RICE UNIVERSITY

Stabilized Finite Element Solution of Optimal Control Problems in Computational Fluid Dynamics

by

Feby V. Abraham

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Yildiz Bayazitoglu, Chair
Professor of Mechanical Engineering and Materials Science

Marek Behr, Thesis Director
Lehrstuhl für Numerische Mechanik
Technische Universität München

Matthias Heinkenschloss
Associate Professor of Computational and Applied Mathematics

Chad Landis
Assistant Professor of Mechanical Engineering and Materials Science

Houston, Texas
April, 2004
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI

UMI Microform 3122446
Copyright 2004 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346
Stabilized Finite Element Solution of Optimal Control Problems in Computational Fluid Dynamics

Feby V. Abraham

Abstract

This thesis discusses the solution of optimal flow control problems, with an emphasis on solving optimal design problems involving blood as the fluid. The discretization of the governing equations of fluid flow is accomplished using stabilized finite element formulations. Although frequently and successfully applied, these methods depend on significant mesh refinement to establish strong consistency properties, when using low-order elements. We present an approach to improve the consistency properties of such methods.

We develop the methodology for the numerical solution of optimal control problems using the aforementioned discretization scheme. For two possible approaches in which the optimal control problem can be discretized—optimize-then-discretize and discretize-then-optimize—we use a boundary control problem governed by the linear Oseen equations to numerically explore the influence of stabilization. We also present indicators for assessing the quality of the computed solution.

We then investigate the influence of the fluid constitutive model on the outcome of shape optimization tasks. Our computations are based on the Navier-Stokes equations generalized to non-Newtonian fluid, with the Carreau-Yasuda model employed
to account for the shear-thinning behavior of blood. The generalized Newtonian treatment exhibits striking differences in the velocity field for smaller shear rates. For a steady flow scenario, we apply gradient-based optimization procedure to a benchmark problem of flow through a right-angle cannula, and to a flow through an idealized arterial graft. We present the issues involved in solving large-scale optimal design problems, and state the numerical formulations for the various approaches that could be used to solve such problems. We numerically demonstrate optimal shape design for unsteady flow in an arterial graft.
To my parents, sister and late grandmother.
Acknowledgments

This thesis is the culmination of about four years of learning, addressing and finding solutions to research issues, and striving for improvements in established methods to solve problems. I take this great opportunity to express my sincere gratitude to all those who have supported, guided and encouraged me in this endeavor.

I thank my advisor Marek Behr for sharing his extensive knowledge, insight and experience in all aspects related to finite element analysis, parallel computing, numerical methods and fluid dynamics. I thank him for exposing me to a range of research problems, and in giving me the freedom to pursue new and challenging problems. I am indebted to him for his patience and time in carefully reviewing this thesis, and my oral defense presentation. My sincere gratitude also goes out to my co-advisor Matthias Heinkenschloss for guiding and inspiring me in all matters related to optimization, partial differential equations and numerical analysis. I am grateful to him for his valuable advice and insightful criticisms in various aspects of my research. I thank Dr. Behr and Dr. Heinkenschloss for this wonderful opportunity, and would always cherish the many fruitful and stimulating discussions we have had.

The research in this thesis has been supported by the National Science Foundation under award ACI-0121360, CTS-ITR-0312764 and by Texas ATP grant 003604-0011-2001. The computing resources were provided by the NSF MRI award EIA-0116289.

I thank Dr. Dan Sorensen, Dr. Richard Tapia, Dr. John Dennis, Dr. Matteo Pasquali and Dr. Scott Collis for introducing me to their respective fields of research;
my course projects in the classes I took with them have significantly contributed towards my thesis objectives. My gratitude also goes to Dr. Yildiz Bayazitoglu, Dr. Scott Collis and Dr. Chad Landis who devoted their time to participate in my qualifier and final examining committees.

I thank my office-mate and friend Dhruv Arora who besides being a great source of encouragement and help, has been kind to share his extensive knowledge on various aspects of blood flow. I would also like to thank my fellow students—Vinod, Agata, Srinivas, Victor, Mehmet, Sushant and Josef—who have helped me in various ways. Vinod, with his strong grasp on computer systems, has been an enormous help at different times.

My friends have been a constant source of encouragement and support. In particular, I would like to thank the “adivaasi group”—Auleen, Soups and Vikram—with whom I have spent countless lunches, dinners, outings and have participated in random, yet pleasant, discussions over a range of issues from existentialism to love. My heart-felt gratitude also goes to Dhar, Malla, Tommy and Haria for their warm company and support. The prayers and fellowship I have enjoyed with Dr. Sarah Shearer and other friends from the First Christian Church have been a source of immense joy. I thank Ingar for her encouragement, support, understanding nature and prayers which have made these past few months a very comfortable and memorable one.

I thank my parents, sister and other family members for their love and affection. I am forever indebted to my parents and late grandmother for the many sacrifices they have made for my education, and for instilling in me an attitude to think big.
Contents

Abstract ii
Acknowledgments v
List of Illustrations xii
List of Tables xix

1 Introduction 1
  1.1 Overview .............................................. 3

2 The Governing Equations 5
  2.1 The governing equations of incompressible fluid flow .......... 5
  2.2 Constitutive equations ..................................... 7

3 Finite Element Formulations 9
  3.1 Introduction ............................................. 9
  3.2 Semi-discrete formulation .................................. 13
  3.3 Space-time formulation .................................... 15
  3.4 Stabilization parameter .................................... 17
  3.5 Solution method .......................................... 18

4 Stress Recovery and the Consistency of the GLS
   Stabilized Finite Element Formulation 20
5 Optimal Flow Control

5.1 Introduction ........................................... 42
5.2 Order of the discretization in the optimization problem ........ 45
5.3 Solution of the optimization problem ....................... 46
  5.3.1 Simultaneous Analysis and Design (SAND) .............. 46
  5.3.2 Nested Analysis and Design (NAND) ................... 47
5.4 Post-optimality sensitivity analysis ........................ 51
5.5 Model problem for code validation ........................ 52

6 The Effect of Stabilization in Finite Element Methods

for a Model Optimal Boundary Control Problem ............... 55

6.1 Introduction ........................................... 56
6.2 A model problem ...................................... 59
  6.2.1 Formulation of the optimal control problem ........... 59
  6.2.2 Discretization of the state equations .................. 65
  6.2.3 Discretization of the optimization problem ........... 66
6.2.4 Discretization of the optimality conditions .......................... 68
6.3 Implementation .................................................................. 70
  6.3.1 Stabilization terms ....................................................... 70
  6.3.2 Differences ................................................................. 71
6.4 Post-optimality sensitivity analysis for choice of stabilization parameter 72
6.5 Numerical results .............................................................. 73
  6.5.1 Test case 1 .................................................................. 74
  6.5.2 Test case 2 ................................................................. 82
6.6 Conclusions ...................................................................... 89

7 Shape Optimization for Steady Blood Flow and Non-Newtonian Effects 90
  7.1 Introduction .................................................................. 91
  7.2 A model shape optimization problem ................................. 93
  7.3 Numerical solution of the optimization problem ............... 94
    7.3.1 Discretization of the optimization problem .................. 94
    7.3.2 Gradient computation ............................................... 96
    7.3.3 Mesh sensitivity and adaptation ................................ 97
    7.3.4 Optimization algorithm ............................................. 97
  7.4 Numerical results ........................................................... 98
    7.4.1 Cannula optimization ............................................... 98
    7.4.2 Arterial grafting ...................................................... 106
  7.5 Conclusions ................................................................. 115

8 Shape Optimization for Unsteady Flows 116
8.1 Introduction ............................................. 116
8.2 A model shape optimization problem .................. 119
8.3 Numerical solution of the optimization problem: semi-discrete
    formulation ............................................. 121
    8.3.1 Discretization of the optimization problem .......... 122
    8.3.2 Discretization of the optimality conditions ........ 125
    8.3.3 Difference in gradients between the discretize-then-optimize
           (do) and optimize-then-discretize (od) approaches .... 127
8.4 Numerical solution of the optimization problem: space-time
    formulation ............................................. 129
    8.4.1 Discretization of the optimization problem .......... 129
    8.4.2 Discretization of the optimality conditions ........ 132
    8.4.3 Difference in gradients between the discretize-then-optimize
           (do) and optimize-then-discretize (od) approaches .... 133
8.5 Reducing storage requirements in adjoint-based gradient calculations 135
8.6 Arterial grafting ........................................ 136
    8.6.1 Case 1 ........................................... 137
    8.6.2 Case 2 ........................................... 143
8.7 Conclusions ............................................ 148

9 Conclusions and Future Directions 149
    9.1 Conclusions ......................................... 149
    9.2 Future directions ..................................... 151

A Implementation of the Semi-Discrete Formulation 153
B Implementation of the Space-Time Formulation 156

C Shape Sensitivity in the Semi-Discrete Formulation 160

D Shape Sensitivity in the Space-Time Formulation 163

Bibliography 166
Illustrations

3.1 Space-time discretization: concept of a space-time slab. .......................... 15

4.1 Computational domain for the test case 1. ........................................... 32
4.2 Horizontal velocity profiles using different stress recovery schemes for various values of stabilization factor $\alpha$ for test case 1. ....................... 32
4.3 The velocity error $E(u)$ as a function of refinement for test case 1. .. 33
4.4 Boundary stress $T_{12}$ on the bottom boundary $\Gamma_b$ for test case 1, after 1 non-linear iteration (left) and after 5 non-linear iterations (right). From top to bottom are discretizations 1 through 3. .......................... 34
4.5 Boundary stress error $E(T_{12})$ as a function of refinement for test case 1. 35
4.6 Computational domain for test case 2. ............................................. 37
4.7 The velocity error $E(u)$ as a function of refinement for test case 2. ... 38
4.8 Boundary stress $T_{12}$ comparison on the bottom boundary $\Gamma_b$ for test case 2, after 1 non-linear iteration (left) and after 5 non-linear iterations (right). From top to bottom are discretizations 1 through 3. The $x$-axis represents the angle $\theta$. .............................. 39
4.9 Boundary stress error $E(T_{12})$ as a function of refinement for test case 2. 40

5.1 Computational domain for flow in a channel with a bump. ............... 53
6.1 Computational domain for test case 1. ........................................ 75
6.2 Finite element mesh for test case 1, discretization 1. ...................... 75
6.3 The control velocities obtained for the two $\text{od}$ and $\text{do}$ approaches using the $\text{diag}$ and $\text{adv}$ element length definition with $\alpha = 1$ for test case 1. The ordering of these plots is $\mu = 0.0005$ on the left and $\mu = 0.00005$ on the right with discretizations 1 through 3 from top to bottom. ....................................................... 78
6.4 The control velocities obtained using different choices of $\alpha$ using $\text{diag}$ element length definition for $\mu = 0.00005$ in discretization 3 for test case 1. The plot on the left uses $\text{od}$ approach and the one on right uses $\text{do}$ approach. For comparison we also present $\text{adv}$ element length definition with $\alpha = 1$. ................................................................. 79
6.5 The weighted sensitivity $|g'(\alpha)_{ij}|/|g(\alpha)_{ij}|$ of the controls for the $\text{do}$ and $\text{od}$ approaches using the $\text{diag}$ and $\text{adv}$ element length definitions for $\mu = 0.00005$ in discretization 3 for test case 1, with $\alpha = 0.5$, $\alpha = 1.0$ and $\alpha = 2.0$. ......................................................... 80
6.6 The control velocities obtained using $\text{diag}$ element length definition for $\mu = 0.00005$ in discretization 3 for test case 1, obtained with $\text{od}$ and $\text{do}$ approaches and their variations. ............................... 81
6.7 Computational domain and boundary conditions for test case 2. In this case, $L = 10$ and $r = L/20$. ......................................................... 82
6.8 Finite element mesh for test case 2, discretization 1. ....................... 83
6.9 The control velocities obtained for the two od and do approaches using the diag and adv element length strategies for test case 2. The ordering of these plots is $\mu = 0.0005$ on the left and $\mu = 0.00005$ on the right with discretizations 1 through 3 from top to bottom.

6.10 The control velocities obtained using different choices of $\alpha$ using diag element length definition for $\mu = 0.00005$ in discretization 3 for test case 2. The plot on the left uses od approach and the one on right uses do approach. For comparison we also present adv element length definition with $\alpha = 1$.

6.11 The weighted sensitivity $|(g'(\alpha))_i|/(|g(\alpha)|_i|$ of the controls for the do and od approaches using the diag and adv element length definitions for $\mu = 0.00005$ in discretization 3 for test case 2, with $\alpha = 0.5$, $\alpha = 1.0$ and $\alpha = 2.0$.

6.12 The control velocities obtained using diag element length definition for $\mu = 0.00005$ in discretization 3 for test case 2, on experimenting with the terms of difference between the od and do approaches.

7.1 Cannula geometry

7.2 The initial and optimal cannula shapes using the Newtonian constitutive equation, $Re = 300$.

7.3 The initial and optimal cannula shapes using the Carreau-Yasuda constitutive model, $Re = 300$.

7.4 Initial shape and optimal shape of the center line for the Newtonian and Carreau-Yasuda constitutive model for the different choices of the Reynolds number.
7.5 Axial velocity plot at various locations of the initial shape of the cannula. Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the left subplot are scaled by a factor of 6 relative to the right subplot to emphasize differences in downstream velocity profiles. ................................................. 104

7.6 Axial velocity plot at various locations of the respective optimal shapes of the cannula. Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the left subplot are scaled by a factor of 6 relative to the right subplot to emphasize differences in downstream velocity profiles. ................................................. 105

7.7 Computational domain for the arterial graft. The initial shape of the design curve of the graft (dashed line) is a semi-circle with center at C. Fixed geometry parameters are $l_1 = 6.0$, $l_2 = 3.0$, $l_3 + d = 2.5$. The downstream section of the host artery is symmetric with respect to the upstream one. ......................................................... 106

7.8 The graft streamline velocity profile for the initial and the optimal shape, using the Newtonian constitutive equation for case 1, $Re = 300$. 108

7.9 The graft streamline velocity profile for the initial and the optimal shape, using the Carreau-Yasuda constitutive model for case 1, $Re = 300$. ................................................................. 109

7.10 Initial shape and optimal shape of the graft design curve for the Newtonian and Carreau-Yasuda constitutive model for case 1. .... 109
7.11 Axial velocity plot at various locations of the initial graft shape for case 1, with $Re = 50$ (top) and $Re = 300$ (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles. . . . . . . . . . . 110

7.12 Axial velocity plot at various locations of the optimal graft shapes using each of the constitutive equation (overlapped here), for case 1, with $Re = 50$ (top) and $Re = 300$ (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles. . . . . . . . . . . 110

7.13 The graft streamline velocity profile for the initial and the optimal shape using the Newtonian constitutive equation for case 2, $Re = 300$. 112

7.14 The graft streamline velocity profile for the initial and the optimal shape, using the Newtonian constitutive equation for case 2, $Re = 300$. 113

7.15 Initial shape and optimal shape of the graft design curve for the Newtonian and Carreau-Yasuda constitutive model for case 2. . . . . 113

7.16 Axial velocity plot at various locations of the initial graft shape for case 2, with $Re = 50$ (top) and $Re = 300$ (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles. . . . . . . . . . . 114
7.17 Axial velocity plot at various locations of the optimal graft shapes using each of the constitutive equation (overlapped here), for case 2, with \( Re = 50 \) (top) and \( Re = 300 \) (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles.

8.1 The function \( \Phi(u^h) \) as a function of time for the initial shape and the optimal shape, using the Newtonian constitutive equation for case 1.

8.2 The function \( \Phi(u^h) \) as a function of time for the initial shape and the optimal shape, using the Carreau-Yasuda constitutive equation for case 1.

8.3 Initial shape and optimal shape of the graft design curve for the Newtonian model for case 1.

8.4 Streamline profiles at \( t = 15.0, 15.25, 15.50, 15.75 \) and 16.0, (left to right, then top to bottom) for the initial shape using Newtonian constitutive equations, for case 1.

8.5 Streamline profiles at \( t = 15.0, 15.25, 15.50, 15.75 \) and 16.0, (left to right, then top to bottom) for the optimal shape using Newtonian constitutive equations, for case 1.

8.6 Streamline profiles at \( t = 15.0, 15.25, 15.50, 15.75 \) and 16.0, (left to right, then top to bottom) for the initial shape using Carreau-Yasuda constitutive equations, for case 1.
8.7 Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the optimal shape using Carreau-Yasuda constitutive equations, for case 1.  142

8.8 The function $\Phi(u^h)$ as a function of time for the initial shape and the optimal shape, using the Newtonian constitutive equation for case 2.  144

8.9 The function $\Phi(u^h)$ as a function of time for the initial shape and the optimal shape, using the Carreau-Yasuda constitutive equation for case 2.  145

8.10 Initial shape and optimal shape of the graft design curve for the Newtonian model for case 2.  145

8.11 Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the initial shape using Newtonian constitutive equations, for case 2.  146

8.12 Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the optimal shape using Newtonian constitutive equations, for case 2.  146

8.13 Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the initial shape using Carreau-Yasuda constitutive equations, for case 2.  147

8.14 Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the optimal shape using Carreau-Yasuda constitutive equations, for case 2.  147
Tables

6.1 Values of $h_e$ and $\tau_e$ for the element at which maximum $\nabla \cdot \mathbf{a}$ at element center is attained in test case 1. .......................... 76
6.2 Values of $h_e$ and $\tau_e$ for the element at which maximum $\nabla \cdot \mathbf{a}$ at element center is attained in test case 2. ......................... 83

7.1 Design variables for the cannula optimization at $Re = 300$. Note that $r_0$ and $r_{18}$ are dependent variables. ......................... 100
7.2 The design variables for the graft optimization for case 1, $Re = 300$. 107
7.3 The angle that the graft makes with the artery at its inlet and exit, for case 1. ................................................................. 108
7.4 The design variables for the graft optimization for case 2, $Re = 300$. 111
7.5 The angle that the graft makes with the artery at its inlet and exit, for case 2. ................................................................. 112

8.1 The design variables for the graft optimization for case 1. ............ 138
8.2 The angle that the graft makes with the artery at its inlet and exit, for case 1. ................................................................. 138
8.3 The design variables for the graft optimization for case 2. ............ 143
8.4 The angle that the graft makes with the artery at its inlet and exit,

for case 2. .............................................................. 144
Chapter 1

Introduction

Statistics by the American Heart Association identify cardiovascular disease as being the principal cause of morbidity and mortality in the western world. Advances in medicine and biomedical engineering provide hope to overcome these staggering statistics. Two of the major types of cardiovascular disease are atherosclerosis and heart failure. Atherosclerosis involves the agglomeration of fatty substances, cholesterol, and other deposits in the inner lining of an artery, which result in reduced blood flow and other pathological complications. The use of a graft, i.e., a bypass conduit, as an alternative route around critically blocked arteries is common clinical practice. Due to the inavailability of natural blood vessels, prosthetic grafts with similar physical and chemical properties as a real artery, are being actively researched [1]. Heart failure is the condition in which the heart loses its effectiveness as a pump. Currently, ventricular assist devices (VADs) provide a short- and medium-term solution until a transplant is found. The goal of ongoing blood pump research is to provide a long-term solution, or with enough progress, even a permanent clinical alternative to heart transplant [2].

For the analysis of blood flow within prosthetic devices, computational fluid dynamics (CFD)—the study of the numerical solutions of the governing equations of fluid dynamics—has become a valuable complement to the experimental approach. CFD has been extensively used in simulating and analyzing fluid flow in a variety of applications, especially in the realm of aerospace engineering. This progress, along
with rapid developments in computer technology, resulted in establishing favorable approximation properties of CFD methodologies over a range of complex geometries and flow conditions. The extension of the traditional CFD approaches to account for the non-Newtonian nature of blood has been a subject of active research. The CFD-based approach can provide prosthetic device engineers with a better understanding of the hemodynamics of the system, and point out both adverse and favorable hemodynamic features in the design. Having defined a quantitative measure, or objective function, representing design effectiveness, the engineer then strives to solve for the best design that optimizes the objective. This may be posed as an optimal design problem, which can then be solved using the CFD approach in conjunction with optimal control theory and numerical optimization techniques.

Motivated by the aforementioned applications in biomedical engineering, this thesis studies the challenges and issues that arise in obtaining a consistent numerical approximation to the optimal design problem. With this objective, we discuss two challenging aspects of these biomedical design endeavors. Firstly, we study the effect of blood-specific constitutive behavior on the solution of the optimal design problem. Secondly, we present a way to incorporate unsteady effects into the design process, with an emphasis on handling large-scale design problems. We use stabilized finite element (FE) formulations to discretize the equations governing blood flow. We discuss the weak consistency property of standard stabilized FE implementations using low-order elements, and propose an algorithm that establishes a better consistency and hence accuracy.

There exist multiple approaches to convert the infinite dimensional optimal design problem to a finite dimensional one amenable to numerical solution. These approaches differ in the order of the application of optimal control theory and the
discretization to the problem. It is not clear a priori which of these approaches consistently approximates the exact solution. Furthermore, the differences between the solutions obtained from the different approaches are dependent on the particular discretization scheme. We investigate this issue numerically for a particular class of optimal control problems, which closely resemble our target optimal design problem.

1.1 Overview

In Chapter 2, we introduce the governing equations of fluid flow used in this thesis, and state the constitutive equation which accounts for the generalized Newtonian behavior of blood. We assume that the fluid is incompressible, and in the laminar regime.

In Chapter 3, following a general background and introduction to stabilized and space-time finite element methods, we present finite element formulations used in this thesis.

In Chapter 4, we discuss the issue of residual incompleteness leading to weak consistency in our finite element formulations, when using low-order elements. Following a review of previous efforts to address this issue, we elaborate on a new approach to strengthen the consistency, and numerically investigate its performance.

In Chapter 5, we review optimal control problems in fluid dynamics, and present an overview of the issues and approaches in optimal flow control relevant to this thesis.

In Chapter 6, we study the effect of stabilization on the finite element discretization of optimal control problems governed by the linear Oseen equations. Control is applied in the form of suction or blowing on part of the boundary. We present various approaches to discretize the optimal control problem, and numerically inves-
tigate the differences in the computed control obtained using these approaches. We also introduce diagnostic tools to assess the quality of the computed control.

In Chapter 7, we investigate the influence of generalized Newtonian constitutive behavior of blood on the outcome of shape optimization tasks for stationary flows.

In Chapter 8, we present a methodology to solve unsteady optimal design problems, and address the various issues that arise in its application. We also show how the constitutive equation choice for blood affects the solution of the unsteady optimal design problem.

In Chapter 9, we summarize the conclusions drawn from the various issues addressed in this thesis, and point out possible directions for future research.
Chapter 2

The Governing Equations

In this chapter, we state the governing equations of a viscous fluid flow, i.e., the incompressible Navier-Stokes equations, allowing for a potential generalization for non-Newtonian fluid. We state the governing equations of fluid flow and the associated boundary and initial conditions in Section 2.1. In Section 2.2, we discuss some of the constitutive models used in this thesis to account for the generalized Newtonian behavior of fluids.

2.1 The governing equations of incompressible fluid flow

We consider a viscous incompressible fluid occupying a bounded region \( \Omega \subset \mathbb{R}^{n_{sd}} \), where \( n_{sd} \) is the number of space dimensions. Let \( \Gamma \) denote the boundary of \( \Omega \). The symbols \( u(x,t) \) and \( p(x,t) \), respectively, represent the velocity and pressure of the fluid. We consider the Navier-Stokes equations, governing the flow of unsteady and incompressible flows:

\[
\rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u - f \right) - \nabla \cdot \sigma = 0 \quad \text{on} \quad \Omega \times [0,T],
\]

(2.1a)

\[
\nabla \cdot u = 0 \quad \text{on} \quad \Omega \times [0,T],
\]

(2.1b)

where \( \rho \), assumed to be constant, represents the density of the fluid, and \( f(x,t) \) represents the external force, such as gravitational or electromagnetic field. The closure to (2.1) is obtained by relating the stress tensor \( \sigma \) to velocity and pressure fields; the constitutive equation will be discussed in Section 2.2. The boundary \( \Gamma \) is
decomposed into two disjoint segments $\Gamma_g$ and $\Gamma_h$. The velocity is assumed to be specified at the Dirichlet boundary $\Gamma_g$:

$$u(x, t) = g \quad \text{on} \quad \Gamma_g \times [0, T]. \quad (2.2)$$

The stress components are specified at the Neumann boundary $\Gamma_h$:

$$n \cdot \sigma(u, p) = h \quad \text{on} \quad \Gamma_h \times [0, T]. \quad (2.3)$$

The initial condition consists of a divergence-free velocity field:

$$u(x, 0) = u_0(x), \quad \nabla \cdot u_0 = 0, \quad \text{on} \quad \Omega. \quad (2.4)$$

Some of the boundary condition sets used in this thesis are:

- No-slip boundary conditions, used to specify all components of the velocity as zero for solid obstacles and walls.

- Inflow boundary conditions, used to specify all components of the velocity at inflow boundaries.

- Outflow boundary conditions, used to specify the stress components at outflow boundaries. Usually, at these boundaries a homogeneous or traction-free version of (2.3) is applied.

The momentum equation (2.1a) represents a nonlinear advective-diffusive system. The pressure variable plays the role of the Lagrange multiplier enabling the incompressibility constraint (2.1b) to be satisfied. The nonlinear nature of the coupled system (2.1) make it very challenging from the numerical standpoint. The relative dominance of the advection and diffusion forces is usually represented by the Reynolds number, a non-dimensional quantity. Physically, this number represents
the ratio of the inertial forces, due to the advective term, and the viscous forces. When the Reynolds number is low, e.g., for highly viscous fluids and creeping flows, viscous forces strongly dominate over the advective forces. Neglecting the advective term altogether, gives rise to the Stokes equations. The linearization of the advective term, on the other hand, gives rise to the Oseen equations. These approximations are widely used in the low Reynolds number flow regime. The other end of the spectrum—high Reynolds number and turbulent flows—is not considered in this thesis.

2.2 Constitutive equations

The rate of strain tensor for the fluid is defined as:

$$\varepsilon(u) = \frac{1}{2} \left( \nabla u + \nabla u^T \right).$$  \hspace{1cm} (2.5)

For a Newtonian fluid, the deviatoric component of the stress tensor is assumed proportional to the strain tensor. Water and most gases are examples of such fluids. Under the Newtonian assumption, the stress tensor can be written as:

$$\sigma(u, p) = -pI + T,$$

$$T = 2\mu\varepsilon(u).$$ \hspace{1cm} (2.6)

Here \(\mu\) is the constant dynamic viscosity, and \(I\) denotes the identity tensor. Other classes of fluids exhibit a dependence of the viscosity \(\mu\) on strain rate. For shear-thinning fluids, such as blood, the viscosity decreases with increasing strain rate whereas for shear-thickening fluids, the viscosity increases with increasing strain rate. The constitutive relation (2.6) can be generalized to account for this dependence. A variety of models have been proposed for this generalization. Let \(\dot{\gamma}\) represent the
square root of the second-invariant of the rate of strain tensor:

\[ \dot{\gamma} = \sqrt{2 \varepsilon^T \varepsilon(\mathbf{u})}, \]  

(2.7)

Two of the models considered here are

- Power-law model:

\[ \mu(\dot{\gamma}) = K \dot{\gamma}^{n-1}, \]  

(2.8)

where \( K \) and \( n \) are the parameters of the model.

- Carreau-Yasuda model:

\[ \mu(\dot{\gamma}) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1 + (\dot{\gamma}/\dot{\gamma}_c)^a)}, \]  

(2.9)

where \( \mu_\infty \) and \( \mu_0 \) are the infinite shear rate limit viscosity and the zero shear rate viscosity, respectively, and \( \lambda, a \) and \( b \) are the parameters of the model.

The power-law model suffers from its inability to describe the low-shear regime. When \( n \) is less than 1, \( \mu \) tends to infinity rather to a constant \( \mu_0 \), as is observed experimentally [3]. An additional level of non-linearity is induced into (2.1) when the generalized Newtonian constitutive equations, as given above, are used. Other classes of fluids, include viscoelastic flows where the stress is not only related to the rate of strain, but also depends on the strain history. We do not consider viscoelastic fluids in this thesis.
Chapter 3

Finite Element Formulations

In this chapter, we present finite element formulations for the discretization of the governing equations of incompressible fluid flow. We start with a general background and introduction to stabilized finite element methods and space-time finite element methods in Section 3.1. In Section 3.2, we present the semi-discrete formulation, which uses a finite element discretization in space and a finite difference discretization in time. In Section 3.3, we present the space-time formulation which uses a finite element discretization in both space and time. In Section 3.4, we discuss the design of stabilization parameters for the stabilized finite element formulations in Sections 3.2 and 3.3. We discuss the formation and solution of the discrete system of equations resulting from these finite element formulations in Section 3.5.

3.1 Introduction

The finite element method is a valuable tool for the solution of many engineering problems governed by partial differential equations (PDEs). It is especially attractive for problems dealing with complex geometries, where other numerical methods are difficult to apply. The use of finite element methods for the solution of the governing equations of fluid flow has become common, and many research and commercial software packages for the solution of fluid flow problems make use of finite element methods. The finite element method constructs a discretization of a weighted resid-
ual formulation to arrive at a system of matrix equations. This discretization can be applied both in space and time, yielding the so-called space-time finite element method. Alternatively, one can construct a discretization in space only, and use a finite differencing scheme for time. Such an approach, summarized as the finite element in space/ finite difference in time scheme, is referred to as the semi-discrete formulation. Our use of the terminology *semi-discretization* follows [4–6]. We caution that semi-discretization frequently refers to the situation in which the problem is only discretized in space or only discretized in time (see, e.g., [7]).

The Galerkin method, the most common weighted residual method, chooses the weighting and interpolation function from the same class of functions. In the context of incompressible fluid flow problems, the Galerkin finite element method suffers from two sources of potential instability. The first of these is common to advective-diffusive systems. The Galerkin method, when applied to an advective-diffusive system, gives rise to a central-difference-type operator, and exhibits a level of diffusion less than that of the physical problem. Therefore, it can be said that the Galerkin method introduces a negative diffusion into the numerical approximation of the problem. In advection dominated flows, spurious node-to-node oscillations are observed in the solution for the advected quantity. These oscillations are seen particularly in the vicinity of high-gradient regions such as boundary layers. Not surprisingly, such oscillations have been also observed when using central-difference finite difference schemes. The second source of instability is unique to mixed methods. The function spaces representing the velocity and pressure must be compatible, i.e., satisfy the inf-sup condition due to Babuška and Brezzi [8, 9]. Unfortunately, many desirable function spaces are precluded because of the inf-sup condition. If appropriate function spaces are not chosen, then oscillatory behavior is observed, primarily in
the pressure field. Disappointingly, most computationally attractive combinations which involve equal order interpolations for the velocity and pressure, do not satisfy the inf-sup condition. Moreover, those combinations which satisfy the inf-sup condition often employ basis functions which are not found in most commonly used engineering codes. This makes these compatible combinations inconvenient from an implementational standpoint.

A possible remedy for the aforementioned instabilities is to use significant mesh refinement for advection-dominated flows, and to always choose function spaces which satisfy the inf-sup condition. An alternative is to introduce an optimal amount of dissipation into the formulation in the advection-dominated regime, and to use methods which circumvent the inf-sup condition. The Streamline Upwind/Petrov-Galerkin (SUPG) technique was introduced by Brooks and Hughes [10], and is widely used in the fluid dynamics community. The SUPG stabilization for incompressible flows is achieved by adding to the Galerkin formulation a series of terms, each in the form of an integral over a different element. These integrals involve the product of the momentum equation and the advective operator acting on the test function [11]. The SUPG method preserves the consistency property of the Galerkin methods, i.e., the stabilized formulation is satisfied by the exact solution. This is because the added terms involve the residual of the momentum equation as a factor, and vanish for the exact solution. A consistent approach to circumvent the inf-sup condition for the Stokes problem was introduced in [12]. This was achieved by perturbing the Galerkin weighting function that multiplies the momentum balance residual, leading again to a Petrov-Galerkin type formulation. A generalization to that work for finite Reynolds number flows led to the Pressure-Stabilized/Petrov-Galerkin formulation (PSPG) developed by Tezduyar et al. [13]. The coefficients of the PSPG formulation
vary with the Reynolds number, in the same way as for the SUPG stabilization. In the zero Reynolds number situation the PSPG formulation reduces to the one proposed by Hughes et al. [12].

The Galerkin/Least-Squares (GLS) method was introduced by Hughes et al. [14] for advective-diffusive systems. In the GLS approach, a term with the residual of the momentum equation weighted by the variation of the momentum equation is added to the original Galerkin method. This stabilizing term is obtained by minimizing the sum of the squared residuals of the momentum equation integrated over each element domain. Since the stabilizing terms involve the residual of the momentum equation, the stabilized formulation satisfies the consistency property. The GLS stabilized method combines the essential ingredients of the SUPG and PSPG stabilized formulations. For time-dependent problems, a strict implementation of the GLS stabilization technique requires the finite element discretization to be both in space and time, i.e., the space-time formulation.

The space-time finite element formulation has been successfully used in conjunction with GLS stabilization for various problems with fixed spatial domain [4,15,16]. These references also present the basics of the space-time formulation, its implementation and associated error estimates. The discontinuous-in-time space-time methods possesses such advantages as: potential for selective temporal refinement, rigorous convergence bounds, and extensibility to moving domains. The potential to simulate flows with moving boundaries or interfaces is perhaps one of the most striking facets of these methods. The Deforming-Spatial-Domain/Space-Time procedure introduced by Tezduyar et al. [5,17], and the streamline-diffusion method introduced by Hansbo et al. [16,18], serve this purpose. Many applications such as incompressible flows past oscillating airfoils and cylinders [19,20], free-surface flows [21–23],
flow over rotating geometries [24–26], to name a few, have been made possible by using this procedure. A pitfall associated with space-time finite element formulations is their computational cost. Dettmer et al. [27] compare various time-integration schemes associated with semi-discrete and space-time finite element formulations for fixed domain problems.

3.2 Semi-discrete formulation

We first present the semi-discrete formulation. Recall that our use of the terminology semi-discretization follows [4–6] and refers to the situation in which we discretize the governing equations (2.1) in space and use a finite difference method for the discretization in time. To discretize the governing equations (2.1) in space, we apply a stabilized finite element discretization using piecewise linear interpolation functions for the velocity and the pressure. Let \( \{ \Omega^e \mid e = 1, 2, \ldots, n_{ed} \} \) be a discretization of \( \Omega \). We define:

\[
H^{1h}(\Omega) = \{ \phi_h \mid \phi_h \in C^0(\Omega), \phi_h|_{\Omega^e} \in P^1, e = 1, 2, \ldots, n_{ed} \},
\]

\[
S^h_u = \{ u_h \mid u_h \in [H^{1h}(\Omega)]^{n_{sd}}, u_h = g^h \text{ on } \Gamma_g \},
\]

\[
Y^h_u = \{ u_h \mid u_h \in [H^{1h}(\Omega)]^{n_{sd}}, u_h = 0 \text{ on } \Gamma_g \}.
\]

(3.1)
The GLS-stabilized finite element discretization of (2.1) is given as follows: find $u^h \in S^h_u$ and $p^h \in H^{1h}(\Omega)$, such that $\forall w^h \in V^h_u$ and $\forall q^h \in H^{1h}(\Omega)$:

$$
\int_{\Omega} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) \, dx + \int_{\Omega} \varepsilon(w^h) : \sigma(u^h, p^h) \, dx \\
+ \sum_{e=1}^{n_e} \int_{\Omega_e} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right] \cdot \\
\left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) - \nabla \cdot \sigma(u^h, p^h) \right] \, dx \\
+ \int_{\Omega} q^h \nabla \cdot u^h \, dx = \int_{\Gamma_h} w^h \cdot n^h \, dx.
$$

(3.2)

Our choice of piecewise-linear functions makes our discrete approximation a low-order method, i.e., the order of the function space is lower than the order of the highest derivative in (3.2). Therefore, we drop the second-order terms in (3.2). This guarantees only a weak consistency [28], since the stabilization term itself vanishes with refinement as $\tau_e$ approaches zero. A variational reconstruction of higher-order terms can be done to achieve a stronger consistency for such low-order function spaces [28]. We will discuss this issue further in Chapter 5. The implemented version of (3.2) is given by: find $u^h \in S^h_u$ and $p^h \in H^{1h}(\Omega)$, such that $\forall u^h \in V^h_u$ and $\forall q^h \in H^{1h}(\Omega)$:

$$
\int_{\Omega} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) \, dx + \int_{\Omega} \varepsilon(w^h) : \sigma(u^h, p^h) \, dx \\
+ \sum_{e=1}^{n_e} \int_{\Omega_e} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) + \nabla q^h \right] \cdot \\
\left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) + \nabla p^h \right] \, dx \\
+ \int_{\Omega} q^h \nabla \cdot u^h \, dx = \int_{\Gamma_h} w^h \cdot n^h \, dx.
$$

(3.3)

We use the generalized midpoint rule for the time-discretization of $u^h$, which can be written as:

$$
u^h = \gamma u_i + (1 - \gamma) u_{i+1}, \quad \frac{\partial u^h}{\partial t} = \frac{u_{i+1} - u_i}{\Delta t},
$$

(3.4)
Figure 3.1: Space-time discretization: concept of a space-time slab.

where $0 \leq \gamma \leq 1$, and $u_i$ represents the velocity field in the domain at time step $i$. The time derivative of the velocity weighting function, $\partial w^h/\partial t$, represents the variation of time-derivative of the velocity. In the case of the generalized midpoint rule with time-step $\Delta t$ (3.4) with $u_i$ known, this term becomes simply $w^h/\Delta t$.

### 3.3 Space-time formulation

We now consider the discontinuous-in-time space-time finite element formulation. The time interval $(0, T)$ is divided into subintervals $I_i = (t_i, t_{i+1})$, where $t_i$ and $t_{i+1}$ belong to an ordered series of time levels $0 = t_0 < t_1 < \ldots t_U = T$. Let $\Omega_i = \Omega_{t_i}$ and $\Gamma = \Gamma_{t_i}$, represent the time-dependent domain; then the space-time slab $Q_i$ is defined as the domain enclosed by surfaces $\Omega_i$, $\Omega_{i+1}$ and $P_i$—the surface described by the boundary $\Gamma_t$ as $t$ traverses $I_i$. As with $\Gamma_t$, surface $P_i$ can be decomposed into disjoint segments $(P_i)_g$ and $(P_i)_n$. This concept is shown in Figure 3.1. Finite element discretization of a space-time slab $Q_i$ is achieved by dividing it into elements.
\{Q^e_i \mid e = 1, 2, \ldots, (n_{el})_i\}, where \( (n_{el})_i \) is the number of elements in the space-time slab \( Q_i \). We define the following finite element interpolation spaces for the velocity and pressure on each space-time slab:

\[
H^{1h}(Q_i) = \{ \phi^h \mid \phi^h \in C^0(Q_i), \phi^h|_{Q^e_i} \in P^1, e = 1, 2, \ldots, (n_{el})_i \},
\]

\[
(S_u^h)_i = \{ u^h \mid u^h \in [H^{1h}(Q_i)]^{n_{el}}, u^h = g^h \text{ on } (P_i)_g \},
\]

\[
(V_u^h)_i = \{ u^h \mid u^h \in [H^{1h}(Q_i)]^{n_{el}}, u^h = 0 \text{ on } (P_i)_g \}. \tag{3.5}
\]

The GLS-stabilized space-time formulation can be written as follows: given \( (u^h)_i^- \), find \( u^h \in (S_u^h)_i \) and \( p^h \in H^{1h}(Q_i) \), such that \( \forall \ w^h \in (V_u^h)_i \) and \( \forall \ q^h \in H^{1h}(Q_i) \):

\[
\int_{Q_i} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) \, dQ + \int_{Q_i} e(w^h) : \sigma(u^h, p^h) \, dQ
\]

\[
+ \sum_{e=1}^{(n_{el})_i} \int_{Q^e_i} \rho \left[ \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right].
\]

\[
\left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) - \nabla \cdot \sigma(u^h, p^h) \right] \, dQ + \int_{Q_i} q^h \nabla \cdot u^h \, dQ
\]

\[
+ \int_{\Omega_i} (w^h)_i^+ \rho \left( (u^h)_i^+ - (u^h)_i^- \right) \, dx = \int_{(P_i)_h} w^h \cdot h^h \, dP. \tag{3.6}
\]

The notational convention used in (3.6) are:

\[
\int_{Q_i} \ldots \, dQ = \int_{t_i} \int_{\Omega_t} \ldots \, dx \, dt, \tag{3.7}
\]

\[
\int_{P_i} \ldots \, dP = \int_{t_i} \int_{\Gamma_t} \ldots \, dx \, dt. \tag{3.8}
\]

Our choice of piecewise-linear functions makes our discrete approximation a low-order method, as mentioned in Section 3.2. The implemented version of (3.6) is:

given \( (u^h)_i^- \), find \( u^h \in (S_u^h)_i \) and \( p^h \in H^{1h}(Q_i) \), such that \( \forall \ w^h \in (V_u^h)_i \) and
\( \forall q^h \in H^1(Q_i) : \)

\[
\int_{Q_i} \mathbf{w}^h \cdot \rho \left( \frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) dQ + \int_{Q_i} \varepsilon(\mathbf{w}^h) : \sigma(\mathbf{u}^h, p^h) dQ
\]

\[
+ \sum_{e=1}^{(n_e)} \int_{Q_i} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h \right) + \nabla q^h \right] dQ
\]

\[
\left[ \rho \left( \frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) + \nabla p^h \right] dQ + \int_{Q_i} q^h \nabla \cdot \mathbf{u}^h dQ
\]

\[
+ \int_{Q_i} (\mathbf{w}^h)^{+} \cdot \rho ((\mathbf{u}^h)^{+} - (\mathbf{u}^h)^{-}) d\mathbf{x} = \int_{(P_i)^h} \mathbf{w}^h \cdot \mathbf{h}^h dP. \tag{3.9}
\]

The solution to (3.9) is obtained for all space-time slabs \( Q_1, \ldots, Q_{\nu-1} \) sequentially, with

\[
(\mathbf{u}^h)^{+}_{1} = \mathbf{u}_0. \tag{3.10}
\]

### 3.4 Stabilization parameter

Our choice [4] for the stabilization parameter \( \tau_e \) is:

\[
\tau_e = \alpha \left[ \left( \frac{2|\mathbf{u}^h|}{h_e} \right)^2 + \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{4\mu}{\rho h_e^2} \right)^2 \right]^{-\frac{1}{2}}, \tag{3.11}
\]

where \( \alpha \) is a factor typically taken to be 1.0.

For the steady case, (3.11) reduces to:

\[
\tau_e = \alpha \left[ \left( \frac{2|\mathbf{u}^h|}{h_e} \right)^2 + \left( \frac{4\mu}{\rho h_e^2} \right)^2 \right]^{-\frac{1}{2}}. \tag{3.12}
\]

The element length \( h_e \) is computed using one of two choices denoted by \texttt{diag} and \texttt{adv}. Let \( h_e^* \) be the longest edge in an element. The \texttt{diag} element length is given by \( h_e = h_e^* \). The \texttt{adv} element length, introduced in [29], is defined as:

\[
h_e = \begin{cases} h_e^* & \text{if } |\mathbf{u}_e^h| = 0, \\ (\sum_a |\mathbf{s} \cdot \nabla N_a|)^{-1} & \text{otherwise}, \end{cases} \tag{3.13}
\]
where $u^h_e$ is the interpolated advection velocity at the center of element $e$, $s = u^h_e/\vert u^h_e\vert$, and $N_a$ represents the basis functions. Both these element length definitions are commonly used in the computation of $\tau_e$. Notice that the adv element length is identical to the diag element length in regions where there is stagnation in the advection field. In many flow scenarios with extended and frequent regions of stagnation, the diag element length is used to avoid abrupt switching of the element length type from streamline interpolation to longest edge of the element, as is the case in the adv element length definition.

### 3.5 Solution method

We can represent (3.2) or (3.6) symbolically as:

$$c^h = 0.$$  \hfill (3.14)

Further, let $c^h_u$ and $c^h_p$ represent respectively the contributions to $c^h$ due to the trial functions $w^h$ and $q^h$. The contributions at the nodal level for $c^h$ is given in Appendix A for the semi-discrete formulation and Appendix B for the space-time formulation. Gaussian quadrature [14] is used to evaluate the integrals in the spatial or space-time domains, which results in a system of first-order non-linear system of equations. This system of non-linear equation is solved using Newton's method. In each step of Newton’s method, we have to solve a linear system:

$$
\left( \begin{array}{c} \frac{\partial c^h_u}{\partial u} & \frac{\partial c^h_u}{\partial p} \\ \frac{\partial c^h_u}{\partial u} & \frac{\partial c^h_u}{\partial p} \\ \frac{\partial c^h_p}{\partial u} & \frac{\partial c^h_p}{\partial p} \\ \end{array} \right) \left( \begin{array}{c} \Delta u \\ \Delta p \\ \end{array} \right) = - \left( \begin{array}{c} c^h_u \\ c^h_p \\ \end{array} \right). \hfill (3.15)
$$

A linear solver must then be used to solve for the updates $(\Delta u, \Delta p)$ in (3.15). In our case, we use ILUT [30] preconditioned GMRES [31].
The second term in (3.3) contains \( \int_{\Omega} \varepsilon(w^h) : 2\mu\varepsilon(u^h) dx^h \). When the Carreau-Yasuda model is used, \( \mu \) depends on \( u \)—see (2.9). In this case, the evaluation of \( \frac{\partial \varepsilon^h}{\partial u^h} \) requires us to compute the derivative of \( \mu \) with respect to \( u^h \). In our computations we drop this derivative. This leads to an inexactness in the Jacobian \( \frac{\partial \varepsilon^h}{\partial u^h} \). While this inexact Jacobian may affect the convergence of the Newton’s method, we have found the performance of our Newton’s method satisfactory, and we choose therefore to avoid the additional implementation effort required to compute the exact Jacobian \( \frac{\partial \varepsilon^h}{\partial u^h} \). A similar situation holds for the space-time implementation (3.9).
Chapter 4

Stress Recovery and the Consistency of the GLS Stabilized Finite Element Formulation

Standard implementations of stabilized finite element methods often use function spaces of order lower than the highest order of derivatives present in the governing equations. Such weak consistency results in reduced accuracy at a given mesh refinement. To enforce a stronger consistency for such methods, variational reconstruction of the higher-order terms is possible, without increasing the order of the interpolation. In this chapter, we analyze the use of a low-cost post-processing approach, previously used to obtain better flux estimates, as a complement to the traditional variational reconstruction methods. In Section 4.1, we address the consistency issue of finite element formulations, the extent of the inconsistency when using low-order elements, and present a standard strategy to overcome this inconsistency. In Section 4.2, we discuss the global conservation property of the Galerkin method, and we point out a modification of the Galerkin method that can help establish this property. We see that in addition, the modified Galerkin method can determine fluxes at Dirichlet boundaries with superior convergence properties. In Section 4.3, we present an algorithm that converts such fluxes to superior stress estimates. These stresses are then used in conjunction with those computed by standard reconstruction techniques for the rest of the domain to improve the consistency of the stabilized FE implementations. In Section 4.4, we present numerical results for two test cases to justify the proposed algorithm. The computations are done using the GLS-stabilized
semi-discrete formulation for stationary flows. The results are however extensible to unsteady flows, space-time formulation and other stabilized finite element formulations. We present our conclusions for this chapter in Section 4.5.

4.1 Introduction

A numerical scheme is said to be consistent, if substituting the exact solution for the discrete solution satisfies the weak form. In our case, it would mean that the exact solution must satisfy (3.2) and (3.6). When using a stabilized finite element method, with a function space of order lower than the highest derivative in the PDE, the consistency of the numerical scheme is not guaranteed. We recall that the stabilization terms added to the Galerkin formulation are based on the complete residual of the weak form. Because the terms with the highest derivative cannot be approximated by the chosen lower-order function space, they are dropped. This issue has been addressed in [28, 32]. It is worthwhile to point out, that this deficiency is not resolved by mesh refinement, since the chosen function space is unable to recreate the dropped term. Higher-order methods, defined as methods employing a function space of order equal or greater than the highest derivative in the PDE, do not suffer from this problem. This is because the function space in this case is able to capture all the terms in the residual.

In the context of incompressible Navier-Stokes equations, we see from (2.1) and (2.6) that the divergence of the strain related term $\nabla \cdot \mathbf{T}$, also referred to as the diffusive flux, is the term with the highest (second) derivative on the velocity. We can observe in the variational formulation for the semi-discrete (3.2) and the space-time formulation (3.6) the presence of this diffusive flux term in the stabilization terms. Since piecewise linear function spaces for the velocity fail to capture this term, the
implemented version of (3.2) and (3.6) were (3.3) and (3.9) respectively. Both these implemented versions suffer from the problem of an incomplete residual. However, as pointed out by Jansen et al. [28], the situation is not so catastrophic. This is because the incomplete residual is multiplied by the stabilization factor $\tau_e$. From the definition of $\tau_e$ using (3.12), it can be observed that $\tau_e$ itself diminishes with mesh refinement and temporal refinement. So, in the limit of significant refinement in space and time, $\tau_e$ is driven to zero. Close to the limit, the stabilized FE formulation closely matches with the Galerkin method, and the consistency of the method is recovered. This is not entirely satisfactory, since very fine discretizations in space and time might be necessary to see an acceptably consistent behavior. It might be recalled from Chapter 3, that one of the reasons for the development of stabilized methods was to overcome significant mesh refinement as a way of avoiding instabilities in advection dominated flows.

Jansen et al. [28] propose to develop improved, low-order stabilized methods by globally reconstructing the dropped diffusive flux term, i.e., recovering nodal stress values. Using this reconstructed term, computed at negligible additional cost, in the element-level residual of a stabilized method helps establish better consistency properties. The recovered stress lags behind the computation of the velocity and pressure. A way to reconstruct these stresses is by the least-squares recovery approach, by solving for the $T^h$ that satisfies:

$$\min \int_{\Omega} (T^h - 2\mu \varepsilon(u^h))^2 \, dx.$$  

(4.1)

Denoting $t^h$ as the vector consisting of the independent terms of $T^h$, and similarly $q^h$ as the vector consisting of the independent terms of $\varepsilon(u^h)$, we define the function space:

$$\mathcal{V}_t^h = \{ w^h \mid w^h \in [H^1(\Omega)]^{n_{ad}(n_{ad}+1)/2} \},$$  

(4.2)
where due to the symmetry of the $\varepsilon(u^h)$, the number of the independent terms in $T^h$ is $n_{sd}(n_{sd} + 1)/2$. For example, if $n_{sd} = 2$, and $T^A$ represents the stress tensor for node $A$, then $T^A = (T_{11}^A, T_{12}^A, T_{22}^A)^T$. The problem (4.1) then becomes: find $t^h \in V^h_t$, such that $\forall w^h \in V^h_t$

$$\int_{\Omega} w^h : t^h \, dx = \int_{\Omega} w^h : 2\mu q^h \, dx. \quad (4.3)$$

This will result in the set of following matrix systems:

$$M\hat{t}_j = R_j, \quad j = 1, \ldots, n_{sd}(n_{sd} + 1)/2, \quad (4.4)$$

with

$$M = [M_{AB}], \quad R_j = \{R_{Aj}\}, \quad \hat{t}_j = \{t_{Aj}\} \quad (4.5)$$

and

$$M_{AB} = I \int_{\Omega} N_A N_B \, dx, \quad R_{Aj} = \int_{\Omega} 2\mu N_A q_j \, dx. \quad (4.6)$$

Here $I$ represents the identity tensor and $N_A, N_B$ represent the basis functions for nodes $A$ and $B$ respectively. The matrices are computed in the usual finite element matter. It is sufficient to lump the off-diagonal terms into the diagonal for the matrix system arising from (4.3) [28]. Representing by $M_L$, the lumped version of $M$, (4.4) is approximated by:

$$M_L\hat{t}_j = R_j, \quad j = 1, \ldots, n_{sd}(n_{sd} + 1)/2. \quad (4.7)$$

There are alternate formulations in which the diffusive part of the stress are treated as additional variables, such as the stress-velocity-pressure formulation of Behr et al. [33]. The linear interpolation of the stresses in that case ensures the consistency of the residual, and such methods are therefore not suffering from the weak consistency. However, the problem size increases considerably, from 3 to 6 degrees of freedom per node in 2D, and from 4 to 10 degrees of freedom in 3D,
making this approach unattractive, outside its intended application area, i.e., flows of viscoelastic fluids.

### 4.2 Consistent evaluation of boundary fluxes

Hughes et al. [34] discuss the global and local conservation properties of the continuous Galerkin method, and point out that global conservation requires that the weighting function whose value is precisely 1 throughout the domain be present in the weighting function space. This is the case only when Dirichlet boundary conditions are absent, since the strong enforcement of Dirichlet boundary condition necessitates the weighting functions to be zero at this portion of the boundary. In the case when only Neumann type boundary conditions are present, the global conservation structure of the continuous Galerkin method is easily established. A similar claim cannot be made when Dirichlet boundary conditions are present. Hughes et al. [34] introduce a modified (mixed) formulation with an auxiliary field, which amounts to the flux on the Dirichlet portion of the boundary, to remedy this problem. The modified formulation reduces to the usual continuous Galerkin method, plus a post-processing step to calculate the flux at the Dirichlet portion of the boundary. The fluxes calculated in this fashion help establish the missing link for global conservation and possess superior convergence properties. Oshima et al. [35] present the application of this approach to evaluate tractions for the Navier-Stokes equations in the context of Large Eddy Simulation (LES) for turbulent flows. To derive the post-processing equation for the semi-discrete GLS formulation (3.2), we introduce the function space:

$$V_u^h = V_u^h \oplus \text{span} \{N_A \}_{A \in \eta^*}$$  \quad (4.8)
where $\eta_g$ is the set of nodes in the discretization on $\Gamma_g$. We will solve for the unknown nodal fluxes at Dirichlet boundaries:

$$r^h = \sigma(u^h, p^h) \cdot n. \quad (4.9)$$

The new problem formulation is then: find $u^h \in S_u^h$ and $p^h \in H^1(\Omega)$, such that $\forall w^h \in V_u^h$ and $\forall q^h \in H^1(\Omega)$:

$$\int_{\Omega} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) \, dx + \int_{\Omega} \varepsilon(w^h) : \sigma(u^h, p^h) \, dx$$
$$+ \sum_{e=1}^{n_\Omega} \int_{\Omega_e} \tau_e \rho \left[ \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right] \, dx$$
$$+ \int_{\Omega} q^h \nabla \cdot u^h \, dx = \int_{\Gamma_h} w^h \cdot h^h \, dx; \quad (4.10a)$$

then find: $r^h \in V_u^h - V_u^h$, such that $\forall w^h \in V_u^h - V_u^h$ and $\forall q^h \in H^1(\Omega)$:

$$\int_{\Gamma_g} w^h \cdot r^h \, d\Gamma = \int_{\Omega} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) \, dx$$
$$+ \int_{\Omega} \varepsilon(w^h) : \sigma(u^h, p^h) \, dx + \int_{\Omega} q^h \nabla \cdot u^h \, dx - \int_{\Gamma_h} w^h \cdot h^h \, dx$$
$$+ \sum_{e=1}^{n_\Omega} \int_{\Omega_e} \tau_e \rho \left[ \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right] \, dx. \quad (4.10b)$$

Equations (4.10a) and (4.10b) can be solved separately, because the spaces $V_u^h$ and $V_u^h - V_u^h$ are independent.

### 4.3 Consistent GLS with consistent boundary fluxes

This chapter explores an approach to combine the flux obtained from (4.10) at Dirichlet boundaries, with the stresses obtained from least-squares recovery technique in
(4.3) in the rest of the domain. When using piecewise linear elements, the implemented version of (4.10) is then: find \( u^h \in S^h \) and \( p^h \in H^{1h}(\Omega) \), such that \( \forall w^h \in V^h_u \) and \( \forall q^h \in H^{1h}(\Omega) \):

\[
\int_{\Omega} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) dx + \int_{\Omega} \varepsilon(w^h) : \sigma(u^h, p^h) dx \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \rho \left[ \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right] + \nabla q^h dx \\
\left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) + \nabla p^h - \nabla \cdot T^h \right] dx \\
+ \int_{\Omega} \n \cdot u^h dx = \int_{\Gamma_h} w^h \cdot h^h dx; \tag{4.11a}
\]

then find: \( r^h \in V^h_u - V^h_u \), such that \( \forall w^h \in V^h_u - V^h_u \) and \( \forall q^h \in H^{1h}(\Omega) \):

\[
\int_{\Gamma_g} w^h \cdot r^h d\Gamma = \int_{\Omega} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) dx \\
+ \int_{\Omega} \varepsilon(w^h) : \sigma(u^h, p^h) dx + \int_{\Omega} q^h \n \cdot u^h dx - \int_{\Gamma_h} w^h \cdot h^h dx \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \rho \left[ \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right] + \nabla q^h dx. \tag{4.11b}
\]

The overall coupled consistent GLS with consistent boundary fluxes procedure is then, within each Newton-Raphson iteration:

1. Solve (4.11a) for \( u^h \) and \( p^h \).

2. Solve post-processing equation (4.11b) to obtain \( r^h \) over \( \Gamma_g \).

3. Solve (4.3) to obtain \( T^h \) over the entire domain.

4. Use \( r^h \) and \( T^h \) to find \( (T^h)_{\Gamma_g} \) and update \( T^h \), as elaborated in Section 4.3.1.

Remarks
1. The right hand side of equation (4.11b) involves the residual of the momentum equation. Therefore the accuracy of $r^h$ in (4.11b) is determined by the consistency of this residual. Consequently, a weakly consistent solution of (4.11a), will result in a lower accuracy of the computed fluxes in (4.11b). On the other hand, a method which solves (4.11a) in a consistent fashion, can result in better accuracy of the computed fluxes.

2. Another level of non-linearity is introduced into the computations due to the reconstruction of the diffusive fluxes. Furthermore, the stresses lag behind the solution for velocity and pressure. Thus even for Stokes equations, which result in a linear operator, we need to use an iterative non-linear solution procedure.

### 4.3.1 Obtaining consistent stresses from consistent fluxes

To simplify notation, we hereafter drop the superscript $h$, used to denote the discretized fields. An exact global reconstruction of the stresses based on the principles of (4.1) would involve the solution of the following problem:

$$
\min \int_\Omega (T - 2\mu \varepsilon(u))^2 \, dx
$$

such that

$$
T_{ij}^B n_j^B = r_i^B + \delta_{ij} p^B n_j^B, \quad \forall \ B \in \eta_B, \quad i, j = 1, \ldots, n_{\text{sd}}, \quad (4.12)
$$

where superscript $B$ represents nodal values at node $B$, and $n_j^B$ are components of the unit normal vector at node $B$. Solving (4.12) for $T$ as a constrained optimization problem exactly is expensive. An inexpensive approximation to (4.12) is to decouple the computation of the stresses on the Dirichlet boundaries from the computation of the stresses on the rest of the domain. To compute the stresses on the Dirichlet boundaries, one would solve:

$$
T_{ij}^B n_j^B = r_i^B + \delta_{ij} p^B n_j^B, \quad (4.13)
$$
for a given degree of freedom $i$ and a given node $B \in \eta_5$. However, (4.13) leads to an under-determined system since there exist $n_{sd}$ unknowns but only one equation for a given $i$. On the other hand, the least-squares recovery procedure (4.3) produces a good estimate of $T_{ij}^B t_j^B$ for 2D problems, and $T_{ij}^B t_j^B$ and $T_{ij}^B b_j^B$ for 3D problems, where $t_j^B$ and $b_j^B$ are the components of the unit tangent and unit bi-normal vectors at node $B$. These estimates are denoted by $T_{ij}^B t_j^B$ and $T_{ij}^B b_j^B$. Using this fact, one arrives at the following system:

\[
\begin{align*}
T_{ij}^B n_j^B &= r_i^B + \delta_{ij} p_j^B n_j^B, \\
T_{ij}^B t_j^B &= T_{ij}^B t_j^B, \\
T_{ij}^B b_j^B &= T_{ij}^B b_j^B,
\end{align*}
\]

(4.14)

for a given degree of freedom $i$ and a given node $B \in \eta_5$, which is sufficient to obtain all $n_{sd}$ components of $T_{ij}^B$ for that $i$ and $B$.

**Remarks**

1. When one or more boundary segments representing $\Gamma_s$ are aligned with coordinate axes, then $n_j$ takes on value either $\pm 1$ or 0, with $\sum_{j=1}^{n_{sd}} n_j^2 = 1$. Consequently, only the single non-zero component of $j$ is active in (4.13). In such a case, $T_{ij}$ can be isolated uniquely for that boundary segment, without having to resort to the approximation (4.14).

2. Note from (4.11b), that the boundary flux $r$ is interpolated using a finite element basis. Most applications involve intersection of boundaries with different boundary conditions applied on each. The intersections of these boundaries often involve a sharp corner or an edge, where the normals at such boundaries are not defined uniquely. For example, the normals are not defined at the four
corner points in a rectangular domain. In the context of the post-processing
equation (4.11b), this means that the value of \( r \) is not uniquely determined,
as discussed by Carey et al. in [36]. The techniques to resolve such ambigu-
ities in [36] do not fit well with our requirements for stress estimate at the
boundary nodes, and so we offer here an alternative. Let \( \eta_g^c \) represent the set
of corner nodes contained in \( \eta_g \). Then for any node \( A \in \eta_g^c \), we make use of
the unhumped least-squares recovery equation (4.4) to get:

\[
[M_{AA}][t_{Aj}] = \{R_{Aj}\} - [M_{AB}][t_{Bj}],
\]  

(4.15)

where \( B \neq A \). Note that, at this stage, stresses \( T \) for all the nodes on Dirichlet
boundaries except the corner nodes have been updated using (4.14). We would
therefore expect (4.15) to produce a better estimate \( T \) for the corner nodes than
those obtained from using just the least-squares recovery process (4.4) at these
nodes.

Our overall algorithm for the conversion of the consistent flux to consistent stress,
is then:

1. Identify \( \eta_g \) and \( \eta_g^c \).

2. Use (4.14) to determine \( T \) for the nodal set \( \eta_g - \eta_g^c \).

3. Use (4.15) to determine \( T \) for the nodal set \( \eta_g^c \).

4.4 Numerical results

We present two test cases to evaluate the proposed methodology. For both these
cases, our computations are based on the steady Stokes equations, with (3.12) as the
stabilization parameter using the adv element length definition. We also experiment
with a high value of the stabilization factor $\alpha$, which will amplify the inconsistency of the standard GLS stabilized implementation. Let us introduce the following short forms of the methods under consideration:

- **UGLS**: the standard low-order GLS formulation (3.3), with no approximation to reconstruct the higher-order derivatives.

- **CLGS**: the GLS formulation (4.11a), with reconstructed stress $\mathbf{T}$ obtained via least-squares recovery.

- **PGLS**: the GLS formulation (4.11a), with reconstructed stress $\mathbf{T}$ obtained via the least-squares recovery, coupled with the post-processing approach (4.11b) at the Dirichlet boundaries.

Further, we use the notation **CGLS $- i$** and **PGLS $- i$** to represent the CGLS and PGLS results, after $i$ non-linear iterations of the Newton's method. When $i = 1$, all results are equivalent to **UGLS**, since the reconstructed stresses lag behind by one iteration.

We also introduce two metrics: $E(T_{ij})$ defined as:

$$ E(T_{ij}) = \left( \int_{\Gamma_g} (T_{ij} - T_{ij}^a)^2 \, dx \right)^{1/2}, $$

(4.16)

where $T_{ij}^a$ represents the analytical stress on the boundary, and

$$ E(\mathbf{u}) = \left( \int_{\Omega} (\mathbf{u} - \mathbf{u}^a)^2 \, dx \right)^{1/2}, $$

(4.17)

where $\mathbf{u}^a$ represents the analytical velocity.

### 4.4.1 Test case 1

Consider a 2D channel flow problem with parabolic inflow and outflow profiles, and no slip at the walls. All the boundaries are thus Dirichlet for both velocity degrees
of freedom. The domain is a unit square shown in Figure 4.1. The inflow velocity is specified as \( \mathbf{u} = (y(1 - y), 0)^T \), with matching outflow profile. The analytical expression for \( T^a \), using \( \mu = 1.0 \) is:

\[
T^a = \begin{bmatrix}
T_{11}^a & T_{12}^a \\
T_{12}^a & T_{22}^a
\end{bmatrix} = \begin{bmatrix}
0 & 1 - 2y \\
1 - 2y & 0
\end{bmatrix}. \tag{4.18}
\]

Figure 4.2 show that the velocity profiles in the CGLS and PGLS formulations are insensitive to the the stabilization factor \( \alpha \). In order to better study the behavior of the reconstructed stresses with respect to mesh refinement, we use three discretizations, with equal mesh spacing in the \( x \)- and \( y \)-directions respectively. The domain is discretized using quadrilateral elements. Discretization 1 has 5 nodes in each direction resulting in element size \( h = 0.25 \) and a total of 25 nodes. Discretization 2 has element size \( h = 0.0625 \) and discretization 3 has element size \( h = 0.0156 \). Figure 4.3 shows the error in the solution \( E(\mathbf{u}) \) as a function of refinement.

We see from Figure 4.4 that PGLS clearly outperforms UGLS and CGLS in approximating the shear stress \( T_{12} \) at the bottom wall \( \Gamma_b \). Furthermore, we observe that at \( x = 0.0 \) and \( x = 1.0 \), which correspond to the corner nodes, \( T_{12} \) is approximated as well as for the interior nodes by CGLS \(-5\). Therefore, the corner correction (4.15), turns out to be a good choice. If we use no corner correction, but used the \( T_{ij} \) from least-squares recovery, then the error at these corners pollutes the \( T_{ij} \) approximations in the remainder of the boundary. Figure 4.5 shows the effect of mesh refinement on the error \( E(T_{ij}) \).
Figure 4.1: Computational domain for the test case 1.

Figure 4.2: Horizontal velocity profiles using different stress recovery schemes for various values of stabilization factor $\alpha$ for test case 1.
Figure 4.3: The velocity error $E(u)$ as a function of refinement for test case 1.
Figure 4.4: Boundary stress $T_{12}$ on the bottom boundary $\Gamma_b$ for test case 1, after 1 non-linear iteration (left) and after 5 non-linear iterations (right). From top to bottom are discretizations 1 through 3.
Figure 4.5: Boundary stress error $E(T_{ij})$ as a function of refinement for test case 1.
4.4.2 Test case 2

To demonstrate the applicability of the stress-recovery procedure in cases where the boundaries are not aligned with coordinate axes, we also consider flow in a circular pipe. The problem geometry is shown in Figure 4.6. The boundary conditions consist of a parabolic inflow profile at boundary $\Gamma_1$, and matching parabolic outflow at boundary $\Gamma_2$. Boundaries $\Gamma_b$ and $\Gamma_t$ are no-slip walls. The pipe has unit diameter and the centerline is an arc with radius 5.0. The domain is discretized using quadrilateral elements. For the given centerline, the nodal points of the discretization are obtained by moving in the radial direction from the discretized centerline by a constant distance. We choose three discretizations with discretization 1 having an element size $h = 0.25$ in both directions, i.e., radial and angular. Discretization 2 has element size $h = 0.125$ and discretization 3 has element size $h = 0.0625$. Figure 4.7 shows the error in the solution $E(u)$ as a function of mesh refinement. The results in Figure 4.8 study the effect of $T_{12}$ on the bottom wall $\Gamma_b$ using the different stress-recovery schemes. Figure 4.9 shows the boundary stress error $E(T_{ij})$ as a function of mesh refinement.
Figure 4.6: Computational domain for test case 2.
Figure 4.7: The velocity error $E(u)$ as a function of refinement for test case 2.
Figure 4.8: Boundary stress $T_{12}$ comparison on the bottom boundary $\Gamma_b$ for test case 2, after 1 non-linear iteration (left) and after 5 non-linear iterations (right). From top to bottom are discretizations 1 through 3. The x-axis represents the angle $\theta$. 
Figure 4.9: Boundary stress error $E(T_{ij})$ as a function of refinement for test case 2.
4.5 Conclusions and summary

We have presented a stress-recovery algorithm that improves the consistency of the GLS-stabilized finite element formulation. The reconstruction of the higher-order terms results in a much better consistency, and thus, accuracy, for relatively coarse meshes. We have used an approach that provides superconvergent fluxes at Dirichlet boundaries in combination with standard variational reconstruction methods, obtaining an algorithm which provides a low-order, yet strongly consistent discretization. For the two test cases considered, we have shown that the PGLS scheme shows a significant reduction in the error for relatively coarse practical meshes. Although, we performed our numerical evaluations using only the steady Stokes equations, we believe that these advantages will also hold for Navier-Stokes equations, as well as for unsteady flows.
Chapter 5

Optimal Flow Control

In this chapter, we review optimization problems in flow control. We start with an introduction to flow control in Section 5.1. We then discuss the various issues arising in the context of optimal flow control relevant to this thesis. In Section 5.2, we review two approaches that transform the continuous optimal control problem to a discrete one. In Section 5.3, we present some of the approaches to solve the discretized optimization problem. In Section 5.4, we present a methodology to evaluate the sensitivity of the solution of the optimal control problem with respect to certain simulation parameters. In Section 5.4, we present the model problem utilized as a first step towards the validation of the computational implementation of our optimization frameworks.

5.1 Introduction

The goal of flow control is to favorably alter selected aspects of a flow field to achieve a desired purpose [37]—typically an improvement in engineering design. For a partial summary of the progress to-date in flow control, one may refer to [38]. Having defined a measure to quantify the desired purpose, the goal of the designer is to find the control that provides the best possible value of this quantitative measure. Such a design task is pervasive across various fluid dynamic endeavors. For example, finding the levels of suction and blowing on a portion of an airplane wing that best reduces
the skin-friction drag, or finding the shape the shape of a left-ventricular assist
device that best reduces the stress levels endured by blood cells, falls in the realm
of flow control. The notion of "best reduction" can be translated into mathematical
language as a minimization or optimization problem. It must be noted that the
measure, or objective, that needs to be minimized is dependent on the flow variables,
thereby making this minimization problem constrained by the physics of the flow.
An inelegant solution is a so-called trial and error method, in which the quantitative
measure is simply evaluated for different values of the controls. The control values
which give rise to the best measure are then identified and selected. Besides lacking
robustness, this method offers no guarantee of arriving at a best solution. In contrast,
this thesis studies the subject of the solution of optimal flow control problems using
techniques of numerical optimization and control theory. Such techniques offer a
robust and automatic framework to arrive at the solution to an optimal flow control
problem, if such a solution exists.

The formulation of the optimal flow control problem proceeds as a three-step
process. First, one specifies the governing equations that determine the flow under
consideration. Second, a suitable objective function is chosen to represent the design
goals. Finally, a control method that is capable of delivering the desired goal is
selected.

We are interested in model optimal control problems based on the governing
equations of fluid flow equations (2.1). Before discretization (3.2) or (3.6), a weak
form of that problem is constructed by introducing the following function spaces:

\begin{align}
\mathcal{S}_u &= \{ u \mid u \in [H^1(\Omega)]^{n_x}, u = g \quad \text{on } \Gamma_g \}, \\
\mathcal{V}_u &= \{ u \mid u \in [H^1(\Omega)]^{n_{n_x}}, u = 0 \quad \text{on } \Gamma_g \},
\end{align}

(5.1)
where $H^1(\Omega)$ is defined in the usual way [39,40]. The weak form of (2.1) is then:

\[ \int_{\Omega} w \cdot \rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u - f \right) dx + \int_{\Omega} \varepsilon(w) : \sigma(u,p) dx + \int_{\Omega} q \nabla \cdot u dx = \int_{\Gamma_h} w \cdot h dx. \] (5.2)

We are interested in the solution of the following optimal control problem:

\begin{align*}
\text{minimize} & \quad J(v, y^h), \\
\text{subject to} & \quad c(v, y) = 0,
\end{align*} (5.3a)

where (5.3b) stands for the state equation (5.2). $v = (u(x), p(x))^T$ are the state variables, and $y$ is the control or design variables vector. The types of controls applied are:

- $g = g(y)$: boundary control problem, where the objective is achieved by suction or blowing of fluid through various parts of the boundary,

- $\Omega = \Omega(y)$: shape (geometry) control problem, where the objective is achieved by varying the shape of the domain.

- $f = f(y)$: distributed control problem, e.g., through a magnetic field acting on an ionized fluid or on a liquid metal [41].

Possible objective functions for flow control include viscous drag, vorticity in the domain, shear stress integral, etc. Often, the objective function $J(v, y)$ does not explicitly depend on the control $y$. This may result in unbounded controls, or controls exhibiting strong oscillations. This problem may be addressed by introducing an additional inequality constraint bounding the norm of the control and its gradient to (5.3). Alternatively, one may solve a penalized (regularized) objective functional,
formed by adding a regularization term of the control and its derivative to (5.3a). We make use of the latter approach in Chapter 6.

The problem (5.3) is an optimization problem constrained by a PDE. Such problems are not unique to flow control, and are found in other engineering disciplines. Therefore, the methods that we outline here for a particular class of PDE-constrained problems arising in flow control are equally applicable to other disciplines.

5.2 Order of the discretization in the optimization problem

Analytical solutions to the continuous problem (5.3) generally cannot be obtained, and so one must rely on approximate solutions. There are two ways in which the infinite dimensional problem is transformed into a finite dimensional one, which would be then amenable to numerical solution. In the first approach, called the optimize-then-discretize approach, one attempts to solve (5.3) by first forming the optimality conditions for this infinite dimensional problem and then discretizing the optimality system. In the second approach, one first discretizes the objective function and the state equation, and then applies optimization techniques to that discretized problem. This approach, called the discretize-then-optimize approach, involves the solution of:

\[
\begin{align*}
\text{minimize} & \quad J(v^h, y), \\
\text{subject to} & \quad c^h(v^h, y^h) = 0.
\end{align*}
\]

The optimality conditions are then derived for (5.4). In Chapter 6, we study the differences in the optimality system, and hence the optimal control problem, that arise when using one or the other of these approaches.
5.3 Solution of the optimization problem

5.3.1 Simultaneous Analysis and Design (SAND)

The constrained discretized optimization problem (5.4) can be recast as an unconstrained problem using the Lagrange multiplier method, by introducing the Lagrange multiplier, or the adjoint variables, \( \lambda^h \). The Lagrangian is then defined as:

\[
L(v^h, \lambda^h, y) = J(v^h, y) + (\lambda^h)^T c^h(v^h, y^h).
\] (5.5)

The first-order necessary optimality conditions require:

\[
\nabla L(v^h, \lambda^h, y^h) = 0.
\] (5.6)

This vanishing gradient condition leads to:

\[
\text{state equation: } \frac{\partial L}{\partial \lambda^h} = c^h(v^h, y^h) = 0, \quad (5.7a)
\]

\[
\text{adjoint equation: } \frac{\partial L}{\partial v^h} = J_v(v^h, y^h) + (\lambda^h)^T c_v^h(v^h, y^h) = 0, \quad (5.7b)
\]

\[
\text{gradient equation: } \frac{\partial L}{\partial y^h} = J_y(v^h, y^h) + (\lambda^h)^T c_y^h(v^h, y^h) = 0. \quad (5.7c)
\]

The equation set (5.7) represents a system of nonlinear equations, which can be solved using Newton’s method.

For convenience, we will drop the superscript \( h \), and the terms within brackets that denote the dependence on \( v^h \) and \( y^h \). The conceptual algorithm for the solution of (5.7) is:

1. Initialize \((v, \lambda, y)\).

2. While not converged:

\[
\text{update } v^h, \lambda^h, y^h \text{ using Newton's method.}
\]
(a) solve for the updates \((\delta v, \delta \lambda, \delta y)\) using:

\[
\begin{pmatrix}
c_v & 0 & c_y \\
J_{vv} + \lambda^T c_{vv} & c_v^T & J_{vy} + \lambda^T c_{vy} \\
J_{vy} + \lambda^T c_{vy} & c_y^T & J_{yy} + \lambda^T c_{yy}
\end{pmatrix}
\begin{pmatrix}
\delta v \\
\delta \lambda \\
\delta y
\end{pmatrix} = -
\begin{pmatrix}
c \\
J_v + \lambda^T c_v \\
J_y + \lambda^T c_y
\end{pmatrix}
\] (5.8)

(b) update the variables \((v, \lambda, y)\) using:

\[
\begin{pmatrix}
v \\
\lambda \\
y
\end{pmatrix} \leftarrow \begin{pmatrix}
v \\
\lambda \\
y
\end{pmatrix} + \begin{pmatrix}
\delta v \\
\delta \lambda \\
\delta y
\end{pmatrix}.
\] (5.9)

This approach, in which the simulation (analysis) variables and the optimization (design) variables are solved for simultaneously, is referred to as the Simultaneous Analysis and Design (SAND). Other names for SAND are one-shot methods and All-At-Once (AAO) methods.

One of the advantages of SAND is that we do not have to exactly solve for the state equation at each iteration. However, this reduction in computational cost is offset by the cost incurred in solving the system (5.8), since the number of the unknowns equals the number of the state variables, adjoint variables and the control variables taken together. In addition, a straightforward usage of existing analysis codes in an optimization framework is more difficult and sometimes impossible. Nonetheless, various efficient SAND-based methodologies are being developed for optimal control flow problems [42, 43].

### 5.3.2 Nested Analysis and Design (NAND)

As an alternative to SAND, one might prefer to work in the space of design variables alone. One could then employ the most suitable optimization algorithm for the
problem defined in this space. Using the implicit function $v^h(y^h)$, we eliminate the state variables and the PDE-constraints $c^h(v^h, y^h) = 0$. The modified optimization problem is:

$$\text{minimize } \tilde{J}^h(y^h) = J^h(v^h(y^h), y^h). \quad (5.10)$$

The Nested Analysis and Design (NAND) approach to solve the optimization problem (5.10) can be further divided into two categories: non-gradient-based and gradient-based. The non-gradient-based methods, also called zeroth-order methods, as the name suggests, involve only objective function evaluations, and no gradient or Hessian of the objective function. Gradient-based optimization algorithms involve both objective function and gradient evaluations. Both non-gradient-based and gradient-based methods have their advantages and disadvantages for certain classes of problems. In this thesis, we focus on gradient-based methods to solve (5.10).

For convenience, we again drop the superscript $h$, and the terms within brackets that denote the dependence on $v^h$ and $y^h$. The gradient of the modified objective function is given as:

$$\nabla \tilde{J}(y) = J_y + \left( \frac{dv}{dy} \right)^T J_v. \quad (5.11)$$

Using the implicit function theorem, we obtain the discrete sensitivity equation given by:

$$c_v \left( \frac{dv}{dy} \right) + c_y = 0. \quad (5.12)$$

Using (5.12) in (5.11), the gradient can be written as:

$$\nabla \tilde{J}(y) = J_y - [c_v^{-1} c_y]^T J_v \quad (5.13)$$

$$= J_y - c_y^T [c_v^{-T} J_v]. \quad (5.14)$$
In order to compute the derivatives in case of the sensitivity approach, one forms 
\[ \mathbf{c}_y^{-1} \frac{\partial \mathbf{c}}{\partial y_i} \] for each \( y_i \), and then computes the derivative from (5.13) for each of these \( n \) design variables. In the adjoint approach, one forms \[ \mathbf{c}_y^{-T} J_y \] once and computes the derivative from (5.14), making the latter approach favorable for problems with large number of design variables. The equation

\[ \mathbf{c}_y^T \lambda = -J_y^T \]

is called the (discrete) adjoint equation. Another way to evaluate the derivative \( \nabla \tilde{J}(y) \) is by utilizing the finite-difference approach to compute \( \frac{d\mathbf{v}}{dy} \). For example, using the one point forward difference scheme, the solution sensitivity \( \frac{d\mathbf{v}}{dy} \) is obtained as:

\[ \frac{d\mathbf{v}}{dy} \approx \frac{\mathbf{v}(y + \delta y_i \mathbf{e}_i) - \mathbf{v}(y)}{\delta y_i} \]

(5.15)

where \( \mathbf{e}_i \) is the unit vector with the non-zero element at index \( i \). Equation (5.15) is solved for each of the design variables \( y_i, i = 1, \ldots, n \). This method has several disadvantages:

1. The large cost of evaluating the derivatives (5.15), since for each of the control variables, one needs to solve two state equations.

2. The difficulty of selecting a suitable step-size \( \delta y_i \), which is problem-specific and not known a priori.

Another option is to use automatic differentiation software to solve for \( \frac{d\mathbf{v}}{dy} \) [44]. However, in this thesis we focus on analytical evaluation of the sensitivity using (5.13) or (5.14).

We can see the equivalence of the derivative (5.14) with that in (5.7c). This suggests that, if a decoupled iterative algorithm to solve the full optimality system (5.7)
was used, that method would be equivalent to the gradient-based NAND approach. In other words, we must solve (5.7c) with an optimization algorithm:

1. Initialize $y$.

2. While not converged:
   
   (a) solve for $v$ using (5.7a),
   
   (b) solve for $\lambda$ using (5.7b) or $\frac{dv}{dy}$ using (5.12),
   
   (c) evaluate $\nabla \hat{J}(y)$ using (5.14) or (5.13),
   
   (d) use $\nabla \hat{J}(y)$ to compute the update $\delta y$,
   
   (e) set $y = y + \delta y$.

In order to compute the control update $\delta y$, one can use any available gradient-based optimization method, such as a line-search method or a trust-region method. For a detailed description of these methods, the interested reader may refer to [45].

The convergence criteria for the optimization algorithm depends on how accurately one solves for $\hat{J}(y)$ and $\nabla \hat{J}(y)$, and also on the definition of satisfactory convergence. For example, methods which make use of incomplete sensitivities [46] and adjoints [47] solve for sensitivity and adjoint variables only in selected regions, which are expected to have most influence over the objective function gradient. In this thesis, we attempt to solve for the gradients exactly, and we clearly state if any of the terms are dropped.

NAND approach has been popular as a way to solve optimal flow control problems, due to its ease of adopting existing optimization frameworks to existing analysis codes. However, the computational expense can be quite high for highly-nonlinear
state equations, e.g., at high Reynolds numbers, since the state equations must be solved exactly, or nearly-so, at each iteration.

5.4 Post-optimality sensitivity analysis

After obtaining the optimal control $y$ for the problem (5.4), the sensitivity of this optimal control to some scalar simulation parameter may also be of interest. That simulation parameter may represent aspects of the numerical scheme, or of the physical model in the governing PDE. Let us denote by $\kappa$ the simulation parameter with respect to which we would like to evaluate this post-optimality sensitivity. The problem is then: given the optimal control $y$ and the corresponding state $v^h$ and adjoint $\lambda^h$, find how sensitive is $(v^h, \lambda^h, y)$ with respect to $\kappa$. The optimal control $y$ and the corresponding velocities and pressure satisfy:

the state equation:

$$c(v^h, y^h, \kappa) = 0, \quad (5.16a)$$

the adjoint equation:

$$J_v(v^h, y^h, \kappa) + \lambda^T c_y(v^h, y^h, \kappa) = 0, \quad (5.16b)$$

and the gradient equation:

$$J_y(v^h, y^h, \kappa) + \lambda^T c_y(v^h, y^h, \kappa) = 0. \quad (5.16c)$$

For convenience, let us hereafter drop the superscript $h$ in (5.16). With $\kappa$ being the parameter for which the optimal control $y(\kappa)$ has been computed, let $\kappa + \delta \kappa$ be another parameter, and examine $y(\kappa + \delta \kappa)$. If $\delta \kappa$ is sufficiently small, and if $y(\cdot)$ is differentiable, then:

$$y(\kappa + \delta \kappa) \approx y(\kappa) + y_\kappa(\kappa) \delta \kappa.$$
The directional derivative \( y_\kappa(\kappa)\delta\kappa \) can be computed as the solution of:

\[
\begin{pmatrix}
c_v & 0 & c_y \\
J_{vv} + \lambda^Tc_v & c_v^T & J_{vy} + \lambda^Tc_y \\
J_{vy} + \lambda^Tc_y & c_y^T & J_{yy} + \lambda^Tc_y
\end{pmatrix}
\begin{pmatrix}
v_\kappa(\kappa)\delta\kappa \\
\lambda_\kappa(\kappa)\delta\kappa \\
y_\kappa(\kappa)\delta\kappa
\end{pmatrix} = -\begin{pmatrix}
c_\kappa\delta\kappa \\
(J_{v\kappa} + \lambda^Tc_v)\delta\kappa \\
(J_{y\kappa} + \lambda^Tc_y)\delta\kappa
\end{pmatrix}
\]  \( 5.17 \)

Note that the system matrix in (5.17) is the same as the system matrix in (5.8).

In Chapter 6, we make use of the post-optimality sensitivity analysis to compute the sensitivity of the optimal boundary velocity control with respect to the stabilization parameter \( \alpha \) introduced in (3.12).

### 5.5 Model problem for code validation

As a first step towards validating the optimization code, while using any of the approaches discussed in this chapter, we utilize a test case where the optimal control \( y \) and the optimal objective \( J \) are known a priori. A flow matching problem attempts to recreate a target flow field, obtained using known design variables, through a minimization problem. When the flow variable being matched is the velocity, then this kind of problem is also referred to as the velocity tracking problem.

Following [48], we consider flow in a 2D channel with extents \( 0 \leq x \leq 10 \) and \( 0 \leq y \leq 3 \), with a bump on the lower wall extending from \( x = 1 \) to \( x = 3 \) as shown in Figure 5.1. The boundary conditions are: no-slip at the walls \( \Gamma_i \) and \( \Gamma_b \), a parallel flow condition at the outlet \( \Gamma_o \), and a parabolic inflow condition at the inflow \( \Gamma_i \) given by \( u = (g_0, y(3 - y), 0)^T \), where \( g_0 \) is a parameter determining the mass flow rate.

The shape of the bump is determined as a sum of Bezier polynomials [49]. We will use a degree 5 polynomial, with two of the coefficients being determined by the requirement that the bump continuously meet the straight channel wall \( \Gamma_b \). Therefore,
the bump profile \( y_B(x, \alpha) \) is:

\[
y_B(x, \alpha) = p(x) + \sum_{k=1}^{3} \alpha_k q_k(x), \quad \text{for} \quad 1 \leq x \leq 3,
\]  

(5.18)

where \( p(x) \) and \( q_k(x) \), for \( k = 1, 2, 3 \), are given functions and \( \alpha = (\alpha_1, \alpha_2, \alpha_3)^T \).

Figure 5.1: Computational domain for flow in a channel with a bump.

In the case of steady flows, we compute a target flow \( \hat{\mathbf{v}}^h \) by solving the governing equations, under consideration, with the parameters chosen as:

\[
\hat{g}_0 = 0.5, \quad \hat{\alpha} = (0.375, 0.5, 0.375)^T.
\]  

(5.19)

The objective function is:

\[
J^h(u^h, y) = \int_{\Omega_{\text{obs}}} (u^h - \hat{u}^h)^2 \, dx,
\]  

(5.20)

where, for a boundary control problem, \( y \) represents the scalar \( g_0 \), and for a shape control problem \( y = \alpha \). The objective function (5.20) measures the discrepancy between the velocity component of the discretized target flow \( \hat{u}^h \) resulting from target control \( \hat{\mathbf{v}} \), with a candidate flow \( u^h \) for a choice of the control \( y \), within the observation region \( \Omega_{\text{obs}} \). Note, that (5.20) represents the discretized functional.
We then attempt to solve (5.4) with objective functional (5.20), in order to approximately recreate the control $\hat{y}$ and the corresponding flow field $\hat{u}^h$. We proceed with this test case as a first step, for validating the optimization frameworks described in this thesis. The discretized governing equation for this problem, i.e., equation (5.3b) will be subsequently defined in the context of each of the following chapters. For the unsteady shape optimization used in Chapter 8, we will add a time integral to (5.20), and the fields $u^h$ and $\hat{u}^h$ will be functions of both space and time.
Chapter 6

The Effect of Stabilization in Finite Element Methods for a Model Optimal Boundary Control Problem

In this chapter, we study the effect of the GLS stabilization on the finite element discretization of optimal control problems governed by the linear Oseen equations. Control is applied in the form of suction or blowing on part of the boundary. We discuss the optimize-then-discretize and the discretize-then-optimize approaches to solve the optimal control problem. Both approaches lead to different discrete adjoint equations and, depending on the choice of the stabilization parameters and grid size, may significantly affect the computed control. Following the problem outline in Section 6.1, we give a precise statement of the optimal control problem in Section 6.2, as well as establish existence of a unique solution and formulate the first-order necessary and sufficient optimality conditions. We also recall the GLS stabilized finite element method for the discretization of the state equation, and describe the two approaches to discretize the optimal control problem. In Section 6.3, we state the implemented version of the discretized state equation and the adjoint equations arising from these two approaches. In Section 6.4, we introduce diagnostic tools that can guide the selection of sensible stabilization parameters. In Section 6.5, we illustrate the effect of the order in which the discretization is applied, and of the choice of the stabilization parameters, using two test problems. We present our conclusions for this chapter in Section 6.6.
6.1 Introduction

As mentioned in Chapter 3, stabilized finite element methods are frequently and successfully used to discretize advection-dominated PDEs [10], or to circumvent the compatibility conditions restricting the choice of interpolation function spaces [12]. One question that arises in the application of these methods is the choice of the stabilization parameter. For many PDE model problems this issue has been studied analytically and numerically [4,50]. The impact on the choice of the stabilization parameter when the stabilized FEM is used in the context of optimal control, however, is not well studied. It is not true that a scheme which gives good approximations to the PDE solution for a fixed simulation is also guaranteed to provide good approximations in the context of optimal control problems. The reason is that solutions of optimal control problems are characterized by the original governing PDE as well as another PDE, the so called adjoint equation. Depending on the approach chosen, a discretization of the original governing PDE may imply a discretization scheme for the adjoint equation with poor approximation properties. This chapter investigates this issue for a class of boundary control problems discretized using the GLS-stabilized FEM.

We consider a class of linear quadratic optimal control problems governed by the Oseen equations, and we study the effect of the GLS-stabilized finite element method on the computed control. The linear Oseen equations were chosen as the governing state equation instead of the nonlinear Navier-Stokes equations, because the resulting optimal control problem has a unique solution and the first-order optimality conditions are necessary and sufficient. Optimal control problems governed by the nonlinear Navier-Stokes equations may have local solutions and their solution requires iterative methods. Since it is difficult to separate the possible effects of local
solutions and iterative solvers on the computed optimal control from the effects of the stabilization on the computed optimal control, we have chosen the Oseen equations. However, the linear quadratic optimal control problems are closely related to the subproblems that arise in the solution of boundary control problems governed by the Navier-Stokes equations using Newton or sequential quadratic programming methods (see, e.g., [51, 52]). Consequently, the results reported in this chapter are also relevant for the optimal control of Navier-Stokes flow.

Collis et al. [53] study the effect of the Streamline-Upwind/Petrov-Galerkin stabilized finite element method on the numerical solution of linear quadratic distributed optimal control problems governed by an advection-diffusion equation. The paper [53] contains both analytical results that describe the convergence of the computed control (state/adjoint) to the exact control (state/adjoint) as the grid is refined as well as numerical convergence studies for the simpler optimal control problem. The analytical results in [53] are accurate asymptotically, but do not describe the numerical behavior well for first-order finite elements on relatively coarse, but for practical purposes acceptable grids. Therefore, this chapter focuses on a numerical study. We expect that most of the analytical results in [53] can be extended to the problems and the GLS stabilization method considered in this paper. However, such a theoretical treatment is beyond the current scope.

We consider a viscous incompressible fluid occupying a bounded region \( \Omega \subset \mathbb{R}^{n_{sd}} \), where \( n_{sd} \) is the number of space dimensions. The symbols \( \mathbf{u} \) and \( p \) represent the velocity and pressure. The boundary \( \partial \Omega \) of \( \Omega \) is decomposed into three disjoint segments \( \Gamma_h, \Gamma_d, \Gamma_g \). Suction and blowing control is applied on \( \Gamma_g \). We are interested in the solution of the following problem.
Minimize:

\[ J(u, g) = J_1(u) + \frac{\beta^2}{2} \int_{\Gamma_g} |g|^2 \, dx \]  

subject to:

\[ a \cdot \nabla u - \nabla \cdot \sigma(u, p) = f \quad \text{on } \Omega, \]  
\[ \nabla \cdot u = 0 \quad \text{on } \Omega, \]  
\[ \sigma(u, p) \cdot n = h \quad \text{on } \Gamma_h, \]  
\[ u = d \quad \text{on } \Gamma_d, \]  
\[ u = g \quad \text{on } \Gamma_g. \]  

The above equations represent the momentum and continuity equations subject to Neumann- and Dirichlet-type boundary conditions. In (6.2), the functions \( f, h, d \) are given and the control \( g \) has to be determined as the solution of the optimization problem. The regularization parameter \( \beta^2 > 0 \) is also given. The stress tensor \( \sigma \) is given by (2.6), and so the fluid is assumed Newtonian. We consider two objective functions \( J(u, g) \) with:

\[ J_1(u) = \frac{1}{2} \int_{\Omega_{\text{obs}}} |\nabla \times u|^2 \, dx, \]  

and

\[ J_1(u) = 2\mu \int_{\Omega_{\text{obs}}} \varepsilon(u) : \varepsilon(u) \, dx, \]  

where \( \Omega_{\text{obs}} \subset \Omega. \)

As mentioned previously, there are at least two approaches for the numerical solution of linear quadratic optimal control problems. In the \textit{discretize-then-optimize} approach, the objective function and governing PDEs are discretized. In our case, the Oseen equation is discretized using the GLS finite element formulation. This leads to a large-scale finite-dimensional quadratic programming problem, or equivalently, to a large-scale linear system that is solved using suitable numerical linear
algebra tools. In the *optimize-then-discretize* approach, the first-order necessary and sufficient optimality conditions are formed and then discretized. The first-order optimality conditions consist of the Oseen equations, the so-called adjoint PDEs, which have a structure similar to the Oseen equations, and an algebraic equation that links controls and adjoint variables. These PDEs are then individually discretized, in our case using the GLS formulation applied to the Oseen and the adjoint PDEs. This also results in a large-scale linear system. The linear systems arising in either approach are not the same, due to the way the stabilization terms affect the adjoint PDEs.

The aim of this chapter is to numerically explore the differences in the computed controls using these two approaches. We will see that an inappropriate choice of the stabilization parameter leads to significant differences in the computed controls. Moreover, the computed control can be very sensitive to a scalar weighting parameter in the stabilization term. In our results, the solutions computed by the optimize-then-discretize approach are more sensitive to the choice of the stabilization parameter and the scalar weighting term. If the stabilization is computed using an element length based on the direction of the advective field $a$, the optimal controls computed by either approach are in good agreement on fine grids.

### 6.2 A model problem

#### 6.2.1 Formulation of the optimal control problem

In the following we use the spaces $L^2(\Omega)$ and $H^1(\Omega)$, which are defined in the usual way [54], and

$$V = \{ v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_d \}, \quad L^2_0(\Omega) = \{ q \in L^2(\Omega) \mid \int_{\Omega} q(x)dx = 0 \}.$$
We set $\mathbf{H}^1(\Omega) = [H^1(\Omega)]^{n_{s,t}}$, $\mathbf{V} = V^{n_{s,t}}$, $L^2(\Gamma_h) = [L^2(\Gamma_h)]^{n_{s,t}}$, and $L^2(\Gamma_g) = [L^2(\Gamma_g)]^{n_{s,t}}$.

The Dirichlet boundary conditions (6.2d,6.2e) can be implemented through interpolation [55], weakly through a Lagrange multiplier technique [56], or via a penalty approach [52,57]. We use interpolation to implement the Dirichlet boundary conditions (6.2d) with fixed data and we replace (6.2e) by the penalized Neumann boundary condition:

$$\sigma(u, p) \cdot n + \frac{1}{\delta} u = \frac{1}{\delta} g \quad \text{on } \Gamma_g,$$

(6.5)

where $\delta > 0$ is a given penalty parameter. This choice enables us to look for controls $g$ in $L^2(\Gamma_g)$, instead of $H^{1/2}(\Gamma_g)$.

The weak form of the partial differential equation (6.2a-6.2d,6.5) is given as follows: find $u \in H^1(\Omega)$ with $u = d$ on $\Gamma_d$ and $p \in L^2(\Omega)$, such that $\forall w \in V$ and $\forall q \in L^2(\Omega)$:

$$\int_{\Omega} (a \cdot \nabla) u \cdot w \, dx + \int_{\Omega} \varepsilon(w) : \sigma(u, p) \, dx - \int_{\Omega} q \nabla \cdot u \, dx + \frac{1}{\delta} \int_{\Gamma_g} w \cdot (u - g) \, dx = \int_{\Gamma_h} w \cdot h \, dx + \int_{\Omega} f \cdot w \, dx.$$

(6.6)

It is also possible to implement (6.2d) via a penalized Neumann approach. See Remark 6.2.3 below. Equation (6.6) motivates the definition of the bilinear and
trilinear forms:

\[ a(u, w) = 2\mu \int_{\Omega} \varepsilon(w) : \varepsilon(u) \, dx \quad \forall w, u \in H^1(\Omega), \quad (6.7a) \]

\[ c(u, w) = \int_{\Omega} (a \cdot \nabla) u \cdot w \, dx \quad \forall w, u \in H^1(\Omega), \quad (6.7b) \]

\[ b(u, q) = \int_{\Omega} q \nabla \cdot u \, dx \quad \forall u \in H^1(\Omega), \forall q \in L^2(\Omega), \quad (6.7c) \]

\[ \langle h, w \rangle_{r_h} = \int_{\Gamma_h} h \cdot w \, dx \quad \forall h \in L^2(\Gamma_h), \forall w \in V, \quad (6.7d) \]

\[ \langle g, w \rangle_{r_g} = \int_{\Gamma_g} g \cdot w \, dx \quad \forall g \in L^2(\Gamma_g), \forall w \in V, \quad (6.7e) \]

and of the linear functional:

\[ \langle f, w \rangle_{\Omega} = \int_{\Omega} f \cdot w \, dx \quad \forall w \in V. \quad (6.7f) \]

For a given \( \delta > 0 \), we consider the optimal control problem:

\[
\begin{align*}
\text{minimize} & \quad J_1(u) + \frac{\beta^2}{2} \int_{\Gamma_g} |g|^2 \, dx, \\
\text{subject to} & \quad a(u, w) + c(u, w) - b(w, p) - b(u, q) \\
& \quad + \delta^{-1}\langle u, w \rangle_{r_g} - \delta^{-1}\langle g, w \rangle_{r_g}
\end{align*}
\]

\[ = \langle h, w \rangle_{r_h} + \langle f, w \rangle_{\Omega} \quad \forall w \in V, \forall q \in L^2(\Omega). \quad (6.8b) \]

**Lemma 6.2.1.** Let \( \delta > 0 \) be given and assume that there exists \( u_d \in H^1(\Omega) \) with \( u_d = d \) on \( \Gamma_d \). If \( a \in H^1(\Gamma_h) \) satisfies \( \nabla \cdot a = 0 \) and \( a \cdot n \geq 0 \) on \( \Gamma \setminus \Gamma_d \), then for any \( h \in L^2(\Gamma_h), g \in L^2(\Gamma_g) \) and \( f \in L^2(\Omega) \) the equation (6.8b) has a unique solution \( (u, p) \in H^1(\Omega) \times L^2(\Omega) \). Moreover, there exists a constant \( C > 0 \) (dependent on \( \delta \), but independent of \( h, g, f, u_d \)) such that

\[
\|u\|_{H^1(\Omega)} + \|p\|_{L^2(\Omega)} \leq C(\|h\|_{L^2(\Gamma_h)} + \|g\|_{L^2(\Gamma_g)} + \|f\|_{L^2(\Omega)} + \|u_d\|_{H^1(\Omega)}).
\]
Proof. The bilinear form \( b \) satisfies the inf-sup condition:

\[
\inf_{q \in L^2(\Omega)} \sup_{w \in V} \frac{b(w, q)}{\|q\|_{L^2(\Omega)} \|w\|_V} \geq \sigma > 0
\]

[39, p. 81]. Integration by parts yields:

\[
c(w, w) = -\frac{1}{2} \int_{\Omega} \nabla \cdot a |w|^2 dx + \frac{1}{2} \int_{\partial \Omega} a \cdot n |w|^2 dx \quad \forall w \in H^1(\Omega).
\]

(6.9)

With the assumptions on \( a \), this implies

\[
c(w, w) \geq 0 \quad \forall w \in V.
\]

Hence, there exists \( \gamma > 0 \) such that

\[
a(w, w) + c(w, w) + \delta^{-1} (w, w)_{\Gamma_g} \geq \gamma \|w\|_{H^1(\Omega)}^2 \quad \forall w \in V.
\]

(6.10)

Continuity of the linear and bilinear forms (6.7) follows using the arguments in [57].

The assertion now follows from the theory of saddle point problems [58, Sec. II.1], [39, Sec. 4].

\[\square\]

**Theorem 6.2.2.** Let the assumptions of Lemma 6.2.1 be satisfied. If \( J_1 \) is weakly lower semicontinuous on \( V \), then (6.8) admits a solution \((u_5, p_5, g_5) \in H^1(\Omega) \times L^2(\Omega) \times L^2(\Gamma_g)\). If \( J_1 \) is convex, e.g., if \( J_1 \) is given by (6.3) or (6.4), the solution is unique.

**Proof.** The result follows from standard arguments, see, e.g., the proof of Theorem 3.5 in [57].

\[\square\]

**Remark 6.2.3.** It is possible to also implement the Dirichlet boundary condition (6.2d) via a penalized Neumann condition:

\[
\sigma(u, p) \cdot n - \frac{1}{2} (a \cdot n) u + \frac{1}{\delta} u = \frac{1}{\delta} d \quad \text{on } \Gamma_d.
\]

(6.11)
The weak form of the partial differential equation (6.2a-6.2c,6.5,6.11) is given as follows: find \( u \in H^1(\Omega) \) and \( p \in L^2(\Omega) \), such that \( \forall w \in H^1(\Omega) \) and \( \forall q \in L^2(\Omega) \)

\[
\int_\Omega (a \cdot \nabla) u \cdot w \, dx + \int_\Omega \varepsilon(w) : \sigma(u, p) \, dx - \int_\Omega q \nabla \cdot u \, dx - \frac{1}{2} \int_{\Gamma_a} (a \cdot n)|u|^2 \, dx \\
+ \frac{1}{\delta} \int_{\Gamma_g} w \cdot (u - g) \, dx + \frac{1}{\delta} \int_{\Gamma_d} w \cdot (u - d) \, dx \\
= \int_{\Gamma_h} w \cdot h \, dx + \int_\Omega f \cdot w \, dx \tag{6.12}
\]

Lemma 6.2.1 with \( V \) replaced by \( H^1(\Omega) \) still holds, since (6.9) and the inclusion of the term \(-\frac{1}{2}(a \cdot n) u \) in (6.11) imply

\[
a(w, w) + c(w, w) + \delta^{-1} \langle w, w \rangle_{\Gamma_g} \geq \gamma \| w \|_{H^1(\Omega)}^2 \quad \forall w \in H^1(\Omega). \tag{6.13}
\]

The convergence behavior of the solutions \( (u_\delta, p_\delta, g_\delta) \) of the optimal control problem (6.8) with penalized Neumann boundary control to the solution of (6.1,6.2) as \( \delta \to 0 \) is discussed in [57] in the context of Navier-Stokes equations. For the purpose of this study, we view (6.8) with a small but fixed \( \delta > 0 \) as our optimal control problem.

The Lagrangian associated with the optimal control problem (6.8) is given by:

\[
L(u, p, \lambda, \theta, g) = J_1(u) + \frac{\beta^2}{2} \int_{\Gamma_g} |g|^2 \, dx \\
+ a(u, \lambda) + c(u, \lambda) - b(\lambda, p) - b(u, \theta) \\
+ \delta^{-1} \langle u, \lambda \rangle_{\Gamma_g} - \delta^{-1} \langle g, \lambda \rangle_{\Gamma_g} - \langle h, \lambda \rangle_{\Gamma_h} - \langle f, \lambda \rangle_{\Omega}. \tag{6.14}
\]

Lemma 6.2.1 ensures that (6.8) satisfies a constraint qualification. Hence, the necessary and sufficient optimality conditions for (6.8) are obtained by setting the Fréchet derivatives of the Lagrangian with respect to the adjoint variables \( (\lambda, \theta) \), with respect to the state variables \( (u, p) \) and with respect to the controls \( g \) equal to zero.
The necessary and sufficient optimality conditions consist of the state equation:

\[
\begin{align*}
  a(u, w) + c(u, w) - b(w, p) - b(u, q) + \delta^{-1}\langle u, w \rangle_{\Gamma_g} - \delta^{-1}\langle g, w \rangle_{\Gamma_g} \\
  = \langle h, w \rangle_{\Gamma_h} + \langle f, w \rangle_{\Omega} \quad \forall w \in V, \forall q \in L^2(\Omega),
\end{align*}
\] (6.15a)

the adjoint equation:

\[
\begin{align*}
  a(w, \lambda) + c(w, \lambda) - b(\lambda, q) - b(w, \theta) + \delta^{-1}\langle w, \lambda \rangle_{\Gamma_g} \\
  = -\langle \{ D_u J_1(u), w \} \rangle \quad \forall w \in V, \forall q \in L^2(\Omega),
\end{align*}
\] (6.15b)

and the gradient equation:

\[
\beta^2\langle g, z \rangle_{\Gamma_g} = \frac{1}{\delta}\langle \lambda, z \rangle_{\Gamma_g} \quad \forall z \in L^2(\Gamma_g).
\] (6.15c)

In the adjoint equation (6.15b), \(D_u J_1(u)\) denotes the Fréchet derivative of \(J_1\) with respect to \(u\) and \(\langle \cdot, \cdot \rangle\) is the duality pairing between \((H^1(\Omega))^*\), the dual of \(H^1(\Omega)\), and \(H^1(\Omega)\), i.e., \(\langle D_u J_1(u), w \rangle\) is the application of the Fréchet derivative of \(J_1\) to the function \(w\). In the case of our objective functions (6.3) and (6.4) these are simply given as:

\[
\langle D_u J_1(u), w \rangle = \int_{\Omega_{\text{obs}}} (\nabla \times u) \cdot (\nabla \times w) dx,
\] (6.16)

and

\[
\langle D_u J_2(u), w \rangle = 4\mu \int_{\Omega_{\text{obs}}} \varepsilon(u) : \varepsilon(w) dx,
\] (6.17)

respectively.

The adjoint equation (6.15b) may formally be interpreted as the weak form of
the adjoint partial differential equation:

\[-(a \cdot \nabla) \lambda - (\nabla \cdot a) \lambda - \nabla \cdot \sigma(\lambda, \theta) = -D_u J_1(u) \quad \text{on } \Omega, \quad (6.18a)\]

\[\nabla \cdot \lambda = 0 \quad \text{on } \Omega, \quad (6.18b)\]

\[\sigma(\lambda, \theta) \cdot n + (a \cdot n) \lambda = 0 \quad \text{on } \Gamma_h, \quad (6.18c)\]

\[\lambda = 0 \quad \text{on } \Gamma_d, \quad (6.18d)\]

\[\sigma(\lambda, \theta) \cdot n + (a \cdot n) \lambda + \frac{1}{\delta} \lambda = 0 \quad \text{on } \Gamma_g, \quad (6.18e)\]

and the equation (6.15c) simply states that

\[g = \frac{1}{\delta \beta^2} \lambda \quad \text{on } \Gamma_g.\]

### 6.2.2 Discretization of the state equations

For the discretization of the state equation, we apply the GLS-stabilized finite element method using piecewise-linear polynomials for both the velocity and the pressure. We divide our domain \(\Omega\) into \(n_{el}\) subdomains \(\Omega^e, e = 1, 2, \ldots, n_{el}\). We assume that our triangulation is such that the controlled boundary \(\Gamma_g\) can be written as the union of edges/faces of elements \(\Omega^e, e = 1, 2, \ldots, n_{el}\). We define the finite dimensional spaces:

\[H^1_h(\Omega) = \{ \phi^h | \phi^h \in C^0(\Omega), \phi^h|_{\Omega^e} \in P^1, e = 1, 2, \ldots, n_{el} \},\]

\[V^h = \{ \phi^h \in H^1_h(\Omega) | \phi^h = 0 \text{ on } \Gamma_d \}, L^{2h}(\Omega) = H^{1h}(\Omega), H^{1h}(\Omega) = [H^{1h}(\Omega)]^{n_{sd}},\]

\[V^h = [V^h]^{n_{sd}}, L^{2h}(\Gamma_g) = \{ \phi^h|_{\Gamma_g} | \phi^h \in H^{1h}(\Omega) \}, \text{ and } L^{2h}(\Gamma_g) = [L^{2h}(\Gamma_g)]^{n_{sd}}.\]

We let \(I_h d\) denote the piecewise linear interpolant of the Dirichlet boundary data \(d\).

The GLS stabilized finite element formulation for the state equation (6.2a-6.2c), (6.5) is given as follows. find \(u^h \in H^{1h}(\Omega)\) with \(u^h = I_h d\) on \(\Gamma_d\) and \(p^h \in L^{2h}(\Omega)\)
such that $\forall w^h \in V^h$ and $\forall q^h \in L^2h(\Omega)$:

\[
\begin{align*}
\int_{\Omega} w^h \cdot (a \cdot \nabla) u^h \, dx + \int_{\Omega} \varepsilon(w^h) : \sigma(u^h, p^h) \, dx \\
+ \sum_{e=1}^{n_\Delta} \int_{\Omega^e} \tau_e \left[ (a \cdot \nabla w^h) - \nabla \cdot \sigma(w^h, q^h) \right] \cdot \left[ ((a \cdot \nabla u^h) - \nabla \cdot \sigma(u^h, p^h)) \right] \, dx \\
+ \int_{\Omega} q^h \nabla \cdot u^h \, dx + \frac{1}{\delta} \int_{\Gamma^h} w^h \cdot (u^h - g^h) \, dx \\
= \int_{\Gamma^h} w^h \cdot h \, dx + \int_{\Omega} f \cdot w^h \, dx \\
+ \sum_{e=1}^{n_\Delta} \int_{\Omega^e} \tau_e \left[ (a \cdot \nabla w^h) - \nabla \cdot \sigma(w^h, q^h) \right] \cdot f \, dx.
\end{align*}
\]

(6.19)

6.2.3 Discretization of the optimization problem

In the discretize-then-optimize (do) approach for the numerical solution of an optimal control, one discretizes the optimal control problem first and then solves the resulting finite-dimensional nonlinear (in our case quadratic) programming problem using a suitable optimization algorithm. Using the GLS-stabilized finite element method described in Section 6.2.2 for the discretization of the state equation, our discretization of the optimal control problem (6.8) is given by:

\[
\begin{align*}
\text{minimize} & \quad J_I(u^h) + \frac{\beta^2}{2} \int_{\Gamma^h} |g^h|^2 \, dx, \\
\text{subject to} & \quad (6.19),
\end{align*}
\]

with $u^h \in H^1h(\Omega), p^h \in L^2h(\Omega), g^h \in L^2h(\Gamma^h), u^h = I_h d$ on $\Gamma_d$. 
The Lagrangian for the discretized problem (6.20) is given by:

\[
L^h(u^h, p^h, \lambda^h, \theta^h, g^h) = J_1(u^h) + \frac{\beta^2}{2} \int_{\Gamma_g} |g^h|^2 dx \\
+ \int_{\Omega} \lambda^h \cdot (a \cdot \nabla) u^h \, dx + \int_{\Omega} \varepsilon(\lambda^h) : \sigma(u^h, p^h) \, dx \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \left[ (a \cdot \nabla) \lambda^h - \nabla \cdot \sigma(\lambda^h, \theta^h) \right] \cdot \left[ (a \cdot \nabla) u^h - \nabla \cdot \sigma(u^h, p^h) \right] \, dx \\
+ \int_{\Omega} \theta^h \nabla \cdot u^h \, dx + \frac{1}{\delta} \int_{\Gamma_g} \lambda^h \cdot (u^h - g^h) \, dx \\
- \int_{\Gamma_h} \lambda^h \cdot h \, dx - \int_{\Omega} f \cdot \lambda^h \, dx \\
- \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \left[ (a \cdot \nabla) \lambda^h - \nabla \cdot \sigma(\lambda^h, \theta^h) \right] \cdot f \, dx. \tag{6.21}
\]

The necessary and sufficient optimality conditions for (6.20) are obtained by setting the derivatives of the Lagrangian with respect to the discrete adjoint variables \((\lambda^h, \theta^h)\), the discretized state variables \((u^h, p^h)\) and the discretized controls \(g^h\) to zero. Setting the derivative of \(L^h\) with respect to the discrete adjoint variables \((\lambda^h, \theta^h)\) to zero gives the discretized state equations (6.19). Setting the derivative of \(L^h\) with respect to the discretized state variables \((u^h, p^h)\) to zero gives the discrete adjoint equations:

\[
\int_{\Omega} \lambda^h \cdot (a \cdot \nabla) w^h \, dx + \int_{\Omega} \varepsilon(\lambda^h) : \sigma(w^h, q^h) \, dx \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \left[ (a \cdot \nabla) \lambda^h - \nabla \cdot \sigma(\lambda^h, \theta^h) \right] \cdot \left[ (a \cdot \nabla) w^h - \nabla \cdot \sigma(w^h, q^h) \right] \, dx \\
+ \int_{\Omega} \theta^h \nabla \cdot w^h \, dx + \frac{1}{\delta} \int_{\Gamma_g} \lambda^h \cdot w^h \, dx \\
= -\langle \langle D_u J_1(u^h), w^h \rangle \rangle \quad \forall w^h \in V^h, \forall q \in L^2(h)(\Omega), \tag{6.22}
\]

where \(\langle \langle D_u J_1(u^h), w^h \rangle \rangle\) is defined as in (6.16) or (6.17), respectively. Finally, setting the derivative of \(L^h\) with respect to the discretized control variables \(g^h\) to zero gives
the discrete gradient equation:

$$\beta^2 \int_{\Gamma_g} g^h \cdot z^h \, dx = \frac{1}{\delta} \int_{\Gamma_g} \lambda^h \cdot z^h \, dx \quad \forall z^h \in L^2_h(\Gamma_g). \tag{6.23}$$

The necessary and sufficient conditions (6.22), (6.23) and (6.19) lead to a system of linear equations:

$$
\begin{pmatrix}
H_{do} & 0 & A^T \\
0 & G^T & -B^T \\
A & -B & 0
\end{pmatrix}
\begin{pmatrix}
\bar{u}_{do} \\
\bar{g}_{do} \\
\bar{\lambda}_{do}
\end{pmatrix}
= 
\begin{pmatrix}
\bar{\delta} \\
\bar{\delta} \\
\bar{f}
\end{pmatrix}. \tag{6.24}
$$

Here $\bar{u}_{do}$ and $\bar{\lambda}_{do}$ are the vectors containing the values of $(u^h, p^h)$ and $(\lambda^h, \theta^h)$ at the grid points, respectively. Vectors $\bar{g}_{do}$ and $\bar{f}$ represent the same extensions of $g_{do}$ and $f$ to velocity-pressure space.

### 6.2.4 Discretization of the optimality conditions

The optimality conditions (6.15) are necessary and sufficient. Therefore, we can tackle the optimality conditions directly to compute an approximate solution of our optimal control problem. This approach, called the optimize-then-discretize (otd) approach, requires us to discretize the state and the adjoint PDE (6.18). In principle, these discretizations do not have to be the same. However, we will use the same mesh and the GLS-stabilized finite element method for the discretization of both. We use the same notation as in section 6.2.2.

Our discretization of the state equation is given in (6.19). GLS stabilized finite
element method applied to the adjoint equation (6.15b), (6.18), yields:

\[
\begin{align*}
\int_{\Omega} \lambda^h \cdot (a \cdot \nabla) w^h \, dx + \int_{\Omega} \varepsilon(w^h) : \sigma(\lambda^h, \theta^h) \, dx \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \left[ -(a \cdot \nabla) w^h - (\nabla \cdot a) w^h - \nabla \cdot \sigma(w^h, q^h) \right] \\
\cdot \left[ -(a \cdot \nabla) \lambda^h - (\nabla \cdot a) \lambda^h - \nabla \cdot \sigma(\lambda^h, \theta^h) \right] \, dx \\
+ \int_{\Omega} q^h \nabla \cdot \lambda^h \, dx + \frac{1}{\delta} \int_{\Gamma_g} \lambda^h \cdot w^h \, dx \\
= -(\langle D_u J_1(u^h), w^h \rangle)_s \quad \forall w^h \in V^h, \forall q \in L^2(\Omega),
\end{align*}
\]

(6.25)

where \(\langle D_u J_1(u^h), w^h \rangle)_s\) is given as follows. For the objective function (6.3),

\[
\begin{align*}
\langle D_u J_1(u^h), w^h \rangle)_s &= \int_{\Omega_{obs}} (\nabla \times u^h) \cdot (\nabla \times w^h) \, dx \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e \cap \Omega_{obs}} \tau_e \cdot (\nabla \times u^h) \cdot \\
(\nabla \times \left[ -(a \cdot \nabla) w^h - (\nabla \cdot a) w^h - \nabla \cdot \sigma(w^h, q^h) \right]) \, dx
\end{align*}
\]

(6.26)

and for the objective function (6.4),

\[
\begin{align*}
\langle D_u J_1(u^h), w^h \rangle)_s &= 4\mu \int_{\Omega_{obs}} \varepsilon(u^h) : \varepsilon(w^h) \, dx \\
+ 4\mu \sum_{e=1}^{n_{el}} \int_{\Omega^e \cap \Omega_{obs}} \tau_e \varepsilon(u^h) : \\
\varepsilon \left( -(a \cdot \nabla) w^h - (\nabla \cdot a) w^h - \nabla \cdot \sigma(w^h, q^h) \right) \, dx.
\end{align*}
\]

(6.27)

The discretization of (6.15c) is again given by (6.23). The necessary and sufficient conditions (6.19), (6.25) and (6.23) lead to a system of linear equations:

\[
\begin{pmatrix}
H_{od} & 0 & \tilde{A} \\
0 & G^T & -B^T \\
A & -B & 0
\end{pmatrix}
\begin{pmatrix}
\tilde{u}_{od} \\
\tilde{g}_{od} \\
\tilde{\lambda}_{od}
\end{pmatrix}
= \begin{pmatrix}
\tilde{0} \\
\tilde{f}
\end{pmatrix}.
\]

(6.28)

Again \(\tilde{u}_{od}\) and \(\tilde{\lambda}_{od}\) are the vectors containing the values of \((u^h, p^h)\) and \((\lambda^h, \theta^h)\) at the grid points, respectively.
Note that because of the stabilization terms in (6.26) and in (6.27), the submatrix \( H_{do} \) in (6.24) is different from the submatrix \( H_{od} \) in (6.28). Moreover, because the advective terms in adjoint equation (6.18) are different from those in the Oseen equation (6.2), the discrete adjoint equations (6.22) are different from the discretized adjoint equations (6.25). Hence the submatrix \( A^T \) in (6.24) is also different from the submatrix \( \tilde{A} \) in (6.28).

### 6.3 Implementation

#### 6.3.1 Stabilization terms

Our choice of piecewise-linear functions makes our discrete approximation a low-order method, i.e. the order of the function space is lower than the order of the highest derivative in (6.2), the stabilized finite element equivalent of which is (6.19). Therefore, we drop the term given by \( \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_e \left[ (\mathbf{a} \cdot \nabla u^h) - \nabla \cdot \sigma(u^h, p^h) - f \right] \cdot \left[ -\nabla \cdot 2\mu \varepsilon(w^h) \right] dx \), and \( \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_e \left[ (\mathbf{a} \cdot \nabla w^h) + \nabla q^h \right] \cdot \left[ -\nabla \cdot 2\mu \varepsilon(u^h) \right] dx \) in (6.19). Note that in this chapter, we have not used the methods to achieve stronger consistency discussed in Chapter 5, in order to avoid the associated nonlinearities. A similar inconsistency is present in (6.22) and (6.25). The implemented version of (6.22) is given by:

\[
\int_{\Omega} \lambda^h \cdot (\mathbf{a} \cdot \nabla)w^h \, dx + \int_{\Omega} \varepsilon(\lambda^h) : \sigma(w^h, q^h) \, dx \\
+ \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_e \left[ (\mathbf{a} \cdot \nabla)w^h + \nabla q^h \right] \cdot \left[ (\mathbf{a} \cdot \nabla)\lambda^h + \nabla \theta^h \right] \, dx \\
+ \int_{\Omega} \theta^h \nabla \cdot w^h \, dx + \frac{1}{\delta} \int_{\Gamma_b} \lambda^h \cdot w^h \, dx \\
= -\langle \langle D_{u^h} J_1(u^h), w^h \rangle \rangle \quad \forall \mathbf{w}^h \in \mathbf{V}^h, \forall q \in L^2(\Omega), \quad (6.29)
\]
and the implemented version of (6.25) is given by:

\[
\begin{align*}
\int_{\Omega} & \lambda^h \cdot (a \cdot \nabla) w^h \, dx + \int_{\Omega} \varepsilon(w^h) : \sigma(\lambda^h, \theta^h) \, dx \\
& + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \left[ -(a \cdot \nabla) w^h - (\nabla \cdot a) w^h + \nabla q^h \right] \\
& \cdot \left[ -(a \cdot \nabla) \lambda^h - (\nabla \cdot a) \lambda^h + \nabla \theta^h \right] \, dx \\
& + \int_{\Omega} q^h \nabla \cdot \lambda^h \, dx + \frac{1}{\delta} \int_{\Gamma^h} \lambda^h \cdot w^h \, dx \\
& = -\langle \langle Du_j(u^h), w^h \rangle \rangle_s \quad \forall w^h \in V^h, \forall q \in L^2(\Omega), \\
& \text{(6.30)}
\end{align*}
\]

where

\[
\langle \langle Du_j(u^h), w^h \rangle \rangle_s = \int_{\Omega_{obs}} (\nabla \times u^h) \cdot (\nabla \times w^h) \, dx \\
& + \sum_{e=1}^{n_{el}} \int_{\Omega^e \cap \Omega_{obs}} \tau_e \left( \nabla \times u^h \right) \cdot \left( \nabla \times \left[ -(a \cdot \nabla) w^h - (\nabla \cdot a) w^h \right] \right) \, dx, \quad \text{(6.31)}
\]

and

\[
\langle \langle Du_j(u^h), w^h \rangle \rangle_s = 4\mu \int_{\Omega_{obs}} \varepsilon(u^h) : \varepsilon(w^h) \, dx \\
& + 4\mu \sum_{e=1}^{n_{el}} \int_{\Omega^e \cap \Omega_{obs}} \tau_e \varepsilon(u^h) : \varepsilon \left( -(a \cdot \nabla) w^h - (\nabla \cdot a) w^h \right) \, dx, \quad \text{(6.32)}
\]

for the objective functions (6.3) and (6.4), respectively.

### 6.3.2 Differences

The use of GLS finite element method in the discretization of the optimal control problem creates differences between the discretize-then-optimize approach and the optimize-then-discretize approach. Specifically, there are differences between the discrete adjoint equation (6.29) and the discretized adjoint equation (6.30). To explore how these differences impact the computed solution, we also implement variations of (6.30).
1. The left hand side and right hand side in (6.30) contain \((\nabla \cdot \mathbf{a})\)-terms, while the divergence of the advective field does not enter into the discrete adjoint equation (6.29). Therefore, we also compute the optimal controls using (6.19), (6.23) and (6.30) where the \((\nabla \cdot \mathbf{a})\)-terms in (6.30) are dropped. The optimal controls computed this way will be labeled as \text{od1}.

2. If we compare (6.29) and (6.30) with \((\nabla \cdot \mathbf{a})\)-terms in (6.30) replaced by zero, we see that the right hand sides of the discretized adjoint equations (6.30) contain \(- (\mathbf{a} \cdot \nabla) w^h\) terms that are not present in the discrete adjoint equations (6.29). Therefore, we also compute the optimal controls using (6.19), (6.23) and (6.30) where all \((\nabla \cdot \mathbf{a})\)-terms in (6.30) as well as the \(- (\mathbf{a} \cdot \nabla) w^h\) term in the right hand side of (6.30) are dropped. The optimal controls computed this way will be labeled as \text{od2}.

3. After the modifications of the discretized adjoint equations (6.30) described above in 2. have been performed, the only remaining difference between (6.29) and (6.30) are the signs of \(\nabla q^h\) and \(\nabla \theta^h\) in (6.30).

### 6.4 Post-optimality sensitivity analysis for choice of stabilization parameter

We are interested in the sensitivity of the computed solution with respect to the stabilization factor \(\alpha\). Equations (6.19), (6.22) and (6.23) reveal that the submatrix \(\mathbf{A}\) and the right hand side component \(\mathbf{f}\) in (6.28) depend on \(\alpha\). Hence we write \(\mathbf{A}(\alpha)\) and \(\mathbf{f}(\alpha)\) instead of \(\mathbf{A}\) and \(\mathbf{f}\), respectively. Moreover, we denote the solution of (6.24) by \(\mathbf{u}_{d\alpha}(\alpha)\), \(\mathbf{g}_{d\alpha}(\alpha)\), \(\mathbf{X}_{d\alpha}(\alpha)\). Examination of (6.19), (6.22), (6.23) reveals that the submatrices \(\mathbf{A}(\alpha)\) and \(\mathbf{H}_{d\alpha}(\alpha)\) in (6.28) are of the form \(\mathbf{A}(\alpha) = \mathbf{A}_1 + \mathbf{A}_2 \alpha\) and
$H_{do}(\alpha) = H_1$, where we use the subscript 1 for all those terms that do not depend on stabilization, and the subscript 2 for those terms that are dependent on stabilization. Hence, if we use $'$ to denote differentiation with respect to $\alpha$, then $A'(\alpha) = A_2$ and $H'_{do}(\alpha) = 0$. By the implicit function theorem, the derivatives of the solution of (6.24) satisfy:

$$
\begin{pmatrix}
    H_{do} & 0 & A(\alpha)^T \\
     0 & G^T & -B^T \\
    A(\alpha) & -B & 0
\end{pmatrix}
\begin{pmatrix}
    \bar{u}_{do}'(\alpha) \\
    \bar{g}_{do}'(\alpha) \\
    \bar{\lambda}_{do}'(\alpha)
\end{pmatrix}
= -
\begin{pmatrix}
    0 & 0 & (A'(\alpha))^T \\
    0 & 0 & 0 \\
    A'(\alpha) & 0 & 0
\end{pmatrix}
\begin{pmatrix}
    \bar{u}_{do}(\alpha) \\
    \bar{g}_{do}(\alpha) \\
    \bar{\lambda}_{do}(\alpha)
\end{pmatrix}
+ 
\begin{pmatrix}
    \bar{0} \\
    \bar{0} \\
    \bar{f}'(\alpha)
\end{pmatrix},
$$

where $\bar{u}_{do}(\alpha), \bar{g}_{do}(\alpha), \bar{\lambda}_{do}(\alpha)$ solve (6.24).

The sensitivity of the solution $\bar{u}_{od}(\alpha), \bar{g}_{od}(\alpha), \bar{\lambda}_{od}(\alpha)$ of (6.28) can be computed analogously. Note that in this case $H_{od}$ also depends on $\alpha$ and is of the form $H_{od}(\alpha) = H_1 + H_2\alpha$, with $H'_{od}(\alpha) = H_2$.

### 6.5 Numerical results

In this section we study the effect of the GLS stabilization on the computed control for two test cases derived from commonly-used model problems. In both cases, the advective field $a$ is computed by solving the Navier-Stokes equations

$$
\rho (a \cdot \nabla) a - \nabla \cdot \sigma(a, p) = 0 \quad \text{on} \quad \Omega,
$$

$$
\nabla \cdot a = 0 \quad \text{on} \quad \Omega, \quad \text{(6.33)}
$$

with appropriate boundary conditions. The Navier-Stokes equations (6.33) are discretized using the GLS-stabilized finite element method and the resulting nonlinear
system is solved using Newton's method with a residual tolerance of $10^{-16}$. Note that while the "exact" advective field $a$ is divergence-free, this is not true for the computed advective field, which is used as the coefficient in our computations.

We solve (6.8) with parameters $\delta = 10^{-5}$ and $\beta^2 = 10^{-5}$. As stated before, we use the notation do to refer to the control computed using the \textit{discretize-then-optimize} approach (6.23), (6.19) and (6.29), and od to refer to the control computed by the \textit{optimize-then-discretize} approach (6.23), (6.19) and (6.30), respectively.

### 6.5.1 Test case 1

The first problem is modeled after the backward-facing step problem [52, pp. 1767,1769]. A schematic of the geometry is given in Figure 6.1. The height of the inflow boundary is 0.5 and that of the outflow boundary is 1. The length of the narrower section of the channel is 1 and that of the wider section is 7, with the total horizontal length being 8. The advective velocity field $a$ is computed by solving the Navier-Stokes equations (6.33) with the following boundary conditions. The inflow velocity is assumed parabolic with a profile $a(0,y) = (8(y - 0.5)(1 - y), 0)^T$. At the outflow boundary $\Gamma_o$ we impose traction-free boundary conditions for $a$ in the $x$-direction. No-slip conditions are imposed at all other boundaries. We define the Reynolds number $Re = \rho U_{\text{max}} H/\mu$, where $U_{\text{max}}$ is the maximum inlet velocity and $H$ is the channel height. The Reynolds number for computing $a$ is $Re = 200$.

The boundary conditions for the Oseen equation are as follows. The inflow velocity field, different from that of $a$, is taken as parabolic given by $u(0,y) = ((y - 0.5)(1 - y), 0)^T$. A traction-free boundary condition for the $x$-velocity is imposed at $\Gamma_o$. The controlled boundary $\Gamma_g$ is the backward facing step. We allow suction and blowing in the normal direction. At the remaining boundaries $\Gamma_b$ and $\Gamma_t$,
no-slip conditions are imposed for $u$. The objective function is given by (6.3), where $\Omega_{\text{obs}}$ is the subdomain of length 2 and height 0.5, as indicated in Figure 6.1.

We present results for two kinematic viscosity coefficients $\mu = 0.0005$ and $\mu = 0.00005$, and for three discretizations. Discretization 1 consists of 1800 triangular elements as shown in Figure 6.2. Discretization 2 consisting of 7200 triangular elements and is a uniform refinement of discretization 1. Discretization 3 is an uniform refinement of discretization 2 and consists of 28800 elements.

Figure 6.2 : Finite element mesh for test case 1, discretization 1.

Figure 6.3 shows the control velocities obtained using the various approaches considered here. We observe that the $od$ and $do$ approaches with the $adv$ element length definition converge with mesh refinement. When we use the $diag$ element length, we observe that the $do$ approach converges to the solution obtained using the $adv$ element length. However, we observe that there is a strong oscillatory behavior in the controls for $od$ $diag$ case, especially for the smallest viscosity, i.e. $\mu = 0.00005$. 
<table>
<thead>
<tr>
<th>Parameter</th>
<th>adv</th>
<th>diag</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h_e )</td>
<td>0.01256</td>
<td>0.02795</td>
</tr>
<tr>
<td>( \tau_e )</td>
<td>0.1434</td>
<td>0.3234</td>
</tr>
</tbody>
</table>

Table 6.1: Values of \( h_e \) and \( \tau_e \) for the element at which maximum \( \nabla \cdot \mathbf{a} \) at element center is attained in test case 1.

Figure 6.4 shows the dependence of control velocities on the factor \( \alpha \), with increasing oscillatory behavior as \( \alpha \) increases. Figure 6.5 shows the sensitivity of the controls to \( \alpha \). The sensitivity of the computed control with respect to \( \alpha \) is large when the \texttt{diag} element length definition is chosen. In particular, for the \texttt{od} approach with \texttt{diag} element length definition large sensitivities are observed near the corner of the backward facing step. This correlates with the 'spikes' in the computed control for this case shown in Figure 6.3. The size of the sensitivities are relatively small for both, the \texttt{do} and the \texttt{od} approach when the \texttt{adv} element length definition is chosen. Again this correlates with the observed grid convergence for both approaches in case of the \texttt{adv} element length definition (see Figure 6.3).

From Figure 6.6, we see that the principal difference between the controls resulting from the \texttt{do} and the \texttt{od} approaches with \texttt{diag} element length definition arises from the terms containing \( \nabla \cdot \mathbf{a} \) in the latter. Such terms manifest themselves due to stabilization and therefore the difference in the results varies with the choice of element length. The maximum and minimum values of \( \nabla \cdot \mathbf{a} \), sampled at the center of each element in \( \Omega \), is 0.8368 and -0.6348. From Figure 6.4, one can also note that for the specific choice of the \texttt{do} approach with the \texttt{diag} element length definition, the controls do not exhibit significant oscillations. From Table 6.5.1, one can see that
with the choice of diag element length, the element length $h_e$ and the stabilization parameter $\tau_e$ are both larger in comparison with the adv element length choice.
Figure 6.3: The control velocities obtained for the two od and do approaches using the diag and adv element length definition with $\alpha = 1$ for test case 1. The ordering of these plots is $\mu = 0.0005$ on the left and $\mu = 0.00005$ on the right with discretizations 1 through 3 from top to bottom.
Figure 6.4: The control velocities obtained using different choices of $\alpha$ using $\text{diag}$ element length definition for $\mu = 0.00005$ in discretization 3 for test case 1. The plot on the left uses $\text{od}$ approach and the one on right uses $\text{do}$ approach. For comparison we also present $\text{adv}$ element length definition with $\alpha = 1$. 
Figure 6.5: The weighted sensitivity $|g'(\alpha)_i|/|g(\alpha)_i|$ of the controls for the do and od approaches using the diag and adv element length definitions for $\mu = 0.00005$ in discretization 3 for test case 1, with $\alpha = 0.5$, $\alpha = 1.0$ and $\alpha = 2.0$. 
Figure 6.6: The control velocities obtained using \texttt{diag} element length definition for $\mu = 0.00005$ in discretization 3 for test case 1, obtained with \texttt{od} and \texttt{do} approaches and their variations.
6.5.2 Test case 2

This test case involves flow past a circular cylinder [55, pp. 19]. A schematic of the geometry is given in Figure 6.7. We use the length $L = 10.0$ and the radius $r = L/20$. The advective velocity field $a$ is obtained by solving (6.33) with Reynolds number $Re = \rho |U| D / \mu = 100$, where $D$ is the diameter of the cylinder. The boundary conditions for $a$ are identical to those indicated in Figure 6.7 with $a = (U, V)^T$ and $T_x$ and $T_y$ being the $x$ and $y$ component of $\sigma(a, p) \cdot n$, respectively.

For the optimal control problem, we use the objective function (6.4), where $\Omega_{\text{obs}}$ is defined as the entire computational domain, excluding the first layer of elements at the inflow. Suction and blowing control is applied in normal direction at the back of the half-cylinder (top right quadrant of the cylinder). The top node in the discretization of the control is forced to have zero control velocity. The inflow velocity is $u(y) = (0.1, 0)^T$. All other boundary conditions for $u$ are indicated in Figure 6.7.

![Figure 6.7: Computational domain and boundary conditions for test case 2. In this case, $L = 10$ and $r = L/20$.](image)

We present results for two kinematic viscosity coefficients $\mu = 0.0005$ and $\mu =$
Table 6.2: Values of $h_e$ and $\tau_e$ for the element at which maximum $\nabla \cdot \mathbf{a}$ at element center is attained in test case 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\text{adv}$</th>
<th>$\text{diag}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_e$</td>
<td>0.00694</td>
<td>0.0496</td>
</tr>
<tr>
<td>$\tau_e$</td>
<td>0.0349</td>
<td>0.2522</td>
</tr>
</tbody>
</table>

0.00005, and for three discretizations. The coarse discretization consists of 2720 triangular elements, and is shown in Figure 6.8—we refer to it as discretization 1. There are 20 control degrees of freedom for this discretization. Discretization 2 consists of 10880 triangular elements is a refinement of discretization 1 and has 40 control degrees of freedom. Discretization 3 consists of 24480 triangular elements has 60 control degrees of freedom.

Similar to the first test case, we observe from Figure 6.9 that there is a strong oscillatory behavior in the controls for the od diag case, especially for the smallest viscosity, i.e., $\mu = 0.00005$. Figure 6.10 shows the dependence of control velocities on
the factor $\alpha$, with increasing oscillatory behavior as $\alpha$ increases. From Figure 6.11, we observe that the sensitivity of the computed control with respect to $\alpha$ is large when the $\text{diag}$ element length definition is chosen. In particular, for the $\text{od}$ approach with $\text{diag}$ element length definition large sensitivities are observed in the region that correlates with the spikes in the computed control shown in Figure 6.9. Again, similar to test case 1, we see from Figure 6.12 that the principal difference between the controls resulting from the $\text{do}$ and the $\text{od}$ approaches with $\text{diag}$ element length definition arises from the terms containing $\nabla \cdot a$ in the latter. The maximum and minimum values of $\nabla \cdot a$, sampled at the center of each element in $\Omega$, are 1.2745 and $-1.126$. As noted in the first test case, one can see from Table 6.2 that with the choice of $\text{diag}$ element length, the element length $h_\varepsilon$ and the stabilization parameter $\tau_\varepsilon$ are both larger in comparison with the $\text{adv}$ element length choice.
Figure 6.9: The control velocities obtained for the two od and do approaches using the diag and adv element length strategies for test case 2. The ordering of these plots is \( \mu = 0.0005 \) on the left and \( \mu = 0.00005 \) on the right with discretizations 1 through 3 from top to bottom.
Figure 6.10: The control velocities obtained using different choices of $\alpha$ using diag element length definition for $\mu = 0.00005$ in discretization 3 for test case 2. The plot on the left uses od approach and the one on right uses do approach. For comparison we also present adv element length definition with $\alpha = 1$. 
Figure 6.11: The weighted sensitivity $|(g'(\alpha))_i|/|g(\alpha)_i|$ of the controls for the do and od approaches using the diag and adv element length definitions for $\mu = 0.00005$ in discretization 3 for test case 2, with $\alpha = 0.5$, $\alpha = 1.0$ and $\alpha = 2.0$. 
Figure 6.12: The control velocities obtained using diag element length definition for $\mu = 0.00005$ in discretization 3 for test case 2, on experimenting with the terms of difference between the od and do approaches.
6.6 Conclusions

The order in which a stabilized finite element discretization is applied to an optimal control problem can have a significant effect on the computed solution. Depending on the choice of the element length, the computed control may be very sensitive to the choice of the weighting parameter used in the stabilization term and it may exhibit large spurious oscillations. It is important to understand and diagnose these effects. Otherwise, it is questionable whether the computed control resembles the true control.

Our computations indicate that a choice (3.13) of the element length based on the direction of the advective field leads to computed controls that appear to converge as the grid is refined and that, for sufficiently fine grids, are almost independent of the order in which the stabilized finite element discretization is applied to the optimal control. In addition, the sensitivity of the computed control with respect to the weighting parameter used in the stabilization term is small.

Our numerical results suggest two indicators for the quality of the computed solution. First, the difference between the controls computed by the discretize-then-optimize approach and the optimize-then-discretize approach should be small. Secondly, the sensitivity of the computed control with respect to the stabilization parameter $\alpha$ should be small. If the discretize-then-optimize approach is chosen as the principal solution approach, the system corresponding to the optimize-then-discretize approach and the sensitivity equations can be obtained by minor modifications.
Chapter 7

Shape Optimization for Steady Blood Flow and Non-Newtonian Effects

In this chapter, we investigate the influence of the fluid constitutive model on the outcome of shape optimization tasks, motivated by optimal design problems in biomedical engineering. Our computations are based on the Navier-Stokes equations generalized to non-Newtonian fluid, with the Carreau-Yasuda model employed to account for the shear-thinning behavior of, e.g., blood. The generalized Newtonian treatment exhibits striking differences in the velocity field for smaller shear rates. Following a motivating remark in Section 7.1, a model shape optimization problem is introduced in Section 7.2. For the numerical solution of the shape optimization problem we discretize the Navier-Stokes equation using GLS-stabilized finite element formulation and piecewise-linear interpolation functions. The discretized shape optimization problem is solved using a gradient-based optimization algorithm. The details of the discretization of the problem, computation of derivatives, as well as specification of the optimization algorithm used is given in Section 7.3. We apply gradient-based optimization procedure to a benchmark problem of flow through a right-angle cannula, and to a flow through an idealized arterial graft. Section 7.4 reports on the numerical results for these two test cases. For each of these problems, we study the influence of the inflow velocity, and thus the shear rate on the outcome of the shape optimization task. Furthermore, for the arterial graft problem we introduce an additional factor in the form of a geometric parameter, and study its effect on the
optimal shape obtained. We present our conclusions for this chapter in Section 7.5.

7.1 Introduction

The use of CFD in the study of physiological flows involving blood is an area of intensive research. Part of the challenge in this field is the accurate treatment of the haemodynamic behavior—see e.g. [59] for a recent review in this field. A variety of models have been proposed that capture the viscoelastic and shear-thinning behavior of blood. The papers [60–63] provide a small sample of the research on non-Newtonian effects on blood flow. A less-studied area is the usage of numerical optimization procedures to guide the design process that involves blood flow.

Given a PDE model of the flow, shape optimization aims to extremize a given objective—subject to physical or geometric constraints—through the variation of the domain or a part of it. Such problems arise in a multitude of engineering applications. Shape optimization procedures have been extensively studied in the context of aerodynamic flows [37, 46], and there has been recent interest to extend these successes, to biomedical applications [64–67]. In the latter references, a Newtonian model is used to represent the blood flow.

This chapter presents a numerical study of non-Newtonian effects on the solution of shape optimization problems for selected idealized biomedical systems. Our work is motivated by the desire to find optimal shapes in the context of continuous-flow centrifugal blood pumps [26], which are being intensively studied as a bridge to transplant, or with enough technological maturity, as a long-term autonomous artificial heart. As a step in this direction, we study stationary flows over 2D idealized geometries. To account for the shear-thinning behavior of blood, we make use of generalized Newtonian constitutive equations of the Carreau-Yasuda model. A
Newtonian assumption is generally considered valid for flows where shear rates are high, thus diminishing the shear-related viscosity differences. Gijsen et al. [60] show higher differences in the axial velocity profile compared to results obtained by [68] for a 3D carotid bifurcation study when using lower mean axial velocity and a larger diameter of the carotid artery, which results in a lower shear rate.

At first, we consider the shape optimization of a two-dimensional inflow cannula of a circulatory assist device. This test case has been studied using Newtonian constitutive equations in [65], where a significant drop in the shear stress level was reported in comparison with the original design. By individually studying geometries differing in the number of side holes in a 3D model of a cannula using Newtonian constitutive equations, Grigioni et al. [69] present significantly lower shear stress and lesser disturbance in the blood flow for one of the geometrical choices. Their work involved topological evaluation, but similar ideas can be extended to the numerical parametric shape optimization, with the same goal of improving the success rates of cannulae in clinical practice. Our first test case can be considered a step in that direction.

Secondly, we examine an idealized bypass graft, commonly used as an alternative route around critically stenosed arteries. Specifically, we study a model with a complete stenosis, one that precludes any flow between the proximal and distal end of the host artery. Guo et al. [70] study the effect of graft placement vis-a-vis the occlusion as an important criterion for improving the outcome of coronary artery bypass grafting. Graft angle has also been identified as a factor that can ensure favorable flow properties [71], which then preclude the development of further occlusions. Quarteroni and Rozza [67] recently studied optimal design in the context of prosthetic grafting using Newtonian constitutive equations. The design is computed using a
lower fidelity Stokes equation. The computed shape is then tested for suitable design control quantities defined over the more expensive unsteady Navier-Stokes flow solution.

### 7.2 A model shape optimization problem

We consider the steady form of the governing equations (2.1), a weak form of which is constructed by introducing the following function spaces:

\[
\mathcal{S}_u = \{ u \mid u \in [H^1(\Omega)]^{n_{ad}}, u = g \text{ on } \Gamma_g \},
\]

\[
\mathcal{V}_u = \{ u \mid u \in [H^1(\Omega)]^{n_{ad}}, u = 0 \text{ on } \Gamma_g \},
\]

where \( H^1(\Omega) \) is defined in the usual way [39, 40]. The weak form of the steady form of (2.1) is then: find \( u \in \mathcal{S}_u \) and \( p \in H^1(\Omega) \), such that \( \forall w \in \mathcal{V} \) and \( \forall q \in H^1(\Omega) \)

\[
\int_{\Omega} w \cdot \rho (u \cdot \nabla u - f) \, dx + \int_{\Omega} \varepsilon(w) : \sigma(u, p) \, dx + \int_{\Omega} q \nabla \cdot u \, dx = \int_{\Gamma_n} w \cdot h \, dx. \tag{7.1}
\]

Our goal is to find a shape \( \Omega \) such that a given objective function \( J \), which depends on \( u, p \), and on \( \Omega \), is minimized. We consider the case where the set of admissible shapes can be parameterized by \( \alpha \in \mathcal{A}_{ad} \subset \mathbb{R}^n \). The optimal shape design problem is given as follows:

\[
\text{minimize} \quad J(u, p, \alpha)
\]

subject to \( (7.1) \) with \( \Omega = \Omega(\alpha) \), \( \alpha \in \mathcal{A}_{ad} \).

A critical criterion for design decisions involving blood, such as for those involving prosthetic devices or artificial heart components, is to minimize the mechanical loading on blood particles, which is related to the shear stress in the flow field.
Processes such as the damage of red blood cells (hemolysis), platelet aggregation, and thrombus formation on artificial surfaces, are all influenced by the shear stress, making it a quantity of significant clinical importance [65,69]. Therefore, we use the shear rate integral in our computations, i.e.,

$$J(u, p, \alpha) = 2 \int_{\Omega_{obs}(\alpha)} \varepsilon(u) : \varepsilon(u) \, dx,$$

(7.3)

where $\Omega_{obs}(\alpha) \subset \Omega(\alpha)$, represents the observation region. We also refer to (7.3) as the dissipation function. Note that in the case of (7.3), $J$ only depends on $u$ but not (explicitly) on $p$ or $\alpha$.

### 7.3 Numerical solution of the optimization problem

#### 7.3.1 Discretization of the optimization problem

To discretize the governing equations (7.1), we again apply a stabilized finite element discretization using conforming piecewise linear finite elements for the velocities and the pressure. Let $\{\Omega^e(\alpha) \mid e = 1, 2, \ldots, n_{el}\}$ be a triangulation of $\Omega(\alpha)$. We set

$$H^1_h(\Omega(\alpha)) = \{ \phi^h \mid \phi^h \in C^0(\Omega(\alpha)), \phi^h|_{\Omega^e(\alpha)} \in P^1, e = 1, 2, \ldots, n_{el} \},$$

$$S^h_u = \{ u^h \mid u^h \in [H^1(\Omega(\alpha))]^{n_{sd}}, u^h = g^h \text{ on } \Gamma_g(\alpha) \},$$

$$V^h_u = \{ u^h \mid u^h \in [H^1(\Omega(\alpha))]^{n_{sd}}, u^h = 0 \text{ on } \Gamma_g(\alpha) \}.$$

Let $x^h(\alpha) \in \Omega(\alpha)$ denote the nodes associated with the nodal basis for $H^1(\Omega(\alpha))$. We can then represent the discretized state equation, (3.2), symbolically as:

$$c^h(u^h(\alpha), p^h(\alpha), x^h(\alpha)) = 0,$$

(7.4)

We assume that for any $\alpha \in A_{ad}$ the equation (7.4) has a unique solution $u^h(\alpha), p^h(\alpha)$. 
The discretized shape optimization problem may now be written as:

\[
\begin{align*}
\text{minimize} \quad & \hat{J}^h(\alpha), \\
\text{subject to} \quad & \alpha \in \mathcal{A}_{ad}.
\end{align*}
\]

where

\[
\hat{J}^h(\alpha) = J^h(u^h(\alpha), p^h(\alpha), x^h(\alpha), \alpha).
\]

Note that in (7.5), the velocities and pressure, \(u^h(\alpha)\) and \(p^h(\alpha)\), are implicit functions of the design parameter \(\alpha\). These implicit functions are defined as the solution of (7.4). This results in the black-box or NAND approach of Section 5.3.2.

In (7.3), the velocities and pressures as well as the design parameters \(\alpha\) are optimization variables. The velocities and pressures are coupled to the design parameters \(\alpha\) through the governing equation, which in (7.3) is included as an explicit constraint. This view is referred to as the SAND approach of Section 5.3.1. The SAND approach could have also been used for the formulation of the discretized problem. The formulation of the optimization problem and the associated optimization algorithm can have a great impact on the efficiency with which the problem can be solved. In particular, SAND formulations combined with sequential quadratic programming (SQP) methods are very attractive. For the purpose of this chapter, however, the efficiency with which the optimization problem (7.5) is solved is secondary. Our optimization is built onto an existing complex flow code. Since SAND formulations combined with SQP methods require more code modifications than the gradient-based method we use to solve (7.5), we have chosen the latter.
7.3.2 Gradient computation

As stated, we use a gradient-based algorithm to solve the problem (7.5). Note that the design variables $\alpha$ enter $\hat{J}^h$ explicitly as well as implicitly through $\mathbf{v}^h = (u^h, p^h)$ and $x^h$. Hence, the gradient of $\hat{J}^h$ with respect to $\alpha$ is given by

$$\nabla \hat{J}^h = \frac{\partial J^h}{\partial \alpha} + \frac{\partial J^h}{\partial x^h} \frac{dx^h}{d\alpha} + \frac{\partial J^h}{\partial \mathbf{v}^h} \frac{d\mathbf{v}^h}{d\alpha}. \quad (7.6)$$

The Jacobian of the state variables $\mathbf{v}^h = (u^h, p^h)$ with respect to $\alpha$ can now be obtained using the implicit function theorem applied to (7.4). This gives

$$\frac{\partial c}{\partial \mathbf{v}^h} \frac{d\mathbf{v}^h}{d\alpha} + \frac{\partial c}{\partial x^h} \frac{dx^h}{d\alpha} = 0. \quad (7.7)$$

Equation (7.7) is referred to as the discrete sensitivity equation.

Using (7.7) in (7.6), we get

$$\nabla \hat{J}^h = \frac{\partial J^h}{\partial \alpha} + \frac{\partial J^h}{\partial x^h} \frac{dx^h}{d\alpha} - \frac{\partial J^h}{\partial \mathbf{v}^h} \left[ \left( \frac{\partial c}{\partial \mathbf{v}^h} \right)^{-1} \frac{\partial c}{\partial x^h} \frac{dx^h}{d\alpha} \right], \quad (7.8)$$

$$= \frac{\partial J^h}{\partial \alpha} + \frac{\partial J}{\partial x^h} \frac{dx^h}{d\alpha} - \left[ \left( \frac{\partial c}{\partial \mathbf{v}^h} \right)^{-1} \frac{\partial J}{\partial \mathbf{v}^h} \right]^T \frac{\partial c}{\partial x^h} \frac{dx^h}{d\alpha}. \quad (7.9)$$

Similar to the discussion in Section 5.3.2, equation 7.8 represents the sensitivity approach and equation 7.9 represents the adjoint approach. The element-wise evaluation of the term $\frac{\partial c}{\partial \alpha} \frac{dx^h}{d\alpha}$ is presented in Appendix C.

When using the Carreau-Yasuda model (2.7, 2.9), the viscosity $\mu$ depends on $u^h(\alpha)$ and on $x^h(\alpha)$. (The dependence on $u^h(\alpha)$ is obvious, the dependence on $x^h(\alpha)$ is due to the fact that the Jacobians of $u^h(\alpha)$ are required (see, e.g., [65] or [72]).)

We have already mentioned in Chapter 3 that in our computations, the terms involving partial derivatives $\frac{\partial c}{\partial \alpha}$ are dropped from the computation of $\frac{\partial c}{\partial x^h}$. We also drop terms involving partial derivatives $\frac{\partial c}{\partial x^h}$ from the computation of $\frac{\partial c}{\partial x^h}$. Therefore
our computed gradient $\nabla \tilde{J}^h(\alpha)$ is inexact when using the Carreau-Yasuda constitutive model.

Note that in this chapter, we follow the discretize-then-optimize approach, i.e., we first discretize the optimization problem to obtain (7.3) and then we solve the resulting nonlinear programming problem (7.5). Other approaches are possible. We refer to our discussion in Chapter 5, and to [37, 73, 74] for more details.

### 7.3.3 Mesh sensitivity and adaptation

For the gradient computation, we need to compute $\frac{dx^h}{d\alpha}$. We also need a method that, given a current design $\alpha$ and a new design $\alpha + \delta \alpha$, computes $x^h(\alpha + \delta \alpha)$ from $x^h(\alpha)$. In some cases, the domains and finite element meshes are simple enough so that we can explicitly determine the map $\alpha \mapsto x^h(\alpha)$. For discussions of other cases see, e.g., [46].

### 7.3.4 Optimization algorithm

The conceptual algorithm for the solution of the shape optimization problem is then

1. Initialize design variable $\alpha$ and compute $x^h(\alpha)$.

2. While not converged:

   (a) solve for $v^h$ using (7.4),
   (b) evaluate $\tilde{J}^h(\alpha) = J(u^h(\alpha), p^h(\alpha), x^h(\alpha), \alpha)$,
   (c) solve mesh sensitivity equation to get $\frac{dx^h}{d\alpha}$,
   (d) solve adjoint equation,
   (e) compute $\nabla \tilde{J}^h(\alpha)$,
(f) use $\nabla \tilde{J}^h(\alpha)$ to determine design update $\delta \alpha$. Set $\alpha = \alpha + \delta \alpha$,

(g) adapt the mesh to get $x^h(\alpha)$.

In our test cases $A_{ad} = \mathbb{R}^n$, and we use a BFGS quasi-Newton method [45, 75] with line search to solve (7.5). Specifically, we have interfaced our flow code with the optimization code [76], which can also handle constraints on the design parameters.

## 7.4 Numerical results

In all computations, the viscosity for the Newtonian case is chosen to be $\mu = \mu_\infty$, which is what one would have done had the model not been available [60]. The observation region $\Omega(\alpha)_{obs}$ for both the test cases is the entire flow domain.

The optimization algorithm is stopped if the norm of the gradient is less than $\| \nabla \tilde{J}(\alpha_0) \| \times 10^{-8}$ for the Newtonian constitutive equations and $\| \nabla \tilde{J}(\alpha_0) \| \times 10^{-7}$ for the Carreau-Yasuda model, where $\nabla \tilde{J}(\alpha_0)$ is the gradient at the initial design. The lower choice of the stopping criterion when using the Carreau-Yasuda model is due to the inexactness in our the gradient computation.

### 7.4.1 Cannula optimization

Our parameterization is motivated by [65] and is based on the centerline of the cannula shown in Figure 7.1:

$$r(\theta) = \sum_{i=0}^{P-1} r_i \cos(2i\theta), \quad \theta \in [0, \frac{\pi}{2}].$$  \hspace{1cm} (7.10)

Two conditions, which ensure that the entrance and the exit of the cannula are fixed, are used to express $r_0$ and $r_{P-1}$ as functions $r_i, i = 1, \ldots, P - 2$. Thus, design vector $\alpha$ consists of the coefficients $r_i, i = 1, \ldots, P - 2$. For a given centerline (7.10), the
nodal points $\mathbf{x}^h$ of the discretization are obtained by moving in the radial direction from the centerline by a constant distance. This defines the map $\alpha \mapsto \mathbf{x}^h(\alpha)$.

![Diagram of cannula geometry](image)

**Figure 7.1 : Cannula geometry**

The boundary conditions for the problem are traction-free at the exit, no-slip at all curved walls, and a specified parabolic inlet velocity. We present results for two different Reynolds numbers 50 and 300, defined as $\rho V_{max} d / \mu_{\infty}$, where $d = 1.0$ is the diameter of the cannula and $V_{max}$ is the maximum velocity at the inlet. The domain is discretized using 3960 triangular elements and 2100 nodes. For the case when the Reynolds number is 300, the optimization process gave a 56.66% reduction
in the dissipation function using the Newtonian constitutive equation, and a 53.86% reduction in the dissipation function using the Carreau-Yasuda model. The streamline plot of the velocity fields for the initial and optimal shapes for the two constitutive model choices are shown in Figures 7.2 and 7.3.

<table>
<thead>
<tr>
<th>Design</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_0$</td>
<td>5.6110</td>
<td>4.8740</td>
<td>4.8570</td>
</tr>
<tr>
<td>$r_1$</td>
<td>0.0000</td>
<td>-0.0424</td>
<td>-0.0340</td>
</tr>
<tr>
<td>$r_2$</td>
<td>-0.7800</td>
<td>0.1050</td>
<td>0.1161</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.0000</td>
<td>0.0170</td>
<td>0.0109</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0.2400</td>
<td>0.0181</td>
<td>0.0211</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0.0000</td>
<td>0.0099</td>
<td>0.0085</td>
</tr>
<tr>
<td>$r_6$</td>
<td>-0.1100</td>
<td>0.0070</td>
<td>0.0081</td>
</tr>
<tr>
<td>$r_7$</td>
<td>0.0000</td>
<td>0.0060</td>
<td>0.0055</td>
</tr>
<tr>
<td>$r_8$</td>
<td>0.0600</td>
<td>0.0035</td>
<td>0.0043</td>
</tr>
<tr>
<td>$r_9$</td>
<td>0.0000</td>
<td>0.0041</td>
<td>0.0038</td>
</tr>
<tr>
<td>$r_{10}$</td>
<td>-0.0300</td>
<td>0.0021</td>
<td>0.0026</td>
</tr>
<tr>
<td>$r_{11}$</td>
<td>0.0000</td>
<td>0.0030</td>
<td>0.0029</td>
</tr>
<tr>
<td>$r_{12}$</td>
<td>0.0200</td>
<td>0.0013</td>
<td>0.0017</td>
</tr>
<tr>
<td>$r_{13}$</td>
<td>0.0000</td>
<td>0.0024</td>
<td>0.0024</td>
</tr>
</tbody>
</table>

Table 7.1 : Design variables for the cannula optimization at $Re = 300$. Note that $r_0$ and $r_{13}$ are dependent variables.

From Figure 7.4, we observe that the optimal shapes depend little on the choice of the constitutive equation. The dependence of the optimal shape on constitutive
equation seems to increase as the Reynolds number increases. The numerical differences of the shape parameters defined in (7.10) is shown in Table 7.1. Figure 7.5 shows the effects of shear-thinning on the axial velocity profile for the initial shape. These results are qualitatively similar to those in [61]. We also observe that for lower Reynolds number, i.e., lower mean inflow axial velocity, the differences in flow profiles are more pronounced. Figure 7.6 shows that shape optimization has removed the sharp bend in the initial configuration of the cannula. This plot again shows that there is very little change in the shapes obtained from each of the constitutive equation.

Figure 7.2: The initial and optimal cannula shapes using the Newtonian constitutive equation, $Re = 300$. 
Figure 7.3: The initial and optimal cannula shapes using the Carreau-Yasuda constitutive model, $Re = 300$. 
Figure 7.4: Initial shape and optimal shape of the center line for the Newtonian and Carreau-Yasuda constitutive model for the different choices of the Reynolds number.
Figure 7.5: Axial velocity plot at various locations of the initial shape of the cannula. Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the left subplot are scaled by a factor of 6 relative to the right subplot to emphasize differences in downstream velocity profiles.
Figure 7.6: Axial velocity plot at various locations of the respective optimal shapes of the cannula. Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the left subplot are scaled by a factor of 6 relative to the right subplot to emphasize differences in downstream velocity profiles.
7.4.2 Arterial grafting

As a second test case, we consider the arterial grafting problem. A graft is attached upstream of the occlusion in the coronary artery, as an alternative route for blood flow. The boundary conditions for the modeled flow field are specified parabolic inlet velocity, no-slip boundary conditions on all walls including the graft, and a parallel flow condition at the outlet. We present results for two different Reynolds numbers 50 and 300, defined as $\rho V_{\text{max}}H/\mu_{\infty}$, where $H = 0.8$ is the height at the inlet and $V_{\text{max}}$ is the maximum velocity at the inlet.

![Diagram of arterial graft](image)

Figure 7.7: Computational domain for the arterial graft. The initial shape of the design curve of the graft (dashed line) is a semi-circle with center at $C$. Fixed geometry parameters are $l_1 = 6.0$, $l_2 = 3.0$, $l_3 + d = 2.5$. The downstream section of the host artery is symmetric with respect to the upstream one.

Figure 7.7 shows the geometry of the problem, with the flow proceeding from left to right. The geometry of the artery (rectangles in Figure 7.7) is fixed. The shape of the graft is optimized. We use the parameterization

$$r(\theta) = \sum_{i=0}^{P} r_i \theta^i$$

(7.11)

with $P = 5$, to represent the centerline of the graft.

Two conditions that ensure that location of the connection between graft and artery is fixed are used to express $r_0$ and $r_1$ in terms of the other ones. Hence, our
design variables $\alpha$ consist of $r_i, i = 2, \ldots, P$.

The mesh in the subdomain representing the artery is fixed. For a given centerline (7.11), the nodal points $x^h$ of the triangulation of the graft are obtained by moving in the radial direction from the centerline by a constant distance. This defines the map $\alpha \mapsto x^h(\alpha)$.

**Case 1**

We define the aspect ratio as the ratio of the diameter $d$ to height $H$. In the first case $d = 0.6$, and the aspect ratio is 0.75. The domain is discretized using 3137 triangular elements and 1774 nodes. For the case when the Reynolds number is 300, the optimization process led to a $14.98\%$ reduction in the dissipation function using the Newtonian constitutive equation, and a $16.38\%$ reduction in the dissipation function using the the Carreau-Yasuda model. The streamline plot of the velocity fields for the initial and optimal shapes for the two constitutive model choices are shown in Figures 7.8 and 7.9.

<table>
<thead>
<tr>
<th>Design Variable</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_2$</td>
<td>0.0</td>
<td>1.703</td>
<td>1.529</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.0</td>
<td>-1.013</td>
<td>-0.880</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0.0</td>
<td>0.265</td>
<td>0.228</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0.0</td>
<td>-0.022</td>
<td>-0.018</td>
</tr>
</tbody>
</table>

Table 7.2 : The design variables for the graft optimization for case 1, $Re = 300$.

Figure 7.10 shows that for an aspect ratio of 0.75 the optimal shapes are independent of the choice of the constitutive model, but vary somewhat with Reynolds
Table 7.3: The angle that the graft makes with the artery at its inlet and exit, for case 1.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 0, , Re = 50$</td>
<td>90.00°</td>
<td>69.70°</td>
<td>70.81°</td>
</tr>
<tr>
<td>$\theta = \pi, , Re = 50$</td>
<td>90.00°</td>
<td>68.21°</td>
<td>68.92°</td>
</tr>
<tr>
<td>$\theta = 0, , Re = 300$</td>
<td>90.00°</td>
<td>61.78°</td>
<td>62.89°</td>
</tr>
<tr>
<td>$\theta = \pi, , Re = 300$</td>
<td>90.00°</td>
<td>60.86°</td>
<td>62.15°</td>
</tr>
</tbody>
</table>

number. Table 7.2 shows the computed optimal shape parameters defined in (7.11), and Table 7.3 shows the values of the angles that the graft makes at the inlet and exit of the artery.

Figure 7.11 shows that that although the axial velocity profiles for the smaller Reynolds number is qualitatively larger, for either Reynolds number the flow profiles are not that different. A similar effect is seen for the optimal shape velocity profiles, as seen in Figure 7.12. Note that for Figure 7.11 and Figure 7.12, we use a different scaling for the two inflow Reynolds numbers.

Figure 7.8: The graft streamline velocity profile for the initial and the optimal shape, using the Newtonian constitutive equation for case 1, $Re = 300$. 
Figure 7.9: The graft streamline velocity profile for the initial and the optimal shape, using the Carreau-Yasuda constitutive model for case 1, $Re = 300$.

Figure 7.10: Initial shape and optimal shape of the graft design curve for the Newtonian and Carreau-Yasuda constitutive model for case 1.
Figure 7.11: Axial velocity plot at various locations of the initial graft shape for case 1, with $Re = 50$ (top) and $Re = 300$ (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles.

Figure 7.12: Axial velocity plot at various locations of the optimal graft shapes using each of the constitutive equation (overlapped here), for case 1, with $Re = 50$ (top) and $Re = 300$ (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles.
Case 2

In the second case we let $d = 1.0$, and the aspect ratio is 1.25. The domain is discretized using 3643 triangular elements and 2017 nodes. By increasing the diameter $d$, the local Reynolds number at the bypass graft inlet is reduced and therefore the shear rate is also lowered. For the case when the Reynolds number is 300, the optimization process gave a 29.0% reduction in the dissipation function using the Newtonian constitutive equation, and a 21.3% reduction in the dissipation function using the the Carreau-Yasuda model. The streamline plot of the velocity fields for the initial and optimal shapes for the two constitutive model choices are shown in Figures 7.13 and 7.14.

<table>
<thead>
<tr>
<th>Design Variable</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_2$</td>
<td>0.0</td>
<td>2.024</td>
<td>1.953</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.0</td>
<td>-1.038</td>
<td>-1.063</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0.0</td>
<td>0.238</td>
<td>0.261</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0.0</td>
<td>-0.014</td>
<td>-0.019</td>
</tr>
</tbody>
</table>

Table 7.4 : The design variables for the graft optimization for case 2, $Re = 300$. 

We see from Table 7.4, the numerical differences in the optimal shape parameters defined in (7.11) when $Re = 300$. Figure 7.15 and Table 7.5 show the significant influence of the shear-thinning property on the obtained optimal shape. Specifically, for the higher Reynolds number there is a larger difference in the computed shape. Moreover, the angle that the graft makes with the artery depends significantly on the constitutive equation used, as well as on the Reynolds number. For $Re = 50$
<table>
<thead>
<tr>
<th>Angle</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 0, \ Re = 50$</td>
<td>90.00°</td>
<td>63.50°</td>
<td>68.35°</td>
</tr>
<tr>
<td>$\theta = \pi, \ Re = 50$</td>
<td>90.00°</td>
<td>59.65°</td>
<td>67.18°</td>
</tr>
<tr>
<td>$\theta = 0, \ Re = 300$</td>
<td>90.00°</td>
<td>50.12°</td>
<td>53.32°</td>
</tr>
<tr>
<td>$\theta = \pi, \ Re = 300$</td>
<td>90.00°</td>
<td>46.27°</td>
<td>52.24°</td>
</tr>
</tbody>
</table>

Table 7.5: The angle that the graft makes with the artery at its inlet and exit, for case 2.

the differences in angles computed with Newtonian and Carreau-Yasuda constitutive equation, respectively, seem slightly larger than for $\Re = 300$.

Figure 7.16 shows a strong influence of the axial velocity profiles due to the inclusion of the shear-thinning model. We see from Figure 7.17, that due to the difference in the shapes obtained from the choice of the respective constitutive equation, there is a qualitative similarity in the flow profiles in the domain.

![Initial and optimal shape](image)

Figure 7.13: The graft streamline velocity profile for the initial and the optimal shape using the Newtonian constitutive equation for case 2, $\Re = 300$. 
Figure 7.14: The graft streamline velocity profile for the initial and the optimal shape, using the Newtonian constitutive equation for case 2, \( Re = 300 \).

Figure 7.15: Initial shape and optimal shape of the graft design curve for the Newtonian and Carreau-Yasuda constitutive model for case 2.
Figure 7.16: Axial velocity plot at various locations of the initial graft shape for case 2, with $Re = 50$ (top) and $Re = 300$ (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles.

Figure 7.17: Axial velocity plot at various locations of the optimal graft shapes using each of the constitutive equation (overlapped here), for case 2, with $Re = 50$ (top) and $Re = 300$ (bottom). Solid line represents Newtonian flow and dashed line represents the flow using the Carreau-Yasuda constitutive model. The velocities in the top subplot are scaled by a factor of 6 relative to the bottom subplot to emphasize differences in downstream velocity profiles.
7.5 Conclusions

We have outlined the solution of shape optimization problems governed by the Navier-Stokes equations with a generalized Newtonian constitutive model. The governing equations are discretized by stabilized finite elements. The resulting optimization problem is solved using a gradient-based method. Our numerical results show the benefits of numerical shape optimization in achieving design improvements.

For the optimization of the cannula shape, we observe that the differences in the optimal shapes due to constitutive equation choice do not seem to be significant. This was true for both inflow Reynolds numbers considered. The computed flow profile did show differences, especially for the lower inflow velocity. However this did not affect the optimal shape of the cannula. The choice of constitutive equation did impact the optimal shape in our second example, the optimization of an idealized arterial graft. For a smaller aspect ratio, the optimal shapes do not differ much for either choice of Reynolds number. However, when we increased the aspect ratio, the flow profiles significantly differ between the generalized Newtonian constitutive model and the Newtonian one, and the optimal shapes obtained were significantly different. This was especially the case for higher Reynolds number, with dominant flow features such as recirculation more prevalent when using the Newtonian constitutive equation.

A Newtonian assumption is valid for flows where shear rates are high, thus diminishing the shear-related viscosity differences as seen in (2.9). The objective function employed here, which is relevant for many blood related design objectives, tries to minimize the global shear rate. Thus, our shape optimization tends to select shapes for which also the local shear rate tends to be smaller. In these cases, we expect a stronger influence of the constitutive model on the computed shape.
Chapter 8

Shape Optimization for Unsteady Flows

In this chapter, we present a gradient-based methodology to solve shape-optimization problems in unsteady fluid flows. Following an introduction to the issues involved in solving unsteady optimal design problems in Section 8.1, we present a model shape optimization problem and the necessary conditions of optimality in Section 8.2. In Section 8.3, we present the methodology to solve the model optimization problem using the semi-discrete finite element discretization. In Section 8.4, we present the methodology to solve the model optimization problem using the space-time finite element method. In both these sections, we discuss the sensitivity-based approach, and the discretize-then-optimize, as well as the optimize-then-discretize adjoint-based method for computing the gradients with respect to the design parameters. In Section 8.5, we discuss a strategy for storing the state solution needed for evaluating the adjoints. In Section 8.6, we report on the numerical results for the unsteady variant of the arterial graft test case discussed in Chapter 7. We will also show how the constitutive equation choice for blood affects the solution of the unsteady optimal design problem.

8.1 Introduction

A major focus of CFD research in the past few years has been on simulating the unsteady behavior of fluid flows. The time-evolving pattern of fluid flows can be
attributed to a variety of factors— the temporal nature of the flow conditions, e.g.,
in pulsatile and excited flows, dominant convective forces, e.g., in bluff body wakes
and turbulent flows, and finally, to deforming spatial domains, e.g., in free-surface
flows, flows past rotating and translating components, fluid-structure interaction
problems.

As previously indicated, significant progress has been made in solving optimal
design problems involving fluid flows. This led to the usage of CFD-based numerical
optimization techniques aiding engineering decisions in an automatic fashion, elimi-
nating the traditional trial and error approach. However, most of the work done in
optimal design in CFD has focused on steady flows. Recently, there has been grow-
ing interest in extending optimal design capabilities to unsteady flows, expanding
greatly the range of applications. A significant portion of recent research in shape
optimization procedures for unsteady flows has been in the context of aerodynamics [46,77].

This chapter studies the solution of optimal design problems in the context of
unsteady flows, with an emphasis on applications in biomedical engineering involving
pulsatile blood flow. As mentioned in Chapter 7, one of the target applications is
the optimal design of centrifugal blood pumps [78]. Blood flow in these pumps is
simulated using the Deformable-Spatial-Domain/Stabilized-Space-Time (DSD/SST)
finite element formulation [5,17], as well as the Shear-Slip Mesh Update Method
(SSMUM) [24,26], which accounts for the mesh deformation due to the rotating
impeller. A shape optimization problem in this context could involve finding the
shape of pump components based on a certain parameterization, that minimizes
hemolysis.

With the final goal of solving large-scale shape optimization problems for un-
steady flows, potentially involving moving boundaries, this chapter builds on concepts introduced in Chapters 5 and 7, and presents a gradient-based optimization framework. Our focus is on the accurate estimation of the gradients. Owing to the large-scale nature of the target optimization problem, adjoint methods may be favored over the sensitivity approach for gradient evaluation. For steady flows, it was possible to clearly see the cost advantage of using the adjoint approach in favor of the sensitivity approach. However for unsteady problems, the adjoint equation is solved backwards in time, and in principle, requires storage of the state variables for all time steps. For large-scale simulations, this poses a formidable obstacle which can be circumvented by recomputing the state variables at intermediate time steps. For sufficiently large-scale optimization problems, these efficient memory storage schemes coupled with the adjoint method might have lower overall computational cost. Therefore, in this chapter we present an approach to gradient computation using the adjoint method as well.

One can evaluate the adjoint variables either through the optimize-then-discretize or the discretize-then-optimize approaches introduced in Chapter 5. In Chapter 6, we numerically investigated the differences in the computed control for a steady Oseen boundary control problem for the GLS-stabilized finite element method. The problem considered in Chapter 6 is convex, linear-quadratic. Hence, it is equivalent to the linear system of optimality conditions. Due to the non-convexity of the problem considered in this chapter, such a reformulation of the optimization problem is no longer possible and we have to use an iterative gradient based method. When using a gradient-based method to solve the optimal control problem, the optimization solver evaluates the next control values, based on the computed gradient to the discretized objective function at the current iteration. The discretize-then-optimize
approach constructs an exact gradient of the discretized objective function, which is an approximation of the continuous objective function. On the other hand, the optimize-then-discretize approach constructs the gradient from the discretized continuous adjoint equation derived from the continuous state equation and continuous objective function. This gradient may not be consistent with the discretized objective function. Therefore, the gradient computed using the optimize-then-discretize approach might drive the optimization algorithm in an inconsistent direction. In the limit of sufficient refinement, both the approaches should yield similar gradients to the continuous objective function [77].

Motivated by the results of Chapter 6, this chapter presents the differences in the adjoint and gradient using the two aforementioned approaches, for both the semi-discrete and space-time GLS formulations, for a model optimal design problem. Further, we continue our study of the influence of constitutive equation on the solution of shape optimization problems from Chapter 7, extending it to unsteady flows. We consider again the arterial grafting problem, introduced in Section 7.4, now in the context of pulsatile flow. Our numerical study is a step in the direction of solving large-scale shape optimization problems for the centrifugal blood pump.

### 8.2 A model shape optimization problem

Our goal is to find a shape $\Omega$, such that a given objective function $J$, which depends on $u(x, t), p(x, t)$ and on $\Omega$ itself, is minimized. We consider the case where the set of admissible shapes can be parameterized by $\alpha \in \mathcal{A}_{ad} \subset \mathbb{R}^n$. The optimal shape
design problem is given as follows:

\[
\begin{align*}
\text{minimize} & \quad J(u, p, \alpha), \\
\text{subject to} & \quad (5.2) \text{ with } \Omega = \Omega(\alpha), \quad \alpha \in \mathcal{A}_{ad}.
\end{align*}
\]

Similarly to Chapter 7, we consider the shear rate integral:

\[
J(u, p, \alpha) = \int_{t_L}^{t_U} \Phi(u, \alpha),
\]

a scalar metric, where \( \Phi(u, \alpha) = 2 \int_{\Omega_{elas}(\alpha)} \varepsilon(u) : \varepsilon(u) dx \) is consistent with (7.3), and \( t_U \) corresponds to the final time \( t_U = T \). We do not include the external force field \( f \) contribution for our discussion in this chapter. The Lagrangian associated with the optimal design problem can be written as:

\[
L(u, p, \lambda, \theta, \alpha) = J(u, p, \alpha) \\
+ \int_{t=0}^{T} \int_{\Omega(\alpha)} \lambda \cdot \left[ \rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) - \nabla \cdot \sigma(u, p) \right] dxdt \\
+ \int_{t=0}^{T} \int_{\Omega(\alpha)} \theta \nabla \cdot u dxdt.
\]

The variables \( \lambda(x, t) \) and \( \theta(x, t) \) respectively represent the adjoint velocity and pressure. The necessary conditions of optimality are obtained by setting the Fréchet derivatives of the Lagrangian with respect to the adjoint variables \( z = (\lambda, \theta)^T \), the state variables \( v = (u, p)^T \) and with respect to \( \alpha \) to zero. The necessary conditions
consist of the state equation (2.1), the adjoint equation:
\[
\rho \left( -\frac{\partial \lambda}{\partial t} + \nabla u^T \lambda - u \cdot \nabla \lambda \right) - \nabla \cdot \sigma(\lambda, \theta) + \chi(t)D_u \Phi(u) = 0 \text{ on } \Omega(\alpha) \quad \forall t \in I,
\]
\[
\nabla \cdot \lambda = 0 \text{ on } \Omega(\alpha) \quad \forall t \in I,
\]
\[
n \cdot \sigma(\lambda, \theta) + (u \cdot n)\lambda = 0 \text{ on } \Gamma_h(\alpha) \quad \forall t \in I,
\]
\[
\lambda(x, t) = 0 \text{ on } \Gamma_g(\alpha) \quad \forall t \in I,
\]
\[
\lambda(x, T) = 0 \text{ on } \Omega(\alpha), \quad (8.4a)
\]
where \( I = [0, T] \), and the gradient equation:
\[
\frac{\partial L}{\partial \alpha} = \frac{\partial J}{\partial \alpha} + \frac{\partial}{\partial \alpha} \int_{t=0}^{T} \int_{\Omega(\alpha)} \lambda \cdot \left[ \rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) - \nabla \cdot \sigma(u, p) \right] dx dt
\]
\[
+ \frac{\partial}{\partial \alpha} \int_{t=0}^{T} \int_{\Omega(\alpha)} \theta \nabla \cdot u dx dt = 0. \quad (8.4b)
\]
For the objective function (8.2), \( D_u \Phi(u) \) is given by
\[
\langle \langle D_u \Phi(u), w \rangle \rangle = 4 \int_{\Omega_{obs}(\alpha)} e(u) : e(w) dx, \quad (8.5)
\]
and
\[
\chi(t) = \begin{cases} 
0 & \text{for } t < t_L, \\
1 & \text{for } t_L \leq t \leq t_U. 
\end{cases} \quad (8.6)
\]

Remark 8.2.1. Note that we have not included the variation of \( \sigma \) with \( \mu(u) \) in the derivation of (8.4a), which arises in the case of a generalized Newtonian fluid. Such a term would be present for the exact continuous adjoint PDE.

8.3 Numerical solution of the optimization problem: semi-discrete formulation

We first present the solution methodology using the semi-discrete formulation, previously described in Section 3.2.
8.3.1 Discretization of the optimization problem

As in Chapter 3, we discretize the governing equations (2.1), by applying the GLS stabilized finite element discretization using conforming piecewise linear finite elements for the velocities and the pressure. Let \( \{ \Omega^c (\alpha) \mid c = 1, 2, \ldots, n_{el} \} \) be a triangulation of \( \Omega(\alpha) \). We set

\[
H^h(\Omega(\alpha)) = \left\{ \phi^h \mid \phi^h \in C^0(\Omega(\alpha)), \phi^h|_{\Omega(\alpha)} \in P^1, c = 1, 2, \ldots, n_{el} \right\},
\]

\[
S^h_u = \{ u^h \mid u^h \in [H^h(\Omega(\alpha))]^{n_{sd}}, u^h = g^h \text{ on } \Gamma_g(\alpha) \},
\]

\[
V^h_u = \{ u^h \mid u^h \in [H^h(\Omega(\alpha))]^{n_{sd}}, u^h = 0 \text{ on } \Gamma_g(\alpha) \}.
\]

Let \( x^h(\alpha) \in \Omega(\alpha) \) denote the nodes associated with the nodal basis for \( H^h(\Omega(\alpha)) \). Furthermore, let \( v_i \) represent the solution \( v^h = (u^h, p^h)^T \) at time step \( i \). For the \( i \)th time step, we can represent (3.2), using (3.4), symbolically as:

\[
c_i(v_{i+1}(\alpha), v_i(\alpha), x^h(\alpha)) = 0, \quad i = 0, \ldots, L, \ldots, U - 1. \tag{8.7}
\]

We assume that for any \( \alpha \in A_{ad} \), the equation (8.7) has a unique solution \( v_i(\alpha), \quad i = 1, \ldots, U \). The discretized shape optimization problem may now be written as:

\[
\begin{align*}
\text{minimize} & \quad \hat{J}^h(\alpha), \\
\text{subject to} & \quad \alpha \in A_{ad}.
\end{align*} \tag{8.8}
\]

where

\[
\hat{J}^h(\alpha) = J^h(v_L(\alpha), \ldots, v_U(\alpha), x^h(\alpha), \alpha),
\]

\[
= \frac{\Delta t}{2} \sum_{i=L}^{U-1} [\Phi^h(u_i(\alpha), x^h(\alpha)) + \Phi^h(u_{i+1}(\alpha), x^h(\alpha))]. \tag{8.9}
\]
Note that in (8.9), the velocities and pressure, $u^h(\alpha)$ and $p^h(\alpha)$, are implicit functions of the design parameter $\alpha$. These implicit functions are defined as the solution of (8.7).

We use a gradient-based algorithm to solve (8.8). The design variables $\alpha$ enter (8.8) implicitly accounting for the dependence of the state field $v_i, i = 0, \ldots, L, \ldots, U$, as well as the nodal coordinates $x^h$ on $\alpha$, and explicitly, whenever regularization terms are included in the objective function. Using the implicit function theorem on (8.7), we obtain the first-order discrete sensitivity equations as:

$$\frac{\partial c_i}{\partial v_{i+1}} \frac{dv_{i+1}}{d\alpha} = -\frac{\partial c_i}{\partial v_i} \frac{dv_i}{d\alpha} - \frac{\partial c_i}{\partial \alpha} \frac{dx^h}{d\alpha}, \quad i = 0, \ldots, L, \ldots, U - 1. \quad (8.10)$$

Let $\Phi_{i+1} = \Phi^h(u_{i+1}(\alpha), x^h(\alpha))$; then the sensitivity $d\Phi_{i+1}/d\alpha$ at time $t_{i+1}$ is:

$$\frac{d\Phi_{i+1}}{d\alpha} = \frac{\partial \Phi_{i+1}}{\partial x^h} \frac{dx^h}{d\alpha} + \frac{\partial \Phi_{i+1}}{\partial v_{i+1}} \frac{dv_{i+1}}{d\alpha}, \quad i = 0, \ldots, L, \ldots, U - 1. \quad (8.11)$$

Using (8.9), the gradient $\nabla \hat{J}^h$ is given as

$$\nabla \hat{J}^h = \frac{\partial J^h}{\partial \alpha} + \frac{\Delta t}{2} \sum_{i=L}^{U-1} \left[ \frac{d\Phi_i}{d\alpha} + \frac{d\Phi_{i+1}}{d\alpha} \right]. \quad (8.12)$$

The term $\frac{dx^h}{d\alpha}$ is evaluated using the techniques mentioned in [79–81]. The recipe for $\frac{\partial c_i}{\partial x^h} \frac{dx^h}{d\alpha}$ is shown in Appendix C.

Let $z_i$ represent the adjoint variables $z^h = (\lambda^h, \theta^h)$ at time step $i$. To evaluate the gradient using the discretize-then-optimize adjoint approach, we write the discrete
Lagrangian as:

\[
L(v_0, \ldots, v_U, z_1, \ldots, z_U, \alpha) = J^h(v_L, \ldots, v_U, x^h(\alpha), \alpha) + \sum_{i=0}^{U-1} z_{i+1}^T c_i(v_{i+1}, v_i, x^h(\alpha)) + \Delta t \sum_{i=L}^{U-1} [\Phi_i + \Phi_{i+1}] + \sum_{i=0}^{U-1} z_{i+1}^T c_i(v_{i+1}, v_i, x^h(\alpha)).
\]

(8.13)

Notice that in (8.13), we have not written the dependence of the state \(v_i\) on \(\alpha\) explicitly. This is because we are explicitly accounting for the state equation. We only write \(v_i(\alpha)\) when using the modified objective function (8.8). The adjoint equation is then obtained by setting \(\frac{\partial L}{\partial v_i} = 0\), for \(i = 1, \ldots, U - 1\). This yields:

\[
\left(\frac{\partial c_{i-1}(v_i, v_{i-1}, x^h(\alpha))}{\partial v_i}\right)^T z_i = -\left(\frac{\partial c_i(v_{i+1}, v_i, x^h(\alpha))}{\partial v_i}\right)^T z_{i+1} - \chi_i \Delta t \left(\frac{\partial \Phi_i}{\partial v_i}\right)^T, \quad i = U - 1, U - 2, \ldots, 1.
\]

(8.14)

where \(\chi_i\) is defined as

\[
\chi_i = \begin{cases} 
0 & \text{for } 0 \leq i < L, \\
0.5 & \text{for } i = L, \\
1 & \text{for } L < i < U - 1.
\end{cases}
\]

(8.15)

The gradient is then obtained as follows:

\[
\frac{\partial L}{\partial \alpha} = \frac{\partial J^h}{\partial \alpha} + \frac{\Delta t}{2} \sum_{i=L}^{U-1} \left[ \frac{\partial \Phi_i}{\partial x^h} + \frac{\partial \Phi_{i+1}}{\partial x^h} \right] \frac{dx^h}{d\alpha} + \Delta t \sum_{i=0}^{U-1} z_{i+1} \frac{\partial c_i(v_{i+1}, v_i, x^h(\alpha))}{\partial x^h} \frac{dx^h}{d\alpha}.
\]

(8.16)
We have already pointed out the equivalence between (8.12) and (8.16) in Chapter 5. The algorithm for the solution of the shape optimization problem is then

1. Initialize design variable \( \alpha \) and generate \( x^h(\alpha) \).

2. Set the initial conditions on the velocity \( u^0 \) and the sensitivity \( \frac{du}{d\alpha} \).

3. While not converged:
   
   (a) solve mesh sensitivity equation to obtain \( \frac{dx^h}{d\alpha} \).
   
   (b) for each time step \( i, i = 1, \ldots, U \):

   i. solve for \( v_{i+1}(\alpha) \) using (8.7).

   ii. if sensitivity approach; solve for \( \frac{dv_{i+1}(\alpha)}{d\alpha} \) using (8.10), accumulate objective function \( \tilde{J} \) if necessary using (8.9), and accumulate \( \nabla \tilde{J}(\alpha) \) using (8.11) and (8.12).

   (c) if adjoint approach, solve for \( z \) and \( \nabla \tilde{J}^h(\alpha) \) using (8.14) and (8.16),

   (d) use \( \nabla \tilde{J}^h \) to determine design update \( \delta \alpha \). Set \( \alpha = \alpha + \delta \alpha \),

   (e) adapt the mesh to obtain \( x^h(\alpha) \).

### 8.3.2 Discretization of the optimality conditions

The optimality conditions consist of the state equation (2.1) and the equations (8.4). In the optimize-then-discretize approach, we discretize the infinite dimensional adjoint PDE given by (8.4a). We can choose different discretization schemes for the state and the adjoint equations; here we will present a formulation when using the same discretization scheme for the adjoint equation and the state equation. Our discretization of the state equation is given by (3.2). The GLS-stabilized formulation for the adjoint equation (8.4a) can be written as follows: given \( u^h \) and \( p^h \) satisfying
(3.2) and \(\lambda_U\), find \(\lambda^h \in V_u^h\) and \(\theta^h \in H^1(a)\), such that \(\forall w^h \in V_u^h\) and \(\forall q^h \in H^1(a)\):

\[
\int_{\Omega(a)} w^h \cdot \rho \left( -\frac{\partial \lambda^h}{\partial t} + (\nabla u^h)^T \lambda^h - u^h \cdot \nabla \lambda^h \right) dx + \int_{\Omega(a)} \varepsilon(w^h) : \sigma(\lambda^h, \theta^h) dx \\
+ \sum_{e=1}^{n_e} \int_{\Omega^e(a)} \frac{\tau_e}{\rho} \left[ \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right] dx \\
+ \int_{\Omega(a)} q^h \nabla \cdot \lambda^h dx = -\chi(t) \langle \langle D_u \Phi^h(u^h), w^h \rangle \rangle_s,
\]

where \(\langle \langle D_u \Phi^h(u^h), w^h \rangle \rangle_s\) is the stabilized functional given by:

\[
\langle \langle D_u \Phi^h(u^h), w^h \rangle \rangle_s = 4 \int_{\Omega(a)} \varepsilon(u^h) : \varepsilon(w^h) dx \\
+ 4 \sum_{e=1}^{n_e} \int_{\Omega^e(\alpha) \cap \Omega(a)} \frac{\tau_e}{\rho} \varepsilon(u^h) dx \\
\varepsilon \left( \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right) dx.
\]

Let \(\lambda_i\) represent the adjoint velocity field in the domain at time step \(i\). The adjoint equation is (8.17) is then solved sequentially for time steps \(i = U, U - 1, \ldots, 1\) with

\[
\lambda_U = 0.
\]

Using the generalized midpoint rule for the time-discretization of \(\lambda^h\), which can be written as:

\[
\lambda^h = \gamma \lambda_i + (1 - \gamma) \lambda_{i-1}, \quad \frac{\partial \lambda^h}{\partial t} = \frac{\lambda_i - \lambda_{i-1}}{\Delta t},
\]

where \(0 \leq \gamma \leq 1\).

The gradient can be then represented as:

\[
\frac{\partial L}{\partial \alpha} = \frac{\partial J^h}{\partial \alpha} + \Delta t \sum_{i=L+1}^{U} \frac{\partial \Phi^h(u^h)}{\partial x^h} \frac{dx^h}{\partial \alpha} + \Delta t \sum_{i=1}^{U} \frac{\partial}{\partial \alpha} \\
\int_{\Omega(a)} \left[ \lambda^h \cdot \left( \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h \right) - \nabla \cdot \sigma(u^h, p^h) \right) + \theta^h \nabla \cdot u^h \right] dx.
\]
where the index $i$ is due to $\lambda^h$ given by (8.20), and $u^h$ given by replacing $\lambda$ with $u$ in (8.20).

### 8.3.3 Difference in gradients between the discretize-then-optimize (do) and optimize-then-discretize (od) approaches

Using integration by parts on (8.17), and dropping the second derivatives in the stabilization terms due to our choice of piecewise-linear elements, the implemented semi-discrete GLS-stabilized formulation for the adjoint equation (8.4a) can be written as follows: given $u^h$ and $p^h$, find $\lambda^h \in V^h_u$ and $\theta^h \in H^1(\Omega(\alpha))$, such that $\forall w^h \in V^h_u$ and $\forall q^h \in H^1(\Omega(\alpha))$:

\[
\int_{\Omega(\alpha)} \lambda^h \cdot \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) \, dx + \int_{\Omega(\alpha)} \varepsilon(w^h) : \sigma(\lambda^h, \theta^h) \, dx \\
+ \sum_{e=1}^{n_e} \int_{\Omega^e(\alpha)} \frac{\tau_e}{\rho} \left[ \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) + \nabla q^h \right] \quad \left[ \rho \left( -\frac{\partial u^h}{\partial t} + (\nabla u^h)^T \lambda^h - u^h \cdot \nabla \lambda^h \right) + \nabla \theta^h \right] \, dx + \int_{\partial \Omega(\alpha)} q^h \nabla \cdot \lambda^h \, dx,
\]

\[
= -\chi(t) \langle \langle D_u \Phi(u^h), w^h \rangle \rangle_s,
\]  

(8.22)

with the implemented version of $\langle \langle D_u J(u^h), w^h \rangle \rangle_s$ given by:

\[
\langle \langle D_u J(u^h), w^h \rangle \rangle_s = 4 \int_{\Omega_{obs}(\alpha)} \varepsilon(u^h) : \varepsilon(w^h) \, dx \\
+ 4 \sum_{e=1}^{n_e} \int_{\Omega^e(\alpha) \cap \Omega_{obs}(\alpha)} \frac{\tau_e}{\rho} \varepsilon(u^h) : \varepsilon \left( \rho \left( -\frac{\partial w^h}{\partial t} + \nabla u^T w^h - u \cdot \nabla w^h \right) \right) \, dx.
\]

(8.23)

The do adjoint, given in (8.14), can be viewed as: given $u^h$ and $p^h$, find $\lambda^h \in V^h_u$.
and $\theta^h \in H^1(\Omega(\alpha))$, such that $\forall w^h \in V^h_u$ and $\forall q^h \in H^1(\Omega(\alpha))$:

$$
\int_{\Omega(\alpha)} \chi^h \cdot \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) \, dx + \int_{\Omega(\alpha)} \varepsilon(w^h) : \sigma(\chi^h, \theta^h) \, dx
+ \sum_{e=1}^{n_{el}} \int_{\Omega^e(\alpha)} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h \right) + \nabla q^h \right] \cdot \left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla \chi^h \right) + \nabla \theta^h \right] \, dx
+ \int_{\Omega(\alpha)} q^h \nabla \cdot \lambda^h \, dx,
$$

$$
= -4\chi(t) \int_{\Omega_{obs}(\alpha)} \varepsilon(u^h) : \varepsilon(w^h) \, dx.
$$

Note that in (8.24), we could have chosen the time-discretization of the objective function analogous to (8.13). However, for this discussion on the comparison of the od and do approaches, we have chosen the same time-discretization scheme of the objective function for both approaches.

The differences between the od adjoint (8.22) and the do adjoint (8.24) are:

1. The term $(\nabla u^h)^T w^h$ and $(\nabla u^h)^T \lambda^h$ in the stabilization terms of od formulation.

2. The stabilization terms on the right-hand side in the od formulation, i.e.,

$$
4 \sum_{e=1}^{n_{el}} \int_{\Omega^e(\alpha) \cap \Omega_{obs}(\alpha)} \frac{\tau_e}{\rho} \varepsilon(u^h) : \varepsilon \left( \rho \left( -\frac{\partial w^h}{\partial t} + \nabla u^T w^h - u \cdot \nabla w^h \right) \right) \, dx.
$$

The gradient calculated using the od approach is given by (8.21). The gradient calculated using the do approach is given by:

$$
\frac{\partial L}{\partial \alpha} = \frac{\partial J}{\partial \alpha} + \Delta t \sum_{i=1}^{U} \frac{\partial \Phi^h(u^h)}{\partial \chi^h} \frac{dx^h}{\partial \alpha} + \Delta t \sum_{i=1}^{U} \frac{\partial}{\partial \alpha}
\int_{\Omega(\alpha)} \chi^h \cdot \left( \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h \right) - \nabla \cdot \sigma(u^h, p^h) \right) + \theta^h \nabla \cdot u^h \right] \, dx
+ \Delta t \sum_{i=1}^{U} \frac{\partial}{\partial \alpha} \sum_{e=1}^{n_{el}} \int_{\Omega^e(\alpha)} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{\partial \chi^h}{\partial t} + u^h \cdot \nabla \chi^h \right) + \nabla \theta^h \right] \cdot \left[ \rho \left( \frac{\partial \chi^h}{\partial t} + u^h \cdot \nabla \chi^h \right) + \nabla \theta^h \right] \, dx,
$$

(8.25)
where the index $i$ is due to $\lambda^h$ given by (8.20), and $u^h$ given by replacing $\lambda$ with $u$ in (8.20). On comparing the gradient equations (8.21) and (8.25), we see that, when using the DO approach, the gradient has added shape-sensitivity terms due to the stabilization in the discretized Lagrangian. These terms are absent when using the OD approach.

### 8.4 Numerical solution of the optimization problem: space-time formulation

We now consider the solution of the optimization problem using the discontinuous-in-time space-time finite element formulation (see Section 3.3).

#### 8.4.1 Discretization of the optimization problem

We define the following finite element interpolation spaces for the velocity and pressure on each space-time slab (recall Figure 3.1):

$$H^1(Q_t(\alpha)) = \{ \phi^h \mid \phi^h \in C^0(Q_t(\alpha)), \phi^h|_t(\alpha) \in P^1, e = 1, 2, \ldots, (n_{el})_t \},$$

$$(S^h_u)_i = \{ u^h \mid u^h \in [H^1(Q_t(\alpha))]^{n_{sl}}, u^h = g^h \text{ on } (P_t(\alpha))_g \},$$

$$(V^h_u)_i = \{ u^h \mid u^h \in [H^1(Q_t(\alpha))]^{n_{sl}}, u^h = 0 \text{ on } (P_t(\alpha))_g \}.$$  

The solution to the state equation (3.6) is obtained for all space-time slabs $Q_1(\alpha), \ldots, Q_L(\alpha), \ldots, Q_{U-1}(\alpha)$ sequentially, with

$$(u^h)_1 = u_0.$$  

(8.26)

We represent (3.6) symbolically for the $i$th time-step as:

$$c_i(v_i(\alpha), u_i^- (\alpha), x_i^h(\alpha)) = 0, \quad i = 1, \ldots, L, \ldots, U - 1.$$  

(8.27)
where $v_i^h(\alpha)$ represents the solution $v^h$ at space-time slab $Q_i(\alpha)$ and $x_i^h(\alpha)$ represents the spatial discretization for that slab. Using the notation $Q^+_i(\alpha)$ to represent the observation region encompassed within $\Omega_{\text{obs}}(\alpha)_i$ and $\Omega_{\text{obs}}(\alpha)_{i+1}$, we define the objective function $\Psi(u_i(\alpha), x_i^h(\alpha)) = 2\int_{Q^+_i(\alpha)} e(u_i(\alpha)) : e(u_i(\alpha)) dQ$. We assume that for any $\alpha \in \mathcal{A}_{ad}$, the equation (8.27) has a unique solution $v_i(\alpha), \quad i = 1, \ldots, U - 1$.

The discretized shape optimization problem may now be written as:

\[
\begin{align*}
\text{minimize} & \quad \hat{J}^h(\alpha), \\
\text{subject to} & \quad \alpha \in \mathcal{A}_{ad}, \\
\end{align*}
\]

where

\[
\hat{J}^h(\alpha) = J^h(v_L(\alpha), \ldots, v_{U-1}(\alpha), x_L^h(\alpha), \ldots, x_{U-1}^h(\alpha), \alpha),
\]

\[
= \sum_{i=L}^{U-1} [\Psi(u_i(\alpha), x_i^h(\alpha))].
\] (8.29)

Note that in (8.28) the velocities and pressure, $u^h(\alpha)$ and $p^h(\alpha)$, are implicit functions of the design parameter $\alpha$. These implicit functions are defined as the solution of (8.27). Using the implicit function theorem on (8.27), we can write the first-order sensitivity equations as:

\[
\frac{\partial c_i}{\partial v_i} \frac{dv_i}{d\alpha} = - \frac{\partial c_i}{\partial v_i} \frac{dv_i}{d\alpha} - \frac{\partial c_i}{\partial x_i^h} \frac{dx_i^h}{d\alpha}, \quad i = 1, \ldots, L, \ldots, U - 1.
\] (8.30)

Let $\Psi_i = \Psi(u_i(\alpha), x_i^h(\alpha))$, then the sensitivity $d\Psi_i/d\alpha$ in the space-time slab $i$ is:

\[
\frac{d\Psi_i}{d\alpha} = \frac{\partial \Psi_i}{\partial x_i^h} \frac{dx_i^h}{d\alpha} + \frac{\partial \Psi_i}{\partial v_i} \frac{dv_i}{d\alpha}, \quad i = 1, \ldots, L, \ldots, U - 1
\] (8.31)

The gradient $\nabla \hat{J}^h$ with respect to $\alpha$ is then given as:

\[
\nabla \hat{J}^h = \frac{\partial J^h}{\partial \alpha} + \sum_{i=L}^{U-1} \left[ \frac{d\Psi_i}{d\alpha} \right].
\] (8.32)
The recipe for \( \frac{\partial e_k}{\partial x_i^h} \) in (8.30) is shown in Appendix D.

Let \( z_i \) represent the adjoint variables \( z^h = (\lambda^h, \vartheta^h) \) in the slab \( Q_i \). To evaluate the gradient using the discretize-then-optimize adjoint approach, we write the discrete Lagrangian as:

\[
L(v_1, \ldots, v_{M-1}, z_1, \ldots, z_{U-1}, \alpha) = J^h(v_L, \ldots, v_U, x_{L}^h(\alpha), \ldots, x_{U-1}^h(\alpha), \alpha) + \sum_{i=1}^{U-1} z_i^T c_i(v_i, u_i^-, x_i^h(\alpha)) \nonumber \\
= \sum_{i=L}^{U-1} [\Psi(u_i, x_i^h(\alpha))] \nonumber \\
+ \sum_{i=1}^{U-1} z_i^T c_i(v_i, u_i^-, x_i^h(\alpha)). \quad (8.33)
\]

Notice that in (8.13), we have not written the dependence of the state \( v_i \) on \( \alpha \) explicitly. This is because we are explicitly accounting for the state equation. We only write \( v_i(\alpha) \) when using the modified objective function (8.28). The adjoint equation is then obtained by setting \( \frac{\partial L}{\partial v_i} = 0 \), for \( i = 1, \ldots, U - 1 \). This yields:

\[
\left( \frac{\partial c_{U-1}(v_{U-1}, u_{U-1}^-, x_{U-1}^h(\alpha))}{\partial v_{U-1}} \right)^T z_{U-1} = - \left( \frac{\partial \Psi_{U-1}}{\partial v_{U-1}} \right)^T, \quad (8.34)
\]

\[
\left( \frac{\partial c_i(v_i, u_i^-, x_i^h(\alpha))}{\partial v_i} \right)^T z_i = - \left( \frac{\partial c_{i+1}(v_{i+1}, u_{i+1}^-, x_{i+1}^h(\alpha))}{\partial v_i} \right)^T - \chi_i \left( \frac{\partial \Psi_i}{\partial v_i} \right)^T, \quad i = U - 2, U - 3, \ldots, 1,
\]

where \( \chi_i \) is defined as

\[
\chi_i = \begin{cases} 
0 & \text{for } 0 \leq i < L, \\
1 & \text{for } L \leq i < U - 1.
\end{cases} \quad (8.35)
\]
The gradient is then obtained as:

$$
\frac{\partial L}{\partial \alpha} = \frac{\partial J^h}{\partial \alpha} + \sum_{i=L}^{U-1} \frac{\partial \Psi_i}{\partial x_i^h} \frac{dx_i^h}{d\alpha} + \sum_{i=1}^{U-1} z_i \frac{\partial c_i(v_i, u_i^h, x_i^h(\alpha))}{\partial x_i^h} \frac{dx_i^h}{d\alpha}.
$$

(8.36)

The algorithm for the solution of the optimization problem is analogous to the one outlined in Section 8.3.

### 8.4.2 Discretization of the optimality conditions

The optimality conditions again consist of the state equation (2.1) and the equations (8.4). In the optimize-then-discretize approach, we discretize the infinite dimensional adjoint PDE given by (8.4a). As mentioned previously, we can choose different discretization schemes for the state and the adjoint equations; here we will present a formulation when using the same discretization scheme for the adjoint equation and the state equation. Our discretization of the state equation is given by (3.6). The GLS-stabilized formulation for the adjoint equation can be written as: given $u^h$, $p^h$ and $(\lambda^h)_{0}^{i+1}$, find $\lambda^h \in (V^h_i)_i$ and $\theta^h \in H^{1h}(Q_i(\alpha))$, such that $\forall w^h \in (V^h_i)_i$ and $\forall q^h \in H^{1h}(Q_i(\alpha))$:

$$
\int_{Q_i(\alpha)} w^h \cdot \rho \left( -\frac{\partial \lambda^h}{\partial t} + (\nabla u^h)^T \lambda^h - u^h \cdot \nabla \lambda^h \right) dQ + \int_{Q_i(\alpha)} \varepsilon(w^h) : \sigma(\lambda^h, \theta^h) dQ \\
+ \sum_{e=1}^{(n_e)_i} \int_{Q_e(\alpha)} \rho \left[ \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right] \\
\left[ \rho \left( -\frac{\partial u^h}{\partial t} + (\nabla u^h)^T \lambda^h - u^h \cdot \nabla \lambda^h \right) - \nabla \cdot \sigma(\lambda^h, \theta^h) \right] dQ \\
+ \int_{Q_i(\alpha)} q^h \nabla \cdot u^h dQ + \int_{Q_{i+1}(\alpha)} (w^h)_{i+1}^+ \cdot \rho ( (\lambda^h)_{i+1}^+ - (\lambda^h)_{i+1}^- ) d\mathbf{x} \\
= -\chi(t) \langle (D_u \Psi(u^h), w^h) \rangle_s,
$$

(8.37)
where \( \langle \langle D_u \Psi(u^h), w^h \rangle \rangle_s \), the stabilized functional, is given by:

\[
\langle \langle D_u \Psi(u^h), w^h \rangle \rangle_s = 4 \int_{Q_i^a(\alpha)} \varepsilon(u^h) : \varepsilon(w^h) dQ
+ 4 \sum_{e=1}^{(n_a)i} \int_{Q_i^e(\alpha) \cap Q_i^a(\alpha)} \tau_e \varepsilon(u^h) : \\
\varepsilon \left( \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) - \nabla \cdot \sigma(w^h, q^h) \right) dQ.
\]  

(8.38)

Let \( \lambda_i \) represent the adjoint velocity field in the space-time slab \( i \). The adjoint equation is solved sequentially for space-time slabs \( i = U - 1, U - 2, \ldots, 1 \) with

\[ \lambda_U^+ = 0. \]  

(8.39)

The gradient can then be represented as:

\[
\frac{\partial L}{\partial \alpha} = \frac{\partial J^h}{\partial \alpha} + \sum_{i=L}^{U-1} \frac{\partial \Psi_i}{\partial x_i} dx_i^h + \sum_{i=1}^{U-1} \frac{\partial}{\partial \alpha} \\
\int_{Q_i(\alpha)} \left[ \lambda_i \cdot \left( \rho \left( \frac{\partial u_i}{\partial t} + u_i \cdot \nabla u_i \right) - \nabla \cdot \sigma(u_i, p_i) + \theta_i \nabla \cdot u_i \right) \right] dQ.
\]

(8.40)

8.4.3 Difference in gradients between the discretize-then-optimize (do) and optimize-then-discretize (od) approaches

Using integration by parts on (8.37), and dropping the second derivatives in the stabilization terms due to our choice of piecewise-linear elements, the implemented GLS space-time stabilized formulation for the od adjoint equation (8.4a) can be written as follows: given \( u^h, p^h \) and \( (\lambda^h)_{i+1}^+ \), find \( \lambda^h \in (Y_u^h)_i \) and \( \theta^h \in H^1(Q_i(\alpha)) \),
such that $\forall w^h \in (V^h_u)_i$ and $\forall q^h \in H^1_i(Q_i(\alpha))$:

$$
\int_{Q_i(\alpha)} \lambda^h \cdot \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) dQ + \int_{Q_i(\alpha)} \varepsilon(w^h) : \sigma(\lambda^h, \theta^h) dQ \\
+ \sum_{e=1}^{(n_{e})_i} \int_{Q^*_i(\alpha) \cap Q^*_e(\alpha)} \tau_e \rho \left[ \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) + \nabla q^h \right] \cdot \\
\left[ \rho \left( -\frac{\partial u^h}{\partial t} + (\nabla u^h)^T \lambda^h - u^h \cdot \nabla \lambda^h \right) + \nabla \theta^h \right] dQ \\
+ \int_{Q_i(\alpha)} g^h \nabla \cdot u^h dQ + \sum_{e=1}^{(n_{e})_i} \int_{\Omega_{i+1}(\alpha)} (w^h)^+_{i+1} \cdot \rho \left( ((\lambda^h)^+_{i+1} - (\lambda^h)^-_{i+1}) \right) dx \\
= -\chi(t) \langle \langle D_u \Psi(u^h), w^h \rangle \rangle_s, \tag{8.41}
$$

where $\langle \langle D_u \Psi(u^h), w^h \rangle \rangle_s$ is implemented as:

$$
\langle \langle D_u \Psi(u^h), w^h \rangle \rangle_s = 4 \int_{Q^*_i(\alpha)} \varepsilon(u^h) : \varepsilon(w^h) dQ \\
+ 4 \sum_{e=1}^{(n_{e})_i} \int_{Q^*_i(\alpha) \cap Q^*_e(\alpha)} \rho \varepsilon(u^h) : \\
\varepsilon \left( \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) \right) dQ. \tag{8.42}
$$

The do adjoint, as seen in (8.34), can be written as:

$$
\int_{Q_i(\alpha)} \lambda^h \cdot \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) dQ + \int_{Q_i(\alpha)} \varepsilon(\lambda^h) : \sigma(\lambda^h, q^h) dQ \\
+ \sum_{e=1}^{(n_{e})_i} \int_{Q^*_i(\alpha) \cap Q^*_e(\alpha)} \tau_e \rho \left[ \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) + \nabla q^h \right] \cdot \\
\left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla \lambda^h \right) + \nabla \theta^h \right] dQ \\
+ \int_{Q_i(\alpha)} g^h \nabla \cdot u^h dQ + \sum_{e=1}^{(n_{e})_i} \int_{\Omega_{i+1}(\alpha)} (w^h)^+_{i+1} \cdot \rho \left( ((\lambda^h)^+_{i+1} - (\lambda^h)^-_{i+1}) \right) dx \\
= -4\chi(t) \int_{Q^*_i(\alpha)} \varepsilon(u^h) : \varepsilon(w^h) dQ. \tag{8.43}
$$

The differences between the od adjoint (8.41) and the do adjoint (8.43) are:

1. The term $(\nabla u^h)^T w^h$ and $(\nabla u^h)^T \lambda^h$ in the stabilization terms of od formulation.
2. The stabilization terms on the right-hand side in the \(od\) formulation, i.e.,
\[
\sum_{e=1}^{(ne)} \int_{Q_e^{(a)}} Q_e^{(a)} \frac{\tau_e}{\rho} \varepsilon(u^h) : \varepsilon \left( \rho \left( -\frac{\partial w^h}{\partial t} + (\nabla u^h)^T w^h - u^h \cdot \nabla w^h \right) \right) dQ.
\]

The gradient calculated using the \(od\) approach is given by (8.40). The gradient calculated using the \(do\) approach is given by:
\[
\frac{\partial L}{\partial \alpha} = \frac{\partial J}{\partial \alpha} + \sum_{i=L}^{U-1} \frac{\partial \Psi_i}{\partial x_i^h} \frac{dx_i^h}{d\alpha} + \sum_{i=1}^{U-1} \frac{\partial}{\partial \alpha} \int_{Q_i^{(a)}} \left[ \lambda_i \cdot \left( \rho \left( \frac{\partial u_i}{\partial t} + u_i \cdot \nabla u_i \right) - \nabla \cdot \sigma(u_i, p_i) \right) + \theta_i \nabla \cdot u_i \right] dQ
\]
\[
+ \sum_{i=1}^{U-1} \frac{\partial}{\partial \alpha} \left[ \int_{Q_i^{(a)}} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{\partial \lambda_i}{\partial t} + u_i \cdot \nabla \lambda_i \right) + \nabla \theta_i \right] \right] dQ
\]
\[
+ \int_{Q_i^{(a)}} \theta_i \nabla \cdot u_i dQ + \int_{\Omega_i^{(a)}} (\lambda_i^+ - (u_i^+ - \rho ((u_i^+) - (u_i^-)) dQ. \tag{8.44}
\]

On comparing the gradient equations (8.40) and (8.44), we see that, when using the \(do\) approach, the gradient has an added shape-sensitivity terms due to the stabilization in the discretized Lagrangian, and the jump term \(\int_{\Omega_i^{(a)}} (w_i^+ - (w_i^-)) d\mathbf{x}\). These terms are absent when using the \(od\) approach.

8.5 Reducing storage requirements in adjoint-based gradient calculations

The adjoint equations—(8.14) and (8.34) for the \textit{discretize-then-optimize} approach, and (8.17) and (8.37) for the \textit{optimize-then-discretize} approach—require the state variables \(\{v_i\}^{i=1}_{i=1}\). This is a consequence of the nonlinearity of the state equation and the choice of the objective function. For large-scale three-dimensional problems, storing the entire state history \(\{v_i\}^{i=1}_{i=1}\) might be impossible. Strategies are needed
to reduce the storage requirements, potentially at a cost of added computation. A simple strategy given in [82] consists of factoring the $U$ time steps as:

$$U = RS,$$  \hspace{1cm} (8.45)

where $R$ and $S$ are positive integers, resulting in a partition of the intervals $(0, U\Delta t)$ into $P$ slices, each consisting of $Q$ time steps. We introduce the following sets of state variable: $R + 1$ samples $\mathcal{H} = \{v_{i,S}\}_{i=0}^{R}$ and $S - 1$ states $\mathcal{G} = \{v_{i}\}_{i=1}^{S-1}$. One starts by solving the state equations for all the time steps and storing the sets $\mathcal{H}$ and $\mathcal{G}$. With these two sets available, one solves the adjoint equation backwards in time. When the data in $\mathcal{G}$ is exhausted, one solves the state equation for the next-to-last slice by using the data of the appropriate stored state in $\mathcal{H}$ as the initial condition. This is repeated until we solve the adjoint equation for all the time-steps. The maximum number of states required to be stored are $R + S$. For example, if $U = 100$, $R = 10$ and $S = 10$, then the maximum storage required is 20 steps.

### 8.6 Arterial grafting

As an application of optimization techniques introduced in this chapter, we consider the arterial grafting problem discussed in Section 7.4. The boundary conditions for the modeled flow field are: specified pulsatile inlet velocity, no-slip boundary conditions at all solid walls including the graft, and a parallel flow condition at the outlet. Let $H = 0.8$ be the height at the inlet. Recall that Figure 7.7 shows the geometry of the problem, with the flow proceeding from left to right. The inflow velocity profile is given as:

$$u_x = q(t)V, \quad u_y = 0.$$  \hspace{1cm} (8.46)
The term $V$ represents the parabolic spatial dependence of the velocity profile at the inlet with maximum velocity $V_{max}$. The temporal flow modulation at the inlet is given by:

$$q(t) = \begin{cases} 
0.1t & \text{for } 0 \leq t < 5, \\
0.5 & \text{for } 5 \leq t < 8, \\
0.5 + 0.1 \sin[\omega(t - 8)] & \text{for } 8 \leq t < \infty, 
\end{cases}$$

(8.47)

with $\omega = 2\pi$. We define Reynolds number as $Re = \rho V_{max} H / \mu_\infty$, and set $V_{max}$ so that $Re = 300$. We define the Womersley number as $Wo = \frac{H}{2} \sqrt{\frac{\omega \rho}{\mu_\infty}}$, which is 5.51 for the flow in (8.47). The Womersley number defines the ratio of inertial forces to viscous forces; a study on Womersley number effects in blood flow in arteries can be found in [83,84]. Our numerical results make use of the the semi-discrete finite element formulation using the sensitivity-based approach to compute gradients. In all computations, the viscosity for the Newtonian case is chosen to be $\mu = \mu_\infty$. The observation region $\Omega_{obs}$ for both the test cases is the entire flow domain, and $t_L = 15$ and $t_U = 20$.

### 8.6.1 Case 1

Similarly to Chapter 7, the first case involves $d = 0.6$ resulting in an aspect ratio of 0.75. The optimization process leads to a 10.98% reduction in the dissipation function using the Newtonian constitutive equation, with a gradient reduction of $\|\nabla \tilde{J}(\alpha_0)\| \times 10^{-3}$, and a 9.30% reduction in the dissipation function using the Carreau-Yasuda model, with a gradient reduction of $\|\nabla \tilde{J}(\alpha_0)\| \times 10^{-3}$. Figures 8.1 and 8.2 show the variation of $\Phi(u^h)$ as a function of time for the initial and optimal shapes, using the Newtonian and Carreau-Yasuda constitutive equations, respec-
tively. Similarly to our observations in the steady flow, Figure 8.3 illustrates that the optimal shapes obtained by the different constitutive equation choices do not differ significantly in this case. Table 8.1 shows the computed optimal shape parameters defined in (7.11), and Table 8.2 shows the values of the angles of the graft (from horizontal) at the inlet and exit of the artery. Figures 8.4 and 8.5 show the streamline plots for the initial and optimal shapes, respectively, using the Newtonian constitutive equation within one period of the pulsatile phase of the flow. Figures 8.6 and 8.7 show the streamline plots for the initial and optimal shapes, respectively, using the Carreau-Yasuda constitutive equation within one period of the pulsatile phase of the flow.

<table>
<thead>
<tr>
<th>Design Variable</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_2$</td>
<td>0.0</td>
<td>1.025</td>
<td>0.7142</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.0</td>
<td>-0.518</td>
<td>-0.2720</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0.0</td>
<td>0.1460</td>
<td>0.0706</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0.0</td>
<td>-0.0136</td>
<td>-0.0057</td>
</tr>
</tbody>
</table>

Table 8.1 : The design variables for the graft optimization for case 1.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 0$</td>
<td>90.00°</td>
<td>64.21°</td>
<td>66.18°</td>
</tr>
<tr>
<td>$\theta = \pi$</td>
<td>90.00°</td>
<td>65.05°</td>
<td>65.80°</td>
</tr>
</tbody>
</table>

Table 8.2 : The angle that the graft makes with the artery at its inlet and exit, for case 1.
Figure 8.1: The function $\Phi(u^h)$ as a function of time for the initial shape and the optimal shape, using the Newtonian constitutive equation for case 1.

Figure 8.2: The function $\Phi(u^h)$ as a function of time for the initial shape and the optimal shape, using the Carreau-Yasuda constitutive equation for case 1.
Figure 8.3: Initial shape and optimal shape of the graft design curve for the Newtonian model for case 1.

Figure 8.4: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the initial shape using Newtonian constitutive equations, for case 1.
Figure 8.5: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the optimal shape using Newtonian constitutive equations, for case 1.

Figure 8.6: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the initial shape using Carreau-Yasuda constitutive equations, for case 1.
Figure 8.7: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the optimal shape using Carreau-Yasuda constitutive equations, for case 1.
8.6.2 Case 2

Similarly to Chapter 7, the second case involves $d = 1.0$, resulting in an aspect ratio of 1.25. The optimization process leads to a 16.57% reduction in the dissipation function using the Newtonian constitutive equation, with a gradient reduction of $\|\nabla \tilde{J}(\alpha_0)\| \times 10^{-7}$, and a 9.62% reduction in the dissipation function using the the Carreau-Yasuda model, with a gradient reduction of $\|\nabla \tilde{J}(\alpha_0)\| \times 10^{-4}$. Figures 8.8 and 8.9 show the variation of $\Phi(u^h)$ as a function of time for the initial and optimal shapes using the Newtonian and Carreau-Yasuda constitutive equations, respectively. Table 8.3 shows the computed optimal shape parameters defined in (7.11), and Table 8.4 shows the values of the angles of the graft, (from horizontal) at the inlet and exit of the artery. Figure 8.10 illustrates that the optimal shapes obtained using the Newtonian and Carreau-Yasuda constitutive equation choice do show a significant difference. Figures 8.11 and 8.12 show the streamline plots for the initial and optimal shapes, respectively, using the Newtonian constitutive equation within one period of the pulsatile phase of the flow. Figures 8.13 and 8.14 show the streamline plots for the initial and optimal shapes, respectively, using the Carreau-Yasuda constitutive equation within one period of the pulsatile phase of the flow.

<table>
<thead>
<tr>
<th>Design Variable</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_2$</td>
<td>0.0</td>
<td>1.4809</td>
<td>1.3285</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.0</td>
<td>-0.7403</td>
<td>-0.7090</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0.0</td>
<td>0.1714</td>
<td>0.1928</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0.0</td>
<td>-0.0108</td>
<td>-0.0177</td>
</tr>
</tbody>
</table>

Table 8.3: The design variables for the graft optimization for case 2.
<table>
<thead>
<tr>
<th>Angle</th>
<th>Initial</th>
<th>Newtonian</th>
<th>Carreau-Yasuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 0$</td>
<td>90.00°</td>
<td>57.27°</td>
<td>60.18°</td>
</tr>
<tr>
<td>$\theta = \pi$</td>
<td>90.00°</td>
<td>54.70°</td>
<td>63.84°</td>
</tr>
</tbody>
</table>

Table 8.4: The angle that the graft makes with the artery at its inlet and exit, for case 2.

Figure 8.8: The function $\Phi(u^h)$ as a function of time for the initial shape and the optimal shape, using the Newtonian constitutive equation for case 2.
Figure 8.9: The function $\Phi(u^h)$ as a function of time for the initial shape and the optimal shape, using the Carreau-Yasuda constitutive equation for case 2.

Figure 8.10: Initial shape and optimal shape of the graft design curve for the Newtonian model for case 2.
Figure 8.11: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the initial shape using Newtonian constitutive equations, for case 2.

Figure 8.12: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the optimal shape using Newtonian constitutive equations, for case 2.
Figure 8.13: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the initial shape using Carreau-Yasuda constitutive equations, for case 2.

Figure 8.14: Streamline profiles at $t = 15.0, 15.25, 15.50, 15.75$ and $16.0$, (left to right, then top to bottom) for the optimal shape using Carreau-Yasuda constitutive equations, for case 2.
8.7 Conclusions

We have outlined the solution of shape optimization problems governed by the unsteady Navier-Stokes equations, as well as a fluid governed by a generalized Newtonian constitutive model. We have discussed the cases in which the unsteady state equations are discretized using the semi-discrete and space-time finite element formulations. The resulting optimization problem is solved using a gradient-based method. We pointed out the differences between the optimize-then-discretize and discretize-then-optimize approaches to the adjoint-based gradient evaluation for both the semi-discrete and space-time finite element formulations.

We considered an unsteady flow shape optimization problem for an idealized arterial graft geometry, with blood as the fluid. Our numerical results show the benefits of numerical shape optimization in achieving design improvements. We see that, similarly to our results for the steady case, the thickness of the graft influenced the differences in the optimal shape obtained with respect to the constitutive equation choice. Therefore, just like for the steady case, we conclude that when considering shape optimization problems involving blood, the shear rate in the domain affects the differences in optimal shapes obtained by the Newtonian and generalized Newtonian constitutive equations.
Chapter 9

Conclusions and Future Directions

We conclude the thesis with a summary of the material presented, in the form of a listing of what we believe are important contributions to the field of optimization in the context of complex fluid flows. We also point out future research directions that build upon the accomplishments of this thesis.

9.1 Conclusions

The goal of this thesis was to gain more insight into the issues involved in the solution of optimal control problems arising in incompressible fluid flows, when these problems are discretized using stabilized finite element methods. A specific emphasis has been on biomedical applications involving blood as the model fluid. In particular, this thesis has four main contributions.

Firstly, we presented a stress-recovery algorithm that improves the consistency of the GLS-stabilized finite element formulation, when using low-order elements. The reconstruction of the higher-order terms results in a much better consistency, and thus, accuracy, for relatively coarse meshes. The proposed method uses an approach that provides superconvergent fluxes at Dirichlet boundaries, in combination with standard variational reconstruction methods in the interior, to yield a low-order, yet strongly-consistent discretization. Our numerical results, based on the steady Stokes equations, showed a significant reduction in the error for relatively coarse practical
meshes, when compared to standard methods. Since the higher-order terms are also present in the Navier-Stokes equations, our method will be equally applicable in that context.

Secondly, we studied the effect of the order in which the GLS-stabilized finite element discretization is applied to an optimal control problem. For a class of optimal control problems governed by the linear Oseen equations, using suction- and blowing-type controls, we saw that the \textit{optimize-then-discretize} and \textit{discretize-then-optimize} approaches give rise to different discrete linear systems, due to the GLS stabilization. Our numerical results showed that the choice of the element length formula in the computation of the stabilization parameter, entering these different linear systems, bears a strong correlation to the quality of the computed control. We showed that, depending on the choice of the element length, the computed control may be very sensitive to the choice of the weighting parameter used in the stabilization term, and it may exhibit large spurious oscillations. We explored the reasons for this behavior, and proposed diagnostic tools that may help to assess the quality of the computed control, and guide the choice of stabilization parameters. Since our model problem closely resembles optimal control problems governed by the Navier-Stokes equations, we believe that our results are also relevant to optimal control of Navier-Stokes flow.

Thirdly, we investigated the influence of the fluid constitutive model on the outcome of shape optimization tasks for stationary flows, with blood as the model fluid. Our computations were based on the Navier-Stokes equations generalized to non-Newtonian fluid, with the Carreau-Yasuda model employed to account for the shear-thinning behavior of blood. Using a gradient-based optimization procedure, we numerically analyzed the influence of non-Newtonian effects on the computed optimal shape for two selected idealized biomedical systems. We saw that the differ-
ences in the computed optimal shape due to the choice of the constitutive equation is dependent on the shear rate prevalent in the flow. For example, in the arterial graft test case, the differences were more prominent when a geometric parameter amplified the shear-related viscosity variations in the generalized Newtonian constitutive equations. In contrast, when the overall shear rates were high, the Newtonian assumption seemed to match closely with the generalized Newtonian one.

Finally, we addressed some of the issues that arise in the solution of optimal design problems involving unsteady flows. In particular, we presented the issues that arise when using the GLS-stabilized finite element method, and pointed out the differences in the discrete system for the optimize-then-discretize and discretize-then-optimize approaches. Our formulations based on the space-time finite element method will later enable us to solve optimal design problems involving a time-dependent spatial domain. We extended our study on the influence of constitutive equation choice to unsteady flows, again, involving blood. We observed that, for the test case we considered, our conclusions were similar to the ones reached in the case of steady flows.

In addition to these contributions, our post-optimality sensitivity analysis with respect to simulation parameters promises to be a useful tool in the realm of PDE-constrained optimization. For non-linear systems, solving the system to compute such sensitivity may not be trivial. However, with a suitable approximation, the designer might find it very useful to know the sensitivity of the optimal solution to certain scalar parameters.

### 9.2 Future directions

Some of the directions for future research, motivated by the results in this thesis are:
- Investigation of the influence of the stress-recovery algorithm in the context of generalized Newtonian fluids, where $T^h$ depends on both $\mu(u^h)$ and $\varepsilon(u^h)$.

- Numerical investigation of the differences in optimize-then-discretize and discretize-then-optimize approaches to solve unsteady optimal design problems.

- The solution of optimal design problems involving unsteady viscoelastic 3D pulsatile flows in complex geometries. Such a tool would allow, e.g., study the distal placement of the graft and the effect of existing partial stenosis, which allows for prograde or retrograde flows between the ends of the graft, on its placement. The levels of existing stenosis are patient specific, and geometric sensitivity analysis of the effect of stenosis on the placement of the graft could be a useful clinical tool for the clinician in search of a robust surgical procedure.

- Optimal design involving moving boundaries such as the shape optimization of a left-ventricular assist device. This will also involve the incorporation of more relevant objective functions quantifying hemolysis or thrombosis.

- The incorporation of convergence acceleration and cost-reduction methods, such as reduced-basis methods and time- and space-domain-decomposition methods, to solve these very-large-scale optimal design problems.
Appendix A

Implementation of the Semi-Discrete Formulation

In this appendix, we present the element-wise evaluation of the residual of the semi-discrete finite element formulation. We recall the GLS-stabilized semi-discrete finite element discretization (3.2), which is implemented as (3.3), when using a low-order function space such as piecewise-linear elements (considered here); and define the following finite element space:

\[ w = \sum_{A=1}^{n_n} N_A w_A, \quad (A-1) \]

\[ q = \sum_{A=1}^{n_n} N_A q_A, \quad (A-2) \]

\[ u = \sum_{A=1}^{n_n} N_A u_A, \quad (A-3) \]

\[ p = \sum_{A=1}^{n_n} N_A p_A, \quad (A-4) \]

where \( n_n \) represents the number of nodes in the spatial discretization. Let \( \Delta u \) represent the increment of velocity at the time step at which we are forming the system. Furthermore, for a given node \( A \), let \( c^i_A \) represent the residual corresponding to the \( i \)th velocity component of \( c_u \) and \( c^p_A \) be the pressure component of the residual \( c_p \) (refer to Sections 3.2 and 3.5). These terms are given as:
\[ c^i_A = \int_\Omega N_A e^i \cdot \rho \left( \frac{N_B \Delta u_B^j}{\Delta t} + u^h \cdot \nabla N_B u_B^j \right) e^j \, dx - \int_\Omega (\nabla N_A \cdot e^i) N_{BP} \, dx + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_e \left[ \rho \left( \frac{N_A}{\Delta t} + u^h \cdot \nabla N_A \right) e^j \right] \cdot \\
 \left[ \rho \left( \frac{N_B \Delta u_B^j}{\Delta t} + u^h \cdot \nabla N_B u_B^j \right) e^j + \nabla N_{BP} \right] \, dx + \int_\Omega \mu \left[ \delta_{ij} \nabla N_A \cdot \nabla N_B + \left( \nabla N_A \cdot e^j \right) \left( \nabla N_B \cdot e^i \right) \right] u_B^j \, dx - \int_{\Gamma_h} N_A e^i \cdot N_B e^j h_B^j \, dx, \quad (A-5) \]

and:

\[ c^p_A = \int_\Omega N_A (e^j \cdot \nabla N_B) p_B \, dx + \int_\Omega \tau_e \left( \nabla N_A e^i \right) \cdot \\
\left[ \rho \left( \frac{N_B \Delta u_B^j}{\Delta t} + u^h \cdot \nabla N_B u_B^j \right) e^j + \nabla N_{BP} \right] \, dx. \quad (A-6) \]

The terms \( c^i_A \) and \( c^p_A \) at node \( A \) are formed by the assembly of element-level contributions. Let \( c^i_A \) and \( c^p_A \) represent these element level terms, with \( a \) denoting nodes numbers at the element level, i.e., \( a = 1, \ldots, n_{en} \), where \( n_{en} \) is the number of nodes in an element. The transformation from the spatial coordinate \( x \) to the local curvilinear coordinate \( \xi \) is given by:

\[ x^e = N_a(\xi)x^e_a, \quad (A-7) \]

where \( x^e_a \) represents the spatial coordinates at node \( a \). The Jacobian matrix \( F \) and the determinant \( F \) of the transformation are:

\[ F = \left( \frac{\partial x}{\partial \xi} \right)^T \in \mathbb{R}^{n_{sd} \times n_{sd}}, \]

\[ F = \det(F). \quad (A-8) \]
The derivatives of the shape functions are then:

\[
\nabla N_a = F^{-1} \nabla_{\xi} N_a. \tag{A-9}
\]

Further, let \( \Omega^\xi \) and \( \Gamma^\xi \) represent the domain and the boundary, respectively, in reference coordinates. The element level contributions in reference coordinates can then be written as:

\[
c^i_a = \int_{\Omega^\xi} N_a e^i \cdot \rho \left( \frac{N_b \Delta u_b^j}{\Delta t} + u^b \cdot F^{-1} \nabla_{\xi} N_b u_b^j \right) e^j F \, d\xi
\]

\[
- \int_{\Omega^\xi} \left( F^{-1} \nabla_{\xi} N_a \cdot e^i \right) N_b p_b F \, d\xi
\]

\[
+ \int_{\Omega^\xi} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{N_a}{\Delta t} + u^b \cdot F^{-1} \nabla_{\xi} N_a \right) e^i \right] \cdot
\left[ \rho \left( \frac{N_b \Delta u_b^j}{\Delta t} + u^b \cdot F^{-1} \nabla_{\xi} N_b u_b^j \right) e^j + F^{-1} \nabla_{\xi} N_b p_b \right] F \, d\xi
\]

\[
+ \int_{\Omega^\xi} \mu \left[ \delta_{ij} F^{-1} \nabla_{\xi} N_a \cdot F^{-1} \nabla_{\xi} N_b + \left( F^{-1} \nabla_{\xi} N_a \cdot e^i \right) \left( F^{-1} \nabla_{\xi} N_b \cdot e^j \right) \right] u_b^j F \, d\xi
\]

\[
- \int_{\Gamma_h^\xi} \delta_{ij} N_a N_b u_b^i F \, d\xi, \tag{A-10}
\]

and

\[
c^i_a = \int_{\Omega^\xi} N_a \left( e^i \cdot F^{-1} \nabla_{\xi} N_b \right) p_b F \, d\xi
\]

\[
+ \int_{\Omega^\xi} \frac{\tau_e}{\rho} \left( F^{-1} \nabla_{\xi} N_a e^i \right) \cdot
\left[ \rho \left( \frac{N_b \Delta u_b^j}{\Delta t} + u^b \cdot F^{-1} \nabla_{\xi} N_b u_b^j \right) e^j + F^{-1} \nabla_{\xi} N_b p_b \right] F \, d\xi. \tag{A-11}
\]
Appendix B

Implementation of the Space-Time Formulation

In this appendix, we present the element-wise evaluation of the residual of the space-time finite element formulation. We recall the GLS-stabilized space-time finite element discretization (3.6), which is implemented as (3.9), when using low-order function space, such as piecewise-linear elements (considered here); and define the following finite element space:

\[
\begin{align*}
  w &= \sum_{A=1}^{n_n} N_A w_A, \\
  q &= \sum_{A=1}^{n_n} N_A q_A, \\
  u &= \sum_{A=1}^{n_n} N_i u_A, \\
  p &= \sum_{A=1}^{n_n} N_A p_A,
\end{align*}
\]  

(B-1) \hspace{1cm} (B-2) \hspace{1cm} (B-3) \hspace{1cm} (B-4)

where \( n_n \) represents the number of nodes in the space-time discretization. Let \( \Delta u \) represent the increment of velocity at the time step at which we are forming the system. Further, for a given node \( A \), let \( c^1_A \) represent the residual corresponding to the \( i \)th velocity component of \( c^h_u \) and \( c^p_A \) be the pressure component of the residual \( c^h_p \) (refer to Sections 3.3 and 3.5). Then, for the space-time slab at time step \( m \), we
obtain:

\[
\hat{c}_A^i = \int_{Q_m} N_A e^i \cdot \rho \left( N_{B,t} + u^h \cdot \nabla N_B \right) u_B^j e^j \, dQ
- \int_{Q_m} (\nabla N_A \cdot e^i) N_{BPB} \, dQ
+ \sum_{e=1}^{(n_e)_m} \int_{Q_m} \frac{\tau_e}{\rho} \left[ \rho \left( N_{A,t} + u^h \cdot \nabla N_A \right) e^i \right] \cdot \\
\left[ \rho \left( N_{B,t} + u^h \cdot \nabla N_B \right) u_B^j e^j + \nabla N_{BPB} \right] \, dQ
+ \int_{Q_m} \mu \left( \delta_{ij} \nabla N_A \cdot \nabla N_B + (\nabla N_A \cdot e^i)(\nabla N_B \cdot e^j) \right) u_B^j \, dQ
+ \int_{Q_m} N_A e^i \cdot \rho \left( (u^h)^+_m - (u^h)^-_m \right) \, dx
- \int_{(P_m)_h} \delta_{ij} N_A N_B h_B^i \, dP, \tag{B-5}
\]

and:

\[
c_A^p = \int_{Q_m} N_A (e^j \cdot \nabla N_B) u_B^j \, dQ
+ \int_{Q_m} \frac{\tau_e}{\rho} (\nabla N_A e^i) \cdot \left[ \rho (N_{B,t} + u^h \cdot \nabla N_B) u_B^j e^j + \nabla N_{BPB} \right] \, dQ. \tag{B-6}
\]

The terms \( c_A^i \) and \( c_A^p \) are formed by the assembly of element-level contributions. Let \( c_a^{ie} \) and \( c_a^{pe} \) represent these element-level terms, with \( a \) denoting nodes numbers at the element level, i.e., \( a = 1, \ldots, n_{en} \), where \( n_{en} \) represents the number of nodes in a space-time element.

The transformation from the spatial coordinate \((x, t)\) to the local curvilinear coordinate \((\xi, \theta)\) is given by:

\[
x^e = N_a(\xi, \theta) x_a^e, \tag{B-7}
\]

\[
t^e = N_a(\xi, \theta) t_a^e,
\]

where again \( n_{en} \) represents the number of nodes that form an element \( e \), and \( t_a^e = t_m \) for \( a = 1, \ldots, n_{en}/2 \) and \( t_a^e = t_m+1 \), for \( a = n_{en}/2 + 1, \ldots, n_{en} \). From this
definition, we obtain $\frac{\partial t}{\partial \xi} = 0$. In the special case when $\theta = \text{const}$ coincides with the $t = \text{const}$ levels, the Jacobian matrix and the determinant of the transformation between the reference and physical domains can be written as:

$$\frac{\partial (x, t)}{\partial (\xi, \theta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \theta} \\ 0 & \frac{\partial t}{\partial \theta} \end{bmatrix}, \quad (B-8)$$

and its inverse as:

$$\frac{\partial (\xi, \theta)}{\partial (x, t)} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial t} \\ 0 & \frac{\partial \theta}{\partial t} \end{bmatrix} = \begin{bmatrix} (\frac{\partial x}{\partial \xi})^{-1} & (\frac{\partial x}{\partial \xi})^{-1}(\frac{\partial x}{\partial \theta}) \frac{2}{\Delta t} \\ 0 & \frac{2}{\Delta t} \end{bmatrix}. \quad (B-9)$$

Noting that $F = \left(\frac{\partial x}{\partial \xi}\right)^T$, we obtain:

$$\begin{pmatrix} \nabla N_a \\ N_{a,t} \end{pmatrix} = \begin{bmatrix} F^{-1} & 0 \\ \frac{2}{\Delta t}(\frac{\partial x}{\partial \theta})^T F^{-1} & \frac{2}{\Delta t} \end{bmatrix} \begin{pmatrix} \nabla_\xi N_a \\ N_{a,\theta} \end{pmatrix}, \quad (B-10)$$

In the case of fixed mesh problems, $\frac{\partial x}{\partial \theta} = 0$, and for this particular case (B-10) transforms to:

$$\begin{pmatrix} \nabla N_a \\ N_{a,t} \end{pmatrix} = \begin{bmatrix} F^{-1} & 0 \\ 0 & \frac{2}{\Delta t} \end{bmatrix} \begin{pmatrix} \nabla_\xi N_a \\ N_{a,\theta} \end{pmatrix}. \quad (B-11)$$

We will denote the Jacobian matrix and the determinant of the transformation by:

$$K = \left(\frac{\partial (x, t)}{\partial (\xi, \theta)}\right)^T \in \mathbb{R}^{(n_{ad} + 1) \times (n_{ad} + 1)},$$

$$K' = \det(K), \quad (B-12)$$

respectively. Let $Q^\xi$ and $P^\xi$ represent the space-time element and the boundary, respectively, in reference coordinates. Additionally let $\Omega^\xi$ represent $\Omega^e_m$ in reference
coordinates. Further, let $K^+$ be the determinant of the Jacobian of the transformation of the spatial coordinates of $\Omega^e_m$, the bottom layer of the space-time slab (see Figure 3.1), to reference coordinates. The element-wise contributions of the equations (B-5) and (B-6) at time step $m$ can be written as:

$$
c^e_a = \int_{Q^e} N_a e^i \cdot \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla_\xi N_b \right) u_b^i e^j K dQ^\xi \\
- \int_{Q^e} \left( F^{-1} \nabla_\xi N_a \cdot e^i \right) N_{b_p} K dQ^\xi \\
+ \int_{Q^e} \frac{\tau_e}{\rho} \left[ \rho \left( N_{a,t} + u^h \cdot F^{-1} \nabla_\xi N_a \right) e^i \right] \cdot \\
\left[ \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla_\xi N_b \right) u_b^i e^j + F^{-1} \nabla_\xi N_{b_p} \right] K dQ^\xi \\
+ \int_{Q^e} \mu \left[ \delta_{ij} \left( F^{-1} \nabla_\xi N_a \cdot F^{-1} \nabla_\xi N_b \right) + \left( F^{-1} \nabla_\xi N_a \cdot e^j \right) \left( F^{-1} \nabla_\xi N_b e^i \right) \right] u_b^i K dQ^\xi \\
+ \int_{Q^e} \left( u^h \right)_m^+ - \left( u^h \right)_m^- \right) K^+ dQ^\xi \\
- \int_{P^e_h} \delta_{ij} N_a N_{b_p} h_{b_p}^i K dP^\xi, 
$$

(B-13)

and

$$
c^e_a = \int_{Q^e} N_a \left( e^j \cdot F^{-1} \nabla_\xi N_b \right) u_b^j K dQ^\xi \\
+ \int_{Q^e} \frac{\tau_e}{\rho} \left( F^{-1} \nabla_\xi N_a e^i \right) \cdot \\
\left[ \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla_\xi N_b \right) u_b^i e^j + F^{-1} \nabla_\xi N_{b_p} \right] K dQ^\xi. 
$$

(B-14)
Appendix C

Shape Sensitivity in the Semi-Discrete Formulation

In this appendix, we present the element-wise evaluation of the sensitivity of the residual of the semi-discrete finite element formulation, i.e., the term $\frac{\partial c_i}{\partial x^h} \frac{dx^h}{d\alpha}$, at a given time step $i$, seen in equations (8.10) and 8.16) of Chapter 8. A steady version of these equations was seen in Chapter 7, (7.8, 7.9). From (A-10) and (A-11), we see that we need to evaluate the terms $\frac{\partial F}{\partial \alpha}$ and $\frac{\partial F^{-1}}{\partial \alpha}$.

Using the techniques of [79–81], for a given element $e$, the derivative of the inverse of the transformation is given by:

$$\frac{\partial (F^{-1})_{ki}}{\partial \alpha} = \frac{\partial (F^{-1})_{ki}}{\partial x^a_j} \frac{dx^a_j}{d\alpha}$$

$$= -(F^{-1})_{ji} \frac{\partial N_a}{\partial x^a_k} \frac{dx^a_j}{d\alpha}, \quad (C-1)$$

at a given node $a$ within the element $e$, and the derivative of the determinant of the transformation is given by:

$$\frac{\partial F}{\partial \alpha} = \frac{\partial F}{\partial x^a_j} \frac{dx^a_j}{d\alpha}$$

$$= F \frac{\partial N_a}{\partial x^a_j} \frac{dx^a_j}{d\alpha}. \quad (C-2)$$

We use the same notation for the terms as in Appendix A. Using (C-1,C-2) in
(A-10) and (A-11), we obtain the sensitivity of the residual with $\alpha$ as:

$$
\frac{\partial e^i}{\partial \alpha} = \int_{\Omega} N_a e^i \cdot \rho \left( \frac{u^h \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla N_b u^i}{\Delta t} + u^h \cdot F^{-1} \nabla N_b u^i \right) e^j F d\xi
$$

$$
+ \int_{\Omega} N_a e^i \cdot \rho \left( \frac{N_b \Delta u^i}{\Delta t} + u^h \cdot F^{-1} \nabla N_b u^i \right) e^j \frac{\partial F}{\partial \alpha} d\xi
$$

$$
- \int_{\Omega} \left( \frac{\partial F^{-1}}{\partial \alpha} \nabla N_a \cdot e^i \right) N_b p_b F d\xi
$$

$$
- \int_{\Omega} \left( F^{-1} \nabla N_a \cdot e^i \right) N_b p_b \frac{\partial F}{\partial \alpha} d\xi
$$

$$
+ \int_{\Omega} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{N_a}{\Delta t} + u^h \cdot F^{-1} \nabla N_a \right) e^i \right] \cdot
\left[ \rho \left( u^h \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla N_b u^i \right) e^j + \frac{\partial F^{-1}}{\partial \alpha} \nabla N_b N_b p_b \right] F d\xi
$$

$$
+ \int_{\Omega} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{N_a}{\Delta t} + u^h \cdot F^{-1} \nabla N_a \right) e^i \right] \cdot
\left[ \rho \left( \frac{N_b \Delta u^i}{\Delta t} + u^h \cdot F^{-1} \nabla N_b u^i \right) e^j + F^{-1} \nabla N_b p_b \right] F d\xi
$$

$$
+ \int_{\Omega} \frac{\tau_e}{\rho} \left[ \rho \left( \frac{N_a}{\Delta t} + u^h \cdot F^{-1} \nabla N_a \right) e^i \right] \cdot
\left[ \rho \left( \frac{N_b \Delta u^i}{\Delta t} + u^h \cdot F^{-1} \nabla N_b u^i \right) e^j + F^{-1} \nabla N_b p_b \right] \frac{\partial F}{\partial \alpha} d\xi
$$

$$
+ \sum_{i=1}^{n_{el}} \int_{\Omega} \mu \delta_{ij} \left[ F^{-1} \nabla N_a \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla N_b + \frac{\partial F^{-1}}{\partial \alpha} \nabla N_a \cdot F^{-1} \nabla N_b \right] u^j b F d\xi
$$

$$
+ \sum_{i=1}^{n_{el}} \int_{\Omega} \mu \left[ \left( \frac{\partial F^{-1}}{\partial \alpha} \nabla N_a \cdot e^i \right) \left( F^{-1} \nabla N_b \cdot e^i \right) \right] u^j b F d\xi
$$

$$
+ \int_{\Omega} \mu \left[ \delta_{ij} \left( F^{-1} \nabla N_a \cdot F^{-1} \nabla N_b \right) + \left( F^{-1} \nabla N_a \cdot e^i \right) \left( F^{-1} \nabla N_b \cdot e^i \right) \right] u^j b \frac{\partial F}{\partial \alpha} d\xi
$$

$$
- \int_{\Omega} N_a N_b \frac{\partial F}{\partial \alpha} d\xi,
$$

(C-3)
\[
\frac{\partial c^\rho}{\partial \alpha} = \int_{\Omega^\kappa} N_a \left( e^j \cdot \frac{\partial \mathbf{F}^{-1}}{\partial \alpha} \nabla_{\xi} N_b \right) p_b F d\xi \\
+ \int_{\Omega^\kappa} N_a \left( e^j \cdot \mathbf{F}^{-1} \nabla_{\xi} N_b \right) p_b \frac{\partial F}{\partial \alpha} d\xi \\
+ \int_{\Omega^\kappa} \frac{\tau_n}{\rho} \left( \frac{\partial \mathbf{F}^{-1}}{\partial \alpha} \nabla_{\xi} N_a e^j \right) \cdot \\
\left[ \rho \left( \frac{N_b \Delta u^j_b}{\Delta t} + \mathbf{u}^h \cdot \mathbf{F}^{-1} \nabla_{\xi} N_b u^j_b \right) e^j + \mathbf{F}^{-1} \nabla_{\xi} N_b p_b \right] F d\xi \\
+ \int_{\Omega^\kappa} \frac{\tau_n}{\rho} \left( \mathbf{F}^{-1} \nabla_{\xi} N_a e^j \right) \cdot \\
\left[ \rho \left( \mathbf{u}^h \cdot \frac{\partial \mathbf{F}^{-1}}{\partial \alpha} \nabla_{\xi} N_b u^j_b \right) e^j + \frac{\partial \mathbf{F}^{-1}}{\partial \alpha} \nabla_{\xi} N_b p_b \right] F d\xi \\
+ \int_{\Omega^\kappa} \frac{\tau_n}{\rho} \left( \mathbf{F}^{-1} \nabla_{\xi} N_a e^j \right) \cdot \\
\left[ \rho \left( \frac{N_b \Delta u^j_b}{\Delta t} + \mathbf{u}^h \cdot \mathbf{F}^{-1} \nabla_{\xi} N_b u^j_b \right) e^j + \mathbf{F}^{-1} \nabla_{\xi} N_b p_b \right] \frac{\partial F}{\partial \alpha} d\xi. \quad (C-4)
\]
Appendix D

Shape Sensitivity in the Space-Time Formulation

In this appendix, we present the element-wise evaluation of the sensitivity of the residual of the space-time finite element formulation, i.e., the term \( \frac{\partial e_i}{\partial x_i^b} \frac{dx_i^b}{d\alpha} \), at a given time step \( i \), seen in equations (8.30) and 8.36 of Chapter 8. From (B-13) and (B-14), we see that we need to evaluate the terms \( \frac{\partial K}{\partial \alpha} \) and \( \frac{\partial F^{-1}}{\partial \alpha} \). Let \( \mathbf{x}^* = (\mathbf{x}, t)^T \in \mathbb{R}^{n_{\text{tot}} + 1} \), then

\[
\frac{d\mathbf{x}^*}{d\alpha} = \left( \frac{d\mathbf{x}}{d\alpha}, \frac{dt}{d\alpha} \right)^T.
\]

But \( \frac{dt}{d\alpha} = 0 \).

Using the techniques of [79–81], for a given element \( e \), the derivative of the inverse of the transformation is given by:

\[
\frac{\partial (K^{-1})_{ki}}{\partial \alpha} = \frac{\partial (K^{-1})_{ki}}{\partial x_j^*} \frac{dx_j^*}{d\alpha} = -(K^{-1})_{ji} \frac{\partial N_a}{\partial x_k^*} \frac{dx_j}{d\alpha}, \quad (D-2)
\]

at a given node \( a \) within the element \( e \), and the derivative of the determinant of the transformation is given by:

\[
\frac{\partial K}{\partial \alpha} = \frac{\partial K}{\partial x_j^{*b}} \frac{dx_j^{*b}}{d\alpha} = K \frac{\partial N_a}{\partial x_j^{*b}} \frac{dx_j^{*b}}{d\alpha}, \quad (D-3)
\]

From (B-10) and (D-2), we can obtain \( N_{a,t,\alpha} \).
We use the same notation for the terms as in Appendix B. Using (D-2-D-3) in (B-13) and (B-14), we obtain the sensitivity of the residual with $\alpha$ as:

\[
\frac{\partial e^{\text{te}}_a}{\partial \alpha} = \int_{\Omega^\xi} N_a e^i \cdot \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla_N N_b \right) e^i u^i_b \frac{\partial K}{\partial \alpha} dQ^\xi
\]

\[
+ \int_{\Omega^\xi} N_a e^i \cdot \rho \left( N_{b,t} + u^h \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla_N N_b \right) e^i u^i_b K dQ^\xi
\]

\[
- \int_{\Omega^\xi} \left( F^{-1} \nabla_N N_a \cdot e^i \right) N_b p_b \frac{\partial K}{\partial \alpha} dQ^\xi
\]

\[
- \int_{\Omega^\xi} \left( \frac{\partial F^{-1}}{\partial \alpha} \nabla_N N_a \cdot e^i \right) N_b p_b K dQ^\xi
\]

\[
+ \tau_e \int_{\Omega^\xi} \rho \left[ \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla_N N_b \right) e^i u^i_b + \frac{\partial F^{-1}}{\partial \alpha} \nabla_N N_b p_b \right] K dQ^\xi
\]

\[
+ \int_{\Omega^\xi} \frac{\tau_e}{\rho} \left[ \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla_N N_b \right) e^i u^i_b + \frac{\partial F^{-1}}{\partial \alpha} \nabla_N N_b p_b \right] K dQ^\xi
\]

\[
+ \int_{\Omega^\xi} \frac{\tau_e}{\rho} \left[ \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla_N N_b \right) e^i u^i_b + F^{-1} \nabla_N N_b p_b \frac{\partial K}{\partial \alpha} dQ^\xi\right]
\]

\[
+ \int_{\Omega^\xi} \mu \left[ \delta_{ij} \left( F^{-1} \nabla_N N_a \cdot F^{-1} \nabla_N N_b \right) + \left( F^{-1} \nabla_N N_a \cdot e^i \right) \left( F^{-1} \nabla_N N_b e^i \right) \right] u^i_b \frac{\partial K}{\partial \alpha} dQ^\xi
\]

\[
+ \int_{\Omega^\xi} \mu \left[ \delta_{ij} \left( \frac{\partial F^{-1}}{\partial \alpha} \nabla_N N_a \cdot F^{-1} \nabla_N N_b \right) + \left( F \nabla_N N_a \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla_N N_b \right) \right] u^i_b K dQ^\xi
\]

\[
+ \int_{\Omega^\xi} \mu \left[ \left( \frac{\partial F^{-1}}{\partial \alpha} \nabla_N N_a \cdot e^i \right) \left( F^{-1} \nabla_N N_b e^i \right) \right] u^i_b K dQ^\xi
\]

\[
+ \int_{\Omega^\xi} N_a e^{i+} \cdot \rho \left( (u^h)^+ - (u^h)^- \right) \frac{\partial K^+}{\partial \alpha} dQ^\xi
\]

\[
- \int_{P^h} \delta_{ij} N_a N_b h^i_b \frac{\partial K}{\partial \alpha} dP^\xi,
\]

(D-4)
\[ \frac{\partial c_b^\text{pe}}{\partial \alpha} = \int_{Q_k} N_a \left( e^j \cdot F^{-1} \nabla \xi N_b \right) w_b^j \frac{\partial K}{\partial \alpha} dQ_k \]
\[ + \int_{Q_k} N_a \left( e^j \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla \xi N_b \right) w_b^j K dQ_k \]
\[ + \int_{Q_k} \tau_e \rho \left[ (F^{-1} \nabla \xi N_a \cdot e^j) \rho \left( N_{b,t} + u^h \cdot F^{-1} \nabla \xi N_b \right) w_b^j \right] K dQ_k \]
\[ + \int_{Q_k} \frac{\tau_e}{\rho} \left[ \left( \frac{\partial F^{-1}}{\partial \alpha} \nabla \xi N_a \cdot e^j \right) \rho \left( N_{b,t} + u^h \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla \xi N_b \right) w_b^j \right] K dQ_k \]
\[ + \int_{Q_k} \frac{\tau_e}{\rho} \left[ \left( F^{-1} \nabla \xi N_a \cdot e^j \right) \rho \left( N_{b,t} \alpha + u^h \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla \xi N_b \right) w_b^j \right] K dQ_k \]
\[ + \int_{Q_k} \frac{\tau_e}{\rho} \left[ \left( F^{-1} \nabla \xi N_a \cdot e^j \right) \right. \]
\[ \left. \left( \frac{\partial F^{-1}}{\partial \alpha} \nabla \xi N_a \cdot F^{-1} \nabla \xi N_b + F^{-1} \nabla \xi N_a \cdot \frac{\partial F^{-1}}{\partial \alpha} \nabla \xi N_b \right) p_b \right] K dQ_k. \] (D-5)
Bibliography


   todynamics: formulations and error estimates”, Computer Methods in Applied

[16] P. Hansbo and A. Szepessy, “A velocity-pressure streamline diffusion finite ele-
   ment method for the incompressible Navier-Stokes equations”, Computer Meth-

[17] T.E. Tezduyar, M. Behr, S. Mittal, and J. Liou, “A new strategy for finite ele-
   ment computations involving moving boundaries and interfaces – the deforming-
   spatial-domain/space-time procedure: II. Computation of free-surface flows,
   two-liquid flows, and flows with drifting cylinders”, Computer Methods in Ap-

[18] P. Hansbo, “The characteristic streamline diffusion method for the time-
   dependent incompressible Navier-Stokes equations”, Computer Methods in Ap-

   past oscillating cylinders and airfoils”, International Journal for Numerical

[20] S. Mittal, Stabilized Space-Time Finite Element Formulations for Unsteady In-
   of Aerospace Engineering and Mechanics, University of Minnesota, 1992.


