\Bigg] d\Omega dt\end{aligned} \tag{3.102}$$

$$\text{[Initial Condition Terms]} = \int_\Omega \left[\mu^p_q \Psi^p_s (\hat{p} - p_0) d\Omega + \alpha^p_q(0) \Psi^p_s \sum_{k=1}^{N} \mu^p_k \phi^p_k\right] d\Omega \tag{3.103}$$

$$\text{[Orthonormality Term]} = 4\sum_{k=1}^{N} \int_\Omega \left(\phi^p_k \Psi^p_s \int_\Omega \phi^p_k \phi^p_q d\Omega - \delta_{kq}\right) d\Omega \tag{3.104}$$

3.7. Possible Goal Functions

Throughout this work, several goal functions will be used for different flow problems. These are summarized in table 3.1. The motivation behind each goal is discussed more extensively in the rest of this section.

Global U

This goal function is the simplest. When performing a POD of reference data, this is the goal which is minimized. Finding GOROM modes from a dataset using this goal thus seems analogous to finding the POD of that same dataset. However, as the temporal behaviour of the GOROM modes is described by nonlinear dynamics rather than the linear projection in the case of POD, the GOROM can in some cases provide a more optimal basis even for this goal (see e.g. the results from Cheng et al. (2015)).

Local U

This goal function is a copy of the global goal function on Ω_s, which is a part of the entire computational domain Ω. Outside of Ω_s, the goal function returns 0. This allows a representation of the goal on part of the domain that is of particular interest. Alternatively, this goal can be augmented by multiplying by a weighting function to allow for a smoother transition of the value of the goal between Ω_s to the value in the rest of the domain.

Table 3.1: Goal functions used throughout this work.

Goal Number	Goal Name	Mathematical Expression
1	Global U	$g(\hat{u}^r) = \hat{u}^r$
2	Local U	$g(\hat{u}^r) = \begin{cases} \hat{u}^r, & for\ [x,y,z] \in \Omega_s \subset \Omega \\ 0, & otherwise \end{cases}$
3	Global U^2	$g(\hat{u}^r) = \hat{u}^{r^2}$
4	Fourier amplitudes	$g(\hat{u}^r) = \begin{cases} \lvert\hat{u}^r(j,0)\rvert, & for\ [x,y,z] \in \Omega_s \subset \Omega \\ 0, & otherwise \end{cases}$
5	Real Fourier amplitudes	$g(\hat{u}^r) = \begin{cases} \hat{u}^r cos(j\omega t), & for\ [x,y,z] \in \Omega_s \subset \Omega \\ 0, & otherwise \end{cases}$
6	nonlinear interaction perturbation field	$g^r\left(\hat{u}, \frac{\partial \hat{u}}{\partial x}\right) = \begin{cases} \left(\hat{u}^j - U_b^j\right)\left(\frac{\partial \hat{u}^r}{\partial x^j} - \frac{\partial U_b^r}{\partial x^j}\right), & for\ [x,y,z] \in \Omega_s \subset \Omega \\ 0, & otherwise \end{cases}$

Global U^2
This goal function is a simple nonlinear goal function used for testing nonlinear implementations.

Fourier Amplitudes
In the previously discussed derivations, the goal term of the Lagrangian (equation 3.39) is defined as in equation 3.105.

$$\int_\Omega \int_0^{t_f} (g - \hat{g})^2 dt \Omega \tag{3.105}$$

For this particular goal however, the function does not have an instantaneous time component and is only integrated in space. That is:

$$\int_\Omega (g - \hat{g})^2 d\Omega \tag{3.106}$$

To derive g, the following decomposition of the flowfield will be used:

$$u = \sum_{j=0}^{N} |u|(j,0) e^{\xi(j,0)} e^{-ij\omega t} + c.c. \tag{3.107}$$

where $i = \sqrt{-1}$, ω is the temporal frequency, $\hat{u}(j,0)$ is the amplitude belonging to the jth harmonic, ξ_0^u is the corresponding phase and $c.c.$ denotes the complex conjugate. Note that the following discussion applies to all variables (i.e. all velocity components and the pressure) exactly in the same way. Therefore, in the following discussions on the goal, the symbol u will be utilized to represent all variables. For simplicity, the superscripts will therefore also be omitted.

Multiplying both sides of equation 3.107 with $e^{ij\omega t}$, integrating in time and assuming that the integration between $t = 0$ and $t = t_f$ covers an integer multiple of the period of the Fourier modes, the following equation can be obtained:

$$|u|(j,0) e^{\xi_0^u} = \int_0^{t_f} \frac{u}{t_f} e^{ij\omega t} dt = \int_0^{t_f} \frac{u}{t_f} cos(j\omega t) dt + i \int_0^{t_f} \frac{u}{t_f} sin(j\omega t) dt = T_r + iT_i \tag{3.108}$$

Where T_r and T_i are the real and imaginary parts of the integral respectively. The amplitude can simply be recovered by using the Euclidean norm of the resulting imaginary number, i.e.:

$$|u|(j,0) = \sqrt{T_r^2 + T_i^2} \tag{3.109}$$

In equation 3.107 the negative wavenumbers are taken into account in the complex conjugate terms. This corresponds to a two-sided spectrum. To recover the equivalent one-sided amplitudes, the amplitudes of the two-sided spectrum are multiplied by two, resulting in the goal function as seen in equation 3.110.

$$g = 2|u|(j,0) = 2\sqrt{T_r^2 + T_i^2} \tag{3.110}$$

These alterations to the goal have an effect on the goal terms of both the adjoint and gradient expressions. Starting with the adjoint, a variation is applied to the temporal amplitudes (α_q). The derivative with respect to the perturbation ϵ is taken to find the stationary point. In other words, the following expression is evaluated:

$$\frac{d}{d\epsilon}\int_{\Omega}(g-\hat{g})^2 d\Omega \tag{3.111}$$

Ignoring the integral over the domain and applying the chain rule several times, one can conclude that:

$$\frac{d}{d\epsilon}(g-\hat{g})^2 = -\int_0^{t_f} \phi_q \frac{4(g-\hat{g})}{t_f\sqrt{T_r^2+T_i^2}}\left[T_r cos(j\omega t) + T_i sin(j\omega t)\right]\eta(t)dt \tag{3.112}$$

The gradient derivation follows a very similar approach. The gradient expression was found by taking the following partial derivative:

$$\frac{\partial}{\partial c_{qs}}\int_{\Omega}(g-\hat{g})^2 d\Omega \tag{3.113}$$

By ignoring the domain integral and following a very similar procedure as done with adjoint term, the derivative can be written as:

$$\frac{\partial}{\partial c_{qs}}(g-\hat{g})^2 = \int_0^{t_f} \alpha_q\psi_s \frac{4(g-\hat{g})}{t_f\sqrt{T_r^2+T_i^2}}\left[T_r cos(j\omega t) + T_i sin(j\omega t)\right]dt \tag{3.114}$$

Although the formulation is complete, the overall implementation of this goal function turned out to be more complex than initially expected. For this reason it is recommended that this goal is implemented in the future. As a stand-in the following goal function is used.

Real Fourier Amplitudes
As the above goal function is relatively difficult to implement in the current tools, an alternative can be found in this goal function. This function does have an instantaneous time component and thus does fit within the current infrastructure. The idea of this function stems from the fact that the flowfield is purely real, as discussed before. This results in complex conjugate amplitudes. Thus when inspecting the decomposition in equation 3.107, the contribution of the jth harmonic to the instantaneous flowfield is:

$$u(j,0) + u(-j,0) = |u|(j,0)e^{\xi(j,0)}e^{-ij\omega t} + |u|(-j,0)e^{\xi(-j,0)}e^{ij\omega t} = 2T_r cos(j\omega t) \tag{3.115}$$

This goal function will thus aim to optimally represent the contribution of each timestep to the integral that makes up T_r.

Nonlinear interaction perturbation field
When inspecting the nonlinear perturbation equations (equation 1.6), a term that is neglected in the PSE and OS equations is the nonlinear interaction term of the perturbation field. Since there might exist nonlinear interaction in the perturbation field, representing the norm of these terms could prove a useful goal function to identify nonlinear effects. As the perturbation field is of interest, the base flow needs to be subtracted first, that is:

$$g^r(\hat{u}) = \sum_{j=1}^{N_{dim}} (\hat{u}^j - U_b^j)\left(\frac{\partial \hat{u}^r}{\partial x^j} - \frac{\partial U_b^r}{\partial x^j}\right) = \sum_{j=1}^{N_{dim}} \hat{u}'^j \frac{\partial \hat{u}'^r}{\partial x^j} \tag{3.116}$$

where u' is introduced to shorten the notation and U_b is the base flow. This goal is considerably more complex than previously discussed goals. First and foremost, it should be noted that the pressure does not have a contribution to this goal (i.e. $g^p = 0$). Furthermore, this goal is different for each velocity component and contains all two (or three) velocity components for each entry in the vector.[2]

---

[2] i.e. where previously the goal vectors looked like: $g = [g(\hat{u}), g(\hat{v}), g(\hat{p})]^T$, this goal vector looks like $g = [g(\hat{u},\hat{v}), g(\hat{u},\hat{v}), 0]^T$

Moreover, this goal poses a problem for the overall formulation. That is, when deriving the goal terms of the adjoint (see e.g. equation 3.62) and the gradient (see e.g. equation 3.78) it is assumed that the goal function only depends on $\hat{u}$ without any of its gradients such that for example:

$$\frac{\partial \hat{g}^r}{\partial \alpha_q^d} = \frac{\partial \hat{g}^r}{\partial \hat{u}^d}\frac{\partial \hat{u}^d}{\partial \alpha_q^d} = \frac{\partial \hat{g}^d}{\partial \hat{u}^d}\phi_q^d \tag{3.117}$$

However, with the addition of the spatial gradients and the addition of the sum, the goal term of the gradient becomes more complex. In fact, equation 3.78 becomes:

$$[\text{Goal Term}] = -\sum_{r=1}^{N_{dim}}\left[\alpha_q^d \psi_s^d \frac{\partial \hat{u}'^r}{\partial x^d}(g^r - \hat{g}^r) + \alpha_q^d \frac{\partial \psi_s^d}{\partial x^r}\hat{u}'^r (g^d - \hat{g}^d)\right] \tag{3.118}$$

Since there are no terms related to the pressure, the goal term in equation 3.95 can be omitted. When considering the adjoint, a similar problem arises. The forcing term in the weak form of the adjoint equation (equation 3.62) needs to be expanded with:

$$\mathscr{F}^d = \int_\Omega \sum_{r=1}^{N_{dim}}\left[\phi_q^d \frac{\partial \hat{u}'^r}{\partial x^d}(g^r - \hat{g}^r) + \frac{\partial \phi_q^d}{\partial x^r}\hat{u}'^r (g^d - \hat{g}^d)\right] d\Omega \tag{3.119}$$

Moreover, the stabilization of the adjoint requires the strong form of the equations. Originally, the forcing on the strong form of the adjoint PDE is recovered by arguing that the weighting function ϕ_q^d that multiplies the term $(g^d - \hat{g}^d)\frac{\partial \hat{g}}{\partial \hat{u}}$ in equation 3.49 is arbitrary, and thus the forcing is simply: $(g^d - \hat{g}^d)\frac{\partial \hat{g}}{\partial \hat{u}}$. In equation 3.119 this is unfortunately not so simple due to the presence of the gradient of the weighting function. This derivative needs to be transferred to the other terms if the strong form of the adjoint equation is to be recovered.

Fortunately, transferring the derivatives can be achieved through integration by parts. When this integration is performed and the boundary terms are omitted, the following integral arises:

$$\begin{aligned}\mathscr{F}^d &= \int_\Omega \phi_q^d \sum_{r=1}^{N_{dim}}\left[\frac{\partial \hat{u}'^r}{\partial x^d}(g^r - \hat{g}^r) - \frac{\partial}{\partial x^r}\left(\hat{u}'^r (g^d - \hat{g}^d)\right)\right] d\Omega \\ &= \int_\Omega \phi_q^d \sum_{r=1}^{N_{dim}}\left[\frac{\partial \hat{u}'^r}{\partial x^d}(g^r - \hat{g}^r) - \frac{\partial \hat{u}'^r}{\partial x^r}(g^d - \hat{g}^d) - \hat{u}'^r\left(\frac{\partial g^d}{\partial x^r} - \frac{\partial \hat{g}^d}{\partial x^r}\right)\right] d\Omega\end{aligned} \tag{3.120}$$

with:

$$\frac{\partial g^d}{\partial x^r} = \sum_{j=1}^{N_{dim}}\left[\frac{\partial u'^j}{\partial x^r}\frac{\partial u'^d}{\partial x^j} + u'^j \frac{\partial^2 u'^d}{\partial x^r \partial x^j}\right] \tag{3.121}$$

which can be used to calculate the forcing needed to make the stabilization of the adjoint consistent.

3.8. Gradient Verification

A manufactured solution is used to verify the implementation of the above gradient calculations, the adjoint solver and the optimization algorithm. This method starts by assuming a solution to the governing equations. This manufactured solution can then be substituted into the governing equations (equations 3.8-3.11) to find the source terms that would reproduce this solution. By applying these source terms in the numerical approximations of the solution, the numerical methods can be tested and compared to the exact solution.

In the rest of this section, a 1D, 2D and 3D verification case will be discussed. Each case has a different manufactured solution with a few commonalities. The first commonality is the presence of a constant term of 1 (see e.g. equations 3.123). Although seemingly obsolete, this term acts as a manufactured mean or base flow. Since the perturbation field is of interest in the applications discussed in chapters 4 and 5, the mean or base flow is given and a constrained degree of freedom in the overall system of equations solved to obtain the solution. The manufactured base flow terms are present to simply test the implementation of these constraints.

Furthermore, the time dependency of each of the manufactured solutions is the same for all cases and specified by $\sin^2(t)$. Each simulation is run for halve of a period, i.e. $t \in [0, \pi/2]$. The spatial domain is a cube of unit length. That is, $x \in [0,1]$, $y \in [0,1]$ & $z \in [0,1]$.[3] The value for viscosity is set to $\nu = 0.0001$, which corresponds to a Reynolds number of $Re = 10^4$ if a dimensional setting was considered. A value for $\beta = 2.0$ is chosen to test the orthonormality constraint.

The final parameter that needs to be determined is the initial trust-region size. In this case it is decided to utilize the simplest form of the three initial trust-region sizes investigated in Cheng (2017). That is, the initial trust-region size is specified by equation 3.122.

$$\Delta_0 = ||\nabla \mathscr{L}|| \tag{3.122}$$

3.8.1. 1D Case

Although the cases of the subsequent chapters will not be 1-dimensional, the manufactured solution case is included here. This is for both completeness and the advantage that it is tractable to construct the response surface of the Lagrangian from equation 3.39. This is in term advantageous to verify the stepping direction of the algorithm. The manufactured solution is specified in equation 3.123.

$$u^i(x,t) = p(x,t) = 1 + \sin^2(t)\sin(\pi x) \tag{3.123}$$

The secondary basis used here are a set of sines with different wavelengths. The set includes the wavelength used in equation 3.123, that is:

$$\psi = [\sin(\pi x)\ \sin(2\pi x)\ \ldots\ \sin(n\pi x)]^T \tag{3.124}$$

Only one primary mode is needed to optimally represent the solution as specified by equation 3.123. The corresponding 'optimal' ψ-to-ϕ coefficients for both velocity and pressure are:

$$c_{ij} = \begin{cases} \sqrt{2}, & for\ j = 1 \\ 0, & for\ j > 1 \end{cases} \tag{3.125}$$

where the value for the first coefficient is found by applying the orthonormality constraint from equation 3.41 and solving for the only non-zero coefficient.

The verification case is subsequently defined by considering two wavelengths for the secondary basis functions. Leading to equation 3.126[4] for the primary bases ϕ. For the goal, a simple global and linear goal is used. This is the same goal function as used when evaluating the POD and corresponds to goal number 1 in table 3.1.

$$\begin{bmatrix} \phi_1^u \\ \phi_1^p \end{bmatrix} = \begin{bmatrix} c_{1,1}^u & c_{1,2}^u \\ c_{2,1}^p & c_{2,2}^p \end{bmatrix} \begin{bmatrix} \sin(\pi x) \\ \sin(2\pi x) \end{bmatrix} \tag{3.126}$$

The first coefficient of both variables is set and constrained to the optimal value of $c_{1,1}^u = c_{2,1}^p = \sqrt{2}$. The second coefficients are then optimized with several starting conditions by defining the initial condition vector as $c_0 = [c_{1,2}^u, c_{2,2}^p]$. Firstly, the case with starting conditions $c_0 = [-0.05, 0.6]$ (the blue line in figures 3.1& 3.2) clearly shows one of the problems highlighted by Bui-Thanh et al. (2007). That is, the shape of the response surface is not guaranteed to be convex, resulting in the algorithm possibly converging to a local minimum. This phenomenon is the reason why these coefficients do not converge to the theoretical optimal values as seen in table 3.2, highlighting the need for a suitable initial guess.

---

[3] This is of course dependent on the amount of dimensions in question.

[4] Note that the equation can be written in this matrix form, because the same secondary basis is used for both variables.

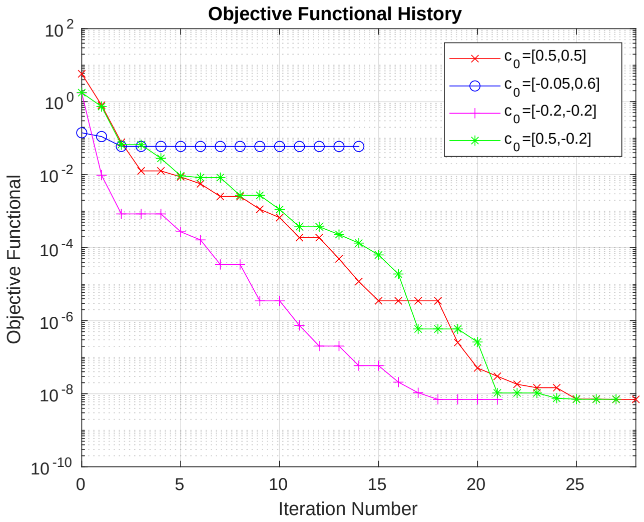
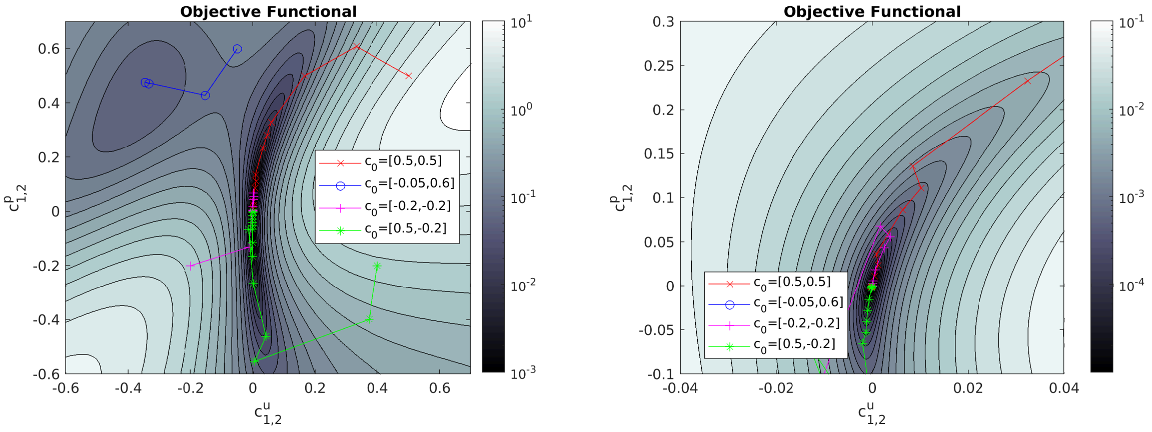


Figure 3.1: Objective functional history for several starting points



(a) Full range

(b) Adjusted Range

Figure 3.2: Response surface of the Lagrangian for the 1D MFS case along with the optimization history.

Table 3.2: Resulting values for the optimization variables and objective functional for a 1D manufactured solution case with different starting conditions.

Initial Coefficients $\left([\mathbf{c}^{\mathbf{u}}_{1,2}, \mathbf{c}^{\mathbf{p}}_{2,2}]\right)$	**Final** $\mathbf{c}^{\mathbf{u}}_{1,2}$	**Final** $\mathbf{c}^{\mathbf{p}}_{2,2}$	**Objective Functional**
$[0.5, 0.5]$	$-9.024 \cdot 10^{-6}$	$2.882 \cdot 10^{-5}$	$7.030 \cdot 10^{-9}$
$[-0.05, 0.6]$	0.3346	0.4715	0.05954
$[-0.2, -0.2]$	$-9.0704 \cdot 10^{-6}$	$3.0978 \cdot 10^{-5}$	$6.9980 \cdot 10^{-9}$
$[0.5, 0.2]$	$-1.0881 \cdot 10^{-5}$	$-1.3220 \cdot 10^{-5}$	$7.0777 \cdot 10^{-9}$

Another interesting observation from these results is the difference in convergence speed of the coefficients for the velocity and pressure. This is illustrated in figure 3.2a, where it can be seen that the value of the objective functional is much more sensitive to changes to the coefficients related to the velocity than changes in the coefficient related to the pressure around the theoretical optimal.

With these two observations in mind, along with the correct convergence behaviour of all four cases, it can be concluded that the simulation is able to accurately perform calculations of the gradient and correctly converges to either a local or a global minimum for a 1D case.

3.8.2. 2D Case

A slightly different manufactured solution is utilized to test the 2D implementation. This manufactured solution is designed to test the applied boundary condition more rigorously, as well as mimic a physical case more closely. That is, the solution is chosen to be divergence free. This is done by defining a streamfunction as seen in equation 3.127:

$$\Psi(x,y) = \sin(\pi x)\sin(\pi y)\sin^2(t) + y - x \tag{3.127}$$

The resulting solution for the variables can then be found in equation 3.128-3.130.

$$u = \pi\sin(\pi x)\cos(\pi y)\sin^2(t) + 1 \tag{3.128}$$

$$v = -\pi\cos(\pi x)\sin(\pi y)\sin^2(t) + 1 \tag{3.129}$$

$$p = \sin(\pi x)\sin(\pi y)\sin^2(t) + 1 \tag{3.130}$$

Due to the presence of the cosines and sines in the manufactured solution, a few more modes per variable are required compared to the 1D case. It is chosen to use the same secondary basis for all variables as seen in equation 3.131.

$$\psi = \begin{bmatrix} \sin(\pi x)\sin(\pi y) \\ \sin(\pi x)\cos(\pi y) \\ \cos(\pi x)\sin(\pi y) \\ \cos(2\pi x)\cos(2\pi y) \end{bmatrix} \tag{3.131}$$

The corresponding ψ-to-ϕ map can be found in equation 3.132.

$$\begin{bmatrix} \phi_1^u \\ \phi_1^v \\ \phi_1^p \end{bmatrix} = \begin{bmatrix} c_{1,1}^u & c_{1,2}^u & c_{1,3}^u & c_{1,4}^u \\ c_{2,1}^v & c_{2,2}^v & c_{2,3}^v & c_{2,4}^v \\ c_{3,1}^p & c_{3,2}^p & c_{3,3}^p & c_{3,4}^p \end{bmatrix} \begin{bmatrix} \sin(\pi x)\sin(\pi y) \\ \sin(\pi x)\cos(\pi y) \\ \cos(\pi x)\sin(\pi y) \\ \cos(2\pi x)\cos(2\pi y) \end{bmatrix} \tag{3.132}$$

Comparing equation 3.132 to equations 3.128-3.130, it can be seen that the theoretical optimal modes are present in the secondary basis. Starting away from this set of optimal modes and running an optimization should then recover the optimal modes for each variable. This theoretical optimal solution is found by applying the orthonormality constraint (equation 3.41) and solving for the optimal coefficient to arrive at:

$$c = \begin{cases} 2, & \text{if } c = c_{1,2}^u \text{ or } c = c_{2,3}^v \text{ or } c = c_{1,1}^p \\ 0, & \text{otherwise} \end{cases} \tag{3.133}$$

Each of the optimal coefficients are then initialized to a value of $c_{1,2}^u = c_{2,3}^v = c_{1,1}^p = 1$ and the coefficient of the fourth mode (i.e. the $cos(2\pi x)\cos(2\pi y)$ mode) is set to a value of $c_{1,4}^u = c_{2,4}^v = c_{3,4}^p = 0.5$ before running the optimization. The optimization is run for both a linear goal (goal number 1 in table 3.1) and a nonlinear goal (goal number 3 in table 3.1).

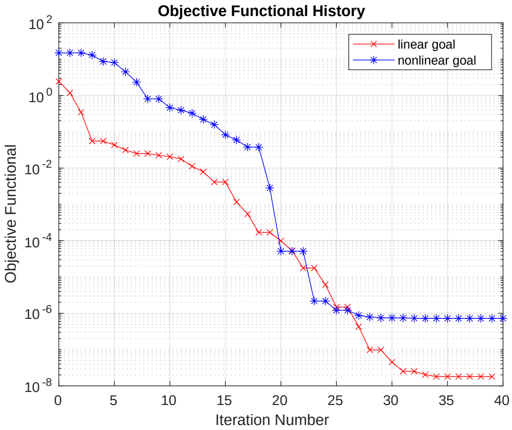
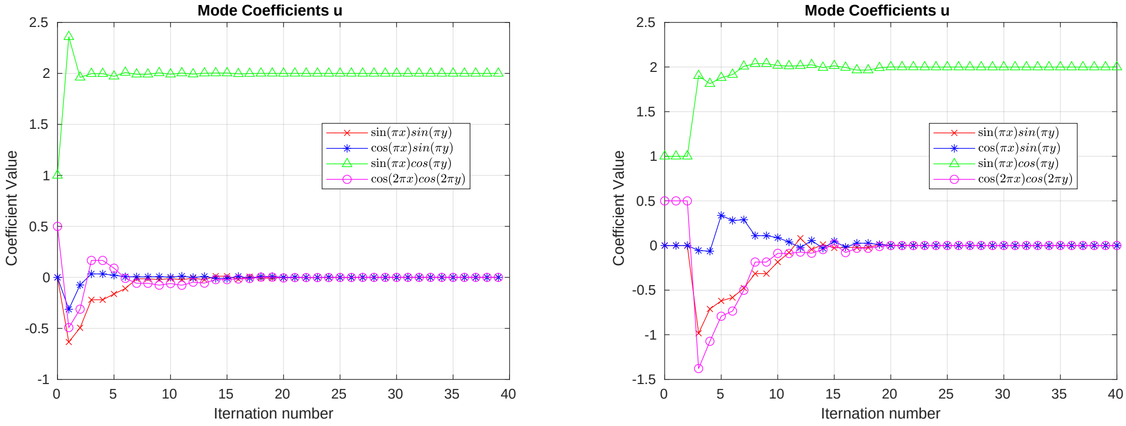


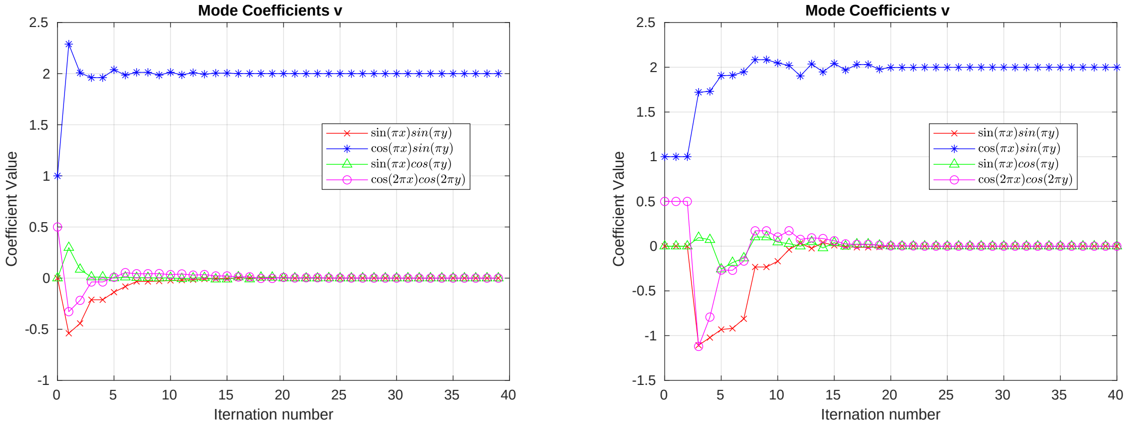
Figure 3.3: History of the value of the objective functional in the 2D manufactured solution case for a linear and non-linear goal.



(a) Linear goal

(b) Nonlinear goal

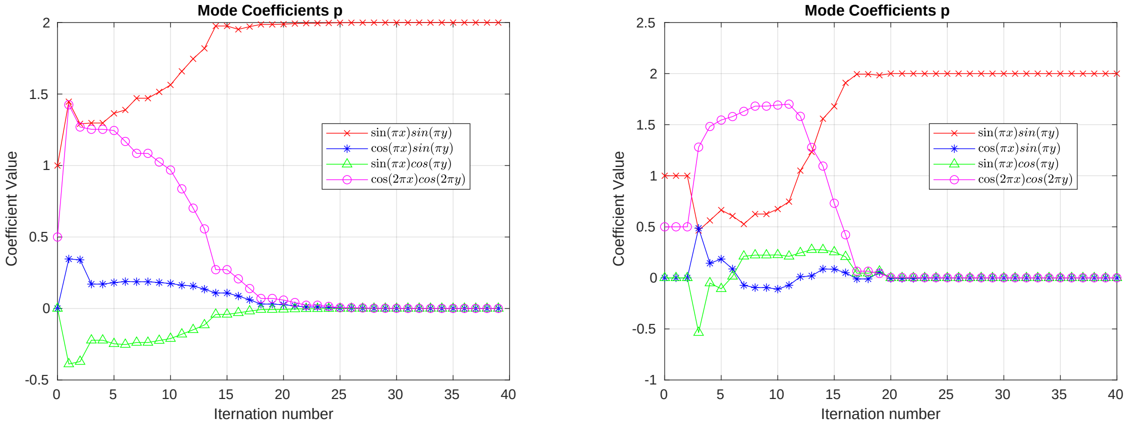
Figure 3.4: Optimization history of the coefficients of the x-velocity components with a linear and non-linear goal.



(a) Linear goal

(b) Nonlinear goal

Figure 3.5: Optimization history of the coefficients of the y-velocity components with a linear and non-linear goal.



(a) Linear goal

(b) Nonlinear goal

Figure 3.6: Optimization history of the coefficients of the pressure with a linear and non-linear goal.

Table 3.3: Resulting values for the optimization variables and objective functional for a 2D manufactured solution case with different goal functions.

Goal Function	**Variable**	$\mathbf{c_{opt}}$	$\lvert\lvert\mathbf{c_{nonopt}}\rvert\rvert$
linear	u	1.99998	$8.8271 \cdot 10^{-5}$
	v	1.99997	$5.2982 \cdot 10^{-5}$
	p	1.99993	$3.9736 \cdot 10^{-4}$
nonlinear	u	1.99945	$1.5974 \cdot 10^{-4}$
	v	1.99953	$1.9537 \cdot 10^{-4}$
	p	1.99988	$4.9972 \cdot 10^{-4}$

For both the linear and nonlinear goal functions the resulting coefficients converge relatively close to their theoretical minimums as seen in table 3.3. It can be seen from figure 3.3 that the linear goal converges faster and smoother than the nonlinear counterpart. Both observations seem to suggest that the algorithm converges faster for linear goal functions then for nonlinear goal functions. Aside from this, the linear goal is able to find a lower value for the objective functional than the linear goal, although the nonlinear goal still converges sufficiently.

From figures 3.4-3.6 it can be concluded that the velocity coefficients seem to converge more quickly to the optimum value than the pressure coefficients. This is an observation made for the 1D case as well, showing a larger dependency of the velocity coefficients on the objective functional.

Finally, where the initial trust-region size determined using equation 3.122 seems to be a good value for the linear goal, for the nonlinear goal the trust-region size is too large, which is the reason why the algorithm does not make any progress in the first two iterations. Although outside the scope of this work, in the future it could be possible to look into alternatives for the initial trust-region sizes, such as the one proposed by Sartenaer (1997).

3.8.3. 3D Case

The transitional flow case considered in chapter 4 will be 2 dimensional. However, the expressions in the state (equations 3.35- 3.38), the adjoint (equations 3.62 & 3.63) and the gradient (equations 3.77 & 3.95) are generalized up to 3 dimensions. A 3D manufactured solution case is considered here as well for the sake of completeness. The manufactured solution case is very similar to the 1D case and is defined by:

$$u^i(\mathbf{x},t) = p(\mathbf{x},t) = 1 + \sin^2(t)\sin(\pi x)\sin(\pi y)\sin(\pi z) \tag{3.134}$$

The secondary basis consists out of a tensor product, i.e.:

$$\psi = \begin{bmatrix} \sin(\pi x) \\ \sin(2\pi x) \end{bmatrix} \otimes \begin{bmatrix} \sin(\pi y) \\ \sin(2\pi y) \end{bmatrix} \otimes \begin{bmatrix} \sin(\pi z) \\ \sin(2\pi z) \end{bmatrix} \tag{3.135}$$

This tensor product results in 8 secondary basis functions. With one primary mode per variable and 4 variables, a total of 32 coefficients are optimized. When comparing the solution to the secondary basis, it can again be seen that the optimal decomposition is present within the secondary basis. The coefficients associated with the optimal mode (i.e. $\sin(\pi x)\sin(\pi y)\sin(\pi z)$) are initialized to a value of 1 and coefficients associated to the mode $\sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$ are initialized to a value of 0.3. To find the theoretically optimal vectors, the orthonormality constraint is solved to arrive at:

$$c_{ij} = \begin{cases} \sqrt{8} \approx 2.828427, & \text{for } j = 1 \\ 0, & \text{for } j > 1 \end{cases} \tag{3.136}$$

The resulting optimization histories for a linear and nonlinear goal can be found in figure 3.7. Table 3.4 shows the final value of the theoretical optimal coefficients (c_{opt}) and the Euclidean norm of the coefficient that in theory should be zero (c_{nonopt}). The linear and nonlinear goal functions again correspond to goal 1 and 3 from table 3.1 respectively.

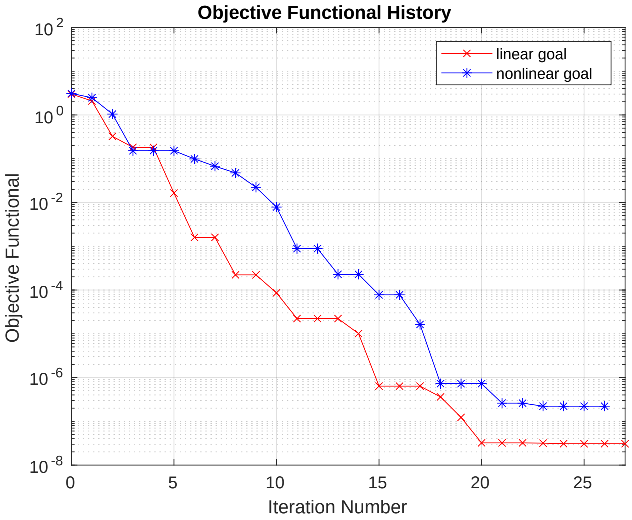


Figure 3.7: History of the value of the objective functional in the 3D manufactured solution case for a linear and non-linear goal.

Table 3.4: Resulting values for the optimization variables and objective functional for a 2D manufactured solution case with different goal functions.

Goal Function	**Variable**	$\mathbf{c_{opt}}$	$\|\|\mathbf{c_{nonopt}}\|\|$
linear	u	2.82842	$3.6755 \cdot 10^{-4}$
	v	2.82842	$3.6711 \cdot 10^{-4}$
	w	2.82842	$3.6718 \cdot 10^{-4}$
	p	2.82842	$2.2662 \cdot 10^{-4}$
nonlinear	u	2.82843	$3.3401 \cdot 10^{-4}$
	v	2.82843	$3.3401 \cdot 10^{-4}$
	w	2.82843	$3.3402 \cdot 10^{-4}$
	p	2.82844	$6.0340 \cdot 10^{-4}$

Similar to the 2D case the method converges to the theoretical optimal values closely. Furthermore, the nonlinear goal converges with a lower rate compared to the linear goal as well. For this 3D case it seems that the initial trust-region size is more appropriate compared to the 2D case for the nonlinear goal function however. With these results in mind, it can be concluded that the expressions found in sections 3.2-3.6 can be applied to problems using the 1D,2D and 3D incompressible Navier-Stokes equations.

4

Tollmien-Schlichting Instability on a Flat Plate

Much simpler methods are available to accurately describe TS waves convecting over a flat plate. Nevertheless, due to the relative simplicity of the problem the GOROM technique will be applied to validate the modal solution techniques, as well as to validate whether GOROM can recover the optimal decomposition in such a case as well.

The structure of this chapter is as follows: firstly, the numerical setup of this case will be discussed. The section after then describes the choice of secondary basis. The subsequent section verifies if the problem is resolved sufficiently in space and time, after which the performance of the secondary basis for several of the goal functions from table 3.1 is tested as a benchmark. The results of the optimizations will be presented and used in the final section to make some concluding remarks on the results.

4.1. Numerical Setup

The integrals from all expressions found in chapter 3 are approximated using numerical quadrature on an integration mesh. Each cell in this integration mesh has one integration point, which is located in the centroid of the cell, allowing the contribution of the cell to be estimated using the midpoint rule.

Since the reference data and POD modes will be defined on the mesh used to resolve the DNS, the integration mesh will be a derivative of this same mesh. The hypothesis is that it is not necessary to use the full DNS mesh to achieve sufficient accuracy for the integration. Therefore the integration mesh will be downsampled relative to the DNS mesh. This is achieved by defining downsampling factors, one for each direction. With a downsampling factor of 1, all vertices in the original DNS mesh will be retained. With a factor of 2, every other vertex will be retained, and so on. The mesh is a nonuniform, Cartesian, multi-block grid. In the wall-normal direction, 192 cells are used using uniform spacing between $y \in [0,3]$. Between $y \in [3,20]$ hyperbolic stretching is applied to generate another 192 cells. For the streamwise direction, uniform spacing is used with 4560 cells between $x \in [0,380]$. A plot of the mesh for the flat plate case can be found in figure 4.1.

To reiterate, the mesh as defined here is based on the mesh used by Shahzad (2020) to generate the reference data. The total size of the domain used in this work is 380x20, which is different from the domain size of 400x20 from Shahzad (2020). This related to the use of outflow boundary conditions. Shahzad (2020) used a method called *selective-frequency damping* to quench the TS waves when flowing out of the domain. Although it is possible to attempt to mimic these outflow conditions for a ROM, the weak boundary condition is used as discussed in 3.2.2. Since the associated forcing term is not modelled, the domain needs to be cut off *before* the region of selective-frequency damping kicks in. As, without the forcing present, the ROM will have to solve a nonphysical solution inside this region. The resulting integration mesh contains 1,751,040 domain integration points.

Finally, the value for the Reynolds number used to normalize the equations, is derived as follows:

$$Re = \frac{U_0 \tilde{L}}{\nu} \tag{4.1}$$

where $U_0 = 10$ m/s, $\tilde{L} = 0.002533$ m and $\nu = 1.5188 \cdot 10^{-5}$ m$^2/s$ are taken from Shahzad (2020). The resulting

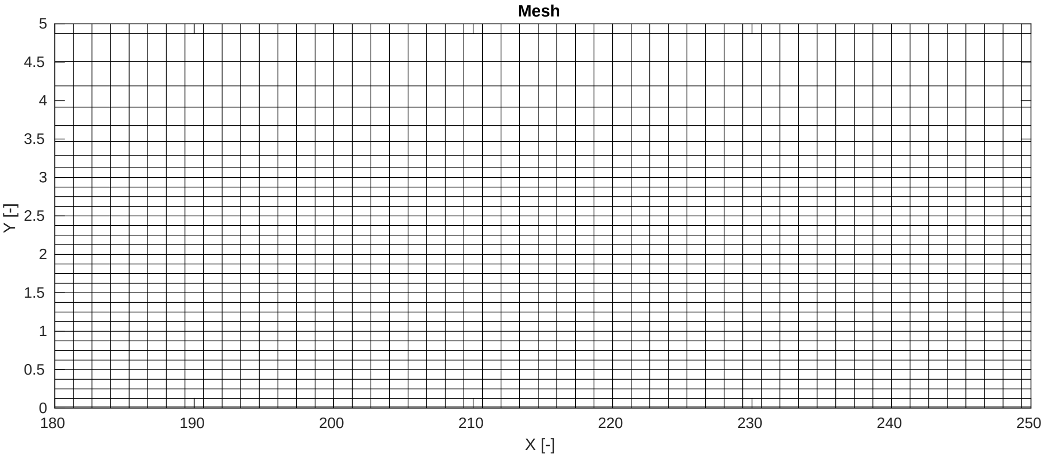


Figure 4.1: Wireframe of part of the DNS mesh used in the simulations of the flat plate. Only every 8th and 16th point are plotted in the wall-normal and streamwise direction respectively.

value for $Re \approx 1667.67$. The value used for the regularization of the orthornormality constraint is $\beta = 2$ and the penalty parameter for the weak boundary condition is $\sigma = 1$.

4.2. POD Reduced-Order Model

The POD modes will be used as the secondary basis and to create a benchmark. The POD is performed on a zero-mean flow field. That is, the mean flow is subtracted from the reference data and a SVD is performed on the resulting snapshots of the data. The mean flow will be constraint in the remaining simulations. For the interested reader, the first four shapes of each variable can be found in appendix E.1.

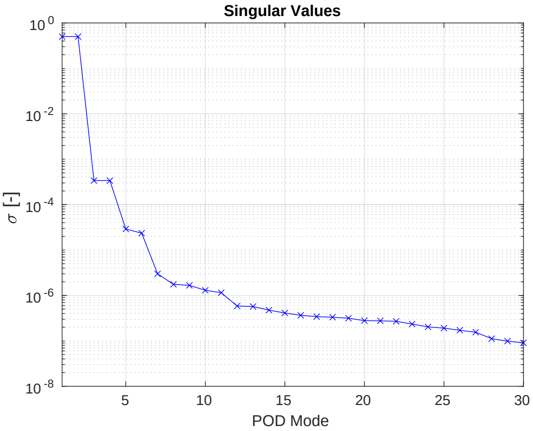
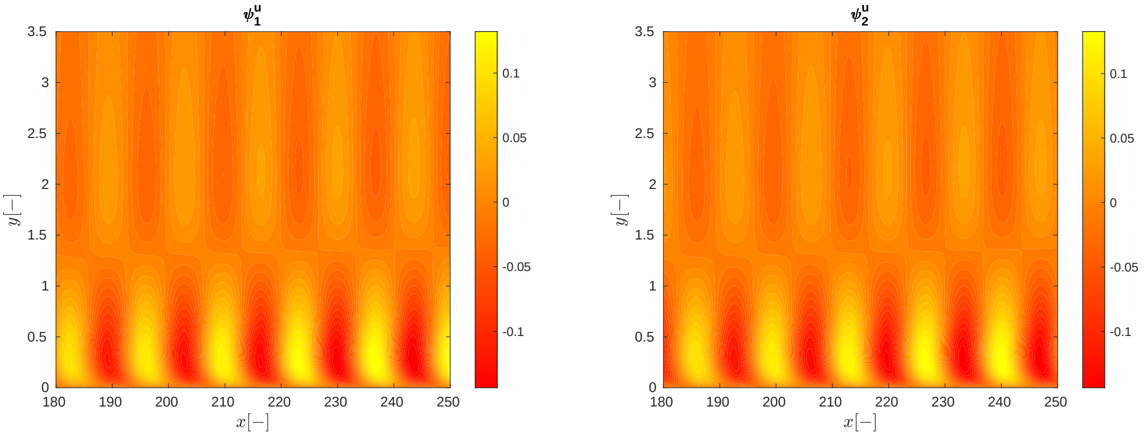


Figure 4.2: Normalized singular values of the reference data for the flat plate case.

Two convection dominated structures can be identified from the singular values in figure 4.2. That is, each convecting structure has a pair of POD modes of similar energy associated with it. Inspecting the first two POD modes of the x-velocity (figure 4.3), the spatial orientation of the modes seems to suggest a phase shift between the two modes, further solidifying the conclusion that these modes capture a convecting structure.

Furthermore, inspecting the shape of the modes also seems to suggest that these capture the fundamental wave convecting over the flat plate. If this is indeed the case, these two modes should also capture most of the content contained within the data. The combined sum of the normalized singular values for these two modes is 0.9993, further strengthening the hypothesis that these modes correspond to the fundamental wave. With this in mind, further analysis of this case will be done with the first two POD modes.



(a) First POD mode

(b) Second POD mode

Figure 4.3: First 2 POD modes of the x-velocity for the flat plate case.

4.2.1. Verification

To verify the modal solver, a spatial refinement test is performed. The value of the objective functional is used as a metric to test convergence. As seen in table 4.1, using the combination $x = 1$ and $y = 2$ results in an integration error two orders of magnitude lower than the value of the objective functional. With this mind, it is decided to use those factors for the downsampling to limit the computational cost of integrating the weak forms.

Table 4.1: Values of the OF for several mesh refinements and number of timesteps. The columns x & y denote the downsampling factor used. Each simulation is ran for 80 timesteps.

x	**y**	**Objective Functional**
1	1	$1.21971 \cdot 10^{-5}$
1	2	$1.29732 \cdot 10^{-5}$
2	1	$1.08652 \cdot 10^{-4}$
2	2	$1.92899 \cdot 10^{-4}$

With the spatial integration tested, the temporal refinement is also tested. Each simulation is performed for one TS period with uniform spacing in time. The simulation is ran for several choices for the total number of timesteps and again, the value of the objective functional is used as a metric to test convergence. Important to note is that the reference data is only available for 20 snapshots per TS period. When running the subsequent simulations with a larger number timesteps, the data is interpolated using linear interpolation between the two closest snapshots.

By inspecting table 4.2, it is decided to use 80 timesteps in the subsequent simulations, as this results in a difference three orders of magnitude lower than the value of the objective functional. With the results from the spatial refinement in mind, this causes the error in the overall discretization to be dominated by the spatial integration, rather than the temporal refinement.

Table 4.2: Values of the OF for several number of timesteps. The difference column is with respect to case with the largest number of timesteps.

Time steps	**Objective Functional**	**Difference**
20	$1.54605 \cdot 10^{-5}$	$3.33254 \cdot 10^{-6}$
40	$1.29732 \cdot 10^{-5}$	$8.45252 \cdot 10^{-7}$
80	$1.21919 \cdot 10^{-5}$	$6.39702 \cdot 10^{-8}$
160	$1.21279 \cdot 10^{-5}$	0

4.2.2. Validation of Modal Solver

To validate the state model, the approach by Shahzad (2020) is mirrored. That is, the resulting flowfield from a simulation of flat plate flow is reconstructed and Fourier transformed in time. The amplitudes of the fundamental wave and first harmonic are computed and the maximum for each streamwise location is plotted. This maximum amplitude plot is compared to the reference data. The resulting streamwise evolution of the amplitudes can be found in figure 4.4.

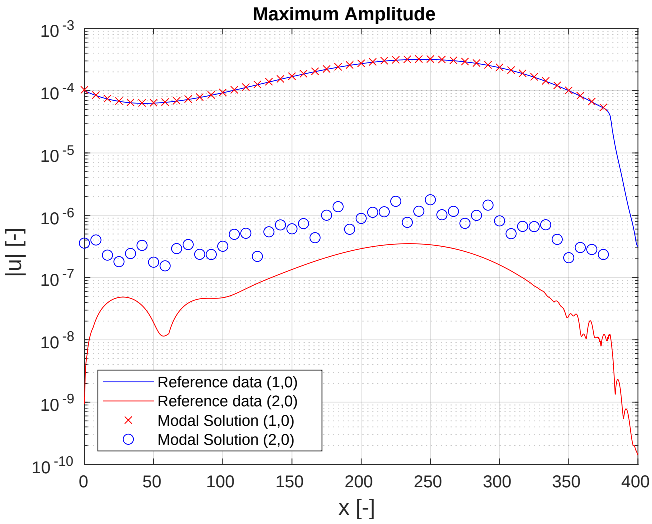


Figure 4.4: Amplitudes of the u(1,0) and u(2,0) component for a flat plate case with different temporal refinements.

The area of selective frequency damping can quite clearly be recognized from figure 4.4, showing the reason for cutting the domain before this region is reached. Furthermore, the state simulations are able to resolve the fundamental wave quite accurately using the first two POD modes, as suspected before.

The result for the first harmonic is more problematic. The first harmonic is not introduced in the inflow boundary conditions, yet grows naturally. This, along with the nature of the problem, does cause the instability to have a very low amplitude. Moreover, the modal solver is not be able to resolve this secondary harmonic sufficiently. This is attributed to the fact that a relatively low number of snapshots of the flow are available per period of this structure. That is, only 10 snapshots are available from which the boundary conditions are inferred. This is particularly relevant on the outflow boundary. Furthermore, only two periods are available in the reference data, which is not perfectly periodic as well. Both these problems might introduce errors in the decomposition. This is clearly not affecting the performance of the modes with the larger amplitudes as much, as predicated by the performance of the primary harmonic. However, at the very small amplitudes observed for the secondary harmonic may lead to errors in the simulation.

4.2.3. ROM Performance

Due to the simplicity of the case in question, the first two POD modes are already able to accurately capture convecting TS waves. Using these modes as the secondary basis and initializing their coefficients to a value of one, will thus most likely already perform well in most norms specified in table 3.1. Quite unsurprisingly, this is indeed the case when evaluating the value of the objective functional for each goal (see table 4.3). Mostly, the fundamental wave is present in the perturbation field. The real Fourier amplitudes and the nonlinear interaction of the perturbation field will thus be captured very well, as also predicated by the results from table 4.3.

Table 4.3: Values of the OF for several goals from table 3.1.

Goal Function	Goal Function Name	Objective Functional
1	Global U	$1.21279 \cdot 10^{-5}$
2	Local U	$7.74121 \cdot 10^{-6}$
3	Global U^2	$1.531967 \cdot 10^{-5}$
5	Real Fourier Amplitudes	$5.19806 \cdot 10^{-6}$
6	Nonlinear Interaction Perturbation Field	$5.88058 \cdot 10^{-6}$

4.3. Goal-Oriented Reduced-Order Model

With results from table 4.3 in mind, running optimizations for this particular case seems illogical, as the truncated POD modes seem to already provide an accurate basis. Therefore, to test the applicability of the GOROM technique in these cases, the optimizations are performed for a set of starting conditions that is perturbed *away* from the POD modes as well. That is, a perturbation (ϵ) is added to the $\psi - to - \phi$ coefficients for two of the POD modes per primary mode. For example, the $\psi - to - \phi$ map for the x-velocity becomes:

$$\begin{bmatrix} \phi_1^u \\ \phi_2^u \end{bmatrix} = \begin{bmatrix} 1.0+\epsilon & \epsilon & 0 & 0 \\ 0 & 1.0+\epsilon & \epsilon & 0 \end{bmatrix} \begin{bmatrix} \psi_1^u \\ \psi_2^u \\ \psi_3^u \\ \psi_4^u \end{bmatrix} \tag{4.2}$$

From the final values of the objective functional from table 4.4 a few observations can be made. Firstly, the results from case A and B show that the GOROM technique slightly modifies the modes to a set that behaves more accurately when used in a ROM. In fact, the GOROM modes outperform the POD modes even when the goal functions are the same, which is a consequence of the model constraint.

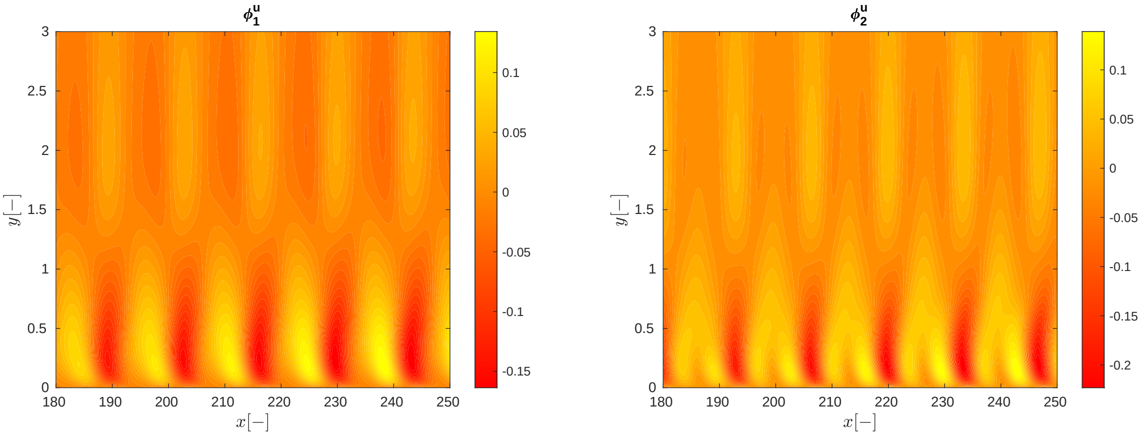
Table 4.4: Optimization parameters and results.

Case Name	Goal Function Name	ϵ	POD OF	GOROM OF
Case A	Global U	0	$1.2128 \cdot 10^{-5}$	$1.4476 \cdot 10^{-6}$
Case B	Local U	0	$7.7412 \cdot 10^{-6}$	$1.0310 \cdot 10^{-6}$
Case C	Global U	0.3	$1.2128 \cdot 10^{-5}$	$1.4669 \cdot 10^{-4}$
Case D	Local U	0.3	$7.7412 \cdot 10^{-6}$	$7.4917 \cdot 10^{-5}$

Case C & D from table 4.4 show two cases where the starting conditions are perturbed away from the POD modes. It can be clearly seen that the GOROM modes both converge to a local minimum. Both sets of GOROM modes are outperformed by the POD modes with an order of magnitude.

The hypothesis that the modes from case C converge to a local minimum is confirmed when considering their shape functions and associated amplitudes, which can be found in figure 4.5 & 4.6 respectively. The resulting shapes seem to be a combination of several harmonics. Moreover, the amplitudes of the modes do show some problematic results as well. That is, the periodicity of the reference data is not represented in any of the amplitudes from figure 4.6. Similar conclusions can be drawn when inspecting the modes from case D.

With these considerations in mind, it can quite safely be presumed that these modes do not correspond to any particular physical instability in the flow, but represent some combination of modes that happens to be close to orthonormal and performs well in a time-averaged sense for one period of the fundamental wave.



(a) First GOROM mode

(b) Second GOROM mode

Figure 4.5: Resulting GOROM shapes for the x-velocity with a perturbed starting condition and the global u goal function.

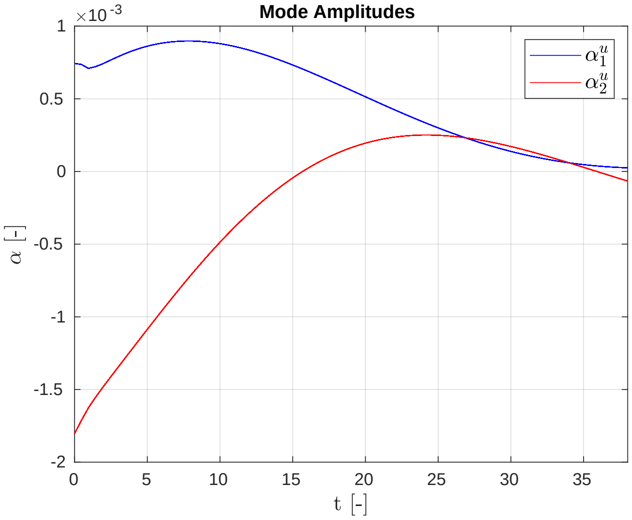


Figure 4.6: Amplitudes of the streamwise GOROM modes of case C from table 4.4.

4.4. Concluding Remarks

From the analysis performed here, it can be concluded that the GOROM method is not preferential in these simple cases. In fact, performing a POD of the reference data seems to already sufficiently recover a set of modes which can be used to interpret the data from the dataset. Furthermore, the added downside possibly encountering a local minimum and the additional cost of constructing modes, makes GOROM not an optimal technique to apply for this flat plate case.

These conclusions are true from the perspective of interpretation. However, the GOROM modes did numerically outperform the POD modes when the optimization is started from the POD. If numerical accuracy of a ROM is important, the GOROM technique can be used to make slight modifications leading to a more accurate ROM even for this relatively simple case.

5

Tollmien-Schlichting Instability with Step

This chapter discusses the results from the optimizations performed on a DNS dataset from a flat plate boundary layer. The difference with the boundary layer from chapter 4 is the presence of a forward-facing step, complicating the flow problem. The DNS is performed and validated by Shahzad (2020). The first section discusses the numerical setup and the different optimizations performed in this work. Section 5.2 then treats the performance of the POD performed on the dataset. The subsequent section then discusses the results from the optimizations to retrieve the GOROM modes. Finally, section 5.4 concludes the chapter with a few additional observations.

5.1. Cases and Numerical Setup

The GOROM technique is applied to two datasets of the cases that include a step. Both datasets result from simulations that use the same inflow amplitude, inflow frequency and step height. These are equivalent to case 12 from Shahzad (2020).

The two datasets result from two simulations with a different domain. To be more precise, one of the datasets has a domain height of 20, where the other has a domain height of 100. The motivation for using the two domain heights can be found when considering the *mean-flow* distortion of both cases. The definition of the mean-flow distortion can be found in equation 5.1.

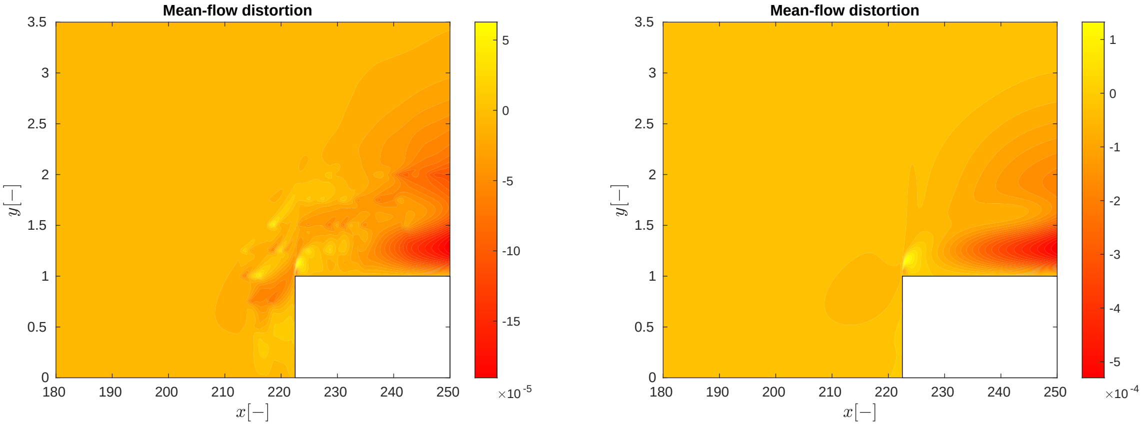
$$u(0,0) = \langle u_{tot} \rangle - U_b \tag{5.1}$$

where $u(0,0)$ is the mean-flow distortion, u_{tot} is the total flow[1], U_b is the base flow and $\langle . \rangle$ is a time-averaging operation. The mean-flow distortion for both cases in an area around the step is plotted in figure 5.1. Some mystifying topology is present in the form of spots in the case with a domain height of 20. The hypothesis for the origin of these 'spots' is that they result from discretization errors due to the mesh quality, model errors due to the top boundary or an interaction between both.

One of these hypotheses can be rejected when considering the mesh of both cases. For clarity, a down-sampled wireframe of the mesh around the step can be found in figure 5.2. The streamwise spacing of the cells is exactly the same for both cases. That is, the first 2160 cells are uniformly spaced between $x \in [0,180]$. Approaching the step, the cells are refined with 600 cells between $x \in [180,212.5]$. Uniform spacing and 1440 cells are then used for the area around the step between $x \in [212.5,242.5]$. The streamwise refinement is then coarsened using hyperbolic stretching with 600 cells between $x \in [242.5,275]$. Finally, uniform cell spacing is used between $x \in [275,380]$ with 1260 cells. The number of streamwise elements per wavelength at two locations is summarized in table 5.1. The wavelengths are inferred from the flat plate case at the relative locations.

---

[1] As discussed in section 1.2, this is the sum of the base flow and perturbation field.



(a) Low top boundary

(b) High top boundary

Figure 5.1: Plot of the mean-flow distortion around the step of the two datasets considered in this work. The x-velocity is plotted in these figures.

Table 5.1: The number of streamwise elements in the mesh per wavelength at the step location and at the outflow of the domain. The wavelengths are inferred from the flat plate case.

Streamwise Location	**Number of Points per Wavelength**
Step Location ($x = 222.5$)	600
Domain Outflow ($x = 350$)	161

Some differences in cell spacing can be observed in wall-normal direction, which is visualized in figure 5.3. The cell index in wall-normal direction is plotted against the y-coordinate of the cell centre. It can be observed that the same uniform spacing is used for both cases between $y \in [0,3]$. The case with a high top boundary then stretches faster compared to the lower boundary. That is, there are *more* cells in wall-normal direction between $y \in [3,20]$ for the case with a low boundary compared to the case with a high boundary. This observation leads to rejecting the hypothesis that the spots from figure 5.1 are purely a result of discretization error. It is still possible the spots originate from numerical errors induced by modeling errors however. Running the DNS of the low boundary case with a refined mesh can be used to definitively show this to be the case. This is outside of the scope of this work however.

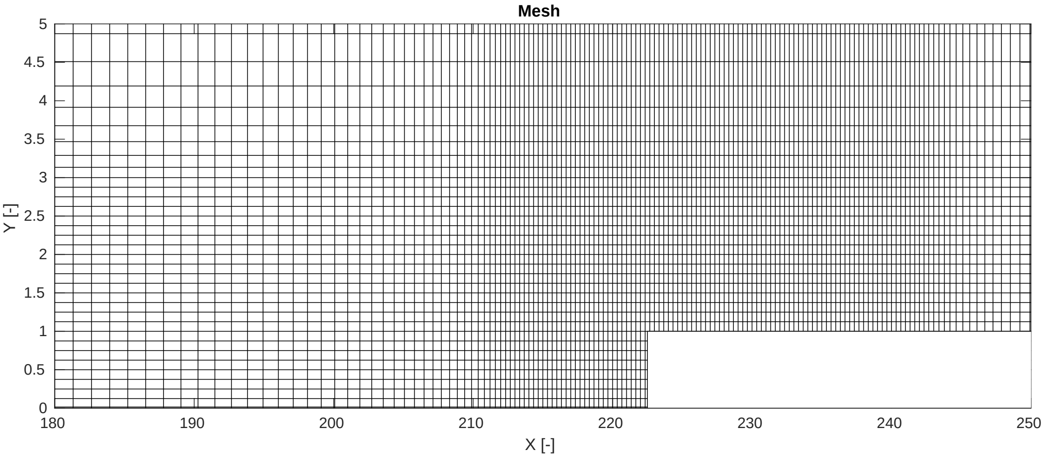


Figure 5.2: Wireframe of part of the DNS mesh used in the simulations of the step case with the lower top boundary. Only every 8th and 16th point are plotted in the wall-normal and streamwise direction respectively

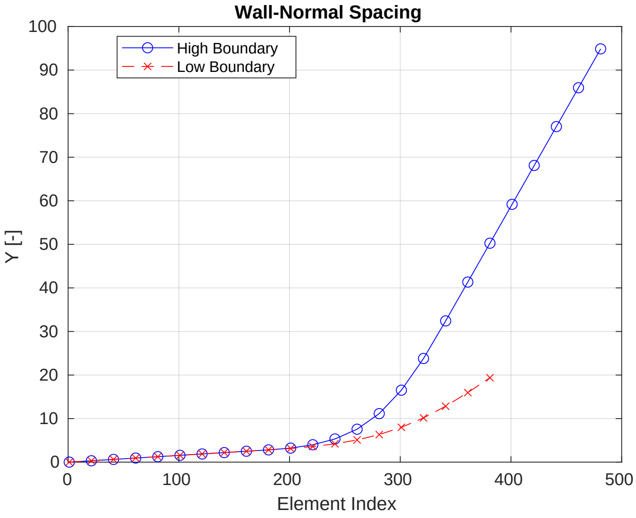


Figure 5.3: Cell location in the wall-normal direction of the two step datasets compared.

From the perspective of the GOROM technique, the mystifying topology induced by the lower top boundary provides an opportunity to test the behaviour of the GOROM technique on datasets in the presence of some of this unnatural topology. The mean or base flow will be constrained during the simulations. This is due to the fact that the mean flow or base flow are not very interesting from the perspective of interpretation. The spots are present in the mean flow however. Thus, if the mean flow is used as the constrained mode, the GOROM technique will not be able to include the topology in the optimizations. Therefore, it is decided to constrain the base flow for the lower boundary cases, rather than the mean flow. This has as a consequence that the mean-flow distortion will show up as part of the POD.

The optimization performed for the step cases can be found in table 5.2. A few notes are of importance before considering the results for these cases. First of all, there is the number of modes present in the GOROM decomposition (i.e. the column N in table 5.2). Case A and B only include 3 GOROM modes to see the effects of the mean-flow distortion. Case C has 4 modes such that convection dominated effects can be captured.

Moreover, the parameter ϵ in this case represents the mode perturbation. This parameter is applied very similarly to the flat plate case. An example of how these perturbations are applied to the coefficients can be found in equation 4.2. This allows some more exploration of the overall shape of the objective functional.

Furthermore, case A-C have a larger streamwise domain than case D-H. As mentioned before, the outflow of the domain uses a technique called selective-frequency damping to quench the TS before flowing out of the domain. The DNS of cases D-H have a larger area in which selective-frequency damping is applied. Thus, cases D-H need to be cut-off earlier than cases A-C.

Table 5.2: Cases used in the optimizations. A description of the goal functions can be found in table 3.1.

Case Name	**Domain Height**	**Constrained Mode**	**Streamwise Domain**	**N**	ϵ	**Goal Function**
Case A	20	Base flow	$y \in [0,380]$	3	0	Global U
Case B	20	Base flow	$y \in [0,380]$	3	0.3	Global U
Case C	20	Base flow	$y \in [0,380]$	4	0.3	Global U
Case D	100	Mean flow	$y \in [0,350]$	4	0.0	Global U
Case E	100	Mean flow	$y \in [0,350]$	4	0.1	Global U
Case F	100	Mean flow	$y \in [0,350]$	4	0.3	Global U
Case G	100	Mean flow	$y \in [0,350]$	4	0.0	Real Fourier Amplitudes
Case H	100	Mean flow	$y \in [0,350]$	4	0.3	Real Fourier Amplitudes

Finally, for the cases considered throughout this chapter, the Reynolds number, regularization parameter β and the weak boundary strength σ all use the same values as used in the flat plate as presented in section 4.1.

5.2. POD Reduced-Order Model

Similar to the flat plate case, the POD modes will be used as a benchmark to compare the performance of the GOROM modes to. Inspecting table 5.2, two sets of POD modes are required. One set for cases A-C and one set for cases D-H. Before performing the SVD to obtain the POD modes, the base flow in cases A-C and the mean flow in cases D-H are subtracted from the snapshots. The resulting normalized singular values of both sets of POD modes can be found in figure 5.4.

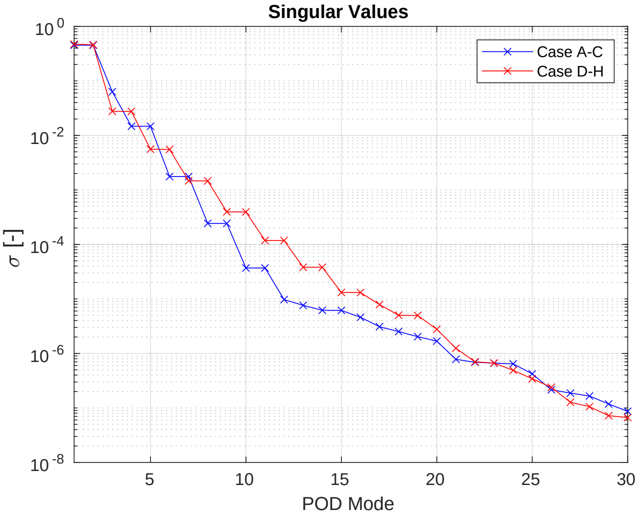
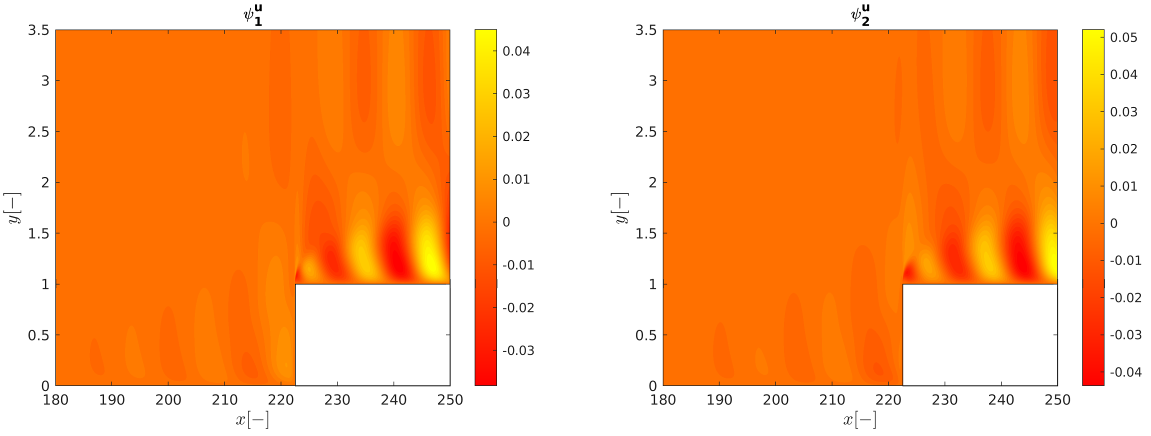


Figure 5.4: Normalized singular values of both the datasets considered in this work.

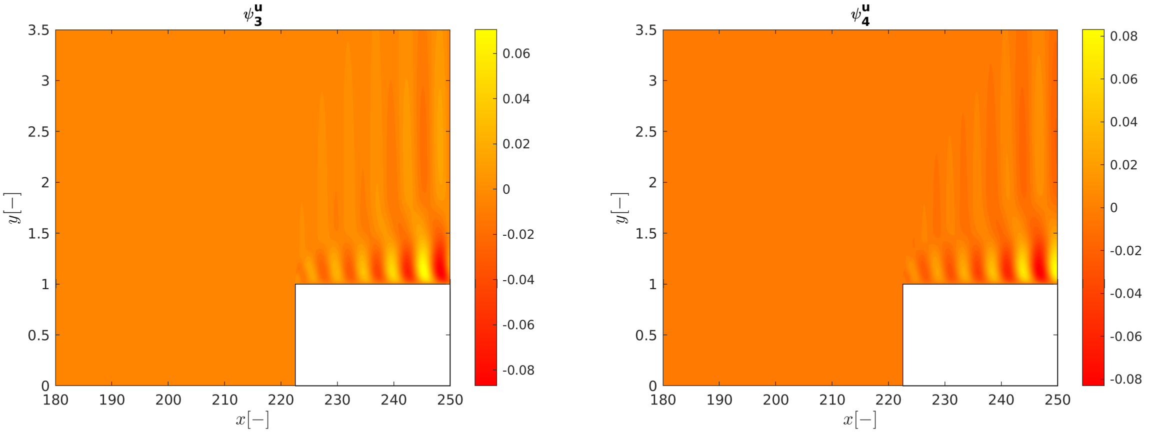
As mentioned before, the convection dominated structures can be identified by pairs of singular values and shape functions that are phase shifted. An example of which would be the first two modes and the third and fourth mode from cases D-H. The shape functions of the POD modes show the phase shift as seen in figure 5.5. Considering their amplitudes plotted in figure 5.6, mode 1 & 2 seem to correspond to the fundamental wave and mode 3 & 4 to the first harmonic. Do note that the projected amplitudes are found by multiplying the singular values by the right singular vectors. All other shape functions can be found in appendix E.2.

The third mode from the low boundary case is of particular interest. This third mode is not paired with any other mode in figure 5.4. Inspecting the shape and projected amplitudes in figure 5.8 reveals the nature of this mode. That is, the mode barely fluctuates in time and the spots observed in the mean-flow distortion are present in this mode as well. Furthermore, the overall topology of the mode is very similar to the topology of the mean-flow distortion. These observation lead to the conclusion that the third mode is equivalent to the mean-flow distortion obtained by other means. Normally, the POD filters the mean-flow from a given set of snapshots in the form of a first (or zeroth) mode. In this case however, the amount of energy in the remaining mean-flow distortion is lower than the amount of energy in the first two modes. This is due to the fact that most of the mean flow has been subtracted from the snapshots through subtracting the base flow. The lower amount of energy results in a smaller singular value and thus the ‘mean’ mode shows up further into the set of modes.



(a) Mode 1

(b) Mode 2



(c) Mode 3

(d) Mode 4

Figure 5.5: Plot of the first two POD modes used in cases D-H. The modes originate from the x-velocity.

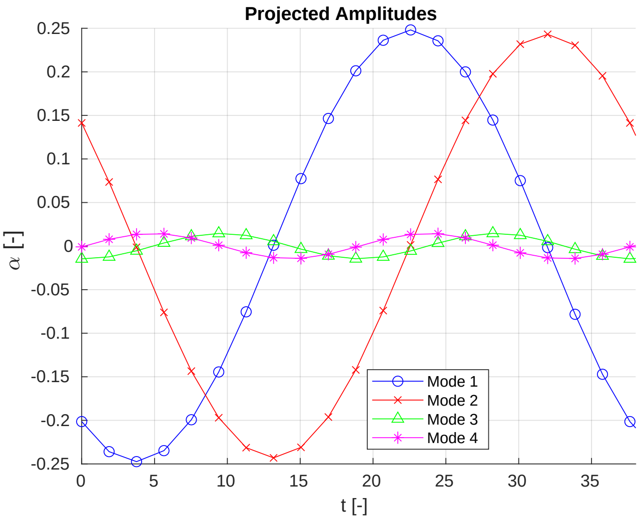


Figure 5.6: Non-normalized, projected amplitudes of the first four POD modes from case D-H.

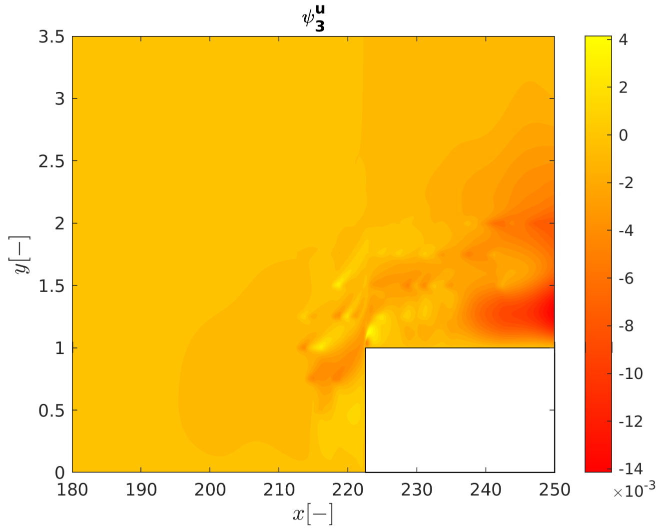


Figure 5.7: Shape function of the third POD mode from the dataset of case A-C.

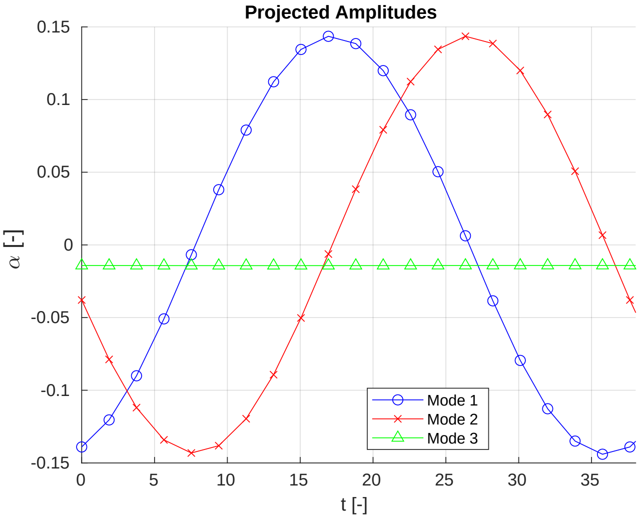


Figure 5.8: Non-normalized, projected amplitudes of the first three POD modes from case A-C.

5.2.1. Verification

To select a sufficient number of timesteps and integration mesh for all cases from table 5.2, a temporal refinement test is performed in table 5.3. In the simulations for cases A-C, a ROM with 5 POD modes is constructed to include the first two convection dominated structures and the mean-flow distortion. In the simulations for cases D-H, a ROM with 4 POD modes is constructed, as the mean-flow distortion is included in the constrained mode. The goal function is the global U goal function, or goal function number 1 from table 3.1. As with the flat plate case, only 20 snapshots are available per period. The value for the reference data is required at each timestep for the weak boundary condition. The reference data is interpolated for the timesteps that are between two snapshots. It is chosen to use a second order Lagrange polynomial for this purpose using the closest three snapshots.

As computational cost is a concern, a difference in objective functional between two consecutive number of time steps of two orders of magnitude lower than the value of objective functional is deemed sufficient. As seen in table 5.3, this achieved for 160 timesteps for case A-C and 320 timesteps for case D-H.

Table 5.3: Values of the OF for several number of timesteps. The difference columns are with respect to case with the largest number of timesteps.

Time steps	**Objective Functional Case A-C**	**Objective Functional Case D-H**
40	$2.5181 \cdot 10^{-3}$	$9.9078 \cdot 10^{-3}$
80	$1.5423 \cdot 10^{-3}$	$1.1485 \cdot 10^{-2}$
160	$1.2679 \cdot 10^{-3}$	$1.2518 \cdot 10^{-2}$
320	$1.2439 \cdot 10^{-3}$	$1.3265 \cdot 10^{-2}$
640	N/A	$1.39169 \cdot 10^{-2}$

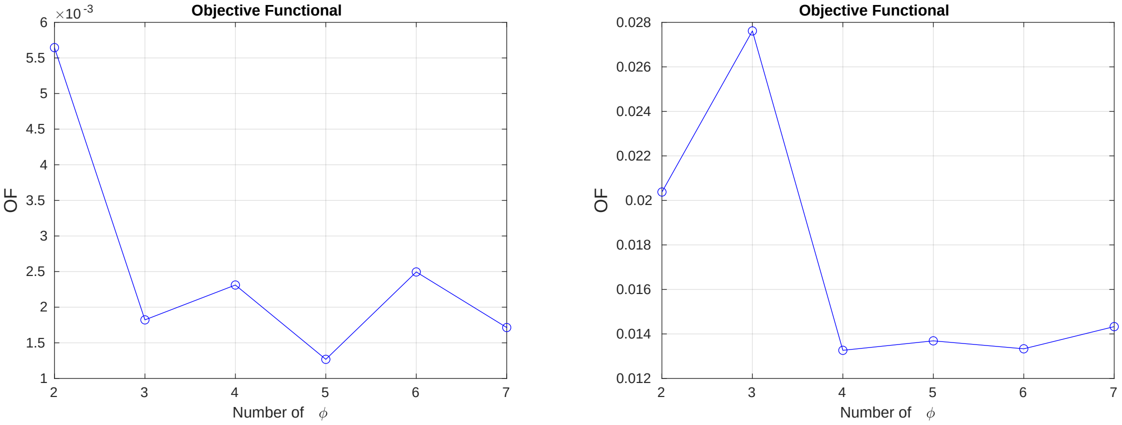
The mesh discussed in section 5.1 is downsampled to investigate if the spatial discretization can be lower than the discretization used to obtain the DNS data. This is achieved by means of downsampling factors, see section 4.1 for an explanation of these factors. From table 5.4 it can be inferred that, counter to the flat plate case, the mesh can not be downsampled and the full DNS mesh needs to be used as the integration mesh.

Table 5.4: Values of the OF for several mesh refinements and number of timesteps. The columns x & y denote the downsampling factor used.

x	**y**	**Objective Functional Case A-C**	**Objective Functional Case D-H**
1	1	$1.2679 \cdot 10^{-3}$	$1.3265 \cdot 10^{-2}$
1	2	4.3818	0.27531
2	1	4.1711	$2.0574 \cdot 10^{-2}$

5.2.2. ROM Performance

Figure 5.9 shows the value of the objective functional plotted against the number of POD modes retained to evaluate the objective functional for cases A-C and cases D-H. Although adding the first few modes does increase the performance of the POD modes in both cases, the performance of the POD rather paradoxically does not increase after 5 and 6 modes for the low and high boundary cases respectively. Increasing the number of POD modes in the simulation should theoretically result in a more accurate ROM. As previously mentioned, the number of snapshots available per period is rather limited. Although the interpolation between snapshots may be sufficient for the fundamental wave, it is questionable whether this holds for the higher harmonics as well.



(a) Low top boundary

(b) High top boundary

Figure 5.9: Value of the objective functional plotted against the number of POD modes retained in the basis. The goal function is the global u goal function from table 3.1.

To create the benchmark to compare the GOROM modes to, the performance of the POD modes in all relevant norms from table 3.1 are presented in table 5.5. Do note that the POD modes for each case are truncated at the same number of modes as the GOROM modes optimized for from table 5.2.

Table 5.5: Values of the OF for several choices of goal function. The number of POD modes retained in the basis corresponds to the column N from table 5.2.

Goal Function	**Goal Function Name**	**Objective Functional Case A-B**	**Objective Functional Case C**	**Objective Functional Case D-H**
1	Global U	$1.8221 \cdot 10^{-3}$	$2.3115 \cdot 10^{-3}$	$1.3262 \cdot 10^{-2}$
2	Local U	$4.9312 \cdot 10^{-5}$	$1.2821 \cdot 10^{-4}$	$2.6962 \cdot 10^{-4}$
3	Global U^2	$2.3814 \cdot 10^{-3}$	$3.3126 \cdot 10^{-3}$	$1.8002 \cdot 10^{-2}$
5	Real Fourier Amplitudes	$3.9899 \cdot 10^{-5}$	$1.1648 \cdot 10^{-4}$	$2.1323 \cdot 10^{-4}$
6	Nonlinear Interaction Perturbation Field	$3.0402 \cdot 10^{-5}$	$1.047 \cdot 10^{-4}$	$1.6036 \cdot 10^{-4}$

Comparing the results from table 5.5 with the results in the flat plate case (table 4.3) shows that the step does add some more complexity to the problem. That is, the step cases have an objective functional a few orders of magnitude higher than the equivalent flat plate case for any choice of goal function. This observation provides a motivation to investigate if the GOROM modes are able to provide a more accurate basis than the POD modes when constructing a ROM with the global u goal function.

A relevant observation is the accurate performance of goal number 6. Goal function 6 aims to quantify how well nonlinear effects in the perturbation field are resolved. It is hypothesized that the step causes nonlinear interactions in the perturbation field to such extend that a truncated POD basis would perform poorly in this norm relative to other norms. This does not seem to be the case however, suggesting that the perturbations grow mostly linear on top of the base flow.

5.3. Goal-Oriented Reduced-Order Model

This section treats the results from the optimizations. First the numerical results from table 5.2 are discussed. The possible physical interpretation of the modes from the relevant cases are then discussed in the subsequent subsection.

5.3.1. Optimizations

The reason for including the mode perturbation ϵ becomes apparent when considering table 5.6. This table shows the results from all cases from table 5.2 compared to the POD results. The parameter in the final column (Δc) is defined as in equation 5.2.

$$\Delta c = \frac{||\mathbf{c_{POD}} - \mathbf{c_{GOROM}}||}{||\mathbf{c_{POD}}||} \tag{5.2}$$

where $\mathbf{c_{POD}}$ and $\mathbf{c_{GOROM}}$ are the coefficient vectors that reproduce the POD and GOROM modes from the secondary basis. This parameter is a scalar value quantifying the relative difference in optimization parameters. This parameter is zero for case A, which thus implies that the GOROM modes are exactly the same as the POD modes. Case B does show a difference. This difference is observed even though the only difference with respect to case A is the starting condition of the optimization. Thus, at least two minima of the objective functional are located relatively close to each other.

Furthermore, the final objective functional for case B is *smaller* than the final objective functional of case A. This observation is interesting from a numerical modeling perspective. That is, a perturbation applied to the starting conditions of the optimization results in a ROM which is more accurate than if started from the POD modes. This suggests that the shape of objective functional for this case is such that the POD modes do not reside in or close to the global minimum of the objective functional.

The results from case D-F confirm the hypothesis that the objective functional is relatively complex and shows several minima in close proximity to each other. That is, the difference in starting conditions of all three cases is not exceptionally large. Despite this fact, each case converges to a different minimum. The perturbed starting conditions do not show improved performance however. Nevertheless, case F is perturbed further away from the POD modes than case E, yet still results in a set of modes that is numerically more accurate than the modes from case E.

The behaviour of the GOROM technique with a different goal function performs considerably better than the POD modes. Interesting is that in case G only minor changes to the set of coefficients are made. The changes in case H are much more pronounced and provide an even lower value for the objective functional, again showing the complexity of the overall shape of the objective functional.

Furthermore, the goal function considered in case G and H is also of note. That is, the objective function is both a local function (i.e. only nonzero around the area of the step) and aims to optimize the performance of the ROM to quantify the fundamental wave. The results presented here show the applicability of the GOROM technique to construct a set of modes better equipped to find ROMs for more relevant goal functions.

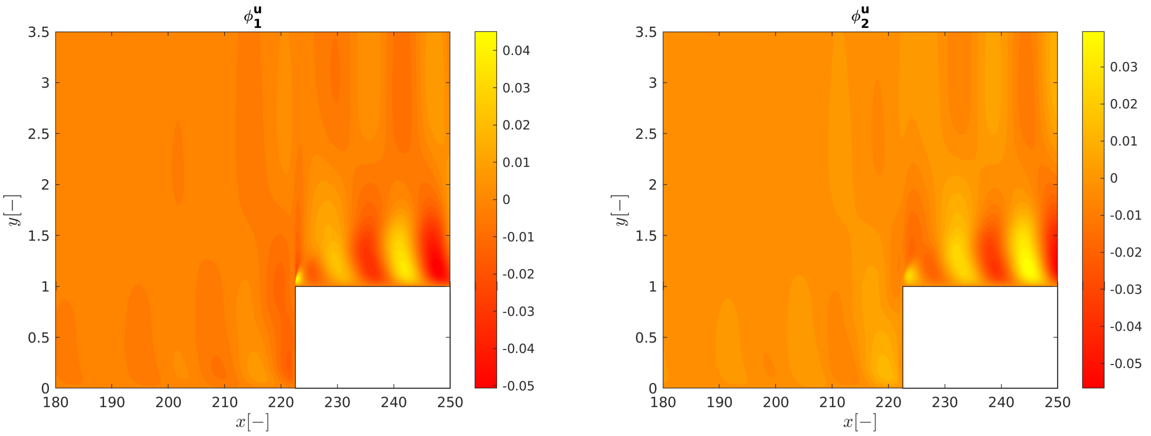
Table 5.6: Results of the optimizations defined in table 5.2. Some of the relevant parameters are present here as well for clarity. The goal function numbers correspond to the numbers found in table 3.1.

Case Name	**Domain Height**	**N**	ϵ	**Goal Function**	**POD OF**	**GOROM OF**	Δc
Case A	20	3	0.0	1	$1.8221 \cdot 10^{-3}$	$1.8221 \cdot 10^{-3}$	0.0
Case B	20	3	0.3	1	$1.8221 \cdot 10^{-3}$	$1.5194 \cdot 10^{-3}$	0.1535
Case C	20	4	0.3	1	$2.3115 \cdot 10^{-3}$	$1.4555 \cdot 10^{-3}$	0.1851
Case D	100	4	0.0	1	$1.3262 \cdot 10^{-2}$	$1.3254 \cdot 10^{-2}$	$7.5539 \cdot 10^{-4}$
Case E	100	4	0.1	1	$1.3262 \cdot 10^{-2}$	$2.1352 \cdot 10^{-2}$	0.075351
Case F	100	4	0.3	1	$1.3262 \cdot 10^{-2}$	$1.7346 \cdot 10^{-2}$	0.31704
Case G	100	4	0.0	5	$2.1323 \cdot 10^{-4}$	$6.1832 \cdot 10^{-5}$	$1.4525 \cdot 10^{-3}$
Case H	100	4	0.3	5	$2.1323 \cdot 10^{-4}$	$2.2376 \cdot 10^{-5}$	0.19426

5.3.2. Mode Shapes and Amplitudes

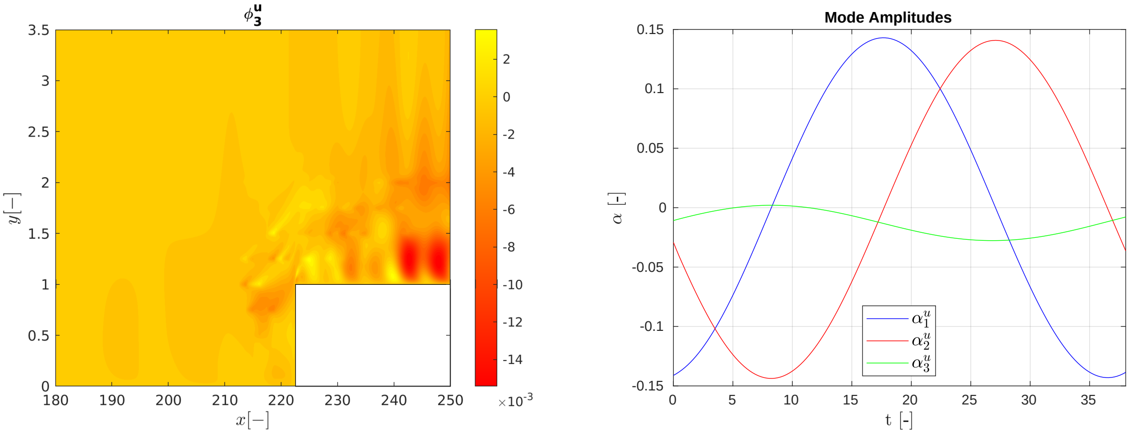
As seen from table 5.6, case A, D and G do not yield a notable change in coefficients and case E and F are outperformed in numerical accuracy by the POD modes. Thus, only case B, C and H are of further interest. All primary shape functions for all variables not discussed in the upcoming discussion can be found in appendix E.3.

The GOROM shapes of the x-velocity from case B and their associated amplitudes in figure 5.10 show some interesting behaviour from the third mode. The shape of the third mode seems a combination of the mean-flow distortion and the first harmonic. With this in mind, the behaviour of the associated amplitudes seems surprising. That is, the time behaviour of this mode corresponds to the fundamental wave where the spatial orientation suggests otherwise. This could be a result of a structure that oscillates at the same frequency as the fundamental harmonic, but has a higher convection velocity compared to the main wave passing over the step.



(a) Mode 1

(b) Mode 2



(c) Mode 3

(d) Amplitudes

Figure 5.10: Plot of the GOROM modes of case B and their associated amplitudes. The modes originate from the x-velocity.

This hypothesis can be tested when considering the shapes (figure 5.12) and amplitudes (figure 5.11) from case C. That is, if the added GOROM mode has a shape with similar topology and an amplitude that is phase shifted in time, the two modes model a convection dominated structure which would provide evidence for the physicality of the modes. This does not seem to be the case however. That is, mode 4 does oscillate at the same frequency as mode 3, yet is not phase shifted in time by 90°, nor has similar energy. The hypothesis is that mode 3 in case B and mode 3 & 4 in case C are correcting for the presence of the earlier observed spots, rather than representing some physical structure.

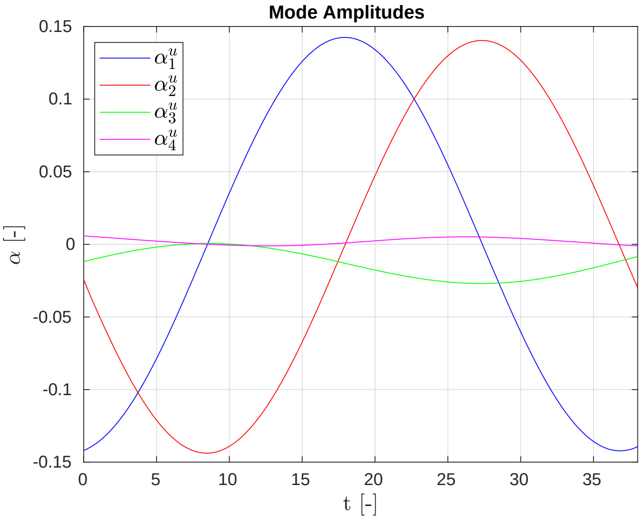
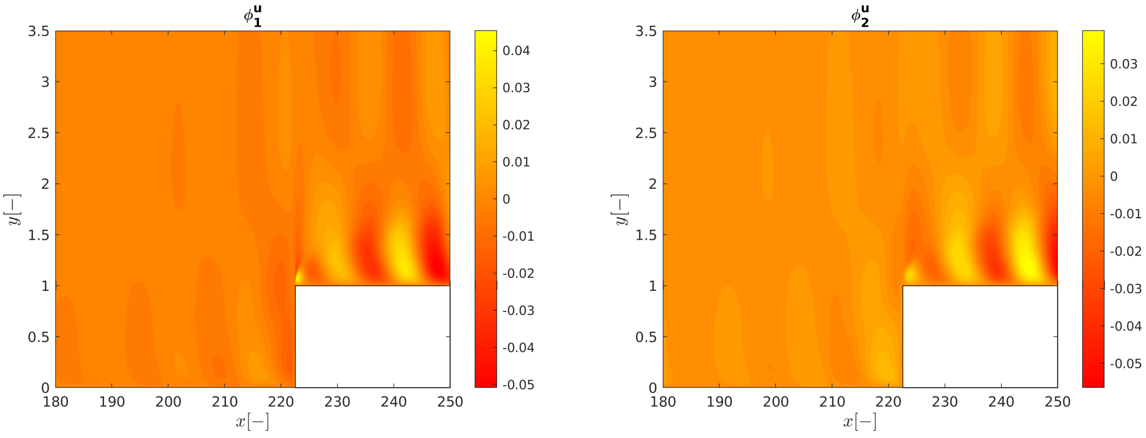
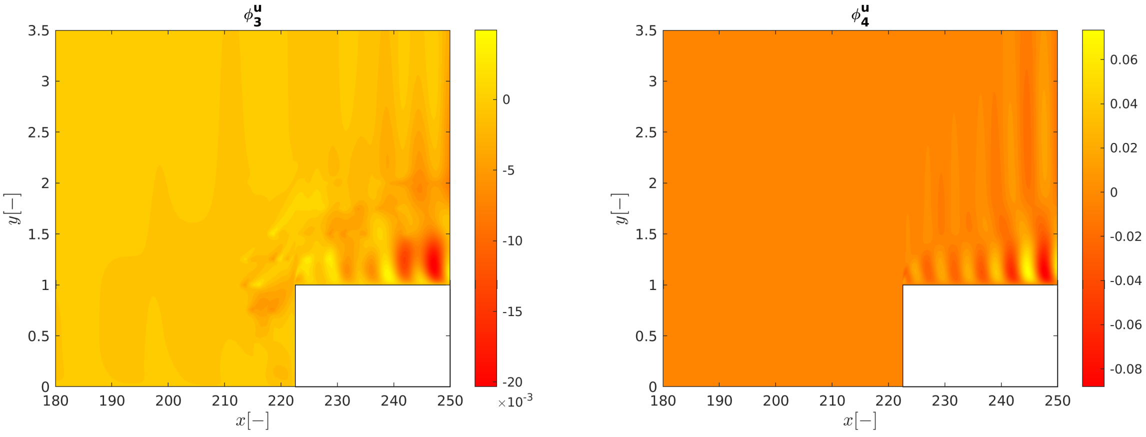


Figure 5.11: Amplitudes of the modes plotted in figure 5.12.



(a) Mode 1

(b) Mode 2



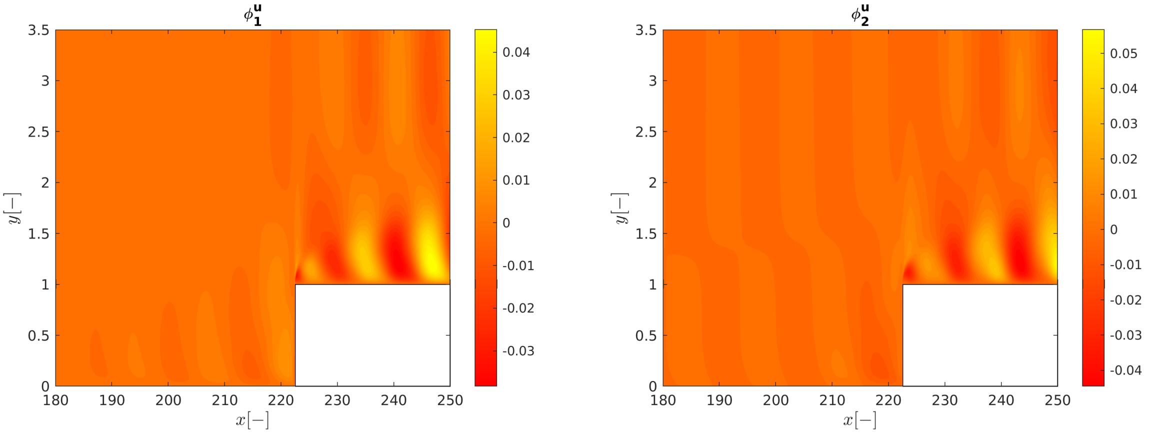
(c) Mode 3

(d) Mode 4

Figure 5.12: Plot of the GOROM modes of case C. The modes originate from the x-velocity.

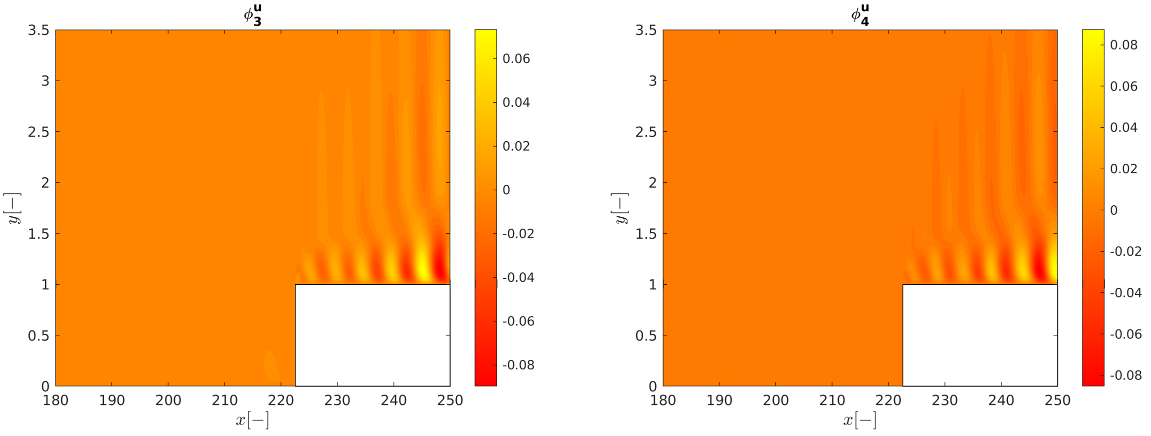
Case H has a top boundary placed further away, which results in the spots not being present in the mean-flow distortion. The mode shapes (figure 5.13) and amplitudes (figure 5.14) show some interesting behaviour of the third and fourth mode. On the surface, mode 3 does not seem to be periodic. However, when a Fourier transform of the amplitude is performed in figure 5.15a, it can be seen that the mode amplitudes are a superposition of the fundamental wave and first harmonic. The same process can be applied to the amplitudes of mode 4 in figure 5.15b. Again, a fundamental and first harmonic component are present. The fundamental component of the amplitudes from mode 3 and 4 have a different amount of energy and are phase shifted by 180°, similar to the results from case C. The fundamental component of mode 4 from case H is more pronounced than the fundamental component from case C however.

The presence of these fundamental components of the amplitudes in the shapes that have a spatial orientation similar to that of the first harmonic structure seems to suggest the presence of a structure with a convection velocity higher than that of the main wave. Problematic however, is that the time behaviour is not necessarily shifted by 90° and the discrepancy in the amount of energy in the mode is equally not convincing. No new conclusions or insights can thus be obtained by inspecting the GOROM modes and their amplitudes for these particular cases that can not be obtained by inspecting the POD modes.



(a) Mode 1

(b) Mode 2



(c) Mode 3

(d) Mode 4

Figure 5.13: Plot of the GOROM modes of case H. The modes originate from the x-velocity.

5.4. Concluding Remarks

The increase in numerical accuracy of the ROM constructed using GOROM shows the potential of this GOROM method. Quite a few different minima are found in close proximity to each other through the perturbation parameter ϵ. The set of modes found in these minima did not provide a set that is any more interpretable than the POD modes, yet the presence of the many minima do lower the confidence that the global minimum of the objective functional has been found. This is relevant for the discussion, as it might be possible that a set of modes exist that shows a value for the objective functional lower than the value found in any of the cases here *and* provide a set of modes that lead to some new insight into the physics of the problem.

If no new physics are captured within the basis however, the increase in numerical accuracy remains

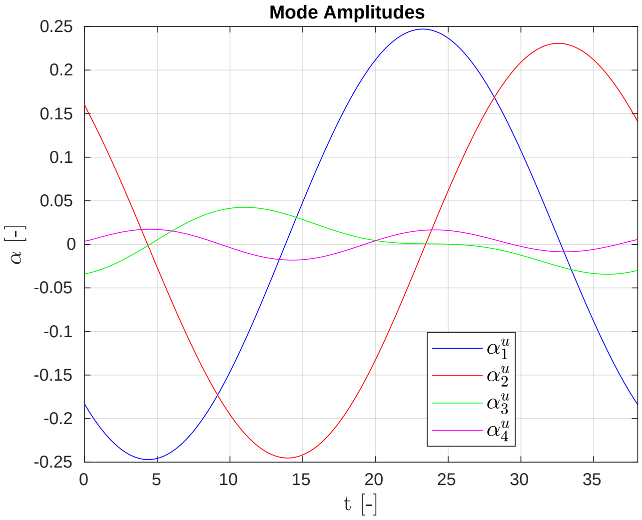
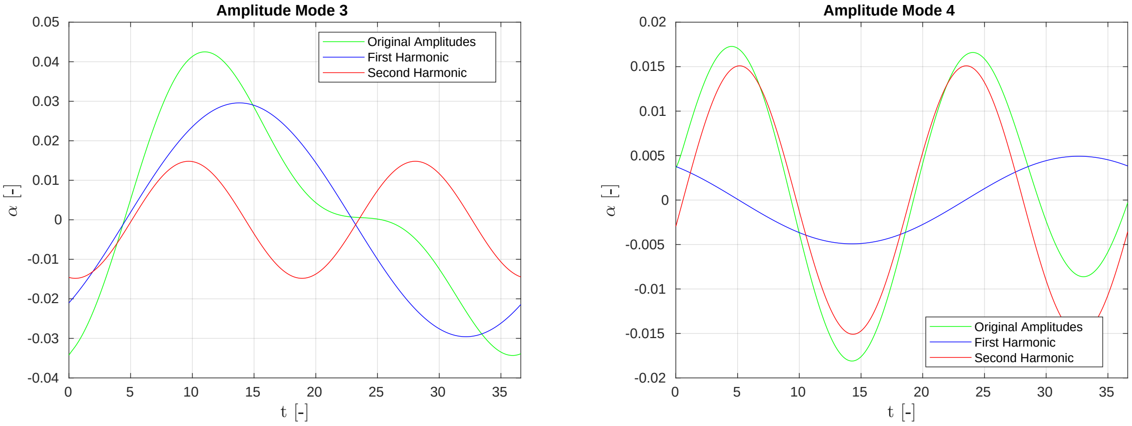


Figure 5.14: Amplitudes of the modes plotted in figure 5.13.



(a) Mode 3

(b) Mode 4

Figure 5.15: Amplitudes of the mode 3 and 4 from case H decomposed into their harmonic components.

unexplained. The subgrid-scale (SGS) model may provide the explanation for the increase in numerical accuracy. The SGS model attempts to model the effect of the truncated scales on the resolved scales when truncating the POD modes. In doing so, the SGS introduces some model error. The current hypothesis is that the GOROM technique is correcting for this model error by slightly modifying the modes. As the perturbations grow largely linear on top of the base flow, the correction made do not considerably change the result, yet do result in a slight increase in numerical performance.

6

Computational Cost and Scaling

This chapter evaluates the computational cost of the problem and investigates the scaling of the parallel implementation. The parallel implementation is achieved by dividing the cost of the spatial integrals over the processors. The case that is investigated to quantify the cost, is the step case with the top boundary at $y = 100$ and a streamwise domain of $x \in [0, 380]$. This totals 2,801,040 domain integration points. As the problem does not change when increasing the number of processors, this chapter will exclusively deal with *strong scaling*. All calculations in this chapter are performed on a Beowulf cluster on one node. This node has a maximum of 20 Intel(R)Xeon(R) E5-2670v2 processors available.

Furthermore, this chapter also will treat the cost of adding more basis functions, both primary and secondary. Shajahan (2016) has already performed such an investigation for the 1D incompressible Navier-Stokes. However, the implementation here is different for two distinct reasons. The first is the fact that the basis functions and reference data are discrete rather than continuous, adding cost for interpolation. The second difference results from the fact that prior to the simulation of the state, adjoint and gradient the implementation here will save value of the basis functions and their first and second gradients at the integration points into memory. No new memory needs to be allocated nor do the basis functions need to be evaluated during any of the simulations, leading to a substantial increase in performance.

The remainder of this chapter is structured as follows. First the parallel scaling of the problem will be discussed, after which the influence of the primary and secondary basis functions are treated. These sections can be used to estimate the order of magnitude of the computational cost for further problems.

6.1. Parallel Scaling

The evaluation time of the total gradient[1] shows a linear decrease as the number of cores increases, as seen in figure 6.1. Do note that the magnitudes of the different lines in figure 6.1 are not directly comparable to each other, as changing the number of primary basis functions will affect the number of degrees-of-freedom in the system. Changing the number of degrees-of-freedom in a system of nonlinear equations may lead to an increased or decreased number of required iterations for one timestep, skewing the comparison. This is clearly very dependent on the physical problem that is solved and thus no definitive conclusions can be made from a comparison of the total evaluation time for different cases.

The speedup ratio[2] for the total gradient and the three components can be found in figure 6.2. Interestingly the speedup for 2 & 3 primary basis functions seems to be higher than the ideal speedup. The components causing this speedup are the state and adjoint rather than the gradient. Furthermore, the speedup seems to jump around four cores. This behaviour can be explained by considering the cache available for each core. For the processors used in these calculations this is 6.4 gigabytes. The memory cost for this particular case is around 20 gigabytes. Aside from the computational cost, the majority of the memory cost will be divided over the cores as well. Thus, from four cores onward, the memory required for each processor fits inside the cache of the processor causing a jump in performance.

This hypothesis is confirmed once considering the total speedup between four and twenty cores. For 2 ϕ, the total evaluation time is reduced by a factor 4.9 and for 3 ϕ this factor is 4.4, both below the ideal speedup.

---

[1] i.e. the cost of the state, adjoint and gradient integrals combined

[2] That is, the evaluation time for a given number of cores divided by the evaluation time for one core.

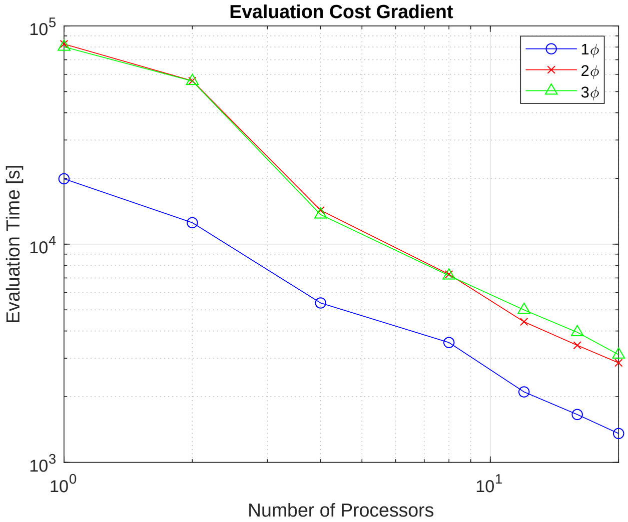


Figure 6.1: Evaluation time of the total gradient plotted against the number of cores used for several numbers of primary basis functions. Note that the problem does not change as the number of cores increases, thus strong scaling is investigated here.

The scaling for 1 ϕ is also interesting. From around 12 cores the performance of the state simulations seems to start to drop off, implying that the processors are waiting for the other processors to complete. The adjoint and gradient still continue along the same slope however, implying a better scaling for these two components.

6.2. Influence of the Number of Basis Functions

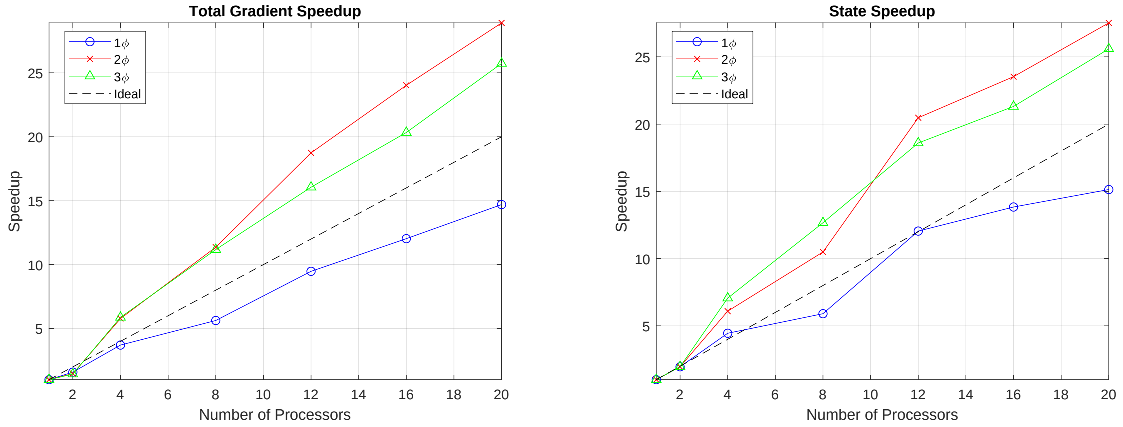
The computational cost is also dependent on the choice of basis. That is, increasing the number of primary and secondary bases will increase the computational cost associated with the problem. The amount by which changing these parameters influences the cost of the integration, is the subject of this section. First the influence of the number of primary bases is discussed, after which the influence of the number of secondary bases is treated. All simulations in this section will be performed using 20 cores on the aforementioned Beowulf-cluster.

6.2.1. Influence of the number of ϕ

To test the increase in computational cost for different numbers of primary basis functions, the cost of evaluating the gradient terms is estimated. The state and adjoint are left out of this comparison as the number of degrees-of-freedom change as ϕ changes, and is thus very case specific. This is not true for the gradient, as this is simply an integral of space and time without any iterations, which allows for a much fairer comparison.

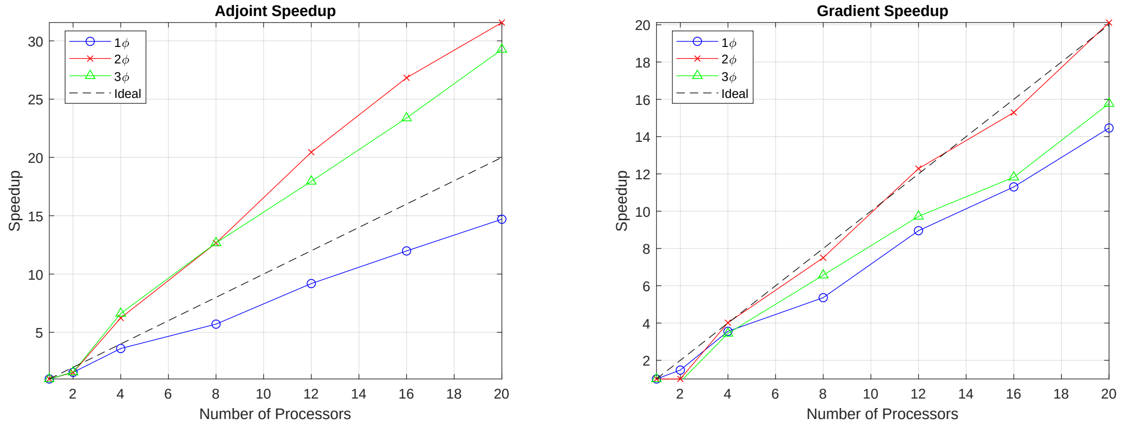
When inspecting the terms in the gradient expressions, some terms show sums over the number of primary basis functions. Hence, the number of primary basis functions will increase the number of terms to integrate per optimization coefficient and thus also the amount of operations that need to be performed to find the gradient for *1* coefficient. Furthermore, the increase in the number of primary basis functions will also increase the number of secondary basis functions, increasing the *number* of optimization coefficients. As these two effects amplify each other, a nonlinear growth in computational cost is expected when increasing the number of primary basis functions.

Inspecting figure 6.3, the nonlinear trend can indeed be observed. That is, when the first four data points are used to construct a linear regression line and extrapolated to 9 primary basis functions, the striped red line emerges. As can be seen, the slowdown ratio diverges from this linear regression line as the number of primary basis functions increases. It should be noted that this nonlinear effect seems to be relatively weak however.



(a) Total Gradient

(b) State



(c) Adjoint

(d) Gradient

Figure 6.2: Speedup of the gradient, including the individual components that make up the total gradient. This investigation uses *strong* scaling.

Figure 6.4 shows a relative breakdown of the components making up the gradient integral. The gradient is divided up into four components:

1. **PDE Terms:** These terms are all the terms related to the model equations (i.e. equations 3.79-3.89 & 3.97-3.102)
2. **Goal Terms:** These terms are related to the total error estimation (i.e. equation 3.78 & 3.96)
3. **Orthonormality Terms:** As the name suggest, these terms are related to the orthonormality of the modes (i.e. equation 3.91 & 3.104)
4. **Initial Condition Terms:** These terms estimate gradient of the initial conditions of the problem (i.e. equations 3.90 & 3.103)

It was found that the cost for the initial conditions and orthonormality terms are negligable compared to the PDE and goal terms. This is not unsurprising as both the initial conditions and orthonormality terms are not integrated in time. Since these two terms are substantially below 1% of the total integration time, they are not plotted in figure 6.4.

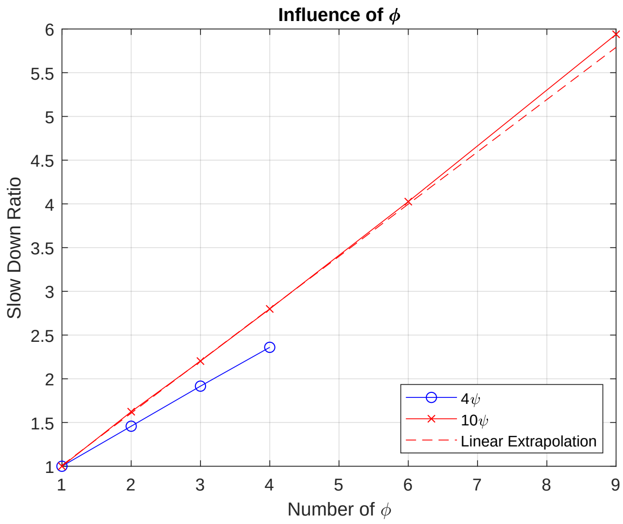


Figure 6.3: Slowdown ratio plotted against the number of primary basis functions for several choices of the number of secondary basis functions.

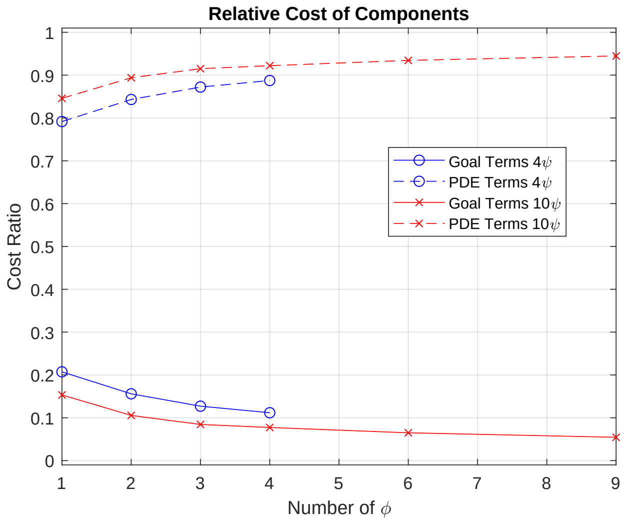


Figure 6.4: Relative computational cost of the different components making up the gradient integral

The trend of the cost for the PDE terms increasing relative to the cost of the goal terms shown in figure 6.4 can be explained by the considering the reference data. The reference data is defined on a discrete grid. Thus interpolation is required to estimate the value of the reference data at a given coordinate. This interpolation is only done once per timestep per integration point for *all* the optimization coefficients. Thus, as the number of optimization coefficients increases, the cost for interpolation does not. The amount of terms integrated does increase as the number of optimization coefficients increases, causing the asymptotic behaviour observed in figure 6.4.

6.2.2. Influence of the number of ψ

To estimate the influence of the number of secondary basis functions the *total* gradient estimation time can be used again, since the number of degrees-of-freedom in the system does not change, nor does the shape of the primary basis functions. The slowdown ratio plotted against the number of secondary basis functions for two different numbers of primary basis functions can be found in figure 6.5.

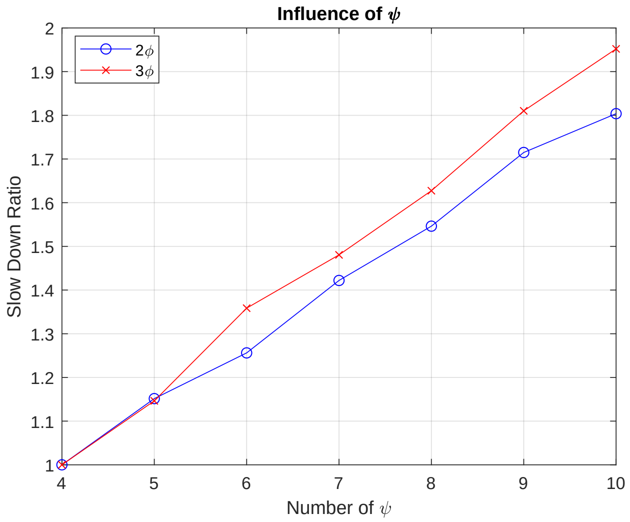


Figure 6.5: Slowdown ration plotted against the number of secondary basis functions.

Figure 6.5 shows a linear increase in computational cost when increasing ψ for a given ϕ. The slope of this increase is dependent on the number of ϕ used. The explanation for this difference in slope can be found in the number of optimization coefficients. That is, when one secondary basis function is added for for 2 primary basis functions, an additional 2 optimization coefficients will be added. When 3 primary basis functions are considered however, 3 optimization coefficients are added. The increase in cost is still linear, yet larger due to the extra added optimization coefficients.

7

Conclusion and Recommendations

7.1. Conclusions

This work extended the semi-continuous formulation of the goal-oriented, reduced-order modeling technique to the incompressible Navier-Stokes equations with arbitrary dimensions and inhomogeneous boundary conditions. This work verified the expressions using manufactured solutions for a 1D, 2D and 3D case, with both a linear and nonlinear goal function. Furthermore, this work applied the GOROM technique to two cases of a 2D transitional boundary layer on a flat plate by recombining a set of modes obtained through the POD into a new, more optimal basis.

The first of the two cases investigated DNS data of a Tollmien-Schlichting wave convecting over a flat plate. As expected, the POD modes captured the most relevant structure in the first two modes. Optimizing starting from these modes lead to a minor numerical improvement, but did not yield considerably different modes. This is attributed to the low amount of content available in the reference data. Furthermore, the behaviour of the structure on top of the base flow can mostly be described using linear dynamics. Thus truncating the POD basis does not lead to notable errors in the representation of the flow field constructed using a ROM of the first two modes.

Still, some optimizations did reach a different set of modes compared to the POD modes. This was only achieved after starting the optimization *away* from the POD modes however. Moreover, the resulting modes reside in a local minimum of the objective functional, as the temporal behaviour of the modes did not show signs of long-term accurate behaviour and the objective functional was higher compared to the one obtained through the POD modes.

The second case considered DNS data of the same boundary layer on a flat plate. This case included a forward-facing step however, complicating the flow problem. The GOROM technique was applied to two datasets of this problem. Both datasets model the same physical problem, but differ in the height of the domain. The lower domain case showed some strange topology occurring in the mean-flow distortion, which is attributed to either purely model error, or an interaction between model and discretization error. Nevertheless, the GOROM technique was applied and found that starting the optimization from the POD modes did not yield any improvement in the objective functional. Contrary to the flat plate case, an improvement was found when using a set of modes started away from the POD modes. This improved set of modes did not yield any more insights into flow physics, yet it does show the applicability of the GOROM technique to provide numerically more accurate ROMs.

The case with a higher domain only showed minor improvements in the objective functional over the POD when using the same goal function as the POD. This minor improvement is attributed to the model constraint. It is postulated that algorithm slightly modifies the modes to reduce the model error introduced by the SGS model. Three different starting conditions were applied to investigate the shape of objective functional for this combination of goal function and domain height. Each starting condition converged to a different minimum. The case with the lower domain height showed similar behaviour. These results imply that the shape of the objective functional is relatively complex and shows several minima in close proximity to each other.

The GOROM technique did make a much more considerable improvement in accuracy when a different goal function was used. Both starting from the POD modes and a perturbed set of modes yielded a ROM which is more accurate. The largest decrease in objective functional was observed in the perturbed case. Furthermore, both starting conditions converged to a different minimum, insinuating a complex objective functional for this choice of goal function as well. The set of modes did not provide new insights, as the behaviour of the amplitudes and shape of the newly found modes did not show signs of what would be expected.

Finally, the scaling of the parallel implementation of the GOROM technique was tested. It was found that the evaluation of the components scale linearly with the number of cores used for a 2D case with 2,801,040 integration points, at least until 20 cores. Some of the terms showed better than ideal scaling, which is attributed to a jump in performance when the required memory per core fits inside the cache of the processor.

Finally, the increase in computational cost was estimated for the same case when increasing the number of primary and secondary basis functions. It was found that the computational cost of the gradient is nonlinearly dependent on the number of primary basis functions and linearly dependent on the number of secondary basis functions.

7.2. Discussion

The results highlight one of the deficiencies of the GOROM technique. That is, the objective functional might hit a local minimum. Specifically for the flat place case this is quite detrimental for the set of modes that result. That is, the POD provides a more optimal set of modes for much less computational cost. Furthermore, the POD is much simpler to perform, as the tools to perform the decomposition are readily available in most popular programming languages. This flat plate case thus provides a very good example that a clear motivation needs to present before applying the GOROM technique.

The optimizations of the step case similarly found several minima of the objective functional in close proximity. The perturbation parameter ϵ was used to enforce different starting conditions in a relatively arbitrary fashion. There many more possibilities to enforce different starting conditions and thus possibly find different minima. The fact that all practical cases considered in this work converged to different minima is evidence that the optimization method applied is not sufficient to confidently say that a global minimum has been found.

Finally, a limitation is present in the reference data. That is, a relatively low number of snapshots per period and a low number of periods available from the DNS data results in problems when simulating higher harmonics. These problems are particularly pronounced for structures with very low amplitudes, as predicated by the results from the first harmonic structure in the flat plate case.

7.3. Recommendations & Future Work

Several recommendations can be made for further applications. Firstly, surrogate models of the objective functional can be used to approximate the shape of the objective functional and more easily be able to find the global minimum, which would overcome the observed problems with local minima in this work. This does come at additional computational cost however.

The limitation regarding the reference data can be solved by adding additional snapshots and periods to the reference data. Alternatively, it is possible to reconstruct the instantaneous flowfield from the Fourier-transformed reference data. This requires less memory and would allow for longer simulation times as well. A downside is that an additional assumption is made that the first few harmonic Fourier components cover all the relevant content.

Furthermore, the POD has been used as a benchmark in this work. However, other decomposition techniques more applicable to these problems do exist. The dynamic mode decomposition and spectral proper orthogonal decomposition have been named as possibilities earlier in this work. In the future, the GOROM modes could be compared to modes from these decomposition techniques to further establish the applicability of the GOROM technique.

Another possible application of the GOROM method can be found in so-called active laminar flow control techniques. These techniques actively try to lower the growth rate of instabilities in the boundary layer, delaying turbulent breakdown and extending the range of laminar flow over this same surface. The GOROM modes could provide a basis for a ROM used to inform the actions of the control scheme. As also observed in this work however, the online evaluation of the nonlinear terms for a ROM of these types of problems can still be relatively expensive. To reduce the online computational cost for the nonlinear terms, methods like discrete empirical interpolation could be used (see e.g. Chaturantabut and Sorensen (2009)).

Furthermore, due to the modeling constraint this method is much more applicable for cases with more nonlinear interactions in the perturbation field. An example of such a case would be any of the cases considered here with a wavepacket introduced at the inflow. The additional nonlinear interactions of the perturbation field will likely lead to a set of POD modes with a much less sharp drop-off of the singular values, increasing the error when using a truncated set of POD modes.

Casacuberta et al. (2021) is investigating a 3D transitional boundary layer with a forward-facing step where the transitional behaviour is dominated by a different type of instability than considered in this work. This so-called *crossflow instability* is more nonlinear in nature, which is a good motivation to study the problem with the GOROM technique. The terms presented in this work are verified for 3D problems, thus a logical extension of this work is to investigate this more complex, nonlinear boundary layer. However, as seen in chapter 6, the computational cost of evaluating the gradient is quite considerable. Moreover, the conjugate-gradient method requires two gradient calculations per iteration and a few iterations to converge. One iteration of the trust-region algorithm thus already requires several calculations of the gradient. Generally speaking, the optimizations for the cases with the step required around 70 hours of walltime to converge using 20 cores. Assuming the same number of iterations are required to converge for similar cases, a total gradient evaluation time of roughly 2,000 seconds should be aimed for.

The refined mesh from Casacuberta et al. (2021) contains 200 million cells, roughly a factor of a 100 more than the case considered here. Making the assumption that the number of timesteps does not need to be altered and the computational cost scales linearly with the number of integration points relative to the 2D case considered in this work[1], the cost for solving the gradient with 1 core can be estimated to be roughly $8.0 \cdot 10^6$ seconds. Assuming the scaling observed in chapter 6 for 3ϕ can be extrapolated to this case, this would result in a little over 3,000 cores to have a gradient update every 2,000 seconds and make this case computationally tractable. According to the author of Casacuberta et al. (2021), data with a coarse mesh of 12 million cells is also available. A final optimization should be performed on the 200 million cell mesh, however it is recommended that the coarse data is to be used first. This makes it possible to test for different parameters and to see if it is even useful to apply the GOROM technique to this case before investing the substantial computational resources on the fine mesh.

Finally, to select a machine architecture the amount of memory required is of importance. It was found that the bulk of the memory requirement comes from the following three sources:

1. Loading the reference data into memory
2. Loading the discrete basis functions into memory
3. Saving the value of the basis functions and their first & second gradients at the integration points

The first two points in this list can easily be estimated for any given case as this information should be readily available. Within the current infrastructure, the memory requirement for the third and final point can be estimated to be:

$$\begin{aligned}\text{Required Memory} &= \left(1+\text{nDim}+\frac{\text{nDim}\cdot(\text{nDim}+1)}{2}\right)\cdot\text{nVar}\cdot\text{nIP}\cdot(N+M)\cdot\text{d2M}+6\text{nIP}\cdot\text{i2M}\\ &= 40\cdot\text{nIP}\cdot(N+M)\cdot\text{d2M}+6\cdot\text{nIP}\cdot\text{i2M} \qquad (7.1)\end{aligned}$$

where $nDim$ is the number of dimensions of the flow problem (i.e. 3 in this case), $nVar$ is the number of variables (i.e. 4 in this case), nIP is the number of integration points used to integrate the weak forms, N is the number of primary basis functions, M is the number of secondary basis functions and $d2M$ & $i2M$ are the memory requirement for a double and integer value respectively.

---

[1] This assumption is faulty due to the fact that adding another dimension increases the number of terms evaluated as well. This is somewhat compensated by the fact that the number of integration points for the 2D case is rounded down for this analysis.

A

Additional Considerations Stability Theory

This appendix is an additional investigation into the literature of stability theory. In what follows, this work follows Schmid and Henningson (2001) unless otherwise specified. This appendix starts by considering the nonlinear perturbation equations as seen below:

$$\frac{\partial u^i}{\partial t} = -U_b^j \frac{\partial u^i}{\partial x^j} - u^j \frac{\partial U_b^i}{\partial x^j} - \frac{\partial p}{\partial x^i} + \frac{1}{Re}\nabla^2 u^i - u^j \frac{\partial u^i}{\partial x^j} \tag{A.1}$$

$$\frac{\partial u^i}{\partial x^i} = 0 \tag{A.2}$$

A.1. Orr-Sommerfeld Equations

The simplifications applied to arrive at the Orr-Sommerfeld (OS) equations are the following. First of all, the nonlinear perturbation term $u^j \frac{\partial u^i}{\partial x^j}$ is dropped from equation A.1 as the perturbations are assumed negligible compared the base flow. Next, the base flow is assumed parallel (that is: $\mathbf{U_b} = \left[U_b(y), 0, 0\right]$). Note that when considering a spanwise invariant flow, it is also possible to include a $z - component$ in the base flow of the form $W_b(y)$. For the sake of simplicity however, this is not included here. From these simplifications, the following fourth-order differential equation for the y perturbation velocity is recovered:

$$\left[\left(\frac{\partial}{\partial t} + U_b \frac{\partial}{\partial x}\right)\nabla^2 - \frac{d^2 U_b}{dy^2}\frac{\partial}{\partial x} - \frac{1}{Re}\nabla^4\right] v = 0 \tag{A.3}$$

The normal vorticity is defined as:

$$\eta = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \tag{A.4}$$

The differential equation describing the normal vorticity can be found to be:

$$\left[\frac{\partial}{\partial t} + U_b \frac{\partial}{\partial x} - \frac{\nabla^2}{Re}\right]\eta = -\frac{dU_b}{dy}\frac{\partial v}{\partial z} \tag{A.5}$$

The final step in obtaining the OS-equations is to assume a solution of the following form:

$$v(x, y, z, t) = \hat{v}(y) e^{i(\alpha x + \beta z - \omega t)} + c.c. \tag{A.6}$$

$$\eta(x, y, z, t) = \hat{\eta}(y) e^{i(\alpha x + \beta z - \omega t)} + c.c. \tag{A.7}$$

where α, β and ω are the streamwise wavenumber, spanwise wavenumber and frequency of the perturbation respectively and $\hat{v}$ and $\hat{\eta}$ are shape functions of the perturbations which are assumed to only change in the y-direction. Furthermore, the term $c.c.$ stands for complex conjugate. Substituting these into equation A.3 and equation A.5 yields the Orr-Sommerfeld equation for the wall normal velocity (equation A.8) derived by both Orr (1907) and Sommerfeld (1908), and the Squire equation (Squire, 1933) (equation A.9).

$$\left[(-i\omega + i\alpha U)\left(\frac{\partial^2}{\partial y^2} - k^2\right) - i\alpha U'' - \frac{1}{Re}\left(\frac{\partial^2}{\partial y^2} - k^2\right)^2\right]\hat{v} = 0 \tag{A.8}$$

$$\left[(-i\omega + i\alpha U) - \frac{1}{Re}\left(\frac{\partial^2}{\partial y^2} - k^2\right)\right]\hat{\eta} = -i\beta U_b'\hat{v} \tag{A.9}$$

where $k = \sqrt{\alpha^2 + \beta^2}$. Note that it is also possible to derive the Orr-Sommerfeld equations independently for the spanwise and freestream perturbation by following a similar process for the spanwise perturbation w and finding the streamwise perturbation u using the continuity equation (c.f. Boiko et al. (2012)). Note also that the operators acting on $\hat{v}$ and $\hat{\eta}$ are linear, meaning that two (or more) perturbations can develop independently from each other, which is a consequence of neglecting the nonlinear $u^j \frac{\partial u^i}{\partial x^j}$ term in equation 1.6.

The general approach to solve the OS equations is not as relevant for the remainder of this work. Therefore, the interested reader can find a discussion of the approach, as well as a few other notes on stability theory in appendix A.2.

A.2. Solving the Orr-Sommerfeld Equations

Usually either the spatial or temporal approach is taken to solve the OS equations. The first subsection will discuss these two approaches in more detail. The second subsection briefly treats the concept of inviscid & viscous instabilities. Finally, the third subsection connects the OS analysis to a popular engineering technique used to estimate the transition point of an airfoil.

A.2.1. Spatial & Temporal Approach

In solving the OS equations, usually either the *temporal* or *spatial* approach is taken. The temporal approach assumes that the wavenumbers (α and β) are known and are real-valued. The OS-equations can then be solved as an eigenvalue problem for the complex-valued frequency ω. The real part of ω (denoted by ω_r) dictates the value of the frequency of the perturbation, whereas the imaginary part (denoted by ω_i) dictates the growth rate of the perturbation in time. From equation A.6 and equation A.7 it can be seen that when $\omega_i = 0$, the perturbation neither grows or decays. This is called a *neutral perturbation*. When $\omega_i < 0$ the perturbation decays, which is called a *decaying perturbation*. And finally, in the case $\omega_i > 0$, the perturbation grows in time, which is fittingly called a *growing perturbation*.

These classifications can be used to construct a so-called *neutral stability curve* for a given base flow, an example of which can be seen in figure A.2. The OS equations are solved for a range of spatial wavelengths and Reynolds numbers. The resulting imaginary parts of the temporal instabilities are used to classify the nature of each combination. A curve can then be constructed in the α, Re-plane for which $\omega_i = 0$ (when a spanwise wavenumber is included, the domain should be extended with a wavenumber β). This is the neutral stability curve which separates the region of unstable perturbations from the region of stable perturbations.

The spatial approach reverses this analysis and finds the imaginary spatial growth rate, given a real-valued temporal frequency. The resulting imaginary wavenumber uses the same classification of neutral, decaying and growing perturbations, although the sign is switched and the perturbations grow in space rather than in time.

The approach that best fits the problem of interest can be chosen. It should be noted however, that perturbations can grow in both space *and* time. Thus, an appropriate reference system needs to be chosen as well when deciding on the approach to take.

Furthermore, solving for the spatial wavelengths (α) is more difficult due to the nonlinear appearance in the OS-equations (see equation A.6). Gaster (1962) introduced a transformation allowing the problem to be formulated as a temporal stability problem. The transformation can then recover the spatial growth rates from the temporal rates as long as the growth rates are small.

Finally, the concept of convective and absolute instabilities are summarized in figure 1.2 from Boiko et al. (2012) which is reprinted in figure A.1. An absolute instability grows in time in the entire domain, as seen in figure A.1a, which is analogous to temporal growth. A convective instability grows while it is convected with the flow, as seen in figure A.1b, which is analogous to spatial growth. One might assume that, when considering a convective instability, there is no downstream influence of a perturbation that has grown upstream. This assumption leads to neglecting the elliptic terms in equation 1.6 and can be used to construct the Parabolized Stability Equations which are discussed in the main body of work.

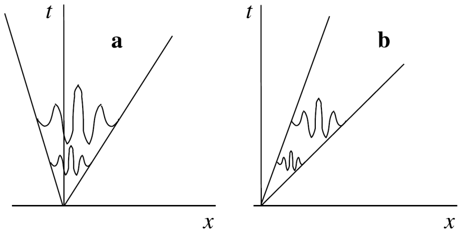


Figure A.1: Graphical representation of an absolute and convective instability, from Boiko et al. (2012)

A.2.2. Inviscid & Viscous Instabilities

The OS-equations include a term that is inversely proportional to the Reynolds number. In practice the Reynolds number can be very high, which can be used as a motivation to simplify equation A.3 and equation A.5 by $Re \to \infty$. In fact, when doing so the inviscid Rayleigh equations can be recovered (Rayleigh, 1880). Rayleigh showed that the presence of an inflection point in the base flow $\left(\frac{\partial^2 U}{\partial y^2} = 0\right)$ is a necessary condition for a perturbation to grow, to which Fjørtoft (1950) added an additional necessary condition that the inflection point needs to be maximum of the spanwise mean vorticity. Regardless, the presence of an inflection point in a base-flow profile is a strong indication of growth of a perturbation.

The instabilities created through these inviscid mechanisms are fittingly called *inviscid instabilities*. Although maybe counterintuitive, viscosity can have a destabilizing effect on perturbations that are stable according to the inviscid equations, which are called *viscous stabilities* (Reed et al., 1996).

A.2.3. e^N Method

The LST forms the basis of the most popular transition prediction method called the e^N method developed by Ingen (1956). The idea behind the method is to relate the ratio of the amplitude of a range of growing eigenmodes to their corresponding initial amplitudes using the N-factor:

$$N = ln\left(\frac{A}{A_0}\right) \tag{A.10}$$

If any of the amplitudes of the eigenmodes is above a certain critical value of N, a boundary layer is considered to transition at that location. The value of this maximum N is found using empirical relations.

A.3. (Nonlinear) Parabolized Stability Equations

In deriving the OS-equations in section 1.2.1, the base flow is assumed to be parallel. This implies that the velocity component V and possibly W of the base flow are only dependent on the wall-normal direction, and that there is no base-flow component in the y direction. In other words, streamwise gradients of the base flow are neglected. Some situations require these streamwise gradients to be taken into account however, which has led to the development of the Parabolized Stability Equations (Herbert, 1997). These equations assume the base flow to be of the from:

$$\mathbf{U_b} = \left[U_b(\xi, y,\zeta), \epsilon V_b(\xi, y,\zeta), W_b(\xi, y,\zeta)\right] \tag{A.11}$$

where ϵ is a small parameter of the order $\epsilon = \mathcal{O}(Re^{-1})$, $\xi = \epsilon x$ and $\zeta = \epsilon z$. Thus the assumption is that the base flow varies slowly in the streamwise and spanwise directions, and that the wall-normal velocity is small compared to the other two components.

Neglecting nonlinear terms and assuming spanwise invariant flow, the ansatz previously seen in equation A.6 is extended as seen in equation A.12 (Westerbeek, 2020). Note that here the q denotes a general perturbation component.

$$q(x, y, z, t) = \hat{q}(x, y)e^{i\left(\int_{x_0}^{x} \alpha(x^*)dx^* + \beta z - \omega t\right)} + c.c. \tag{A.12}$$

Assuming that the shape function slowly varies in the streamwise direction (i.e. $\hat{q}(x, y) = \hat{q}(\xi, y)$) can together with an order of magnitude analysis be used to motivate neglecting the elliptic terms. The slow variation of the shape function is enforced through the normalization condition, which is defined as:

$$\int \hat{q}^{\dagger} \frac{\partial \hat{q}}{\partial y} = 0 \tag{A.13}$$

where the dagger denotes the complex conjugate. With the elliptic terms neglected, the condition for a perturbation at the inflow of the domain can be prescribed, which can consequently be marched in the streamwise direction.

An even higher-fidelity model of perturbation growth is obtained when the nonlinear terms in equation 1.6 are not neglected. This does come with a cost: the growth of perturbations can not be individually analyzed and several harmonics of the fundamental perturbation need to be accounted for at the same time. The perturbation ansatz seen before extends further to equation A.14 (Westerbeek, 2020).

$$q(x,y,z,t) = \sum_{m=-M}^{m=M} \sum_{n=-N}^{n=N} \hat{q}_{m,n}(x,y) e^{\int_{x_0}^{x_e} \alpha(x)_{m,n} dx + \beta_m z - \omega_n t} \tag{A.14}$$

Note that the addition of the complex conjugate is added by summing over the negative wavenumbers as well. Furthermore, the summations should extend from $-\infty$ to ∞, which in practice is not possible due to the discrete representations necessary for any practical application. Therefore, usually the finite summations are shown.

Finally, although not explicitly shown in the above equation, a slowly varying base flow and shape function are also assumed, the latter implied through the normalization condition.

A.4. Additional Considerations Tollmien-Schlichting Problem

The main body of this work leaves out a few important results from investigations into transitional boundary layers that are not as relevant for the work performed here, yet are important when discussing Tollmien-Schlichting problem in general. The first of which is the Squire theorem, discussed here in the first subsection. The second topic is the neutral stability curve for a flat plate boundary layer, which is on the of the results that can be attributed to Tollmien (1929) and Schlichting (1933). Finally, the main body of the text only discusses primary instabilities in the context of path A from figure 1.1. However, a discussion of secondary instabilities and breakdown are required to encompass the full transition scenario of path A. This is subject of the last subsection.

A.4.1. Squire Theorem

Squire (1933) proved that for any 3D perturbation, there is a 2D perturbation that has a lower critical Reynolds number. As pointed out by Reed et al. (1996) however, this theorem only holds for incompressible, parallel flows near the critical Reynolds number. Nevertheless, this theorem can be used to motivate investigating a 2D case rather than a 3D case in a flat-plate boundary layer, as the 2D waves provide the more ‘critical’ case. Consequently, for the discussion surrounding the Tollmien-Schlichting problem, the spanwise wavenumber β is assumed zero.

A.4.2. Neutral Stability Curve

Tollmien and Schlichting managed to calculate the neutral stability curve for a Blasius boundary layer by solving an OS problem locally for increasing streamwise distances along the boundary layer. An example of a neutral stability curve using a temporal approach for a Blasius boundary layer can be found in figure A.2.

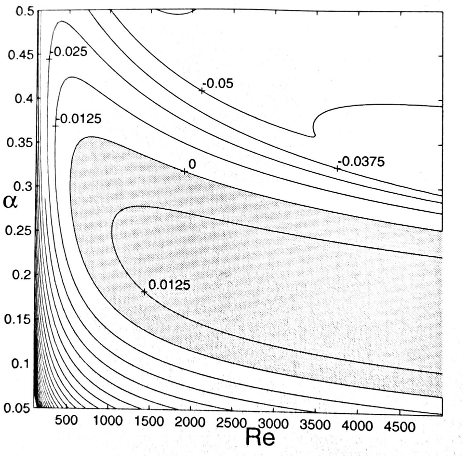


Figure A.2: Neutral stability curve of the Blasius boundary layer with contours of the growth rate. The shaded area has positive growth rates and are thus unstable. Taken from Schmid and Henningson (2001).

The streamwise distance x can be related to the boundary layer (or displacement) thickness and thus to the local Reynolds number for which the OS-equations is solved. This can all be used in a local streamwise Reynolds number (Re_x), which uses the x-location as the length scale. This means that the curve in figure A.2 can be interpreted as follows: a perturbation with a given frequency 'enters' the boundary layer at $Re_x = 0$. This perturbation then convects along the flat plate, increasing the Re_x (moving to the right in figure A.2). As Re_x is currently smaller than the critical Reynolds number for the given frequency, the perturbation decays. At some point the perturbation reaches a value for Re_x corresponding to the point for which $\alpha_i = 0$. At this point, the perturbation hits the neutral stability curve. Moving further to the right, the perturbation enters an unstable area of the boundary layer and starts to grow.

This growth is continued until a Re_x is reached that corresponds to $\alpha_i = 0$ again. Moving further to the right in figure A.2 now corresponds to entering a stable area. From here on out, the perturbation decays. The first line of the neutral curve that perturbations encounter is called *branch I* of the neutral stability curve. The second line is similarly called *branch II.*

A.4.3. Secondary Instabilities and Breakdown

Clearly the analysis above does not completely encompass the full transition scenario (as also seen in figure 1.1). If the analysis is followed, the perturbations grow between branch I and II and then decay further downstream to zero. The final mechanisms causing transition are not captured due to the validity of the assumptions made in the derivation. LST assumes that the perturbations grow by extracting energy from the base flow (Schmid & Henningson, 2001). The assumption of a parallel base flow implies that this energy extraction does not affect the base flow. This is only valid as long as the perturbations are indeed small. When the perturbation amplitudes reach high enough values, the perturbations can saturate and cause a 'new' base flow existing out of the original base flow and the saturated perturbations (Herbert, 1988). This new base flow can be unstable to other 3D perturbations that are present in the boundary layer but have previously not been unstable or negligibly unstable compared to the primary instabilities. These perturbations are usually referred to as *secondary instabilities.*

In case of a flat plate boundary layer with a sufficiently low disturbance environment, the 2D TS waves are modulated by a spanwise secondary instability whenever the amplitude of the TS reaches a value of around 1% of the freestream velocity (Herbert, 1988). These 3D modulations of the TS waves grow further into Λ-vortices and eventually lead locally to profiles with inflection points. These profiles are again unstable to high frequency perturbations that consequently lead to turbulent spots, finally resulting in fully turbulent breakdown (Herbert, 1988).

This transition scenario is usually referred to as *K-type transition* (Klebanoff et al., 1962) if the resulting Λ-vortices show an aligned pattern and *H-type* transition (Herbert, 1988) if the pattern is staggered. Usually, K-type transition is connected to the growth of a fundamental mode[1], whereas the H-type transition is connected to the growth of subharmonic modes[2] (Herbert, 1988). Which type of transition is dominant is dependent on the freestream conditions, as shown using flow visualization by Saric (1986).

---

[1] An oblique wave with the same streamwise wavelength and spanwise wavelength as the primary instability's streamwise wavelength, see the DNS results by Rist and Fasel (1995) for a detailed discussion

[2] An oblique wave with the same spanwise and halve the streamwise wavelength of the primary instability's streamwise wavelength

B

Derivations of the Adjoint and Gradient for the General Formulation

This appendix has the full derivations of the adjoint and gradient equations. The Lagragian constructed is:

$$\mathscr{L} = \int_\Omega \left(\frac{1}{2} \int_0^{t_f} E dt + \sum_{k=1}^{m} \int_0^{t_f} \lambda_k G_2^k dt + \sum_{k=1}^{m} \mu_k G_3^k \right) d\Omega + \frac{\beta}{2} G_1 \tag{B.1}$$

With:

$$E = \left(g - \hat{g}\right)^2 \tag{B.2}$$

$$G_1 = \sum_{i,j=1}^{m} \left(\delta_{ij} - \phi_i \phi_j\right)^2 \tag{B.3}$$

$$G_2^k = \phi_k \left(\mathcal{L}\left(\hat{u}_t\right) - f\right) \tag{B.4}$$

$$G_3^k = \phi_k \left(\hat{u}_{t_0} - u_{t_0}\right) \tag{B.5}$$

B.1. Derivation of the Adjoint

The adjoint equations are found by applying a small variation. The variation is applied on one of the temporal coefficients α, i.e.:

$$\alpha_q = \alpha_q^* + \epsilon \eta(t) \tag{B.6}$$

$$\dot{\alpha}_q = \dot{\alpha}_q^* + \epsilon \dot{\eta}(t) \tag{B.7}$$

where α_q^* is the value for α_q that extremises the Lagrangian. Note that it is assumed the model constraint only depends on the first time derivative at most. Using calculus of variations it can be argued that for α_q^* to be the optimum, it is necessary that:

$$\left.\frac{d\mathscr{L}}{d\epsilon}\right|_{\epsilon=0} = 0 \tag{B.8}$$

Using the chain rule, the total derivative of a function that depends on ϵ only through α_q and $\dot{\alpha}_q$ can be calculated to be:

$$\frac{dh(\alpha_q, \dot{\alpha}_q)}{d\epsilon} = \frac{\partial \alpha_q}{\partial \epsilon} \frac{\partial h(\alpha_q, \dot{\alpha}_q)}{\partial \alpha_q} + \frac{\partial \dot{\alpha}_q}{\partial \epsilon} \frac{\partial h(\alpha_q, \dot{\alpha}_q)}{\partial \dot{\alpha}_q} = \eta \frac{\partial h(\alpha_q, \dot{\alpha}_q)}{\partial \alpha_q} + \dot{\eta} \frac{\partial h(\alpha_q, \dot{\alpha}_q)}{\partial \dot{\alpha}_q} \tag{B.9}$$

where the final step comes from calculating the derivatives using Equation B.6 and Equation B.7. Applying this derivative to each term in equation B.1:

Goal function term:
The term that includes the goal function is:

$$\frac{d}{d\epsilon}\int_0^{t_f} E dt = \int_0^{t_f} \frac{\partial E}{\partial \alpha_q}\eta dt \tag{B.10}$$

Where the derivative with respect to $\dot{\alpha}_q$ is omitted by assuming the goal does not depend on a time derivative.

Model constraint:
The term that includes the model constraint is:

$$\frac{d}{d\epsilon}\int_0^{t_f} \lambda_k G_2^k dt = \int_0^{t_f} \left[\eta\frac{\partial(\lambda_k G_2^k)}{\partial \alpha_q} + \dot{\eta}\frac{\partial(\lambda_k G_2^k)}{\partial \dot{\alpha}_q}\right] dt \tag{B.11}$$

Using integration by parts, the second term in the above integral can be written as:

$$\int_0^{t_f}\left[\dot{\eta}\frac{\partial(\lambda_k G_2^k)}{\partial \dot{\alpha}_q}\right] dt = -\int_0^{t_f}\left[\frac{d}{dt}\left(\lambda_k\frac{\partial G_2^k}{\partial \dot{\alpha}_k}\right)\eta\right] dt + \lambda_k(t_f)\eta(t_f)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=t_f} - \lambda_k(0)\eta(0)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=0} \tag{B.12}$$

Substituting the result back into equation B.11 leads to:

$$\frac{d}{d\epsilon}\int_0^{t_f} \lambda_k G_2^k dt = \int_0^{t_f}\left[\frac{\partial(\lambda_k G_2^k)}{\partial \alpha_q} - \frac{d}{dt}\left(\lambda_k\frac{\partial G_2^k}{\partial \dot{\alpha}_k}\right)\right]\eta dt + \lambda_k(t_f)\eta(t_f)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=t_f} - \lambda_k(0)\eta(0)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=0} \tag{B.13}$$

Initial conditions:
The initial condition constraint leads to:

$$\frac{d}{d\epsilon}\left(\mu_k G_3^k\right) = \frac{\partial\left(\mu_k G_3^k\right)}{\partial \alpha_q}\eta\Big|_{t=0} = \mu_k\eta(0)\frac{\partial G_3^k}{\partial \alpha_q}\Big|_{t=0} \tag{B.14}$$

Where the second part of the total derivative is omitted since the initial condition does not depend on the time derivative of the state variables.

Combining Everything
Combining equations B.10-B.14 with equation B.1 results in equation B.15. Note that the orthonormality constraint G_1 does not show up, since the constraint does not depend on the temporal coefficients α and thus drops out during the differentiation.

$$\begin{aligned}\frac{d\mathscr{L}}{d\epsilon} = \int_\Omega\int_0^{t_f}\left[\frac{1}{2}\frac{\partial E}{\partial \alpha_q} + \sum_{k=1}^{m}\frac{\partial(\lambda_k G_2^k)}{\partial \alpha_q} - \frac{d}{dt}\left(\sum_{k=1}^{m}\lambda_k\frac{\partial G_2^k}{\partial \dot{\alpha}_k}\right)\right]\eta dt + \\ \sum_{k=1}^{m}\left(\lambda_k(t_f)\eta(t_f)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=t_f} - \lambda_k(0)\eta(0)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=0} + \mu_k\eta(0)\frac{\partial G_3^k}{\partial \alpha_q}\Big|_{t=0}\right) d\Omega = 0\end{aligned} \tag{B.15}$$

This relation needs to hold for an arbitrary η with arbitrary start- and endpoints. To be able to formalize a differential equation, conditions on the boundary terms in the final sum of the above expression need to be applied, that is:

$$\lambda_k(t_f)\eta(t_f)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=t_f} - \lambda_k(0)\eta(0)\frac{\partial G_2^k}{\partial \dot{\alpha}_q}\Big|_{t=0} + \mu_k\eta(0)\frac{\partial G_3^k}{\partial \alpha_q}\Big|_{t=0} = 0 \tag{B.16}$$

Collecting terms in front of $\eta(t_f)$ and $\eta(0)$ makes it is clear that the following conditions need to hold:

$$\lambda_k(t_f) = 0 \tag{B.17}$$

$$\mu_k = \lambda_k(0) \tag{B.18}$$

Which leads to the boundary terms being dropped in equation B.15 resulting in:

$$\frac{d\mathscr{L}}{d\epsilon} = \int_\Omega \int_0^{t_f} \left[\frac{1}{2} \frac{\partial E}{\partial \alpha_q} + \sum_{k=1}^{m} \lambda_k \frac{\partial G_2^k}{\partial \alpha_q} - \frac{d}{dt} \left(\sum_{k=1}^{m} \lambda_k \frac{\partial G_2^k}{\partial \dot{\alpha}_k} \right) \right] \eta \, dt \, d\Omega = 0 \tag{B.19}$$

Using the fact that η is an arbitrary function, the term in the square brackets in the above equation needs to be zero for this relation to hold. The resulting adjoint differential equation that is solved for $\lambda(t)$ becomes:

$$\int_\Omega \left[\frac{1}{2} \frac{\partial E}{\partial \alpha_q} + \sum_{k=1}^{m} \lambda_k \frac{\partial G_2^k}{\partial \alpha_q} - \frac{d}{dt} \left(\sum_{k=1}^{m} \lambda_k \frac{\partial G_2^k}{\partial \dot{\alpha}_k} \right) \right] d\Omega = 0 \tag{B.20}$$

With the initial conditions specified by equation B.17. Note that the initial conditions here are in reality final conditions, causing the adjoint differential equation to be solved backward in time.

B.2. Derivation of the Gradient

With the definitions of the state and adjoint equations, the gradient can be derived. This follows a procedure very similar to the procedure defined in the adjoint. Problematic is the dependency of the model constraint on the second spatial derivative. To solve this problem, integration by parts can be utilized to reduce the order of the spatial differentiation by one. This results in the following when one spatial mode includes a small variation of the form:

$$\begin{aligned} \phi_q &= \phi_q^* + \epsilon \eta(x) \\ \phi_{q_x} &= \phi_{q_x}^* + \epsilon \eta_x(x) \end{aligned}$$

where a subscript x denotes differentiation with respect to x. Due to the boundary conditions, the variation $\eta(x)$ on the boundaries is not arbitrary, but can be assumed zero. To find the gradient, the Lagrangian is differentiated with respect to ϵ, where again each term will be treated independently in the following:

Goal functional term:
The term that includes the goal is:

$$\frac{d}{d\epsilon} \int_\Omega E \, d\Omega = \int_\Omega \frac{\partial E}{\partial \phi_q} \eta \, d\Omega \tag{B.21}$$

where it is assumed that the goal does not depend on spatial derivatives.

Model contraint:
The term that includes the model constraint is:

$$\frac{d}{d\epsilon} \int_\Omega \lambda_k G_2^k \, d\Omega = \int_\Omega \left[\eta \frac{\partial(\lambda_k G_2^k)}{\partial \phi_q} + \eta_x \frac{\partial(\lambda_k G_2^k)}{\partial \phi_{q_x}} \right] d\Omega \tag{B.22}$$

Using integration by parts, the second term in the above integral can be written as:

$$\int_\Omega \left[\eta_x \frac{\partial(\lambda_k G_2^k)}{\partial \phi_{q_x}} \right] d\Omega = - \int_\Omega \left[\frac{d}{dx} \left(\lambda_k \frac{\partial G_2^k}{\partial \phi_{q_x}} \right) \eta \right] d\Omega \tag{B.23}$$

where the variation η is assumed zero on the boundaries. Substituting this back in leads to:

$$\frac{d}{d\epsilon} \int_\Omega \lambda_k G_2^k \, d\Omega = \int_\Omega \left[\lambda_k \frac{\partial G_2^k}{\partial \phi_q} - \frac{d}{dx} \left(\lambda_k \frac{\partial G_2^k}{\partial \phi_{q_x}} \right) \right] \eta \, d\Omega \tag{B.24}$$

Initial Conditions
The initial condition constraint leads to:

$$\frac{d}{d\epsilon} \left(\int_\Omega \mu_k G_3^k \, d\Omega \right) = \int_\Omega \mu_k \frac{\partial G_3^k}{\partial \phi_q} \eta \, d\Omega \tag{B.25}$$

Orthonormality Constraint
The orthonormality constraint leads to:

$$\sum_{i,j=1}^{m} \frac{d}{d\epsilon}\left(\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega\right)^2 = \sum_{i,j=1}^{m} 2\left(\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega\right)\frac{d}{d\epsilon}\left(\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega\right) \tag{B.26}$$

Consider the derivative term:

$$\frac{d}{d\epsilon}\left(\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega\right) = \begin{cases} 0, & \text{if } i \neq q \;\&\; j \neq q \\ -\int \phi_j \eta d\Omega, & \text{if } i = q \;\&\; j \neq q \\ -\int \phi_i \eta d\Omega, & \text{if } i \neq q \;\&\; j = q \\ -2\int \phi_q \eta d\Omega, & \text{if } i = q \;\&\; j = q \end{cases} \tag{B.27}$$

All terms in equation B.26 not related to ϕ_q drop out. With this in mind, the index j can be replaced by q and the double sum over i and j is replaced by a single sum over i. It can also be shown that the following then results for the derivative term:

$$\frac{d}{d\epsilon}\left(\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega\right) = -2\int_\Omega \phi_i \eta d\Omega \tag{B.28}$$

Combining this result with equation B.26 leads to the following conclusion:

$$\sum_{i,j=1}^{m} \frac{d}{d\epsilon}\left(\delta_{ij} - \int_\Omega \phi_i\phi_j d\Omega\right)^2 = \sum_{i=1}^{m} 4\left(\int_\Omega \phi_i\phi_q d\Omega - \delta_{iq}\right)\left(\int_\Omega \phi_i \eta d\Omega\right) = 4\sum_{i=1}^{m}\int_\Omega \left(\int_\Omega \phi_i\phi_q d\Omega - \delta_{iq}\right)\phi_i \eta d\Omega \tag{B.29}$$

Combining Everything
Combing equation B.1 with equation B.21, B.24, B.25 and B.29 forms the following equation:

$$\frac{d\mathscr{L}}{d\epsilon} = \int_\Omega \left\{ \int_0^{t_f} \left[\frac{1}{2}\frac{\partial E}{\partial \phi_q} + \sum_{i=1}^{m} \lambda_i \frac{\partial G_2^i}{\partial \phi_q} - \frac{d}{dx}\left(\sum_i^m \lambda_i \frac{\partial G_2^i}{\partial \phi_{q_x}}\right)\right] dt + \sum_{i=1}^{m} \mu_i \frac{\partial G_3^i}{\partial \phi_q} + 2\beta \sum_{i=1}^{m}\left(\int_\Omega \phi_i\phi_q d\Omega - \delta_{iq}\right)\phi_i \right\} \eta d\Omega = 0 \tag{B.30}$$

Arguing again that the variation η is arbitrary, the term inside the curly braces needs to go to zero. The gradient is thus obtained by finding this expression, i.e. equation B.31.

$$\frac{\partial\mathscr{L}}{\partial \phi_q} = \int_0^{t_f} \left[\frac{1}{2}\frac{\partial E}{\partial \phi_q} + \sum_{i=1}^{m} \lambda_i \frac{\partial G_2^i}{\partial \phi_q} - \frac{d}{dx}\left(\sum_i^m \lambda_i \frac{\partial G_2^i}{\partial \phi_{q_x}}\right)\right] dt + \sum_{i=1}^{m} \mu_i \frac{\partial G_3^i}{\partial \phi_q} + 2\beta \sum_{i=1}^{m}\left(\int_\Omega \phi_i\phi_q d\Omega - \delta_{iq}\right)\phi_i \tag{B.31}$$

C

Algorithms

C.1. Trust-Region Method

Algorithm 1 shows the algorithm used in the trust-region method. The formulation here mostly follows the formulation found in Cheng (2017) with a few minor modifications.

Algorithm 1 Trust Region Algorithm. Mostly follows the formulation from Cheng (2017).

```
Given $\Delta_{max} > 0$, $\Delta_{min} \in (0, \Delta_{max})$, $\Delta_0 \in (\Delta_{min}, \Delta_{max})$, $\eta \in \left[0, \frac{1}{4}\right)$
for k = 0,1,2,... do
    Obtain $p_k$ by approximately solving equation 2.17
    Evaluate $\rho_k$ using equation 2.19

    if $\rho_k < \frac{1}{4}$ then
        $\Delta_{k+1} = \frac{1}{4}\Delta_k$                          ▷ reduce trust region radius
    else
        if $\rho_k > \frac{3}{4}$ and $||p_k||_2 \geq \Delta_k$ then
            $\Delta_{k+1} = \min(2\Delta_k, \Delta_{max})$            ▷ enlarge trust region radius
        else
            $\Delta_{k+1} = \Delta_k$                                 ▷ keep trust region radius
        end if
    end if

    if $\rho_k > \eta$ then
        $x_{k+1} = x_k + p_k$                  ▷ enough reduction made, advance step
    else
        $x_{k+1} = x_k$                                               ▷ do not advance
    end if
end for
```

C.2. Conjugate-Gradient Method

To obtain p_k approximately by solving equation 2.17, the conjugate-gradient algorithm by Steihaug (1983) is used. The algorithm can be found in algorithm 2. Note that the formulation of this algorithm is a reprint from Cheng (2017). In the algorithm d_j denotes the search direction of the iteration j.

Algorithm 2 Steihaug CG Algorithm. Follows the formulation from Cheng (2017).

```
Given tolerance $\epsilon_k > 0$
Set $z_0 = 0$, $r_0 = -\nabla f_k - \nabla^2 f_k p_k = -\nabla f_k - \nabla^2 f_k z_0 = -\nabla f_k, d_0 = r_0$

if $||r_0||_2 < \epsilon_k$ then
    return $p_k = z_0 = 0$
end if

for j = 0,1,2,... do
    if $d_j^T \nabla^2 f_k d_j \leq 0$ then
        Compute $\tau > 0$ such that $||z_j + \tau d_j||_2 = \Delta_k$
        return $p_k = z_j + \tau d_j$
    end if
    Compute $\alpha_j = r_j^T r_j / d_j^T \nabla^2 f_k d_j$
    Compute $z_{j+1} = z_j + \alpha_j d_j$
    if $||z_{j+1}||_2 \geq \Delta_k$ then
        Compute $\tau > 0$ such that $||z_j + \tau d_j||_2 = \Delta_k$
        return $p_k = z_j + \tau d_j$
    end if
    Compute $r_{j+1} = r_j - \alpha_j \nabla^2 f_k d_j$
    if $||r_{j+1}||_2 < \epsilon_k$ then
        return $p_k = z_{j+1}$
    end if

    Compute $\beta_{j+1} = r_{j-1}^T r_{j+1} / r_j^T r_j$
    Compute $d_{j+1} = r_{j+1} + \beta_{j+1} d_j$
end for
```

D

Derivation of the Model Constraint

Starting from the governing equations. That is:

$$\dot{u} + \nabla \cdot \mathcal{F}^x_{inv}(\boldsymbol{u}) - \frac{1}{Re}\nabla \cdot \mathcal{F}^x_{visc}(\boldsymbol{u}) + \frac{\partial p}{\partial x} = f^x \tag{D.1}$$

$$\dot{v} + \nabla \cdot \mathcal{F}^y_{inv}(\boldsymbol{u}) - \frac{1}{Re}\nabla \cdot \mathcal{F}^y_{visc}(\boldsymbol{u}) + \frac{\partial p}{\partial y} = f^y \tag{D.2}$$

$$\dot{w} + \nabla \cdot \mathcal{F}^z_{inv}(\boldsymbol{u}) - \frac{1}{Re}\nabla \cdot \mathcal{F}^z_{visc}(\boldsymbol{u}) + \frac{\partial p}{\partial z} = f^z \tag{D.3}$$

$$\nabla \cdot \boldsymbol{u} = f^c \tag{D.4}$$

To construct a ROM of the governing equations, the equations are projected onto the basis functions using a Bubnov-Galerkin method. This will be discussed in the first section. Furthermore, the weak form of the INS equations require stabilization terms, which will be achieved through the Variational-Multiscale Method introduced by Hughes et al. (2001). This method, along with the model for the unresolved scales, will be discussed in the second section. The final sections combines the conclusions from both preceding sections into one final set of equations.

D.1. Projection of Governing Equations

This section discusses the procedure for projecting the governing equations onto the basis functions and constructing the equivalent weak form.

D.1.1. Momentum equation

First, the $x\text{-}momentum$ equation (equation D.1) is projected onto the primary basis function ϕ^u_k using a Galerkin projection. That is:

$$(\phi^u_k, \dot{u})_\Omega + (\phi^u_k, \nabla \cdot \mathcal{F}^x_{inv})_\Omega - \frac{1}{Re}(\phi^u_k, \nabla \cdot \mathcal{F}^x_{visc})_\Omega + (\phi^u_k, \frac{\partial p}{\partial x})_\Omega = (\phi^u_k, f^x)_\Omega \tag{D.5}$$

Integration by parts (equation D.6) can be used to rewrite some of the terms in equation D.5 into equations D.7-D.9.

$$\int_\Omega a \nabla \cdot \boldsymbol{b} d\Omega = \int_\Gamma a\boldsymbol{b} \cdot \boldsymbol{n} d\Gamma - \int_\Omega \nabla a \cdot \boldsymbol{b} d\Omega \tag{D.6}$$

$$(\phi^u_k, \nabla\cdot\mathcal{F}^x_{inv})_\Omega = \int_\Gamma \phi^u_k\mathcal{F}^x_{inv}\cdot\boldsymbol{n}d\Gamma - \int_\Omega \nabla\phi^u_k\cdot\mathcal{F}^x_{inv}d\Omega \tag{D.7}$$

$$(\phi^u_k, \nabla\cdot\mathcal{F}^x_{visc})_\Omega = \int_\Gamma \phi^u_k\mathcal{F}^x_{visc}\cdot\boldsymbol{n}d\Gamma - \int_\Omega \nabla\phi^u_k\cdot\mathcal{F}^x_{visc}d\Omega \tag{D.8}$$

$$\begin{aligned}(\phi^u_k, \frac{\partial p}{\partial x})_\Omega &= \int_\Omega \phi^u_k\nabla\cdot p\boldsymbol{e}^x d\Omega = \int_\Gamma \phi^u_k p\boldsymbol{e}^x\cdot\boldsymbol{n}d\Gamma - \int_\Omega \nabla\phi^u_k\cdot p\boldsymbol{e}^x d\Omega \\ &= \int_\Gamma \phi^u_k p n^x d\Gamma - \int_\Omega \frac{\partial\phi^u_k}{\partial x} p d\Omega \end{aligned}\tag{D.9}$$

where $\boldsymbol{e}^x = [1,0,0]^T$ is the unit vector in x-direction and $\boldsymbol{n} = [n^x, n^y, n^z]^T$ is the outward-normal unit vector on the boundary. Substituting equations D.7-D.9 into equation D.5 leads to the following equation:

$$(\phi^u_k, \dot{u})_\Omega + I^x_{bound}(\boldsymbol{u}) + I^x_{domain}(\boldsymbol{u}) = (\phi^u_k, f^x)_\Omega \tag{D.10}$$

with:

$$I^x_{bound}(\boldsymbol{u}) = \int_\Gamma \phi^u_k\mathcal{F}^x_{inv}(\boldsymbol{u})\cdot\boldsymbol{n}d\Gamma - \frac{1}{Re}\int_\Gamma \phi^u_k\mathcal{F}^x_{visc}(\boldsymbol{u})\cdot\boldsymbol{n}d\Gamma + \int_\Gamma \phi^u_k p n^x d\Gamma \tag{D.11}$$

$$I^x_{domain}(\boldsymbol{u}) = -\int_\Omega \nabla\phi^u_k\cdot\mathcal{F}^x_{inv}(\boldsymbol{u})d\Omega + \frac{1}{Re}\int_\Omega \nabla\phi^u_k\cdot\mathcal{F}^x_{visc}(\boldsymbol{u})d\Omega - \int_\Omega \frac{\partial\phi^u_k}{\partial x} p d\Omega \tag{D.12}$$

When this procedure is repeated by projecting the y and z momentum equations onto the basis functions ϕ^v_k and ϕ^w_k, the expressions can be written as:

$$(\phi^r_k, \dot{u^r})_\Omega + I^r_{bound} + I^r_{domain} = (\phi^r_k, f^r)_\Omega \tag{D.13}$$

with:

$$I^r_{bound} = \int_\Gamma \phi^r_k \sum_{j=1}^{N_{dim}} \mathcal{F}^r_{inv}(u^j)n^j d\Gamma - \frac{1}{Re}\int_\Gamma \phi^r_k \sum_{j=1}^{N_{dim}} \mathcal{F}^r_{visc}(u^j)n^j d\Gamma + \int_\Gamma \phi^r_k p n^r d\Gamma \tag{D.14}$$

$$I^r_{domain} = -\int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial\phi^r_k}{\partial x^j}\mathcal{F}^r_{inv}(u^j)d\Omega + \frac{1}{Re}\int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial\phi^r_k}{\partial x^j}\mathcal{F}^r_{visc}(u^j)d\Omega - \left(\frac{\partial\phi^r_i}{\partial x^r}, p\right)_\Omega \tag{D.15}$$

and:

$$\mathcal{F}^r_{inv}(u^j) = u^r u^j \tag{D.16}$$

$$\mathcal{F}^r_{visc}(u^j) = \frac{\partial u^r}{\partial x^j} + \frac{\partial u^j}{\partial x^r} \tag{D.17}$$

where the superscripts denote the spatial direction of the particular variable[1].

D.1.2. Continuity Equation

Applying a similar technique to the continuity equation, the following weak form is recovered:

$$\int_\Omega \sum_{j=1}^{N_{dim}} \phi^p_k \frac{\partial u^j}{\partial x^j} d\Omega = (\phi^p_k, f^c)_\Omega \tag{D.18}$$

D.2. Varitiational-Multiscale Method

To model the effect of unresolved scales on the resolved scales, the variational-multiscale method (VMM) will be used. The decomposition thus results in resolved and unresolved scales as:

$$u^r = \hat{u}^r + u^{r\prime} \tag{D.19}$$

$$p = \hat{p} + p' \tag{D.20}$$

---

[1] i.e. u^r with $r = 1$ corresponds to the x-velocity, $r = 2$ corresponds to the y-velocity and $r = 3$ corresponds to the z-velocity

with:

$$\hat{u}^r = \sum_{i=1}^{m} \alpha_i^r \phi_i^r \tag{D.21}$$

$$\hat{p} = \sum_{i=1}^{m} \alpha_i^p \phi_i^p \tag{D.22}$$

Momentum Equations

In this work, both the unresolved viscous and unresolved time derivative will be omitted from here on out, simplifying the viscous term to:

$$\mathcal{F}^r_{visc}(u^j) = \mathcal{F}^r_{visc}(\hat{u}^j) = \mathcal{F}^r_{visc}(\hat{u}^j) = \frac{\partial \hat{u}^r}{\partial x^j} + \frac{\partial \hat{u}^j}{\partial x^r} \tag{D.23}$$

and the time derivative term to:

$$(\phi_k^r, \dot{u}^r)_\Omega = (\phi_k^r, \dot{\hat{u}}^r)_\Omega \tag{D.24}$$

Similarly, all unresolved variables on the boundaries will be omitted, allowing similar simplifications to equation D.14. The momentum terms can then be completed by simply substituting in equations D.19 and D.20 into equation D.15, resulting in equation D.25.

$$(\phi_k^r, \dot{\hat{u}}^r)_\Omega + I^r_{bound} + I^r_{domain} = (\phi_k^r, f^r)_\Omega \tag{D.25}$$

with:

$$I^r_{bound} = \int_\Gamma \phi_k^r \sum_{j=1}^{N_{dim}} \mathcal{F}^r_{inv}(\hat{u}^j) n^j \, d\Gamma - \frac{1}{Re} \int_\Gamma \phi_k^r \sum_{j=1}^{N_{dim}} \mathcal{F}^r_{visc}(\hat{u}^j) n^j \, d\Gamma + \int_\Gamma \phi_k^r \hat{p} n^r \, d\Gamma \tag{D.26}$$

$$I^r_{domain} = -\int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} \mathcal{F}^r_{inv}(\hat{u}^j + u^{j\prime}) d\Omega + \frac{1}{Re} \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} \mathcal{F}^r_{visc}(\hat{u}^j) d\Omega - \int_\Omega \frac{\partial \phi_i^r}{\partial x^r} \left(\hat{p} + p'\right) d\Omega \tag{D.27}$$

Continuity Equation

Before completing the substitution of the unresolved scales in the continuity equation, the continuity equation requires some more attention. First of all, it is noted that the term is linear, simplifying the expansions to be:

$$\int_\Omega \phi_k^p \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^j} d\Omega + \int_\Omega \phi_k^p \sum_{j=1}^{N_{dim}} \frac{\partial u'^j}{\partial x^j} d\Omega = \left(\phi_k^p, f^c\right)_\Omega \tag{D.28}$$

For the sake of implementation, it is more convenient to remove the spatial derivative in front of the unresolved scales. This is again achieved by using integration by parts (equation D.6), resulting in:

$$\int_\Omega \phi_k^p \sum_{j=1}^{N_{dim}} \frac{\partial u'^j}{\partial x^j} d\Omega = -\int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^p}{\partial x^j} u'^j \, d\Omega \tag{D.29}$$

where the boundary integral is omitted. The continuity equation then finally becomes:

$$\int_\Omega \phi_k^p \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^j} d\Omega - \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^p}{\partial x^j} u'^j \, d\Omega = \left(\phi_k^p, f^c\right)_\Omega \tag{D.30}$$

SGS Model

Although it is possible to solve the unresolved scales u'^r & p' directly, the process is computationally expensive. Therfore, the terms will be modelled through a SGS model of the form seen in equations D.31 and D.32.

$$u'^r = -\tau_m R^r_m \tag{D.31}$$

$$p' = -\tau_c R_c \tag{D.32}$$

where τ_m and τ_c are two stabilization parameters and:

$$R_m^r = \dot{\hat{u}}^r + \sum_{j=1}^{N_{dim}} \frac{\partial(\hat{u}^r \hat{u}^j)}{\partial x^j} - \frac{1}{Re} \sum_{j=1}^{N_{dim}} \frac{\partial}{\partial x^j} \left(\frac{\partial \hat{u}^r}{\partial x^j} + \frac{\partial \hat{u}^j}{\partial x^r} \right) + \frac{\partial \hat{p}}{\partial x^r} - f^r \tag{D.33}$$

$$R_c = \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^j} - f^c \tag{D.34}$$

are the residuals of the momentum equations and continuity equation respectively.

Boundary Conditions

This work will utilize weak Dirichlet-type boundary conditions for the velocity. This takes the form of a boundary integral added to the momentum equations of the form seen in equation D.35.

$$I_{bound,bc}^r = \int_\Gamma \sigma^r (\hat{u}^r - u_{bc}^r) d\Gamma \tag{D.35}$$

D.3. Final Model Constraint

Recombining each term into a final set of equations, the model constraint can be summarized as follows:

$$(\phi_k^r, \dot{\hat{u}}^r)_\Omega + B_1^r - \frac{1}{Re} B_2^r + B_3^r + B_4^r - D_1^r + \frac{1}{Re} D_2^r - D_3^r = (\phi_i^r, f^r)_\Omega \tag{D.36}$$

$$D_4 = \left(\phi_k^p, f^c\right)_\Omega \tag{D.37}$$

where $r = [1,2,3]$ denotes the direction of the momentum equation.[2] The boundary integrals B are defined from equations D.26 and D.35 and can be found in equations D.38-D.41:

$$B_1^r = \int_\Gamma \phi_k^r \sum_{j=1}^{N_{dim}} \mathcal{F}_{inv}^r (\hat{u}^j) n^j d\Gamma = \int_\Gamma \phi_k^r \hat{u}^r \sum_{j=1}^{N_{dim}} \hat{u}^j n^j d\Gamma \tag{D.38}$$

$$B_2^r = \int_\Gamma \phi_k^r \sum_{j=1}^{N_{dim}} \mathcal{F}_{visc}^r (\hat{u}^j) n^j d\Gamma = \int_\Gamma \phi_k^r \sum_{j=1}^{N_{dim}} \left(\frac{\partial \hat{u}^r}{\partial x^j} + \frac{\partial \hat{u}^j}{\partial x^r} \right) n^j d\Gamma \tag{D.39}$$

$$B_3^r = \int_\Gamma \phi_k^r \hat{p} n^r d\Gamma \tag{D.40}$$

$$B_4^r = \int_\Gamma \sigma^r (\hat{u}^r - u_{bc}^r) d\Gamma \tag{D.41}$$

and the domain integrals are defined from equations D.27 and D.30 and can be found in equations D.42-D.45.

$$D_1^r = \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} \mathcal{F}_{inv}^r (u^j) d\Omega = \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} (\hat{u}^r - \tau_m R_m^r)(\hat{u}_j - \tau_m R_m^j) d\Omega \tag{D.42}$$

$$D_2^r = \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} \mathcal{F}_{visc}^r (u^j) d\Omega = \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^r}{\partial x^j} \left(\frac{\partial \hat{u}^r}{\partial x^j} + \frac{\partial \hat{u}^j}{\partial x^r} \right) d\Omega \tag{D.43}$$

$$D_3^r = \int_\Omega \frac{\partial \phi_k^r}{\partial x^r} p d\Omega = \int_\Omega \frac{\partial \phi_k^r}{\partial x^r} \hat{p} d\Omega - \int_\Omega \frac{\partial \phi_k^r}{\partial x^r} \tau_c R_c d\Omega \tag{D.44}$$

$$D_4 = \int_\Omega \phi_k^p \sum_{j=1}^{N_{dim}} \frac{\partial \hat{u}^j}{\partial x^j} d\Omega + \int_\Omega \sum_{j=1}^{N_{dim}} \frac{\partial \phi_k^p}{\partial x^j}, \tau_m R_m^j d\Omega \tag{D.45}$$

---

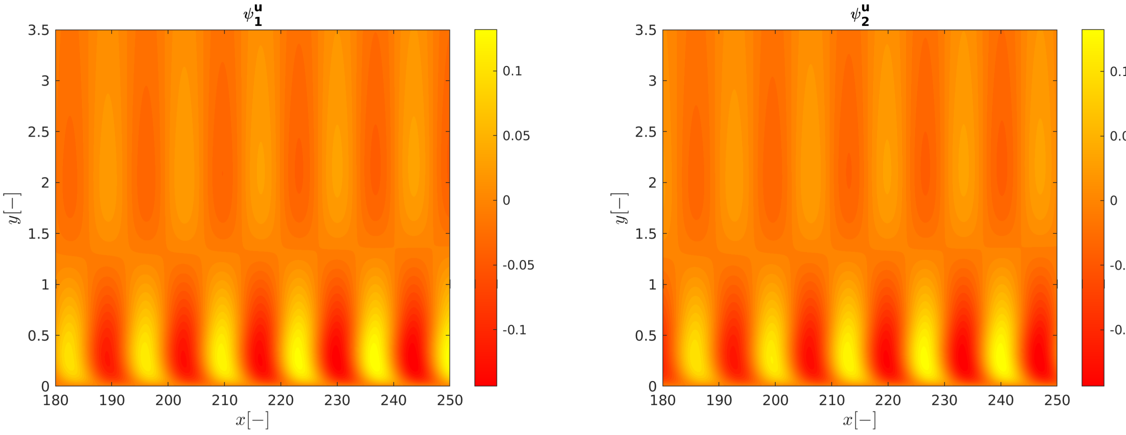
[2] i.e. 1 denotes the x-direction, 2 the y-direction and 3 the z-direction

E

Shape Functions

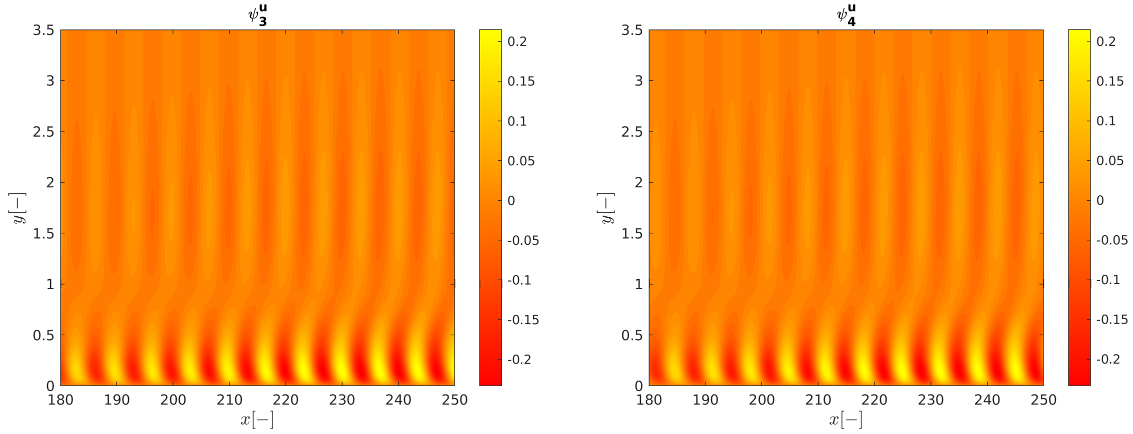
This appendix contains the primary and secondary shape functions of the cases considered throughout this work. The secondary shape functions of the flat plate case can be found on pages 79-81. The secondary shapes of the step case with the low boundary can be found on pages 82- 84. The secondary shapes of the step case with the high boundary can be found on pages 85-87. The GOROM modes from case B and C can be found in pages 88-89 and 90-92. Finally, the GOROM modes from case H can be found on pages 93-95.

E.1. Secondary Shapes Flat Plate



(a) First POD mode

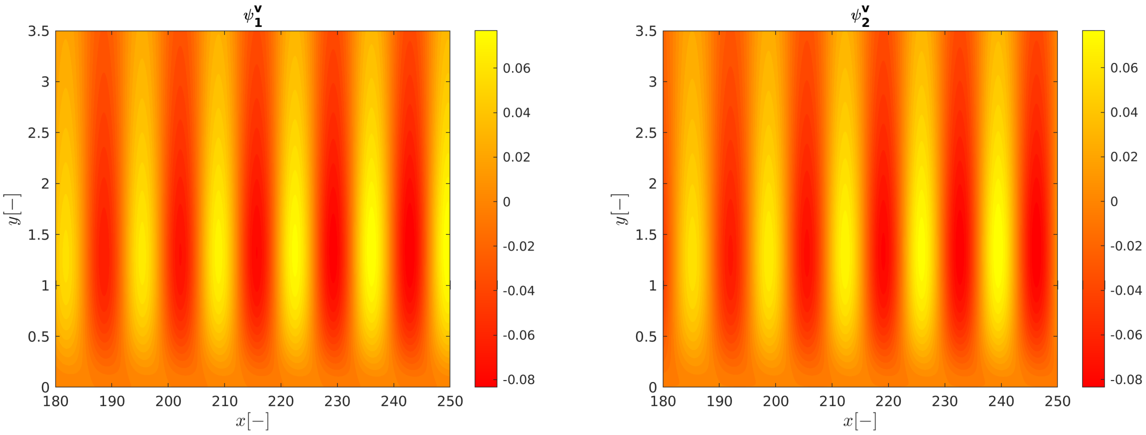
(b) Second POD mode



(c) Third POD mode

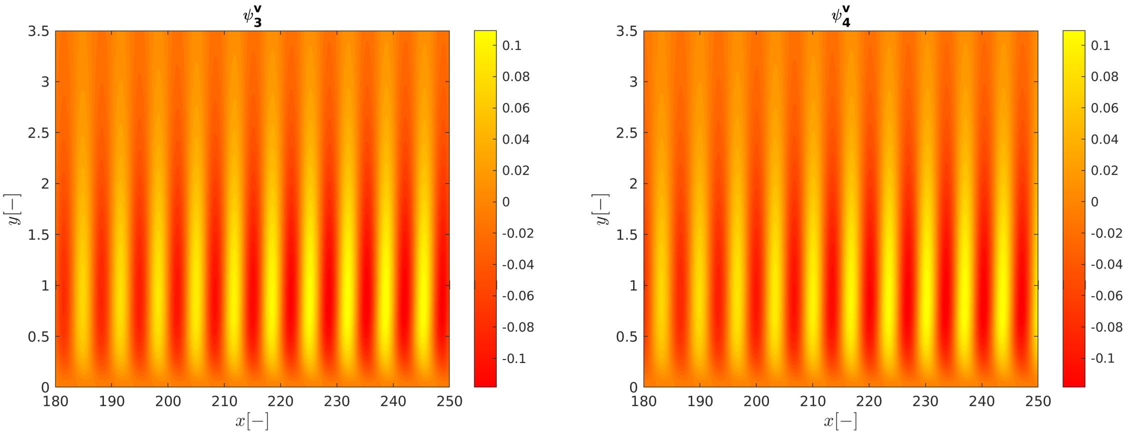
(d) Fourth POD mode

Figure E.1: First 4 POD modes of the x-velocity for the flat plate case.



(a) First POD mode

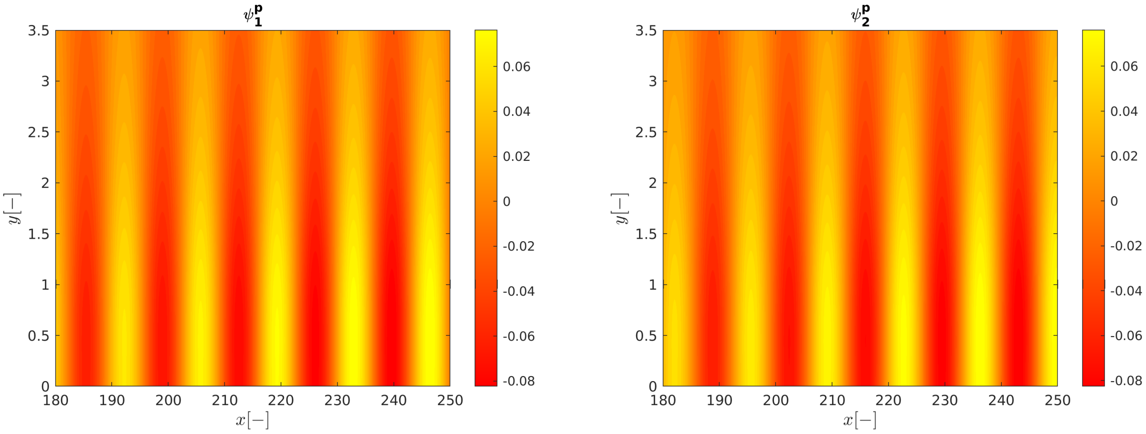
(b) Second POD mode



(c) Third POD mode

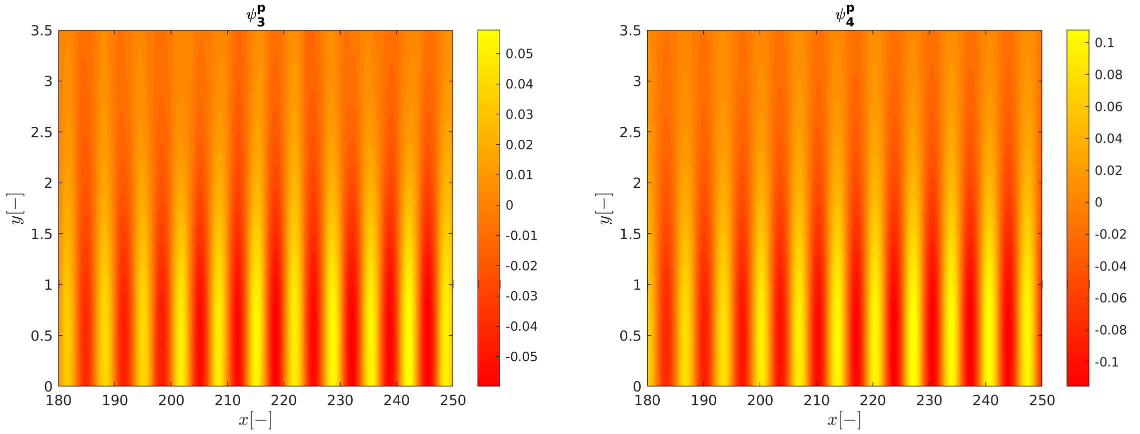
(d) Fourth POD mode

Figure E.2: First 4 POD modes of the y-velocity for the flat plate case.



(a) First POD mode

(b) Second POD mode



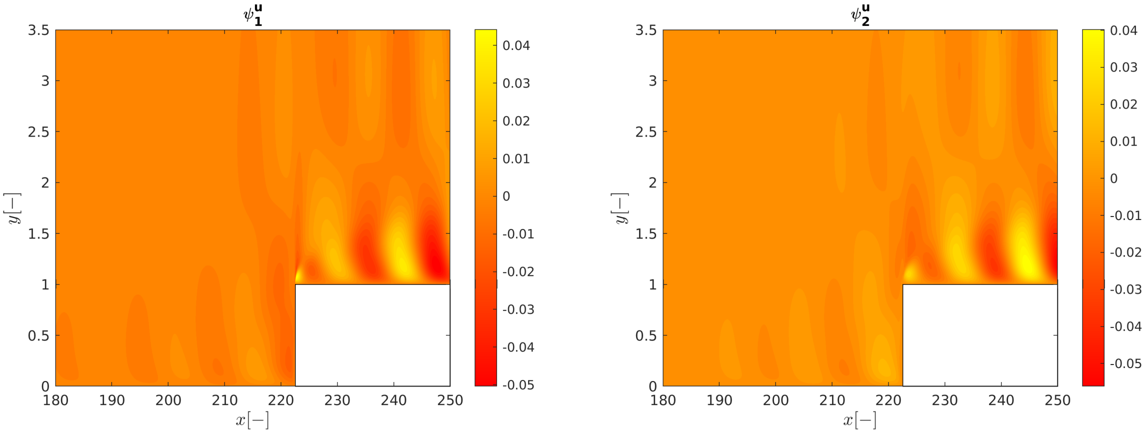
(c) Third POD mode

(d) Fourth POD mode

Figure E.3: First 4 POD modes of the pressure for the flat plate case.

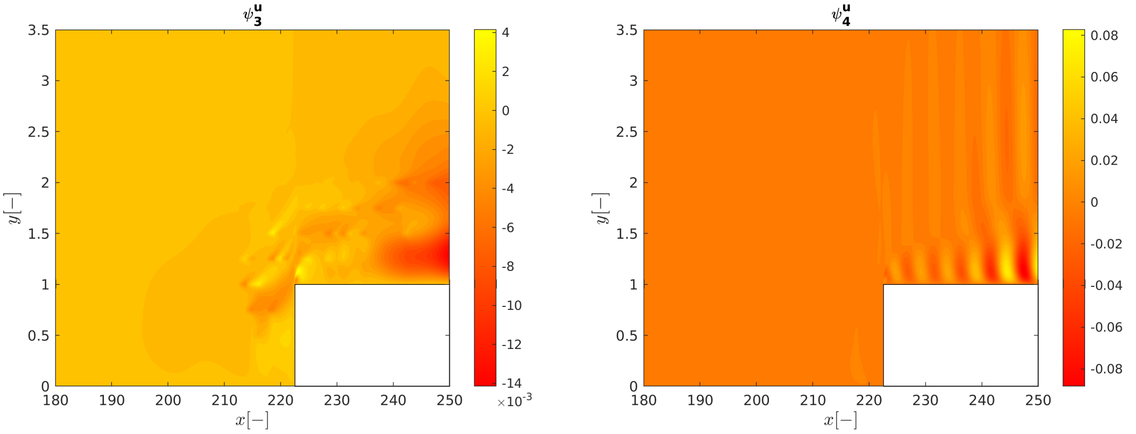
E.2. Secondary Shapes Step Case

E.2.1. Secondary Shapes Low Boundary Case



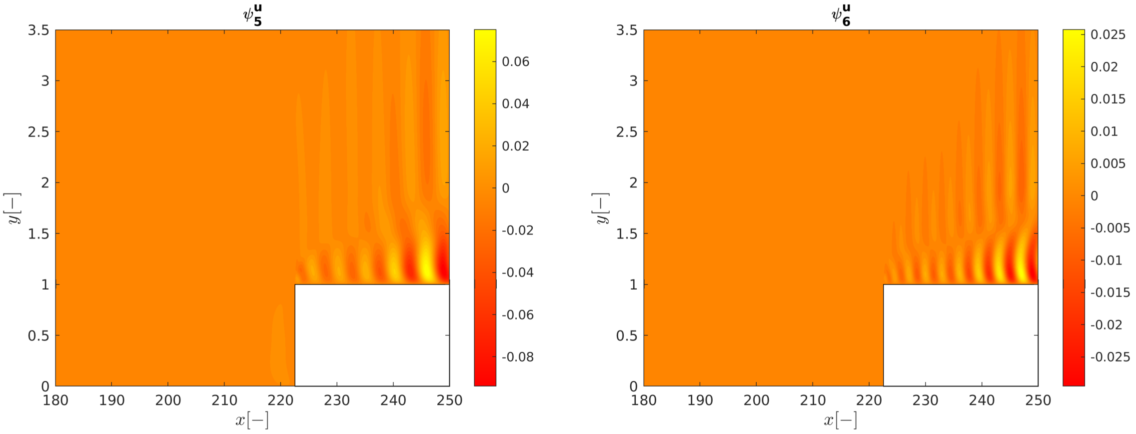
(a) First POD mode

(b) Second POD mode



(c) Third POD mode

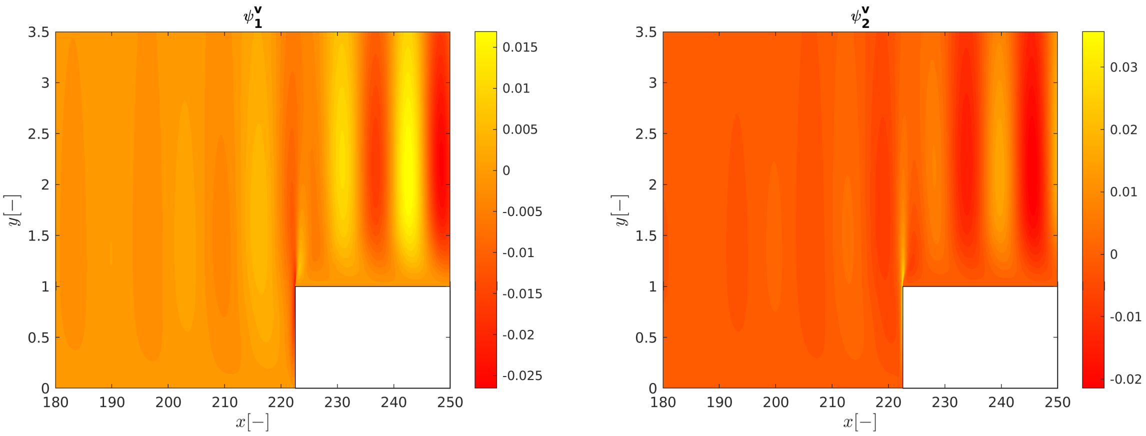
(d) Fourth POD mode



(e) Fifth POD mode

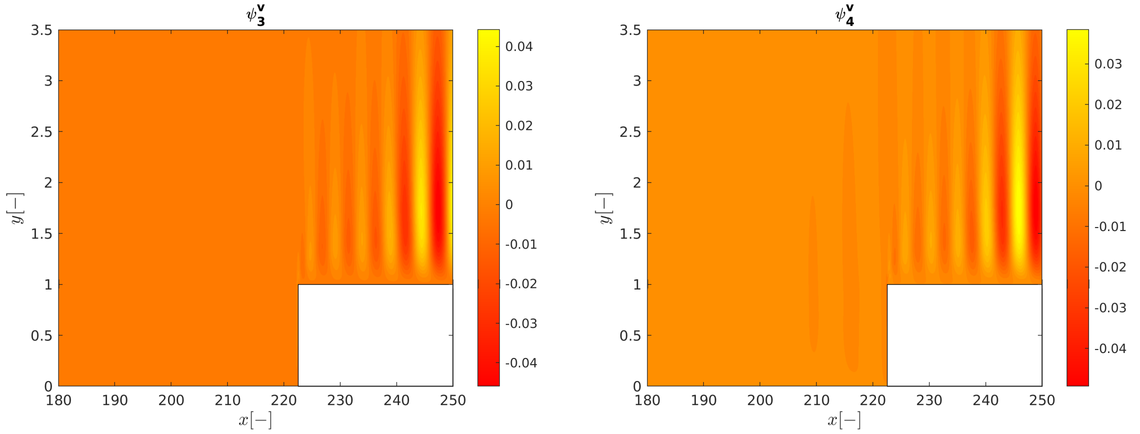
(f) Sixth POD mode

Figure E.4: First 6 POD modes of the x-velocity for the step case with a low top boundary.



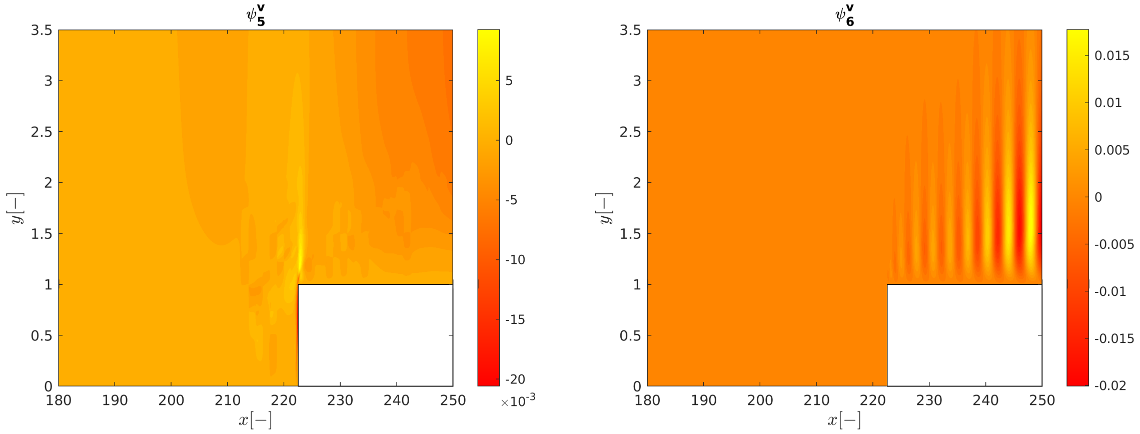
(a) First POD mode

(b) Second POD mode



(c) Third POD mode

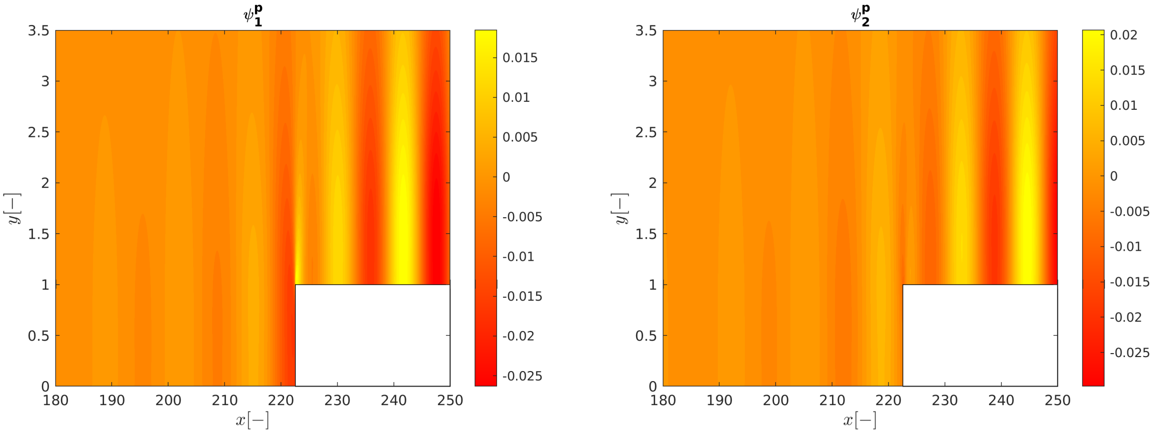
(d) Fourth POD mode



(e) Fifth POD mode

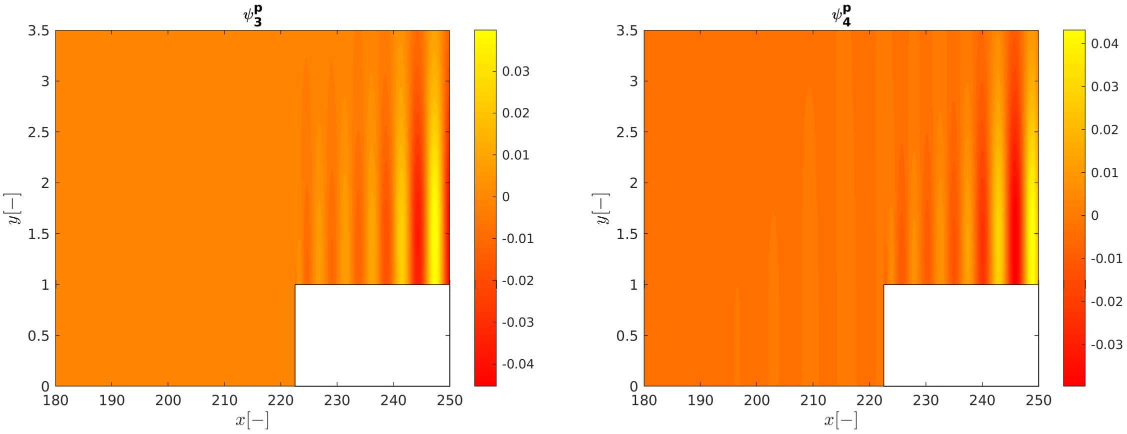
(f) Sixth POD mode

Figure E.5: First 6 POD modes of the y-velocity for the step case with a low top boundary.



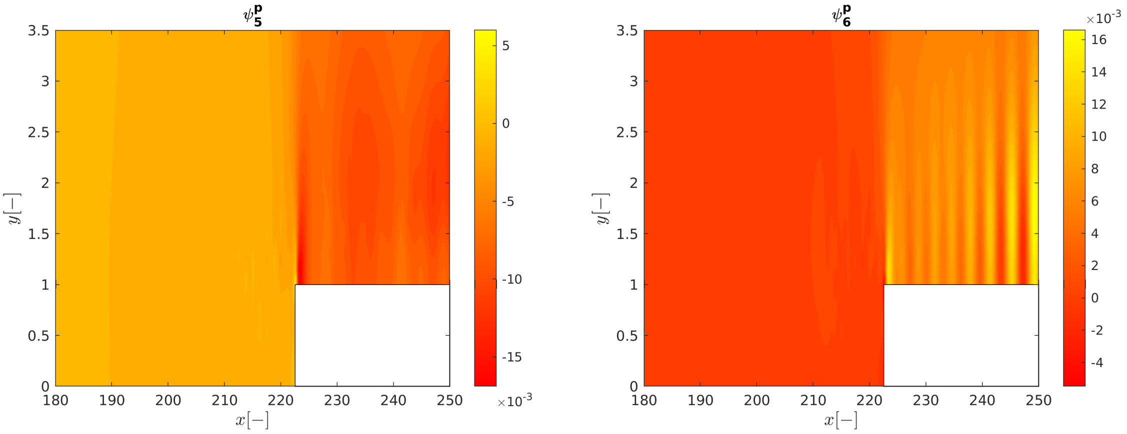
(a) First POD mode

(b) Second POD mode



(c) Third POD mode

(d) Fourth POD mode

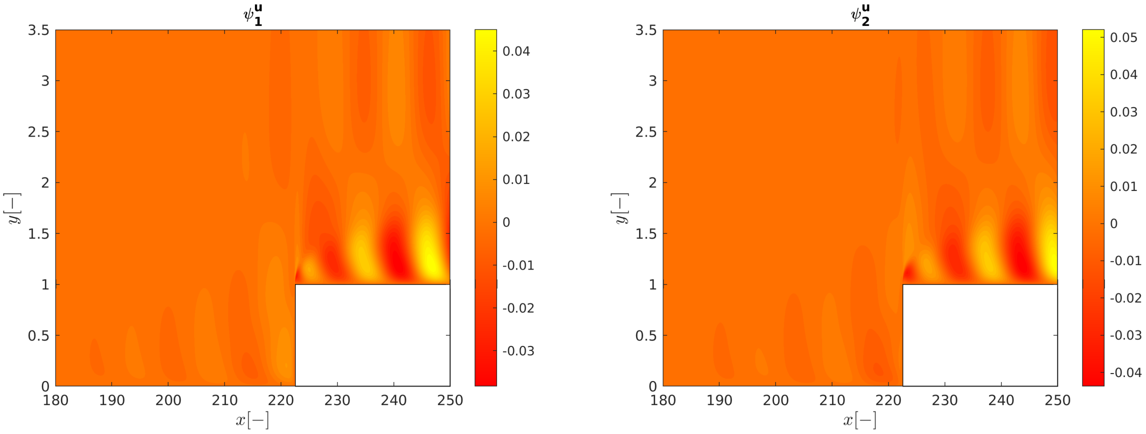


(e) Fifth POD mode

(f) Sixth POD mode

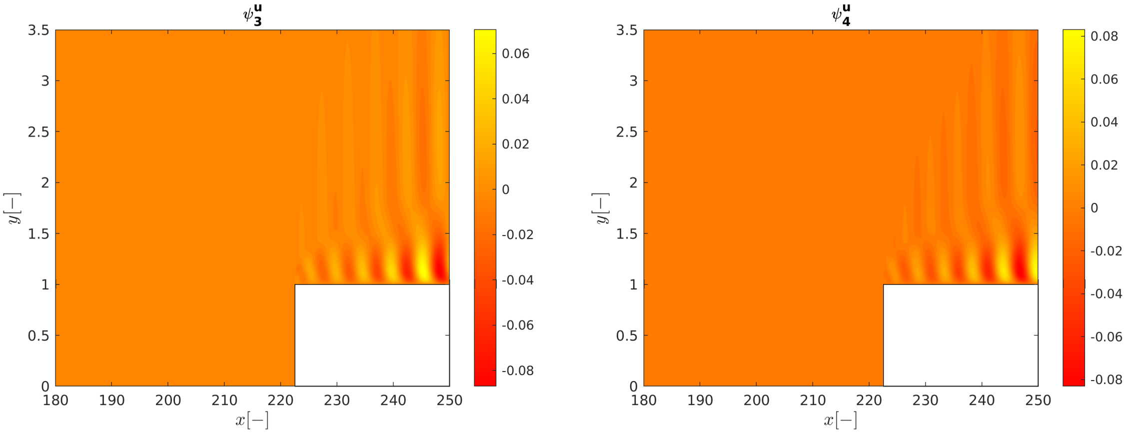
Figure E.6: First 6 POD modes of the pressure for the step case with a low top boundary.

E.2.2. Secondary Shapes High Boundary Case



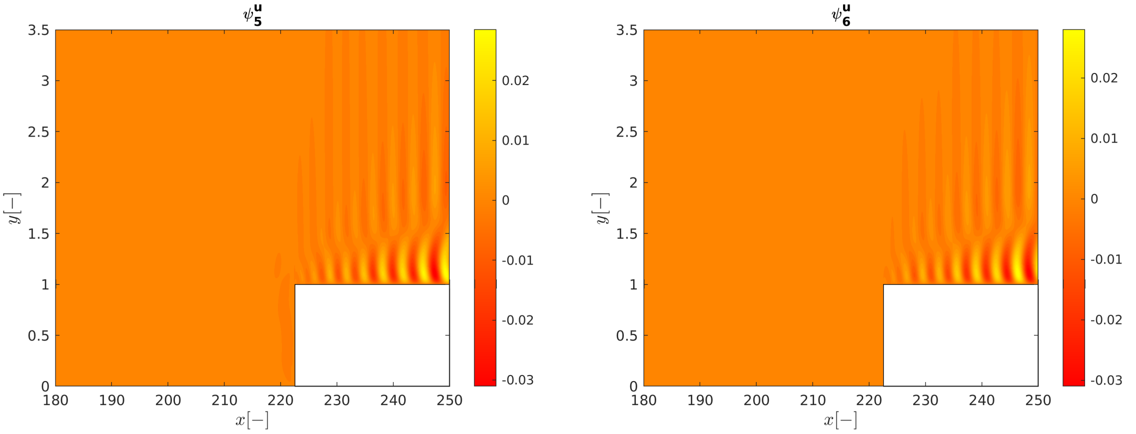
(a) First POD mode

(b) Second POD mode



(c) Third POD mode

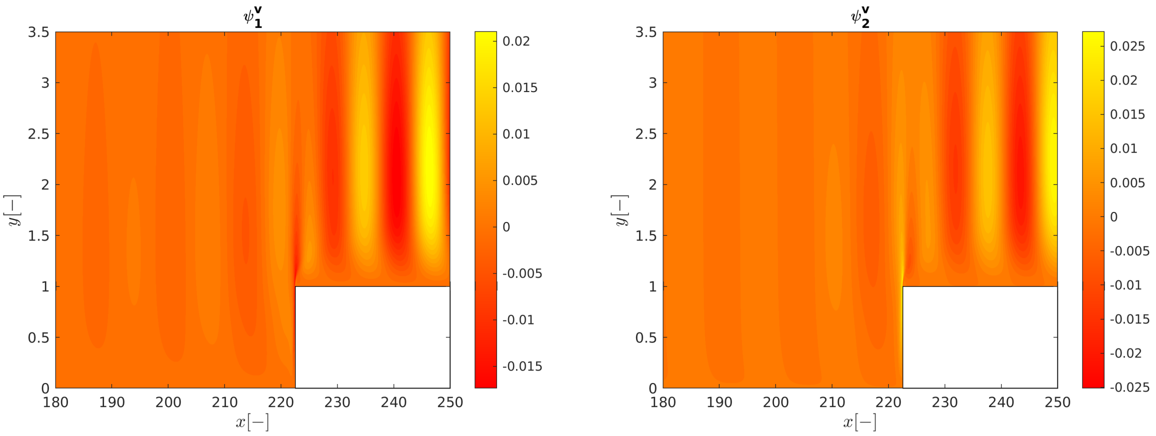
(d) Fourth POD mode



(e) Fifth POD mode

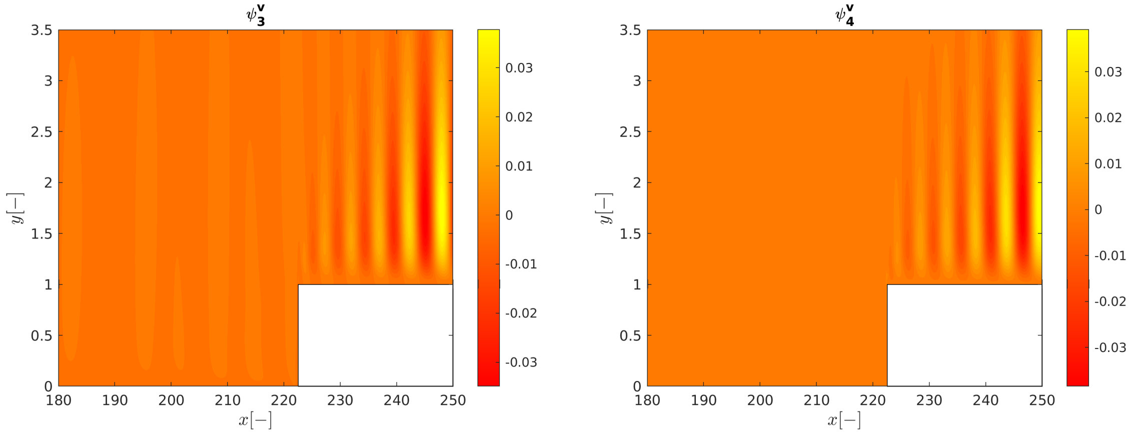
(f) Sixth POD mode

Figure E.7: First 6 POD modes of the x-velocity for the step case with a high top boundary.



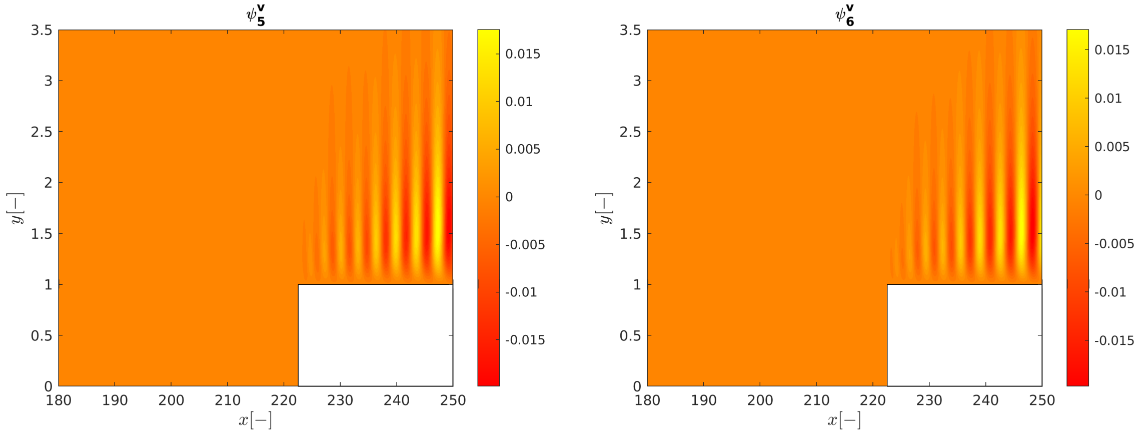
(a) First POD mode

(b) Second POD mode



(c) Third POD mode

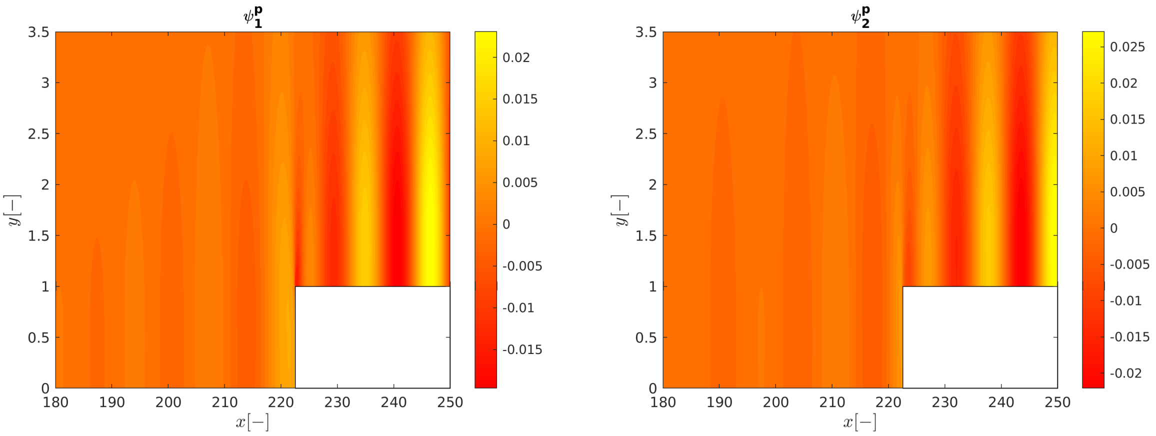
(d) Fourth POD mode



(e) Fifth POD mode

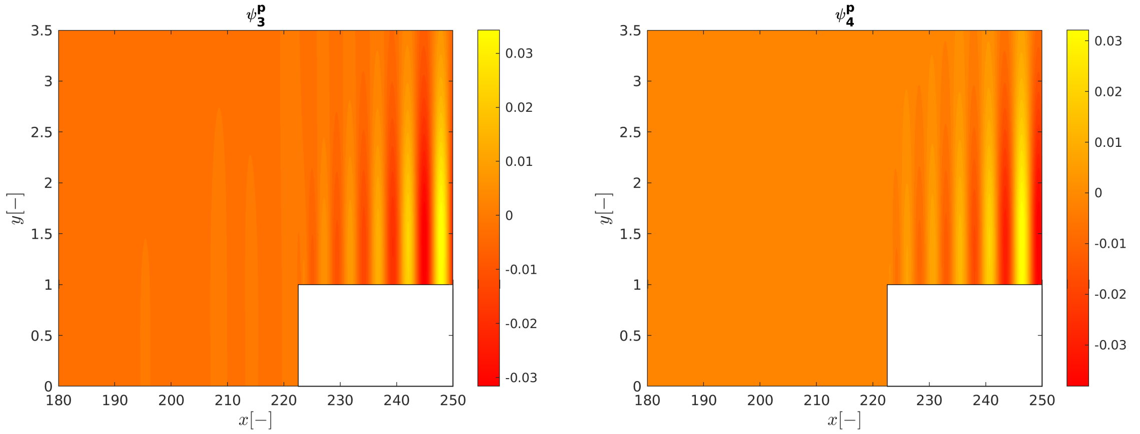
(f) Sixth POD mode

Figure E.8: First 6 POD modes of the y-velocity for the step case with a high top boundary.



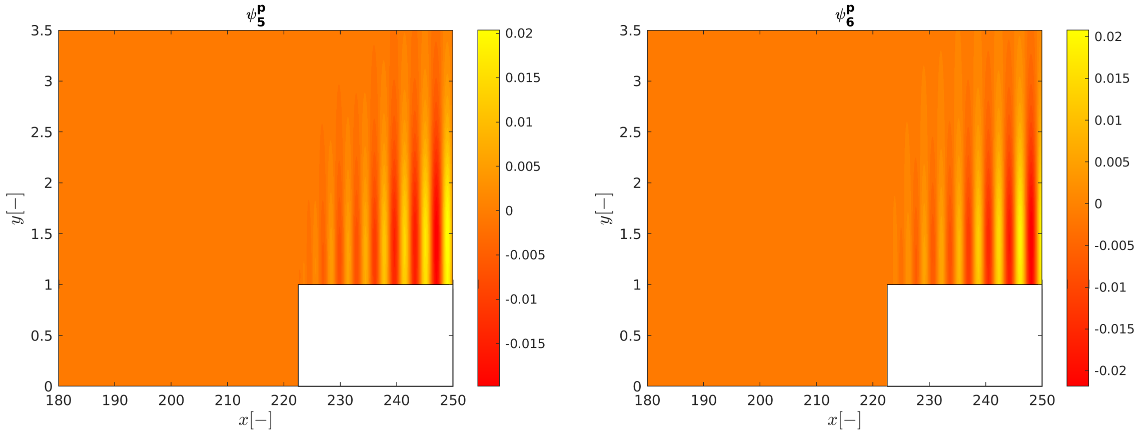
(a) First POD mode

(b) Second POD mode



(c) Third POD mode

(d) Fourth POD mode



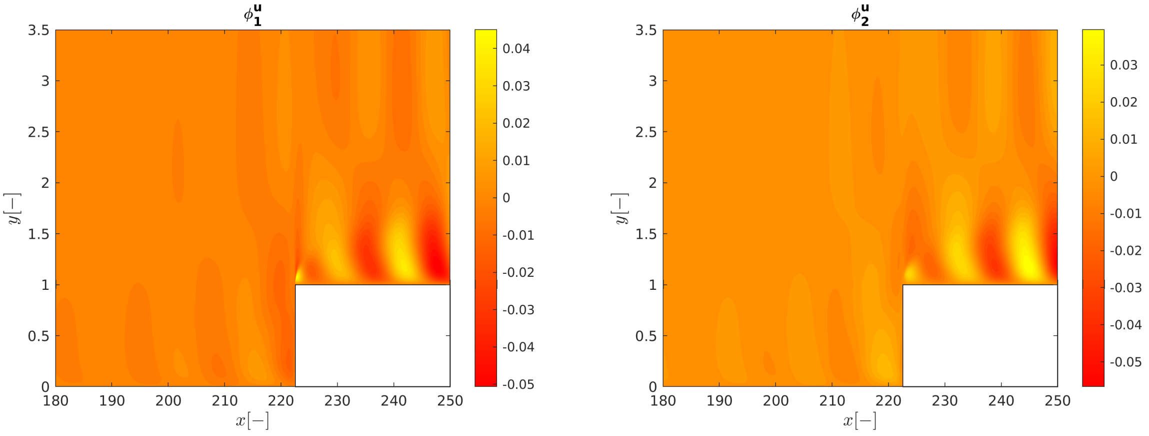
(e) Fifth POD mode

(f) Sixth POD mode

Figure E.9: First 6 POD modes of the pressure for the step case with a high top boundary.

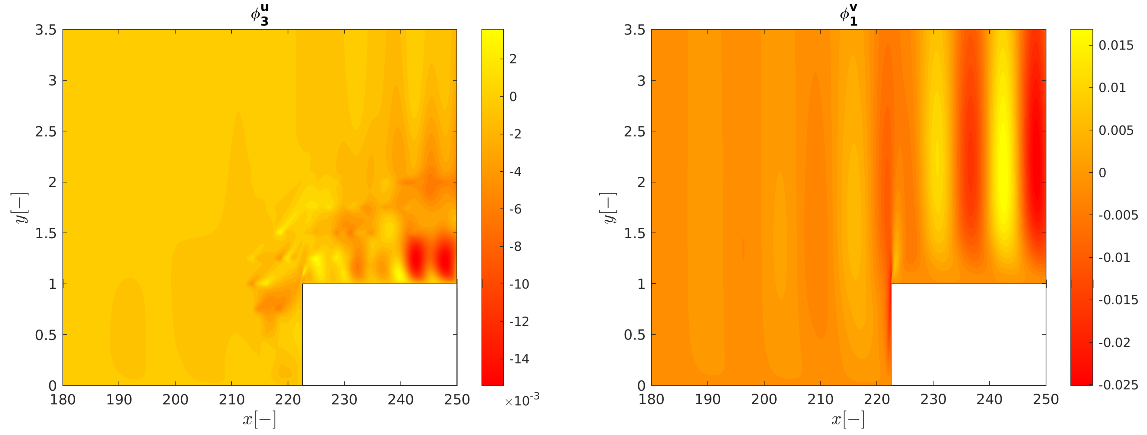
E.3. Primary Shapes Step Case

E.3.1. Case B



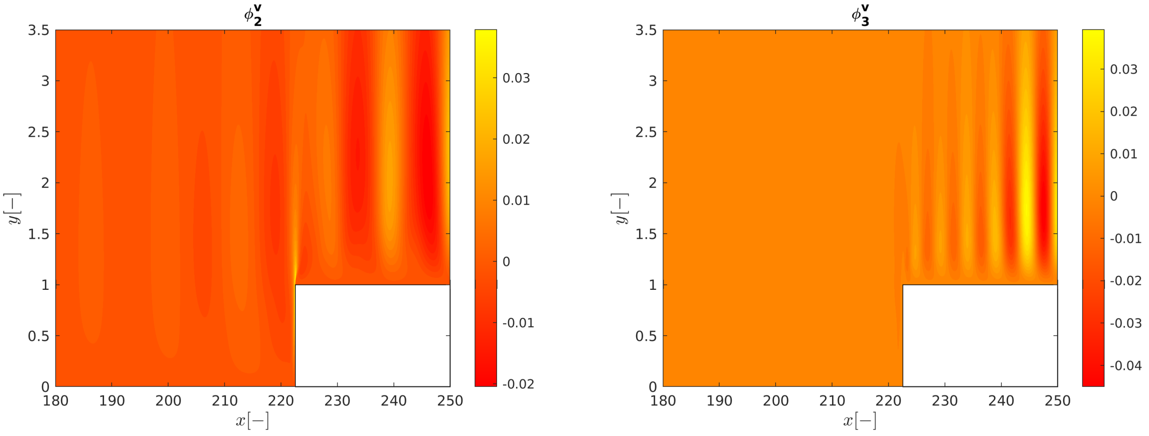
(a) First GOROM mode u

(b) Second GOROM mode u



(c) Third GOROM mode u

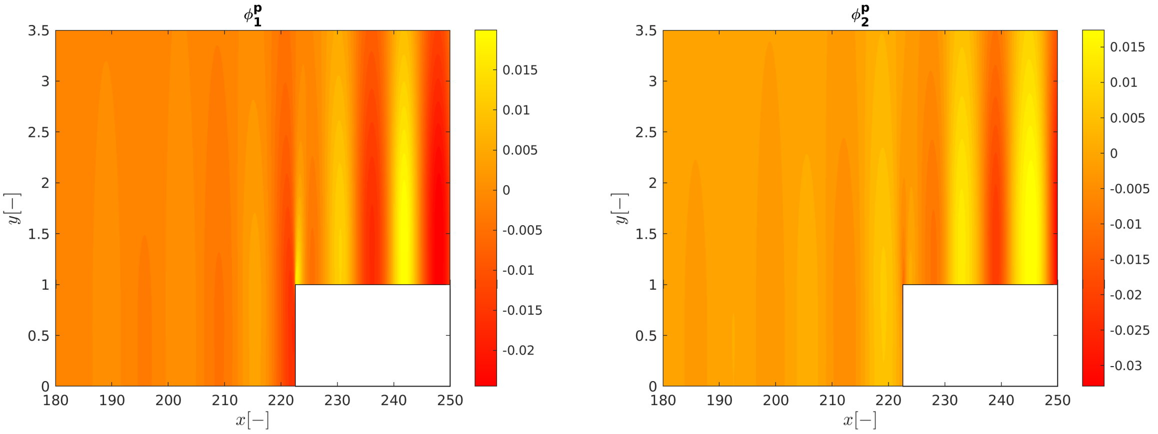
(d) First GOROM mode v



(e) Second GOROM mode v

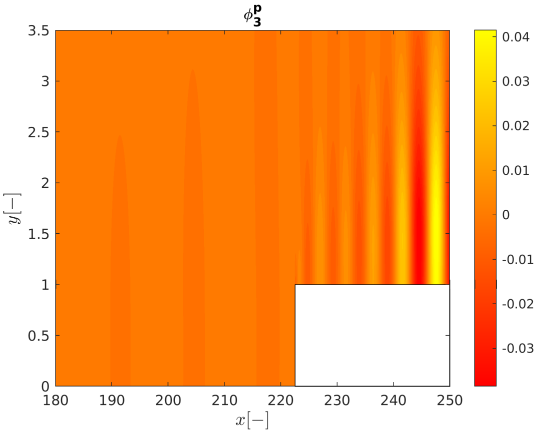
(f) Third GOROM mode v

Figure E.10: GOROM modes of the velocities for case B.



(a) First GOROM mode

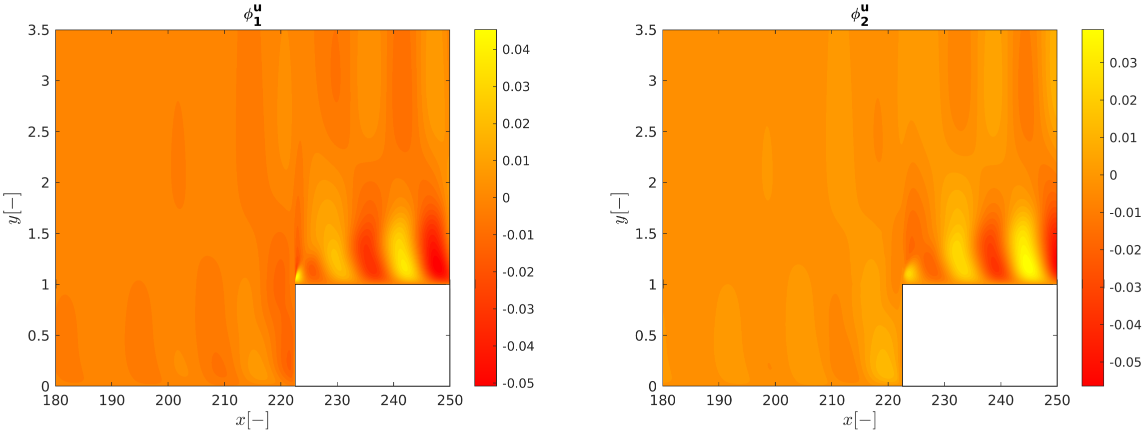
(b) Second GOROM mode



(c) Third GOROM mode

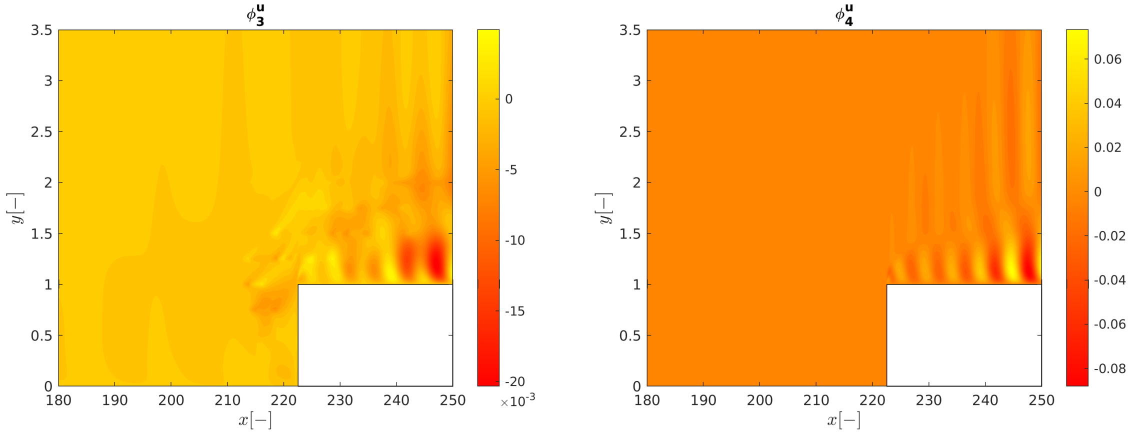
Figure E.11: GOROM modes of the pressure for case B.

E.3.2. Case C



(a) First GOROM mode

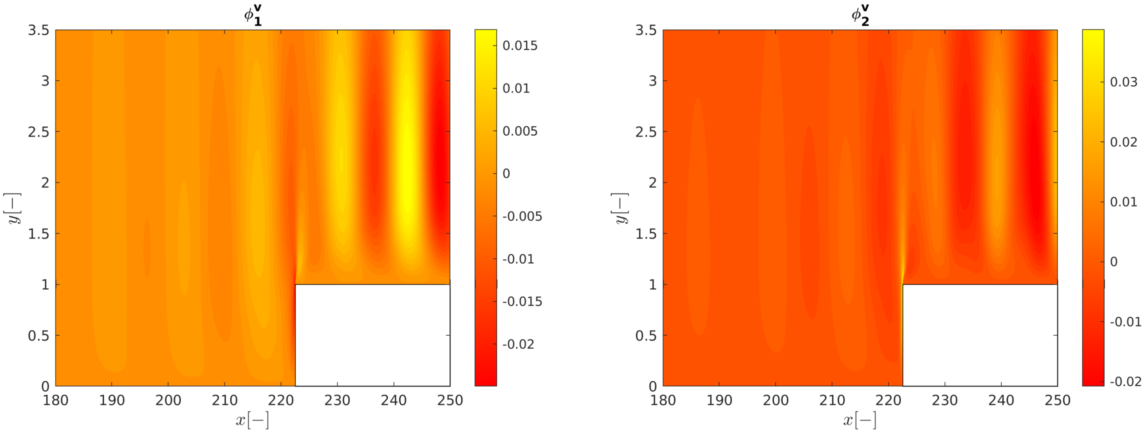
(b) Second GOROM mode



(c) Third GOROM mode

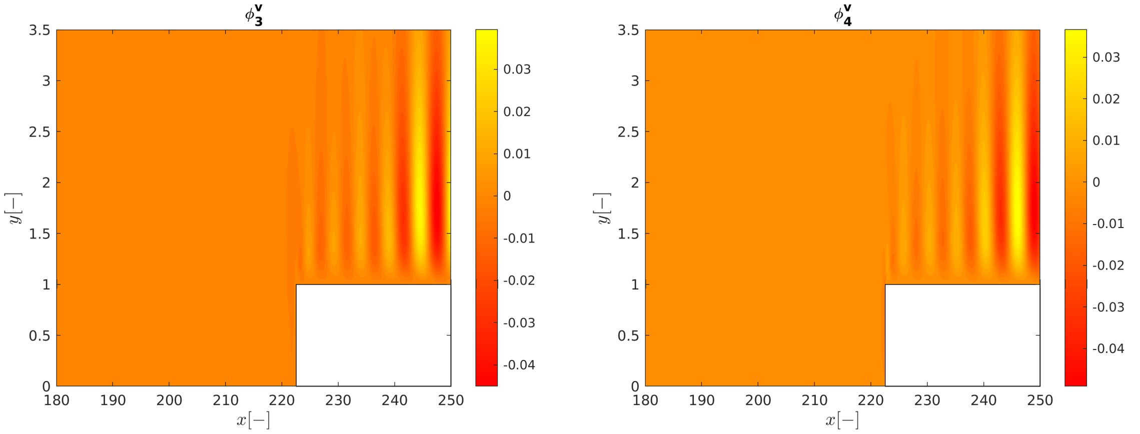
(d) Fourth GOROM mode

Figure E.12: GOROM modes of the x-velocity for case C.



(a) First GOROM mode

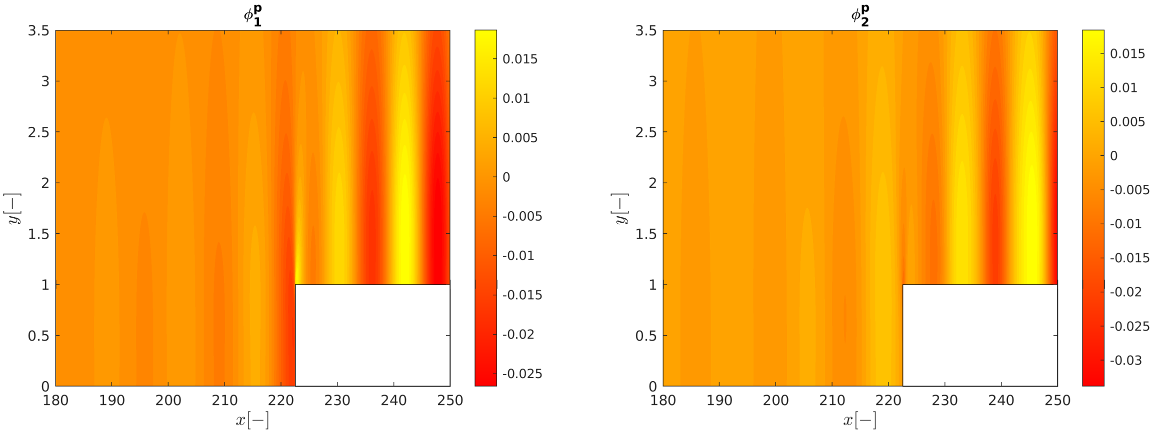
(b) Second GOROM mode



(c) Third GOROM mode

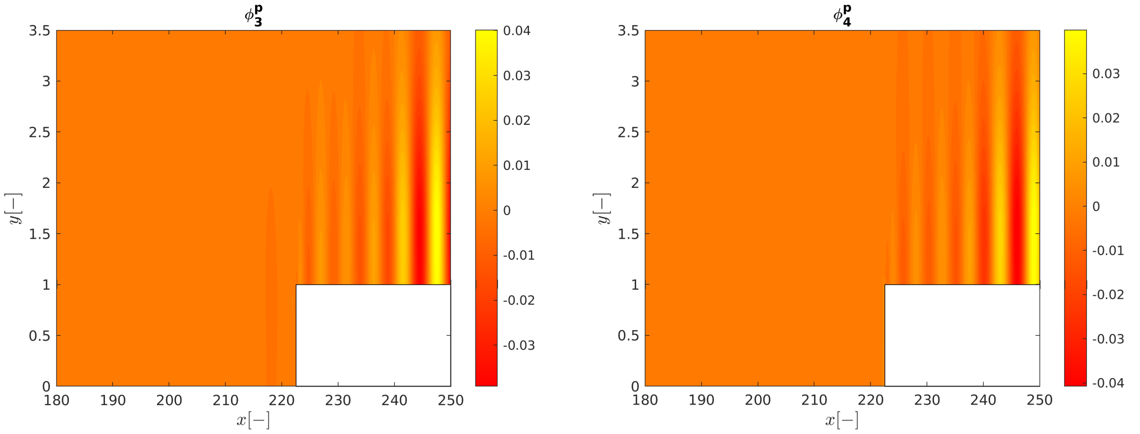
(d) Fourth GOROM mode

Figure E.13: GOROM modes of the y-velocity for case C.



(a) First GOROM mode

(b) Second GOROM mode

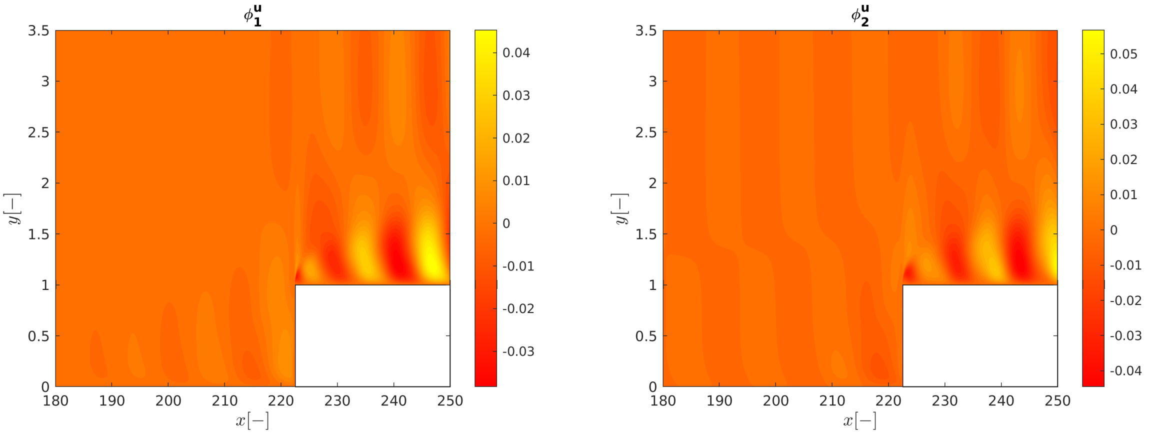


(c) Third GOROM mode

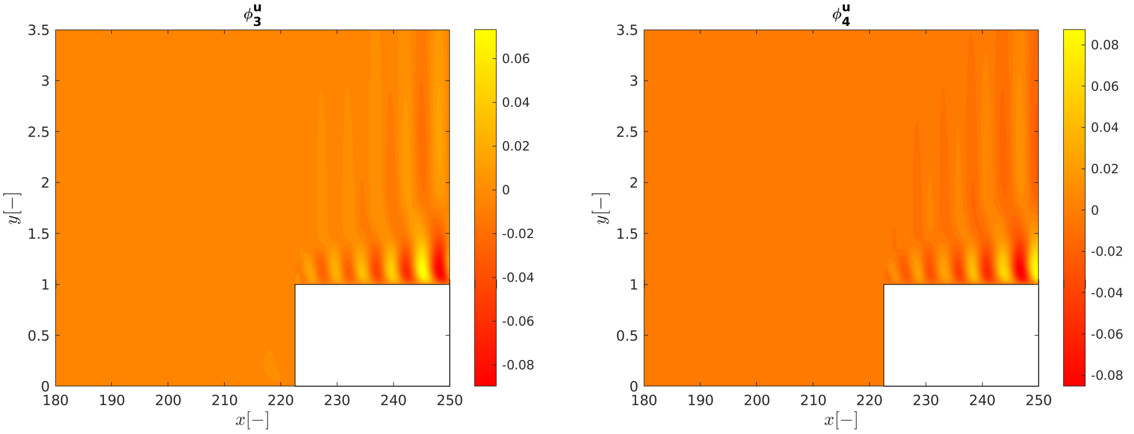
(d) Fourth GOROM mode

Figure E.14: GOROM modes of the pressure for case C.

E.3.3. Case H

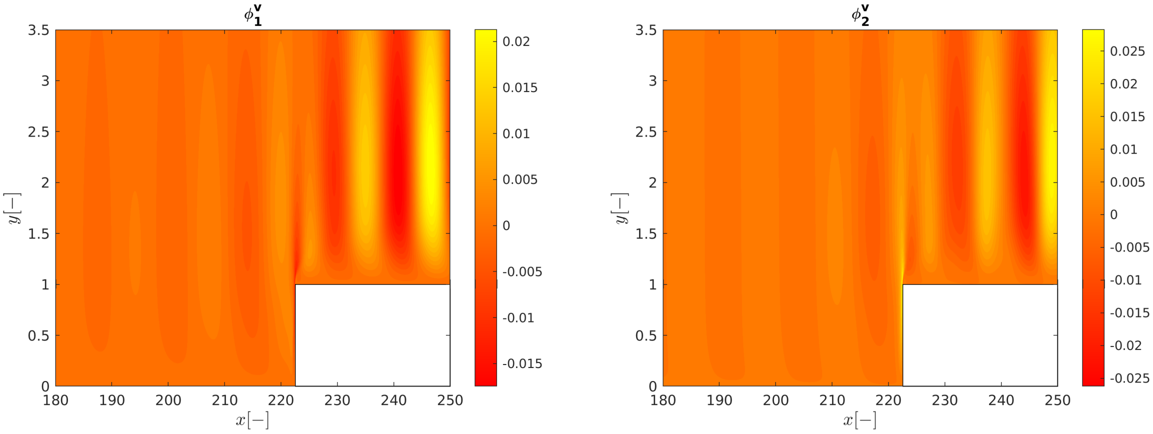


(a) First GOROM mode (b) Second GOROM mode



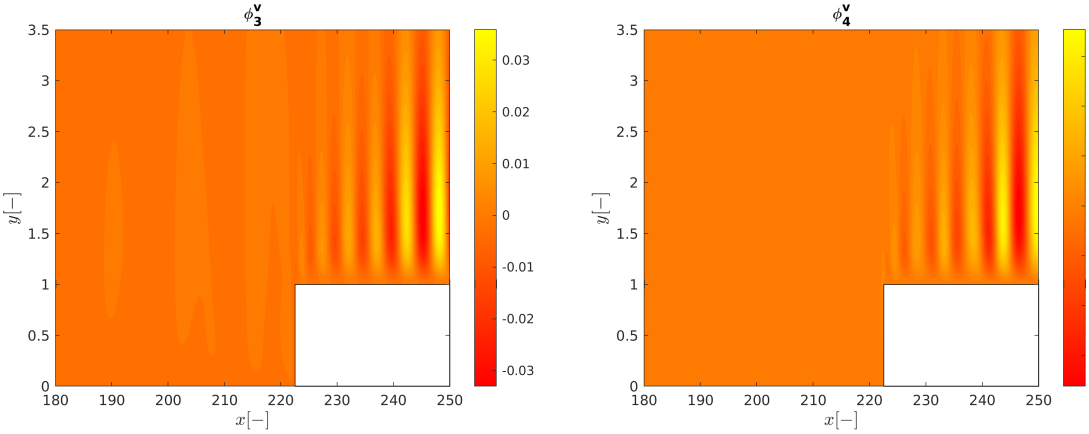
(c) Third GOROM mode (d) Fourth GOROM mode

Figure E.15: GOROM modes of the x-velocity for case H.



(a) First GOROM mode

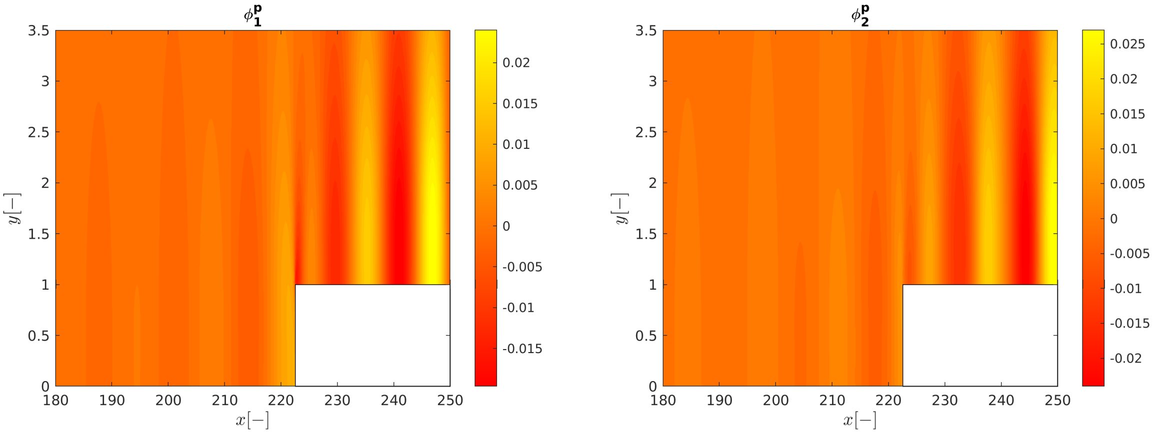
(b) Second GOROM mode



(c) Third GOROM mode

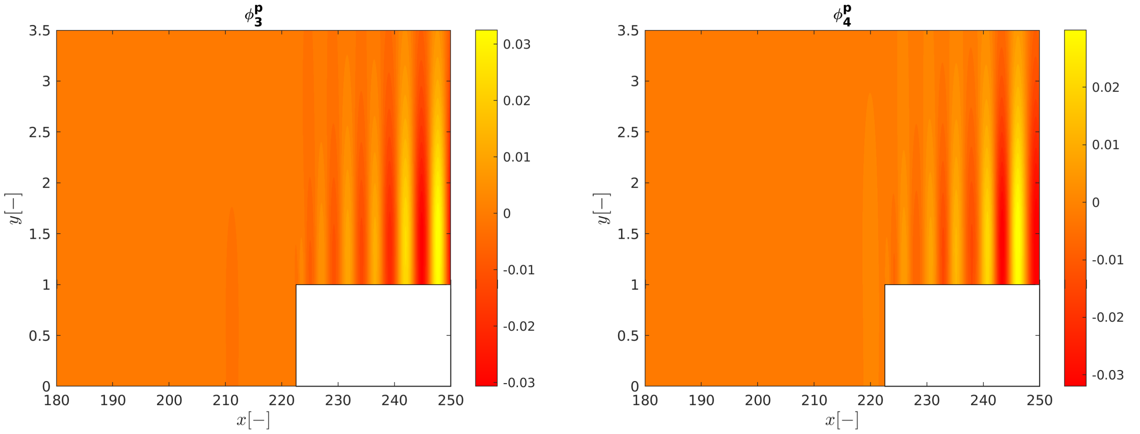
(d) Fourth GOROM mode

Figure E.16: GOROM modes of the y-velocity for case H.



(a) First GOROM mode

(b) Second GOROM mode



(c) Third GOROM mode

(d) Fourth GOROM mode

Figure E.17: GOROM modes of the pressure for case H.

Bibliography

Anderson, J. (2017). *Fundamentals of aerodynamics* (Vol. 6). New York: McGraw-Hill Education.

Bergmann, M., Bruneau, C.-H., & Iollo, A. (2009). Enablers for robust POD models. *Journal of Computational Physics*, *228*(2), 516–538.

Berkooz, G., Holmes, P., & Lumley, J. (2003). The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annual Review of Fluid Mechanics*, *25*, 539–575.

Boiko, A. V., Dovgal, A. V., Grek, G. R., & Kozlov, V. V. (2012). *Physics of transitional shear flow* (Vol. 98). Springer Science+Business Media B.V.

Borggaard, J., Wang, Z., & Zietsman, L. (2016). A goal-oriented reduced-order modeling approach for nonlinear systems. *Computers and Mathematics with Applications*, *71*(11), 2155–2169.

Brunton, S., Noack, B., & Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, *52*, 477–508.

Bui-Thanh, T., Willcox, K., Ghattas, O., & van Bloemen Waanders, B. (2007). Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, *224*(2), 880–896.

Casacuberta, J., Hickel, S., & Kotsonis, M. (2021). Mechanisms of interaction between stationary crossflow instabilities and forward-facing steps. *AIAA Scitech 2021 Forum*.

Chaturantabut, S., & Sorensen, D. (2009). Discrete empirical interpolation for nonlinear model reduction. *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 4316–4321. https://doi.org/10.1109/CDC.2009.5400045

Cheng, L. (2017). A semi-continuous formulation for goal-oriented reduced-order models [Phd Thesis].

Cheng, L., Mattei, S., Fick, P., & Hulshoff, S. (2015). A semi-continuous formulation for goal-oriented reduced-order models: 1d problems. *International Journal for Numerical Methods in Engineering*, *105*(6), 464–480.

Drake, A., Bender, A., Korntheuer, A., Westphal, R., Rohe, W., Dale, G., McKeon, B., & Geraschchenko, S. (2010). Step excrescence effects for manufacturing tolerances on laminar flow wings. *AIAA paper 2010-375*.

Fjørtoft, R. (1950). Application of integral theorems in deriving criteria of stability for laminar flows and for the baroclinic circular vortex. *Geofysiske Publikasjoner*, *17*(6), 1–52.

Gaster, M. (1962). A note on the relation between temporally-increasing and spatially increasing disturbances in hydrodynamic stability. *Journal of Fluid Mechanics*, *14*, 222–224.

Herbert, T. (1988). Secondary instability of boundary layers. *Annual Review of Fluid Mechanics*, *20*(1), 487–526.

Herbert, T. (1997). Parabolized stability equations. *Annual Review of Fluid Mechanics*, *29*, 245–283.

Hughes, T., Mazzei, L., Oberai, A., & Wray, A. (2001). The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of Fluids*, *13*(2), 505–512.

Ingen, J. (1956). A suggested semi-empirical method for the calculation of the boundary layer transition region. *Department of Aeronautical Engineering Delft: Technical Reports, VTH-74*.

Jansen, K., Whiting, C., & Hulbert, G. (2000). A generalized-α method for integrating the filtered navier-stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, *190*, 305–319.

Klebanoff, P. S., Tidstrom, K. D., & Sargent, L. M. (1962). The three-dimensional nature of boundary-layer instability. *Journal of Fluid Mechanics*, *12*, 1–34.

Lanczos, C. (1952). *The variational principles of mechanics* (Vol. 4). Oxford University Press.

Lu, K., Jin, Y., Chen, Y., Yang, Y., Hou, L., Zhang, Z., Li, Z., & Fu, C. (2019). Review for order reduction based on proper orthogonal decomposition and outlooks of applications in mechanical systems. *Mechanical Systems and Signal Processing*, *123*, 264–297.

Lumley, J. (1967). The structure of inhomogeneous turbulence. *Atmosperic Turbulence and Wave Propagation*, 166–178.

Nesselrooij, M., Veldhuis, L., Oudheusden, B., & Schrijer, F. (2016). Drag reduction by means of dimpled surfaces in turbulent boundary layers. *Experiments in Fluids*, *57*(9), 1–14.

Orr, W. (1907). The stability or instability of the steady motions of a perfect liquid and of a viscous liquid. part ii: A viscous liquid. *Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences*, *27*, 69–138.

Rayleigh, R. (1880). On the stability, or instability, of certain fluid motions. *Proceedings of the London Mathematical Society*, *9*, 57–70.

Reed, H., Saric, W., & Arnal, D. (1996). Linear stability theory applied to boundary layers. *Annual Review of Fluid Mechanics*, *28*, 389–428.

Reshotko, E. (2001). Transient growth: A factor in bypass transition. *Physics of fluids*, *13*(5), 1067–1075.

Reynolds, O. (1883). An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philosophical Transactions of the Royal Society of London*, *174*, 935–982.

Rist, U., & Fasel, H. (1995). Direct numerical simulation of controlled transition in a flat-plate boundary layer. *Journal of Fluid Mechanics*, *298*, 211–248.

Rowley, C. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, *15*(3), 997–1013.

Rowley, C., Mezi, I., Bagheri, S., Schlatter, P., & Henningson, D. (2009). Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, *641*, 115–127.

Saric, W. (1986). Visualization of different transition mechanisms. *Physics of Fluids*, *29*(9), 2770.

Saric, W., Reed, H., & Kerschen, E. (2002). Boundary-layer receptivity to freestream disturbances. *Annual Review of Fluid Mechanics*, *34*, 291–319.

Sartenaer, A. (1997). Automatic determination of an initial trust region in nonlinear programming. *SIAM Journal on Numerical Analysis*, *18*(6), 1788–1803.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2002). Incremental singular value decomposition algorithms for highly scalable recommender systems. *Fifth International Conference on Computer and Information Science*.

Schlichting, H. (1933). Zur entstehung der turbulenz bei der plattenströmung. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen: Mathematisch-Physikalische Klasse*, 181–208.

Schmid, P. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5–28.

Schmid, P., & Henningson, D. (2001). *Stability and transition in shear flows* (Vol. 142). New York: Springer-Verlag New York, Inc.

Serpieri, J., & Kotsonis, M. (2016). Three-dimensional organisation of primary and secondary crossflow instability. *Journal of Fluid Mechanics*, *799*, 200–245.

Shahzad, H. (2020). Flow stability around forward facing steps in unswept incompressible boundary layers [Msc Thesis].

Shajahan, L. (2016). Reduced order modeling using goal-oriented optimization [Msc Thesis].

Sieber, M., Paschereit, C., & Oberleithner, K. (2016). Spectral proper orthogonal decomposition. *Journal of Fluid Mechanics*, *792*, 798–828.

Sirovich, L. (1987). Turbulence and the dynamics of coherent structures. i - coherent structures. ii - symmetries and transformations. iii - dynamics and scaling. *Quarterly of Applied Mathematics*, *45*(3), 561–571.

Sommerfeld, A. (1908). Ein beitrag zur hydrodynamischen erklärung der turbulenten flüssigkeitsbewegungen. *Proceedings of the 4th International Congress of Mathematicians*, *3*, 116–124.

Squire, H. (1933). On the stability for three-dimensional disturbances of viscous fluid flow between parallel walls. *Proceedings of the Royal Society of London A*, *142*, 621–628.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, *20*(3), 626–637.

Taira, K., Brunton, S., Dawson, S., Rowley, C., Colonius, T., McKeon, B., Schmidt, O., Gordeyev, S., Theofilis, V., & Ukeiley, L. (2017). Modal analysis of fluid flows: An overview. *AIAA Journal*, *55*(12), 4013–4041.

Taylor, C., Hughes, T., & Zarins, C. (1997). Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering*, *158*, 155–196.

Tollmien, W. (1929). Über die entstehung der turbulenz. 1. mitteilung [translated as NACA TM 609]. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen: Mathematisch-Physikalische Klasse*, 21–44.

Torenbeek, E. (2013). *Advanced aircraft design : Conceptual design, analysis and optimization of subsonic civil airplanes.* (Vol. 1). John Wiley & Sons, Incorporated.

Towne, A., Schmidt, O., & Colonius, T. (2018). Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *Journal of Fluid Mechanics*, *847*, 821–867.

Tu, J., Rowley, C., Luchtenburg, D., Brunton, S., & Kutz, J. (2014). On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, *1*(2), 391–421.

Westerbeek, S. (2020). Development of a nonlinear parabolized stability equation (NPSE) analysis tool for span-wise invariant boundary layers [Msc Thesis].

Willcox, K., & Peraire, J. (2002). Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, *40*, 2323–2330.