OPTIMAL-CONTROL THEORETIC METHODS FOR OPTIMIZATION AND REGULATION OF DISTRIBUTED PARAMETER SYSTEMS

by

JENNIFER DAWN GOSS

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2009

To Brendan, my loving and supportive husband,

and to my parents Doug and Wilma Cunningham

who never said the words “you can’t”

ACKNOWLEDGEMENTS

The most difficult task I have found in completing this dissertation is to find the words of significant gravitas to adequately represent the depth of gratitude that is owed to so many for their support and assistance over many years. Without their support, advice and constructive criticism none of this would have been possible. Each person mentioned had a major impact on not only my research, but my successful journey to becoming an engineer.

I would like to thank my supervising professor Dr. Kamesh Subbarao. He has been a tremendous mentor, a true inspiration and an outstanding resource for me. He was always there to challenge and push me to produce more, better, faster. From very early on in my masters research it was clear that Dr. Subbarao was going to be an engineering force at the University of Texas at Arlington, and all of us who worked with him would be doing our absolute best just to keep pace. Over the next 6 years it was his indefatigable attitude and demand for excellence that set a bar for how thorough I had to be in all my research. This is something that has stuck with me as a measuring stick. I will use this in all my future engineering endeavors. It has truly been a pleasure to work with someone so enthusiastic about engineering and research.

I wish to also recognize the efforts of my committee members; Dr. Don Wilson, Dr. Seiichi Nomura, Dr. Frank Lu, and Dr. Hua Shan for their interest in my research and for taking time to serve in my dissertation committee. Dr. Don Wilson, the patriarch of the aerospace engineering department, was always available for advice and guidance throughout my graduate career. His insight to the magnetohydrodynamic

problem studied during my masters was invaluable. The finite element methods course, taught by Dr. Hua Shan, was essential in understanding the POD reduced order modeling approach used in depth in this dissertation. Dr. Nomura had the misfortune of teaching two of my favorite classes in engineering math, and therefore was someone I had to have on my committee. Special thanks to Dr. Frank Lu for his detailed review and notes. Thank you also to Pat Graham who took the time to review this dissertation long before it had order, enduring its narcoleptic effects. Your suggestions were greatly appreciated and indispensable.

I am grateful to the College of Engineering and the Department of Mechanical and Aerospace Engineering for their support throughout my studies. The assistantships and scholarships were greatly appreciated and eased the stresses of academic life. I must also mention two members of the administration staff, Sally Thompson and Janet Gober who were always available for advice, assistance and a warm smile. Thanks to Zonta International for honoring me with the Amelia Earhart Fellowship.

I would also like to recognize the help that I received from prior work completed in this field. The AUSM flow solver code and the smoothing function for the inlet grid generator were provided by Ramakanth Munipalli for my masters work and they were put to more good use here, thank you Ramakanth.

I would like to express my deepest gratitude to my husband, CFO, transportation coordinator and taskmaster, without whom this dissertation would not have been possible. My parents, in-laws and sisters and extended family were always supportive and encouraging.

Finally to everyone in the Aerospace Systems Laboratory, thank you for your endless support and laughter. It has been the journey of a lifetime and I am glad I was able to share it with all of you!

April 20, 2009

ABSTRACT

OPTIMAL-CONTROL THEORETIC METHODS FOR
OPTIMIZATION AND REGULATION OF
DISTRIBUTED PARAMETER
SYSTEMS

JENNIFER DAWN GOSS, Ph.D.
The University of Texas at Arlington, 2009

Supervising Professor: Kamesh Subbarao

Optimal control and optimization of distributed parameter systems are discussed in the context of a common control framework. The adjoint method of optimization and the traditional linear quadratic regulator implementation of optimal control both employ adjoint or costate variables in the determination of control variable progression. As well both theories benefit from a reduced order model approximation in their execution. This research aims to draw clear parallels between optimization and optimal control utilizing these similarities. Several applications are presented showing the use of adjoint/costate variables and reduced order models in optimization and optimal control problems.

The adjoint method for shape optimization is derived and implemented for the quasi-one-dimensional duct and two variations of a two-dimensional double ramp inlet. All applications are governed by the Euler equations. The quasi-one-dimensional

duct is solved first to test the adjoint method and to verify the results against an analytical solution. The method is then adapted to solve the shape optimization of the double ramp inlet. A finite volume solver is tested on the flow equations and then implemented for the corresponding adjoint equations. The gradient of the cost function with respect to the shape parameters is derived based on the computed adjoint variables.

The same inlet shape optimization problem is then solved using a reduced order model. The basis functions in the reduced order model are computed using the method of snapshots form of proper orthogonal decomposition. The corresponding weights are derived using an optimization in the design parameter space to match the reduced order model to the original snapshots. A continuous map of these weights in terms of the design variables is obtained via a response surface approximations and artificial neural networks. This map is then utilized in an optimization problem to determine the optimal inlet shape. As in the adjoint method of optimization, the methodology for a reduced order model is validated using the quasi-one-dimensional duct. The reduced order model is tested for efficiency and accuracy by performing an inverse optimization to match the pressure along the duct to a desired pressure profile. The method is then extended to generate a reduced order model for the two dimensional double ramp inlet. In this case, we optimize the inlet shape to minimize the mass weighted total pressure loss.

The optimal control problem addressed is a two-dimensional channel flow governed by the Burgers equation. An obstacle in the flow is utilized for the implementation of boundary control to influence the flow. The Burgers equation is written in the abstract Cauchy form to allow for the implementation of linear control routines.

The Riccati and Chandrasekhar equations are used to solve for the optimal control input to influence a region downstream of the obstacle. The results of both the controlled and uncontrolled scenarios are presented, and the Riccati and Chandrasekhar methods of gain calculation are compared. Reduced order modelling of the channel flow is performed using proper orthogonal decomposition and standard projection techniques. The reduced order model is then used for feedback control of the system in both set point and time-varying tracking problems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Acronyms

CFD	Computational Fluid Dynamics
CVT	Centroid Voronoi Tessellation
DPS	Distributed Parameter Systems
FEM	Finite Element Method
GLS	Galerkin Least Squares
LQR	Linear Quadratic Regulator
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PFA	Principal Factor Analysis
POD	Proper Orthogonal Decomposition
RBM	Reduced Basis Methods
ROM	Reduced Order Model

Roman

a	Weight coefficients
$\mathbf{A},\ \mathbf{B}$	Flux Jacobians
$\mathcal{A},\ \mathcal{B}$	State and control input matrices
c	Speed of sound
$\mathbf{C}$	Spatial correlation matrix
C_i	Number of control inputs on surface Γ_j
E	Total energy
$\mathbf{F},\ \mathbf{G}$	Flux Vectors

$\mathcal{F}$	Forcing term for inflow conditions
$\mathcal{G}$	Nonlinear operator
H	Hamiltonian
I	First variation of the cost function
J	Cost function
$\mathcal{K}$	Functional gain
$K_1,\ K_2$	Convective scaling constants in the Burger equation
L	General cost function at intermediate times/steps
M	Mach number
$\mathbf{n}$	Normal vector
p	Pressure
p_o	Total pressure
$\mathcal{P}$	Solution to the Riccati equation
Re	Dissipation scaling constant in the Burger equation
$\mathbf{R}$	Residual operator
R	Gas Constant
s	Shape function for 1-D duct
$\mathcal{S},\ \mathcal{Q},\ \mathcal{R}$	Weight matrices for linear quadratic cost function
t	Time
T	Temperature
$u,\ v$	Velocity components
$\mathbf{U}$	State variables
$\mathbf{V}$	Velocity vector
w_i	Eigenvector components
$\mathbf{W}$	Eigenvectors
$x,\ y$	Spatial coordinates

Greek

$\boldsymbol{\alpha}$	Control variables
α_s	Shift parameter for the LQR implementation
Γ	Boundary of the domain
λ	Eigenvalues
$\boldsymbol{\Lambda}$	Lagrange multiplier
	Costate / adjoint variables
Ω	Domain
ϕ	Control influence
$\boldsymbol{\Phi}$	Basis functions
ψ	Cost function at final time/step
ρ	Density

Symbols

$\tilde{(\cdot)}$	Variation in the quantity $(\cdot)$
$*$	Multiplication
$\otimes$	Tensor product

CHAPTER 1

INTRODUCTION

1.1 Dissertation objectives and contributions

Main objective: Formulate a framework through optimal control theory to elucidate methods for solving optimization and optimal control problems using adjoint and reduced order methods for general distributed parameter systems (DPS).

Supporting objectives:

1. Investigate methods to derive and solve the adjoint system of equations for DPS.
2. Research intelligent control techniques of DPS
3. Investigate methods to generate reduced order models for general DPS.
4. Use the above methods in optimization and optimal control problems, recognize the commonalities and illustrate them through the framework formulation.

Research contributions: The following are the main contributions of the proposed work,

a. Domain dependent adjoint formulation

b. System-theoretic approach to POD

c. Inlet shape optimization using POD and Adjoint formulations

d. Intelligent boundary control of DPS (2-D Burgers equation)

Published articles based on current research

1. Jennifer Goss and Kamesh Subbarao, *'Adjoint Methods for Inlet Shape Optimization',* AIAA 38th Fluid Dynamics Conference and Exhibit, Seattle, Washington, 23-26 June, 2008.
2. Jennifer Goss and Kamesh Subbarao, *'Inlet Shape Optimization Based on POD Model Reduction of the Euler Equations'*, 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia, Canada. 10-12 September, 2008.
3. Jennifer Goss and Kamesh Subbarao, *'Inlet Shape Optimization for a Two-Dimensional Hypersonic Flow using Adjoint Methods',* AIAA 39th Fluid Dynamics Conference and Exhibit, San Antonio, Texas, 22-25 June, 2009.

Publications under preparation

1. Jennifer Goss and Kamesh Subbarao, *'Inlet Shape Optimization Based on POD Model Reduction of the Euler Equations' - To be submitted to AIAA Journal*
2. Jennifer Goss and Kamesh Subbarao, *'Intelligent Boundary Control of DPS (2-D Burgers Equations)' - To be submitted to American Control Conference, 2010 & IEEE Transactions in Neural Networks*

1.2 Motivation and background

Optimal control and optimization can be regarded as related problems. In many flow problems optimization is performed on steady-state flow results to determine an optimal geometric configuration (parameterized as control variables). Therefore, optimal control can be considered as a dynamic optimization problem where the control variables are optimized over time. Both problems seek to minimize or maximize some performance index while satisfying specified constraints. Adjoint methods based op-

timization and optimal control problems using the linear quadratic regulator (LQR) approach are equivalent on steady and unsteady systems respectively. Both require the solution of a set of costate variables which are then used to update the control variables. The use of reduced order models for the governing input/output descriptions in both problems dramatically reduce the computational costs. Drawing on these similarities, a framework is formulated for both the optimization and the optimal control problems, the basis of which is illustrated in Fig.1.1.

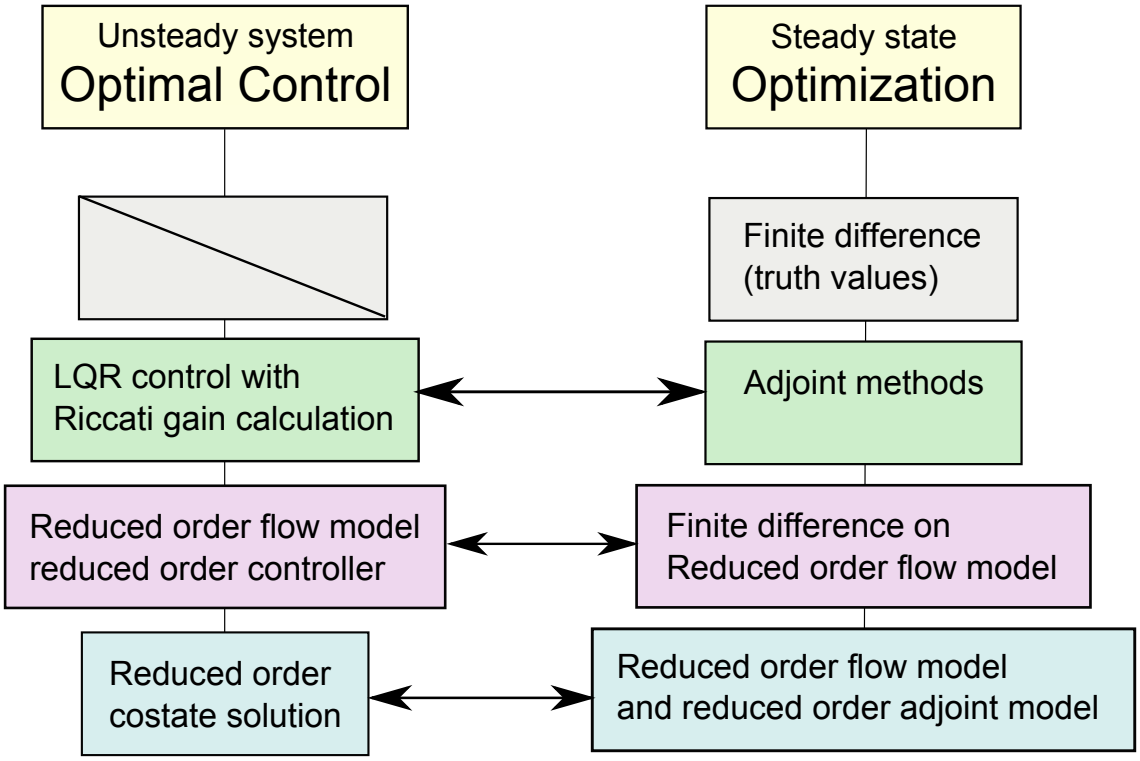


Figure 1.1. Flow chart demonstrating the correlations between optimal control and optimization.

1.3 Relation to state-of-the-art research and literature survey

1.3.1 Optimization and control of discretized distributed parameter systems

Most physical systems are distributed parameter systems. They are, in general, systems described by partial differential equations (PDEs) and their state space is infinite dimensional. The control of these systems is then in terms of a time-dependent PDE with control actions either distributed throughout or applied at the boundaries of the spatial domain. Design of such control, given today's state-of-technology, requires simplification of both the system model and the control strategy. Special care needs to be taken in maintaining the intrinsic characteristics of the system while still allowing the implementation of known control strategies.

Control theoretic approaches for DPS have made great strides since Lions' (1971) landmark book on optimal control [1]. Early developments around that time addressed issues of controllability and stabilization in finite and infinite dimensions. Formulation of linear distributed control problems was fairly complete by the late seventies including the extension of LQR theory and the related Riccati equations [2].

Optimization problems, specifically geometric shape design problems, can be considered in the context of control theory. If the boundary of the shape is given as the control variables, then optimal control theory allows for the determination of an optimal set of those shape variables to extremize some performance cost function. For instance, the goal of aerodynamic shape optimization is the determination of a geometric shape that minimizes some performance functional without violating specific geometric and/or aerodynamic constraints. To this end, there are essentially three sources to the computational cost of finding this optimal geometry: 1) evaluating the

flow, 2) evaluating the gradient, 3) the number of iterations and gradient evaluations needed to minimize the performance function. The adjoint method of optimization presented in the next chapter addresses the third issue above that leads to the evaluation of the gradient. Later in the investigation of reduced order models, the first issue will be addressed.

Optimal control problems are essentially dynamic optimization problems. The set of optimal control inputs varies with time such that the system is coerced along a desired trajectory. This is a powerful approach especially when combined with feedback information for the control of flow separation, drag reduction, and lift enhancement just to mention a few aerodynamic applications. Feedback control of dynamical systems is simply a formula relating the quantity being measured by a sensor to the quantity being controlled by an actuator. Various methods and examples of optimal feedback control will follow the next section.

Adjoint-based optimization

The main advantage of the adjoint approach to design is in the calculation of the gradient of the cost function. Traditional gradient-based optimization methods require the calculation of the flow parameters for an initial state as well as for a perturbation in each of the design variables. For a large number of design variables this can become computationally prohibitive. The adjoint variables which can be solved with equal computational cost of one flow solution can be used to compute the gradient without the need for additional flow solutions.

Limitations to the adjoint approach to design include [3]

- **System constraints**: Soft constraints or constraints on the geometry of the problem are easily dealt with through adjustments to the search direction after the gradient calculation. Hard constraints or constraints on the flow variables which must be satisfied at all stages of the optimization can only be enforced through additional Lagrange multiplier constraints on the cost function. These would then require the solution of additional adjoint systems to determine their sensitivities. Therefore if the number of hard constraints is almost equal to the number of design variables then all benefits of the adjoint method are lost.
- **Convergence**: If the cost function is of a least-squares type, then the direct linear perturbation approach to optimization provides an approximation to the Hessian leading to rapid convergence. Conversely, the adjoint method does not provide any information on the Hessian and will require more steps to converge.
- **Local optima**: Given that the adjoint method provides an alternative means of generating the gradient of the cost function it is still subject to all the limitations of gradient based optimization. The design variables must be continuous and convergence is only guaranteed to a local minimum. Careful choice of the cost function can help alleviate the latter.

The approach of using adjoint variables in shape optimization has been in practice for many years. The basis for the adjoint methods we know today originated with Lions [1] use of mathematical theory for control of system governed by partial differential equations. This work led Pironneau (1974) [4] to use adjoint methods for shape optimization in fluid dynamics problems. Modern studies on the use of adjoint methods in aerodynamic shape optimization were pioneered by Jameson [5, 6]. The first decade of the twenty-year history of adjoint methods in aerodynamic optimization progressed quickly, from potential flow equations to Euler then to Navier-Stokes

equations and from 2-D airfoil design to 3-D wings and then to full aircraft configurations. The last ten years have seen refinements in the implementation of these methods.

Early investigations into adjoint methods brought thorough theoretical studies. Giles and Pierce[7, 8, 9] derived analytic solutions for 1-D and 2-D flows governed by the Euler equations through the use of Green's functions. These studies were verified numerically. Major topics emphasized in the studies by Giles and Pierce included the proper derivation and implementation of the boundary conditions and numerical error analysis. More recent studies by Ferlauto et al.[10] have shown that internal flows require additional studies into the well posedness of the boundary conditions. Numerical techniques for external flows such as airfoil and wing design cannot be applied directly to internal flows due to the strong coupling between inner flow fields and their boundaries.

Airfoil and wing optimization problems have been studied by many groups with the majority imposing inverse design routines for matching a given pressure distribution and/or drag reduction [11, 12, 13, 14, 15, 16, 17, 18, 19]. Jameson[12, 13] implemented a continuous adjoint formulation on wings, wing-body configurations and full aircraft. The geometry was modified using a grid perturbation technique which proved to be difficult and computationally expensive, leading to the use of unstructured grids. Xie[14] performed airfoil optimization where the shape was parameterized by Hicks-Henne functions and the NACA 4-digit airfoil series. Xie added the angle of attack as an additional control variable to generate a super-reduced design formulation. Both cases were optimized to minimize the drag at a fixed lift coefficient. Anderson and Venkatakrishnan[15] utilized a mixed continuous and discrete adjoint

formulation to draw on the benefits of each method. They also suggested a unique design problem using two offset airfoils, the purpose being to influence the pressure distribution on one airfoil by modifying the shape of the other. This test demonstrated a flexibility of the adjoint method that would be impossible with traditional inverse design methods.

Adjoint methods have also been used on internal flow studies; Iollo and Salas[20] applied the adjoint method to the optimization of 2-D internal flows with embedded shocks. Iollo et al.[21] implemented an adjoint optimization of a diffuser and turbomachinery. The wall pressures of the diffuser were optimized to minimize axial flow deviations at the outlet. In the turbine blade problem, the objective function was to maximize thrust with the control being the force on the turbine blade. Note that in [21], the control variables were not shape geometries but flow variables. This approach to flow design removed the need to impose additional flow constraints and hence the need for additional adjoint solutions. Zhang and Lum[22] utilized the adjoint method in optimizing an S-shaped duct with fixed entrance and exit geometry. The centerline and radius functions for the inlet were parameterized with Hicks-Henne functions and the objective function minimized the wall pressure drag in the x-direction. This was an indirect way of minimizing the exit face total pressure distortion while minimizing the loss in total pressure.

Feedback control of distributed parameter systems

Feedback control is still evolving in the field of aerodynamic flow control. Difficulties in solving large systems of nonlinear equations resulting from feedback control have limited advancements in this field until recently. High performance computing platforms and new numerical algorithms are allowing for the development of tools

for design of realistic 3-D flow control problems. Some examples of optimal feedback control follow.

Choi et al.[23] implemented a sub-optimal control and feedback routine using the adjoint method described above and a gradient algorithm. The optimization problem was executed to compute the control input at each time step of the control implementation. This is a clear example of the relationship between optimization and optimal control. The method, though not the most efficient, was effective in controlling the stochastic Burgers equation turbulence model. Kang et al. [24] implemented distributed and boundary control on the 1-D Burgers equation through feedback control derived from a linearized flow model. The control was effective at relaxing steep gradients in the flow variables. Burns et al.[25] utilized boundary control to influence the boundary layer by reducing the energy in the flow near the boundary. The simulation was performed using the 1-D Burgers equation and Dirichlet boundary conditions.

Atwell and King[26] utilized feedback control to stabilize the FEM solutions of open and closed loop simulations of the Burgers equation. The Galerkin FEM solution is unstable for low viscosity coefficients of the Burgers equation. However, the Galerkin Least Squares (GLS) FEM has added stabilizing terms and a broader stable range for the viscosity. LQR feedback control designed from the unstable system is effective at stabilizing the Galerkin and GLS approximations. King and Krueger[27] then investigated how these stabilizing terms affected the resulting controller and found the influence to be minimal.

Comparison of the Riccati and Chandrasekhar equations in computing the functional gains for the LQR problem is a topic of several studies. Borggaard et al.[28] conducted this comparison on the 2-D heat equation and the 2-D Stokes flow. Camphouse and Myatt[29] also ran this comparison on the Burgers equation and added a study of the influences of various boundary conditions on the functional gains and resulting control development. The Chandrasekhar equations themselves have been the topic of many studies, Lainiotis [30] derived a generalized algorithm as a solution to the matrix Riccati equation for the application to time varying models. Ito and Powers[31] derived the equations for infinite dimensional systems using functional analysis techniques.

1.3.2 Optimization and control using reduced order models

There are several methods available to generate reduced order models of infinite dimensional DPS and all of them can essentially be categorized as reduced basis methods. Finite difference and finite element solutions are also considered reduced order models in that they approximate a system governed by partial differential equations. Whilst finite difference methods use grid functions, finite element methods use piecewise polynomials as basis functions [32].

The first presentations of the principal orthogonal decomposition (POD) method for model reduction were independent works by Loeve (1945) [33] and Karhunen (1946) [34]. after which the method was known as the Karhunen-Loeve (K-L) expansion. The method has been called by many names over the years including principal component analysis (PCA), principal factor analysis (PFA), and Hotalling transformation. It was not until 1967 when Lumley first named the method “Proper Orthogonal Decomposition” in his studies of turbulent structures. The early uses of the

POD method involved studies of coherent structures in turbulent flows which also led Sirovich [35] to introduce the method of snapshots in 1987. Today POD model reduction has been used in a wide range of scientific fields including; image processing, signal analysis, data compression, oceanography, processes identification, and control in chemical engineering. More details on the history and a thorough development of the POD method can be found in Ref. [36].

Within the fields of control and optimization, reduced order modeling has been studied to great extent. Early work by Ito and Ravindran [32] investigated the use of various reduced basis methods (RBM). With these methods, the basis functions are generated from the problem being solved through either a Lagrange or Taylor method or a hybrid of both known as the Hermite method. The RBM's often require many basis functions which typically contain redundant information. There is apparently no systematic way to reduce the number of basis. Later, Ravindran [37] went on to compare the RBM with the POD method. The POD method provides a means of sorting the basis functions which allows the dominant modes to be clearly identified and extracted to obtain a *reduced* reduced-basis model. Other methods of determining dominant modes are also available including a centroid Voronoi tessellation (CVT) method presented by Burkardt et al.[38]. The CVT method naturally introduces the concept of clustering in the construction of the basis functions and is very useful in a variety of applications including; optimal representation, quantization, cell division, optimal distribution of sensors and actuators, grid generation, etc. See Ref.[38] for more details on the CVT method.

Once the basis functions are generated, there are a variety of methods available to project the governing equations onto the reduced order space. The most common

method is the Galerkin approach of which there are several minor variations including the discontinuous Galerkin approach. In Ref.[39], Iollo studies the stability and accuracy of the Galerkin projection methods for several 1-D problems. The generally accepted rule is that any POD-Galerkin scheme based on an underlying stable finite difference or finite volume scheme is also stable. This is not necessarily the case. It is found that Galerkin POD schemes for Euler equations need additional stabilization from that provided by a straight forward discretization of the projection of the equations in finite dimensional space. This result was corroborated by Lucia and Beran [40] who compared Galerkin projection with a direct projection based on a Taylor series expansion. This work found the direct projection to require less stabilizing influence. In both cases as with other projection methods in general, they are very difficult to apply to nonlinear systems such as the Euler and Navier Stokes equations, and the results are not very robust. LeGresley and Alonso [41, 42] presented a finite volume projection that is solved using a nonlinear least squares methodology. This method has similarities to the method presented in this dissertation in that we also use a sub-optimization routine to determine the approximate flow solution given the set of basis modes (see sections 4.4 and 5.4).

Optimization and optimal control problems have been addressed using POD model reduction on various flow systems such as; the heat equation [43, 44, 45], Burgers equation [43, 46], Euler equations [41, 42] and the Navier-Stokes equations [37]. The heat equation studies have involved optimal control problems with LQR controllers [44] and adaptive critic neural networks [45] in the solution of the control input. The Burgers equation studies have included suboptimal feedback control implementation [46] and studies into design-then-reduce vs. reduce-then-design perspec-

tives. Optimization studies have mainly concerned airfoil shape optimization governed by the Euler equations [41, 42].

Feedback control using POD models

Special consideration needs to be taken when applying POD model reduction to a system subject to feedback control. This consideration is focused on the generation of the snapshots. Specifically, basis functions generated from a snapshot ensemble of uncontrolled or even open loop control data may not correlate with the energy of the system under feedback control. An interesting solution to this problem was presented by Camphouse[44] who utilized the functional gains to select the dominant modes of the POD model instead of the standard energy function. The functional gains indicate states that are most significant to the controller and proved to be an effective means of determining the model fidelity. This method is an excellent option where optimal feedback control design from the full order system is difficult. Ravindran[47] utilized an adaptive technique to improve the POD basis iteratively resulting in a set of modes that best represent the influence of the control action. A set of snapshots are initially generated through a linear profile of control inputs. The model is generated and the controller is applied to the model for a given time interval and more snapshots are generated. The original snapshot ensemble is then either augmented or replaced with the new set of snapshots. Ravindran found that concatenating the snapshots results in larger and larger numbers of basis required to capture a fixed amount to energy whereas replacing the ensemble each time keeps the number of basis fixed.

POD model reduction has been applied to many complex systems for feedback control. For example, Ravindran[48, 49] implemented reduced order controllers for flow past an airfoil with the goal of controlling the vortex shedding behind a thin

airfoil. The optimal control was found to be a downstream directed unsteady blowing from the leading edge of the airfoil. This blowing mitigates flow separation which in turn suppresses vortex shedding. Singh et al.[50] used feedback control to control unsteady wake flow behind a circular cylinder. Buffoni et al.[51] have solved the full 3-D problem of confined wake flow behind a square cylinder. Djouadi et al.[52] compared POD model reduction with a balanced truncation scheme with similar results.

The next chapter develops the general theory of the adjoint formulation, intelligent control of DPS and reduced order models. The following chapters apply this theory to specific flow problems for optimization and optimal control. Three different shape optimization problems are considered in this dissertation; a quasi-1-D duct (chapter 3), a full inlet (chapter 4), and a cowl style inlet (chapter 5). All of these cases are governed by the Euler equations. The 1-D duct has been addressed in other works and had well published results for the optimization routines presented here. The problem allows for the validation of the methods derived herein and verification of their implementation. The methodologies are then applied to the full inlet problem which is also well published for the conventional optimization problem. The main component of the unique contribution of this dissertation comes in the final case of the cowl style inlet. Chapter 6 considers an optimal control problem for a channel flow governed by the Burger equation. This case has been published in previous works and provides full coverage of the optimization/regulation analogy we wish to clarify. Table 1.1 summarizes chapter and section information for the case studies

Table 1.1. Chapter and section number for the case studies

	Optimization			**Optimal Control**
	1-D Duct	2-D Full Inlet	2-D Cowl Inlet	2-D Channel
Conventional	——	4.2	5.2	——
Adjoint	3.2	4.3	5.3	6.2
POD	3.3	4.4	5.4	6.3

CHAPTER 2

OPTIMIZATION AND CONTROL OF DISTRIBUTED PARAMETER SYSTEMS

2.1 General description of the problem

A general description of a system with state variables $\mathbf{U}$ and some control variables $\boldsymbol{\alpha}$ can be written as follows,

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{R}(\mathbf{U}, \boldsymbol{\alpha}) \tag{2.1}$$

The operator $\mathbf{R}$ may contain many different operations on the state variables. In general, we consider this operator to contain spatial derivatives of the state variables, giving us a set of partial differential equations. An example of a system that can be written in this form is given by the 2-D Euler equations for a gas

$$\frac{\partial \mathbf{U}(x, y, \boldsymbol{\alpha})}{\partial t} = -\frac{\partial \mathbf{F}(\mathbf{U}, x, y, \boldsymbol{\alpha})}{\partial x} - \frac{\partial \mathbf{G}(\mathbf{U}, x, y, \boldsymbol{\alpha})}{\partial y} \tag{2.2}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u(E + p) \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v(E + p) \end{bmatrix}$$

where ρ is the density, u, v are the x and y components of the velocity such that $\mathbf{V} = (u, v)$, E is the total internal energy, and p is the pressure, while $\boldsymbol{\alpha}$ are the

control variables. To close the system we assume that the perfect gas relationship holds

$$E = \frac{p}{\rho(\gamma - 1)} + \frac{1}{2}(u^2 + v^2), \qquad p = \rho RT, \qquad c = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\gamma RT}$$

where $\gamma = 1.4$ is the ratio of specific heats, and T, R and c are the temperature, the gas constant and the speed of sound respectively.

In both *optimal boundary control* and *optimization* problems, the goal is to ***extremize*** some scalar performance measure. We will assume that the performance measure/cost function can be written in the general form

$$J_i = \psi(N, \mathbf{U}_N) + \sum_{k=i}^{N-1} L^k(\mathbf{U}_k, \boldsymbol{\alpha}_k) \tag{2.3}$$

Depending upon what problem is being solved, the interval $[i, N]$ is the time interval of interest or the optimal parameter sequence index. The latter implies that J_i is the cost at the i^{th} iteration. An optimal correction to $\boldsymbol{\alpha}_i$ is sought that will seek to extremize J_{i+1}. If the optimization problem is solved at each step, we have a one-step ahead predictor-corrector set of equations. For completion, we mention that ψ is a function of the final time/step, $\mathbf{U}_N$ the state at the final time/step and L^k is a general function of the state and control input at intermediate steps, k.

In summary, the *optimal control* problem is to find the control $\boldsymbol{\alpha}_k$ on the interval $[i, N]$ that drives the system (Eq. 2.1) along a trajectory $\mathbf{U}_k$ such that the cost function (Eq. 2.3) is extremized. On the other hand, the *optimization* problem is the extremization of an objective function which contains a set of variables that will affect the value of the objective function, subject to a set of constraints (the flow

equations plus any parametric bounds). In this case the objective function must be differentiable on the domain Ω and an extremum must exist.

Minimization/maximization of a functional: Let J, our objective function, be a functional defined on a fixed nonempty set Ω of a normed vector space $\mathcal{X}$. The necessary condition for an extremum is given as: [53]

Theorem 1 *(Necessary condition for an extremum of a functional) If a functional J defined on an open set Ω contained in a normed vector space $\mathcal{X}$ has a local extremum at a vector $\mathbf{U}^*$ in Ω, and if J has a variation at $\mathbf{U}^*$, then the variation of J at $\mathbf{U}^*$ must vanish; that is*

$$\delta J(\mathbf{U}^*; \boldsymbol{h}) = 0 \quad \textit{for all vectors } \boldsymbol{h} \textit{ in } \mathcal{X}$$

must hold.

Proof The proof can be found in Ref.[53].

Minimize/maximize a cost functional subject to constraints: Consider two functions J and R both defined on the fixed non-empty set Ω of a normed vector space $\mathcal{X}$. The goal is to find the extremum vectors $\mathbf{U}^*$ for J among all those vectors $\mathbf{U}$ in Ω which satisfy the constraint $R = r_o$. We assume there exists a non-empty subset in Ω which satisfy the constraint; $\Omega[R = r_o]$, in general this is not an open set in $\mathcal{X}$. Now the problem is to find all local extrema in the subset $\Omega[R = r_o]$ for J. However, since the subset $\Omega[R = r_o]$ is not guaranteed to be an open set, the variation in J

does not necessarily vanish at a local extremum vector $\mathbf{U}^*$ in $\Omega[R = r_o]$. Therefore, the previous theorem for the necessary condition of an extremum may not hold.

$$\delta J(\mathbf{U}^*; \boldsymbol{h}) \neq 0 \quad \text{for all vectors } \boldsymbol{h} \text{ in } \mathcal{X}$$

To solve this new problem for the local extremum vector in $\Omega[R = r_o]$ for J we look to the Euler-Lagrange multiplier theorem for a single constraint. The following theorem is stated without proof [53].

Theorem 2 *(Euler-Lagrange Multiplier Theorem for a Single Constraint) Let J and R be functionals which are defined and have variations on an open set Ω contained in a normed vector space $\mathcal{X}$, and let $\mathbf{U}^*$ be a local extremum vector in $\Omega[R = r_o]$ for J, where r_o is any given fixed number for which the set $\Omega[R = r_o]$ is nonempty. Assume that both the variations of J and R are weakly continuous near $\mathbf{U}^*$. Then at least one of the following two possibilities must hold:*

1. *The variation of R at $\mathbf{U}^*$ vanishes identically, i.e.,*

$$\delta R(\mathbf{U}^*; \Delta\mathbf{U}) = 0$$

   for every vector $\Delta\mathbf{U}$ in $\mathcal{X}$; or

2. *The variation of J at $\mathbf{U}^*$ is a constant multiple of the variation of R at $\mathbf{U}^*$, i.e.,there is a constant λ such that*

$$\delta J(\mathbf{U}^*; \Delta\mathbf{U}) = \lambda \delta R(\mathbf{U}^*; \Delta\mathbf{U})$$

   for every vector $\Delta\mathbf{U}$ in $\mathcal{X}$

The second condition in the above theorem guarantees that the level curves/ surfaces of the two functions are tangential at all extremum vectors. In other words the two surfaces $D[J = J(\mathbf{U}^*)]$ and $\Omega[R = r_o]$ coincide at local extremum vectors $\mathbf{U}^*$ if the first condition fails.

To add a constraint requirement to our general cost function, we follow the method given in the above theorem. The general cost function is augmented to include the constraint that the flow equations are also satisfied, resulting in the Hamiltonian

$$H^k = L^k + \mathbf{\Lambda}_{k+1}^T \mathbf{R}_k \tag{2.4}$$

By introducing the Lagrange multiplier, we replaced the problem of minimizing $L^k(\mathbf{U}_k, \boldsymbol{\alpha}_k)$ subject to a constraint $\mathbf{R}_k = 0$ with the problem of minimizing the Hamiltonian $H^k(\mathbf{U}_k, \boldsymbol{\alpha}_k, \mathbf{\Lambda}_k)$ without constraints. This form of the objective function is used in both the optimal control and optimization procedures that follow.

2.2 Domain dependent adjoint formulation

Consider the augmented cost function mentioned earlier (Eq. 2.4) and restated here

$$H(\mathbf{U}, \boldsymbol{\alpha}) = L(\mathbf{U}, \boldsymbol{\alpha}) - \mathbf{\Lambda}^T \mathbf{R}(\mathbf{U}, \boldsymbol{\alpha})$$

General perturbations $d\mathbf{U}$ and $d\boldsymbol{\alpha}$ result in a variation of the cost function given as

$$dH = \left(\frac{\partial L}{\partial \mathbf{U}} - \mathbf{\Lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right) d\mathbf{U} + \left(\frac{\partial L}{\partial \boldsymbol{\alpha}} - \mathbf{\Lambda}^T \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} \right) d\boldsymbol{\alpha}$$

Choose $\mathbf{\Lambda}^T$ to satisfy the adjoint equation

$$\frac{\partial L}{\partial \mathbf{U}} - \mathbf{\Lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{U}} = 0 \qquad \Rightarrow \qquad \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \mathbf{\Lambda} = \left(\frac{\partial L}{\partial \mathbf{U}}\right)^T$$

The remaining terms in the variation of the cost function are no longer dependent on changes in the flow variables and are strictly functions of the design parameters $\boldsymbol{\alpha}$

$$dH = \left(\frac{\partial L}{\partial \boldsymbol{\alpha}} - \mathbf{\Lambda}^T \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}\right) d\boldsymbol{\alpha}$$

The adjoint derivations for shape optimization problems, i.e., the 1-D duct flow as well as the 2-D double ramp inlet are detailed in subsequent chapters. In chapter 3 the quasi 1-D duct flow problem is investigated to validate the adjoint and gradient computation procedures. These results correlate well with previously published results[14]. In this case, the duct shape is obtained via an inverse design process to match the pressure for some pre-specified shape. The validation of the adjoint variables is achieved based on the analytical solution developed for this specific problem[9].

In chapters 4 and 5 the derivation of the adjoint variables is extended to the 2-D double ramp inlet governed by the Euler equations. These case studies address the development of the adjoint method to maximize the mass weighted total pressure loss at the exit of a double ramp inlet with and without a cowl, where the control variable is the shape of the inlet wall. In the present work, we derive the adjoint variables and outline a procedure to compute the gradient based on the adjoints. The computation of the gradients in this case is *unique* as the cost-function sensitivity to the design variables turns out to be a *domain integral* as opposed to boundary

integrals as in most prior literature. The method of deriving and utilizing the adjoint variables is a fairly general one but has to be redone on a case-by-case basis. The resulting adjoint system of equations to be solved is unique to each problem owing to differences in the boundary conditions and the cost function. Any changes in the flow boundary conditions for different scenarios result in changes in the adjoint boundary conditions. As well any modification in the cost function results in changes to the adjoint system. Future applications of this work will better utilize the adjoint formulation by increasing the number of design variables for a smooth inlet shape variation.

2.3 Intelligent control of DPS

In this section, we summarize the details of the control problem that usually accompanies distributed parameter systems. Consider the same cost function as above now in linear quadratic form

$$J_i = \frac{1}{2}\mathbf{U}_N \mathcal{S}_f \mathbf{U}_N + \sum_{k=i}^{N-1} \mathbf{U}_k^T \mathcal{Q}_k \mathbf{U}_K + \boldsymbol{\alpha}_k^T \mathcal{R}_k \boldsymbol{\alpha}_k$$

where $\mathcal{S}$, $\mathcal{Q}$ and $\mathcal{R}$ are weight matrices for the final state, intermediate state and control inputs respectively. The system is also subject to the constraint of a linear flow equation given by

$$\mathbf{U}_{k+1} = \mathcal{A}_k \mathbf{U}_k + \mathcal{B}_k \boldsymbol{\alpha}_k$$

where $\mathcal{A}$ and $\mathcal{B}$ are the state and control input matrices. The Hamiltonian is then given as

$$H_k = \frac{1}{2}\left[\mathbf{U}_k^T \mathcal{Q}_k \mathbf{U}_K + \boldsymbol{\alpha}_k^T \mathcal{R}_k \boldsymbol{\alpha}_k\right] + \boldsymbol{\Lambda}_{k+1}^T \left[\mathcal{A}_k \mathbf{U}_k + \mathcal{B}_k \boldsymbol{\alpha}_k\right]$$

According to optimal control theory [54], the necessary conditions for minimization of the functional are

$$\begin{aligned}
\mathbf{U}_{k+1} = \quad & \frac{\partial H_k}{\partial \mathbf{\Lambda}_{k+1}} && = \mathcal{A}_k \mathbf{U}_k + \mathcal{B}_k \boldsymbol{\alpha}_k && \text{State equation} \\
\mathbf{\Lambda}_k = \quad & -\frac{\partial H_k}{\partial \mathbf{U}_k} && = \mathcal{A}_k^T \mathbf{\Lambda}_{k+1} + \mathcal{Q}_k \mathbf{U}_k && \text{Costate equation} \\
0 = \quad & \frac{\partial H_k}{\partial \boldsymbol{\alpha}_k} && \Rightarrow \boldsymbol{\alpha}_k = -\mathcal{R}_k^{-1} \mathcal{B}_k^T \mathbf{\Lambda}_{k+1} && \text{Stationarity condition}
\end{aligned}$$

Rearranging the equations slightly, the state and costate equations can be coupled and written as a single unforced system

$$\begin{bmatrix} \mathbf{U}_{k+1} \\ \mathbf{\Lambda}_k \end{bmatrix} = \begin{bmatrix} \mathcal{A} & -\mathcal{B}\mathcal{R}^{-1}\mathcal{B}^T \\ \mathcal{Q} & \mathcal{A}^T \end{bmatrix} \begin{bmatrix} \mathbf{U}_k \\ \mathbf{\Lambda}_{k+1} \end{bmatrix}$$

This system is difficult to solve because the state equation is integrated forwards and the costate equation is integrated backwards in time. The most popular method to solve this system of equations is through the use of the Riccati equation. Implementation of this problem on a 2-D channel flow will be given in detail in chapter 6.

2.4 System-theoretic approach to proper orthogonal decomposition

We follow a similar notation as in [37] for the derivation of the POD model: $\mathcal{L}^2(\Omega)$ denotes the collection of all square-integrable functions defined on a flow region $\Omega \subset \mathbb{R}^2$ and the associated norm is denoted by $\|\cdot\|_0$; also,

$$\mathcal{H}^1(\Omega) = \{v \in \mathcal{L}^2(\Omega) : \frac{\partial v}{\partial x_i} \in \mathcal{L}^2(\Omega) \text{ for } i = 1,\ 2\}$$

and the norm on it is $\|\cdot\|_1$. In addition, $\mathcal{L}^2(0,T;\mathcal{H}^1)$ denotes the space of all measurable functions $f:(0,T)\rightarrow\mathcal{H}^1$ such that,

$$\|f\|_{\mathcal{L}^2(0,T;\mathcal{H}^1)}=\left(\int_0^T\|f\|_1^2dt\right)^{1/2}<\infty$$

Bold-face symbols denote the vector valued counterparts of the above-mentioned spaces. The inner products are denoted by $<\cdot,\cdot>_\Gamma$ where Γ denotes the boundary of Ω.

In this section of the dissertation, we are interested in a reduced-order model which reduces the computational burden and as such we consider a set of basis functions $\mathbf{\Phi}_i,\ (i=1,\ldots,N)$ where N is much less than the number of grid points used in an equivalent finite difference approximation or the number of functions used in a finite element method. The approximation to the state variables $\widetilde{\mathbf{U}}\in\mathcal{V}\triangleq\text{span}\{\mathbf{\Phi}_1,\ldots,\mathbf{\Phi}_N\}$ can then be written as a linear combination $\widetilde{\mathbf{U}}=\sum_{i=1}^N a_i\mathbf{\Phi}_i$ where a_i are weighting coefficients corresponding to the basis functions $\mathbf{\Phi}_i$. Selection methods for the weights will be discussed later in this section.

The essential approach to POD following the method of snapshots described here is a summary of the work by Ravindran[37]. The basic ideas and the formulation are reproduced for the sake of completeness. The POD method seeks to identify a structure in a random vector field. The objective is to seek a function $\mathbf{\Phi}$ that has a structure typical of the members of an ensemble of random vector fields, $\mathbf{U}^{(i)}$. To resolve this problem, one would project the field ensemble on $\mathbf{\Phi}$ and find a $\mathbf{\Phi}$ that maximizes $<\mathbf{U}^{(i)},\mathbf{\Phi}>$ while ensuring that the amplitude effects are removed through normalization ($\mathbf{\Phi}$ is being made parallel to the ensemble). The solution $\mathbf{\Phi}$ is sought

from the space of functions for which the inner-product exists i.e. $\mathbf{\Phi}$ must be $\mathcal{L}^2(\Omega)$. To include the statistics, we maximize

$$\frac{< \mathbf{U}^{(i)}, \mathbf{\Phi} >}{\sqrt{< \mathbf{\Phi}, \mathbf{\Phi} >}}$$

in some average sense. Also, since the maximization only needs to consider the magnitude and not the sign, we would consider the mean of the square of the above.

Now consider a set of N snapshots that form the ensemble set:

$$\mathcal{S} = \{\mathbf{U}^{(i)} : 1 \leq i \leq N\}$$

In the case of control of fluid flows or any DPS, these snapshots are the solutions at N different time steps, t_i. The objective in that case is to **seek a function $\mathbf{\Phi} \in \mathcal{L}^2(\Omega)$ that maximizes**

$$\frac{1}{N} \sum_{i=1}^{N} \frac{| < \mathbf{U}^{(i)}, \mathbf{\Phi} > |^2}{< \mathbf{\Phi}, \mathbf{\Phi} >} \tag{2.5}$$

It has been shown in [35] that when the number of degrees of freedom required to describe $\mathbf{U}^{(i)}$ is larger than the number of snapshots N, it is efficient to express the basis functions as a linear combination of the snapshots. Thus, it is proposed that $\mathbf{\Phi}$ has a special form in terms of the original data as

$$\mathbf{\Phi} = \sum_{i=1}^{N} w_i \mathbf{U}^{(i)} \tag{2.6}$$

where w_i need to be determined such that $\mathbf{\Phi}$ maximizes Eq.(2.5). This maximization problem can be cast in an equivalent eigenvalue problem. To this effect, define

$$\mathbf{K\Phi} = \frac{1}{N}\sum_{i=1}^{N}\int_{\Omega}\mathbf{U}^{(i)}(x)\mathbf{U}^{(i)}(x')\mathbf{\Phi}(x')dx' \tag{2.7}$$

Then

$$\begin{aligned} < \mathbf{K\Phi}, \mathbf{\Phi} > &= \frac{1}{N}\sum_{i=1}^{N}\int_{\Omega}\int_{\Omega}\mathbf{U}^{(i)}(x)\mathbf{\Phi}(x)dx\mathbf{U}^{(i)}(x')\mathbf{\Phi}(x')dxdx' \\ &= \frac{1}{N}\sum_{i=1}^{N}| < \mathbf{U}^{(i)}, \mathbf{\Phi} > |^2 \end{aligned}$$

Moreover,

$$\frac{< \mathbf{K\Phi}, \mathbf{\Phi} >}{< \mathbf{\Phi}, \mathbf{\Phi} >} = \frac{\frac{1}{N}\sum_{i=1}^{N}| < \mathbf{U}^{(i)}, \mathbf{\Phi} > |^2}{< \mathbf{\Phi}, \mathbf{\Phi} >} = \lambda$$

The maximization of the above can be performed using straightforward calculus of variations, i.e., assuming $\mathbf{\Phi}^*$ as the function that maximizes λ and using small perturbations to expand, $\mathbf{\Phi} = \mathbf{\Phi}^* + \epsilon\mathbf{\Phi}'$, one can find the necessary conditions and show that

$$< \mathbf{K\Phi}^*, \mathbf{\Phi}' > \ = \lambda < \mathbf{\Phi}^*, \mathbf{\Phi}' >$$

It is hence clear that the maximization of Eq.(2.5) is equivalent to finding the solution of the eigenvalue problem

$$\mathbf{K\Phi}^* = \lambda\mathbf{\Phi}^* \tag{2.8}$$

Therefore, if Eqs.(2.6) and (2.7) are used in Eq.(2.8), we have

$$\mathbf{CW} = \lambda \mathbf{W} \tag{2.9}$$

where

$$\mathbf{C}_{ij} = \frac{1}{N} \int_{\Omega} \mathbf{U}^{(i)}(x)\mathbf{U}^{(j)}(x)dx, \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

and where $\mathbf{C}$ is a spatial correlation matrix which is nonnegative and Hermitian such that Eq.(2.9) can be decomposed into a complete set of orthogonal eigenvectors

$$\mathbf{W}_1 = \begin{bmatrix} w_1^1 \\ w_2^1 \\ \vdots \\ w_N^1 \end{bmatrix}, \quad \mathbf{W}_2 = \begin{bmatrix} w_1^2 \\ w_2^2 \\ \vdots \\ w_N^2 \end{bmatrix}, \quad \ldots\ldots, \quad \mathbf{W}_N = \begin{bmatrix} w_1^N \\ w_2^N \\ \vdots \\ w_N^N \end{bmatrix}$$

along with a set of eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq 0$. We can then obtain the solutions to Eq.(2.5) as

$$\Phi_1 = \sum_{i=1}^{N} w_i^1 \mathbf{U}^{(i)}, \quad \Phi_2 = \sum_{i=1}^{N} w_i^2 \mathbf{U}^{(i)}, \quad \ldots, \quad \Phi_N = \sum_{i=1}^{N} w_i^N \mathbf{U}^{(i)} \tag{2.10}$$

The functions are then normalized by requiring,

$$< \mathbf{W}_j, \mathbf{W}_j > \; = \frac{1}{N\lambda_j}$$

This now provides us with a set of orthonormal basis functions $\{\mathbf{\Phi}_1,\ \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_N\}$ i.e.,

$$< \mathbf{\Phi}_l, \mathbf{\Phi}_m > \ = \begin{cases} 1 & l = m \\ 0 & l \neq m \end{cases}$$

The POD subspace is then essentially defined as, $\mathcal{V}^{\text{POD}} = \text{span}\{\mathbf{\Phi}_1,\ \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_N\}$. The energy of a given data set associated with the corresponding mode $\mathbf{\Phi}_i$ can be quantified based on Eq. 2.7.

$$\lambda_i = \frac{1}{N} \sum_{j=1}^{N} < \mathbf{\Phi}_i, \mathbf{U}^{(j)} >^2$$

Ravindran [37] then also shows that the POD subspace calculated above is optimal in the sense that the approximation of the snapshots,

$$\mathbf{U}^{(l)} = \sum_{i=1}^{N_k} a_i^l \mathbf{\Phi}_i, \quad a_i^l =< \mathbf{\Phi}_i, \mathbf{U}^{(l)} >$$

maximizes the captured energy

$$\begin{aligned} E &= \frac{1}{N} \sum_{i=1}^{N} < \mathbf{U}^{(i)}, \mathbf{U}^{(i)} > \\ &= \sum_{i=1}^{N_k} \lambda_i \quad \text{for all } N_k < N \end{aligned}$$

To accurately capture the underlying dynamics of the system, N has to be large. In such a case, using a Galerkin procedure, one can obtain a high fidelity model for large N. In many cases the majority of the energy capture is contained in the first

few modes. If this is the case then the number of functions required to accurately describe the system may be significantly less than N. The set can be truncated to N_k to obtain an optimal set.

2.4.1 Application of POD to Euler equations

For the Euler equations (2-D as well as 1-D), we apply the procedure discussed in the previous sections for each of the primitive variables, $(\rho,\ u,\ v,\ p)$ that would in turn be used to compute the conservative state vector $(\rho,\ \rho u,\ \rho v,\ \rho E)$ if need be. Having obtained the basis functions (modes) within acceptable accuracy, we expand the flow solution about an arbitrary geometry. To illustrate, the flow fields for a double ramp 2-D inlet case would be expressed as

$$\begin{aligned}
\rho(x,y,\boldsymbol{\alpha}) &= \sum_{i=1}^{N_k} a_{\rho i}(\boldsymbol{\alpha})\boldsymbol{\Phi}_i^{(\rho)} \\
u(x,y,\boldsymbol{\alpha}) &= \sum_{i=1}^{N_k} a_{ui}(\boldsymbol{\alpha})\boldsymbol{\Phi}_i^{(u)} \\
v(x,y,\boldsymbol{\alpha}) &= \sum_{i=1}^{N_k} a_{vi}(\boldsymbol{\alpha})\boldsymbol{\Phi}_i^{(v)} \\
p(x,y,\boldsymbol{\alpha}) &= \sum_{i=1}^{N_k} a_{pi}(\boldsymbol{\alpha})\boldsymbol{\Phi}_i^{(p)}
\end{aligned} \tag{2.11}$$

where $\boldsymbol{\alpha}$ is the vector of control inputs.

To generate the basis functions for these cases, we first convert the flow solutions (snapshot solutions) for a judiciously chosen grid of design parameters into groups of vectors. If the number of nodes on the computational grid are $[I_{\mathrm{MAX}},\ \ J_{\mathrm{MAX}}]$, then $\mathbf{U}^{(i)}$ is a $[I_{\mathrm{MAX}} * J_{\mathrm{MAX}},\ \ 1,\ \ 4]$ matrix. The last index refers to number of primitive variables i.e., $sizeof(\rho,\ u,\ v,\ p)$. Thus, for N_p design parameter combinations, the

ensemble of snapshots will be a $[I_{\mathrm{MAX}} * J_{\mathrm{MAX}}, \quad N_p, \quad 4]$ matrix.

The snapshots are then obtained following the approach outlined in the previous section. Since the computation of the basis functions is done separately for each of the primitive variables, all calculations shown below are repeated for each of the primitive variables. To obtain the spatial correlation matrix,

$$\mathbf{C}_{ij} = \frac{1}{N} \int_{\Omega} \mathbf{U}^{(i)}(x)\mathbf{U}^{(j)}(x)dx, \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$$

The above is approximated for the discrete computational domain as follows [45],

$$\mathbf{C}_{ij} = \frac{1}{N} \sum_{k=1}^{I_{\mathrm{MAX}} * J_{\mathrm{MAX}}} \mathbf{U}^{(i)}(k)\mathbf{U}^{(j)}(k)\Delta A_k$$

where ΔA_k is corresponding cell area. This evaluation poses a problem in situations where unstructured meshes would be utilized and when the domain of the i^{th} and j^{th} snapshots are different. We use a common domain for all snapshots thereby avoiding this problem. This requires us to apply changes to the boundary conditions to account for the change in the boundary [55].

Determination of the weights for the reduced order model (ROM)

Having obtained the basis functions as outlined earlier, we can now seek to obtain the weights $a_{\rho i}$ etc., Eq.(2.11). The most common methods used to calculate the weights are projection based methods, Galerkin projection being the most popular among them. In these cases, the governing partial differential equations (PDE) are

recast into a system of ordinary differential equations (ODE). While the Galerkin projection is one of the most common methods, it is generally limited to incompressible flows due to its sensitivity to errors in boundary conditions and the need for simple, smooth geometries [40]. The Galerkin method can be applied to the compressible case if these issues are addressed carefully and the projection is of minimal order (small number of basis functions).

If the system is well known and well behaved, then projection methods can work well although they do get computationally expensive for more complex systems. However, for purely data driven systems and reduced order modelling of such, the projection methods would no longer be possible as the governing PDEs are unavailable. This necessitates that the weights be determined in a different manner. Here it is considered as a system identification problem and approached as such. Since the actual snapshots (solutions) are available at discrete design values, the following approach to generating the weights is proposed. The weights are derived by minimizing the following objective function for each snapshot condition

$$J_{POD} = \sum_{N_k} \|\mathbf{U}^{(i)} - \widetilde{\mathbf{U}}^{(i)}\|^2 \tag{2.12}$$

where the approximation $\widetilde{\mathbf{U}}^{(i)}$ to the desired data $\mathbf{U}^{(i)}$ is written in terms of the orthonormal basis $\mathbf{\Phi}_i(x),\ i = 1, 2, \ldots, N_k$ as

$$\mathbf{U}^{(i)}(x, \alpha_1, \ldots) = \sum_{i=1}^{N_k} a_i(\alpha_1, \ldots)\mathbf{\Phi}_i(x) \tag{2.13}$$

where, again the a_i's are the weighting coefficients of the i^{th} mode in the function expansion. This defines the solution for the given set of basis most closely resembling

the snapshot data. Given a set of basis functions and the generated snapshot data, the cost of the reduced order model approximation is simply the cost of the optimization of weighting coefficients as given in Eq.(2.12).

2.4.2 Application of POD to Burgers equation

Application of POD model reduction to the Burgers equation is slightly different than for the Euler equations. Since the Burgers equation describes the evolution of a single variable, we have a simplification in that there is no need to calculate basis functions and weights for four different flow variables. However, the *evolution* of that variable adds a small complication in that we are considering the change of that flow variable with time. In the previous section we assumed the flow solutions governed by the Euler equations were in steady state. To illustrate the difference, the flow field for the 2-D channel flow would be expressed as

$$U(x, y, t, \boldsymbol{\alpha}(t)) \quad = \quad \sum_{i=1}^{N_k} a(\boldsymbol{\alpha}(t), t)\boldsymbol{\Phi}_i \tag{2.14}$$

where a is now a function of the control input and time.

In order to capture the dynamics of the system, the snapshots are generated using random control inputs of varying strength and at various times in the flow progression. Again, if the number of nodes on the computational grid are $[I_{\mathrm{MAX}}, \quad J_{\mathrm{MAX}}]$, then a single snapshot $\mathbf{U}^{(i)}$ is a $[I_{\mathrm{MAX}} * J_{\mathrm{MAX}}, \quad 1]$ column vector. For N_p design parameter and time combinations, the ensemble of snapshots will be a $[I_{\mathrm{MAX}} * J_{\mathrm{MAX}}, \quad N_p]$ matrix.

Determination of the weights for the ROM

Due to the temporal nature of the Burgers equation control problem, the previous method of determining the weights would not be as beneficial. In order to structure the final POD model for use in a feedback control routine we use the traditional Galerkin projection techniques. We will follow the methodology given in [56] for the weak formulation and projection of Burgers equation onto the POD subspace. The weak formulation allows for the extraction of the boundary condition information prior to the projection and results in an explicit expression for the boundary control in the reduced model equations. This methodology is presented under the appropriate case study as it depends on a full description of the specific problem.

CHAPTER 3

QUASI-1-D DUCT SHAPE OPTIMIZATION

3.1 Quasi-1-D duct flow

The "quasi-1-D" term is used because we neglect non-streamwise velocity components and assume an axisymmetric shape (see Fig. 3.1). This is a valid assumption if the cross-section of the duct is smooth and does not change dramatically. For most duct flow problems the quasi-1-D duct is a reasonable approximation of the full 3-D flow. In the following sections, we first summarize the governing equations for the quasi-1-D duct and specify the boundary conditions. Then the derivation of the adjoint variables is presented followed by the results of the optimization involving the adjoint variables. Finally the reduced order model derivation, the formal optimization procedure and the results utilizing the POD model are presented.

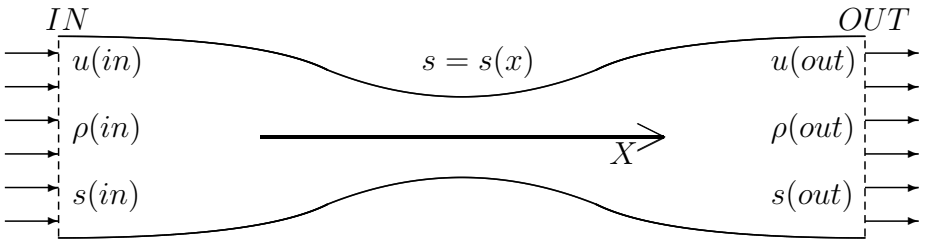


Figure 3.1. 1-D duct schematic.

3.1.1 Governing equations for quasi-1-D duct flow

A full derivation of the quasi-1-D duct equations is given in appendix A. The equations are written in conservative form as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial (s\mathbf{F})}{\partial x} - \mathbf{P}\frac{ds}{dx} = 0 \tag{3.1}$$

where $s(x)$ represents the duct shape (see Fig. 3.1). The state variables, flux and pressure terms are given as follows

$$\mathbf{U} = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u(E + p/\rho) \end{bmatrix}, \qquad \mathbf{P} = \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix},$$

3.1.2 Boundary conditions for the quasi-1-D duct problem

The boundary conditions of any flow problem governed by the Euler equations are set by the characteristics of the flow. The characteristics are the eigenvalues of the Jacobian matrix $d\mathbf{F}/d\mathbf{U}$, that can be analytically computed to be u, $u - c$, and $u + c$ where c is the speed of sound. The number of necessary conditions to be specified at a boundary is set by the number of characteristics that point into the system from that boundary.

For instance, at the IN-flow boundary if the flow is subsonic, there would be two characteristics pointing into the domain (u and $u + c$) and one pointing out ($u - c$). Figure 3.2 demonstrates the characteristics for supersonic and subsonic flow conditions. In this paper, we study the supersonic case with no shocks present. The resulting boundary conditions require that three IN-flow conditions need to be specified and all OUT-flow conditions are left unspecified. The inlet conditions are

fixed at a given total enthalpy of $H = E + p/\rho = 4$, total pressure $p_o = 2$, and Mach number $M = 3$.

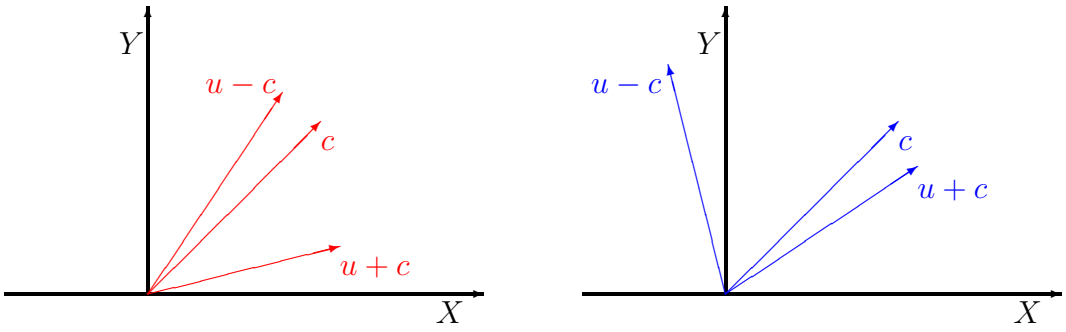


Figure 3.2. Flow characteristics for (a) supersonic and (b) subsonic flows.

3.2 Shape optimization using the adjoint method based gradients

The quasi 1-D duct is used in this research as a verification case where the methods of the adjoint derivation can be tested and validated. This work focuses on the continuous adjoint approach where the adjoint variables are derived through the Lagrange multiplier method.

3.2.1 Adjoint derivation for quasi-1-D duct flow

The Euler equations for a 1-D duct are given in Eq.(3.1) and are restated here for steady flow

$$\boldsymbol{R}(\mathbf{U}, s) = \frac{d}{dx}(s\mathbf{F}) - \boldsymbol{P}\frac{ds}{dx} = 0 \tag{3.2}$$

again $s(x)$ represents the shape of the duct. The cost function for the optimization is chosen such that the pressure p along the duct is to approach some desired pressure distribution p^*

$$J = \frac{1}{2}\int_{-1}^{1}(p-p^*)^2 dx$$

The cost function is augmented with the constraint that the flow equations must be satisfied. The Lagrange multiplier method for adding constraints is used to facilitate the derivation of the adjoint equations resulting in the Hamiltonian given as

$$H = \frac{1}{2}\int_{-1}^{1}(p-p^*)^2 dx - \int_{-1}^{1}\mathbf{\Lambda}^T \boldsymbol{R} dx$$

The adjoint equations are derived by first taking the variation of the cost function with respect to perturbations in the flow variables $\mathbf{U}$ and the design parameter s. We denote by $\tilde{(\cdot)}$ a variation in the quantity $(\cdot)$.

$$\begin{aligned} I = \delta H &= \int_{-1}^{1}(p-p^*)\frac{\partial p}{\partial \mathbf{U}}^T \widetilde{\mathbf{U}} dx \\ &\quad - \int_{-1}^{1}\mathbf{\Lambda}^T\left[\frac{d}{dx}\left(s\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\widetilde{\mathbf{U}}\right) - \frac{ds}{dx}\frac{\partial \boldsymbol{P}}{\partial \mathbf{U}}\widetilde{\mathbf{U}} + \frac{d}{dx}\left(\mathbf{F}\tilde{s}\right) - \boldsymbol{P}\frac{d\tilde{s}}{dx}\right] dx \\ &= \int_{-1}^{1}(p-p^*)\frac{\partial p}{\partial \mathbf{U}}^T \widetilde{\mathbf{U}} dx - \int_{-1}^{1}\mathbf{\Lambda}^T\left[\left(\frac{d}{dx}\left(s\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right) - \frac{ds}{dx}\frac{\partial \boldsymbol{P}}{\partial \mathbf{U}}\right)\widetilde{\mathbf{U}}\right] dx \\ &\quad + \int_{-1}^{1}\mathbf{\Lambda}^T\left[\frac{d}{dx}\left(\mathbf{F}\tilde{s}\right) - \boldsymbol{P}\frac{d\tilde{s}}{dx}\right] dx \end{aligned}$$

Applying integration by parts and rearranging the middle term, the variation of the cost function becomes

$$\begin{aligned} I &= \int_{-1}^{1}(p-p^*)\frac{\partial p}{\partial \mathbf{U}}\widetilde{\mathbf{U}}dx - \int_{-1}^{1}\left(-s\left[\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right]^T\frac{d\mathbf{\Lambda}}{dx} - \frac{ds}{dx}\left[\frac{\partial \boldsymbol{P}}{\partial \mathbf{U}}\right]^T\mathbf{\Lambda}\right)^T\widetilde{\mathbf{U}}dx \\ &\quad + \int_{-1}^{1}\mathbf{\Lambda}^T\left[\frac{d}{dx}(\mathbf{F}\tilde{s}) - \boldsymbol{P}\frac{d\tilde{s}}{dx}\right]dx - \mathbf{\Lambda}^T\left(s\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\widetilde{\mathbf{U}}\right)\Big|_{-1}^{1} \end{aligned}$$

By setting the terms containing $\widetilde{\mathbf{U}}$ to zero, we can remove any dependence of the cost function on variations in the flow and define the adjoint equation with its corresponding boundary conditions. The adjoint equations for the Euler equations and the given cost function are given in Eq.(3.3) and the boundary conditions are given in Eq.(3.4):

$$\frac{\partial \mathbf{\Lambda}}{\partial t} - \left[\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right]^T\frac{\partial \mathbf{\Lambda}}{\partial x} - \frac{1}{s}\frac{\partial s}{\partial x}\left[\frac{\partial \boldsymbol{P}}{\partial \mathbf{U}}\right]^T\mathbf{\Lambda} - \frac{1}{s}(p-p^*)\left(\frac{\partial p}{\partial \mathbf{U}}\right)^T = 0 \tag{3.3}$$

$$\mathbf{\Lambda}^T\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\widetilde{\mathbf{U}}\Big|_{-1}^{1} = 0 \tag{3.4}$$

where the Lagrange multiplier variable $\mathbf{\Lambda}$ is a vector of the adjoint variables. Recall $\widetilde{\mathbf{U}}$ is a small perturbation in $\mathbf{U}$

$$\widetilde{\mathbf{U}} = \begin{bmatrix} \tilde{\rho} \\ \widetilde{\rho u} \\ \widetilde{\rho E} \end{bmatrix}, \qquad \mathbf{\Lambda} = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{bmatrix}$$

For the case when the flow *entering* the duct is supersonic the boundary conditions for the flow equations are fully specified; therefore, $\widetilde{\mathbf{U}} = 0$. The resulting adjoint boundary variables Λ_1, Λ_2 and Λ_3 are all free. For the case where the flow *exiting* the duct is supersonic, the boundary conditions for the flow equations are free, therefore $\widetilde{\mathbf{U}}$ is arbitrary and the adjoint boundary variables are set to zero, i.e., $\Lambda_1, = \Lambda_2 = \Lambda_3 = 0$. Once the constraint equation for the adjoint variables is satisfied, the final form of the first variation of the cost function is given as,

$$\begin{aligned} I &= \int_{-1}^{1} \mathbf{\Lambda}^T \left[\frac{d}{dx}(\tilde{s}\mathbf{F}) - \boldsymbol{P}\frac{d\tilde{s}}{dx} \right] dx \\ &= \int_{-1}^{1} \mathbf{\Lambda}^T \left[\mathbf{F}\frac{d\tilde{s}}{dx} + \tilde{s}\frac{d\mathbf{F}}{dx} - \boldsymbol{P}\frac{d\tilde{s}}{dx} \right] dx \\ &= \int_{-1}^{1} \mathbf{\Lambda}^T \left[\tilde{s}\frac{d\mathbf{F}}{dx} + (\mathbf{F} - \boldsymbol{P})\frac{d\tilde{s}}{dx} \right] dx \end{aligned} \tag{3.5}$$

where $\tilde{s}$ is a small perturbation in s.

The gradient of the cost function is derived from the variation of the cost function. In the 1-D case with only one design parameter, this calculation is straight forward.

$$\begin{aligned} \Delta H &= \frac{\partial H}{\partial \alpha}\Delta\alpha \\ \Rightarrow \quad \left(H^{n+1} - H^n\right) &= \frac{\partial H}{\partial \alpha}\left(\alpha^{n+1} - \alpha^n\right) \\ \therefore \ \alpha^{n+1} &= \alpha_n + \frac{1}{(\partial H/\partial\alpha)|_{\alpha_n}}\left(H^{n+1} - H^n\right) \end{aligned}$$

The desired $H^{n+1} = 0$ and because the control variable is a scalar in this case, we can substitute in $I = (\partial H/\partial \alpha)\,\tilde{\alpha}$ and the update equation becomes

$$\therefore \quad \alpha^{n+1} \;=\; \alpha_n - \left[\frac{\partial H}{\partial \alpha}\right]^{-1} H^n$$

The numerical solution of the flow uses a modified one-dimensional advection upstream splitting method with low speed correction ($\mathrm{AUSM}^{+} - \mathrm{up}$) [57, 58] with adaptive limiters applied to the MUSCL scheme [59] to improve accuracy to second order. For the adjoint equations we utilized the Steger-Warming scheme in space and a 2^{nd} order Runge-Kutta scheme in time.

3.2.2 Results for the adjoint method based quasi 1-D duct optimization

The results presented here correlate well with previously published results [14]. In the present case, the duct shape is obtained via an inverse design process to match the pressure for some pre-specified shape. The validation of the adjoint variables is achieved based on the analytical solution developed for this specific problem [9]. For a sample case, the duct is parameterized [14] as follows for $\alpha \in [0, 1]$

$$s(x, \alpha) = \alpha x^2 - \sqrt{(0.8\alpha)}x + 1$$

where

$$ds(x, \alpha) = \left(x^2 - \frac{1}{2\sqrt{0.8\alpha}}x\right) d\alpha + \left(2\alpha x - \sqrt{0.8\alpha}\right) dx$$

The target pressure profile is generated for $\alpha = 0.8$ and the initial conditions are set to $\alpha = 0.6$. The flow is set to be supersonic with Mach 3 at the inlet. The flow through the duct remains supersonic and there are no shocks. The results of the optimization are shown in Fig. 3.3 where it can be clearly seen that the objective

function approaches zero as the design parameter α approaches the target profile at 0.8.

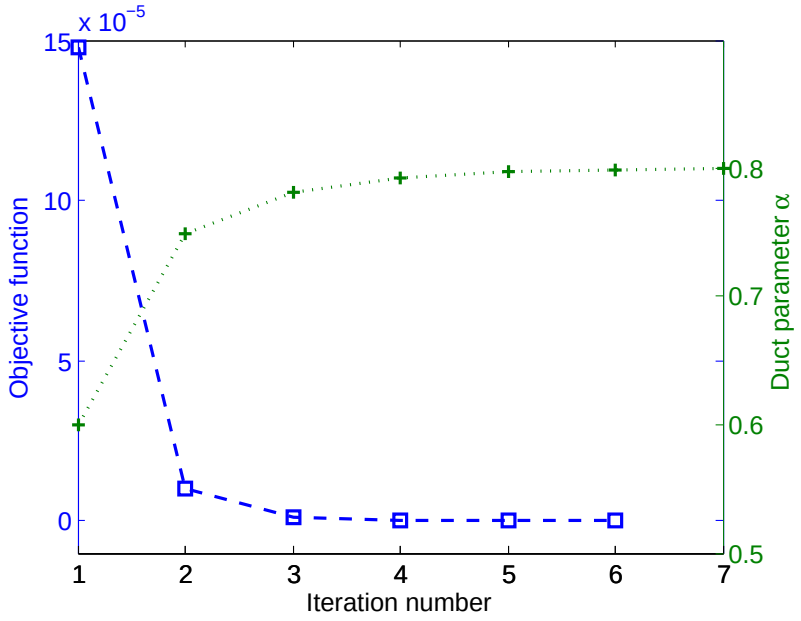


Figure 3.3. Optimization results for the quasi-1-D duct.

3.3 Shape optimization using POD based reduced order models

The flow equations for the 1-D duct are given in conservative form in Eq.(3.2) and are restated here

$$\boldsymbol{R}(\mathbf{U}, s) = \frac{d}{dx}(s\mathbf{F}) - \boldsymbol{P}\frac{ds}{dx} = 0 \tag{3.2}$$

where again $s(x)$ represent the shape of the duct and the state variables, flux and pressure terms are given as follows:

$$\mathbf{U} = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u(E + p/\rho) \end{bmatrix}, \qquad \mathbf{P} = \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix}$$

3.3.1 POD based reduced order model for the quasi 1-D duct flow

An inverse pressure matching optimization problem is solved using a desired pressure profile that corresponds to some optimal shape of the duct. The shape of the duct is parameterized through a parameter α and governed by the following equation.

$$s(x, \alpha) = \alpha\ x^2 - (\sqrt{0.8\alpha})\ x + 1, \qquad 0 < \alpha \leq 1 \tag{3.6}$$

Without loss of generality the optimal pressure profile is chosen corresponding to $\alpha = 0.8$

The following algorithm describes the process of determining the complete POD reduced order model. Let $\mathbf{U}^{(i)}(x, \alpha)$ be a steady state solution of the flow for a given α defining the duct shape from Eq.(3.6):

1. Generate a set of N snapshots $\mathbf{U}^{(i)}$ from steady state flow solutions of Eq.(3.2) for a random set of α's.
2. Compute the correlation matrix $\mathbf{C}$ from Eq.(2.9).
3. Solve the eigenvalue problem $\mathbf{CW} = \lambda\mathbf{W}$ to get a complete set of eigenvectors w_i.

4. Obtain the basis functions $\mathbf{\Phi}_i$ using Eq.(2.10).
5. Determine the corresponding weights in Eq.(2.13) by minimizing the cost function given in Eq.(2.12).
6. Determine a relationship for the weights as a function of (α_1, α_2).

In this case, the steady-state duct flow is computed for 12 discrete values of the control variable α ranging between 0.55 and 0.95. The eigenvalues of the correlation matrix computed in step 3 are shown in Fig. 3.4. Notice that the amount of information contained in each mode drops off quickly and only the first 2 basis functions are required to capture 99.99% of the energy. Figure 3.5 shows the results of step 4, the first two basis functions. The first mode has the same shape as the true response from the system and contains the majority of the information of the system. Steps 5 and 6 are presented in Fig 3.6. Here we see clearly that the weights and the design parameter α bear a simple linear relationship.

With the basis functions and weights computed the reduced order model is complete. To illustrate how well the reduced order model captures the original snapshots, we show a comparison of the POD model with the CFD results for some specific control (design) values. Fig. 3.7 shows this comparison for the two extreme values of α that were used to generate the snapshots, i.e., $\alpha = 0.55$ and $\alpha = 0.95$.

3.3.2 Optimization of the quasi-1-D duct shape

As mentioned earlier, the duct shape optimization problem is posed as an inverse optimization for pressure matching along the duct. The cost function is chosen such that the pressure p along the duct is to match some desired pressure distribution p^* subject to the flow equations given in Eq.(3.2) and corresponding boundary conditions

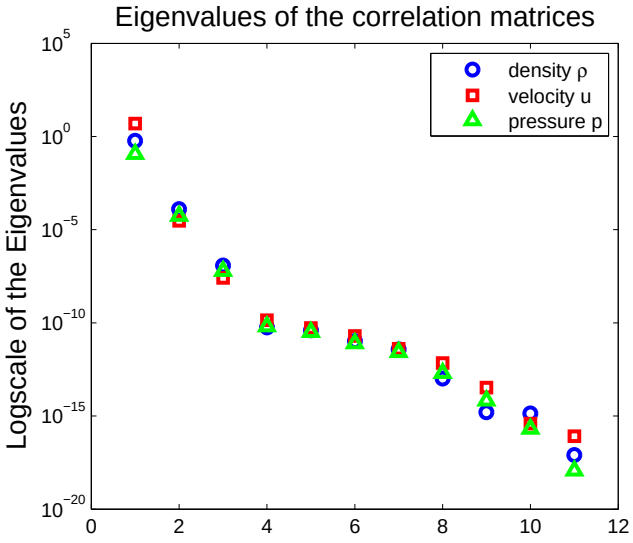


Figure 3.4. Eigenvalues for correlation matrices for each of the primitive variables.

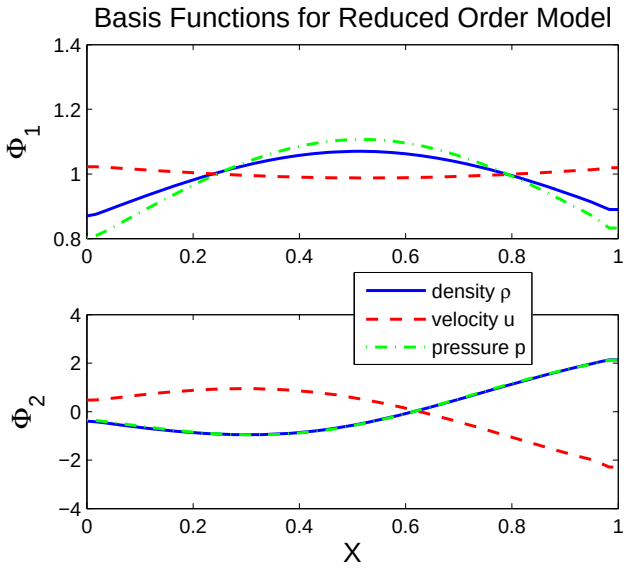


Figure 3.5. First two basis function generated from the eigenvalues of the correlation matrix.

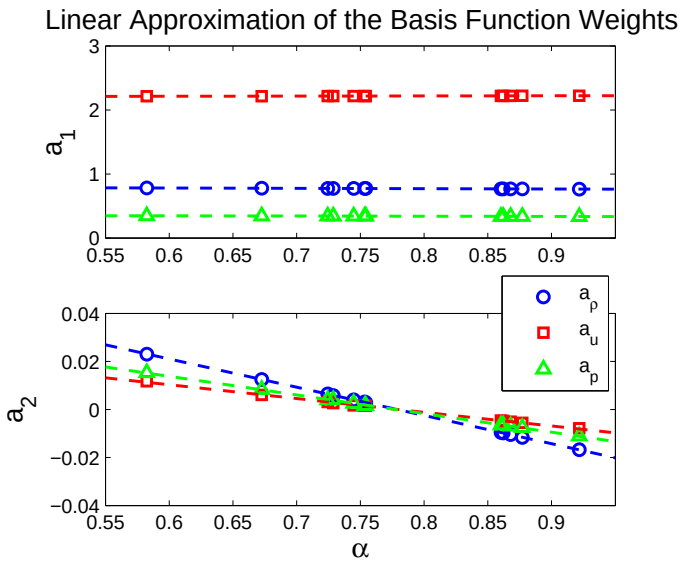


Figure 3.6. Weights corresponding to the basis functions as a function of the design parameter α.

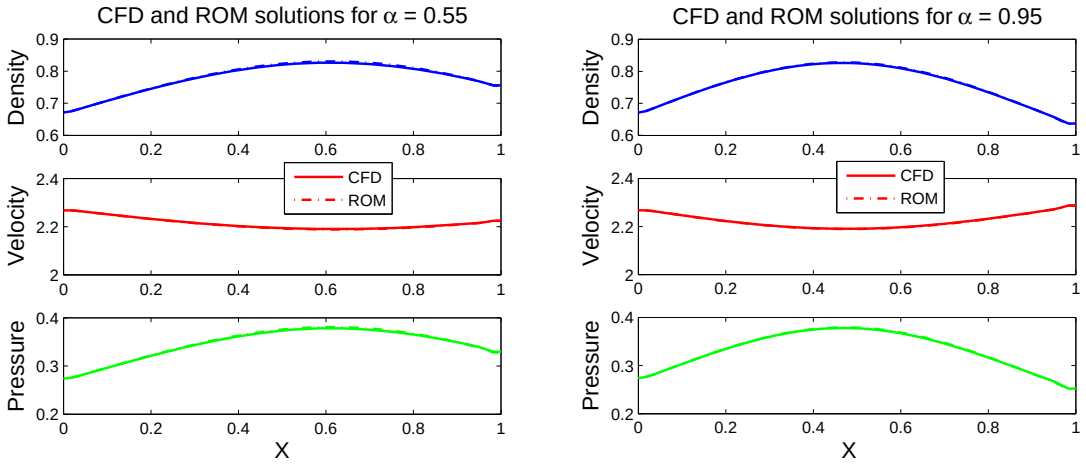


Figure 3.7. Comparing the results of the CFD solution with the final POD reduced order model for the extremum values of the control variable.

$$J = \frac{1}{2}\int_{-1}^{1} \|p - p^*\|^2 dx$$

With only one control variable the optimization routine used here is very straight forward. We use a Levenberg-Marquardt based approach (that combines the quasi-Newton and steepest descent) to optimize the pressure matching scenario with a target pressure profile. The target profile is generated using the same CFD solver as that used to generate the snapshot data. As mentioned before, we chose a target profile corresponding to $\alpha^* = 0.8$ and then began the optimization based on the POD model with an initial condition of $\alpha_0 = 0.2$. This initial condition is a good test of the POD model because is is well outside the range of values used to generate the snapshots from which the reduced order model was obtained. As shown in Fig. 3.8 the optimization was successful and was completed in only two iterations.

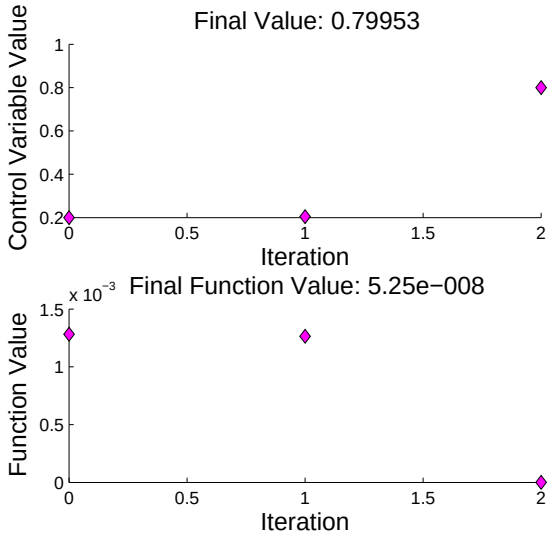


Figure 3.8. Results of the POD model optimization ($\alpha^* = 0.8$).

CHAPTER 4

2-D FULL INLET SHAPE OPTIMIZATION

4.1 2-D full double ramp inlet

The full inlet presented here is used as a 2-D verification case for the conventional optimization approach. The adjoint and POD optimization that follow are unique contributions of this dissertation. Figure 4.1 gives a sketch of the inlet under consideration. The inlet is assumed to be symmetric about a centerline and therefore only half of the inlet is simulated as the flow would also be symmetric. In the following sections, we will first review the details of the inlet and check the performance values against analytic values. Then we perform a conventional finite-difference optimization routine and again verify the results. The derivation of the adjoint variables will be presented next followed by the details of the numeric solver and the results. Finally, the reduced order model derivation and the optimization results of that model will be presented.

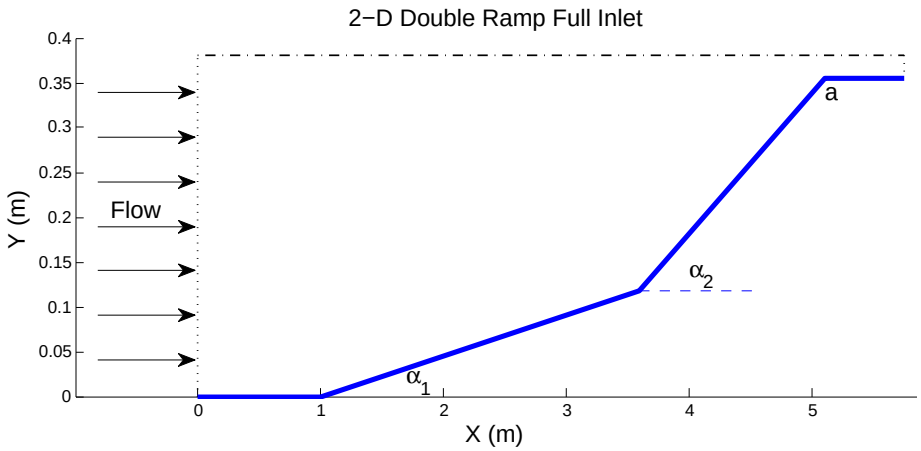


Figure 4.1. 2-D full inlet schematic.

4.1.1 Governing equations for the 2-D full inlet flow

The 2-D Euler equations are given in Eq.(2.2) and a full derivation is given in appendix B. The equations in steady state are given as follows:

$$\boldsymbol{R}(\mathbf{U}, \boldsymbol{\alpha}) = \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{4.1}$$

The state variables $\mathbf{U}$ and the flux vectors $\mathbf{F}$ and $\mathbf{G}$ are given as before

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \end{bmatrix}$$

where ρ, p, u and v are the density, pressure and x and y velocity components respectively. The geometry of the inlet is shown in Fig. 4.1

4.1.2 Boundary conditions for the 2-D full inlet

As in the 1-D case, the boundary conditions are determined by the characteristics of the flow, u, $u - c$, and $u + c$ where c is the speed of sound. The number of necessary conditions to be specified at a boundary is set by the number of characteristics that point into the system from that boundary. See Fig. 3.2 in the previous section for a graphic description of the characteristics.

In this case the IN-flow boundary is supersonic and has all characteristics pointing in and none pointing out. This requires that the entire IN-flow conditions be specified and the OUT-flow conditions are all left unspecified. The inlet conditions are fixed at $p = 1, \rho = 1$, and Mach number $M = 14$ such that $u = M/a$ and $v = 0$.

All wall boundaries and the centerline (upper boundary) are treated as reflection surfaces for the surface normal component of the velocity vector.

4.1.3 Numerical solution of the 2-D Euler equations

The Euler equations are solved using the advection upstream splitting method (AUSM) scheme. The method was originally proposed by Liou and Steffen[57] for typical compressible aerodynamic flows. It is a straightforward and efficient method to implement and offers accurate shock capturing capabilities. The grid utilized for both the flow and the adjoint calculations is an algebraic grid with a Thomas Middlecoff function for smoothing.

4.2 Generation of true optimized values for comparison — Finite difference gradient approach

In this section, we focus on first validating the inlet performance, then on generating the true optimal inlet ramp angles based on a finite difference based gradient approach as well as exhaustive parametric studies of the 2-D double ramp inlet. This specific inlet configuration was originally studied by Korte and Auslender[60] and later by Munipalli et.al,[61]. Both groups utilized this inlet in optimization problems and the extended the inlet description from two linear ramps to multiple cubic splines. In these earlier studies the inlet was designed to have a contraction ratio of 15 : 1 from the inlet to the throat radius. As well, the length of the inlet from the initiation of the first ramp to the termination of the second ramp (point **a** in Fig. 4.1) was 150 times the throat radius. In this dissertation the contraction ratio is the same, 15 : 1 however the length scale is slight longer at 162.

Validation of the inlet is most convenient in the shock canceled configuration. In this configuration the shocks generated by the ramps converge to a single point

at the center of the inlet and the reflection of those shocks impinge on the entry to the throat, this point is labeled as point **a** on the schematic given in Fig. 4.1. (See Fig. 4.2 for a comparison of the analytic and CFD results for the shock locations.) The shock canceled inlet makes the calculation of the analytic pressure loss much simpler as there is no need to consider interactions and multiple reflections of the shocks. The total pressure recovery ($PR = p_{o_{exit}}/p_{o_{in}}$) for the analytic case and the CFD solution are $PR = 0.3270$ and $PR = 0.3182$ respectively. These values along with the result shown in Fig. 4.2 are sufficient to validate the CFD solution.

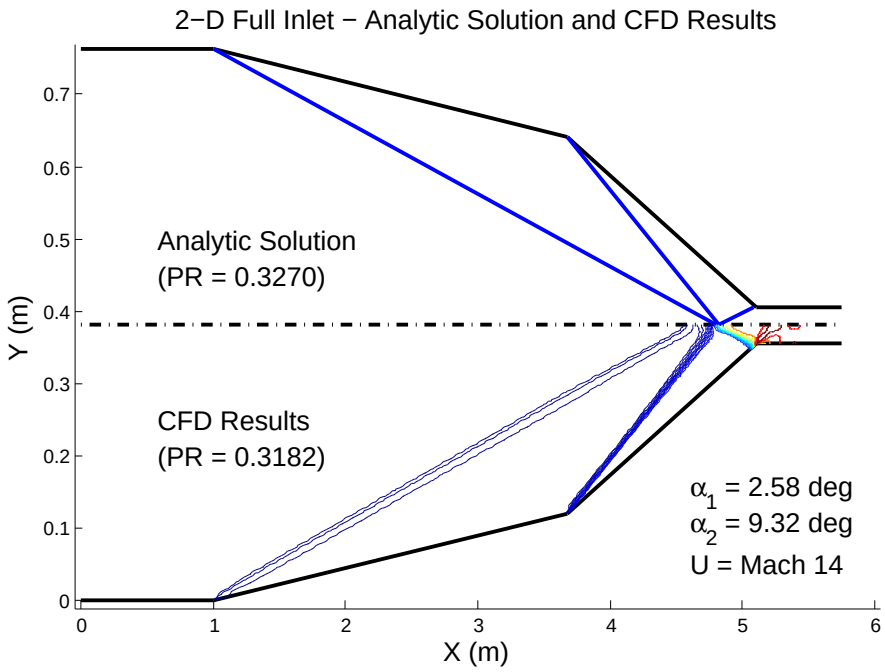


Figure 4.2. Analytic shock locations and CFD pressure contours for the shock canceled inlet.

The full inlet is optimized to find the minimum total pressure loss from the inflow boundary to the throat. This condition will minimize the energy loss through

the inlet and result in greater efficiency. The mass weighted total pressure loss cost function is given as

$$J_{PressureLoss} = \frac{\sum \rho_{i,j} u_{i,j} \Delta y_{i,j} \left(p_{o_{i,j}} - p_{o_\infty} \right)}{\dot{m} p_{o_\infty}} \tag{4.2}$$

A conventional optimization routine using a Levenberg–Marquardt scheme is implemented in MATLAB®[1] is implemented to optimize the inlet geometry. The initial condition is that of the shock cancelled inlet and the final optimized values were $\alpha_1 = 3.7129°$ and $\alpha_2 = 7.6724°$. The optimized inlet is given in Fig. 4.3. These values are verified against a parametric study in which the flow solution is generated and the cost function is evaluated for a range of α_1 and α_2 values to generate the contour plot given in Fig. 4.4. This contour plot clearly shows a large local minimum in the cost function at $\alpha_1 = 3.75°$ and $\alpha_2 = 7.6°$. These values correspond well with the finite difference results mentioned earlier. See table 4.1 for a summary of these results.

Table 4.1. Values of the objective function for the full inlet

Case	Angles (deg)		Mass Weighted Total Pressure Loss
	α_1	α_2	
Full inlet (parametric study)	3.75	7.60	0.1638
Full inlet (initial condition)	2.58	9.32	0.2452
Full inlet (optimization)	3.7129	7.6724	0.1639

[1]MATLAB/SIMULINK is the trademark of The MathWorks, Inc, Natick, MA,USA

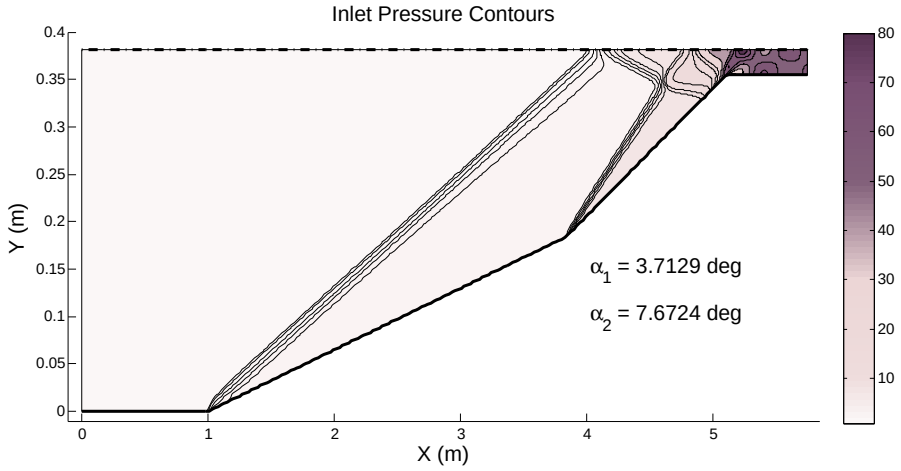


Figure 4.3. Pressure contours of the optimal configuration for the full inlet.

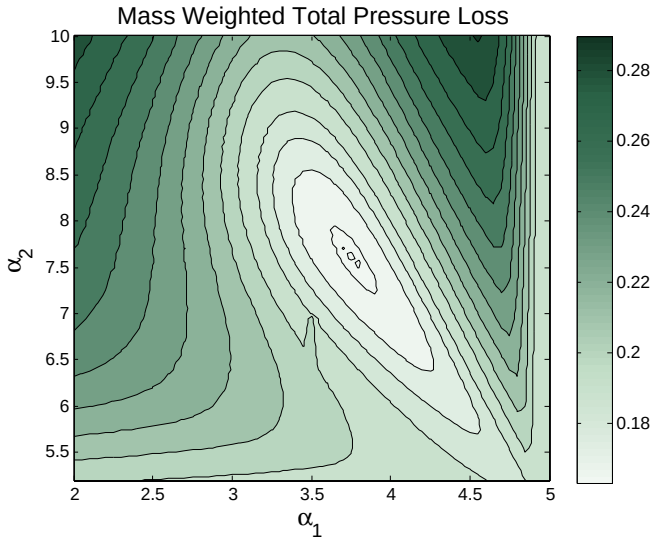


Figure 4.4. Cost function contour for various α_1 and α_2 for the double ramp full inlet.

4.3 Shape optimization using the adjoint method based gradients

4.3.1 Adjoint derivation for 2-D full inlet

We now explore the use of adjoints in the optimization of the shape parameters. While we recognize that the use of adjoints for optimization in this problem (only

two design variables) is an overkill, the development of the framework is necessitated by the fact that future work is related to using a continuously varying inlet geometry parameterized as several segments.

The cost function is the same as that presented earlier for the finite difference optimization, i.e. the mass weighted total pressure loss. The cost function given in Eq.(4.2) is written in integral form as follows

$$J(\mathbf{U}, \boldsymbol{\alpha}) = \oint_{exit} \frac{\rho u(p_o - p_{o_\infty})}{\dot{m} p_{o_\infty}} \cdot \boldsymbol{d}s$$

As in the 1-D case, the cost function is augmented with the constraint that the flow equations must be satisfied. Again, we use the Lagrange multiplier method to enforce that constraint and generate the Hamiltonian:

$$H(\mathbf{U}, \boldsymbol{\alpha}) = \oint_{exit} \frac{\rho u(p_o - p_{o_\infty})}{\dot{m} p_{o_\infty}} \cdot \boldsymbol{d}s + \int_{\Omega} \boldsymbol{\Lambda}^T \boldsymbol{R}(\mathbf{U}, \boldsymbol{\alpha}) d\Omega$$

Let $\mathbf{A} = \partial \mathbf{F}/\partial \mathbf{U}$ and $\mathbf{B} = \partial \mathbf{G}/\partial \mathbf{U}$. Then the first variation of the cost function with respect to perturbations in the flow variables $\mathbf{U}$ and the design parameters $\boldsymbol{\alpha}$ is given as

$$I = \delta H \quad = \quad \delta H_{\mathbf{U}} + \delta H_{\boldsymbol{\alpha}} + \delta H_{\boldsymbol{\Lambda}} \tag{4.3}$$

where

$$\begin{aligned}
\delta H_{\mathbf{U}} &= \frac{1}{\dot{m} p_{o_\infty}} \oint_{exit} \left(\frac{\partial(\rho u(p_o - p_{o_\infty}))}{\partial \mathbf{U}} \right)^T \widetilde{\mathbf{U}} ds + \int_\Omega \mathbf{\Lambda}^T \left[\frac{\partial}{\partial x} \left(\mathbf{A} \widetilde{\mathbf{U}} \right) + \frac{\partial}{\partial y} \left(\mathbf{B} \widetilde{\mathbf{U}} \right) \right] d\Omega \\
&= \frac{1}{\dot{m} p_{o_\infty}} \oint_{exit} \left(\frac{\partial(\rho u(p_o - p_{o_\infty}))}{\partial \mathbf{U}} \right)^T \widetilde{\mathbf{U}} ds + \oint_s \mathbf{\Lambda}^T \left[\mathbf{A} n_{sx} + \mathbf{B} n_{sy} \right] \widetilde{\mathbf{U}} ds \\
&\quad - \int_\Omega \left[\left(\frac{\partial \mathbf{\Lambda}}{\partial x} \right)^T \mathbf{A} + \left(\frac{\partial \mathbf{\Lambda}}{\partial y} \right)^T \mathbf{B} \right] \widetilde{\mathbf{U}} d\Omega
\end{aligned}$$

$$\delta H_{\boldsymbol{\alpha}} = \frac{1}{\dot{m} p_{o_\infty}} \oint_{exit} \left(\frac{\partial(\rho u(p_o - p_{o_\infty}))}{\partial \boldsymbol{\alpha}} \right)^T \tilde{\boldsymbol{\alpha}} \, ds + \int_\Omega \mathbf{\Lambda}^T \left[\frac{\partial}{\partial x} \left(\frac{\partial \mathbf{F}}{\partial \boldsymbol{\alpha}} \, \tilde{\boldsymbol{\alpha}} \right) + \frac{\partial}{\partial y} \left(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\alpha}} \, \tilde{\boldsymbol{\alpha}} \right) \right] d\Omega$$

$$\delta H_{\mathbf{\Lambda}} = \int_\Omega \tilde{\mathbf{\Lambda}}^T \left(\mathbf{A} \, \frac{\partial \mathbf{U}}{\partial x} + \mathbf{B} \, \frac{\partial \mathbf{U}}{\partial y} \right) d\Omega$$

where n_{sx} and n_{sy} are components of the surface normal vector $\mathbf{n_s}$. The last term $\delta H_{\mathbf{\Lambda}}$ is equal to zero because the steady-state flow condition is $\mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} = 0$.

By setting $\delta H_{\mathbf{U}} = 0$ we extract the adjoint form of the Euler equations and the boundary conditions. The adjoint equations and boundary conditions for the 2-D Euler equations and this cost function are given in Eqs.(4.4) and (4.5)

$$\mathbf{A}^T \left[\frac{\partial \mathbf{\Lambda}}{\partial x}\right] + \mathbf{B}^T \left[\frac{\partial \mathbf{\Lambda}}{\partial y}\right] = 0 \qquad \text{in } \Omega \tag{4.4}$$

$$\begin{aligned} &\frac{1}{\dot{m} p_{o_\infty}} \left[\frac{\partial}{\partial \mathbf{U}} \left(\rho u (p_o - p_{o_\infty})\right)\right] \Delta s \\ &+ \left[\frac{\partial \mathbf{F}}{\partial \mathbf{U}} n_{sx} + \frac{\partial G}{\partial \mathbf{U}} n_{sy}\right]^T \mathbf{\Lambda} = 0 \qquad \text{at the exit} \\ &\left[\frac{\partial \mathbf{F}}{\partial \mathbf{U}} n_{sx} + \frac{\partial \mathbf{G}}{\partial \mathbf{U}} n_{sy}\right]^T \mathbf{\Lambda} = 0 \qquad \text{on all other boundaries} \end{aligned} \tag{4.5}$$

The boundary conditions on the inlet, the ramp wall and the supersonic outflow planes are given as follows. The wall boundary conditions are obtained by applying the restriction that the flow perpendicular to the wall is zero, i.e., $un_{sx} + vn_{sy} = 0$.

$$\left.\begin{aligned} \Lambda_1 &= \text{free} \\ \Lambda_2 &= \text{free} \\ \Lambda_3 &= \text{free} \\ \Lambda_4 &= \text{free} \end{aligned}\right|_{inlet}, \quad \left.\begin{aligned} \Lambda_1 &= \text{free} \\ \Lambda_2 &= \text{free} \\ \Lambda_3 &= -\frac{n_{sx}}{n_{sy}} \Lambda_2 \\ \Lambda_4 &= \text{free} \end{aligned}\right|_{wall}, \quad \left.\begin{aligned} \Lambda_1 &= 0 \\ \Lambda_2 &= 0 \\ \Lambda_3 &= 0 \\ \Lambda_4 &= 0 \end{aligned}\right|_{outflow}$$

The exit plane of the inlet is a little more complex and involves the integral of the cost function. Since the flow conditions at the boundary are fully free and $\tilde{U}$ is arbitrary the adjoint conditions must be fully specified.

$$\left.\begin{aligned}
\Lambda_1 = \; & -\frac{\Delta y}{\dot{m}P_{o\infty}}\left[(\gamma-1)\rho E + \frac{(\gamma-2)}{2}\rho(3u^2+v^2) - p_{o_\infty}\right] \\
& -u(3-\gamma)\Lambda_2 - v\Lambda_3 - \left(\tfrac{(1-\gamma)}{2}(3u^2+v^2) + \gamma E\right)\Lambda_4 \\
\Lambda_2 = \; & -\frac{\Delta y}{\dot{m}P_{o\infty}}\rho u - \frac{\gamma}{(\gamma-1)}u\Lambda_4 \\
\Lambda_3 = \; & \frac{\Delta y}{\dot{m}P_{o\infty}}(2-\gamma)\rho v - (\gamma u + (1-\gamma)v)\Lambda_4 \\
\Lambda_4 = \; & -\frac{\Delta y}{\dot{m}p_{o_\infty}}\frac{\frac{2\gamma-5}{2(\gamma-1)}\rho u\mathbf{V}^2}{\frac{\gamma}{2}u\mathbf{V}^2 - \frac{1}{\gamma-1}u^3 - \gamma u E}
\end{aligned}\right|_{exit}$$

4.3.2 Numerical solution of the 2-D adjoint equations

The CFD algorithm used to solve the adjoint equations is given in full detail in appendix C. The solver uses a finite volume technique and is second-order accurate. With appropriate adjustments to the solution equation and the boundary conditions, the algorithm is adapted to solve the flow equations in order to validate the method and code. Figure 4.5 shows the flow equation residual versus the iteration number. With $\Delta t = 1\mathrm{E}-5$, the flow converges in $30,000$ iterations.

Figure 4.6 shows the pressure contours of the flow solver. The pressure contours demonstrate a good correlation with analytic results for both the shock locations and pressure changes across the shocks. In this case, the incoming flow was at Mach 14 and the ramp angles were set at 2.58° and 9.32° respectively.

The adjoint code is able to run at $\Delta t = 1\mathrm{E}-5$ and converges in $40,000$ iterations as can be seen in Fig. 4.7. Both solvers used a fixed time step and an Euler

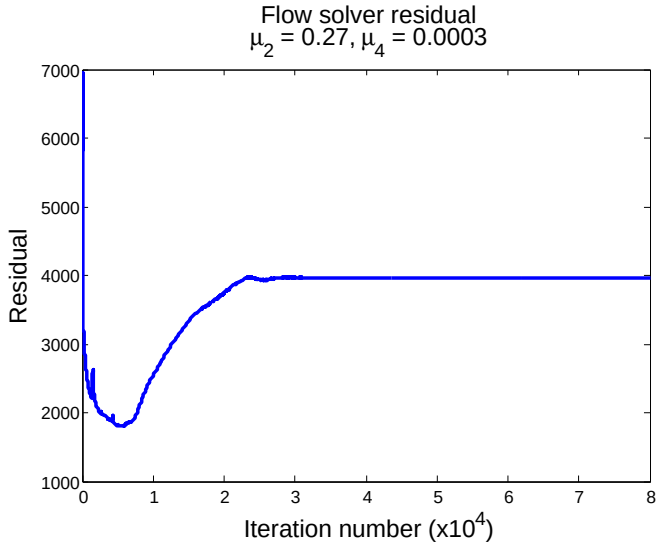


Figure 4.5. Residual values of the flow solver.

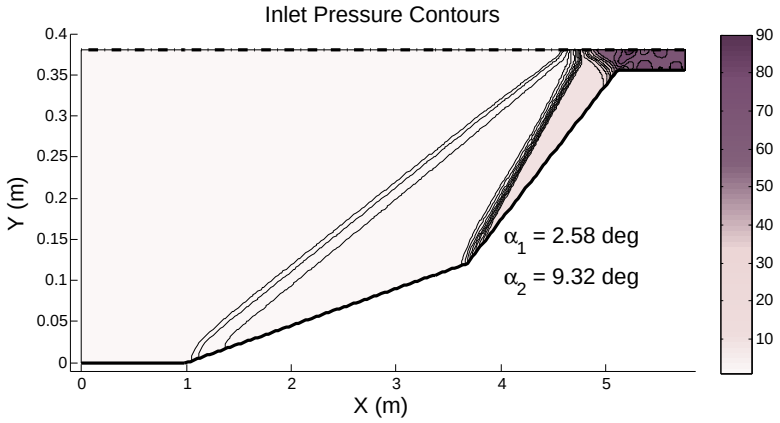


Figure 4.6. Pressure contours of the flow.

integration in time. Results could likely be improved with convergence accelerators such as those utilized in [14] and [11].

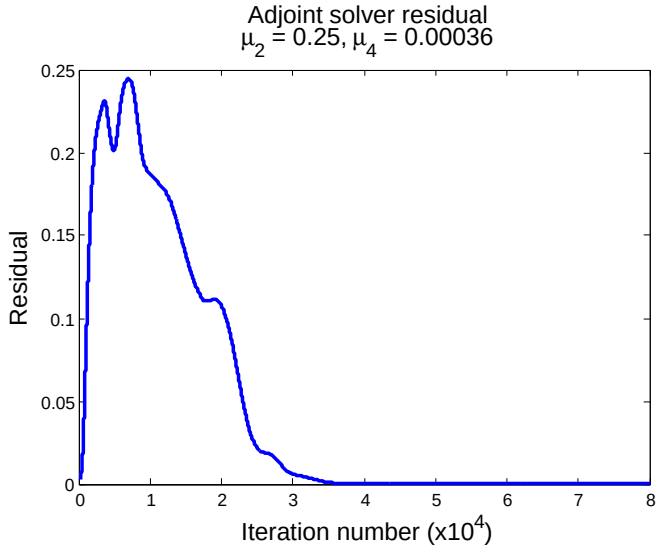


Figure 4.7. Residual values of the adjoint solver.

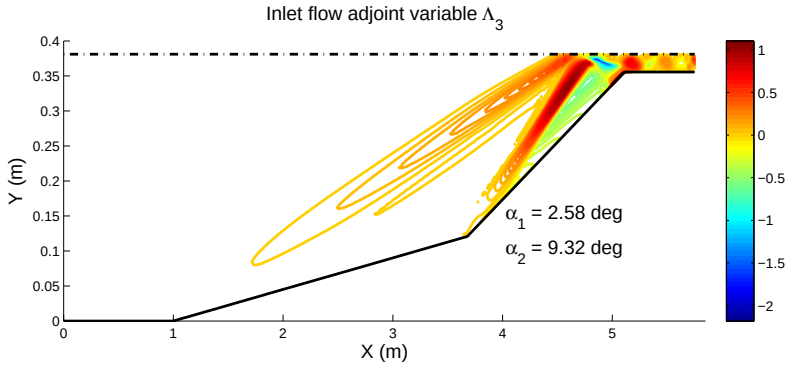


Figure 4.8. Contours of the adjoint variable Λ_3.

Given that we do not know what the true values of the adjoint variables should be, we must assess the success of the adjoint solver by it's ability to conform to the given boundary conditions. Figure 4.8 shows the contour of the third adjoint variable Λ_3; from this figure, we can see that the trends of the shock structures are carried

through from the flow solution.

On closer inspection of the boundaries, however, we find that the values are not exactly as prescribed in the boundary conditions, but are within reason. Figure 4.9 show the values of Λ_3 on the exit plane where the boundary condition is $\Lambda_3 = 0$ and Fig. 4.10 gives the values of Λ_3 on the upper wall. Here, the boundary conditions set $\Lambda_3 = 0$ on the entire boundary. Overall, we are confident in the adjoint solver's ability to capture the flow trends and generate reasonable values.

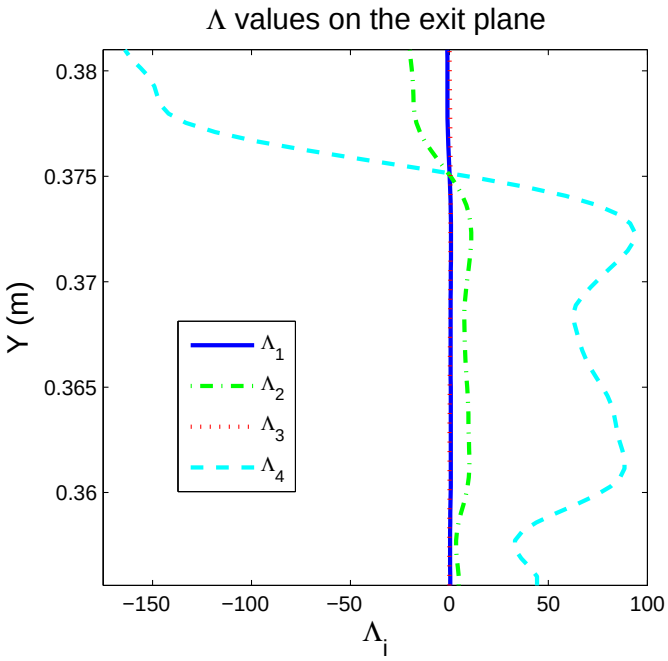


Figure 4.9. Λ_3 values on the exit boundary.

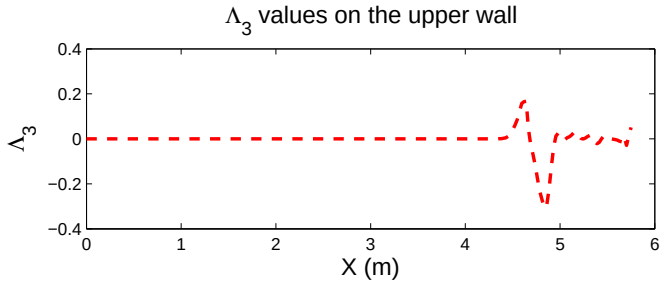


Figure 4.10. Λ_3 values on the upper wall boundary.

4.3.3 Computation of the gradients using the adjoint variables

With the adjoint equation and boundary conditions satisfied, the remaining term of Eq.(4.3) namely δH_α, makes up the first variation of the cost function

$$I = \frac{1}{\dot{m} p_{o_\infty}} \oint_{exit} \left(\frac{\partial \rho u (p_o - p_{o_\infty})}{\partial \boldsymbol{\alpha}} \right)^T \tilde{\boldsymbol{\alpha}} \boldsymbol{d} s + \int_\Omega \boldsymbol{\Lambda}^T \left[\frac{\partial}{\partial x} \left(\frac{\partial \mathbf{F}}{\partial \boldsymbol{\alpha}} \right) + \frac{\partial}{\partial y} \left(\frac{\partial \mathbf{G}}{\partial \boldsymbol{\alpha}} \right) \right] \tilde{\boldsymbol{\alpha}} d\Omega$$

Since the pressure loss parameter is only defined on the exit boundary, it does not explicitly depend on the control variable $\boldsymbol{\alpha}$ which is only defined on the lower wall of the ramp inlet. Hence, the first integral is zero and the final cost variation is given as

$$I = \int_\Omega \Lambda^T \left[\frac{\partial}{\partial x} \left(\mathbf{A} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} \right) + \frac{\partial}{\partial y} \left(\mathbf{B} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} \right) \right] \tilde{\boldsymbol{\alpha}} d\Omega \tag{4.6}$$

The $\partial \mathbf{F}/\partial \boldsymbol{\alpha}, \partial \mathbf{G}/\partial \boldsymbol{\alpha}$ terms are difficult to evaluate in this state because we cannot evaluate the change in the flow variables without running the flow solver again and this would null any benefits we hope to achieve using the adjoint variables. One method of evaluating these terms is presented in [3] and can be applied here. We

begin by linearizing the residual equation with respect to perturbations in the design parameter $\tilde{\boldsymbol{\alpha}}$

$$\boldsymbol{R}(\mathbf{U}, \boldsymbol{\alpha}) = \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0$$

When applied to the algebraic grid utilized here the fluxes are adjusted to account for the cell face normal, i.e.,

$$\boldsymbol{R}(\mathbf{U}, \boldsymbol{\alpha}) = \frac{\partial}{\partial x}\left(\mathbf{F} n_{xx} + \mathbf{G} n_{xy}\right) + \frac{\partial}{\partial y}\left(\mathbf{F} n_{yx} + \mathbf{G} n_{yy}\right) = 0$$

where n_{xx} and n_{xy} are the components of the surface normal vector in the x-direction $\mathbf{n_x}$, and similarly for n_{yx} and n_{yy} in the y-direction $\mathbf{n_y}$ (see Fig. 4.11). Small perturbations in the design parameter will produce changes to the components of this expression as

$$\mathbf{F} \rightarrow \mathbf{F} + \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} \tilde{\boldsymbol{\alpha}}, \quad \mathbf{G} \rightarrow \mathbf{G} + \frac{\partial \mathbf{G}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} \tilde{\boldsymbol{\alpha}}$$
$$n_{\star x} \rightarrow n_{\star x} + \frac{\partial n_{\star x}}{\partial \boldsymbol{\alpha}} \tilde{\boldsymbol{\alpha}}, \quad n_{\star y} \rightarrow n_{\star y} + \frac{\partial n_{\star y}}{\partial \boldsymbol{\alpha}} \tilde{\boldsymbol{\alpha}}$$

where the $\star$ in the normal components is either x or y. Substituting these changes

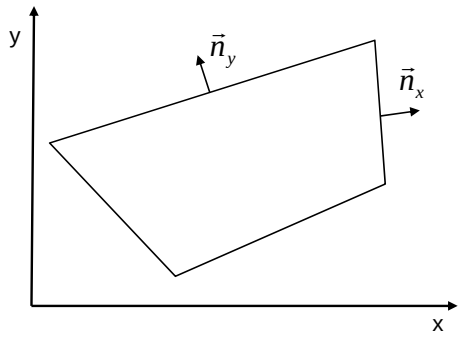


Figure 4.11. Cell face normal vectors.

into the residual equation and simplifying, we find that all terms not involving $\tilde{\boldsymbol{\alpha}}$ cancel out. Neglecting all terms containing $\tilde{\boldsymbol{\alpha}}^2$, the resulting linearized equation is given as

$$\begin{aligned}
\delta \boldsymbol{R} &= \frac{\partial}{\partial x}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}} n_{xx} + \frac{\partial \mathbf{G}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}} n_{xy}\right) + \frac{\partial}{\partial y}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}} n_{yx} + \frac{\partial \mathbf{G}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}} n_{yy}\right) + \\
&\quad \frac{\partial}{\partial x}\left(\mathbf{F}\frac{\partial n_{xx}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{xy}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}}\right) + \frac{\partial}{\partial y}\left(\mathbf{F}\frac{\partial n_{yx}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{yy}}{\partial \boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}}\right) \\
&= \left[\frac{\partial}{\partial x}\left(\mathbf{A}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{xx} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{xy}\right) + \frac{\partial}{\partial y}\left(\mathbf{A}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{yx} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{yy}\right) + \right. \\
&\quad \left. \frac{\partial}{\partial x}\left(\mathbf{F}\frac{\partial n_{xx}}{\partial \boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{xy}}{\partial \boldsymbol{\alpha}}\right) + \frac{\partial}{\partial y}\left(\mathbf{F}\frac{\partial n_{yx}}{\partial \boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{yy}}{\partial \boldsymbol{\alpha}}\right)\right]\tilde{\boldsymbol{\alpha}} = 0
\end{aligned}$$

Therefore we have

$$\begin{aligned}
&\left[\frac{\partial}{\partial x}\left(\mathbf{A}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{xx} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{xy}\right) + \frac{\partial}{\partial y}\left(\mathbf{A}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{yx} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} n_{yy}\right)\right] = \\
&\qquad - \left[\frac{\partial}{\partial x}\left(\mathbf{F}\frac{\partial n_{xx}}{\partial \boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{xy}}{\partial \boldsymbol{\alpha}}\right) + \frac{\partial}{\partial y}\left(\mathbf{F}\frac{\partial n_{yx}}{\partial \boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{yy}}{\partial \boldsymbol{\alpha}}\right)\right]
\end{aligned}$$

Substituting this into Eq.(4.6), we can eliminate the dependence on $\partial \mathbf{U}/\partial \alpha$. Thus we obtain the domain integral

$$\frac{\delta H}{\delta \boldsymbol{\alpha}} \approx -\int_{\Omega} \boldsymbol{\Lambda}^T \left[\frac{\partial}{\partial x}\left(\mathbf{F}\frac{\partial n_{xx}}{\partial \boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{xy}}{\partial \boldsymbol{\alpha}}\right) + \frac{\partial}{\partial y}\left(\mathbf{F}\frac{\partial n_{yx}}{\partial \boldsymbol{\alpha}} + \mathbf{G}\frac{\partial n_{yy}}{\partial \boldsymbol{\alpha}}\right)\right] d\Omega \qquad (4.7)$$

This integral is simple to evaluate and only requires that the grid be generated for each perturbation in the design variable. The computational cost of the grid generation is much less than the flow solution.

4.3.4 Optimization algorithm based on the adjoint variables

The following steps outline the algorithm implemented for the adjoint-based optimization routine:

1. Provide an initial condition $\boldsymbol{\alpha}_0$ for the inlet and generate the grid.
2. Solve the flow equations to steady state, Eq.(4.1).
3. Solve the adjoint equations to steady state, Eq.(4.4).
4. Solve for the changes in the cell surface normals due to changes in the design parameters. This is done numerically by perturbing each design variable and evaluating the changes.
5. Evaluate the cost function, Eq.(4.2).
6. Evaluate the variation in the cost function, Eq.(4.7) for each design variable.
7. Update the design variables and regenerate the grid.
8. Return to step 2 until cost function is extremized.

4.3.5 Results of the adjoint based gradient calculation for the 2-D full inlet

The adjoint gradients are computed for a range of α_1 and α_2 values to generate a vector map of the gradient trends. Consider a basic gradient descent optimization routine; the method is based on the observation that if any real-values function $F(x)$ is defined and differentiable in a neighbourhood of a point a, then $F(x)$ decreases the fastest if one goes in a direction of the negative gradient of F at a, $-\nabla F(a)$. Therefore we consider the negative of the gradients calculated using the method presented in the earlier section to determine if there is a trend to a stable point. The adjoint gradients can be seen in Fig. 4.12. For comparison the finite-difference gradients are shown in Fig. 4.13. The finite-difference gradients are plotted on top of the contour plot for the cost function to better understand their implication. The gradient vectors should

point normal to the contour lines of the cost function and towards the optimal value. The adjoint gradient in Fig. 4.12 do not suggest convergence to the optimal α_1 and α_2 however there is a trend to a stable region in close proximity to the optimal.

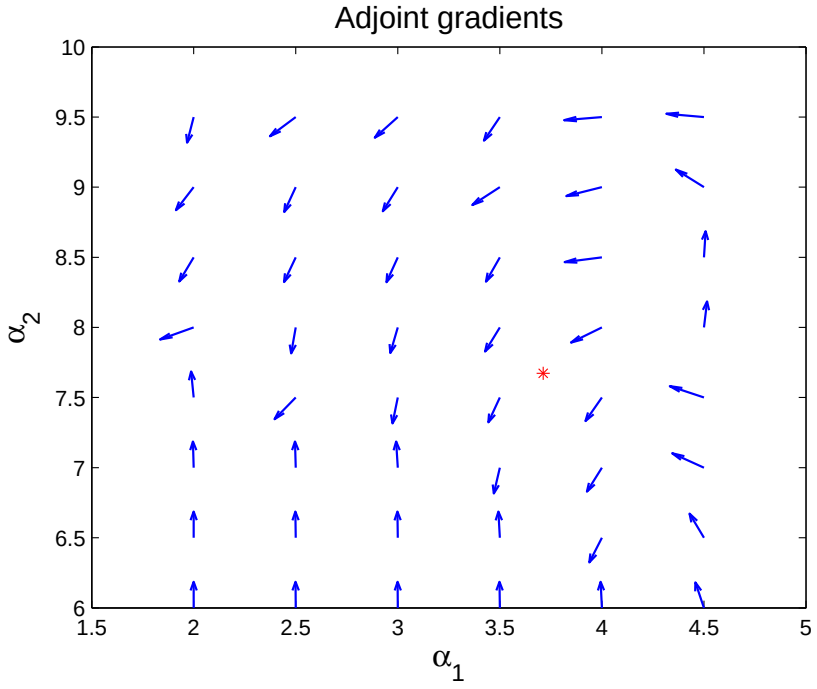


Figure 4.12. Adjoint gradient survey for the full inlet ($* = (\alpha_1, \alpha_2)_{optimal}$).

4.4 Shape optimization using POD based reduced order models

We now investigate the use of a POD reduced order model for shape optimization of the 2-D inlet. To our knowledge this is the first instance of the use of POD based optimization for this specific problem. The method of developing the POD reduced order model and using it in an optimization problem was demonstrated in the previous section for the quasi 1-D duct. The basis functions for the POD model are derived from snapshots of the converged flow for various values of the control variable.

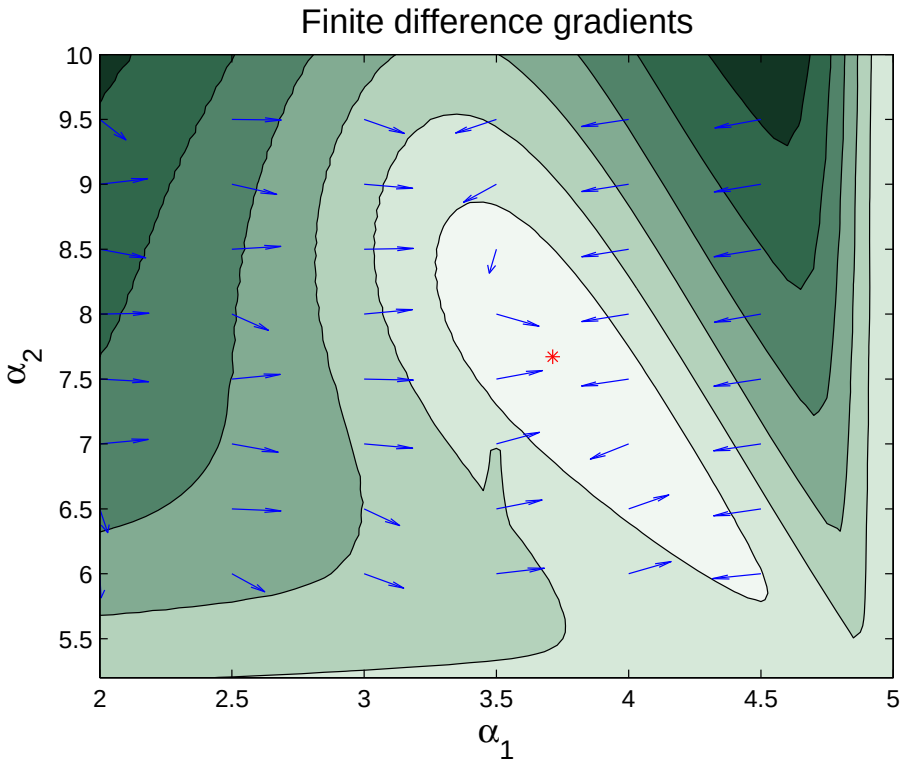


Figure 4.13. Finite difference gradient survey for the full inlet ($* = (\alpha_1, \alpha_2)_{optimal}$).

Weights corresponding to each basis function are determined in a simple and unique way by minimizing the variation between the individual snapshot solutions and the POD model output. Finally, a response surface approximation is obtained for the weights as a function of the design variables. This smooth approximation enables the overall optimization process that is verified in simulation studies.

4.4.1 POD based reduced order model for the 2-D full inlet

The POD model reduction method is applied to the 2-D double ramp inlet problem illustrated in Fig. 4.1. The POD reduced order model will be used in the optimization of the shape parameters. Again the control variables, or shape parame-

ters, are the angles of the two ramp portions of the inlet (α_1, α_2).

The 2-D full inlet is approached in the same manner as the quasi 1-D duct except this time the governing equations are the 2-D Euler equations given in Eq.(4.1). Following the algorithm given in section 3.3 for determining the POD model, we let $\mathbf{U}^{(i)}(x, y, \alpha_1, \alpha_2)$ be a steady-state solution of the flow equations for a set of design variables (α_1, α_2):

1. Generate a set of N snapshots $\mathbf{U}^{(i)}$ from steady state flow solutions of Eq.(4.1) for a random set of α_i's.
2. Compute the correlation matrix $\mathbf{C}$ from Eq.(2.9).
3. Solve the eigenvalue problem $\mathbf{CW} = \lambda\mathbf{W}$ to get a complete set of eigenvectors w_i.
4. Obtain the basis functions $\mathbf{\Phi}_i$ using Eq.(2.10).
5. Determine the corresponding weights in Eq.(2.13) by minimizing the cost function given in Eq.(2.12).
6. Determine a relationship for the weights as a function of (α_1, α_2).

For this case, the steady-state inlet flow is computed for 504 permutations of the control variables with α_1 ranging between 2.5° and 4.8° and α_2 ranging between 6.5° and 8.5°. The eigenvalues for the correlation matrix computed in step 3 are shown in Fig. 4.14. In order to better visualize the trends we plotted every eighth eigenvalue. To retaining 99.5% of the energy in the system we must keep the first 11 modes. In step 6 we consider two methods for generating this functional relationship: a p^{th}-order polynomial response surface, and an artificial neural network.

$\mathbf{p^{th}}$-order response surface: The weights computed in step 5 above are used to generate a response surfaces to the N_k combination of the design parameters. We

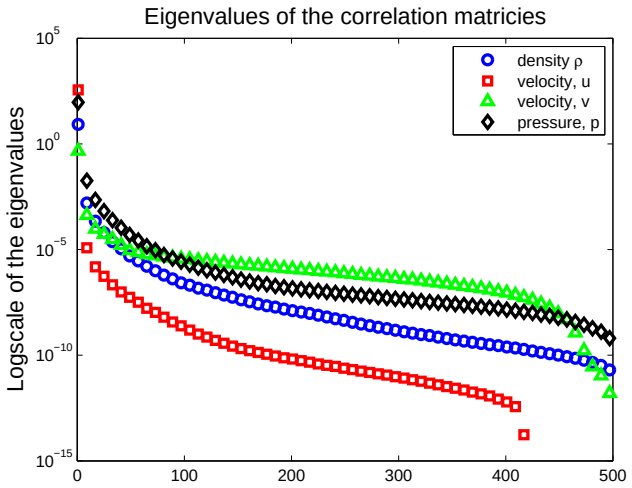


Figure 4.14. Eigenvalues for correlation matrices for each of the primitive variables.

suppose that the desired ROM will best approximate the snapshot solutions. The functional representation for the weights is obtained using a least squares polynomial approximation for any specified order. The results for 3$^{\text{rd}}$- and 5$^{\text{th}}$-order surfaces are compared. For the sake of illustration, we will outline the procedure to obtain the least squares based response surface for the vector weight function a_ρ, see Eq.(2.11). For each of the N_k combinations of the parameters, we first assume that the function can be represented by the following

$$
\begin{aligned}
a_{\rho i,j} &= (c_{\rho,0} + c_{\rho,1}\alpha_{1,j} + c_{\rho,2}\alpha_{1,j}^2 + \cdots + c_{\rho,p}\alpha_{1,j}^p) \\
&\quad \otimes \quad (b_{\rho,0} + b_{\rho,1}\alpha_{2,j} + b_{\rho,2}\alpha_{2,j}^2 + \cdots + b_{\rho,p}\alpha_{2,j}^p) \\
&= \begin{bmatrix} 1 & \alpha_{2,j} & \alpha_{2,j}^2 & \cdots & \alpha_{1,j} & \alpha_{1,j}\alpha_{2,j} & \alpha_{1,j}\alpha_{2,j}^2 & \cdots & \alpha_{1,j}^p\alpha_{2,j}^p \end{bmatrix} \\
&\quad * \begin{bmatrix} c_{\rho,0}b_{\rho,0} & c_{\rho,0}b_{\rho,1} & c_{\rho,0}b_{\rho,2} & \cdots & c_{\rho,1}b_{\rho,0} & c_{\rho,1}b_{\rho,1} & c_{\rho,1}b_{\rho,2} & \cdots & c_{\rho,p}b_{\rho,p} \end{bmatrix}^T \\
&= \mathbf{H}_j \boldsymbol{X}
\end{aligned}
$$

where $j = 1, \ldots, N_k$. The parameters $c_{\rho,i}$ and $b_{\rho,i}$ are obtained as a linear least-squares solution

$$\mathbf{X} = \left[\mathbf{H}^T\mathbf{H}\right]^{-1}\mathbf{H}^T\mathbf{a}_{\rho i}$$

Neural networks: The neural network function approximation utilizes the weights calculated from the snapshots and their corresponding design parameters to train a basic feedforward backpropagation network. The feedforward type network allows only the forward transmission of information through the network without cycles or loops and the backpropagation is used in the training of the network. Backpropagation training involves calculating the gradient of the error with respect to the network's modifiable weights. Figures 4.15 and 4.16 demonstrate the network structure. In this study the network is trained using a Levenberg–Marquardt optimization routine to update the weight and bias values. The backpropagation routine requires the activation functions used by the nodes to be differentiable, therefore we chose the common tan-sigmoid transfer function for all layers. In this work we compare the results for both 2- and 3-layer networks with 7 nodes for the hidden layer of the 2-layer network and 5 and 7 nodes for the hidden layers of the 3-layer network.

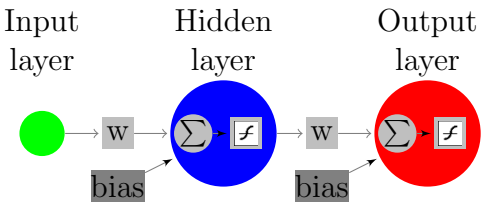


Figure 4.15. Typical 2-layer neural network with bias.

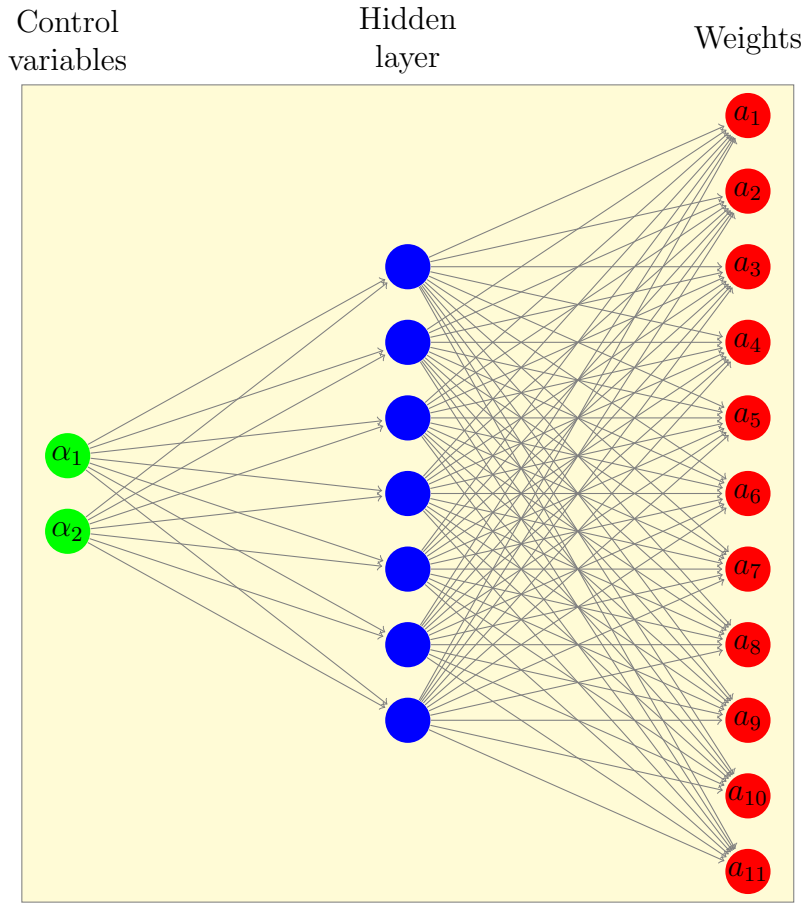
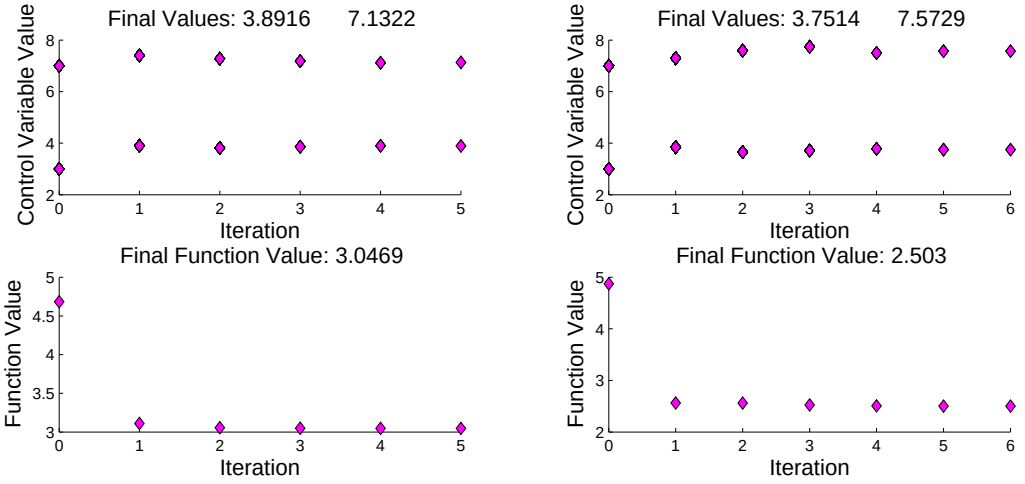


Figure 4.16. Actual 2-layer neural network structure for POD model weight function.

4.4.2 Optimization of the 2-D full inlet using the POD model

Consider the same cost function is that used in the finite difference and adjoint problems; the mass weighted total pressure loss. Figure 4.17 gives the results of the POD model optimization for the 3^{rd}- and 5^{th}-order polynomial surfaces respectively. The top figure in each shows the evolution of the control variables and the lower figure gives the evolution of the cost function for each iteration. The 3^{rd}-order polynomial approximation does not perform as well as the 5^{th}-order polynomial. The 5^{th}-order polynomial fit was better able to captured the trends of the weights, and therefore

was able to converge to values very close to the optimal.



(a) (b)

Figure 4.17. Results of the full inlet POD model optimization using (a) 3^{rd}-order and (b) 5^{th}-order polynomial response surfaces.

The optimization results based on the neural network weight function showed minor variances for different network generations. The training process was not consistent for networks generated from the same data, we therefore ran through a Monte–Carlo simulation to determine the mean and standard deviation of the optimization results. In the Monte–Carlo simulation the neural network is generated and used in the optimization of the inlet angles up to 900 times. After each batch of simulations the mean and standard deviation are calculated as well as the percentage of converged solutions. The converged solutions consist of those in which the optimization did not result in any of the constraint conditions. In this case the constraints for the optimization routine were the same as the limits used to generate the original POD snapshots, $2.5° < \alpha_1 < 4.8°$ and $6.5° < \alpha_2 < 8.5°$. The results of the Monte–Carlo

simulation for both the 2- and 3-layer networks are given in Fig. 4.18. Note that the 3-layer network has a very high percentage for convergence and consistent mean values for the different solution batches.

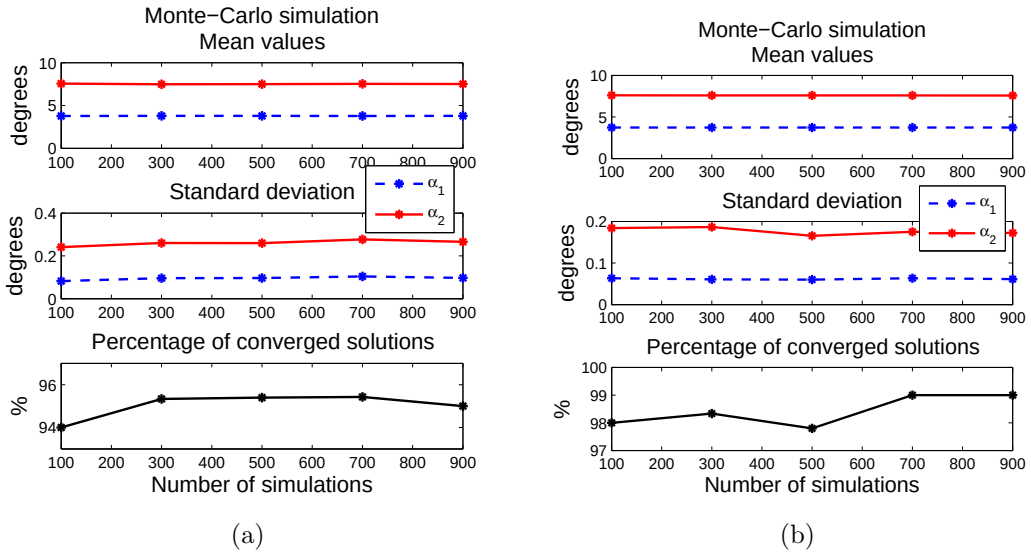


Figure 4.18. Monte–Carlo results of the full inlet POD optimization using (a) 2-layer and (b) 3-layer neural networks.

The results of the POD model optimization are summarized in Table 4.2 and are compared with the results of a finite difference optimization. Note that the results of the POD based optimization matches very closely with the finite difference optimization run on the high-fidelity model.

Table 4.2. Comparison of the optimal shape parameters from the full-order and POD reduced order model finite-difference optimizations

Case	Angles (deg)	
	α_1	α_2
Full-order model	3.7129	7.6724
POD model		
3^{rd}-order response surface	3.892	7.132
5^{th}-order response surface	3.751	7.573
2-layer NN	3.786 $\pm$ 0.095	7.517 $\pm$ 0.26
3-layer NN	3.728 $\pm$ 0.062	7.591 $\pm$ 0.18

CHAPTER 5

2-D COWL INLET SHAPE OPTIMIZATION

5.1 2-D cowl inlet

This case study is similar to the previous full inlet study except here the upper half of the inlet is removed and a short cowl is used to capture the incoming air. Figure 5.1 illustrates the 2-D cowl inlet problem under consideration. The inlet is characterized by a lower wall designed to influence the flow entering the engine. The cowl depicted in the figure is necessary to capture the flow. As in the full inlet case, the shape of the lower wall is typically designed for a small operating range. Changing the shape of the inlet would allow it to perform optimally over a broader range of flight conditions. This section explores the use of adjoints and reduced order models in the optimization of the shape parameters. Again we recognize that the use of adjoints for optimization of only two design variables is an overkill; future work is related to using a continuously varying inlet geometry parameterized as several segments.

5.2 Generation of true optimized values for comparison — Finite difference gradient approach

In this section, we focus on first generating the true values based on a finite difference based gradient approach as well as brute force parametric studies of the 2-D cowl inlet. We study this inlet in a bit more detail by evaluating the benefit of moving from a single ramp inlet to multiple segments. The optimization approach uses a Levenberg–Marquardt scheme and the gradients are evaluated based on a finite

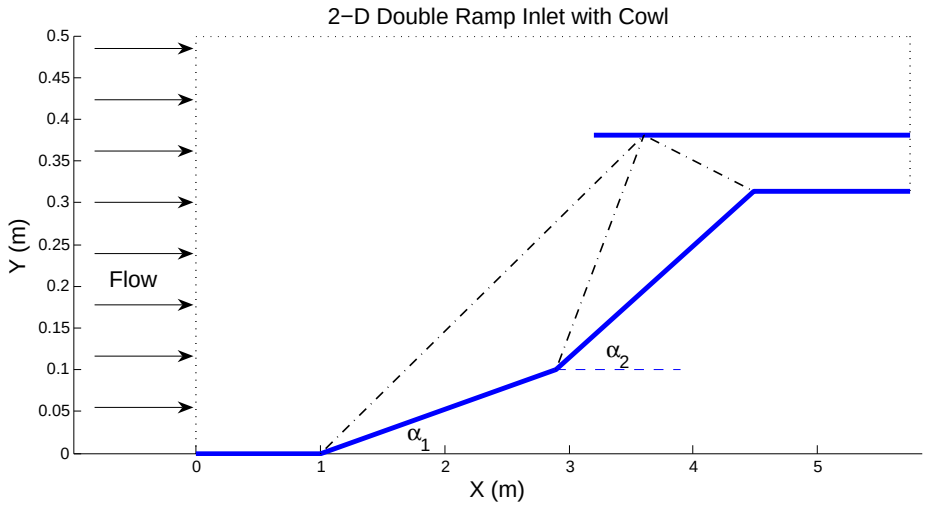


Figure 5.1. 2-D ramp schematic.

difference scheme. This serves as the "truth" to which the results obtained via the POD and the adjoint methods are compared.

5.2.1 Single ramp cowl inlet performance

The optimization set up begins with establishing a reliable and well-posed problem. To this end we will first consider the single ramp inlet to a scramjet engine. Given an inflow Mach number of 14, the various geometric parameters of the inlet need to be specified such that an appropriate objective function will have a local minimum within the design space.

To optimize the geometry of the single ramp inlet, the cowl lip x location and the shoulder x location are chosen as control variables. The radii of the inlet and exit are fixed as is the length of the inlet and the initial ramp turning location. At this point we need to choose the objective function; as in the previous case we must

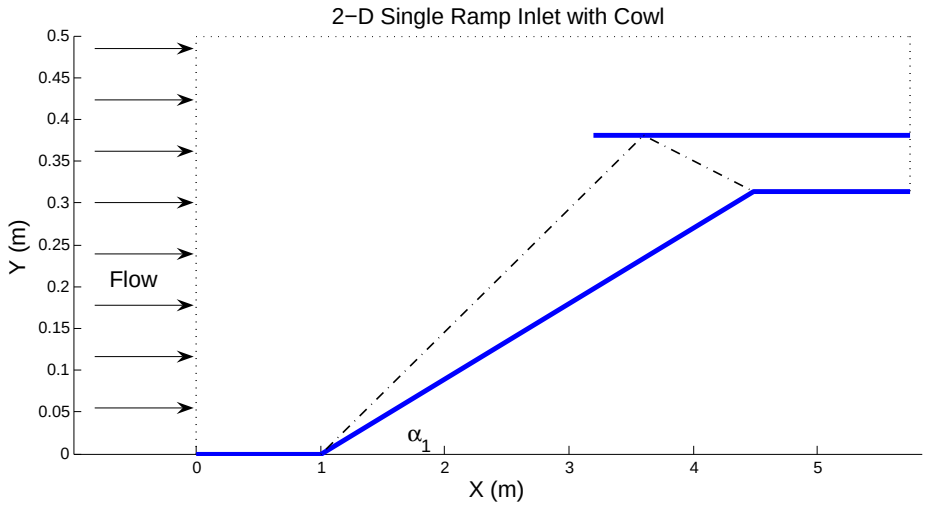


Figure 5.2. Schematic of the single ramp cowl inlet.

consider the mass weighted total pressure loss but with this cowl style inlet we can also consider the mass capture.

$$J_{PressureLoss} = \frac{\sum \rho_{i,j} u_{i,j} \Delta y_{i,j} \left(p_{o_{i,j}} - p_{o_\infty} \right)}{\dot{m} p_{o_\infty}} \tag{4.2}$$

$$J_{MassCapture} = \sum \rho_{i,j} u_{i,j} \Delta y_{i,j} \tag{5.1}$$

Both objective functions have a direct implication on the performance of the inlet. The mass capture measures the amount of air entering the engine; this should be maximized in order to draw as much energy as possible from the combustion process. The total pressure loss essentially measures the efficiency of the engine; minimizing the total pressure loss is the same as minimizing the loss in energy as the air enters the inlet.

The optimization routine based on the finite difference based gradients is implemented in MATLAB® to obtain the optimal design variables. This led to a definite and repeatable result for the shoulder location and a more vague result for the cowl lip location. The shoulder was consistently located at the point where the shock reflecting off the cowl wall met the lower wall. This scenario is depicted in the inlet schematic, Fig. 5.2. The optimal cowl lip location is a bit more vague. In order to better understand the consequences of this variable we investigated the cost function over a range of values for the cowl lip location. Figure 5.3 gives the results of this study for both the mass weighted pressure loss and the mass capture. In this figure the objectives are a maximum in the mass capture and a minimum in the pressure loss. The mass capture of the inlet increases very gradually as the cowl lip is moved from the left and reaches a maximum when the cowl lip is at $x = 3.55$. It drops very dramatically for values greater than $x = 3.55$. The pressure loss has a local minimum at $x = 3.2$ with a spike at the maximum mass capture value. It then drops continually as the cowl lip location moves further to the right. On comparing this with the mass capture variation we can see that this essentially removes the effect of the cowl altogether such that the flow is only subject to the ramp wall and we no longer have an inlet. Our optimal value then is chosen to be $x = 3.2$ since it still provides good mass capture and minimizes the pressure loss.

Figure 5.4 gives the pressure contours from the CFD solution of our optimal single ramp configuration. Notice the shock that is generated from the cowl lip is bent slightly. This and the reflected shock from the ramp meet at the shoulder of the inlet. There is minimal reflection of the shocks downstream of the shoulder.

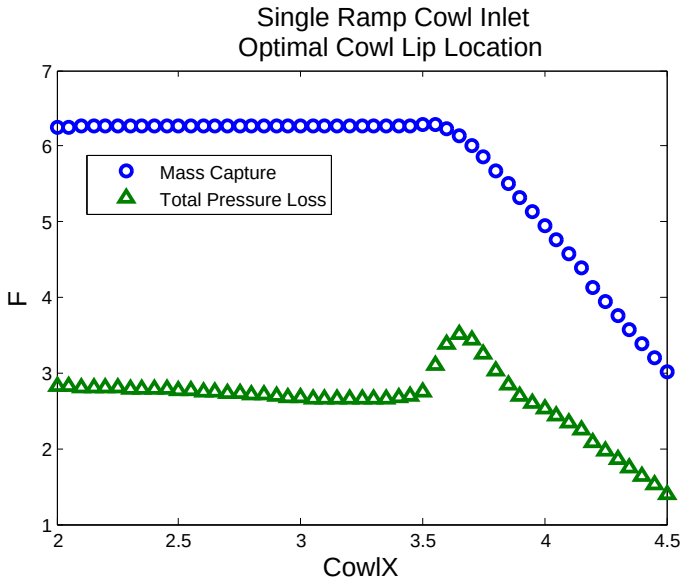


Figure 5.3. Plot of the pressure loss and mass capture vs the cowl lip location for the single ramp inlet.

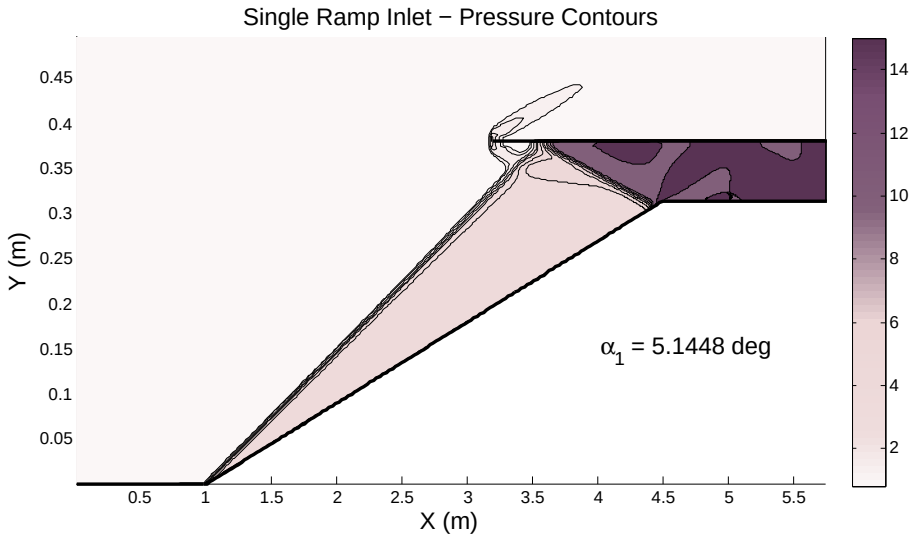


Figure 5.4. Pressure contours of the optimal configuration for the single ramp cowl inlet.

5.2.2 Double ramp cowl inlet performance

With the single ramp optimized we can now fix the basic geometry and look at the use of multiple ramps and optimize to improve the mass weighted pressure loss. This case study will modify the single ramp inlet to two ramps of increasing slope where the maximum allowable α_1 and the minimum allowable α_2 are the single ramp value of $(\alpha_1 \leq \alpha_{SingleRamp} \leq \alpha_2)$.

The true optimized ramp angles are generated using the same Levenberg Marquardt based optimization procedure that uses the finite difference based gradients. The optimization routine is implemented in MATLAB®. The initial condition to the optimization is given as $\alpha_1 = 3.0°$ and $\alpha_2 = 8.0°$. The pressure contour plot of this ramp configuration is shown in Fig. 5.5. The optimized true values were obtained as $\alpha_1 = 4.8446°$ and $\alpha_2 = 5.6458°$; the pressure contours of this optimized inlet are shown in Fig. 5.6. The pressure loss is improved from 2.6566 for the optimal single ramp to 2.2581 for the optimal 2 ramp inlet. The initial conditions supplied to the optimizer gives the pressure loss to be 6.3444.

In order to further validate the results of the optimization routine we computed the flow solution and evaluated the cost function for a range of α_1 and α_2 to generate a contour plot. This plot is shown in Fig 5.7. As can be seen in the figure there is a single minimum in the lower right corner. The values at this minimum point correspond with the results of optimization routine and are found to be $\alpha_1 = 4.85°$ and $\alpha_2 = 5.55°$ with a pressure loss of 2.2560. A summary of all the validation results is given in table 5.1.

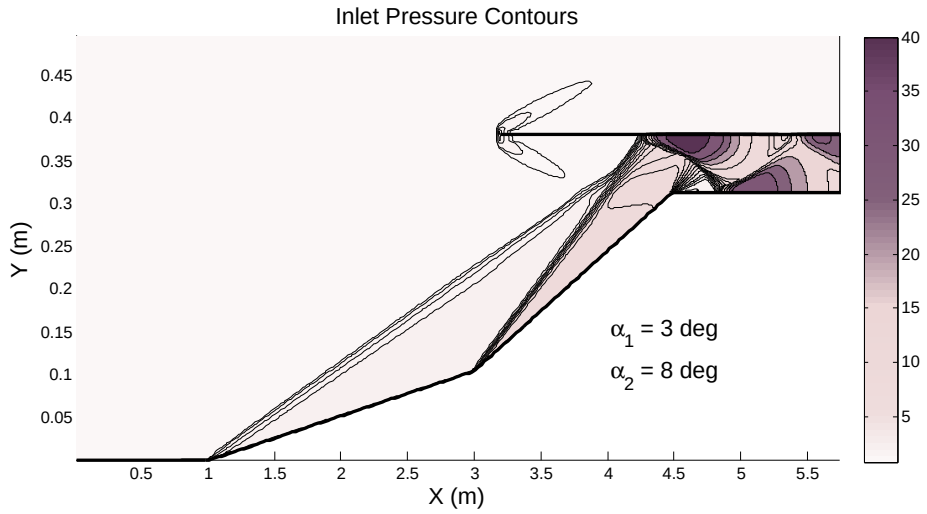


Figure 5.5. Pressure contours of the initial condition for the cowl inlet optimization.

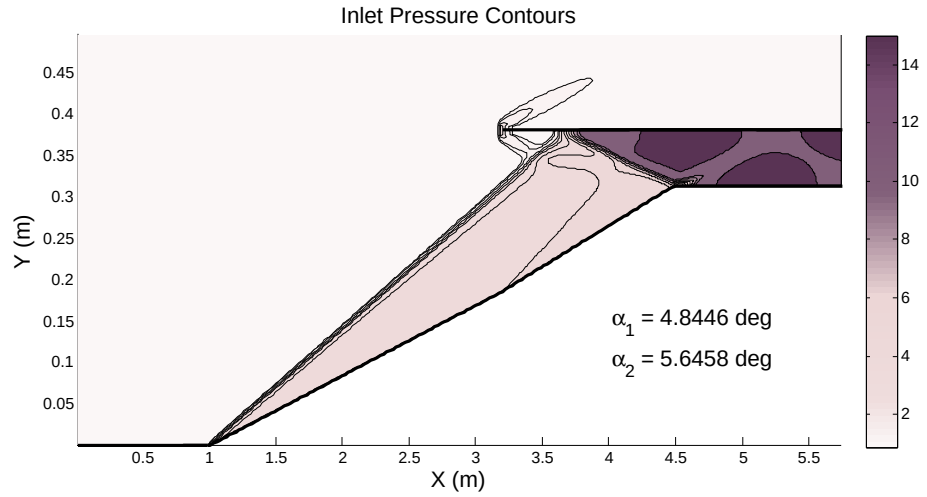


Figure 5.6. Pressure contours of the optimal configuration for the cowl inlet.

Table 5.1. Values of the objective function for single and double ramp cowl inlets

Case	Angles (deg) α_1	α_2	Mass Weighted Total Pressure Loss
Single Ramp (parametric study)	5.1448	—	2.6566
Double Ramp (parametric study)	4.85	5.55	2.2560
Double Ramp (initial condition)	3.0	8.0	6.3444
Double Ramp (optimization)	4.8446	5.6458	2.2581

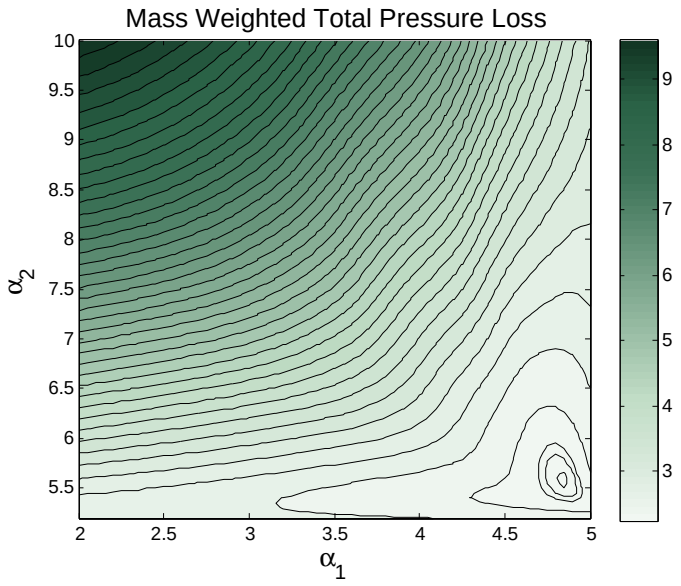


Figure 5.7. Cost function contour for various α_1 and α_2 of the cowl inlet.

5.3 Shape optimization using the adjoint method based gradients

Given the similarities with the previous case, the adjoint derivation is identical that presented in section 4.3. The only difference here is in the application of the boundary conditions. The boundary conditions themselves remain the same but where they are applied is slightly different. The wall boundary condition is applied to the lower wall as well as the cowl wall. The exit flow condition is applied only to that small section inside the inlet and in this case the upper boundary is treated as a free stream (out-flow) condition.

5.3.1 Results of the adjoint based gradient calculation for the 2-D cowl inlet

The adjoint gradients are computed for a range of α_1 and α_2 to generate the vector map discussed in the previous case. Figure 5.8 gives the results of this survey. Again we plot the negative of the calculated gradients to determine if there there is

a trend to a stable point (see Fig. 5.8). As in the full inlet case, the adjoint gradients do not imply convergence to the optimal value but the vectors do suggest that there is a stable region in close proximity to the optimal.

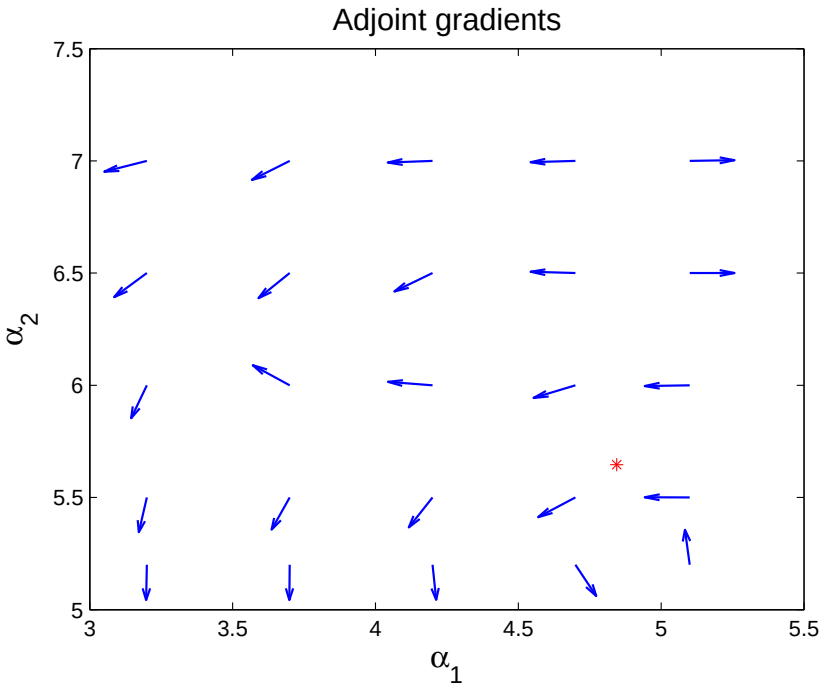


Figure 5.8. Adjoint gradient survey for the cowl inlet ($* = (\alpha_1, \alpha_2)_{optimal}$).

5.4 Shape optimization using POD based reduced order models

5.4.1 POD based reduced order model for the 2-D cowl inlet

The POD model generation procedure is identical to the previous case. The only variations are in the number of snapshots and the range and number of control variations. In this case the steady state inlet flow is computed for 88 permutations of the control variables with α_1 ranging between 4.1° and 5.15° and α_2 ranging between 5.2° and 6.7°. The eigenvalues for the correlation matrix are shown in Fig. 5.9 (every

second eigenvalue is plotted for clarity). Notice the amount of information contained in each mode drops off quickly initially then becomes very gradual. The first 3 modes contain 99.0% of the energy in the system, where the first 14 modes contain 99.9%. To gain one more 0.09% to reach 99.99% we need to jump from 14 to 53 modes. Thus to maintain accuracy yet keep the computation time low, we chose to keep 14 modes. The methods for generating the functional relationship is step 6 of the algorithm also remain the same. We use a 3^{rd}- and 5^{th}-order polynomial surface fit and 2- and 3-layer neural networks with 7 nodes for the hidden layer of the 2-layer network and 5 and 7 nodes for the hidden layers of the 3-layer network.

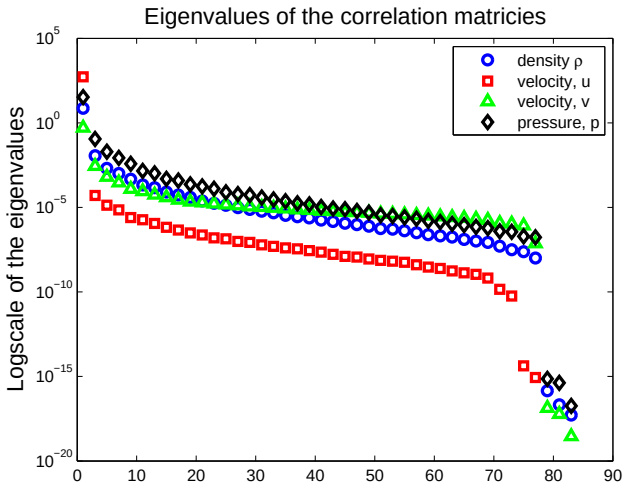
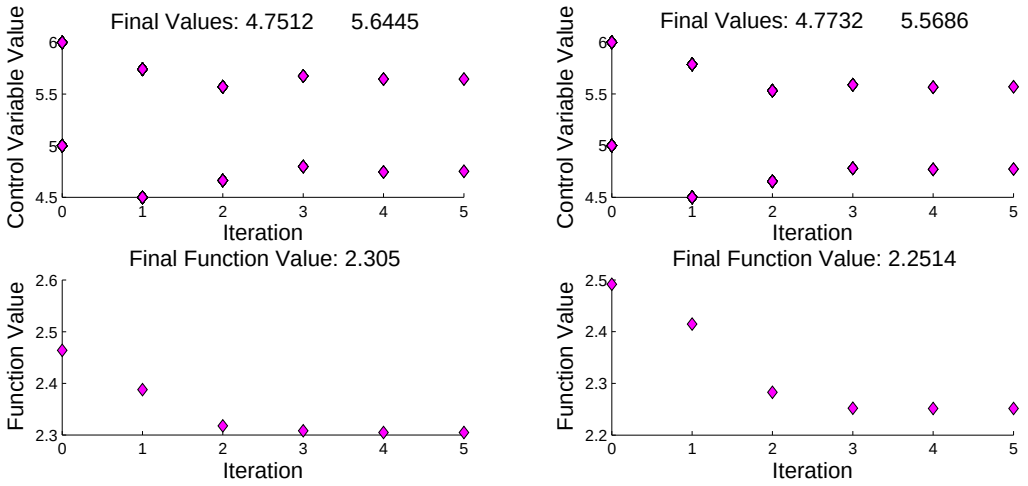


Figure 5.9. Eigenvalues for correlation matrices for each of the primitive variables.

5.4.2 Optimization of the 2-D cowl inlet using the POD model

Figure 5.10 shows the POD model optimization results for 3^{rd}- and 5^{th}-order response surfaces respectively. The top figure in each shows the evolution of the control variable for each iteration of the optimization and the lower figure gives the

evolution of the cost function. The optimization is completed in only 5 iterations.



(a) (b)

Figure 5.10. Results of the POD model optimization using (a) 3^{rd}-order and (b) 5^{th}-order polynomial response surfaces.

As in the previous case, the neural network training process was not consistent a Monte–Carlo simulation was run to determine the mean and standard deviation of the optimization results. In this case the neural network is generated and used in the optimization of the inlet angles up to 900 times. After each batch of simulations the mean and standard deviation are calculated as well as the percentage of converged solutions. The constraints for the optimization routine were the same as the limits used to generate the original POD snapshots, $4.1° < \alpha_1 < 5.15°$ and $5.2° < \alpha_2 < 6.7°$. The results of the Monte–Carlo simulation for both the 2- and 3-layer networks are given in Fig. 5.11. Note that the 2-layer network has a higher percentage of convergence but the 3-layer network has lower standard deviations of the mean values. The results of this study may vary depending on the number of snapshots as well.

The network is trained using all the data from the original set of snapshots, if that number were increased the network training may be more consistent.

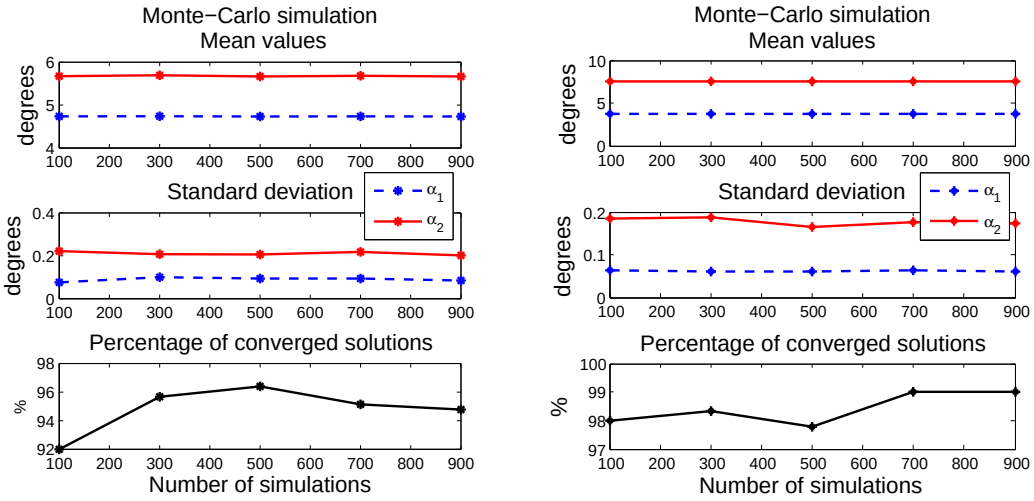


Figure 5.11. Monte–Carlo results of the POD model optimization using (a) 2-layer and (b) 3-layer neural networks.

The results of the POD model optimization are summarized in Table 5.2 and are compared with the results of a finite difference optimization. As in the full inlet case, the results of the POD based optimization matches very closely with the finite difference optimization run on the high fidelity model.

Table 5.2. Comparison of the optimal shape parameters from the full-order and POD reduced order model finite-difference optimizations

Case	Angles (deg)	
	α_1	α_2
Full-order model	4.8446	5.6458
POD optimization		
3^{rd}-order response surface	4.751	5.645
5^{th}-order response surface	4.773	5.569
2-layer NN	4.736 ± 0.090	5.677 ± 0.21
3-layer NN	4.756 ± 0.078	5.663 ± 0.12

CHAPTER 6

2-D CHANNEL FLOW CONTROL

For this study we focus on yet another set of PDEs, namely, the 2-D Burgers equation for a channel flow. The Burgers equation is essentially a model of nonlinear wave propagation subject to dissipation. The dissipative term is linear and the convective term is nonlinear. It is an initial boundary value problem which describes the evolution of the function, $U(t, x, y)$. In this study, we implement a linear controller on the nonlinear system.

6.1 Burgers equation

Johannes Burgers introduced his equation in 1940 as a simple model for turbulence; however, it is also capable of representing the convection/diffusion phenomena seen in the Navier-Stokes equations making it a popular means of testing control strategies. It is a simple and convenient model for traffic flows, acoustic transmission, shock waves, and supersonic flow about airfoils. The Burgers equation is given as follows

$$\frac{\partial}{\partial t}U(t,x,y)+K_1\frac{\partial}{\partial x}U^2(t,x,y)+K_2\frac{\partial}{\partial y}U^2(t,x,y) = \frac{1}{Re}\left[\frac{\partial^2}{\partial x^2}U(t,x,y) + \frac{\partial^2}{\partial y^2}U(t,x,y)\right] \tag{6.1}$$

for $t > 0$ and where K_1 and K_2 are scaling constants for the convective terms and Re is a nonnegative scaling constant for the dissipation analogous to the Reynolds number in the Navier-Stokes equations.

The specific problem addressed here was originally developed by Camphouse and Myatt [29] as a means to develop control methodologies for fluid flow governed by the nonlinear Navier-Stokes equations. Consider a channel or pipe flow with an obstacle as shown in figure 6.1. This case simulates a right travelling shock around an obstacle. Control is implemented through the walls of the obstacle to stabilize desired flow field structures.

Let us first define the domain $\Omega_1 \subseteq \Re^2$ as an open rectangle given by $(x_1, x_2) \times (y_1, y_2)$ and the domain $\Omega_2 \subseteq \Re^2$ as the rectangle given by $[a_1, a_2] \times [b_1, b_2]$. We restrict the size of Ω_2 such that $x_1 < a_1 < a_2 < x_2$ and $y_1 < b_1 < b_2 < y_2$ or $\Omega_2 \subset \Omega_1$. The final problem domain Ω is now given by $\Omega = \Omega_1 \backslash \Omega_2$ where Ω_2 defines the obstacle in the flow.

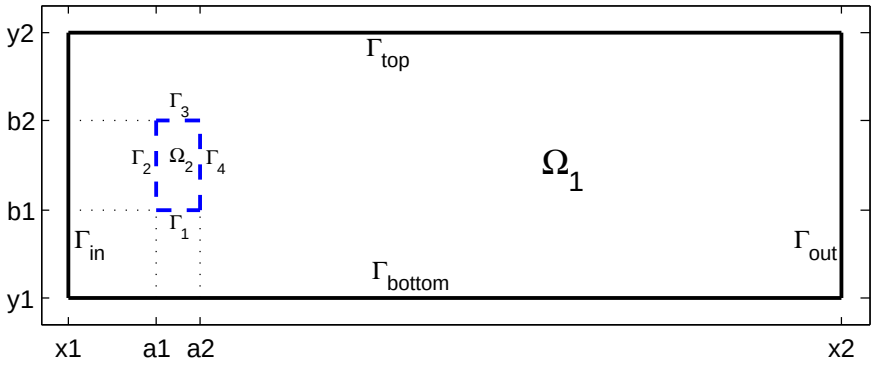


Figure 6.1. Channel flow schematic.

To fully define the problem we must specify the boundary conditions $\partial\Omega_1$ and $\partial\Omega_2$ and initial condition. On the Ω_1 domain, the top and bottom boundaries are treated as walls and are defined by Dirichlet boundary conditions. The outflow is

defined by a Neumann condition and the inflow is defined by a forcing function given in figure 6.2.

$$\Gamma_{in} = \{(x_1, y) \mid y_1 \leq y \leq y_2\}, \qquad \Gamma_{bottom} = \{(x, y_1) \mid x_1 \leq x \leq x_2\}$$
$$\Gamma_{out} = \{(x_2, y) \mid y_1 \leq y \leq y_2\}, \qquad \Gamma_{top} = \{(x, y_2) \mid x_1 \leq x \leq x_2\}$$

$$U(t, \Gamma_{in}) = f(y), \qquad U(t, \Gamma_{bottom}) = 0$$
$$U_x(t, \Gamma_{out}) = 0, \qquad U(t, \Gamma_{top}) = 0 \tag{6.2}$$

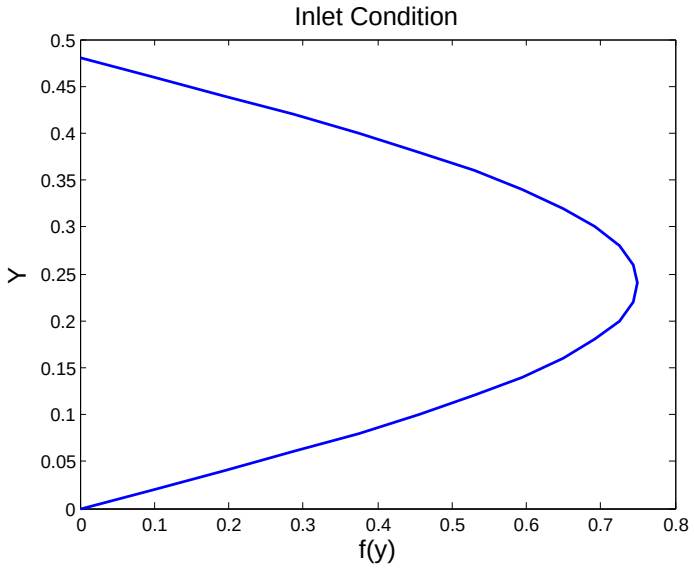


Figure 6.2. Inlet condition for channel flow.

The obstacle boundaries are implemented as Dirichlet boundaries and are denoted as

$$\Gamma_1 = \{(x, b_1) \mid a_1 \leq x \leq a_2\}, \qquad \Gamma_2 = \{(a_1, y) \mid b_1 \leq y \leq b_2\}$$
$$\Gamma_3 = \{(x, b_2) \mid a_1 \leq x \leq a_2\}, \qquad \Gamma_4 = \{(a_2, y) \mid b_1 \leq y \leq b_2\}$$

Since the control is implemented through this boundary, we specify the conditions here as a linear combination of a spatial and a temporal term

$$U_1(t, \Gamma_1) = \sum_{i=1}^{C_1} \alpha_{1,i}(t)\phi_{1,i}(x), \qquad U_2(t, \Gamma_2) = \sum_{i=1}^{C_2} \alpha_{2,i}(t)\phi_{2,i}(x)$$
$$U_3(t, \Gamma_3) = \sum_{i=1}^{C_3} \alpha_{3,i}(t)\phi_{3,i}(x), \qquad U_4(t, \Gamma_4) = \sum_{i=1}^{C_4} \alpha_{4,i}(t)\phi_{4,i}(x)$$

where C_i is the number of controls on Γ_j, $\alpha_{i,j}$ is the i_{th} control on Γ_j, and the function $\phi_{i,j}$ is a function describing the influence of the i_{th} control on Γ_j. In the cases presented here, there are two controls on the front and back of the obstacle, $C_2 = C_4 = 2$ and one control on the top and bottom, $C_1 = C_3 = 1$. The controls are shown graphically in figure 6.3. The boundary conditions are piecewise continuous on each surface and are now of the form

$$U_1(t, \Gamma_1) = \alpha_1\phi_1, \qquad U_2(t, \Gamma_2) = \alpha_{2,1}\phi_{2,1} + u_{2,2}\phi_{2,2} \tag{6.3}$$
$$U_3(t, \Gamma_3) = \alpha_3\phi_3, \qquad U_4(t, \Gamma_4) = \alpha_{4,1}\phi_{4,1} + u_{4,2}\phi_{4,2}$$

To complete the problem description the initial condition on the flow is specified as

$$U(0, x, y) = U_o(x, y) \in \Omega \tag{6.4}$$

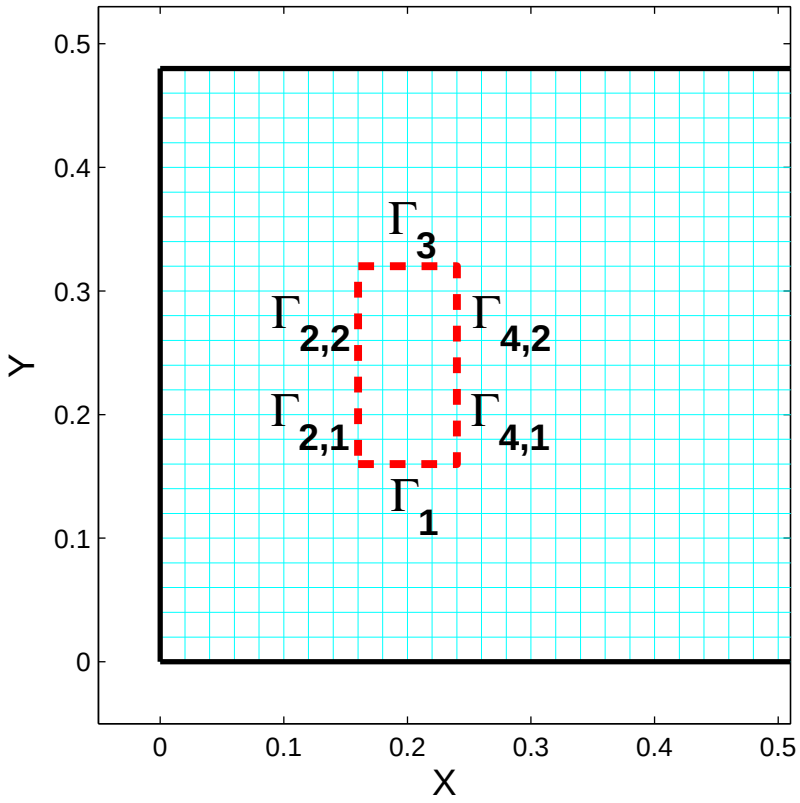


Figure 6.3. Obstacle control surfaces used in channel flow.

6.2 Boundary control using linear control routines

There are numerous control strategies available for the linear problem. However, to implement a linear controller on the system model, the governing equation must be linearized and rewritten as an abstract Cauchy problem of the form

$$\frac{\partial}{\partial t}U = \mathcal{A}U + \mathcal{F} + \mathcal{B}\alpha, \qquad t > 0$$

where $\mathcal{A}$ is the state matrix for the linearize flow equation $\mathcal{F}$ is a forcing term setting the inflow boundary condition and $\mathcal{B}$ is the control input matrix. The resultant form

of this equation is dependent on the chosen discretization method.

The goal with any control strategy is to determine the optimal control variables necessary to achieve a zero flow state in a downstream region from the obstacle. To this end, the cost function is formulated in a standard LQR format with an α_s shift (α_s-LQR),

$$J_\alpha(U_0, \alpha) = \int_0^\infty \{\langle \mathcal{Q}U, U\rangle + \langle \mathcal{R}\alpha, \alpha\rangle\} e^{2\alpha_s t} dt$$

where $\mathcal{Q} = \mathcal{C}^T\mathcal{C}$. It can be shown that the optimal solution for the α_s-LQR is the same as that for the standard LQR problem, namely

$$\begin{aligned} \alpha_{opt} &= -\mathcal{K}U \\ &= \mathcal{R}^{-1}\mathbf{B}^T\mathcal{P}U \end{aligned}$$

where $\mathcal{K}$ is the functional gain and $\mathcal{P}$ is the solution to the algebraic Riccati equation given as

$$(\mathcal{A} + \alpha_s I)^T\mathcal{P} + \mathcal{P}(\mathcal{A} + \alpha_s I) - \mathcal{P}\mathcal{B}\mathcal{R}^{-1}\mathcal{B}^T\mathcal{P} + \mathcal{Q} = 0$$

With the optimal control variables determined we can now use them to solve the nonlinear system as

$$\begin{aligned} \frac{\partial}{\partial t}U &= \mathcal{A}U + \mathcal{F} + \mathcal{G}(U) + \mathcal{B}\alpha, \qquad t > 0 \\ &= \mathcal{A}U + \mathcal{F} + \mathcal{G}(U) - \mathcal{B}(\mathcal{K}U) \\ &= (\mathcal{A} - \mathcal{B}\mathcal{K})U + \mathcal{F} + \mathcal{G}(U) \end{aligned}$$

where $\mathcal{G}(U)$ is a nonlinear operator which incorporates all the nonlinear terms that are omitted from the linear model.

Alternatively, the gains can be solved using the Chandrasekhar equations which are derived by taking the time derivative of the Riccati equation. For more details on this derivation, see [54]. The Chandrasekhar equations are given as

$$\begin{aligned} -\dot{\mathcal{K}}(t) &= \mathcal{R}^{-1}\mathcal{B}^T\mathcal{L}^T(t)\mathcal{L}(t) \\ -\dot{\mathcal{L}}(t) &= \mathcal{L}(t)[(\mathcal{A}+\alpha_s) - \mathcal{B}\mathcal{K}(t)] \end{aligned}$$

where the final values are known; $\mathcal{K}(T) = 0$, $\mathcal{L}(T) = \mathcal{C}$. These equations are integrated backwards in time until the steady state solution for $\mathcal{K}(t)$ is reached.

As can been seen from the above equations, the solution of the Chandrasekhar equations is more efficient and should yield the same results as the Riccati method due to the nature of it's derivation. There are situations where each method has it's problems, for instance the Riccati equations involve finding $(N^2+N)/2$ matrix unknowns, where N is the number of degrees of freedom or the size of the $\mathcal{A}$ matrix. The Riccati method grows quadratically and quickly becomes unmanageable for large N or finer computational grids. However, the Chandrasekhar equations involve the calculation of $N(m+p)$ matrix unknowns where m is the number of controls and p is the number of outputs. The calculations then grow linearly with increasing degrees of freedom N. Alternatively, the Chandrasekhar equations are integrated backwards in time until a steady state is reached. The approach to this backward integration may greatly impact the performance of the simulation[28].

The above linear and nonlinear models are simulated without control over the grid given in Fig. 6.4 with the inlet condition $f(x, y)$ shown in Fig. 6.2. The results of the uncontrolled simulations can be compared in Fig. 6.5. As expected the linear model is not as accurate as the nonlinear model, but it is accurate enough to generate the control inputs that drive the nonlinear system as will be shown shortly. To test the efficacy of the control variables we test each control variable one at a time. Figure 6.6 shows the results of this test.

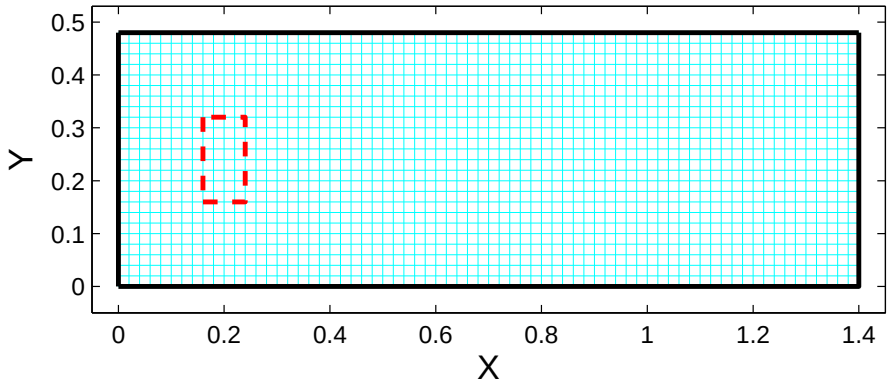
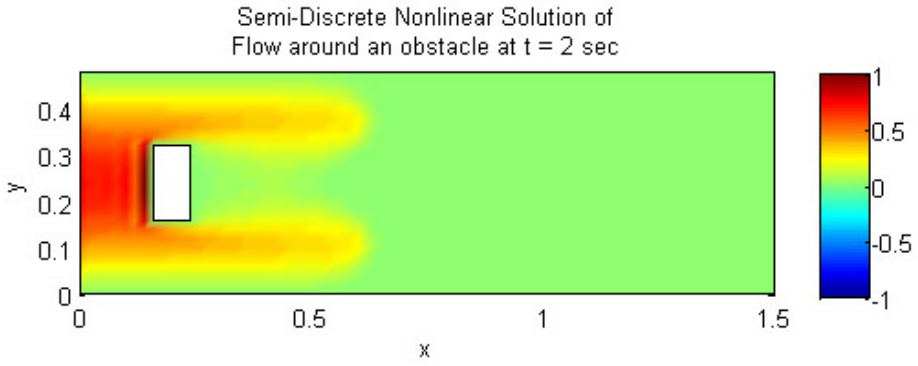
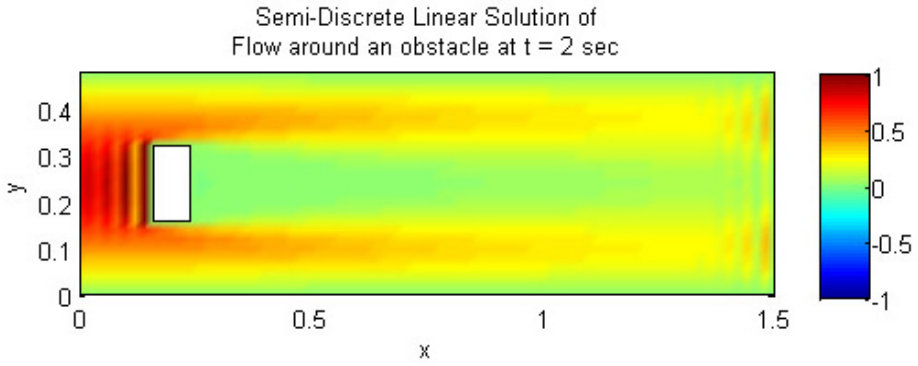


Figure 6.4. Grid used for channel flow problem.

The results of the Riccati and Chandrasekhar optimization methods are shown in the following figures. The flow solution given in Fig. 6.7 demonstrates the efficacy of the control schemes in modifying the downstream flow conditions. Our goal in this situation was to use controls on the obstacles surface to influence the flow downstream between $x = 0.5$ to 0.75 and $y = 0.16$ to 0.32 (the same height as the obstacle). The cost function was set up to drive this region to zero, i.e., $U(t, x, y) \Rightarrow 0$. The control schemes are compared to the uncontrolled case in Fig. 6.7, it is obvious that there is some influence due to the controls and the two control methods produce very similar



(a)

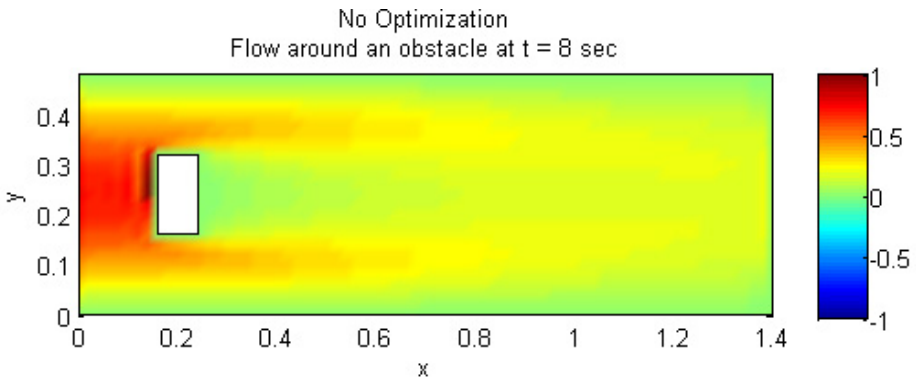


(b)

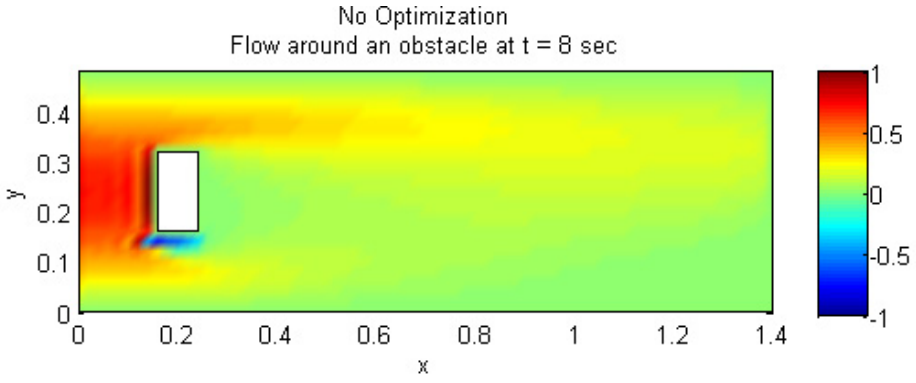
Figure 6.5. Comparison of flow solutions using (b) linear and (a) nonlinear models.

results.

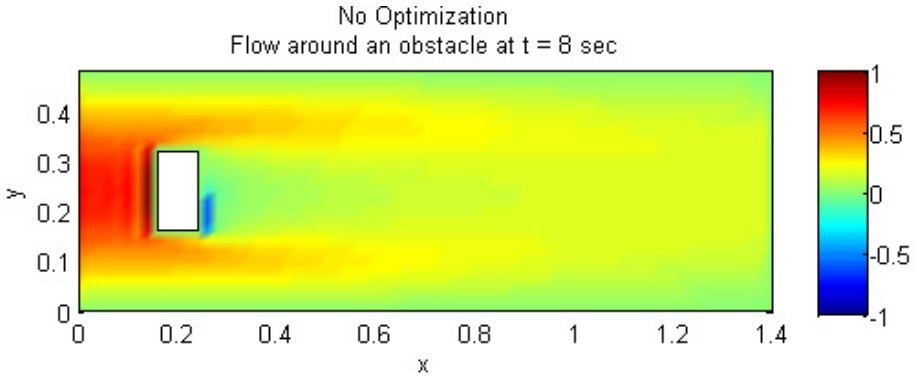
In order to show a more qualitative result of the control influence, we decided to look at the norm of the flow value $\|U(t,x,y)\|$ in the design region as a function of time. Figure 6.8 demonstrates these results. It is now clear that the optimal control schemes do indeed have an effect on the downstream flow; however, we had hoped to see no flow in this region. The control influence is a factor of several different inputs such as the values of $\mathcal{R}$, $\mathcal{C}$ and ϕ. Adjusting these values greatly impacts the ability of the control to achieve the desired effect. For instance, by increasing



(a)

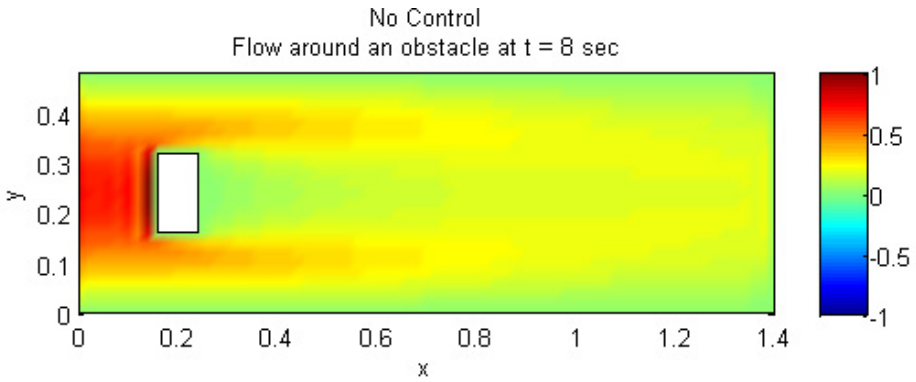


(b)

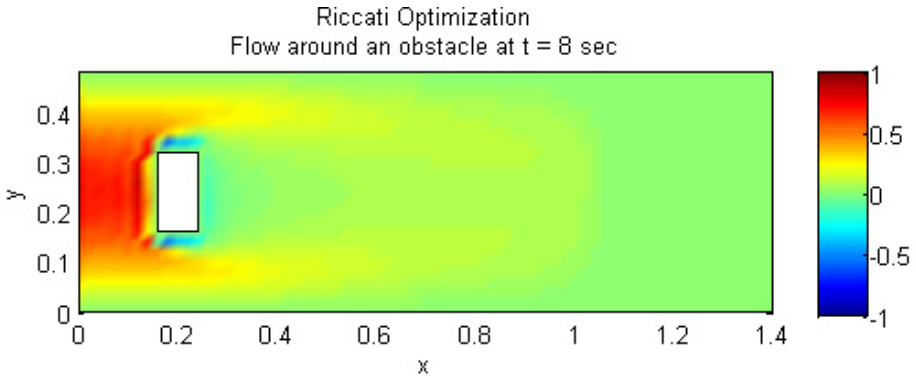


(c)

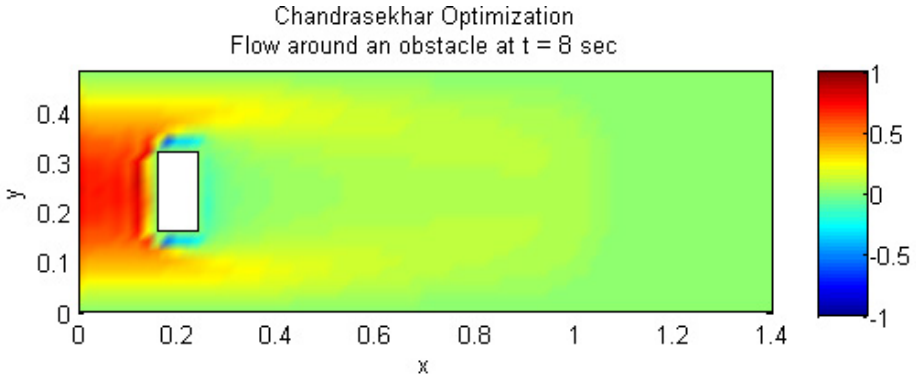
Figure 6.6. Test of manual control inputs on three different control surfaces; (a) $\Gamma_{2,1}$, (b) Γ_1 and (c) $\Gamma_{4,1}$.



(a)



(b)



(c)

Figure 6.7. Comparison of the uncontrolled channel flow given in (a) and optimal control strategies; (b) Riccati control, (c) Chandrasekhar control.

the values of the $\mathcal{R}$ matrix we are essentially putting a greater cost on the use of controls, the system will decrease the control amount in order not to have the cost function blow up. Also, by decreasing the values of the $\mathcal{C}$ matrix (recall $\mathcal{Q} = \mathcal{C}^T\mathcal{C}$) we are essentially decreasing the importance of influencing the flow in the control region in the cost function, thereby decreasing the amount of control needed to minimize the cost function. As well, by decreasing ϕ we are decreasing the controls effectiveness.

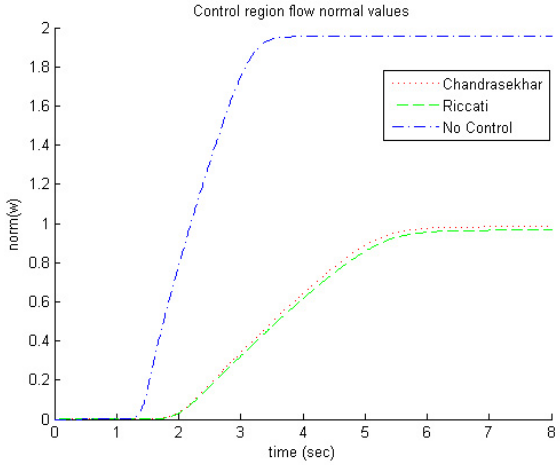


Figure 6.8. $\|U\|$ inside control region.

The results may also be a product of implementing the gain calculation as a constant, calculated at the initial time of the simulation. If we adjust the gain calculation to vary with time we may see the desired results.

Figures 6.9 and 6.10 show the histories of the controls for the Riccati and Chandrasekhar optimal control schemes. These figures clearly shows the similarity in the results between the Riccati and Chandrasekhar methods.

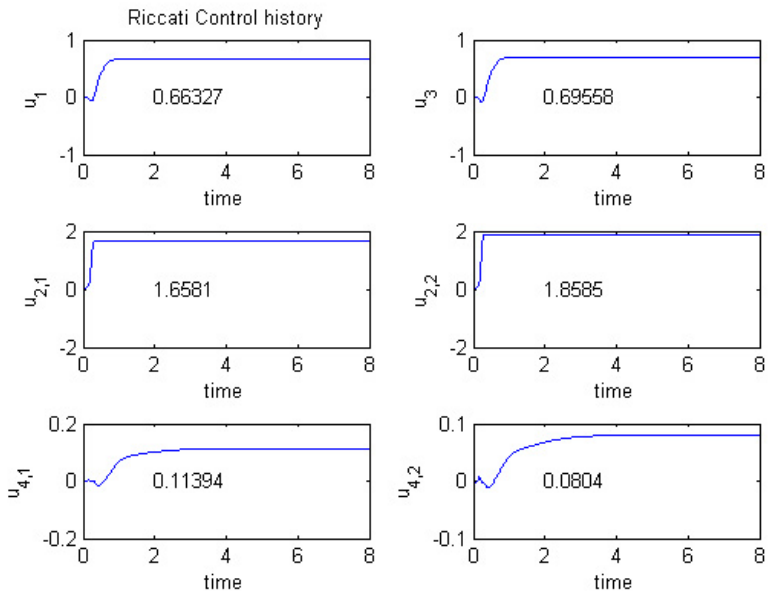


Figure 6.9. Riccati control history.

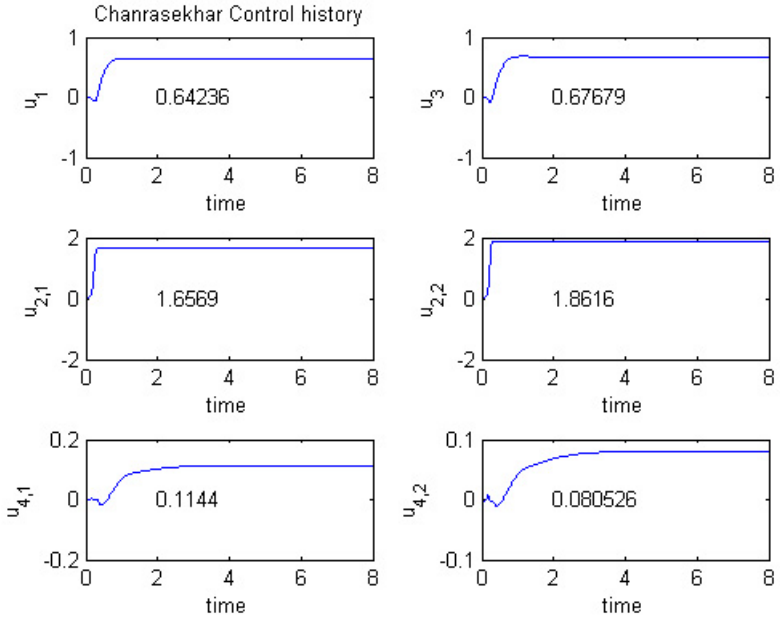


Figure 6.10. Chandrasekhar control history.

6.3 Boundary control using POD based reduced order model

The POD model reduction technique given for the inlet optimization is applied to the optimal control problem of the channel flow. There are minor changes for this application when compared with that derived for the Euler equations. In this case, we are generating the reduced order model for the Burgers equation which is a single PDE not the system of PDEs as we had for the Euler equation. Another change is that we are now considering an unsteady system. Considering that we wish to implement an optimal control technique on a dynamic system we need the reduced model to accurately capture all of those dynamics. The basis functions for the POD model are derived from snapshots of the flow solution at various instances in time and for various values and combinations of the control variables.

6.3.1 Weak form of the governing equation

Restating the problem at hand, let us rewrite the governing equation as

$$\frac{\partial}{\partial t}U(t,x,y) + \nabla \cdot D(U) = \frac{1}{Re}\nabla^2 U(t,x,y) \tag{6.5}$$

where $D(U)$ is given as

$$D(U) = \left[\begin{array}{cc} K_1 U^2(t,x,y), & K_2 U^2(t,x,y) \end{array} \right]^T$$

The flow boundary conditions remain the same as in the previous problem; we restate them here for convenience

$$U(t,\Gamma_{in}) = f(y), \quad U_x(t,\Gamma_{out}) = 0, \quad U(t,\Gamma_{bottom}) = 0, \quad U(t,\Gamma_{top}) = 0 \tag{6.2}$$

The control input on the obstacle are slightly varied in this problem, now we will consider controls only on the top and bottom of the obstacle, Γ_1 and Γ_3

$$U_1(t,\Gamma_1) = \alpha_1\phi_1, \qquad U_3(t,\Gamma_3) = \alpha_3\phi_3, \tag{6.6}$$

The initial condition is unchanged

$$U(0,x,y) = U_o(x,y) \in \Omega \tag{6.4}$$

Following the methodology presented in [56] we first generate the weak form of the governing equation by taking the inner product of both sides of Eq.(6.5) with the i^{th} POD mode (see Eq.(2.14))

$$\int_\Omega U(t,x,y)\Phi_i(x,y)d\Omega + \int_\Omega \nabla D(U)\Phi_i(x,y)d\Omega = \frac{1}{Re}\int_\Omega \nabla^2 U(t,x,y)\Phi_i(x,y)d\Omega$$

Using Green's identities the spatial derivative terms can be rewritten to involve domain and boundary integrals

$$\begin{aligned}\int_\Omega \frac{\partial U(t,x,y)}{\partial t}\Phi_i(x,y)d\Omega &= \int_\Omega D(U)\cdot\nabla\Phi_i(x,y)d\Omega - \int_\Gamma (D(U)\cdot\mathbf{n})\Phi_i(x,y)ds \\ -\frac{1}{Re}\int_\Omega \nabla U(t,x,y)\cdot\nabla\Phi_i(x,y)d\Omega &+ \frac{1}{Re}\int_\Gamma (\nabla U(t,x,y)\cdot\mathbf{n})\Phi_i(x,y)ds\end{aligned} \tag{6.7}$$

where $\mathbf{n}$ is the outward facing surface normal. Implementing the boundary conditions on the boundary integrals we can greatly simplify this equation. Consider the second boundary integral

$$
\begin{aligned}
&\int_{\Gamma} (\nabla U(t,x,y)\cdot \mathbf{n})\Phi_i(x,y)ds \\
&= \int_{\Gamma_{top}} \left(\frac{\partial U(t,\Gamma_{top})}{\partial y}\right)^{0} \Phi_i(\Gamma_{top})dx - \int_{\Gamma_{bottom}} \left(\frac{\partial U(t,\Gamma_{bottom})}{\partial y}\right)^{0} \Phi_i(\Gamma_{bottom})dx \\
&+ \int_{\Gamma_{out}} \left(\frac{\partial U(t,\Gamma_{out})}{\partial x}\right)^{0} \Phi_i(\Gamma_{out})dy - \int_{\Gamma_{in}} \left(\frac{\partial U(t,\Gamma_{in})}{\partial x}\right) \Phi_i(\Gamma_{in})dy \\
&+ \int_{\Gamma_3} \left(\frac{\partial U(t,\Gamma_3)}{\partial y}\right) \Phi_i(\Gamma_3)dx - \int_{\Gamma_1} \left(\frac{\partial U(t,\Gamma_1)}{\partial y}\right) \Phi_i(\Gamma_1)dx \\
&+ \int_{\Gamma_4} \left(\frac{\partial U(t,\Gamma_4)}{\partial x}\right)^{0} \Phi_i(\Gamma_4)dy - \int_{\Gamma_2} \left(\frac{\partial U(t,\Gamma_2)}{\partial x}\right)^{0} \Phi_i(\Gamma_2)dy
\end{aligned}
$$

Similarly for the first boundary integral

$$
\begin{aligned}
&\int_{\Gamma} (D(U)\cdot \mathbf{n})\Phi_i(x,y)ds \\
&= \int_{\Gamma_{top}} \left(U^2(t,\Gamma_{top})\right)^{0} \Phi_i(\Gamma_{top})dx - \int_{\Gamma_{bottom}} \left(U^2(t,\Gamma_{bottom})\right)^{0} \Phi_i(\Gamma_{bottom})dx \\
&+ \int_{\Gamma_{out}} \left(U^2(t,\Gamma_{out})\right) \Phi_i(\Gamma_{out})dy - \int_{\Gamma_{in}} \left(U^2(t,\Gamma_{in})\right)^{f(y)} \Phi_i(\Gamma_{in})dy \\
&+ \int_{\Gamma_3} \left(U^2(t,\Gamma_3)\right)^{0} \Phi_i(\Gamma_3)dx - \int_{\Gamma_1} \left(U^2(t,\Gamma_1)\right)^{0} \Phi_i(\Gamma_1)dx \\
&+ \int_{\Gamma_4} \left(U^2(t,\Gamma_4)\right)^{0} \Phi_i(\Gamma_4)dy - \int_{\Gamma_2} \left(U^2(t,\Gamma_2)\right)^{0} \Phi_i(\Gamma_2)dy
\end{aligned}
$$

The partial derivatives on the boundaries can be discretized such that the control inputs and inflow condition are incorporated explicitly, for example

$$\frac{\partial}{\partial y}U(t,\Gamma_1) \approx \frac{\alpha_1\psi_1 - U(t,\Gamma_1 - h)}{h}$$
$$\frac{\partial}{\partial y}U(t,\Gamma_3) \approx \frac{U(t,\Gamma_3 + h) - \alpha_3\psi_3}{h}$$
$$\frac{\partial}{\partial x}U(t,\Gamma_{in}) \approx \frac{U(t,\Gamma_{in} + h) - f(y)}{h}$$

In the simulation, we set $K_1 = 1$ and $K_2 = 0$ for a positive inflow conditions with convection from left to right and the grid is discretized such that $\Delta x = \Delta y = h$. Substituting these terms and the reduced order model, $U = \sum_j a_j\Phi_j$ into Eq.(6.7) the resulting equation is given as

$$\frac{\partial}{\partial t}a = -\frac{1}{hRe}\left[\int_{\Gamma_3}\Phi_{j\Gamma_3+h}\Phi_{i\Gamma_3}dx + \int_{\Gamma_1}\phi_{j\Gamma_1-h}\Phi_{i\Gamma_1}dx + \int_{\Gamma_{in}}\phi_{j\Gamma_{in}+h}\Phi_{i\Gamma_{in}}dy\right.$$
$$\left.+h\int_{\Omega}\nabla\Phi_j\cdot\nabla\Phi_i d\Omega\right]a + \int_{\Omega}\left(\sum_{j=1}^{M}a_j\Phi_j\right)^2\frac{\partial\Phi_i}{\partial x}d\Omega - \int_{\Gamma_{out}}\left(\sum_{j=1}^{M}a_j\Phi_{j\Gamma_{out}}\right)^2\Phi_{i\Gamma_{out}}dy$$
$$+\frac{1}{hRe}\left[\int_{\Gamma_1}\alpha_1\psi_1\Phi_{i\Gamma_1}dx + \int_{\Gamma_3}\alpha_3\psi_3\Phi_{i\Gamma_3}dx\right] + \int_{\Gamma_{in}}\left(f^2(y) + \frac{1}{hRe}f(y)\right)\Phi_{i\Gamma_{in}}dy$$

This gives the final reduced order model of the form

$$\dot{a} = \hat{\mathcal{A}}a + \hat{\mathcal{B}}\alpha + \hat{\mathcal{G}}(a) + \hat{\mathcal{F}}$$

where

$$\hat{\mathcal{A}} = -\mu \left[\int_{\Gamma_3} \Phi_{j\Gamma_3+h} \Phi_{i\Gamma_3} dx + \int_{\Gamma_1} \phi_{j\Gamma_1-h} \Phi_{i\Gamma_1} dx + \int_{\Gamma_{in}} \phi_{j\Gamma_{in}+h} \Phi_{i\Gamma_{in}} dy + h \int_{\Omega} \nabla \Phi_j \cdot \nabla \Phi_i d\Omega \right]_{M \times M}$$

$$\hat{\mathcal{B}} = \mu \begin{bmatrix} \int_{\Gamma_1} \psi_1 \Phi_{1\Gamma_1} dx & \int_{\Gamma_3} \psi_3 \Phi_{1\Gamma_3} dx \\ \vdots & \vdots \\ \int_{\Gamma_1} \psi_1 \Phi_{M\Gamma_1} dx & \int_{\Gamma_3} \psi_3 \Phi_{M\Gamma_3} dx \end{bmatrix}_{1 \times M}$$

$$\hat{\mathcal{G}}(a) = \begin{bmatrix} \int_{\Omega} \left(\sum_{j=1}^{M} a_j \Phi_j \right)^2 \frac{\partial}{\partial x} \Phi_1 d\Omega - \int_{\Gamma_{out}} \left(\sum_{j=1}^{M} a_j \Phi_{j\Gamma_{out}} \right)^2 \Phi_{1\Gamma_{out}} dy \\ \vdots \\ \int_{\Omega} \left(\sum_{j=1}^{M} a_j \Phi_j \right)^2 \frac{\partial}{\partial x} \Phi_M d\Omega - \int_{\Gamma_{out}} \left(\sum_{j=1}^{M} a_j \Phi_{j\Gamma_{out}} \right)^2 \Phi_{M\Gamma_{out}} dy \end{bmatrix}_{1 \times M}$$

$$\hat{\mathcal{F}} = \begin{bmatrix} \int_{\Gamma_{in}} \left(f^2(y) + \frac{1}{hRe} f(y) \right) \Phi_{1\Gamma_{in}} dy \\ \vdots \\ \int_{\Gamma_{in}} \left(f^2(y) + \frac{1}{hRe} f(y) \right) \Phi_{M\Gamma_{in}} dy \end{bmatrix}_{1 \times M}$$

The initial condition a_o is generated by projecting the initial flow condition U_o into the POD basis functions.

6.3.2 Generation of a POD based reduced order model for the 2-D channel flow

The model is generated using snapshots of the flow solution for various control inputs and at various time steps. Following [62] the snapshots are generated for control inputs given as

$$\begin{aligned} \alpha_1(t) &= \beta \sin(0.25t^2) & \alpha_3(t) &= 0 \\ \alpha_1(t) &= 0 & \alpha_3(t) &= \beta \sin(0.25t^2) \\ \alpha_1(t) &= \beta \sin(0.25t^2) & \alpha_3(t) &= \beta \sin(0.25t^2) \end{aligned}$$

where the values of β are $-3,\ -2,\ -1$. The snapshots are taken every 0.05 seconds from $t = 0$ to 10 seconds. The eigenvalues of the correlation matrix are given in Fig 6.11, every tenth eigenvalue is plotted to better visualize the data. In order to capture 99.99% of the energy in the system we maintain 17 basis functions. The first 8 basis functions are given in Fig. 6.12

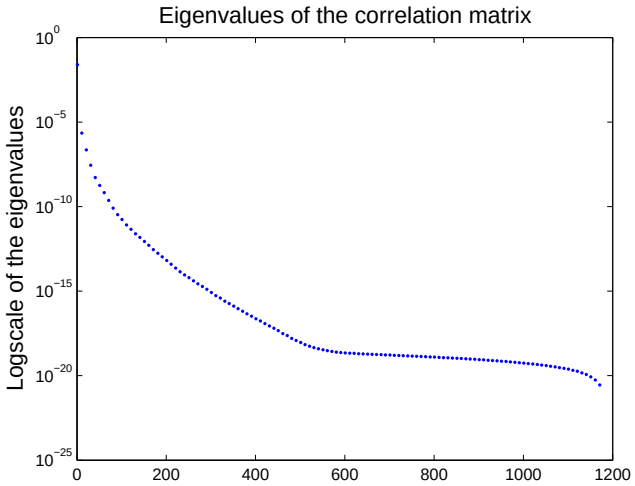


Figure 6.11. Eigenvalues of the correlation matrix for the channel flow.

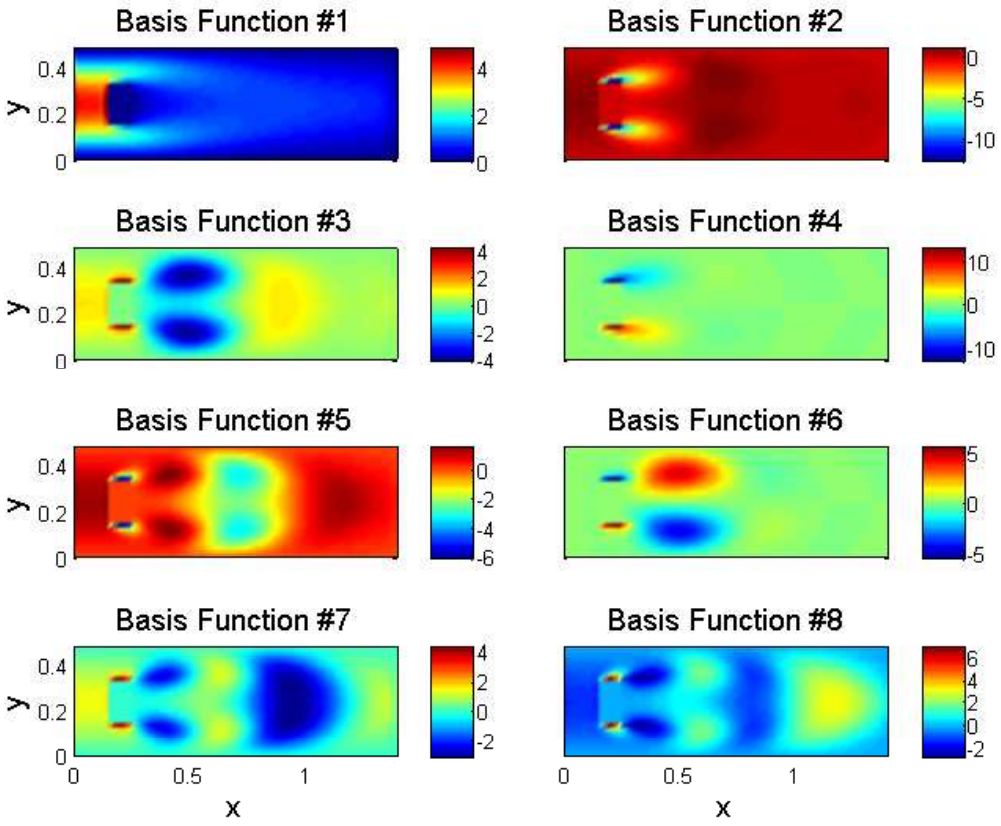


Figure 6.12. The first 8 basis functions for the POD model.

6.3.3 Validation of the POD Model

To validate the POD model we can compare the inputs on the control boundaries with prescribed control inputs where

$$\sum_{i=1}^{M} a_i(t)\Phi_i(\Gamma_1) \approx U(t,\Gamma_1) = \alpha_1(t)$$

$$\sum_{i=1}^{M} a_i(t)\Phi_i(\Gamma_3) \approx U(t,\Gamma_3) = \alpha_3(t)$$

The test control input to the system is given as

$$\alpha_1(t) = \sin\left(\frac{3}{4}\pi t\right), \qquad \alpha_3(t) = \sin\left(\frac{3}{2}\pi t\right)$$

Figure 6.13 shows the results of this validation, notice the good correlation between the reduced order model and prescribed control input. Additional validation can be achieved through comparison of the POD temporal coefficient and equivalent coefficients derived by projecting the full order model onto the basis functions at each time step. Utilizing the same control inputs as those described in the previous validation case, the temporal coefficients are given in Fig. 6.14

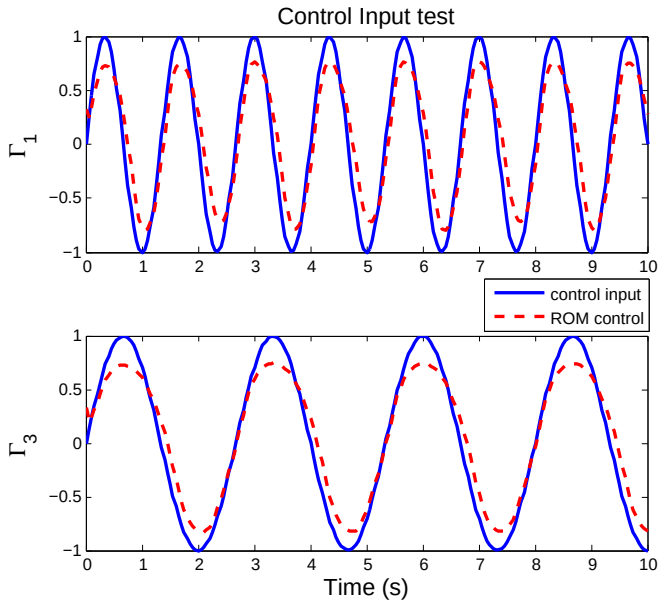


Figure 6.13. POD model boundary control accuracy.

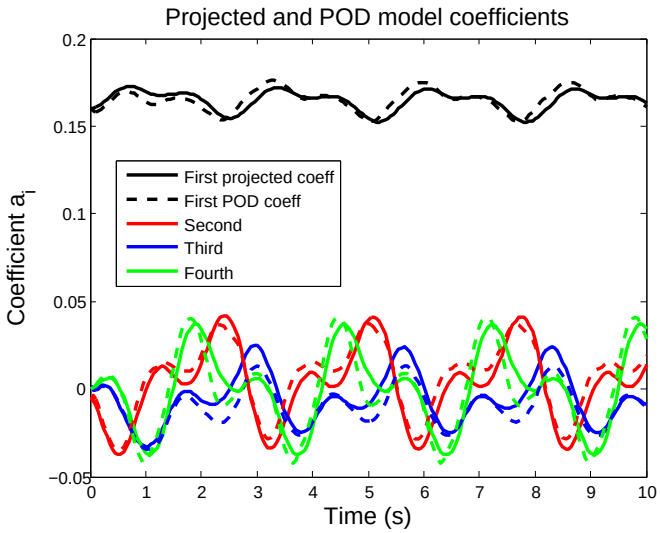


Figure 6.14. POD and projected temporal coefficients.

6.3.4 Boundary control utilizing the POD model

We now consider the use of the reduced order model in the development of a control routine to be applied to the full order model. The control routine is developed from the linearized reduced order model given as

$$\dot{a}(t) = \hat{\mathcal{A}}a + \hat{\mathcal{B}}\alpha$$

$$a(0) = a_o$$

The control problem addressed here is a tracking problem such that flow will track some desired reference signal. In this case the reference signal is constant in time and is denoted as $U_{ref}(x, y)$. Projecting this reference signal onto the POD basis functions

results in a set of tracking coefficients a_{ref} for the POD model. The dynamics of the tracking control problem are given as follows

$$\begin{bmatrix} \dot{a} \\ \dot{a}_{ref} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{A}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ a_{ref} \end{bmatrix} + \begin{bmatrix} \hat{\mathcal{B}} \\ 0 \end{bmatrix} \alpha \tag{6.8}$$

$$\dot{\mathcal{X}} = \bar{\mathcal{A}}\mathcal{X} + \bar{\mathcal{B}}\alpha \tag{6.9}$$

where the augmented state vector is given as

$$\mathcal{X} = \begin{bmatrix} a \\ a_{ref} \end{bmatrix}, \qquad \mathcal{X}_o = \begin{bmatrix} a_o \\ a_{ref} \end{bmatrix}$$

The cost function given in the same format to the previous channel flow case were we again use an α_s shifted LQR

$$J_\alpha(U_0, \alpha) = \int_0^\infty \{(a - a_{ref})^T \mathcal{Q}(a - a_{ref}) + \alpha^T \mathcal{R}\alpha\} e^{2\alpha_s t} dt \tag{6.10}$$

The optimal control problem is to minimize the given cost function, Eq.6.10 subject to the constraints, Eq. 6.9. Again we will use the Riccati equation to find the optimal feedback gains, $\mathcal{K}$.

$$\begin{aligned} \alpha_{opt} &= -\mathcal{K}\mathcal{X} \\ &= -\begin{bmatrix} \mathcal{K}_1 & \mathcal{K}_2 \end{bmatrix} \mathcal{X} \\ &= -\begin{bmatrix} \mathcal{R}^{-1}\hat{\mathcal{B}}^T\mathcal{P}_1 & \mathcal{R}^{-1}\hat{\mathcal{B}}^T\mathcal{P}_2 \end{bmatrix} \mathcal{X} \end{aligned}$$

where $\mathcal{P}_1$ is the solution to the algebraic Riccati equation

$$(\hat{\mathcal{A}} + \alpha_s I)^T \mathcal{P}_1 + \mathcal{P}_1(\hat{\mathcal{A}} + \alpha_s I) - \mathcal{P}_1 \hat{\mathcal{B}} \mathcal{R}^{-1} \hat{\mathcal{B}}^T \mathcal{P}_1 + \mathcal{Q} = 0$$

and $\mathcal{P}_2$ satisfies the equation

$$\left[(\hat{\mathcal{A}} + \alpha_s I)^T - \mathcal{P}_1 \hat{\mathcal{B}} \mathcal{R}^{-1} \hat{\mathcal{B}}^T \right] \mathcal{P}_2 = \mathcal{Q}$$

The gains computed from the linear model are used in feedback control of the full nonlinear system to generate the optimal temporal coefficients for the reduced order model

$$\begin{aligned} \dot{\mathcal{X}} &= \left(\bar{\mathcal{A}} - \bar{\mathcal{B}}\mathcal{K}\right)\mathcal{X} + \begin{bmatrix} \hat{\mathcal{G}}(a) & 0 \end{bmatrix}^T + \begin{bmatrix} \hat{\mathcal{F}} & 0 \end{bmatrix}^T \\ \mathcal{X}(0) &= \mathcal{X}_o \end{aligned}$$

6.3.5 Closed-loop control results for set point tracking

The reference signal utilized in this case is a steady-state flow solution for a flow with $Re = 50$. The signal is projected onto the POD basis functions to generate a set of reference coefficients a_{ref}. The full nonlinear flow solution and the POD flow solution corresponding to the projected coefficients are given in Fig. 6.15. Notice the POD model is a close match to the reference signal even though it is not representative of the original snapshot data used to generate the reduced order model.

The uncontrolled system is given in Fig. 6.16 for visual comparison with the final controlled system in Fig. 6.17. A more accurate comparison is done through the norm of the flow for some downstream region. In this case we take the norm of all the flow values down stream of $x = 0.6$. We compare the norm $\|U\| - \|U_{ref}\|$ for both

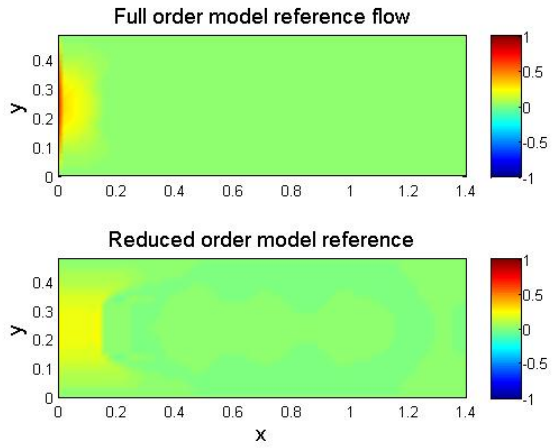


Figure 6.15. Reference signal for full order model and projected reduced order model.

the controlled and uncontrolled cases, the results are given in Fig. 6.18. As can be seen in the figure, the feedback control is quite capable of bringing the initial flow to the desired state. Changing the LQR parameters; the weight matrices $\mathcal{Q}$ and $\mathcal{R}$ would influence the rate and accuracy of the tracking.

6.3.6 Closed-loop control results for time varying tracking

In this case we use a time varying reference signal generated using a sinusoidal input on the top and bottom boundaries of th obstacle. Again, the signal is projected onto the POD basis functions to generate the set of reference coefficients a_{ref}. In order to validate the reduced order reference signal with the original full nonlinear reference signal we plotted the downstream norm values of each and calculated the error between them. The results of this validation are given in Fig. 6.19. As we saw in the set point tracking problem, the POD model is still a close match to the reference signal. In this case the reference inputs are representative of the snapshot data and a close correlation is expected.

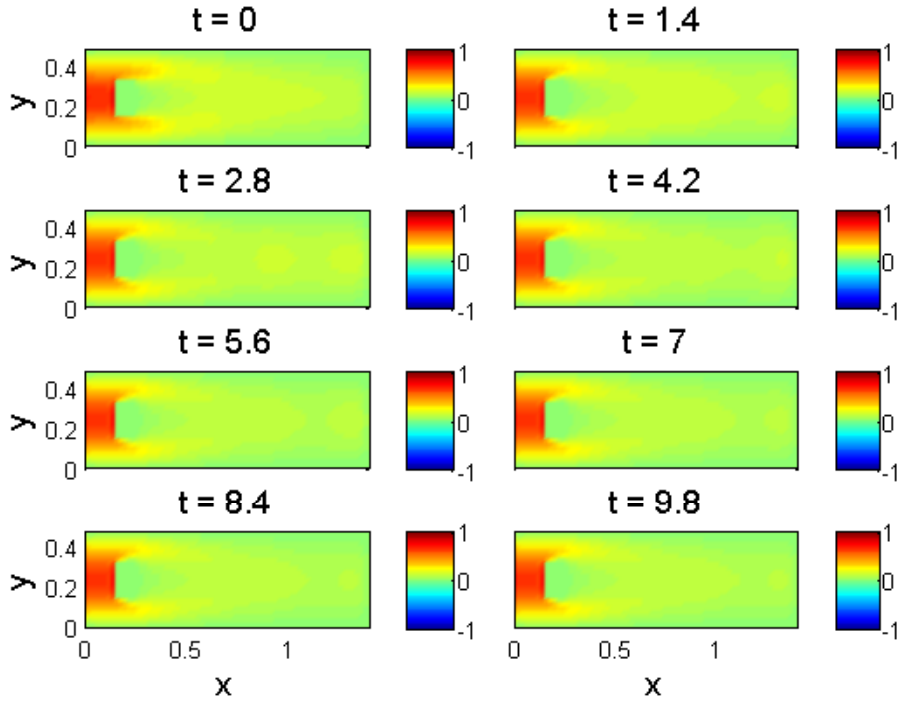


Figure 6.16. Uncontrolled POD model.

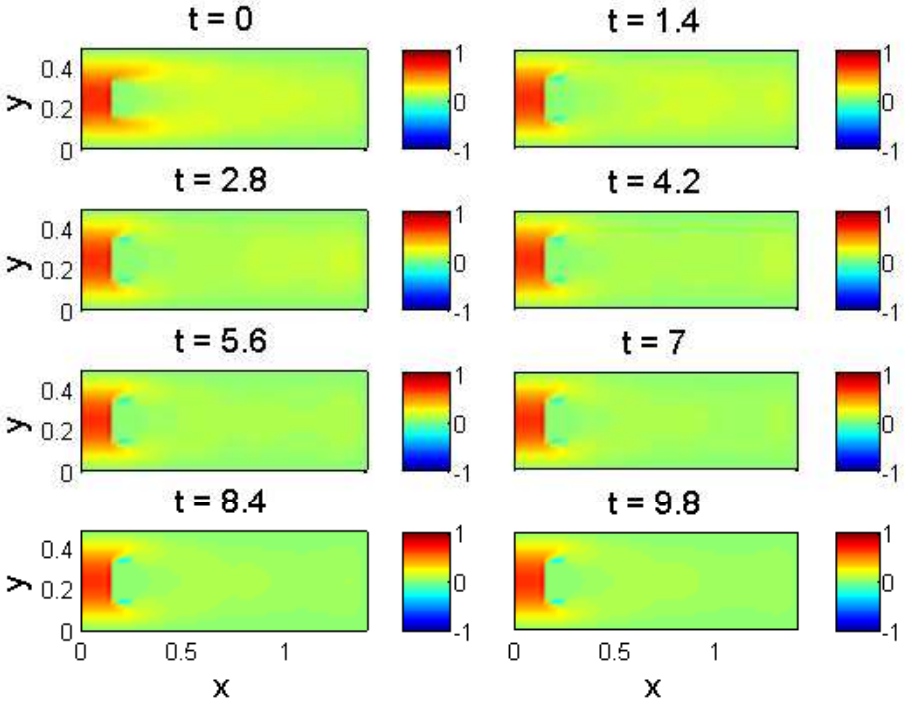


Figure 6.17. POD model with control.

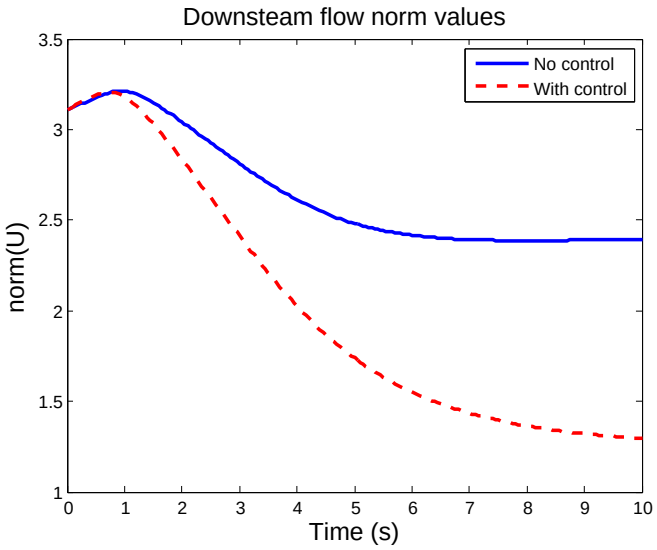


Figure 6.18. $\|U\|$ downstream of $x = 0.6$.

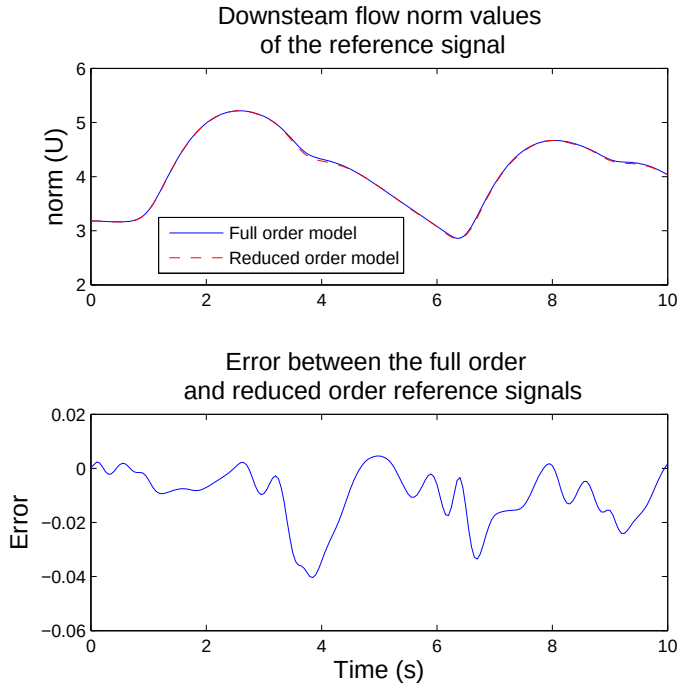


Figure 6.19. Validation of the reference signal for the time varying tracking problem.

The results of the control implementation are given in Fig. 6.20. Again we compare the norm values of the flow downstream of $x = 0.6$. Various values of the state weight matrix $\mathcal{Q}$ were chosen to illustrate its effect. As seen in the figure, higher values of $\mathcal{Q}$ resulted in better tracking of the reference signal. A final step is to compare the control inputs resulting from the LQR implementation on the reduced order model to those used to generate the reference signal. Figure 6.21 demonstrates this result. Here we see a strong correlation between the control inputs and the original reference signal inputs.

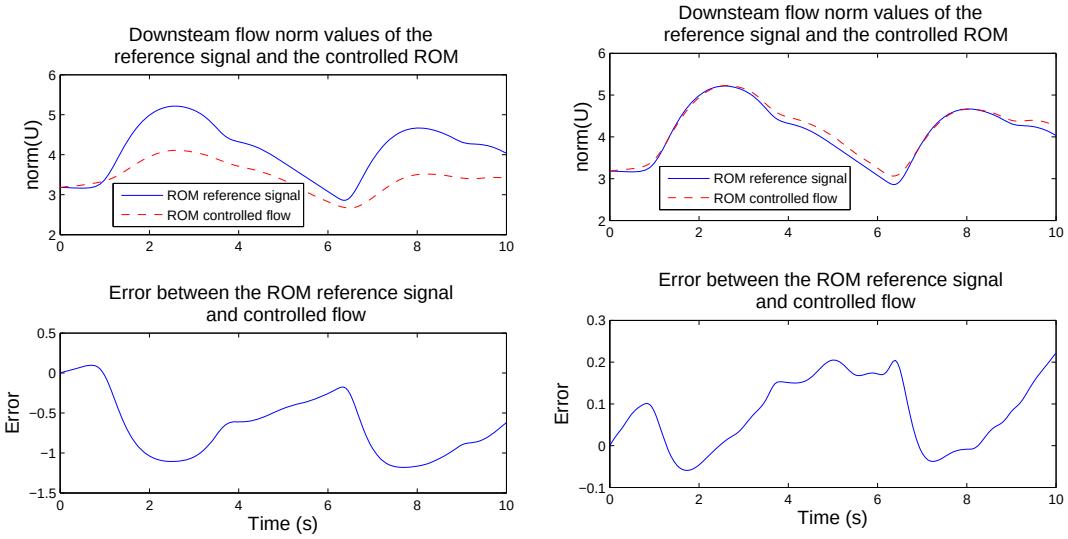


Figure 6.20. Comparison of the downstream norm values for the reference signal and the controlled flow for state weight values of (a) $\mathcal{Q} = 4000$ and (b) $\mathcal{Q} = 200000$.

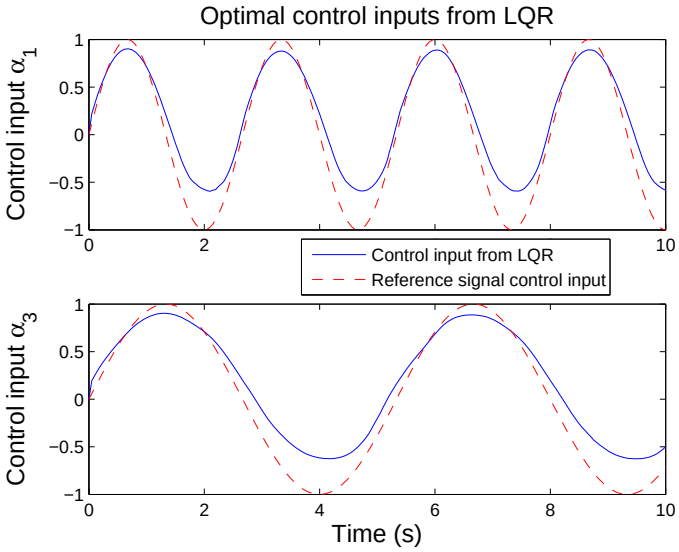


Figure 6.21. Comparison of the reduced order model control inputs with the control inputs used to generate the reference signal.

CHAPTER 7

CONCLUDING REMARKS

7.1 Optimization and optimal control using adjoint/costate variables

The relationship between the adjoint methods of optimization and the Riccati solution of the LQR optimal control problem have been laid out in detail. There are clear parallels between the steady-state optimization problem and optimal control applied to unsteady systems that can be seen through the use of the costate/adjoint variables.

Adjoint methods have been investigated for gradient calculations of 2-D double ramp inlet optimization. For the double ramp inlets considered here, the adjoint gradient calculation does not produce ideal results. The gradient vectors suggest possible local stable regions near the known optimal values but their regions of convergence are quite small. This method may produce better results with a different cost function. In this research the cost function was defined at the exit plane of the inlet and the control input was defined on lower wall boundary. There is a section of the lower wall inside the throat that separates the control surface from the exit plane where the cost function is evaluated. This decoupling of the control surface from cost function may be a hindrance to the adjoint gradient calculation method. Future work should investigate the use of a different cost function defined either on the control boundary or through out the domain. Another potential improvement may be seen with the implementation of a higher-order accurate flow solver.

Optimal control of a 2-D channel flow was investigated through the implementation of the Riccati equation on an LQR problem. As was demonstrated in the

optimization problem, the results of the optimal control solution are dependent on the chosen cost function. Unlike the adjoint optimization problem, the feedback gains are not difficult to recalculate for changes in the cost function.

7.2 Reduced order models

Optimization and optimal control problems were solved using POD reduced order models with great success. The POD method for reduced order modeling proved to be very robust. In the full inlet study we used 504 snapshots whereas in the cowl inlet study we used 88 snapshots. Both models yielded good results in their respective optimization problems. The majority of the energy in all the systems studied was captured in a very small number of basis functions.

The reduced order models for the optimization problem where generated using a projection-free sub-optimization technique based on system identification theory. This method proved to be very effective and eliminated complexities associated with traditional projection techniques.

Future studies will investigate possible projection-free techniques for calculating the weights in the optimal control problem. The difficulties here are in the application of linear control techniques which require the problem to be given in Cauchy form. Traditional projection techniques naturally give a system of this type, but when applied to more complex governing equations the projections become quite cumbersome. Ideally the projection free technique could be written in the desired Cauchy form.

APPENDIX A

DERIVATION OF THE QUASI-1-D EULER EQUATIONS

The governing equations for the quasi-1-D duct are derived from the full 1-D system of equations beginning with the continuity equation. Consider a duct such as the one shown in Fig. 3.1. The flow variables are the velocity $\mathbf{u}$, density ρ and pressure p, and the duct shape $s = s(x)$. Given a small control volume of area ν and with a boundary S, the quasi 1-D continuity equation is

$$\begin{aligned} & \frac{\partial}{\partial t}\int_{\nu} \rho d\nu + \oint_{S} \rho \mathbf{u} \cdot d\mathbf{S} = 0 \\ \Rightarrow \quad & \frac{\partial}{\partial t}\int_{\nu} \rho d\nu + \oint_{S(in)} \rho \mathbf{u} \cdot d\mathbf{S} + \oint_{S(out)} \rho \mathbf{u} \cdot d\mathbf{S} + \oint_{S(x)} \rho \mathbf{u} \cdot d\mathbf{S} = 0 \end{aligned}$$

Restricting the solution to steady-state and imposing the wall condition ($\mathbf{u} \cdot d\mathbf{S} = 0$), the following terms are eliminated.

$$\oint_{S(x)} \rho \mathbf{u} \cdot d\mathbf{S} = 0, \qquad \frac{\partial}{\partial t}\int_{\nu} \rho d\nu = 0$$

With $d\mathbf{S}$ defined as the outward normal being positive, the equation can be written as

$$\begin{aligned} \Rightarrow \quad & -s(in)\rho(in)u(in) + s(out)\rho(out)u(out) = 0 \\ \Rightarrow \quad & s(x)\rho(x)u(x) = \text{constant} \\ \Rightarrow \quad & \boxed{\frac{d}{dx}(\rho u s) = 0} \end{aligned} \tag{A.1}$$

Considering the same control volume, the quasi 1-D momentum equation is derived as follows

$$\frac{\partial}{\partial t}\int_{\nu} \rho \mathbf{u} d\nu + \oint_{S} (\rho \mathbf{u} \cdot d\mathbf{S})\mathbf{u} = -\oint_{S} p d\mathbf{S}$$

Again restricting the solution to steady-state and imposing the wall condition ($\mathbf{u} \cdot d\mathbf{S} = 0$), the equation simplifies to

$$\Rightarrow \quad \oint_{S(in)} (\rho \mathbf{u} \cdot d\mathbf{S})\mathbf{u} + \oint_{S(out)} (\rho \mathbf{u} \cdot d\mathbf{S})\mathbf{u} = -\oint_{S(in)} p d\mathbf{S} - \oint_{S(out)} p d\mathbf{S} + \oint_{S(x)} p d\mathbf{S}$$

$$\Rightarrow \quad -s(in)\rho(in)u^2(in) + s(out)\rho(out)u^2(out) = s(in)p(in) - s(out)p(out) + \oint_{S(x)} p d\mathbf{S}$$

$$\Rightarrow \quad \frac{d}{dx}(s(x)\rho(x)u^2(x)) + \frac{d}{dx}(s(x)p(x)) = p\frac{d}{dx}(s(x))$$

$$\Rightarrow \quad \boxed{\frac{d}{dx}(s\rho u^2) + \frac{d}{dx}(sp) - p\frac{d}{dx}(s) = 0} \tag{A.2}$$

For notational convenience, from here on we will denote $s(x)$ by s. The energy equation is similarly derived as follows

$$\frac{\partial}{\partial t}\int_{\nu} \rho E d\nu + \oint_{S} \rho E \mathbf{u} \cdot d\mathbf{S} = -\oint_{S} p\mathbf{u} \cdot d\mathbf{S}$$

which in steady-state can be written as

$$\Rightarrow \quad \oint_{S} \rho E \mathbf{u} \cdot d\mathbf{S} = -\oint_{S} p\mathbf{u} \cdot d\mathbf{S}$$

$$\therefore \quad \boxed{\frac{d}{dx}(s\rho E u) + \frac{d}{dx}(spu) = 0} \tag{A.3}$$

Assembling the conservation equation for mass Eq.(A.1), momentum Eq.(A.2), and energy Eq.(A.3) together gives the final system of equations for the duct

$$\begin{aligned} \frac{d}{dx}(s\rho u) &= 0 \\ \frac{d}{dx}(s\rho u^2 + sp) - p\frac{d}{dx}(s) &= 0 \\ \frac{d}{dx}(s\rho E u + spu) &= 0 \end{aligned}$$

where E is the total energy for an ideal gas:

$$E = \frac{p}{\rho(\gamma - 1)} + \frac{1}{2}u^2$$

The flow equations are written in conservative form as follows

$$R(\mathbf{U}, s) = \frac{d}{dx}(s\mathbf{F}) - \mathbf{P}\frac{ds}{dx} = 0$$

where $s(x)$ represent the height of the duct and the state variables, flux and pressure terms are given as follows

$$\mathbf{U} = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u(E + p/\rho) \end{bmatrix}, \qquad \mathbf{P} = \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix},$$

APPENDIX B

DERIVATION OF THE 2-D EULER EQUATIONS

A fluid in motion is governed by fundamental laws of classical physics; conservation of mass, Newton's second law, and the first law of thermodynamics. We will proceed through these principles one at a time to derive the flow equations that will be solved in our simulations.

B.1 Continuity equation

The conservation of mass is a fairly straightforward concept; for a given, fixed mass of fluid inside a volume element dV moving with the flow, the size and shape of the element may change but the mass inside does not. This gives us the continuity equation written in nonconservative form

$$\frac{D}{Dt}\int_V \rho dV = 0$$

or

$$\frac{D\rho}{Dt} + \rho\nabla \cdot V = 0$$

The same equation can be derived in conservative form by considering a given volume element fixed in space with a fluid moving through it with velocity $\mathbf{V}$, the mass inside the volume does not change with time:

$$\frac{\partial\rho}{\partial t} + \nabla \cdot (\rho\mathbf{V}) = 0 \tag{B.1}$$

B.2 Momentum equation

The conservation of momentum is derived from the application of Newton's second law $\mathbf{F} = m\mathbf{a}$, which says that the rate at which the momentum of a volume of fluid is changing is due to the sum of the external forces acting on the fluid element.

External forces can be placed in two different categories, body and surface forces. Body forces include those forces that '*act at a distance*' on the entire fluid volume such as gravity or electric fields. If $\mathbf{f}$ represents the resultant body force per unit mass, then the net external body force acting on the fluid volume is $\int_V \rho \mathbf{f} dV$. Surface forces act only on the surface of the fluid element and include pressure and shear forces. If $\boldsymbol{P}$ is a resultant stress tensor, then the net external surface force acting on the fluid surface S is $\int_S \boldsymbol{P} dS$.

$$\frac{D}{Dt}\int_V \rho u dV = \int_S \boldsymbol{P} dS + \int_V \rho \mathbf{f} dV$$

The stress tensor terms can be denoted by nine stress components at any given point, one normal and two shear components on each coordinate plane, see Fig. B.1. The conservation of momentum can then be written as

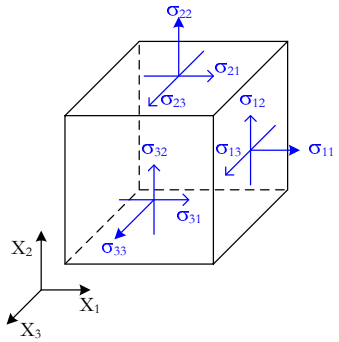


Figure B.1. Cartesian stress components on a volume element.

$$\frac{D}{Dt}\int_V \rho u_j dV = \int_S \sigma_{ij} n_i dS + \int_V \rho f_j dV$$

where f_j is the component of the body force in the j direction and the stress tensor σ_{ij} can also be written as

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij}$$

where p is the pressure, δ_{ij} is the Kronecker delta function and τ_{ij} is the shear stress tensor and is dependent only on the motion of the fluid. In this dissertation we will neglect viscosity such that the shear stress terms $\tau_{ij} = 0$. The surface integral term can be converted to a volume integral by use of Gauss's Theorem.

Theorem 3 (Gauss's Theorem) *Let V be a closed region in 3-space, the boundary of which is a piecewise smooth orientable surface S, and let $\mathbf{v}$ be a vector field that is defined and C^1 in V. Assuming mass is not created or destroyed and there are no discontinuities, and if $\hat{\mathbf{n}}$ denotes the outward unit normal on S, then*

$$\int_V (\nabla \cdot \mathbf{v}) dV = \oint_S \hat{\mathbf{n}} \cdot \mathbf{v} d\mathbf{S}$$

giving us

$$\int_S \sigma_{ij} n_i dS = \int_V \frac{\partial \sigma_{ij}}{\partial x_i} dV$$

such that

$$\frac{D}{Dt} \int_V \rho u_j dV = \int_V \frac{\partial \sigma_{ij}}{\partial x_i} dV + \int_V \rho f_j dV$$

Grouping all of the terms on one side of the equation allows us to set the integrand equal to zero:

$$\frac{D}{Dt} \rho u_j = \frac{\partial \sigma_{ij}}{\partial x_i} + \rho f_j$$

For inviscid flow, viscous effects or shear stresses are neglected and the surface forces are only due to pressure distributions acting normal to the surface.

$$\frac{D}{Dt}\rho u_j = \frac{\partial}{\partial x_i}(-p\delta_{ij}) + \rho f_j$$

Or, in conservative form, the momentum equation is given as:

$$\frac{\partial}{\partial t}(\rho u_j) + \nabla \cdot (\rho u_j \mathbf{V}) = \frac{\partial}{\partial x_i}(-p\delta_{ij}) + \rho f_j \tag{B.2}$$

B.3 Energy equation

For the conservation of energy equation, we look at the first law of thermodynamics which can be stated as follows for a fluid element moving with the flow: "*The rate of change in internal energy de inside the fluid element is equal to the net flux of heat δq into the element plus the rate of work done δw on the element due to body and surface forces.*" [63]

$$de = \delta q + \delta w \tag{B.3}$$

Let us first consider the work done on the moving fluid element. Since work is the scalar product of force and distance, we really need only to consider the same forces that we had for the momentum equation: body forces $\mathbf{f}$, and surface forces $\boldsymbol{P}$. Starting with the body forces moving at some velocity $\mathbf{V}$, the work done by them is given by

$$\delta w_{\mathbf{f}} = \rho \mathbf{f} \cdot \mathbf{V}(dx\ dy\ dz)$$

Next, the work done due to surface forces, again for inviscid flow

$$\delta w_{\boldsymbol{P}} = -\nabla \cdot (p\mathbf{V})(dx\ dy\ dz)$$

The heat flux into the fluid element can be attributed to two main factors; volumetric heating due to absorption or emission of radiation, and heat transfer across the surface due to temperature gradients (which in the case of inviscid flow can be neglected). If we define $\dot{q}$ as the rate of volumetric heat addition per unit mass, the volumetric heating of the fluid element is then given by

$$\delta q = \rho \dot{q}\ dx\ dy\ dz$$

The energy term in the first law of thermodynamics Eq.(B.3) is the internal energy of the system; however this assumes a stationary system. In the case of a moving fluid, this is not correct. We must therefore include the effects of a local velocity $\mathbf{V}$ and consequently the kinetic energy. Thus the total energy per unit mass is given by

$$E = e + \frac{\mathbf{V}^2}{2}$$

Finally, for the rate of change in total energy of the system we will take as the time rate of change per unit mass.

$$dE = \rho \frac{DE}{Dt}\ dx\ dy\ dz$$

The first law can now be rewritten to include all of our terms giving us the final form of the energy equation:

$$\rho \frac{DE}{Dt} = \rho \mathbf{f} \cdot \mathbf{V} - \nabla \cdot (p\mathbf{V}) + \rho \dot{q}$$

or written in conservative form:

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\rho E \mathbf{V}) = \rho \dot{q} - \nabla \cdot (p\mathbf{V}) + \rho \mathbf{f} \cdot \mathbf{V} \tag{B.4}$$

B.4 Summary of the Euler flow equations

The equations of motion that govern fluid flow in a region are given by Eqs. (B.1), (B.2) and (B.4) and are summarized as follows:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) &= 0 \\ \frac{\partial}{\partial t}(\rho u_j) + \nabla \cdot (\rho u_j \mathbf{V}) + \frac{\partial}{\partial x_i}(-p\delta_{ij}) &= \rho f_j \\ \frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\rho E \mathbf{V}) + \nabla \cdot (p\mathbf{V}) &= \rho \dot{q} + \rho \mathbf{f} \cdot \mathbf{V} \end{aligned}$$

The above set of equations are specific to inviscid, compressible flows and are termed the Euler equations [63].

If we expand the ∇ derivatives for a Cartesian system

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} &= 0 \\ \frac{\partial}{\partial t}(\rho u_j) + \frac{\partial \rho u u_j}{\partial x} + \frac{\partial \rho v u_j}{\partial y} + \frac{\partial \rho w u_j}{\partial z} + \frac{\partial}{\partial x_i}(p\delta_{ij}) &= \rho f_j \\ \frac{\partial}{\partial t}(\rho E) + \frac{\partial \rho u E}{\partial x} + \frac{\partial \rho v E}{\partial y} + \frac{\partial \rho w E}{\partial z} + \frac{\partial p u}{\partial x} + \frac{\partial p v}{\partial y} + \frac{\partial p w}{\partial z} &= \rho \dot{q} + \rho \mathbf{f} \cdot \mathbf{V} \end{aligned}$$

and expand the momentum equation into its three coordinate components,

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} &= 0 \\
\frac{\partial}{\partial t}(\rho u) + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho uv}{\partial y} + \frac{\partial \rho uw}{\partial z} + \frac{\partial}{\partial x}p &= \rho f_x \\
\frac{\partial}{\partial t}(\rho v) + \frac{\partial \rho uv}{\partial x} + \frac{\partial \rho v^2}{\partial y} + \frac{\partial \rho vw}{\partial z} + \frac{\partial}{\partial y}p &= \rho f_y \\
\frac{\partial}{\partial t}(\rho w) + \frac{\partial \rho uw}{\partial x} + \frac{\partial \rho vw}{\partial y} + \frac{\partial \rho w^2}{\partial z} + \frac{\partial}{\partial z}p &= \rho f_z \\
\frac{\partial}{\partial t}(\rho E) + \frac{\partial \rho uE}{\partial x} + \frac{\partial \rho vE}{\partial y} + \frac{\partial \rho wE}{\partial z} + \frac{\partial pu}{\partial x} + \frac{\partial pv}{\partial y} + \frac{\partial pw}{\partial z} &= \rho\dot{q} + \rho\mathbf{f}\cdot\mathbf{V}
\end{aligned} \tag{B.5}$$

The complete, nonlinear, Euler equations in conservative form for a compressible, inviscid flow are finally summarized in Eq. (B.5). These equations can be recast into the following form in order to facilitate the implementation of the flux splitting flow solver method described in the next section. We are limiting the equations to two-dimensions here to simplify the problem and because the inlets studied are easily represented in two-dimensions.

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S} \tag{B.6}$$

where $\mathbf{U}$ is called the *solution vector*, $\mathbf{F}$ and $\mathbf{G}$ are know as the *flux vectors* and $\mathbf{S}$ represents a *source term*. They are all column vectors given as:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho\dot{q} + \rho\mathbf{f}\cdot\mathbf{V} \end{bmatrix}$$

where ρ, p, u and v are the density, pressure and x and y velocity components respectively. The energy per unit volume, ρE as well as the following perfect gas relations are assumed to hold

$$\rho E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2), \qquad p = \rho RT, \qquad c = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\gamma RT}$$

where $\gamma = 1.4$ is the ratio of specific heats; T, R and c are the temperature, the gas constant and the speed of sound respectively. If we neglect the body forces and volumetric heating, then $\mathbf{S} = 0$.

APPENDIX C

CFD ALGORITHM FOR THE ADJOINT EQUATIONS

Following the methodology of Jameson's finite volume scheme [14, 11], the steady-state adjoint equations are solved by adding the time derivative term into the equation as shown in Eq.(C.1). The negative sign on the spatial terms is the result of integration by parts in the derivation of this equation and implies a reversal on the convective direction from the flow solution:

$$\frac{\partial \mathbf{\Lambda}}{\partial t} - \mathbf{A}^T \frac{\partial \mathbf{\Lambda}}{\partial x} - \mathbf{B}^T \frac{\partial \mathbf{\Lambda}}{\partial y} = 0 \tag{C.1}$$

In integral form, the equation is written as

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{\Lambda} d\Omega - \int_{s} \mathbf{F} \cdot \mathbf{n} \; ds = 0 \tag{C.2}$$

where $\mathbf{F}$ is the surface flux and $\mathbf{n}$ is the unit normal directed outward from the surface.

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix}, \quad \mathbf{F} \cdot \mathbf{n} = \mathbf{A}^T \mathbf{\Lambda} n_x + \mathbf{B}^T \mathbf{\Lambda} n_y$$

The volume integral in Eq.(C.2) can be discretized over a cell (i, j) with volume $V_{i,j}$ as

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{\Lambda} d\Omega = \frac{d}{dt} \left(\mathbf{\Lambda}_{i,j} V_{i,j} \right)$$

Next the surface integral in Eq.(C.2) is decomposed into constituent surface elements as

$$\int_s \mathbf{F} \cdot \mathbf{n} ds = \sum_k \mathbf{F}_k \mathbf{S}_k$$

where $\mathbf{F}_k$ are the discrete fluxes at each cell face and $\mathbf{S}_k$ are the surface normals, see Fig. C.1. The cell face fluxes are approximated by averaging the cell-centered values from the two adjoining cells. For example, Face 2 is given as follows

$$\begin{aligned}
\mathbf{F}_2 &= \frac{1}{2}\left(\mathbf{F}_{i+1,j} + \mathbf{F}_{i,j}\right) \\
\mathbf{F}_2\mathbf{S}_2 &= \frac{1}{2}\left(\underbrace{\left(\mathbf{A}^T_{i+1,j}\mathbf{\Lambda}_{i+1,j}dy_2 - \mathbf{B}^T_{i+1,j}\mathbf{\Lambda}_{i+1,j}dx_2\right)}_{\hat{\mathbf{A}}^T_{i+1,j}\mathbf{\Lambda}_{i+1,j}} + \underbrace{\left(\mathbf{A}^T_{i,j}\mathbf{\Lambda}_{i,j}dy_2 - \mathbf{B}^T_{i,j}\mathbf{\Lambda}_{i,j}dx_2\right)}_{\hat{\mathbf{A}}^T_{i,j}\mathbf{\Lambda}_{i,j}}\right) \\
&= \frac{1}{2}\Big(\quad \hat{\mathbf{A}}^T_{i+1,j}\mathbf{\Lambda}_{i+1,j} \quad + \quad \hat{\mathbf{A}}^T_{i,j}\mathbf{\Lambda}_{i,j} \quad \Big)
\end{aligned}$$

So the final differential equation for a cell (i, j) is given as,

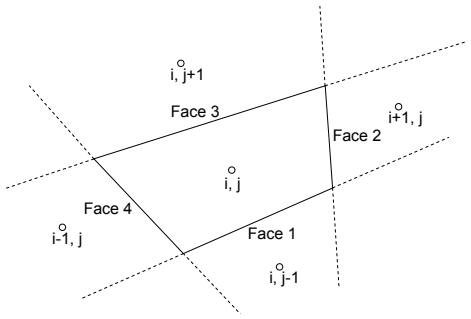


Figure C.1. Cell-centered mesh.

$$\begin{aligned}\frac{d}{dt}\left(\mathbf{\Lambda}_{i,j}V_{i,j}\right) &= \sum_k \mathbf{F}_k\mathbf{S}_k \\ &= \frac{1}{2}\Big[\left(\hat{\mathbf{A}}^T_{i+1,j}\mathbf{\Lambda}_{i+1,j}+\hat{\mathbf{A}}^T_{i,j}\mathbf{\Lambda}_{i,j}\right)_2-\left(\hat{\mathbf{A}}^T_{i,j}\mathbf{\Lambda}_{i,j}+\hat{\mathbf{A}}^T_{i-1,j}\mathbf{\Lambda}_{i-1,j}\right)_4+ \\ &\quad \left(\hat{\mathbf{B}}^T_{i,j+1}\mathbf{\Lambda}_{i,j+1}+\hat{\mathbf{B}}^T_{i,j}\mathbf{\Lambda}_{i,j}\right)_3-\left(\hat{\mathbf{B}}^T_{i,j}\mathbf{\Lambda}_{i,j}+\hat{\mathbf{B}}^T_{i,j-1}\mathbf{\Lambda}_{i,j-1}\right)_1\Big] \quad \text{(C.3)}\end{aligned}$$

The solution method adopted is a second-order central difference scheme for the spatial derivatives and a first-order Euler integration in time. Dissipation terms are added to provide damping to avoid typical second-order instability issues. A fourth-order term is used to smooth odd-even oscillations in smooth areas and a second-order term is used to smooth areas of steep gradients. The semi-discrete form is given as

$$\frac{d}{dt}\left(\mathbf{\Lambda}_{i,j}V_{i,j}\right) = \mathbf{Q}_{i,j}+\mathbf{D}_{i,j}$$

where $\mathbf{Q}_{i,j}$ is the right hand side of Eq.(C.3) and $\mathbf{D}_{i,j}$ is given as

$$\mathbf{D}_{i,j} = \mathbf{D}^2_{i,j}-\mathbf{D}^4_{i,j}$$

where

$$\begin{aligned}\mathbf{D}^2_{i,j} &= d^2_{i+\frac{1}{2},j}-d^2_{i-\frac{1}{2},j}+d^2_{i,j+\frac{1}{2}}-d^2_{i,j-\frac{1}{2}} \\ \mathbf{D}^4_{i,j} &= d^4_{i+\frac{1}{2},j}-d^4_{i-\frac{1}{2},j}+d^4_{i,j+\frac{1}{2}}-d^4_{i,j-\frac{1}{2}}\end{aligned}$$

The terms on the right hand side all have a similar form and can be extrapolated from the following example

$$\begin{aligned}d^2_{i+\frac{1}{2},j} &= \epsilon^2_{i+\frac{1}{2},j}(\mathbf{\Lambda}_{i+1,j}-\mathbf{\Lambda}_{i,j}) \\ d^4_{i+\frac{1}{2},j} &= \epsilon^4_{i+\frac{1}{2},j}(\mathbf{\Lambda}_{i+2,j}-3\mathbf{\Lambda}_{i+1,j}+3\mathbf{\Lambda}_{i,j}-\mathbf{\Lambda}_{i-1,j})\end{aligned}$$

The ϵ^2 and ϵ^4 terms allow for a blending of the two types of dissipation. When scaled properly the d^4 terms dominate in areas of smooth flow and will eliminate the effects of odd-even decoupling of the central difference scheme, and the d^2 term will take over in areas of steep gradients such as around shocks. The ϵ^2 term is appropriately scaled by the normalized second difference of the pressure.

$$\begin{aligned}
\epsilon^2_{i+\frac{1}{2},j} &= \frac{\mu^2 V_{i+\frac{1}{2},j}}{\Delta t_{i,j}} \max\left(\nu^i_{i+2,j},\ \nu^i_{i+1,j},\ \nu^i_{i,j},\ \nu^i_{i-1,j}\right) \\
\epsilon^4_{i+\frac{1}{2},j} &= \max\left[0, \left(\frac{\mu^4 V_{i+\frac{1}{2},j}}{\Delta t_{i,j}} - \epsilon^2_{i+\frac{1}{2},j}\right)\right]
\end{aligned}$$

$$\begin{aligned}
\nu^i_{i,j} &= \left|\frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}}\right| \\
\nu^j_{i,j} &= \left|\frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{p_{i,j+1} + 2p_{i,j} + p_{i,j-1}}\right|
\end{aligned}$$

The μ^2 and μ^4 terms are selected by the user to scale the influence of the damping terms, they are typically set to 1/4 and 1/256 respectively. To simplify the solver we chose not to implement a variable time step, high order Runge-Kutta time integration or the convergence accelerators utilized in [14, 11]. We chose instead to implement an Euler integration for the time derivative and the update equation is given as

$$\mathbf{\Lambda}^{n+1}_{i,j} = \mathbf{\Lambda}^n_{i,j} + \frac{\Delta t}{V_{i,j}}\left(\mathbf{Q}_{i,j} + \mathbf{D}_{i,j}\right)$$

APPENDIX D

DERIVATION OF THE BURGERS EQUATION IN CAUCHY FORM

D.1 Semi-discrete linear model

To write Burgers equation in the abstract Cauchy form, we must linearize the equation about some equilibrium point U_o. To this end, we substitute in the small perturbation $U = U_o + \Delta U$

$$\frac{\partial}{\partial t}(U_o + \Delta U) + K_1(U_o + \Delta U)\frac{\partial}{\partial x}(U_o + \Delta U) + K_2(U_o + \Delta U)\frac{\partial}{\partial y}(U_o + \Delta U) = \frac{1}{Re}\left(\frac{\partial^2}{\partial x^2}(U_o + \Delta U) + \frac{\partial^2}{\partial y^2}(U_o + \Delta U)\right)$$

Since the equilibrium point U_o is constant, the equation simplifies to

$$\frac{\partial}{\partial t}(\Delta U)+K_1(U_o+\Delta U)\frac{\partial}{\partial x}(\Delta U)+K_2(U_o+\Delta U)\frac{\partial}{\partial y}(\Delta U) = \frac{1}{Re}\left[\frac{\partial^2}{\partial x^2}(\Delta U) + \frac{\partial^2}{\partial y^2}(\Delta U)\right]$$

Neglecting the higher-order terms, $\Delta U \frac{\partial}{\partial x}(\Delta U)$ gives the linear Burgers equation

$$\frac{\partial}{\partial t}(\Delta U) + K_1 U_o \frac{\partial}{\partial x}(\Delta U) + K_2 U_o \frac{\partial}{\partial y}(\Delta U) = \frac{1}{Re}\left[\frac{\partial^2}{\partial x^2}(\Delta U) + \frac{\partial^2}{\partial y^2}(\Delta U)\right] \quad \text{(D.1)}$$

Next, we discretize the spatial derivatives using central differencing (since we no longer have a time derivative we can drop the superscript n notation for the time step)

$$\begin{aligned}
\frac{\partial}{\partial x}(\Delta U) &= \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta x} \\
\frac{\partial}{\partial y}(\Delta U) &= \frac{U_{i,j+1} - U_{i,j-1}}{2\Delta y} \\
\frac{\partial^2}{\partial x^2}(\Delta U) &= \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta x^2} \\
\frac{\partial^2}{\partial y^2}(\Delta U) &= \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta y^2}
\end{aligned}$$

substitute these into the linearized Burgers Eq.(D.1)

$$\begin{aligned}\frac{\partial}{\partial t}(\Delta w) &= -K_1 U_o \frac{U^n_{i+1,j} - U^n_{i-1,j}}{2\Delta x} - K_2 U_o \frac{U^n_{i,j+1} - U^n_{i,j-1}}{2\Delta y} \\ &\quad + \frac{1}{Re}\left(\frac{U^n_{i+1,j} - 2U^n_{i,j} + U^n_{i-1,j}}{\Delta x^2} + \frac{U^n_{i,j+1} - 2U^n_{i,j} + U^n_{i,j-1}}{\Delta y^2}\right)\end{aligned}$$

For now we set $a_x = \frac{K_1 U_o \Delta t}{2\Delta x}$ and $b_x = \frac{\Delta t}{Re \Delta x^2}$ and similarly $a_y = \frac{K_2 U_o \Delta t}{2\Delta y}$ and $b_y = \frac{\Delta t}{Re \Delta y^2}$. The equation simplifies to

$$\begin{aligned}\frac{\partial}{\partial t}(\Delta U) &= -a_x(U_{i+1,j} - U_{i-1,j}) - a_y(U_{i,j+1} - U_{i,j-1}) \\ &\quad + b_x\left(U_{i+1,j} - 2U_{i,j} + U_{i-1,j}\right) + b_y\left(U_{i,j+1} - 2U_{i,j} + U_{i,j-1}\right) \\ &= (-2b_x - 2b_y)U_{i,j} + (b_x - a_x)U_{i+1,j} + (b_y - a_y)U_{i,j+1} \\ &\quad + (b_x + a_x)U_{i-1,j} + (b_y + a_y)U_{i,j-1}\end{aligned}$$

The boundary conditions are given as

$$\begin{aligned} U(x_0, y_j) = U_{0,j} = f(y_j), \qquad & U(x_{p+1}, y_j) = U_{p+1,j} = U_{p,j} \\ U(x_i, y_0) = U_{i,0} = 0, \qquad & U(x_i, y_{q+1}) = U_{i,q+1} = 0 \end{aligned}$$

Grouping the j terms together, we can build a matrix of the form

$$\mathbf{H} = \begin{bmatrix} -2b_x - 2b_y & b_x - a_x & 0 & \dots & 0 & 0 \\ b_x + a_x & -2b_x - 2b_y & b_x - a_x & \dots & 0 & 0 \\ 0 & b_x + a_x & -2b_x - 2b_y & \dots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & b_x - a_x & 0 \\ 0 & 0 & 0 & \dots & -2b_x - 2b_y & b_x - a_x \\ 0 & 0 & 0 & \dots & b_x + a_x & -2b_x - 2b_y \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} b_y - a_y & 0 & 0 & \dots \\ 0 & b_y - a_y & 0 & \\ 0 & 0 & b_y - a_y & \\ \vdots & & & \ddots \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} b_y + a_y & 0 & 0 & \dots \\ 0 & b_y + a_y & 0 & \\ 0 & 0 & b_y + a_y & \\ \vdots & & & \ddots \end{bmatrix}$$

associated with the vectors

$$U_j = \begin{bmatrix} U_{1,j} \\ U_{2,j} \\ U_{3,j} \\ \vdots \\ U_{p-1,j} \\ U_{p,j} \end{bmatrix}, \qquad U_{j+1} = \begin{bmatrix} U_{1,j+1} \\ U_{2,j+1} \\ U_{3,j+1} \\ \vdots \\ U_{p-1,j+1} \\ U_{p,j+1} \end{bmatrix}, \qquad U_{j-1} = \begin{bmatrix} U_{1,j-1} \\ U_{2,j-1} \\ U_{3,j-1} \\ \vdots \\ U_{p-1,j-1} \\ U_{p,j-1} \end{bmatrix}$$

Now the semi-discretized Burgers equation can be written as

$$\frac{\partial}{\partial t}(\Delta U) = \mathbf{H}U_j + \mathbf{K}U_{j-1} + \mathbf{J}U_{j+1}$$

Adding in the boundary conditions

$$\frac{\partial}{\partial t}(\Delta U) = \underbrace{\begin{bmatrix} -2b_x - 2b_y & b_x - a_x & 0 & \dots \\ b_x + a_x & -2b_x - 2b_y & b_x - a_x & \dots \\ 0 & b_x + a_x & -2b_x - 2b_y & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{H} \begin{bmatrix} U_{1,j} \\ U_{2,j} \\ \vdots \\ U_{p,j} \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} b_y - a_y & 0 & 0 & \dots \\ 0 & b_y - a_y & 0 & \dots \\ 0 & 0 & b_y - a_y & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{J} \begin{bmatrix} U_{1,j+1} \\ U_{2,j+1} \\ \vdots \\ U_{p,j+1} \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} b_y + a_y & 0 & 0 & \dots \\ 0 & b_y + a_y & 0 & \dots \\ 0 & 0 & b_y + a_y & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{K} \begin{bmatrix} U_{1,j-1} \\ U_{2,j-1} \\ \vdots \\ U_{p,j-1} \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} b_x + a_x \\ 0 \\ \vdots \\ 0 \\ b_x - a_x \end{bmatrix} \begin{bmatrix} U_{0,j} \\ 0 \\ \vdots \\ 0 \\ U_{p,j} \end{bmatrix}}_{f}$$

Let

$$\mathcal{A} = \begin{bmatrix} H & J & 0 & \dots & 0 & 0 & 0 \\ K & H & J & \dots & 0 & 0 & 0 \\ 0 & K & H & \dots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \dots & H & J & 0 \\ 0 & 0 & 0 & \dots & K & H & J \\ 0 & 0 & 0 & \dots & 0 & K & H \end{bmatrix}, \qquad \mathcal{F} = \begin{bmatrix} f \\ f \\ f \\ \vdots \\ f \\ f \\ f \end{bmatrix}, \qquad U = \begin{bmatrix} U_{1,1} \\ U_{2,1} \\ \vdots \\ U_{p,1} \\ U_{1,2} \\ U_{2,2} \\ \vdots \\ U_{p,2} \\ U_{1,3} \\ \vdots \\ \vdots \\ U_{p-1,q} \\ U_{p,q} \end{bmatrix}$$

The final uncontrolled linear model is given as

$$\frac{\partial}{\partial t}(\Delta U) = \mathcal{A}U + \mathcal{F}$$

D.2 Semi-discrete nonlinear model

$$\frac{\partial}{\partial t}U(t,x,y) + K_1\frac{\partial}{\partial x}U(t,x,y)^2 + K_2\frac{\partial}{\partial y}U(t,x,y)^2 = \frac{1}{Re}\left(\frac{\partial^2}{\partial x^2}U(t,x,y) + \frac{\partial^2}{\partial y^2}U(t,x,y)\right)$$

The nonlinear solution looks a little different than the linear model but the results are derived in the same manor. This time we will use the discretization method given by Camphouse and Myatt [29]:

$$\begin{aligned}
\frac{\partial U^2}{\partial x} &= \frac{(U_{i,j}+U_{i+1,j})^2-(U_{i-1,j}+U_{i,j})^2}{4\Delta x} \\
&\quad +\gamma\frac{|U_{i,j}+U_{i+1,j}|(U_{i,j}-U_{i+1,j})-|U_{i-1,j}+U_{i,j}|(U_{i-1,j}-U_{i,j})}{4\Delta x} \\
\frac{\partial U^2}{\partial y} &= \frac{(U_{i,j}+U_{i,j+1})^2-(U_{i,j-1}+U_{i,j})^2}{4\Delta y} \\
&\quad +\gamma\frac{|U_{i,j}+U_{i,j+1}|(U_{i,j}-U_{i,j+1})-|U_{i,j-1}+U_{i,j}|(U_{i,j-1}-U_{i,j})}{4\Delta y} \\
\frac{\partial^2 U}{\partial x^2} &= \frac{U_{i+1,j}-2U_{i,j}+U_{i-1,j}}{\Delta x^2} \\
\frac{\partial^2 U}{\partial y^2} &= \frac{U_{i,j+1}-2U_{i,j}+U_{i,j-1}}{\Delta y^2}
\end{aligned}$$

Substituting these into the semi discrete equation

$$\begin{aligned}
\frac{\partial}{\partial t}U(t,x,y) &= -K_1\left[\frac{(U_{i,j}+U_{i+1,j})^2-(U_{i-1,j}+U_{i,j})^2}{4\Delta x}\right. \\
&\quad \left.+\gamma\frac{|U_{i,j}+U_{i+1,j}|(U_{i,j}-U_{i+1,j})-|U_{i-1,j}+U_{i,j}|(U_{i-1,j}-U_{i,j})}{4\Delta x}\right] \\
&\quad -K_2\left[\frac{(U_{i,j}+U_{i,j+1})^2-(U_{i,j-1}+U_{i,j})^2}{4\Delta y}\right. \\
&\quad \left.+\gamma\frac{|U_{i,j}+U_{i,j+1}|(U_{i,j}-U_{i,j+1})-|U_{i,j-1}+U_{i,j}|(U_{i,j-1}-U_{i,j})}{4\Delta y}\right] \\
&\quad +\frac{1}{Re}\left(\frac{U_{i+1,j}-2U_{i,j}+U_{i-1,j}}{\Delta x^2}+\frac{U_{i,j+1}-2U_{i,j}+U_{i,j-1}}{\Delta y^2}\right)
\end{aligned}$$

Notice that the linear terms are now limited to the last line of the equation and the remaining terms are nonlinear. The linear terms are dealt with in the same manor as

before to build the **A** matrix and the nonlinear terms are grouped together to make up the $G(U)$ term in the matrix equation.

If we assume that $\Delta x = \Delta y = h$ and $K_2 = 0$ the equation simplifies to

$$\begin{aligned}\frac{\partial}{\partial t}U(t,x,y) &= -\frac{K_1}{4h}\Big[(U_{i,j}+U_{i+1,j})^2-(U_{i-1,j}+U_{i,j})^2 \\ &\quad +\gamma\big(|U_{i,j}+U_{i+1,j}|(U_{i,j}-U_{i+1,j})-|U_{i-1,j}+U_{i,j}|(U_{i-1,j}-U_{i,j})\big)\Big] \\ &\quad +\frac{1}{Re\cdot h^2}\left(U_{i+1,j}-4U_{i,j}+U_{i-1,j}+U_{i,j+1}+U_{i,j-1}\right)\end{aligned}$$

again, grouping the j terms together the H matrix is given by

$$\mathbf{H} = \frac{1}{Re\cdot h^2}\begin{bmatrix} -4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -4 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & -4 & \dots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \dots & -4 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & -4 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -4 \end{bmatrix}$$

The $j+1$ and $j-1$ terms can be constructed into identity matrices

$$\mathbf{J} = \frac{1}{Re\cdot h^2}I_{p\times p}$$

The boundary conditions are included as

$$f = \begin{bmatrix} \frac{1}{Re \cdot h^2} \\ 0 \\ \vdots \\ 0 \\ \frac{1}{Re \cdot h^2} \end{bmatrix} \begin{bmatrix} U_{0,j} \\ 0 \\ \vdots \\ 0 \\ U_{p,j} \end{bmatrix}$$

This results in

$$\frac{\partial}{\partial t} U = \mathbf{H} U_j + \mathbf{J} U_{j-1} + \mathbf{J} U_{j+1} + f$$

It is easily seen that the $\mathcal{A}$ matrix and $\mathcal{F}$ vector are given by

$$\mathcal{A} = \begin{bmatrix} H & J & 0 & \dots & 0 & 0 & 0 \\ J & H & J & \dots & 0 & 0 & 0 \\ 0 & J & H & \dots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \dots & H & J & 0 \\ 0 & 0 & 0 & \dots & J & H & J \\ 0 & 0 & 0 & \dots & 0 & J & H \end{bmatrix} \qquad \mathcal{F} = \begin{bmatrix} f \\ f \\ f \\ \vdots \\ f \\ f \\ f \end{bmatrix}$$

The nonlinear terms are then assembled into the $\mathcal{G}(U)$ vector as follows

$$\mathcal{G} = \frac{K_1}{4h}\begin{bmatrix} (U_{1,1}+U_{2,1})^2-(U_{0,1}+U_{1,1})^2 \\ +\gamma\big(|U_{1,1}+U_{2,1}|(U_{1,1}-U_{2,1})-|U_{0,1}+U_{1,1}|(U_{0,1}-U_{1,1})\big) \\ (U_{2,1}+U_{3,1})^2-(U_{1,1}+U_{2,1})^2 \\ +\gamma\big(|U_{2,1}+U_{3,1}|(U_{2,1}-U_{3,1})-|U_{1,1}+U_{2,1}|(U_{1,1}-U_{2,1})\big) \\ (U_{3,1}+U_{4,1})^2-(U_{2,1}+U_{3,1})^2 \\ +\gamma\big(|U_{3,1}+U_{4,1}|(U_{3,1}-U_{4,1})-|U_{2,1}+U_{3,1}|(U_{2,1}-U_{3,1})\big) \\ \vdots \\ (U_{p,q}+U_{p,q})^2-(U_{p,q}+U_{p,q})^2 \\ +\gamma\big(|U_{p,q}+U_{p+1,q}|(U_{p,q}-U_{p+1,q})-|U_{p-1,q}+U_{p,q}|(U_{p-1,q}-U_{p,q})\big) \end{bmatrix}$$

The uncontrolled nonlinear model is given as

$$\frac{\partial}{\partial t}U = \mathcal{A}U + \mathcal{F} + \mathcal{G}(U)$$

D.3 Control variables

In this section we build the control matrix $\mathcal{B}$ associated with the control variables u. This matrix is the same in both the linear and nonlinear cases. In the problems studied in this dissertation the controls are implemented on the surface of the obstacle in the flow. There are two controls on each the front and back of the obstacle and one control on the top and bottom. The spatial influence of each control is piecewise constant. Our controls are then simplified to

$$U(t,\Gamma_1) = \alpha_1\phi_1, \qquad U(t,\Gamma_2) = \alpha_{2,1}\phi_{2,1} + \alpha_{2,2}\phi_{2,2}$$

$$U(t,\Gamma_3) = \alpha_3\phi_3, \qquad U(t,\Gamma_4) = \alpha_{4,1}\phi_{4,1} + \alpha_{4,2}\phi_{4,2}$$

where $\alpha_{j,i}$ is the i^{th} control on Γ_j and $\phi_{j,i}$ is a function describing the influence of the i^{th} control on Γ_j.

We can now build the control matrix as

$$\mathcal{B} = \begin{bmatrix}
(0) & (0) & (0) & (0) & (0) & (0)_{1\to p,1} \\
 & \vdots & & & \vdots & \\
(b)_{1\to p,b1}*\phi_1 & & & & & \\
(0) & (b)_{1\to p,b1+1}*\phi_{2,1} & (0) & (0) & (b)_{1\to p,b1+1}*\phi_{4,1} & (0) \\
(0) & (b)_{1\to p,b1+2}*\phi_{2,1} & (0) & (0) & (b)_{1\to p,b1+2}*\phi_{4,1} & (0) \\
(0) & (b)_{1\to p,b1+3}*\phi_{2,1} & (0) & (0) & (b)_{1\to p,b1+3}*\phi_{4,1} & (0) \\
(0) & \vdots & (0) & & \vdots & (0) \\
(0) & (0) & \vdots & & (0) & \vdots \\
(0) & (0) & (b)_{1\to p,b2-3}*\phi_{2,2} & (0) & (0) & (b)_{1\to p,b2-2}*\phi_{4,2} \\
(0) & (0) & (b)_{1\to p,b2-2}*\phi_{2,2} & (0) & (0) & (b)_{1\to p,b2-1}*\phi_{4,2} \\
(0) & (0) & (b)_{1\to p,b2-1}*\phi_{2,2} & (0) & (0) & (b)_{1\to p,b2}*\phi_{4,2} \\
(0) & (0) & (0) & (b)_{1-p,b2}*\phi_3 & (0) & (0) \\
 & \vdots & & & \vdots & \\
(0) & (0) & (0) & (0) & (0) & (0)_{1\to p,q}
\end{bmatrix}$$

where $(\cdot)_{1\to p,j}$ denotes the j_{th} row of the grid. The final matrix size is $pq \times 6$ in this case for the 6 control variables.

The final nonlinear model with control given in Cauchy form is given as

$$\frac{\partial}{\partial t}U = \mathcal{A}U + \mathcal{F} + \mathcal{G}(U) + \mathcal{B}\alpha$$

REFERENCES

[1] Lions, J. L., *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag New York Heidelberg Berlin, 1971.

[2] Lasiecka, I., "Control of Systems Governed by Partial Differential Equations — A Historical Perspective," *34th Conference on Decision and Control, New Orleans, LA*, December 1995, pp. 2792–2795.

[3] Giles, M. and Pierce, N., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, No. 3-4, 2000, pp. 393–415.

[4] Pironeau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 1, 1974, pp. 97–110.

[5] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260.

[6] Jameson, A., "Optimum Aerodynamic Design Using CFD and Control Theory," *12th Computational Fluid Dynamics Conference, San Diego, CA*, June 1995, AIAA Paper 1995-1729.

[7] Giles, M. and Pierce, N., "Adjoint Equations in CFD: Duality, Boundary Conditions and Solution Behavior," *AIAA Journal*, 1997, pp. 182–198, AIAA Paper 97-1850.

[8] Giles, M. and Pierce, N., "On the Properties of Solutions of the Adjoint Euler Equations," *6th ICFD Conference on Numerical Methods for Fluid Dynamics*, 1998.

[9] Giles, M. and Pierce, N., "Analytic Adjoint Solutions for the Quasi-One-Dimensional Euler Equations," *Journal of Fluid Mechanics*, Vol. 426, 2001, pp. 327–345.

[10] Ferlauto, M., Iollo, A., and Zannetti, L., "Set of Boundary Conditions for Aerodynamic Design," *AIAA Journal*, Vol. 42, No. 8, 2004, pp. 1582–1592.

[11] Reuther, J. J., *Aerodynamic Shape Optimization Using Control Theory*, Ph.D. thesis, University of California Davis, 1996.

[12] Jameson, A., Alonso, J. J., Reuther, J. J., Martinelli, L., and Vassberg, J. C., "Aerodynamic Shape Optimization Techniques Based on Control Theory," *29th Fluid Dynamics Conference, Albuquerque, NM*, June 15-18 1998, AIAA Paper 1998-2538.

[13] Jameson, A., "Aerodynamic Shape Optimization Using the Adjoint Method," *Lectures at the Von Karman Institute, Brussels*, Febuary 2003.

[14] Xie, L., *Gradient-Based Optimum Aerodynamic Design Using Adjoint Methods*, Ph.D. thesis, Virginia Polytechnic Institute and State University, April 2002.

[15] Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," *35th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, January 6-9 1997, AIAA Paper 1997-0643.

[16] Baysal, O. and Ghayour, K., "Continuous Adjoint Sensitivities for Optimization with General Cost Functionals on Unstructured Meshes," *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 48–55.

[17] Carpenieri, G., Koren, B., and van Tooren, M., "Adjoint-Based Aerodynamic Shape Optimization on Unstructured Meshes," *Journal of Computational Physics*, Vol. 224, 2007, pp. 267–287.

[18] Reuther, J. J., Rimlinger, M. J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1," *Journal of Aircraft*, Vol. 36, No. 1, January-February 1999, pp. 51–60.

[19] Reuther, J. J., Rimlinger, M. J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 2," *Journal of Aircraft*, Vol. 36, No. 1, January-February 1999, pp. 61–74.

[20] Iollo, A. and Salas, M. D., "Contribution to the Optimal Shape Design of Two-Dimensional Internal Flows with Embedded Shocks," *Journal of Computational Physics*, Vol. 125, 1996, pp. 124–134.

[21] Iollo, A., Ferlauto, M., and Zannetti, L., "An Aerodynamic Optimization Method Based on the Inverse Problem Adjoint Equations," *Journal of Computational Physics*, Vol. 173, 2001, pp. 87–115.

[22] Zhang, Z. and Lum, K.-Y., "S-Shaped Inlet Design Optimization Using the Adjoint Equation Method," *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, July 9-12 2006, AIAA Paper 2006-4453.

[23] Choi, H., Temam, R., Moin, P., and Kim, J., "Feedback Control for Unsteady Flow and Its Application to the Stochastic Burgers Equation," *Journal of Fluid Mechanics*, Vol. 253, 1993, pp. 509–543.

[24] Kang, S., Ito, K., and Burns, J. A., "Unbounded Observation and Boundary Control Problems for Burgers Equation," *30th Conference on Decision and Control, Brighton England*, December 1991.

[25] Burns, J. A., Zietsman, L., and Myatt, J. H., "Boundary Layer Control for the Viscous Burgers' Equation," *IEEE International Conference on Control Applications*, September 18-20 2002.

[26] Atwell, J. A. and King, B. B., "Stabilized Finite Element Methods and Feedback Control for Burgers' Equation," *American Control Conference*, June 2000, pp. 2745–2749.

[27] King, B. B. and Krueger, D. A., "Burgers' Equation: Galerkin Least-Squares Approximations and Feedback Control," *Mathematical and Computer Modelling*, Vol. 38, 2003, pp. 1075–1085.

[28] Borggaard, J., Burns, J., and Zietsman, L., "Computational Challenges in Control of Partial Differential Equations," *Second AIAA Flow Control Conference*, June 28-July 1 2004, AIAA Paper 2004-2526.

[29] Camphouse, R. C. and Myatt, J. H., "Feedback Control for a Two-Dimensional Burgers' Equation System Model," *Second AIAA Flow Control Conference*, June 28-July 1 2004, AIAA Paper 2004-2411.

[30] Lainiotis, D., "Generalized Chandrasekhar Algorithms: Time-Varying Models," *IEEE Transaction on Automatic Control*, Vol. 21, No. 5, October 1976, pp. 728–732.

[31] Ito, K. and Powers, R. K., "Chandrasekhar Equations for Infinite Dimensional Systems," *SIAM Journal on Control and Optimization*, Vol. 25, No. 3, May 1987, pp. 596–611.

[32] Ito, K. and Ravindran, S. S., "A Reduced Basis Method for Control Problems Goverened by PDEs," *Control and Estimation of Distributed Parameter Systems*, edited by K. K. Wolfgang Desch, F. Kappel, Vol. 126, International Series of Numerical Mathematics, Birkhauser Verlag, Basel, 1998, pp. 173–191, ISBN 3764358351.

[33] Loéve, M., "Fonctions Aléatoire de Second Ordre," *Compte Rendus des Academie des Sciences*, Vol. 220, 1945.

[34] Karhunen, K., "Zur Spektraltheorie Stokastisher Prozesse," *Annales Academiae Scientiorum Fennicae*, 1946.

[35] Sirovich, L., "Turbulence and the Dynamics of Coherent Structures: Part I-III," *Quarterly of Applied Mathematics*, Vol. 45(3), 1987, pp. 561–590.

[36] Berkooz, G., Holmes, P., and Lumley, J. L., "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, Vol. 25, 1993, pp. 539–575.

[37] Ravindran, S. S., "Proper Orthogonal Decomposition in Optimal Control of Fluids," Tech. rep., NASA Langley Research Center, Hampton, Virginia, March 1999, TM-1999-209113.

[38] Burkardt, J., Du, Q., Gunzburger, M., and Lee, H.-C., "Reduced Order Modeling of Complex Systems," *20th Biennial Conference on Numerical Analysis, University of Dundee, Scotland, UK*, June 24-27 2003, pp. 29–38.

[39] Iollo, A., "Remarks on the Approximation of the Euler Equations by a Low Order Model," Research Report RR-3329, Inria, December 1997.

[40] Lucia, D. J. and Beran, P. S., "Projection Methods for Reduced Order Models of Compressible Flows," *Journal of Computational Physics*, Vol. 188, No. 1, June 2003, pp. 252–280.

[41] LeGresley, P. A. and Alonso, J. J., "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition," *AIAA Fluids 2000 Conference and Exhibit*, June 12-22 2000, AIAA Paper 2000-2545.

[42] LeGresley, P. A. and Alonso, J. J., "Investigation of Non-Linear Projection for POD Based Reduced Order Models for Aerodynamics," *39th AIAA Aerospace Sciences Meeting and Exhibit*, January 8-11 2001, AIAA Paper 2001-0926.

[43] Atwell, J. A., *Proper Orthogonal Decomposition for Reduced Order Control of Partial Differential Equations*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, April 10 2000.

[44] Camphouse, R. C., "Boundary Feedback Control Using Proper Orthogonal Decomposition Models," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, September-October 2005, pp. 931–938.

[45] Padhi, R. and Balakrishnan, S. N., "Proper Orthogonal Decomposition Based Feedback Optimal Control Sysnthesis of Distributed Parameter Systems Using Neural Networks," *American Control Conference*, May 8-10 2002.

[46] Kunisch, K. and Volkwein, S., "Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition," *Journal of Optimization Theory and Applications*, Vol. 102, No. 2, August 1999, pp. 345–371.

[47] Ravindran, S. S., "Reduced Order Adaptive Controllers for Fluids Using Proper Orthogonal Decomposition," *39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, Jan 8-11 2001, AIAA Paper 2001-0925.

[48] Ravindran, S. S., "Reduced-Order Controllers for Control of Flow Past an Airfoil," *International Journal for Numerical Methods in Fluids*, Vol. 50, 2006, pp. 531–554.

[49] Ravindran, S. S., "Optimal Boundary Feedback Flow Stabilization by Model Reduction," *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, 2007, pp. 2555–2569.

[50] Singh, S. N., Myatt, J. H., Addington, G. A., Banda, S., and Hall, J. K., "Optimal Feedback Control of Vortex Shedding Using Proper Orthogonal Decomposition Models," *Journal of Fluids Engineering*, Vol. 123, No. 3, September 2001, pp. 612–618.

[51] Buffoni, M., Camarri, S., Iollo, A., and Salvetti, M., "Low-Dimensional Modelling of a Confined Three-Dimensional Wake Flow," *Journal of Fluids Mechanics*, Vol. 569, November 2006, pp. 141–150.

[52] Djouadi, S. M., Camphouse, R. C., and Myatt, J. H., "Optimal Order Reduction for the Two-Dimensional Burgers Equation," *46th IEEE Conference on Decision and Control, New Orleans, LA*, Dec 12-14 2007.

[53] Smith, D. R., *Variational Methods in Optimization*, Prentice-Hall, Inc., 1974.

[54] Stengel, R. F., *Optimal Control and Estimation*, Dover Publications Inc., 1994.

[55] Pettit, C. and Beran, P., "Reduced-Order Modeling for Flutter Prediction," *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Atlanta, Georgia, USA*, April 2000, AIAA Paper 2000-1446.

[56] Camphouse, R. C., Djouadi, S. M., and Myatt, J. H., "Feedback Control for Aerodynamics," Tech. rep., AFRL Air Vehicle Directorate, Air Force Materiel Command, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, March 2006, AFRL-VA-WP-TP-2006-348.

[57] Liou, M. and Steffen, C., "A New Flux Splitting Scheme," *Journal of Computational Physics*, Vol. 107, No. 1, July 1993.

[58] Liou, M., "A Sequel to AUSM, Part II: $AUSM^+ - up$ For All Speeds," *Journal of Computational Physics*, Vol. 214, 2006, pp. 137–170.

[59] Billet, G. and O.Louedin, "Adaptive Limiters for Improving the Accuracy of the MUSCL Approach for Unsteady Flows," *Journal of Computational Physics*, Vol. 170, 2001, pp. 161–183.

[60] Korte, J. and Auslender, A., "Optimization of Contoured Hypersonic Scramjet Inlets with a Least-Squares Parabolized Navier-Stokes Procedure," *Computing Systems in Engineering*, Vol. 4, No. 1, 1993, pp. 13–26.

[61] Munipalli, R., Wasawadigi, G., Anderson, D. A., and Wilson, D., "Application of Optimization Techniques in Inlet Design," *AIAA 13th Applied Aerodynamics conference*, June 19-22 1995, AIAA Paper 95-1824.

[62] Djouadi, S. M., Camphouse, R. C., and Myatt, J. H., "Reduced Order Models for Boundary Feedback Flow Control," *American Control Conference, Seattle, WA*, June 11-13 2008.

[63] Anderson, J. D., *Computational Fluid Dynamics*, McGraw-Hill, Inc., 1995.

BIOGRAPHICAL STATEMENT

Jennifer was born in Fort McMurray, Alberta, Canada in 1973. Her family later moved to Edson, Alberta where Jennifer completed high school with a senior matriculation diploma in 1991 and earned her private pilots license that same year. She attended the University of Calgary from 1992 to 1997 and graduated with a Bachelors of Science degree in Astrophysics.

After graduation Jennifer worked in Geophysics, first as a research assistant for a research group at the University of Calgary then later in industry as a seismic data processor. Concurrently Jennifer became involved with a local air cadet squadron where she served as the operations officer for the unit. She planned all of the squadrons activities and instructed cadets on leadership, aerospace science and survival skills. It was during one of the flights with the squadron that Jennifer met her husband Brendan. In 2000 Jennifer and Brendan moved to the United States for employment opportunities where she took a job as a simulation physicist for Terminal Reality, a Texas based gaming company making a PC flight simulator. Jennifer left the company and began graduate studies in 2003 at the University of Arlington in Texas to pursue a masters in Aerospace Engineering. She graduated with her masters in 2005 and completed the PhD program in May 2009.