National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Optimal Control of Time-Dependent Viscous Flow with Potential Application to Artificial Hearts

by

AKBAR SAHRAPOUR

A Thesis

Presented to the University of Ottawa

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

Ottawa-Carleton Institute for

Mechanical and Aerospace Engineering

December 1995

ISBN 0-612-11599-2

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

The present research is concerned with the application of optimal control theory to time-dependent viscous flows. Some specific designs of artificial hearts are described briefly, followed by a review on common hemodynamic problems created by artificial hearts and heart valves. Some mathematical models for certain types of artificial hearts are formulated. Optimal control theory has been applied to some of these models to obtain the necessary conditions of optimality. The optimization problem consists of determining the optimal control policy in order to minimize the value of an appropriate cost functional, which accounts for differences from a desired flow rate and the mean squared shear stresses and vorticity. The necessary conditions of optimality include the Navier-Stokes equations, the associated adjoint equations and an inequality.

For the purpose of the numerical solution of the governing equations, a numerical code has been developed, which solves the Navier-Stokes and the corresponding adjoint equations by the Finite Element Method with penalty function formulation. The Navier-Stokes and the adjoint equations were solved iteratively. Each system of equations was solved separately but the two systems were iteratively coupled in the control problem context. At the end of each iteration a new control was computed using the inequality, and the process was repeated until convergence was achieved. Numerical solution of the optimization algorithm was achieved for a two-dimensional model with an initial velocity distribution as the control variable, and a two-dimensional model with a time-dependent boundary velocity applied at some part of the boundary as the control. The latter case was further extended to an approximate three-dimensional model of an artificial heart. In all cases, it was observed that, in the absence of shear stress and vorticity in the cost functional, the desired flow rate was almost achieved, while including these parameters

resulted in a decreased optimal flow rate but with smoother velocity distributions. In the case of two- and three-dimensional boundary control, it was found that, when the stress and vorticity levels were minimized, the optimization algorithm could successfully rearrange the flow distribution, smoothen the sharp gradients and remove the vortices that would occur otherwise in the velocity field.

This study demonstrates that optimal control of flow systems is practical and can be used as a design tool.

Acknowledgements

I express my sincere gratitude to Professors N.U. Ahmed and S. Tavoularis for suggesting the topic of the thesis and continuous guidance, encouragement and support throughout this research. I am also thankful to Professor S. Tavoularis for teaching me the scientific way of writing and presenting the results.

I am indebted to my caring parents for all their great efforts in supporting and directing me throughout my life.

My special thanks are due to my wife Maryam for sacrificing her own interests and making it possible for me to carry on this research. I am also indebted to my son Mohammad-Mehdi and my daughter Fatemeh for their great patience and understanding.

The financial support received from the Ministry of Culture and Higher Education of the Islamic Republic of Iran and partly from the Natural Sciences and Engineering Research Council of Canada are gratefully acknowledged.

Contents

List of Tables

List of Figures

Nomenclature

Roman symbols

A_{ij}	integral coefficient in global matrix
A_i	inlet area of the domain
A_o	outlet area of the domain
C^i	functions with i continuous derivatives
C_i	integral coefficient in global matrix
d_{ij}	rate of deformation tensor
E_i	integral coefficient in global matrix
e_i	unit vectors in coordinate directions
F_i	integral coefficient in global matrix
$\mathbf{f}(f_i)$	body force per unit mass
$\mathbf{G}^1(G_i^1)$	Gateaux derivative of $J(\mathbf{v})$ with respect to $\mathbf{u}$
$\mathbf{G}^2(G_i^2)$	Gateaux derivative of $J(\mathbf{v})$ with respect to $\mathbf{v}$
$g(\mathbf{u},\mathbf{v})$	integrand of the cost functional
I	time interval
J	Jacobian matrix
$J(\mathbf{v})$	cost functional
$J'(\mathbf{v})$	Gateaux derivative of the cost functional $J(\mathbf{v})$
K_{ij}	integral coefficient in global matrix
M	integral coefficient in global matrix
m	mass of a system
N_i	integral coefficient in global matrix

$\mathbf{n}(n_i)$	outward normal vector
$P(P^m)$	nodal values of the pressure
p	pressure
p_c	compressor pressure
p'	pressure corresponding to $\mathbf{v}'$
p_ϵ	pressure in penalty function formulation
$Q(Q^m)$	nodal values of the adjoint pressure
$Q(t)$	flow rate
$Q^d(t)$	a desired flow rate
q_w	adjoint pressure corresponding to $\mathbf{w}$
q_z	adjoint pressure corresponding to $\mathbf{z}$
R, R_i	residuals
R^2, R^3	two- and three-dimensional domains
r, s, t	local coordinates
r_i, s_j, t_k	quadrature points
T	terminal time
t	time variable
$\mathbf{U}(U_i)$	vector of nodal values of the velocity vector
$U_i(U_i^n)$	nodal values of the velocity vector
U_{ad}	set of admissible controls
$\mathbf{u}(u_i)$	velocity vector
$\mathbf{u}^d$	desired velocity field
$\mathbf{u}'$	velocity vector corresponding to $\mathbf{v}'$
$\mathbf{u}_\epsilon$	velocity vector in penalty function formulation
$\mathbf{v}(v_i)$	control parameter
$\mathbf{v}'$	arbitrary control
$\mathbf{w}$	Gateaux derivative of $\mathbf{u}$ with respect to $\mathbf{v}$
W_i, W_j, W_k	weight coefficients
w	weight function
$\mathbf{x}(x_i)$	a generic point of the domain

x, y, z	global coordinates
$Z(Z_i^n)$	nodal values of the adjoint state
$\mathbf{z}(z_i)$	adjoint state
z	height of the membrane
X, Y, Z	nodal values of the global coordinates

Greek symbols

$\alpha_c, \alpha_g, \alpha_d, \alpha_i$	weight coefficients
Γ	boundary of the domain
Γ_f	fixed boundary
Γ_i	inlet boundary
Γ_o	outlet boundary
Γ_m	moving boundary
γ	convergence tolerance in optimization algorithm
Δ	Laplacian operator
δ_{ij}	Kronecker delta
Λ	approximating function, membrane boundary
λ	descent parameter
λ	second coefficient of viscosity
μ	coefficient of viscosity
ν	kinematic viscosity
ρ	density
$\sigma(\sigma_{ij})$	stress tensor
$\tau(\tau_i)$	traction vector
Φ	approximating function
Ψ	approximating function
ψ	streamfunction vector
ψ	streamfunction
Ω	physical domain

$\boldsymbol{\omega}$	vorticity vector
ω	vorticity in two-dimensional problems

Superscripts

$(\ )^{-1}$	inverse of a matrix
$\dot{(\ )}$	time derivative
$(\ )^e$	values in element level
$(\ )^m, (\ )^n$	nodal unknowns
$(\ )^o$	optimal value
$(\ )^T$	transpose of a vector or matrix
$(\ )^\epsilon$	values corresponding to $\mathbf{v}^\epsilon$

Subscripts

$(\ )_0$	initial value
$(\ )_\epsilon$	values in penalty function formulation
$(\ )_i$	vector components
$(\ )_{ij}$	matrix components
$(\ )_n$	n-th step

Other symbols

∇	del operator
$\|\ \|^2$	L_2 norm
$\|\ \|$	determinant

Chapter 1

Introduction

The outline of the present thesis is as follows. Chapter 1 is devoted to a literature review of subjects related to the present optimal control problem. Chapter 2 contains the formulation of simplified mathematical models of the artificial heart system and an analytical expression of the optimal control problem. The finite element formulation of the governing equations is presented in Chapter 3 for blood flow as well as the optimality conditions. Some results of the application of the optimal control algorithm to a few models of artificial hearts are reported in Chapter 4. The main conclusions of the present work are summarized in Chapter 5, followed by some suggestions for future research.

1.1 Motivation and Objectives

In recent years, there has been a growth of interest in theoretical studies of human organs. The heart, being of great importance to human life, has attracted the attention of many researchers in different areas of science and engineering. Indeed, certain aspects of cardiology lend themselves in a natural way to physical and mathematical description, modeling and analysis. Artificial hearts have been devised to assist and/or replace end stage failing natural hearts.

The objectives of this study are as follows:

a) to develop a mathematical model describing the blood flow in artificial hearts,

b) to propose some simplified models, suitable for the numerical study of the problem,

c) to apply optimal control theory to the governing equations in order to obtain the necessary conditions for optimal performance of the above models, and

d) to develop the necessary computer codes for the numerical solution of the governing equations and the corresponding optimality conditions.

1.2 Mathematical Modeling of the Human Heart

The process of analyzing and modeling of any physical system requires an identification of the geometry and materials, an expression of the governing natural laws and their mathematical formulation, the solution of the corresponding equations and the verification of the results.

Recent advances in the development of the theory of heart function and suitable engineering models have not yet had a significant impact in clinical cardiology. Theoretical studies deal with different aspects of the cardiac system, e.g. stress analysis of the heart, finite element modeling of ventricular mechanics and mathematical models of pacemaker tissue in the heart (Glass *et al.*, 1991). Some numerical studies on heart valves have been reported by Peskin (1972 and 1982). The purpose of heart is to pump blood to the circulation system. To do so continuously for over 70 years or more requires a very sophisticated organ (Glass *et al.*, 1991). Figure 1.1 shows a schematic of the natural heart. One may refer to Fung (1990) for details on the blood circulation inside the heart and other related aspects of blood flow.

Related to the study of the natural heart is the modeling of artificial hearts. The need for artificial hearts stems from the lack of adequate availability of natural hearts for transplantation and the need for supporting devices for partly disabled hearts. Artificial hearts will be discussed briefly in the following section.

1.3 Artificial Hearts

Although controversial, the development and clinical testing of artificial hearts has been necessitated by inadequate availability of heart donors for transplantation. By 1965, a number of research teams around the world, including teams in the USA, Japan and Europe, were involved in the development of artificial blood pumps as temporary support or permanent replacement of a failing heart. During the last two decades, this extensive research activity has led to the successful development of a number of experimental artificial blood pumps, such as the Penn State total artificial heart (TAH) and ventricular assist device (VAD), the Jarvik 7 TAH (University of Utah), the Novacor VAD, the Termedics VAD (Thermo Cardiosystems Inc.), the NIMBUS-Cleveland Clinic Foundation TAH and VAD, and the University of Ottawa EVAD.

The first generation of clinically applicable TAHs and VADs was based on models activated by external pneumatic drivers, while newer permanent devices, currently under development and testing, are meant to be totally implantable and powered by an internal source, such as electrical, thermal or even thermonuclear energy. An early review of the world experience in using artificial hearts has been presented by Joyce *et al.* (1986). One may also refer to Kolff (1988), Bernhard (1989) and Nose (1990) for detailed information on artificial hearts. Atsumi *et al.* (1990) report some developments of artificial hearts in Japan. Some artificial hearts consist of a blood chamber, a pusher plate or diaphragm and two valves. Some of these mechanisms are shown in Figure 1.2. For example, in one design (Figure 1.2b), the diaphragm divides the blood chamber into the pump atrium and the pump ventricle. In this design, the inflow valve is located on the diaphragm and the

outflow valve is incorporated into the outlet port of the blood chamber. In the systolic phase, the inflow valve is closed and the blood in the pump ventricle is sent out to the arterial side through the outflow valve. Suction is generated inside the pump atrium and blood enters simultaneously into the pump atrium from the left or right atrium. In the diastolic phase, the pressure inside the pump ventricle is reduced following the movement of the diaphragm, the inflow valve is opened and the blood enters into the pump atrium (Atsumi *et al.*, 1990). One may refer to Sakakibara *et al.* (1989) and Cheon *et al.* (1990) for alternative designs. Some experimental studies and flow measurements in artificial hearts have been reported by Tarbell *et al.* (1986), Mann *et al.* (1987), and Mussivand *et al.* (1988). In a recent report, Jin and Clark (1993) reported an experimental study of unsteady flow in a sac-type VAD and illustrated the development of flow patterns and the velocity distributions within the VAD chamber.

1.4 Optimization of Artificial Hearts

Since its early conceptualization in the 1960s, the artificial heart has been under continuous development. Considerable amount of studies have been directed towards artificial heart valves, which have been used both as components of total artificial hearts and as artificial cardiac prostheses. The most serious problems and complications experienced both by total artificial hearts and by heart valves may be divided in two distinct categories as follows.

Problems related to blood-contacting surfaces. Thromboembolism has been one of the major clinical complications in all long-term artificial heart recipients. Thromboembolism is believed to originate from the intrinsic thrombogenicity of the blood contacting surfaces within the cardiac devices (Lelkes and Samet, 1991).

Problems related to undesirable flow patterns inside the blood container. One of the problems and complications experienced by artificial hearts and artificial heart valves are related to the dynamics of blood flow. Damage to red cells and platelets could be

the result of elevated turbulent shear stresses (Woo and Yoganathan, 1986). Elevated shear stress levels can also cause element trauma (Williams, 1978). Blackshear (1972) has shown that red blood cells, which adhere to foreign surfaces, may be damaged by shear stresses of the order of 10-100 $dyn\ cm^{-2}$. Suterd and Mehrjardi (1975) have demonstrated that red blood cells lose their bi-concavity when subjected to bulk turbulent stresses of 500 $dyn\ cm^{-2}$ for a duration of 4 min. Clinically significant levels of hemolysis as well as thromboembolic complications are problems associated with implanted prosthetic valves (Yoganathan *et al.*, 1979 and 1982).

According to Blackshear (1972), Coulantuoni (1977), and Sallam and Hwang (1983), high turbulence activity can cause lysis of red cells (cell destruction) and platelet activation. The latter may lead to clot formation and thrombosis (Stein and Sabbah, 1974). The severity of damage to blood constituents by fluid shear is highly influenced by both the magnitude and the duration of the shear stress (Schwartz *et al.*, 1980; Tiederman *et al.*, 1986). According to the experimental study of heart valves by Galanga and Lloyd (1981), areas of high shear rates, slow-moving separated regions and stagnation point flows contribute to thrombus formation. These authors found high mass transfer rates both in regions of high velocity gradients and in regions of flow separation with low shear stresses, which, in their view, provide support to the hypothesis that thrombus formation occurs in both high- and low-shear regions.

Optimization of artificial hearts attempts to reduce and, if possible, eliminate the above problems. Significant resources are directed towards reducing the thrombogenicity of the blood containing surfaces. One of the proposed approaches include casting of a smooth polymeric surface, which would ensure minimal cell adhesion/growth improving hemocompatibility of the biomaterials by organo-chemical modification of their surfaces. One may refer to Lelkes and Samet (1991) for a detailed study on this issue.

The above review of experimental results is aimed at providing some initial insight towards formulating realistic optimization criteria, which identifies certain types of hemodynamic problems that the designer of artificial hearts has to take into consideration. It

should also be noted that the shape and size of artificial hearts are usually the results of empirical design, within constraints dictated by physiological considerations, power source limitations, biocompatibility of materials and ease in surgical implantation (Mussivand *et al.*, 1995). On the other hand, for a given heart design, it appears possible to minimize the probability of blood damage and thrombus formation by controlling the flow-generating system.

The main goal of this study is the application of optimal control theory to artificial hearts in an attempt to minimize the flow-related problems. In order to achieve this optimization, one should develop necessary conditions of optimality for blood flow and construct an efficient algorithm for the numerical solution of the problem.

In the present analysis, blood is assumed to be a homogeneous Newtonian fluid. Although it is known that, as a suspension, whole blood has non-Newtonian characteristics (Chein *et al.*, 1966; Shah, 1980; Bird *et al.*, 1983), at the high shear rates occurring near the walls of the heart and the pulmonary and systemic arteries and veins, whole blood can be modeled as a Newtonian fluid with a constant coefficient of viscosity (Pedley, 1980; Fung, 1981; Fung, 1990). A similar assumption may reasonably be made for blood flow in artificial hearts.

1.5 Optimal Control Theory and the Navier-Stokes Equations

The optimal control problem in its general sense consists of the following essential elements:

a) a mathematical model of a dynamical system to be controlled,

b) a specified target (or desired state) for the system,

c) a cost functional or performance index, and

d) a set of admissible controls.

Consider the differential equation in R^n

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{v}, t) \tag{1.1}$$

as the equation to be controlled. The vector $\mathbf{x}$ is the state vector and $\mathbf{v}(t)$ is the control parameter vector. The set of admissible controls, U_{ad}, is an arbitrary set in the space of control parameters R^r. This set is said to be the set of admissible values of the control parameter. An arbitrary measurable and bounded function, $\mathbf{v}(t)$, which is defined in the set U_{ad} is said to be an admissible control. The set of all admissible controls is called the class of admissible controls (Gamkrelidze, 1978).

Given a dynamical system, we are interested in moving the state from its specified initial condition to the desired state or specified target, while minimizing a given cost functional, $J(\mathbf{x}, \mathbf{v})$. This cost functional is supposed to quantify the performance of the system under application of different controls so that its value decreases as the quality of the control parameter improves. The optimal control problem is to find an admissible control, $\mathbf{v}^o(t)$, which simultaneously ensures the attainment of the system objective and the minimization of the cost functional, $J(\mathbf{x}, \mathbf{v})$, such that

$$J(\mathbf{x}, \mathbf{v}^o) \leq J(\mathbf{x}, \mathbf{v}) \tag{1.2}$$

A pair, $(\mathbf{x}, \mathbf{v}^o)$, which satisfies this property is called the optimal pair for the system (Owens, 1981).

Given an optimal control problem, two different aspects of the problem should be studied: the existence of an optimal control and the necessary (and possibly sufficient) conditions of optimality. The term "necessary conditions of optimality" refers to the set of conditions which should be satisfied in order to minimize the cost functional, $J(\mathbf{x}, \mathbf{v})$. For further studies on optimal control problems, the existence problem and the necessary conditions of optimality for specific differential equations, one may refer to Gamkrelidze (1978), Ahmed and Teo (1981), and Ahmed (1988). The subject of the optimal control theory has been under rapid development over the last three decades. In the early 1960s, Pontryagin *et al.* (1962) made a fundamental contribution to optimal

control theory, followed by further works of Gamkrelidze (1978) in the theory of necessary conditions for general optimization problems. Further developments of the subject have been achieved, among others, through the works of Warga (1972) in delay differential equations, Wonham (1970), Fleming and Rishel (1975), Ahmed (1979), and Krylov (1980) in stochastic differential equations and Butkovskiy (1969), Lions (1971), and Ahmed and Teo (1981) in partial differential equations.

Optimal control theory has found many interesting applications in physics, systems theory, chemical processes, mechanics, and many other science and engineering fields. In recent years, a considerable effort has been spent on the application of optimal control theory to fluid dynamics systems described by the Navier-Stokes equations. Among the earliest studies one may refer to the work of Cuvelier (1976), who applied optimal control theory to the Navier-Stokes equations coupled with the heat equation. Forsikov (1982 and 1983) has studied some control problems concerning the Navier-Stokes equations. More recently, Abergel and Temam (1990) have used optimal control theory towards minimizing turbulence in a few fluid mechanical problems through distributed and boundary controls. Fattorini and Sritharan (1992) have provided a mathematical formulation for the existence of optimal controls for certain boundary control problems, while Sritharan (1992) has studied an optimal control problem in exterior hydrodynamics. Gunzburger *et al.* (1992) have studied the problem of drag reduction in an incompressible flow through boundary velocity control. In a recent paper, Ahmed (1992b) has proved the existence of an optimal control for fluid systems with an initial velocity distribution or a boundary velocity distribution as the control variables. Finally, some general issues regarding shape optimization for the Navier-Stokes equations have been outlined by Gunzburger and Peterson (1992).

1.6 Modeling of Blood Flow

For Newtonian fluids, the relation between stress and deformation is given by a linear constitutive relation (see equation (1.14) in the next section). In these fluids the viscosity is assumed to be constant at a given temperature. Non-Newtonian fluids are those for which the constitutive relation (1.14) does not hold. For non-Newtonian fluids the interaction between stress, deformation and time can be much more complex. Non-Newtonian behaviour of fluids is observed in a number of different ways. Such fluids may be divided into two different categories: inelastic fluids and viscoelastic fluids. Inelastic fluids, also viewed as *generalized Newtonian fluids*, exhibit a viscosity that depends on shear rate. Pseudo-plastic (shear-thinning) fluids have a viscosity that decreases with increasing shear rate, while dilatant (shear-thickening) fluids have a viscosity that increases with decreasing shear rate. Viscoelastic fluids on the other hand, behave much more differently than Newtonian fluids do. In such fluids, the memory of the fluid is important, and the motion of a material element depends on both the present stress state and the deformation history of the element. This time-dependent behaviour leads to more complex constitutive relations. A comprehensive discussion of non-Newtonian fluids may be found in the texts by Bird *et al.* (1987) and Tanner (1985).

Human blood is a suspension of cells and plasma. The cellular contents are the red cells, white cells and platelets. Red cells which are the most numerous, completely dominate the mechanical properties of blood and occupy about 50% of the blood volume. Human red cells are deformable, disk shaped, with a diameter of 7.6 μm and an average thickness of 2.8 μm. They number about 5 million per mm^3. The normal white cell count is considered to be from 5,000 to 8,000 per mm^3 and, therefore, dynamically unimportant. White cells are slightly larger than red cells and equally deformable. The platelets are more numerous than white cells (250,000 to 300,000 per mm^3) and have a diameter of about 2.5 μm. They too are unimportant dynamically (see Pedley, 1980; Fung, 1981).

Experimental results show that plasma behaves like a Newtonian fluid (Merrill *et*

al., 1965; Gabe and Zazt, 1968), while the whole blood is found to be non-Newtonian (Chein *et al.*, 1966; Shah, 1980; Bird *et al.*, 1983). Therefore, the non-Newtonian features of whole blood come from the suspended cellular bodies. Some authors have sought to explain the flow behaviour of blood in terms of continuum theories. Among the available literature on the bulk properties of blood, one may refer to Goldsmith and Skalak (1975) and Fung (1981). Regarding the constitutive relation of blood, Fung (1981) has proposed the following equation

$$\sigma_{ij} = -p\delta_{ij} + 2\mu(I)d_{ij} \tag{1.3}$$

in which σ_{ij} is the stress tensor, p is the pressure, μ is the coefficient of viscosity, $I = \frac{1}{2}d_{ij}d_{ij}$ and d_{ij} is the rate-of-deformation tensor, defined by equation (1.15) in the next section. This equation describes a strain-rate dependent viscosity for the whole blood. However, Fung (1981) discusses that, at high shear rates, whole blood behaves like a Newtonian fluid with a constant coefficient of viscosity, while, at low shear rates, non-Newtonian behaviour of blood can be described by the constitutive relation (1.3). In relatively large animals such as cat and man, the shear strain rate of blood near the walls of the heart and the pulmonary and systemic arteries and veins exceeds 100 s^{-1}, so that in these regions the viscosity of blood may be regarded as constant, and flowing blood in these vessels may be treated as Newtonian (Pedley, 1980; Fung, 1990). A similar assumption appears to apply to blood flow in artificial hearts, at least for regions where the shear stress is not extremely low.

Whole blood cannot be regarded as a homogeneous fluid in the small blood vessels, because the diameters and spacings of red cells are comparable to the capillary vessel diameters. However, in analyzing blood flow in large blood vessels, whose diameters exceed 100 μm, and, consequently, in blood chambers of artificial hearts, blood can be regarded as effectively homogeneous (Pedley, 1980; Fung, 1981).

According to Peskin (1982), the most difficult experimental information to obtain *in vivo* is a measurement of the flow pattern of blood in the heart. This problem was first attacked by Taylor and Wade (1973), who used thin, carefully injected streams of

contrast material to trace the flow pattern of blood around the atrioventricular valves on cineangiograms. Their principal finding was that the flow pattern of blood in the heart forms a stable system with no evidence of large-scale mixing, disorder or turbulence. This is an experimental result of great importance for any theoretical treatment of the fluid dynamics of heart and heart valves. However, there are some indications (Mussivand *et al.*, 1988) that the blood flow in artificial hearts, at least in some regions, is turbulent.

In conclusion, we shall take blood to be a homogeneous Newtonian fluid of density ρ=1.05$\times 10^3 kg\ m^{-3}$, viscosity μ=0.004 $kg\ m^{-1}s^{-1}$ (at 37°C) and, hence, kinematic viscosity ν=3.5 $\times 10^{-6} m^2 s^{-1}$ (Pedley, 1980). Blood flow in artificial hearts will be assumed to be laminar (Peskin, 1982).

Concerning the non-Newtonian behaviour of blood, Shah (1980) has written a review paper on the subject and Bird *et al.* (1983) have reported on the application of the Casson model for representing the blood flow. Nakamura (1990) has also reported a numerical study on the non-Newtonian blood flow in an axisymmetric stenosis. Pham and Mitsoulis (1994) have numerically studied the non-Newtonian blood flow in an abrupt contraction using the Casson model. Other works in this area include papers by Perktold (1991 and 1994), in which the pulsatile non-Newtonian blood flow in carotid artery models have been numerically studied.

1.7 Governing Equations of the Fluid Flow

In this section the governing equation of blood flow in an artificial heart will be considered. Because blood flow is characterized by non-Newtonian behaviour of the fluid, we shall first consider the general equations of motion. For an isothermal flow, this consists of the equations for the conservation of mass and momentum and general constitutive relations. Afterwards, we shall simplify the general equations for Newtonian fluids to obtain the so-called Navier-Stokes equations.

Conservation of Mass:

Let Ω denote a physical closed domain in R^3 with the boundary Γ. A generic point in the domain is represented by $\mathbf{x} = (x_1, x_2, x_3)$. According to the principle of the conservation of mass, the total mass of a system occupying the domain Ω does not change when the domain moves with the system

$$\frac{dm}{dt} = \frac{d}{dt}\int_\Omega \rho d\Omega = 0 \tag{1.4}$$

in which m denotes the total mass of the fluid and ρ the density of the fluid. Consider the Reynolds transport theorem (Tanner, 1985)

$$\frac{d}{dt}\int_\Omega \rho\psi d\Omega = \int_\Omega \frac{\partial}{\partial t}(\rho\psi)d\Omega + \int_S \rho\psi \mathbf{u}\cdot\mathbf{n}dS \tag{1.5}$$

where S is the surface of the domain Ω and $\mathbf{n}$ is the outward unit vector normal to the boundary. The quantity ψ may be any specific (per unit mass) quantity of a scalar, vector or tensor nature, and $\mathbf{u} = (u_1, u_2, u_3)$ denotes the velocity vector. Applying (1.5) with ψ =1 to (1.4) yields

$$\int_\Omega \frac{\partial}{\partial t}\rho d\Omega + \int_S \rho\mathbf{u}\cdot\mathbf{n}dS = 0. \tag{1.6}$$

By using the divergence theorem,

$$\int_S \Phi\cdot\mathbf{n}dS = \int_\Omega \nabla\cdot\Phi d\Omega, \tag{1.7}$$

where Φ is any vector, the second term in (1.6) can be converted to a volume integral, so that

$$\int_\Omega (\frac{\partial\rho}{\partial t} + \nabla\cdot\rho\mathbf{u})d\Omega = 0. \tag{1.8}$$

Since the domain Ω is arbitrary, one may conclude that

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\rho\mathbf{u} = 0 \tag{1.9}$$

which is called the continuity equation.

Conservation of Momentum:

According to the principle of conservation of momentum, the rate of change of linear momentum, $\rho\mathbf{u}d\Omega$, of a system, occupying the material volume Ω, is equal to the net

force on the system, consisting of surface and body forces. Using the transport theorem (1.5) with $\psi = \mathbf{u}$ for the rate of change of linear momentum, one finds

$$\int_\Omega \frac{\partial}{\partial t}(\rho \mathbf{u}) d\Omega + \int_S (\rho \mathbf{u})\mathbf{u}\cdot\mathbf{n} dS = \int_\Omega \rho \mathbf{f} d\Omega + \int_S \tau dS \tag{1.10}$$

in which $\mathbf{f} = (f_1, f_2, f_3)$ is the body force per unit mass and τ is the traction vector defined as

$$\tau_i = \sigma_{ij} n_j \tag{1.11}$$

where σ_{ij} are the components of the stress tensor. Using the divergence theorem (1.7) for surface integrals in (1.10) yields, in cartesian indicial notation,

$$\int_\Omega \left\{ \frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) \right\} d\Omega = \int_\Omega \left\{ \frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i \right\} d\Omega. \tag{1.12}$$

Considering that equation (1.12) should apply to any volume Ω and using the continuity equation (1.9), equation (1.12) can be written in differential form as

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = \rho f_i + \frac{\partial \sigma_{ij}}{\partial x_j}. \tag{1.13}$$

This equation is the general differential momentum equation for fluids. Together with the continuity equation it forms a set of governing equations for fluid flow. Because the number of unknowns is higher than the number of equations, the concept of constitutive relations has been introduced (see e.g. Landau and Lifshitz, 1959) which relates the stress to the motion of the material. Newtonian and non-Newtonian fluids differ from each other in the way these constitutive relations are defined. A Newtonian fluid is defined by the following constitutive relation for the stress tensor

$$\sigma_{ij} = -p\delta_{ij} + \lambda d_{kk}\delta_{ij} + 2\mu d_{ij} \tag{1.14}$$

where λ and μ are coefficients of viscosity, p is the pressure and d_{ij} is the rate-of-deformation tensor, defined as

$$d_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \tag{1.15}$$

If the fluid is incompressible, then the continuity equation (1.9), and accordingly the Newtonian constitutive relation (1.14), can be simplified to the following equations, respectively

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1.16}$$

$$\sigma_{ij} = -p\delta_{ij} + 2\mu d_{ij}. \tag{1.17}$$

By substituting the Newtonian constitutive relation (1.17) in the momentum equation (1.13) and assuming a constant viscosity, one may obtain the so-called Navier-Stokes equations

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} - \nu\frac{\partial}{x_j}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) + \frac{1}{\rho}\frac{\partial p}{\partial x_i} = f_i \tag{1.18}$$

in which $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity. In vector notation these equations can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}.\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f}. \tag{1.19}$$

Equations (1.16) and (1.19) are the governing equations of a Newtonian incompressible fluid, which may be solved over the domain Ω after introducing appropriate initial and boundary conditions.

1.8 Alternative Formulations of the Navier-Stokes Equations

In this section, some alternative formulations of the Navier-Stokes and continuity equations will be briefly discussed.

1.8.1 The Velocity-Pressure Formulation

This formulation, also called "primitive formulation", refers to the Navier-Stokes and continuity equations in their original form with $(\mathbf{u}, p)$ as unknowns. Using the symbols as defined in section 1.6, this formulation will have the following form:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{t=0} = \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega \ ; \ \mathbf{u} = g(\mathbf{x},t) \quad \text{on the boundary } \Gamma \times [0,T] \end{cases} \tag{1.20}$$

where $g(\mathbf{x},t)$ is a given function and $[0,T]$ denotes the time interval.

1.8.2 The Penalty Function Method

The presence of the continuity equation in the governing equations introduces difficulties in the numerical treatment of the problem (Reddy, 1982; Cuvelier *et al.*, 1986). The so called penalty method has been introduced to decouple the momentum equations and the continuity equation by eliminating the pressure from the momentum equations. In this method, the continuity equation is perturbed with a sufficiently small proportion of the pressure. The formulation procedure is the same for transient or steady, Navier-Stokes or Stokes (neglecting the inertia term) equations. For simplicity, the steady Stokes equations will be considered first in the following discussion and then the results will be extended to the unsteady Navier-Stokes equations.

The original formulation of the penalty method has been on the basis of a minimization problem. It follows from the calculus of variations that a necessary condition to make a functional stationary is that the Euler-Lagrange equation of the functional be satisfied together with the boundary conditions. This Euler-Lagrange equation is precisely the governing partial differential equation of the problem (Cuvelier *et al.*, 1986, and Ahmed, 1988). Consider the following Stokes equations with homogeneous boundary conditions

$$\begin{aligned} -\nu\nabla^2\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in } \Omega \\ \nabla\cdot\mathbf{u} &= 0 \quad \text{in } \Omega \\ \mathbf{u} &= 0 \quad \text{on the boundary } \Gamma. \end{aligned} \tag{1.21}$$

It can be shown (Girault and Raviart, 1986) that the Stokes equations (1.21) are the Euler-Lagrange equations for the following minimization problem

$$\min\{J(\mathbf{u},p) : \ \mathbf{u} = 0 \quad \text{on } \Gamma\}$$

where

$$J(\mathbf{u},p) = \nu\int_\Omega \nabla\mathbf{u}\cdot\nabla\mathbf{u}^T d\Omega - 2\int_\Omega p\nabla\cdot\mathbf{u} d\Omega - 2\int_\Omega \mathbf{f}\cdot\mathbf{u} d\Omega. \tag{1.22}$$

It is also known that one may introduce a constraint in a minimization problem through penalization. Thus, for $\epsilon > 0$, one may consider a minimization problem of the form

$$\min\{J_\epsilon(\mathbf{u}) : \ \mathbf{u} = 0 \quad \text{on } \Gamma\}$$

$$J_\epsilon(\mathbf{u}) = \nu \int_\Omega \nabla \mathbf{u} \cdot \nabla \mathbf{u}^T d\Omega - \frac{1}{\epsilon} \int_\Omega (\nabla \cdot \mathbf{u}) \cdot (\nabla \cdot \mathbf{u}) d\Omega - 2 \int_\Omega \mathbf{f} \cdot \mathbf{u} d\Omega. \tag{1.23}$$

Then the Euler-Lagrange equations of the functional (1.23) are given by

$$\begin{aligned} -\nu \nabla^2 \mathbf{u}_\epsilon - \tfrac{1}{\epsilon} \nabla (\nabla \cdot \mathbf{u}_\epsilon) &= \mathbf{f} \quad \text{in } \Omega \\ \mathbf{u}_\epsilon &= 0 \quad \text{on the boundary } \Gamma. \end{aligned} \tag{1.24}$$

By introducing $p_\epsilon = -\frac{1}{\epsilon}(\nabla \cdot \mathbf{u}_\epsilon)$, equations (1.24) can be written as

$$\begin{aligned} -\nu \nabla^2 \mathbf{u}_\epsilon + \nabla p_\epsilon &= \mathbf{f} \quad \text{in } \Omega \\ \nabla \cdot \mathbf{u}_\epsilon &= -\epsilon p_\epsilon \end{aligned} \tag{1.25}$$

which implies that the effect of penalization is simply to relax the incompressibility condition (Gunzburger, 1989). Given a small parameter, ϵ, one may extend the above result in the Navier- Stokes equations by considering a perturbed version of these equations, as follows

$$\begin{aligned} -\tfrac{\partial \mathbf{u}_\epsilon}{\partial t} + (\mathbf{u}_\epsilon \cdot \nabla)\mathbf{u}_\epsilon - \nu \nabla^2 \mathbf{u}_\epsilon + \nabla p_\epsilon &= \mathbf{f} \quad \text{in } \Omega \\ \nabla \cdot \mathbf{u}_\epsilon &= -\epsilon p_\epsilon \quad \text{in } \Omega \\ \mathbf{u}_\epsilon &= 0 \quad \text{on } \Gamma. \end{aligned} \tag{1.26}$$

Then the pressure may be easily eliminated from the above equations to yield the following penalty formulation of the Navier-Stokes equations

$$\begin{aligned} -\tfrac{\partial \mathbf{u}_\epsilon}{\partial t} + (\mathbf{u}_\epsilon \cdot \nabla)\mathbf{u}_\epsilon - \nu \nabla^2 \mathbf{u}_\epsilon - \tfrac{1}{\epsilon} \nabla (\nabla \cdot \mathbf{u}_\epsilon) &= \mathbf{f} \quad \text{in } \Omega \\ \mathbf{u}_\epsilon &= 0 \quad \text{on } \Gamma. \end{aligned} \tag{1.27}$$

Once the decoupled equations (1.27) are solved, the pressure field may be recovered using the second equation in (1.25). Regarding the implementation of this method, the convergence of the solution of the perturbed system to the exact solution and the effects of the penalty parameter, ϵ, on the accuracy of the results, one may refer to Polya (1971), Hughes *et al.* (1979), Reddy (1979 and 1982), Cuvelier *et al.* (1986), and Gunzburger (1989). Mathematical aspects of the method may be found in Temam (1979) and Girault and Raviart (1986).

1.8.3 The Streamfunction-Vorticity Formulation

For an incompressible flow, a vector ψ, can be introduced as

$$\mathbf{u} = \nabla \times \psi \tag{1.28}$$

In a two-dimensional medium, this is reduced to $\psi = (0, 0, \psi)$, where ψ is known as the streamfunction.

Similarly, the (vector) vorticity, ω, is introduced to be

$$\omega = \nabla \times \mathbf{u} \tag{1.29}$$

which in the case of a two-dimensional flow, reduces to $\omega = (0, 0, \omega)$.

To derive the streamfunction-vorticity formulation, we limit our discussion to two-dimensional problems. In this context we shall utilize a definition of curl operators, which is distinct from the one customarily used in three-dimensional flows (Gunzburger, 1989). The curl operators in R^2 are defined as

$$\mathbf{u} = \nabla \times \psi = (\frac{\partial \psi}{\partial x_2}, -\frac{\partial \psi}{\partial x_1}) \tag{1.30}$$

and

$$\omega = \nabla \times \mathbf{u} = (\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}). \tag{1.31}$$

In the above equations, the first curl operator takes the scalar ψ into the vector $\mathbf{u}$, while the second does the opposite and takes the vector $\mathbf{u}$ into the scalar ω. Since for any scalar function, ϕ, $\nabla \cdot (\nabla \times \phi) = 0$, the continuity equation is satisfied by the above definition of the stream function ψ. The following equalities can be easily verified using the above definition for ω in (1.29) as

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = (\nabla \times \mathbf{u}) \times \mathbf{u} + \nabla(\frac{1}{2}(\mathbf{u} \cdot \mathbf{u})) = \omega \times \mathbf{u} + \nabla(\frac{1}{2}(\mathbf{u} \cdot \mathbf{u})) \tag{1.32}$$

$$\nabla \times \omega = \nabla \times (\nabla \times \mathbf{u}) = -\nabla^2 \mathbf{u}. \tag{1.33}$$

Taking the curl of the momentum equation in (1.20), using the identities in equations (1.32) and (1.33), using the continuity equation and also noting the fact that, for any

scalar function, ϕ, $\nabla \times (\nabla \phi) = 0$, one may obtain the following equation

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega - \nu \nabla^2 \omega = \frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2} \tag{1.34}$$

which is the vorticity transport equation. Substituting $\mathbf{u}$ from (1.28) in the above equation yields

$$\frac{\partial \omega}{\partial t} + (\nabla \times \psi) \cdot (\nabla \omega) - \nu \nabla^2 \omega = \frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2} \tag{1.35}$$

in which ν is the kinematic viscosity. y combining equations (1.30) and (1.31), one may obtain

$$\nabla^2 \psi = -\omega. \tag{1.36}$$

The set of equations (1.35) and (1.36), with appropriate boundary and initial conditions, form the streamfunction-vorticity formulation equivalent to the original formulation (1.20) of the Navier-Stokes equations for two-dimensional flows. One may note that these are two scalar equations with two unknowns ω and ψ.

Introducing the appropriate boundary and initial conditions is an important issue in this formulation. For a detailed discussion on initial and boundary conditions for this formulation, one may refer to Cuvelier *et al.* (1986) and Gunzburger (1989). Several solutions are available in the literature for the above equations in particular configurations. Some of them may be found in Campion and Crochet (1978), and Dhatt *et al.* (1981).

1.8.4 The Streamfunction Formulation

Another alternative formulation for two-dimensional flows, may be obtained by substituting (1.36) in equation (1.35) as

$$\frac{\partial \Delta \psi}{\partial t} + \frac{\partial \psi}{\partial x_2}\frac{\partial \Delta \psi}{\partial x_1} - \frac{\partial \psi}{\partial x_1}\frac{\partial \Delta \psi}{\partial x_2} - \nu \Delta^2 \psi = \frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2}. \tag{1.37}$$

This fourth-order partial differential equation is called the streamfunction formulation, which is equivalent to the original Navier-Stokes formulation of the problem. One may note that in this formulation, the problem has been reduced to one scalar equation, in

which the only unknown is the streamfunction ψ. On the other hand, in this formulation, a fourth-order partial differential equation should be dealt with.

Similarly to the streamfunction-vorticity formulation, the continuity equation is automatically accounted for in this formulation. In both formulations the velocity and pressure fields can be recovered. Once the streamfunction, ψ, is determined, equation (1.30) may be used to recover the velocity field. The most common method to recover the pressure field is taking the divergence of the original momentum equation in (1.20) and using the continuity equation and equation (1.27) to obtain the following Poisson equation for the pressure

$$\Delta p = 2\left\{(\frac{\partial^2\psi}{\partial x_1^2})(\frac{\partial^2\psi}{\partial x_2^2}) - (\frac{\partial^2\psi}{\partial x_1 \partial x_2})^2\right\} + \frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} \tag{1.38}$$

One may refer to Gunzburger (1989) for more discussion on the recovery of the velocity and pressure fields. The boundary and initial conditions in this formulation can be found in the same references stated for the streamfunction-vorticity formulation. Some mathematical aspects of this formulation have been presented in Girault and Raviart (1986).

1.8.5 Evaluation of the Various Formulations

In this section we shall briefly compare the various formulations of the Navier-Stokes equations. In the velocity-pressure formulation, the unknowns are the velocity and the pressure fields, which are physical quantities. This method can be applied to both two- and three-dimensional problems. No boundary condition for pressure is specified in this method, while boundary conditions are implicitly applied in terms of boundary stresses. The numerical solution of the equations requires a considerable amount of computing time and memory space.

In the streamfunction-vorticity and streamfunction formulations, the continuity equation is automatically satisfied, so that the set of equations to be solved does not

contain that equation. In the streamfunction-vorticity formulation one has to deal only with two unknowns ψ and ω, and in the case of the streamfunction formulation only with one unknown ψ. These methods, however, can be applied only to two-dimensional and axi-symmetric problems, due to the use of the streamfunction. In these formulations, the velocity is obtained by numerical differentiation of the computed streamfunction, ψ, which is an inaccurate process. Finally, defining the appropriate boundary conditions for vorticity is not very straightforward. In the streamfunction formulation, the problem is reduced to one scalar, fourth-order, differential equation. Numerical solution of this equation by the finite element method is not an easy task and it is usually solved by the finite difference method.

In the penalty formulation, one needs smaller amounts of computing time and memory, compared to those required for the velocity-pressure formulation. In this method, however, the accuracy of the results depends on a suitable choice of the penalty parameter.

We shall use the penalty function formulation of the Navier-stokes equations for numerical simulation of these equations. This formulation applies equally to two- and three-dimensional problems and, especially in the later case, it reduces substantially the amount of memory size and CPU time. There is no difficulty in applying this method to the adjoint equations, to be introduced in Chapter 2.

1.9 Computational Fluid Dynamics

Because an exact solution of the general governing equations for viscous flow is unavailable, it is necessary to seek a computational solution. Computational techniques replace the governing partial differential equations with systems of algebraic equations, so that a digital computer can be used to obtain the solution. Among the available numerical techniques in computational fluid dynamics, one may refer to the finite difference and the finite element methods as the most commonly used ones. The finite difference model of a

problem gives a pointwise approximation to the governing equations. In this method the set of algebraic equations is obtained by writing the difference equations for all the grid points of the domain (Huebner, 1975). The difference equation is obtained by expressing the differentials in the governing equation as finite differences using a Taylor series expansion. One may refer to Forsythe and Wasow (1960) for an extensive discussion on finite difference methods for partial differential equations.

The finite difference formulation of the Navier-Stokes equations has been known for many years, but the extensive resources and time required to carry out step-by-step solution of the equation were prohibitive until about 1950, when the electronic computer made high-speed calculations possible, and the works of Von Neumann gave a considerable impetus to this field. The report by Cebeci and Smith (1968) explains an early application of the method for solving the incompressible, turbulent, boundary-layer equations.

In the finite element method, unlike the finite difference method, which replaces the solution region by an array of grid points, the domain is replaced by many small, interconnected subregions or elements. This method also gives a piecewise approximation to the governing equations (Huebner, 1975). The finite element method was originally developed by engineers in the 1950s to analyze large structural systems for aircrafts (Baker, 1983). Later, Zienkiewicz and Cheung (1965) reported the applicability of this method to field problems like flow and electromagnetism. One may consult Brebbia and Connor (1975), Huebner and Thornton (1982), Rao (1982), Zienkiewicz (1979) and Zienkiewicz and Morgan (1983) for more recent works in structural finite element analysis.

Despite having a long history in solving structural problems, the finite element method has only recently been applied to computational fluid dynamics. Oden (1972) was among the first who derived the finite element equations for the Navier-Stokes equations. Some results on elementary incompressible flow were first reported by Baker (1971, 1973 and 1974), Talor and Hood (1973) and Tanner (1973). Ladyzhenskaya (1969, 1970a and 1970b), Oden and Reddy (1976), Thomasset (1981), Glowinski (1984) and Girault

and Ravier (1986), have contributed to the development of the mathematical theory and numerical analysis of the Navier-Stokes equations. Baker (1983), Cuvelier *et al.* (1986), and Glowinski and Pironneau (1992) have described some finite element aspects of the equations.

One interesting aspect in the application of the finite element method in fluid flows is the modeling of the flow domains with moving and free surface boundaries. Some examples of such problems are extrusion processes, flow in cylinder-piston systems and blood flow in an artificial heart. Floryan and Rasmussen (1989) have provided a useful review on the subject.

1.10 The Finite Element Method

The finite element method is a numerical technique which can be applied to a wide variety of ordinary and partial differential equations of interest to physics and engineering to obtain approximate solutions. In this method, like for any numerical technique, the governing continuous differential equation, having infinite degrees of freedom, is reduced to a discrete set of equations with a finite number of degrees of freedom. The solution procedure in the finite element method can be summarized in the following steps.

1. *Definition of the Governing Partial Differential Equation.* The first step is to define the equations to be solved and the appropriate boundary and initial conditions. The physical region in which the solution of the governing equation is sought must also be defined.

2. *Discretization of the Domain.* The continuous region is divided into a number of simple subdomains which are called *finite elements.* The collection of these elements is called the finite element mesh. A wide variety of element shapes has been used, including triangles, rectangles and quadrilaterals. Different element shapes may be used in the same solution domain. In general, the solution becomes more accurate as the number of the

elements increases. However, the optimal number and the types of the elements to be used depends on the particular problem to be solved and the analyst's experience.

3. *Polynomial Approximation.* In the finite element method, method. In variational methods, the differential equation is transformed into an equivalent integral form and then the approximate solution is assumed to be of the form

$$u = \sum_{i=1}^{n} c_i \Phi_i \tag{1.39}$$

where Φ_i are known approximate functions and c_i are unknown parameters to be determined in such a way that the integral form of the original equation is satisfied. This can be achieved by a variational approach such as the Ritz method. In the Ritz method, a potential functional equivalent to the differential formulation of the problem should be known. The approximate solution is substituted into the functional, which is then differentiated with respect to each coefficient c_i, and the resulting expressions are set equal to zero. The solution of these simultaneous equations determines the unknown coefficients c_i, which in turn determine the approximate solution. In this method, the approximating functions are required to be defined over the whole domain and to satisfy the boundary conditions (Huebner, 1975).

Another common approach is the weighted residual method, in which the approximate functions are defined only over one element and they have to satisfy no boundary conditions, while the coefficients c_i are taken to be nodal values of the solution variable, U_i, at each element, namely $u = \sum_{i=1}^{n} U_i \Phi_i$. Moreover in this method there is no need for the equivalent functional of the governing differential equations. Substitution of the approximation (1.39) into the governing differential equation $\ell(\Phi, u) = 0$ would result to the residual (error) R, written as

$$\ell(\Phi, u) = R. \tag{1.40}$$

In the weighted residual method, this residual is reduced to zero by making the residual orthogonal to a weight function, w, as

$$\int_{\Omega} w \cdot R d\Omega = \int_{\Omega} w \cdot \ell(\Phi, u) d\Omega = 0. \tag{1.41}$$

The weighted residual method is known as Galerkin method, if the weight function w is chosen to be the same as the approximation function Φ in (1.41). Repeating the same procedure for all elements of the whole domain would result in integral equations for the elements (Huebner, 1975).

4. *Assembly of the Element Equations.* After determining the matrix equations for the individual elements, they should be combined to form the global system of equations describing the entire domain. The basis for the assembly procedure is to require that, at each node which is common to neighbouring elements, the solution value has to be the same for all elements containing that node.

5. *Imposition of the Boundary and Initial Conditions.* Once the global system of equations is obtained, the boundary and initial conditions are applied.

6. *Solution of the Algebraic Equations.* The global system of equations should be solved to obtain the unknown nodal values of the state variable. This may be performed by either direct matrix solvers, like the Gaussian elimination method or the Cholesky decomposition method, or iterative methods, like the Gauss-Seidel method.

7. *Post-Processing of the Results.* The final stage in a finite element solution is to present the results in such a way as to facilitate their interpretation. This is usually done by graphic display of the solution variables and other related quantities.

For further details concerning the finite element method, one may refer to standard texts by Huebner (1975), Hinton and Owen (1979), Davis (1980), Becker *et al.* (1981), Huebner and Thornton (1982), Rao (1982), Carey and Oden (1983), Zienkiewicz (1979), Zienkiewicz and Morgan (1983), and Reddy (1984).

Chapter 2

Mathematical Formulation of the Optimal Control Problem

This Chapter describes models of the artificial heart and the optimal control problem related to it. In section 2.1 we present a mathematical model for a generic commercial-type artificial heart. Some simplified models are then discussed, which will be used later to develop the formulation of the optimal control problem and the necessary computational codes. In section 2.2, we introduce the cost functional related to the physical problem to be used in formulation of the optimal control problem. In section 2.3, a model of the artificial heart will be considered and the corresponding optimal control formulation will be stated, followed by a discussion on another simplified model of the problem in section 2.4.

2.1 Modeling of the Physical Problem

In this section the mathematical formulation of the blood flow in an artificial heart is presented. The formulation of a general model will be followed by some simplified formulations , which are more suitable for numerical study of the optimal control problem.

In section 1.3 some artificial hearts were schematically shown. For the purpose of the formulation, we consider the artificial heart in Figure 2.1, which consists of two chambers, two valves for inflow and outflow of the blood and a membrane. The membrane is the moving part of the flow boundary. On one side of the membrane there is blood flow and on the other side there is a chamber, connected to an air supply. The air pressure varies periodically with the use of pneumatic controls. The domain of the flow and its boundaries are denoted as follows,

Ω: the blood flow domain, $\Omega \in R^3$,

Γ: the whole boundary of the flow domain,

Γ_f: fixed boundary of the domain,

Γ_i: the boundary of the inlet valve, with cross-sectional area A_i,

Γ_o: the boundary of the outlet valve, with cross-sectional area A_o,

Γ_m: moving boundary of the flow (membrane surface),

Λ: boundary of the membrane fixed to the walls.

The above boundaries satisfy the relation: $\Gamma_f \cup \Gamma_i \cup \Gamma_o \cup \Gamma_m = \Gamma$. In the following models, we assume that they are isolated from the characteristics of the cardiovascular systems.

Model No. 1

In this model the artificial heart of Figure 2.1 is considered. The governing equation of the membrane is coupled to those of the flow inside the blood chamber, for the location of the membrane is the time-dependent boundary for the blood flow. The motion of the membrane depends on the pressure and flow patterns inside both the blood and the air chambers. The formulation of this model would result in a system of coupled nonlinear partial differential equations. The coupled system of governing equations of the flow in this model has been presented by Ahmed (1992a) as follows.

Membrane dynamics:

$$\begin{cases} \rho_m \frac{\partial^2 z}{\partial t^2} - (\sigma_{11}\frac{\partial^2 z}{\partial x^2} + 2\sigma_{12}\frac{\partial^2 z}{\partial x \partial y} + \sigma_{22}\frac{\partial^2 z}{\partial y^2}) = p_c(t) - p(x,y,t); \quad (x,y) \in \Gamma_m \\ z(x,y,t)|_\Lambda = 0 \\ z(x,y,0) = z_0; \quad (x,y) \in \Gamma_m \\ \frac{\partial z}{\partial t}(x,y,0) = z_1; \quad (x,y) \in \Gamma_m. \end{cases} \tag{2.1}$$

Fluid dynamics:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{t=0} = \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0 \quad \text{in } \Omega \\ \mathbf{u}|_{\Gamma_f} = 0; \quad \mathbf{u}\cdot\mathbf{n}|_{\Gamma_m} = -\frac{\partial \mathbf{Z}}{\partial t}\cdot\mathbf{n} \\ \begin{cases} \mathbf{u}\cdot\mathbf{n}|_{\Gamma_i} = 0; \quad \mathbf{u}\cdot\mathbf{n}|_{\Gamma_o} = \frac{1}{A_o}\int_{\Gamma_m}\frac{\partial z}{\partial t}(x,y,t)dxdy \quad \text{for systole half cycle} \\ \mathbf{u}\cdot\mathbf{n}|_{\Gamma_i} = \frac{1}{A_i}\int_{\Gamma_m}\frac{\partial z}{\partial t}(x,y,t)dxdy; \quad \mathbf{u}\cdot\mathbf{n}|_{\Gamma_o} = 0 \quad \text{for diastole half cycle} \end{cases} \end{cases} \tag{2.2}$$

where $\mathbf{z} = (0,0,z)$, $\mathbf{x} = (x,y,z)$, $\mathbf{u} = (u_1,u_2,u_3)$ and

$p_c(t)$: compressor pressure applied to one side of the membrane,

$p(t,x,y)$: fluid pressure on the surface of the membrane inside the blood chamber,

z: height of the membrane,

ρ_m: mass density of the membrane,

σ_{ij}: components of the stress tensor on the surface of the membrane.

The above equations contain a number of simplifying assumptions, including the following:

1) Shear stresses acting on the membrane surface from the fluids on either side are negligible.

2) The body force on the membrane is neglected due to its negligible mass.

3) The supplied air pressure, $p_c(t)$, is uniform in space.

4) The tangential velocity of the fluid on the membrane is neglected.

One may note that the dynamics of the fluid are coupled to that of the membrane through the boundary condition for the fluid. This formulation represents a coupled and moving boundary problem, which is difficult to deal with both in deriving the necessary

conditions of optimality and in solving them numerically. To avoid such complications, we are going to consider some simplified models, to be presented in the remaining of this section.

Model No. 2

In order to simplify the general formulation described above, we decouple the dynamic equation of the membrane from the governing equations of the fluid. This requires the assumption that the membrane has a known, time-dependent motion, or, in other words, that the moving boundary Γ_m of the domain Ω has a prescribed motion. In that case, the governing equations and the related boundary and initial conditions are reduced to the following form

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f} \quad \text{in } \Omega\times(0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{t=0} = \mathbf{u}(\mathbf{x},0) = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{\Gamma_f} = 0; \quad \mathbf{u}|_{\Gamma_m} = \frac{\partial \mathbf{h}(\mathbf{x},t)}{\partial t} \\ \begin{cases} \mathbf{u}|_{\Gamma_i} = 0; \quad \mathbf{u}\cdot\mathbf{n}|_{\Gamma_o} = \frac{1}{A_o}\int_{\Gamma_m}\frac{\partial \mathbf{h}}{\partial t}\cdot\mathbf{n}ds \quad \text{for systole half cycle} \\ \mathbf{u}\cdot\mathbf{n}|_{\Gamma_i} = \frac{1}{A_i}\int_{\Gamma_m}\frac{\partial \mathbf{h}}{\partial t}\cdot\mathbf{n}ds; \quad \mathbf{u}\cdot\mathbf{n}|_{\Gamma_o} = 0 \quad \text{for diastole half cycle} \end{cases} \end{cases} \tag{2.3}$$

in which $\mathbf{h}(\mathbf{x},t)$ is the prescribed position of the membrane. One may note that this problem is a moving boundary problem and the size of the domain is varying as a result of the movement of the boundary Γ_m.

Model No. 3

Some further simplification of the problem can be made by replacing the moving boundary by a fixed boundary. This is done by assuming that the boundary Γ_m is no longer moving but, instead, a prescribed fluid velocity, equivalent to the motion of the boundary Γ_m is imposed on a fixed surface. This assumption results in the following

equations

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{t=0} = \mathbf{u}(\mathbf{x},0) = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{\Gamma_f} = 0; \quad \mathbf{u}|_{\Gamma_m} = \mathbf{v}(\mathbf{x},t) \\ \begin{cases} \mathbf{u}|_{\Gamma_i} = 0; \quad \boldsymbol{\tau}|_{\Gamma_o} = 0; \quad 0 < t \le T_1 \\ \boldsymbol{\tau}|_{\Gamma_i} = 0; \quad \mathbf{u}|_{\Gamma_o} = 0; \quad T_1 < t \le T \end{cases} \end{cases} \tag{2.4}$$

in which $\boldsymbol{\tau}$ is the traction vector defined by equations (1.11) and (1.17). In this model, $\mathbf{v}(\mathbf{x},t)$ is a known velocity profile applied on the boundary, Γ_m, which is now assumed to be fixed. This time-dependent velocity profile causes the flow through the inlet and outlet valves.

Model No. 4

In this model, the problem is reduced to an initial-value problem. No moving boundary or boundary velocity are present in this formulation, and the forcing function is assumed to be the initial velocity field. At time $t = 0$, the blood flow is being given an impulse through the initial velocity field $\mathbf{u}_0(\mathbf{x})$, which causes blood to be discharged through the outlet valve. Obviously, this model is quite unrealistic, but we shall take advantage of its simplicity to develop the optimal control formulation of the simplest relevant problem. The governing equations for this model are as follows

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{t=0} = \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega \\ \mathbf{u}|_{\Gamma_f} = 0; \quad \boldsymbol{\tau}|_{\Gamma_m} = 0; \quad 0 < t \le T \\ \mathbf{u}|_{\Gamma_i} = 0; \quad \boldsymbol{\tau}|_{\Gamma_o} = 0; \quad 0 < t \le T \end{cases} \tag{2.5}$$

in which $\mathbf{u}_0(\mathbf{x})$ is the initial velocity distribution applied to the system at time $t = 0$, which should satisfy the continuity equation. This initial velocity distribution is assumed to cause the flow through the outlet valve during the period of the flow.

2.2 Formulation of an Appropriate Cost Functional

In this section we present a cost functional, which will be used in the following sections, in the formulation of the optimal control problem.

An optimization problem, in general, requires the specification of a "target" or "desired state" for the system as well as a suitable optimization criterion, to be determined from physical considerations. As briefly discussed in section 1.5, a cost functional is the expression used in optimal control terminology, which provides a measure of those parameters which should be minimized inside the flow domain. In view of the experimental results discussed in section 1.4, an optimization criterion appropriate to artificial heart design should probably include parameters such as shear stress, vorticity and residence time of blood in the heart, as it is known that regions with excessively high stresses and recirculation should be avoided. Since the control variable $\mathbf{v}(\mathbf{x},t)$ is a measure of the energy spent on the system, it may also be included in the criterion. In addition, the solution may be required to satisfy certain constraints. In the case of cardiac assist devices, the desired blood flow rate, averaged over a cycle, can be viewed as a constraint.

To introduce the constraint of desired flow rate, two approaches are considered here. In a less realistic approach, we assume that a desired velocity field, $\mathbf{u}^d(\mathbf{x},t)$ is known. The integral of this desired velocity over Γ_m, would produce the desired flow rate as a function of time. Then, it is necessary to minimize the difference between $\mathbf{u}^d$ and the velocity produced by the device. Including all the above requirements would result in the following definition for the cost functional

$$J(\mathbf{v}) = \frac{1}{2}\int_I\int_\Omega \{\alpha_c|\nabla\times\mathbf{u}(\mathbf{x},t)|^2 + \alpha_g|\nabla\mathbf{u}(\mathbf{x},t)|^2 + \alpha_d|\mathbf{u}(\mathbf{x},t)-\mathbf{u}^d(\mathbf{x},t)|^2 + \alpha_i|\mathbf{v}(\mathbf{x})|^2\}d\Omega dt \qquad (2.6)$$

where $\mathbf{u}$ is a velocity produced by the device corresponding to the control $\mathbf{v}$ and I is the period of the flow.

A more realistic and less restrictive criterion can be formulated by using the time-

dependent, "desired flow rate", Q^d, produced by the device as

$$J(\mathbf{v}) = \frac{1}{2}\int_I\int_\Omega \{\alpha_c|\nabla \times \mathbf{u}(\mathbf{x},t)|^2 + \alpha_g|\nabla \mathbf{u}(\mathbf{x},t)|^2\}d\Omega dt + \frac{1}{2}\int_I \alpha_d\{|Q(t) - Q^d(t)|^2\}dt$$
$$+ \frac{1}{2}\int_I\int_{\Gamma_m} \alpha_v|\mathbf{v}(\mathbf{x},t)|^2 ds dt, \qquad (2.7)$$

where $Q(t) = \int_{\Gamma_m} \mathbf{u}(\mathbf{x},t)ds$. The requirement of minimizing the mean squared $|Q - Q^d|$ over the period of the flow introduces the constraint of a desired flow rate.

In the above expressions, $\nabla \times \mathbf{u}$ and $\nabla \mathbf{u}$ denote the curl and the gradient of the velocity vector, $\mathbf{u}(\mathbf{x},t)$, respectively, and α_c, α_g, α_d and α_v are positive constant weights. By choosing different values for the weights, one can assign different importance to the various components of (2.6) and (2.7).

2.3 Optimal Control Using Boundary Velocity

The main goal of the present research is the application of optimal control theory to artificial hearts in an attempt to minimize the flow-related problems. In order to achieve this optimization, one should define a suitable cost functional and develop necessary conditions of optimality for blood flow and construct an efficient algorithm for the numerical solution of the problem.

To accomplish the above objective, it would be more appropriate to model the actions of moving boundaries (diaphragm and valves) and of the cardiovascular system. However, in order to simplify the problem numerically, we consider an idealization of an isolated artificial heart and demonstrate how optimal control can effectively modify the flow pattern, by attaining goals prescribed by a performance functional. A realistic control parameter for several popular designs of artificial hearts and ventricular assist devices would be the oscillatory motion of a pusher plate or diaphragm. For mathematical and numerical simplicity, we model this motion by a time-dependent boundary velocity distribution over the boundary occupied by the diaphragm at its equilibrium position (model No. 3). The time-dependent boundary velocity, $\mathbf{v}(\mathbf{x},t)$, is applied on some part

of the boundary, Γ_m, of the physical domain, which causes the blood flow inside the domain.

It is clear that different boundary velocity profiles would result in different velocity fields inside the domain, each producing different distributions of shear stresses and vorticity. We are interested in finding such a boundary velocity that would minimize the cost functional (2.7). This boundary velocity and the resulting velocity field inside the domain are the optimal control, $\mathbf{v}^o$, and the optimal velocity, $\mathbf{u}^o$, respectively. The formulation of the control problem of this model will contain all the essential aspects of an optimal control problem.

2.3.1 Statement of the Boundary Control Problem

In this section we proceed with a formal statement of the optimal control problem of the Navier-Stokes equations with the control on the boundary. The governing equations of model No. 3 are repeated here from section 2.1 for the purpose of convenience. The discussion and derivations will be represented for the general three-dimensional problem, which could be easily applied to two-dimensional problems.

The time-dependent, three-dimensional flow of an incompressible, Newtonian fluid in the domain Ω is described by the system of the momentum (Navier-Stokes) and the continuity equations,

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{t=0} = \mathbf{u}(\mathbf{x},0) = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{\Gamma_f} = 0; \quad \mathbf{u}|_{\Gamma_m} = \mathbf{v}(\mathbf{x},t) \\ \begin{cases} \mathbf{u}|_{\Gamma_i} = 0; \quad \tau|_{\Gamma_o} = 0; \quad 0 < t \le T_1 \\ \tau|_{\Gamma_i} = 0; \quad \mathbf{u}|_{\Gamma_o} = 0; \quad T_1 < t \le T \end{cases} \end{cases} \tag{2.8}$$

where $\mathbf{x} = (x_1, x_2, x_3)$ denotes the coordinates, $\mathbf{u} = (u_1, u_2, u_3)$ is the velocity vector, p is the pressure, ν is the coefficient of kinematic viscosity, ρ is the density and $\mathbf{f} = (f_1, f_2, f_3)$

is the body force per unit volume and τ is the traction vector. In equations (2.8), Ω represents the domain occupied by the fluid, which is assumed to be a three-dimensional open bounded set in R^3, whose boundary is denoted by $\Gamma = \Gamma_m \cup \Gamma_f \cup \Gamma_i \cup \Gamma_o$. Γ_m is the part of the boundary which has the prescribed velocity $\mathbf{v}$, Γ_f is the part with zero velocity, Γ_i is the inlet and Γ_o the outlet boundary of the domain, n is the unit normal vector and $I = [0, T]$ is a time interval and T_1 is the end of systolic and beginning of diastolic cycle. The control variable is the boundary velocity, $\mathbf{v}(\mathbf{x}, t)$, in equation (2.8), applied over the boundary Γ_m.

Our objective is to determine a control variable $\mathbf{v}(\mathbf{x}, t) \in U_{ad}$ (where the set of admissible controls, U_{ad}, is a closed convex set in $L_2((0, T) \times \Gamma_m)$, which would produce the optimal state that minimizes the cost functional in equation (2.7). The weight coefficients α_c, α_g, α_d and α_v in equation (2.7) may be adjusted according to the perceived importance of the various components of the cost functional. The pair $(\mathbf{v}^o, \mathbf{u}^o)$ satisfying these requirements is called the optimal pair.

The existence of an optimal control, namely of at least one element $\mathbf{v}^o \in U_{ad}$ such that $J(\mathbf{v}^o) \leq J(\mathbf{v})$, for all $\mathbf{v} \in U_{ad}$, has been proved by Ahmed (1992) (with a slightly different cost functional). However, the question of uniqueness of optimal control has not yet been resolved. In general, because the mapping $\mathbf{v} \rightarrow \mathbf{u}_\mathbf{v}$ is nonlinear, $J(\mathbf{v})$ is nonconvex, and uniqueness cannot be guaranteed (see Abergel and Temam, 1990).

The necessary conditions of optimality for a similar problem were derived by Ahmed (1992). In that paper, the problem (2.8) was cast on the space of solenoidal vector fields and the pressure was removed by applying the orthogonal projection operator on both sides of equation (2.8). However, in practical applications, it is more convenient to maintain the pressure and to derive the necessary conditions of optimality in a form that is more suitable for numerical implementation. The formal necessary conditions of optimality, corresponding to the present cost functional, are outlined below, while derivation details are presented in Appendix A.

2.3.2 Necessary Conditions of Optimality

Consider the system of equations (2.8). Assume that the integrand, g, of the cost functional in (2.7) is once continuously Gateaux differentiable, and denote the Gateaux derivatives of g with respect to the state $\mathbf{u}$ and the control $\mathbf{v}$ by $\mathbf{G}^1$ and $\mathbf{G}^2$, respectively. Then, in order that the pair $(\mathbf{v}^o, \mathbf{u}^o)$ be the optimal pair, it is necessary that there exists a $\mathbf{z}^o(\mathbf{x}, t)$ such that the triplet $(\mathbf{v}^o, \mathbf{u}^o, \mathbf{z}^o)$ satisfies the following equations and inequality

$$\begin{cases} \frac{\partial \mathbf{u}^o}{\partial t} + (\mathbf{u}^o \cdot \nabla)\mathbf{u}^o - \nu\nabla^2\mathbf{u}^o + \frac{1}{\rho}\nabla p^o = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla \cdot \mathbf{u}^o = 0 \quad \text{in } \Omega \\ \mathbf{u}^o|_{t=0} = \mathbf{u}^o(\mathbf{x},0) = 0 \quad \text{in } \Omega \\ \mathbf{u}^o|_{\Gamma_f} = 0; \quad \mathbf{u}^o|_{\Gamma_m} = \mathbf{v}^o(\mathbf{x},t) \\ \begin{cases} \mathbf{u}^o|_{\Gamma_s} = 0; \quad \tau|_{\Gamma_o} = 0; \quad 0 < t \le T_1 \\ \tau|_{\Gamma_s} = 0; \quad \mathbf{u}^o|_{\Gamma_o} = 0; \quad T_1 < t \le T \end{cases} \end{cases} \tag{2.9}$$

$$\begin{cases} -\frac{\partial \mathbf{z}^o}{\partial t} - (\mathbf{u}^o \cdot \nabla)\mathbf{z}^o + (\nabla \mathbf{u}^o)^T\mathbf{z}^o - \nu\nabla^2\mathbf{z}^o + \frac{1}{\rho}\nabla q_z^o = \mathbf{G}^1(\mathbf{u}^o, \mathbf{v}^o) \quad \text{in } \Omega \times (0,T] \\ \nabla \cdot \mathbf{z}^o = 0 \quad \text{in } \Omega \\ \mathbf{z}^o|_{t=T} = \mathbf{z}^o(\mathbf{x},T) = 0 \quad \text{in } \Omega \\ \mathbf{z}^o|_{\Gamma_f} = 0; \quad \mathbf{z}^o|_{\Gamma_m} = 0 \\ \begin{cases} \mathbf{z}^o|_{\Gamma_s} = 0; \quad \mathbf{z}^o|_{\Gamma_o} = 0; \quad 0 < t \le T_1 \\ \mathbf{z}^o|_{\Gamma_s} = 0; \quad \mathbf{z}^o|_{\Gamma_o} = 0; \quad T_1 < t \le T \end{cases} \end{cases} \tag{2.10}$$

$$J'(\mathbf{v}^o, \mathbf{v} - \mathbf{v}^o) = \int_I \int_{\Gamma_m} \left| \left\{ \left(-\nu(\mathbf{n} \cdot \nabla)\mathbf{z}^o + q_z^o \cdot \mathbf{n} + \mathbf{G}^2 \right) \cdot (\mathbf{v} - \mathbf{v}^o) \right\} \right| ds dt \ge 0; \quad \forall\, \mathbf{v} \in U_{ad}. \tag{2.11}$$

The system of equations (2.10) represents the "adjoint equations" along the optimal flow $\mathbf{u}^o$. The variables $\mathbf{z}$ and q_z are called the "adjoint state" and "adjoint pressure", respectively. It should be noted that the definition of the initial condition for the adjoint state $\mathbf{z}$ is at time $t = T$, in contrast to the definition of the initial condition in the Navier-Stokes equations. This is an important feature of the adjoint field. In the following we present a brief derivation of the above "necessary conditions of optimality".

The basic idea in deriving the necessary conditions of optimality is to require that

the cost functional attains its minimum for the control $\mathbf{v}^o$, which means that

$$J(\mathbf{v}^o) \leq J(\mathbf{v}) \quad \forall\, \mathbf{v} \in U_{ad}. \tag{2.12}$$

Since $\mathbf{v}^o + \epsilon(\mathbf{v} - \mathbf{v}^o) \in U_{ad}$ for $0 \leq \epsilon \leq 1$, then the following inequality holds

$$\lim_{\epsilon \to 0} \{\frac{J(\mathbf{v}^o + \epsilon(\mathbf{v} - \mathbf{v}^o)) - J(\mathbf{v}^o)}{\epsilon}\} \geq 0 \quad \forall\, \epsilon \in (0, 1). \tag{2.13}$$

The left-hand side of this inequality coincides with the definition of the Gateaux derivative of $J(\mathbf{v})$ with respect to $\mathbf{v}$ at $\mathbf{v}^o$ in the direction of $(\mathbf{v} - \mathbf{v}^o)$. (The Gateaux derivative of a functional J at v^o in the direction of v, denoted by $J'(v^o, v)$, is defined as

$$J'(v^o, v) = \lim_{\epsilon \to 0} \{\frac{J(v^o + \epsilon v) - J(v^o)}{\epsilon}\} \quad \forall\, \epsilon > 0). \tag{2.14}$$

The cost functional, $J(\mathbf{v})$, in its general form, is as follows

$$J(\mathbf{v}) = \int_I \int_\Omega g(\mathbf{u}, \mathbf{v}) d\Omega dt. \tag{2.15}$$

One may note that the functional $J(\mathbf{v})$ is a function of the control $\mathbf{v}$ and the state $\mathbf{u}$, which is in turn a function of $\mathbf{v}$. Carrying out the differentiation and using the chain rule, one would obtain

$$J'(\mathbf{v}, \mathbf{v} - \mathbf{v}^o) = \int_I \int_\Omega \mathbf{G}^1 \cdot \mathbf{w} d\Omega dt + \int_I \int_\Omega \mathbf{G}^2 \cdot (\mathbf{v} - \mathbf{v}^o) d\Omega dt \tag{2.16}$$

in which $\mathbf{G}^1$ and $\mathbf{G}^2$ are the Gateaux derivatives of the cost functional with respect to the state $\mathbf{u}$ and the control $\mathbf{v}$, respectively. $\mathbf{w}$ is the Gateaux derivative of the state $\mathbf{u}$ with respect to $\mathbf{v}$ in the direction of $(\mathbf{v} - \mathbf{v}^o)$.

In order to find an expression for $\mathbf{w}$, consider the optimal control $\mathbf{v}^o$ and an arbitrary control $\mathbf{v}^\epsilon = \mathbf{v}^o + \epsilon(\mathbf{v} - \mathbf{v}^o) \in U_{ad}$. The corresponding solutions to these controls are obtained from equations (2.8) by applying the boundary conditions at Γ_m as $\mathbf{u}|_{\Gamma_m} = \mathbf{v}^o(\mathbf{x}, t)$ and $\mathbf{u}|_{\Gamma_m} = \mathbf{v}^\epsilon(\mathbf{x}, t)$, respectively. We denote the resulting system of equations for the boundary controls $\mathbf{v}^o$ and $\mathbf{v}^\epsilon$ as N^o and N^ϵ, respectively. The velocity field and the pressure corresponding to the equations N^o and N^ϵ are denoted as $(\mathbf{u}^o, p^o)$ and $(\mathbf{u}^\epsilon, p^\epsilon)$, respectively. In the limit, $\epsilon \to 0$, $\mathbf{v}^\epsilon \to \mathbf{v}^o$ and it can be shown that

$$\mathbf{u}^\epsilon \to \mathbf{u}^o \quad , \quad \frac{\partial \mathbf{u}^\epsilon}{\partial x_i} \to \frac{\partial \mathbf{u}^o}{\partial x_i} \qquad i = 1, 2 \tag{2.17}$$

in some appropriate function spaces. If we subtract the equations N^o from the equations N^ϵ, divide the result by ϵ and let $\epsilon \to 0$, we obtain the following equations

$$\begin{cases} \frac{\partial \mathbf{w}}{\partial t} + (\mathbf{u}^o \cdot \nabla)\mathbf{w} + (\mathbf{w}.\nabla)\mathbf{u}^o - \nu\nabla^2\mathbf{w} + \frac{1}{\rho}\nabla q_w = 0 \quad \text{in } \Omega \times (0,T] \\ \nabla \cdot \mathbf{w} = 0 \quad \text{in } \Omega \\ \mathbf{w}|_{t=0} = \mathbf{w}(\mathbf{x},0) = 0 \quad \text{in } \Omega \\ \mathbf{w}|_{\Gamma_f} = 0; \quad \mathbf{w}|_{\Gamma_m} = \mathbf{v}(\mathbf{x},t) - \mathbf{v}^o(\mathbf{x},t) \\ \begin{cases} \mathbf{w}|_{\Gamma_i} = 0; \quad \frac{\partial \mathbf{w}}{\partial \mathbf{n}}|_{\Gamma_o} = 0; \quad 0 < t \le T_1 \\ \frac{\partial \mathbf{w}}{\partial \mathbf{n}}|_{\Gamma_i} = 0; \quad \mathbf{w}|_{\Gamma_o} = 0; \quad T_1 < t \le T \end{cases} \end{cases} \tag{2.18}$$

in which

$$q_w = \lim_{\epsilon \to 0}\{\frac{p^\epsilon - p^o}{\epsilon}\} = \lim_{\epsilon \to 0}\{\frac{p(\mathbf{v}^o + \epsilon(\mathbf{v} - \mathbf{v}^o)) - p(\mathbf{v}^o)}{\epsilon}\}; \quad \forall \epsilon \in (0,1) \tag{2.19}$$

and

$$\mathbf{w} = \lim_{\epsilon \to 0}\{\frac{\mathbf{u}^\epsilon - \mathbf{u}^o}{\epsilon}\} = \lim_{\epsilon \to 0}\{\frac{\mathbf{u}(\mathbf{v}^o + \epsilon(\mathbf{v} - \mathbf{v}^o)) - \mathbf{u}(\mathbf{v}^o)}{\epsilon}\}; \quad \forall \epsilon \in (0,1). \tag{2.20}$$

It is important to note that, in the above equation, $\mathbf{w}$ is the Gateaux derivative of $\mathbf{u}$ with respect to $\mathbf{v}$ in the direction of $(\mathbf{v} - \mathbf{v}^o)$, as it appeared in (2.16) and it is given by the solution of (2.18).

At this stage we consider the adjoint equations as stated in (2.10), in which the adjoint operator $\mathbf{z}$ can be directly obtained by multiplying both sides of (2.18) by $\mathbf{z}^o$, integrating over the domain and integrating by parts. The right-hand side of the first equation in (2.10) and the initial and boundary conditions are defined with consideration of the cost functional (2.7). Substituting in (2.16) for $\mathbf{G}^1$ from the first equation of (2.10) results in

$$\begin{aligned} J'(\mathbf{v}, \mathbf{v} - \mathbf{v}^o) = & \int_I \int_\Omega \{-\frac{\partial \mathbf{z}^o}{\partial t} - (\mathbf{u} \cdot \nabla)\mathbf{z}^o + (\nabla \mathbf{u})^T \mathbf{z}^o - \nu\nabla^2\mathbf{z}^o + \frac{1}{\rho}\nabla q_z\}.\mathbf{w} d\Omega dt \\ & + \int_I \int_\Omega \mathbf{G}^2 \cdot (\mathbf{v} - \mathbf{v}^o) d\Omega dt. \end{aligned} \tag{2.21}$$

The next step is to trade the derivatives in the above equality from $\mathbf{z}^o$ to $\mathbf{w}$. This can be performed by integration by parts using the following Green-Gauss formula

$$\int_\Omega w \frac{\partial z}{\partial x_i} d\Omega = -\int_\Omega \frac{\partial w}{\partial x_i} z d\Omega + \int_\Gamma n_i w z dS; \quad i = 1,2,3 \tag{2.22}$$

where $\mathbf{n} = n_i e_i$ is the normal to the boundary and e_i are unit vectors in the coordinate directions. Performing the required integrations, the final expression for the Gateaux derivative of the cost functional (inequality (2.11)) may be obtained (see Appendix A for details).

Finally, performing the required differentiations and omitting intermediate steps, one gets the following expressions for the Gateaux derivatives of the integrand of the cost functional

$$\begin{cases} \mathbf{G}^1 = \alpha_c\{[\nabla \times (\nabla \times \mathbf{u})]|_\Omega - [\mathbf{n} \times (\nabla \times \mathbf{u})]|_\Gamma\} + \alpha_g A(\mathbf{u}) + \alpha_d(Q - Q^d) \\ \mathbf{G}^2 = \alpha_v \mathbf{v}|_{\Gamma_m} \\ \text{where} \\ A(\mathbf{u}) = -\left(\frac{\partial^2 u_1}{\partial x_3^2} + \frac{\partial^2 u_1}{\partial x_2^2},\quad \frac{\partial^2 u_2}{\partial x_1^2} + \frac{\partial^2 u_2}{\partial x_3^2},\quad \frac{\partial^2 u_3}{\partial x_1^2} + \frac{\partial^2 u_3}{\partial x_2^2}\right)|_\Omega \\ \qquad + \left(\frac{\partial u_1}{\partial x_2} n_2 + \frac{\partial u_1}{\partial x_3} n_3,\quad \frac{\partial u_2}{\partial x_1} n_1 + \frac{\partial u_2}{\partial x_3} n_3,\quad \frac{\partial u_3}{\partial x_1} n_1 + \frac{\partial u_3}{\partial x_2} n_2\right)|_\Gamma. \end{cases} \tag{2.23}$$

The details of derivation of $\mathbf{G}^1$ and $\mathbf{G}^2$ are given in Appendix A.

2.3.3 Numerical Algorithm and Solution Procedure

In order to compute the optimal control-state pair $(\mathbf{v}^o, \mathbf{u}^o)$, we use a gradient-type algorithm, based on the necessary conditions of optimality stated in equations (2.9)-(2.11). This approach uses the gradient of the cost functional $J(\mathbf{v})$ with respect to the control, which leads to an iterative computation of the optimal control. This iterative algorithm is summarized in the following steps.

Step 1: The system of equations (2.9) is provided with a guess for the boundary velocity $\mathbf{v}$, which is the control in the first iteration.

Step 2: At the n-th step, with $\mathbf{v}_n$ known, the system of equations (2.9) is solved to obtain $(\mathbf{u}_n, p_n)$. These equations are solved by forward marching in time.

Step 3: With $(\mathbf{u}_n, \mathbf{v}_n)$ known, equation (2.10) is solved for the adjoint pair $(\mathbf{z}_n, q_n)$. These equations are solved by backward marching in time.

Step 4: With $(\mathbf{v}_n, \mathbf{u}_n, \mathbf{z}_n)$ known, the following gradient is computed at each discrete time step

$$J'(\mathbf{v}_n) = -\nu(\mathbf{n}\cdot\nabla)\mathbf{z}_n + q_z\cdot\mathbf{n} + \mathbf{G}^2(\mathbf{u}_n, \mathbf{v}_n) \quad \text{on} \quad \Gamma_m. \tag{2.24}$$

Step 5: A new control $\mathbf{v}_{n+1}(t)$ is calculated from the following equation

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \lambda J'(\mathbf{v}_n) \tag{2.25}$$

for a suitable choice of the descent parameter $\lambda > 0$, so that $J(\mathbf{v}_{n+1}) \leq J(\mathbf{v}_n)$.

Step 6: The norm $\|\mathbf{v}_{n+1}-\mathbf{v}_n\|$ is computed using an appropriate definition. If $\|\mathbf{v}_{n+1}-\mathbf{v}_n\| \leq \gamma$, for $\gamma > 0$ sufficiently small, the computation is stopped, and $\mathbf{v}_{n+1}(t)$ and the corresponding $\mathbf{u}_{n+1}(t)$ are taken as the approximate optimal control and the optimal state, respectively. Otherwise the steps 1 through 6 are repeated with $\mathbf{v}_{n+1}(t)$ as the new control.

In step 3 of the above algorithm, the right-hand side, $\mathbf{G}^1$, of the first equation in (2.10) is found by computing the Gateaux derivative of the cost functional with respect to the state variable $\mathbf{u}$ (see equation (2.23)). Then, the Gateaux derivative, $\mathbf{G}^2$, of the cost functional with respect to the control variable $\mathbf{v}$ is computed, which in turn is used in (2.24) together with the adjoint state $\mathbf{z}$ to find $J'(\mathbf{v}_n)$. In the numerical implementation of the algorithm, we have removed the term $\alpha_d(Q-Q^d)$ from $\mathbf{G}^1$ in equation (2.23) and have imposed it on the boundary conditions on inlet and outlet ports (Γ_i and Γ_o) in adjoint equations (2.10). Therefore, the boundary conditions for the adjoint state, $\mathbf{z}$, on these ports will be replaced by

$$\begin{cases} \mathbf{z}^o|_{\Gamma_i} = Q^o - Q^d; \quad \mathbf{z}^o|_{\Gamma_o} = 0; \quad 0 < t \leq T_1 \\ \mathbf{z}^o|_{\Gamma_i} = 0; \quad \mathbf{z}^o|_{\Gamma_o} = Q^0 - Q^d; \quad T_1 < t \leq T. \end{cases} \tag{2.26}$$

2.4 Optimal Control Using Initial Velocity

In this section we study the optimal control using an initial velocity distribution as control, model No. 4 in section 2.1. In this model, the forcing function is the initial

velocity field, which implies that the flow of blood is caused by a known initial velocity distribution, which is applied to the physical domain at time $t = 0$. It is clear that different initial velocities would result in different velocity fields inside the domain, each having different patterns for shear stresses and vorticity. Similarly to the previous case, we are interested in finding such an initial velocity field which would minimize the cost functional (2.6). All the essential aspects of derivation of the necessary conditions of optimality for this model are present in model No. 3, discussed in the previous section. Then, we shall outline the optimality conditions for this model based on the results of the previous section, while the details will be given in Appendix A.

2.4.1 Statement of the Control Problem

Consider the time-dependent, three-dimensional flow of an incompressible, viscous, Newtonian fluid in the domain Ω. Then the flow is described by the system of the momentum (Navier-Stokes) and the continuity equations

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \\ \mathbf{u}|_{t=0} = \mathbf{u}(\mathbf{x},0) = \mathbf{v}(\mathbf{x}) \quad \text{in } \Omega \\ \mathbf{u}|_{\Gamma_f} = 0; \quad \tau|_{\Gamma_m} = 0; \quad 0 < t \le T \\ \mathbf{u}|_{\Gamma_i} = 0; \quad \tau|_{\Gamma_o} = 0; \quad 0 < t \le T \end{cases} \tag{2.27}$$

where in the present problem $\mathbf{v} = (v_1, v_2, v_3)$ is the initial velocity distribution and the rest of the variables were defined previously.

In the present case we assume that the control variable is the initial velocity field $\mathbf{v}(\mathbf{x})$ in equation (2.27), which should of course satisfy the continuity equation. This requires that $\mathbf{v}$ should belong to a set of admissible controls, $U_{ad} \equiv \{\mathbf{v} : \mathbf{v} \in L_2(\Omega);\ \text{div}\,\mathbf{v} = 0\}$. The optimization problem is to find a control $\mathbf{v} \in U_{ad}$ which would minimize the cost functional (2.6).

2.4.2 Necessary Conditions of Optimality

In order to develop the necessary conditions of optimality, we consider the system of equations (2.27). Assume that the integrand g of the cost functional in (2.6) is once continuously Gateaux differentiable. As before, denote the Gateaux derivatives of g with respect to the state $\mathbf{u}$ and the control $\mathbf{v}$ by $\mathbf{G}^1$ and $\mathbf{G}^2$, respectively. Then, in order that the pair $(\mathbf{v}^o, \mathbf{u}^o)$ be an optimal pair, it is necessary that there exists a $\mathbf{z}^o(\mathbf{x}, t)$ such that the triplet $(\mathbf{v}^o, \mathbf{u}^o, \mathbf{z}^o)$ satisfies the following equations and inequalities

$$\begin{cases} \frac{\partial \mathbf{u}^o}{\partial t} + (\mathbf{u}^o \cdot \nabla)\mathbf{u}^o - \nu\nabla^2\mathbf{u}^o + \frac{1}{\rho}\nabla p^o = \mathbf{f} \quad \text{in } \Omega \times (0,T] \\ \nabla \cdot \mathbf{u}^o = 0 \quad \text{in } \Omega \\ \mathbf{u}^o|_{t=0} = \mathbf{u}^o(\mathbf{x},0) = \mathbf{v}^o(\mathbf{x}) \quad \text{in } \Omega \\ \mathbf{u}^o|_{\Gamma_f} = 0; \quad \tau|_{\Gamma_m} = 0; \quad 0 < t \leq T \\ \mathbf{u}^o|_{\Gamma_i} = 0; \quad \tau|_{\Gamma_o} = 0; \quad 0 < t \leq T \end{cases} \tag{2.28}$$

$$\begin{cases} -\frac{\partial \mathbf{z}^o}{\partial t} - (\mathbf{u} \cdot \nabla)\mathbf{z}^o + (\nabla \mathbf{u})^T\mathbf{z}^o - \nu\nabla^2\mathbf{z}^o + \frac{1}{\rho}\nabla q_z^o = \mathbf{G}^1(\mathbf{u}^o, \mathbf{v}^o) \quad \text{in } \Omega \times (0,T) \\ \nabla \cdot \mathbf{z}^o = 0 \quad \text{in } \Omega \\ \mathbf{z}^o|_{t=T} = \mathbf{z}^o(\mathbf{x},T) = 0 \quad \text{in } \Omega \\ \mathbf{z}^o|_{\Gamma_f} = 0; \quad \mathbf{z}^o|_{\Gamma_m} = 0; \quad 0 < t \leq T \mathbf{z}^o|_{\Gamma_i} = 0; \quad \frac{\partial \mathbf{z}^o}{\partial \mathbf{n}}|_{\Gamma_o} = 0; \quad 0 < t \leq T \end{cases} \tag{2.29}$$

$$J'(\mathbf{v}^o, \mathbf{v} - \mathbf{v}^o) = |(\mathbf{z}^o(0) + \int_I \mathbf{G}^2(\mathbf{u}^o, \mathbf{v}^o)dt, \mathbf{v} - \mathbf{v}^o)| \geq 0; \quad \forall\, \mathbf{v} \in U_{ad}. \tag{2.30}$$

The system of equations (2.29) represents the adjoint equations along the optimal flow $\mathbf{u}$ of the Navier-Stokes equations, and $\mathbf{z}$ and q_z are the adjoint state and adjoint pressure, respectively.

The details of the derivation of the above conditions of optimality are similar to those in the case of boundary control problem discussed in the previous section, except that the boundary integrals in Appendix A will no longer exist for this case. Regarding the Gateaux derivatives of the cost functional, a similar analysis as in Appendix A for

the cost functional (2.6) would result in

$$\begin{cases} \mathbf{G}^1 = \alpha_c\{\nabla \times (\nabla \times \mathbf{u})|_\Omega + \alpha_g A(\mathbf{u}) + \alpha_d(\mathbf{u} - \mathbf{u}^d)|_\Omega \\ \mathbf{G}^2 = \alpha_v \mathbf{v}|_\Omega \\ \text{where} \\ A(\mathbf{u}) = -\left(\frac{\partial^2 u_1}{\partial x_3^2} + \frac{\partial^2 u_1}{\partial x_2^2}, \quad \frac{\partial^2 u_2}{\partial x_1^2} + \frac{\partial^2 u_2}{\partial x_3^2}, \quad \frac{\partial^2 u_3}{\partial x_1^2} + \frac{\partial^2 u_3}{\partial x_2^2}\right)|_\Omega. \end{cases} \tag{2.31}$$

2.4.3 Numerical Algorithm and Solution Procedure

Similarly to the case of boundary control problem, in order to compute the optimal control-state pair $(\mathbf{v}^o, \mathbf{u}^o)$, we use a gradient-type algorithm based on the above necessary conditions of optimality. This iterative algorithm is summarized in the following steps.

Step 1: The system of equations (2.28) is provided with a guess for the initial velocity distribution $\mathbf{v}_0$, which is the control in first iteration.

Step 2: At the n-th step, with $\mathbf{v}_n$ known, the system of equations (2.28) is solved to obtain $(\mathbf{u}_n, p_n)$. These equations are solved by forward marching in time.

Step 3: With $(\mathbf{u}_n, \mathbf{v}_n)$ known, equation (2.29) is solved for the adjoint pair $(\mathbf{z}_n, q_n)$. These equations are solved by backward marching in time.

Step 4: With $(\mathbf{v}_n, \mathbf{u}_n, \mathbf{z}_n)$ known, the following gradient is computed at the end of time integration

$$J'(\mathbf{v}_n) = \mathbf{z}_n(0) + \int_I \mathbf{G}^2(\mathbf{u}_n(t), \mathbf{v}_n)dt \quad \text{in } \Omega. \tag{2.32}$$

Step 5: A new control $\mathbf{v}_{n+1}$ is calculated from the following equation

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \lambda J'(\mathbf{v}_n) \tag{2.33}$$

for a suitable choice of the descent parameter $\lambda > 0$, so that $J(\mathbf{v}_{n+1}) \leq J(\mathbf{v}_n)$.

Step 6: The norm $\|\mathbf{v}_{n+1} - \mathbf{v}_n\|$ is computed using an appropriate definition. If $\|\mathbf{v}_{n+1} - \mathbf{v}_n\| \leq \gamma$, for $\gamma > 0$ sufficiently small, the computation is stopped, and $\mathbf{v}_{n+1}$ and the

corresponding $\mathbf{u}_{n+1}(t)$ are taken as the approximate optimal control and the optimal state, respectively. Otherwise the steps 1 through 6 are repeated with $\mathbf{v}_{n+1}$ as the new control.

Chapter 3

Finite Element Formulation of the Control Problem

3.1 Introduction

The solution of the set of the Navier-Stokes equations and the corresponding adjoint equations described in Chapter 2 can only be achieved numerically. The finite element method appears to be most suitable for the solution of such equations, because it can deal effectively with complex geometries and boundary conditions, which are both present in our problem. Hence, this method will be employed in the numerical analysis of these equations.

In this section we shall demonstrate the discretization of the Navier-Stokes and corresponding adjoint equations by the Galerkin finite element method. For this purpose the velocity-pressure formulation of the Navier-Stokes equations (see section 1.8.1) in indicial notation will be used. The penalty function formulation of the Navier-Stokes equations will be used in our numerical code. Then, the finite element discretization of this formulation (see section 1.8.2) will also be presented. The flow will be considered to be three-dimensional, time-dependent, incompressible, laminar and Newtonian.

The finite element method replaces the continuous physical domain by a number of simple subdomains, called finite elements. For one-dimensional domains these subdomains are intervals, for two-dimensional domains they are triangles or quadrilaterals, and for three-dimensional domains they are tetrahedrons or hexahedrons. Within each element the solution of the governing equations is approximated by interpolating polynomials. Finally, the discrete representation of the whole domain is obtained through assemblage of the element equations. This assembly procedure is performed in such a way that interelement continuity of the nodal unknowns is met.

3.2 Finite Element Discretization of the Navier-Stokes Equations

The Navier-Stokes and continuity equations are first written for each individual finite element as follows

$$\begin{gathered} \rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} - \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{\partial p}{\partial x_i} = \rho f_i \\ \frac{\partial u_i}{\partial x_i} = 0; \quad i,j = 1,2,3. \end{gathered} \tag{3.1}$$

Over a typical element, Ω^e, the (continuous) velocity components and pressure (u_i and p, respectively) are approximated as

$$\begin{gathered} u_i^e(\mathbf{x},t) = \sum_{n=1}^{N} U_i^n(t)\Phi^n(\mathbf{x}) = \Phi^T(\mathbf{x})U_i(t); \quad i = 1,2,3 \\ p^e(\mathbf{x},t) = \sum_{m=1}^{M} P^m(t)\Psi^m(\mathbf{x}) = \Psi^T(\mathbf{x})P(t) \end{gathered} \tag{3.2}$$

where Φ and Ψ are column vectors of interpolating functions for velocity components and pressure, respectively, and the superscript T denotes their transposes. $\mathbf{x} = (x_1, x_2, x_3)$ is a generic point in the domain. U_i are N nodal unknowns of the velocity components, u_i, within the element and P^m are M nodal unknowns of the pressure. As can be seen from the above expressions, the interpolating functions for velocity and pressure do not belong to the same class of functions because they are not of the same order. We shall consider this in more details in Appendix B.

Because the expressions in equations (3.2) are approximations to the solution of equations (3.1), substituting them in those equations would produce errors ("residuals") in (3.1), rather than exactly satisfying them. These residuals are presented in the form

$$\begin{aligned} L_i(\Phi, \Psi, \mathbf{U}, P) = R_i; \quad i = 1,2,3 \quad \text{for momentum} \\ L_4(\Psi, \mathbf{U}) = R_4 \quad \text{for continuity} \end{aligned} \tag{3.3}$$

where $\mathbf{U} = (U_1, U_2, U_3)^T$ is the vector of nodal values of the velocity vector.

The basic premise in the formulation of (3.1) is to require that these residuals be zero. This can be performed in a weighted form by making the residuals to be orthogonal to the corresponding interpolation functions in a proper vector space, namely

$$\begin{aligned} (R_i, \Phi) = \int_\Omega R_i \cdot \Phi d\Omega = 0; \quad i = 1,2,3 \\ (R_4, \Psi) = \int_\Omega R_4 \cdot \Psi d\Omega = 0. \end{aligned} \tag{3.4}$$

Multiplying both sides of the first equation in (3.1) by the weight function, w, and integrating over an element area produces integral equations of the form

$$\begin{aligned} \int_\Omega w \left[\rho \frac{\partial u_i}{\partial t} + \rho(u_1 \frac{\partial u_i}{\partial x_1} + u_2 \frac{\partial u_i}{\partial x_2} + u_3 \frac{\partial u_i}{\partial x_3}) - \mu \frac{\partial}{\partial x_j}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) + \frac{\partial p}{\partial x_i}\right] d\Omega \\ = \int_\Omega w \rho f_i d\Omega. \end{aligned} \tag{3.5}$$

Similarly, the second equation in (3.1) can be integrated as

$$\int_\Omega w [\frac{\partial u_i}{\partial x_i}] d\Omega = 0. \tag{3.6}$$

The presence of second-order derivatives with respect to velocity components requires that the approximating function $\Phi \in C^2(\Omega)$ ($C^2(\Omega)$ is the set of functions on Ω that have two continuous derivatives) and $w \in C^0(\Omega)$. The so-called "weak variational formulation" has been introduced to trade the differentiation from u_i to w once. This can be done by virtue of the following Green-Gauss formula

$$\int_\Omega w \frac{\partial \Phi}{\partial x_i} d\Omega = -\int_\Omega \frac{\partial w}{\partial x_i} \Phi d\Omega + \int_\Gamma n_i w \Phi ds; \quad i = 1,2,3 \tag{3.7}$$

where $\mathbf{n} = n_i e_i$ is the normal to the element boundary and e_i are unit vectors in the coordinate directions.

Integration by parts, applied to the x component of equation (3.5) and use of (3.6) will give rise to the following weak variational form of the x component of the momentum equation in (3.1) and the continuity equation

$$\begin{aligned}
&\int_\Omega \rho w \frac{\partial u_1}{\partial t} d\Omega + \int_\Omega \rho w (u_1 \frac{\partial u_1}{\partial x_1}) d\Omega + \int_\Omega \rho w (u_2 \frac{\partial u_1}{\partial x_2}) d\Omega + \int_\Omega \rho w (u_3 \frac{\partial u_1}{\partial x_3}) d\Omega \\
&+2\mu \int_\Omega \frac{\partial w}{\partial x_1} \frac{\partial u_1}{\partial x_1} d\Omega + \mu \int_\Omega \frac{\partial w}{\partial x_2} \frac{\partial u_1}{\partial x_2} d\Omega + \mu \int_\Omega \frac{\partial w}{\partial x_2} \frac{\partial u_2}{\partial x_1} d\Omega \\
&+\mu \int_\Omega \frac{\partial w}{\partial x_3} \frac{\partial u_1}{\partial x_3} d\Omega + \mu \int_\Omega \frac{\partial w}{\partial x_3} \frac{\partial u_3}{\partial x_1} d\Omega - \int_\Omega \frac{\partial w}{\partial x_1} p d\Omega \\
&= \int_\Omega \rho f_1 w d\Omega + 2\mu \int_\Gamma w n_1 \frac{\partial u_1}{\partial x_1} ds + \mu \int_\Gamma w n_2 \frac{\partial u_1}{\partial x_2} ds + \mu \int_\Gamma w n_2 \frac{\partial u_2}{\partial x_1} ds \\
&-\int_\Gamma w n_1 p d\Omega + \mu \int_\Gamma w n_3 \frac{\partial u_1}{\partial x_3} ds + \mu \int_\Gamma w n_3 \frac{\partial u_3}{\partial x_1} ds
\end{aligned} \tag{3.8}$$

$$\int_\Omega w [\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}] d\Omega = 0.$$

Note that, in the momentum equation, the derivative in pressure has also been traded. If the weighting function, w, is chosen to be the same as the approximating function, Φ, the resulting formulation is called "Galerkin finite element model". By doing so and substituting (3.2) in (3.8), one gets the final Galerkin finite element model equivalent to equations (3.1) and (3.2), as

$$\begin{aligned}
&\left\{\int_\Omega \rho \Phi \Phi^T d\Omega\right\} \frac{\partial U_1}{\partial t} + \left\{\int_\Omega \rho \Phi (u_1 \frac{\partial \Phi^T}{\partial x_1} + u_2 \frac{\partial \Phi^T}{\partial x_2} + u_3 \frac{\partial \Phi^T}{\partial x_3}) d\Omega\right\} U_1 \\
&+\left\{\int_\Omega 2\mu \frac{\partial \Phi}{\partial x_1} \frac{\partial \Phi^T}{\partial x_1} d\Omega\right\} U_1 + \left\{\int_\Omega \mu \frac{\partial \Phi}{\partial x_2} \frac{\partial \Phi^T}{\partial x_2} d\Omega\right\} U_1 + \left\{\int_\Omega \mu \frac{\partial \Phi}{\partial x_3} \frac{\partial \Phi^T}{\partial x_3} d\Omega\right\} U_1 \\
&+\left\{\int_\Omega \mu \frac{\partial \Phi}{\partial x_2} \frac{\partial \Phi^T}{\partial x_1} d\Omega\right\} U_2 + \left\{\int_\Omega \mu \frac{\partial \Phi}{\partial x_3} \frac{\partial \Phi^T}{\partial x_1} d\Omega\right\} U_3 \\
&-\left\{\int_\Omega \frac{\partial \Phi}{\partial x_1} \Psi^T d\Omega\right\} P = \int_\Omega \rho f_1 \Phi d\Omega + \int_\Gamma \tau_1 \Phi ds
\end{aligned} \tag{3.9}$$

$$\left\{\int_\Omega \Psi \frac{\partial \Phi^T}{\partial x_1} d\Omega\right\} U_1 + \left\{\int_\Omega \Psi \frac{\partial \Phi^T}{\partial x_2} d\Omega\right\} U_2 + \left\{\int_\Omega \Psi \frac{\partial \Phi^T}{\partial x_3} d\Omega\right\} U_3 = 0$$

where

$$\tau_1 = \mu \left\{ n_1 (2\frac{\partial u_1}{\partial x_1}) + n_2 (\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}) + n_3 (\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}) \right\} - n_1 p \tag{3.10}$$

is the x component of the traction vector τ_i, defined by equations (1.11) and (1.17). Note that, in the continuity equation, the weighting function has been chosen as $w = \Psi$. The same procedure can be applied to other components of equation (3.5) to obtain similar expressions for the y and z components of the momentum equation.

One can combine the three components of the momentum equation and the conti-

nuity equation to obtain one single matrix equation, as

$$\begin{bmatrix} M & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & M & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{U}_1 \\ \dot{U}_2 \\ \dot{U}_3 \\ \dot{P} \end{bmatrix} + \begin{bmatrix} \sum N_i(u_i) & 0 & 0 & 0 \\ 0 & \sum N_i(u_i) & 0 & 0 \\ 0 & 0 & \sum N_i(u_i) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ P \end{bmatrix}$$

$$+ \begin{bmatrix} 2K_{11} + K_{22} + K_{33} & K_{12} & K_{13} & -C_1 \\ K_{21} & K_{11} + 2K_{22} + K_{33} & K_{23} & -C_2 \\ K_{31} & K_{32} & K_{11} + K_{22} + 2K_{33} & -C_3 \\ -C_1^T & -C_2^T & -C_3^T & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ P \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ 0 \end{bmatrix} \tag{3.11}$$

where the coefficients are defined as follows

$$M = \int_\Omega \rho \Phi \Phi^T d\Omega \tag{3.12}$$

$$N_i(u_i) = \int_\Omega \rho u_i \Phi \frac{\partial \Phi^T}{\partial x_i} d\Omega \tag{3.13}$$

$$K_{ij} = \int_\Omega \mu \frac{\partial \Phi}{\partial x_j} \frac{\partial \Phi^T}{\partial x_i} d\Omega \tag{3.14}$$

$$C_i = \int_\Omega \frac{\partial \Phi}{\partial x_i} \Psi^T d\Omega \tag{3.15}$$

$$F_i = \int_\Gamma \tau_i \Phi ds + \int_\Omega \rho f_i \Phi d\Omega \tag{3.16}$$

and τ_i is defined as

$$\tau_i = \sigma_{ij} n_j \tag{3.17}$$

where σ_{ij} is the stress tensor defined in (1.14). As mentioned earlier, the above equations have been derived for one element and the matrices are called "element matrices". After assemblage for all elements in the solution domain, the global set of equations can be obtained. Solving the global system of equations, after imposing boundary conditions, would result in the set of nodal values of the velocity components and pressure.

3.3 Finite Element Discretization of the Adjoint Equations

The optimal control problem described in the previous Chapter, requires that the adjoint system of equations corresponding to the Navier-Stokes equations be solved at each control iteration. Similarly in the case of the Navier-Stokes equations, the Galerkin finite element method is employed to discretize these equations. In this section we present the finite element formulation of the adjoint equations.

The adjoint equations (2.10) are rewritten here in standard indicial notation as

$$\begin{gathered} -\frac{\partial z_i}{\partial t} - u_j \frac{\partial z_i}{\partial x_j} + z_j \frac{\partial u_j}{\partial x_i} - \nu \frac{\partial}{\partial x_j}\left(\frac{\partial z_i}{\partial x_j} + \frac{\partial z_j}{\partial x_i}\right) + \frac{1}{\rho}\frac{\partial q_z}{\partial x_i} = G_i^1 \\ \frac{\partial z_i}{\partial x_i} = 0 \\ z_i(\mathbf{x}, T) = 0 \end{gathered} \tag{3.18}$$

where z_i are adjoint state components, q_z the adjoint pressure and G_i^1 are components of the Gateaux derivatives of the cost functional, $J(\mathbf{u}, \mathbf{v})$, with respect to the state variable $\mathbf{u} = (u_1, u_2, u_3)$ (see equation (2.23)). This system contains four equations with four unknowns, z_1, z_2, z_3 and q_z. It should be noted that the velocity components, u_1, u_2 and u_3, and their spatial derivatives are known quantities, obtained from the solution of the Navier-Stokes equations.

One may note that the above system of equations is a terminal value problem for the condition $\mathbf{z}(\mathbf{x}, T)$. This system of equations together with the Navier-Stokes equations, which are initial value equations, form a two-point boundary-value problem. To solve these equations efficiently, we reverse the time in adjoint equations by using $t \rightarrow T - t$. Hence the adjoint equations will have the following form

$$\begin{gathered} \frac{\partial z_i}{\partial t} - u_j \frac{\partial z_i}{\partial x_j} + z_j \frac{\partial u_j}{\partial x_i} - \nu \frac{\partial}{\partial x_j}\left(\frac{\partial z_i}{\partial x_j} + \frac{\partial z_j}{\partial x_i}\right) + \frac{1}{\rho}\frac{\partial q_z}{\partial x_i} = G_i^1 \\ \frac{\partial z_i}{\partial x_i} = 0 \\ z_i(\mathbf{x}, 0) = 0 \end{gathered} \tag{3.19}$$

which is now converted to an initial-value problem. In the above equations, the value of the adjoint state in the first time-step corresponds to the value of the velocity vector in

the final time-step and so on, which should be kept in mind during the numerical solution of the adjoint equations.

The procedure of discretizing the above equations is essentially the same as in the case of the Navier-Stokes equations. Over a typical element Ω^e, the (continuous) adjoint velocity components and the adjoint pressure (z_i and q_z, respectively) are approximated as follows

$$\begin{aligned} z_i^e(\mathbf{x},t) &= \sum_{n=1}^{N} Z_i^n(t)\Phi^n(\mathbf{x}) = \Phi^T(\mathbf{x})Z_i(t); \quad i=1,2,3 \\ q_z^e(\mathbf{x},t) &= \sum_{m=1}^{M} Q_z^m(t)\Psi^m(\mathbf{x}) = \Psi^T(\mathbf{x})Q_z(t) \end{aligned} \tag{3.20}$$

where Φ and Ψ are described in the previous section, and Z_i^n are N nodal unknowns of the adjoint state components, z_i, within the element, and Q_z^m are M nodal unknowns of the adjoint pressure.

Then multiplying both sides of equations (3.19) by the weight function, w, and integrating over the element area, integrating by parts using the Green-Gauss formula (3.7), using $w = \Phi$ in the adjoint momentum equations and $w = \Psi$ in the adjoint continuity equation, and combining the three components of the adjoint momentum and the adjoint continuity equation, one obtains the following single matrix equation

$$\begin{bmatrix} M & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & M & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \\ \dot{Q}_z \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} & 0 \\ B_{21} & B_{22} & B_{23} & 0 \\ B_{31} & B_{32} & B_{33} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Q_z \end{bmatrix}$$

$$+ \begin{bmatrix} 2K_{11}+K_{22}+K_{33} & K_{12} & K_{13} & -C_1 \\ K_{21} & K_{11}+2K_{22}+K_{33} & K_{23} & -C_2 \\ K_{31} & K_{32} & K_{11}+K_{22}+2K_{33} & -C_2 \\ -C_1^T & -C_2^T & -C_3^T & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Q_z \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ 0 \end{bmatrix} \tag{3.21}$$

in which

$$\begin{aligned} B_{11} &= \sum N_i(u_i) + A_{11}\left(\tfrac{\partial u_1}{\partial x_1}\right), \quad B_{12} = A_{12}\left(\tfrac{\partial u_2}{\partial x_1}\right), \quad B_{13} = A_{13}\left(\tfrac{\partial u_3}{\partial x_1}\right) \\ B_{21} &= A_{21}\left(\tfrac{\partial u_1}{\partial x_2}\right), \quad B_{22} = \sum N_i(u_i) + A_{22}\left(\tfrac{\partial u_2}{\partial x_2}\right), \quad B_{23} = A_{23}\left(\tfrac{\partial u_3}{\partial x_2}\right) \\ B_{31} &= A_{31}\left(\tfrac{\partial u_1}{\partial x_3}\right), \quad B_{32} = A_{32}\left(\tfrac{\partial u_2}{\partial x_3}\right), \quad B_{33} = \sum N_i(u_i) + A_{33}\left(\tfrac{\partial u_3}{\partial x_3}\right) \end{aligned} \tag{3.22}$$

where the coefficients M, $N_i(u_i)$, K_{ij} and C_i are defined by equations (3.12)-(3.15), respectively, and the new coefficient matrices, A_{ij} and E_i, are defined as

$$A_{ij} = \int_\Omega \rho \Phi \Phi^T \frac{\partial u_j}{\partial x_i} d\Omega \tag{3.23}$$

$$E_i = \int_\Gamma \tau_i \Phi ds + \int_\Omega \rho G_i^1 \Phi d\Omega \tag{3.24}$$

in which τ_i is defined by equation (3.17).

The two- and three-dimensional elements used in the numerical solution of the above equations and some other related issues are discussed in Appendix B.

3.4 Penalty Function Formulation

The derivation of the finite element equations in section 3.2 for the Navier-Stokes equations was based on the velocity-pressure formulation, in which both the velocity and the pressure unknowns are directly approximated and both are present in the global matrix of unknowns. In this section, we derive the finite element equations for the so-called penalty function formulation of the Navier-Stokes equations as described by equation (1.27) (see Gartling, 1987). Using the approximation in equation (1.25) for the original continuity constraint and applying the usual Galerkin weighted residuals method with the approximating functions (3.2) produces the following discretized form for the perturbed continuity equation

$$\left\{ \int_\Omega \Psi \frac{\partial \Phi^T}{\partial x_i} d\Omega \right\} U_i = - \left\{ \int_\Omega \epsilon \Psi \Psi^T d\Omega \right\} P \tag{3.25}$$

which in terms of a matrix equation will be

$$\mathbf{C}^T \mathbf{U} = -\epsilon M_p P \tag{3.26}$$

where the components of matrix $\mathbf{C} = (C_1, C_2, C_3)^T$ are defined by equation (3.15), $\mathbf{U} = (U_1, U_2, U_3)^T$ is the vector of unknown nodal velocities and

$$M_p = \int_\Omega \epsilon \Psi \Psi^T d\Omega. \tag{3.27}$$

By assuming that M_p can be inverted, the equation (3.26) can be written as

$$P = -\frac{1}{\epsilon} M_p^{-1} \mathbf{C}^T \mathbf{U}. \tag{3.28}$$

Inserting (3.28) into the x component of the momentum equation in (3.9) and repeating the same procedure for the other components, one gets the following matrix equation

$$\begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix} \begin{bmatrix} \dot{U}_1 \\ \dot{U}_2 \\ \dot{U}_3 \end{bmatrix} + \begin{bmatrix} \sum N_i(u_i) & 0 & 0 \\ 0 & \sum N_i(u_i) & 0 \\ 0 & 0 & \sum N_i(u_i) \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

$$+ \begin{bmatrix} 2K_{11} + K_{22} + K_{33} & K_{12} & K_{13} \\ K_{21} & K_{11} + 2K_{22} + K_{33} & K_{23} \\ K_{31} & K_{32} & K_{11} + K_{22} + 2K_{33} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

$$+ \begin{bmatrix} K_{p_{11}} & K_{p_{12}} & K_{p_{13}} \\ K_{p_{21}} & K_{p_{22}} & K_{p_{23}} \\ K_{p_{31}} & K_{p_{32}} & K_{p_{33}} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{3.29}$$

where $\mathbf{K}_p = K_{p_{ij}}$ is given by

$$\mathbf{K}_p = \frac{1}{\epsilon} \mathbf{C} M_p^{-1} \mathbf{C}^T. \tag{3.30}$$

The system of equations (3.29) represents the penalty form of the flow problem. The penalty parameter ϵ is a small constant, typically in the range of 10^{-6}-10^{-9}. After solving this equation, the pressure can be recovered from equation (3.28) with a known velocity.

It is important to note that in the above representation, the discretized (projected or weaker) continuity equation (equation (3.26) or equivalently (3.28)) was used as the penalty approximation to eliminate the pressure from the momentum equation. This formulation is commonly referred to as "consistent penalty method". Other formulations are possible, such as the so-called "reduced integration penalty method", in which the original perturbed continuity equation (1.25) is substituted into the original partial differential form of the momentum equation. Application of the Galerkin method to the resulting perturbed partial differential equations leads to a system of equations similar

to (3.29). These two formulations are in general different, hence their choice would affect the numerical implementation. For details on the reduced integration penalty method see Carey and Oden (1983).

The successful numerical implementation of the penalty method relies on the construction of the $\mathbf{K}_p$ matrix. Similar to the other finite element matrices in (3.11), this matrix should also be evaluated at the element level. We note in equation (3.30) that, to construct $\mathbf{K}_p$ efficiently, $\mathbf{C}$ and M_p^{-1} should be first constructed. The $\mathbf{C}$ matrix can be easily constructed and poses no difficulty. However, evaluation of M_p^{-1} requires that M_p be invertible at the element level, and this requirement restricts the choices for the basis function, Ψ, approximating the pressure. To be invertible at the element level, the function Ψ should be defined only within the element level, or, in other words, it should be discontinuous between consecutive elements.

In the following, we shall concentrate on the construction of $\mathbf{K}_p$ matrix for 9-node, two-dimensional elements and 27-node, three-dimensional elements. In the case of 9-node quadrilateral elements, two possible approximating functions for pressure are described in section B.1, from which only the second one (equation (B.3)) is discontinuous between elements. When this discontinuous function is used for pressure approximation, a 3×3 Gaussian integration rule must be used to evaluate $\mathbf{C}$ and M_p matrices. The resulting pressure mass matrix, M_p, is a 3×3 matrix, and it can be easily inverted by standard techniques to produce M_p^{-1}.

It should be noted that another discontinuous pressure approximation is possible for this quadrilateral element, which is a bilinear interpolation defined at the 2×2 Gauss points. However, this interpolation function will lead to a larger matrix for M_p (a full 4×4 matrix) and undesirable checkerboards (spurious oscillations) for the pressure (see Sani *et al.*, 1981).

In the case of 27-node, three-dimensional elements, two discontinuous pressure approximations are possible, as described by equations (B.7) and (B.8). The use of the

discontinuous linear approximation function (B.8), will lead to a 4×4 matrix for M_p, which, as in the two-dimensional case, can be easily inverted to produce M_p^{-1}. However, if the discontinuous trilinear approximation function (B.7) is used, the resulting M_p matrix will be a full 8×8 matrix which requires an expensive operation to invert, and the spurious checkerboards will be present in the pressure mode.

In the light of the above discussion, we shall use the discontinuous linear approximation functions (B.3) and (B.8) in our penalty function formulation for two- and three-dimensional elements, respectively.

Regarding the penalty function formulation for the adjoint equations (3.18), a similar procedure could be performed, which would result in a system of equations similar to (3.29).

3.5 Nondimensionalization

It is well known that the dimensionless formulation of engineering problems has considerable advantages over dimensional representations. In numerical studies of engineering problems, the relative importance of the various terms in the equations can be systematically studied by nondimensional formulations to identify the dominant terms. It can also reduce large differences in orders of magnitude between different terms in the equation and allows an estimate of the difficulty of the problem. In this section, we consider the nondimensional form of the Navier-Stokes equations (3.1).

In general, two characteristic scales are needed to scale the velocities and the lengths (physical coordinates). The velocities are scaled with respect to a characteristic speed U, and a characteristic length L is used to scale the physical coordinates, as follows

$$x_i^* = \frac{x_i}{L}; \qquad u_i^* = \frac{u_i}{U} \tag{3.31}$$

where superscript * denotes a dimensionless quantity. These two scales can be combined

to produce scales for time, pressure and body forces, as $\frac{L}{U}$, ρU^2 and $\frac{U^2}{L}$, respectively. Then

$$t^* = \frac{tU}{L}; \qquad p^* = \frac{p}{\rho U^2}; \qquad f_i^* = \frac{fL}{U^2}. \tag{3.32}$$

Using the relations (3.31) and (3.32), the dimensional momentum equation in (3.1) is transformed to the following dimensionless equation

$$\frac{\partial u_i^*}{\partial t^*} + u_j^* \frac{\partial u_i^*}{\partial x_j^*} - \frac{1}{Re} \frac{\partial}{\partial x_j^*} \left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} \right) + \frac{\partial p^*}{\partial x_i^*} = f_i^* \tag{3.33}$$

in which $Re = \frac{\rho UL}{\mu}$ is the Reynolds number.

An alternative dimensionless form of equation (3.33) is possible as

$$Re \left(\frac{\partial u_i^*}{\partial t^*} + u_j^* \frac{\partial u_i^*}{\partial x_j^*} \right) - \frac{\partial}{\partial x_j^*} \left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} \right) + \frac{\partial p^{**}}{\partial x_i^*} = Re f_i^* \tag{3.34}$$

where $p^{**} = \frac{L}{\mu U} p$. If equation (3.33) is used in the numerical simulation, then the input data for density and viscosity should be chosen as

$$\rho = 1; \qquad \mu = \frac{1}{Re} \tag{3.35}$$

and if equation (3.34) is used, then this requirement would be of the form

$$\rho = Re; \qquad \mu = 1. \tag{3.36}$$

Note that the pressure will have different interpretation in these two formulations. It is also noted that the form of the continuity equation does not change with nondimensionalization.

Regarding the adjoint equations, the same procedure will result in nondimensional forms similar to the above.

Chapter 4

Numerical Results

4.1 Introduction

The numerical algorithms described in sections 2.3 and 2.4 show that, at each optimization iteration, first the Navier-Stokes equations should be solved for the velocity field, based on which the corresponding adjoint equations should be subsequently solved. Therefore, the solution of the present problem requires a computer code, which would be able to solve both systems of equations for time-dependent boundary conditions.

Among the available commercial computer codes for solving the Navier-Stokes equations, the computer code FIDAP (FIDAP7.07, 1993) was tried to solve some preliminary examples. Although FIDAP, to some extent, accommodates time-dependent boundary conditions, it was found impractical in the simulation of the time-varying boundary conditions for two outlet valves in the domain shown in Figure 4.1. Moreover, neither FIDAP nor any other commercial computer code can easily be adopted for the solution of the adjoint equations. These reasons necessitated the development of our own specialized computer code, FLOPTIM, which is described in some detail in Appendix C.

4.2 Two-Dimensional Systems

4.2.1 Optimal Control Using Boundary Velocity

In this section the control problem corresponding to model No. 3 is studied numerically. The governing equations for this model and the mathematical formulation of the necessary conditions of optimality and the numerical algorithm were discussed in section 2.3.

In order to illustrate the optimization algorithm, three numerical examples are presented here. The physical domain used in these examples is a two-dimensional idealization of a prototype, moving diaphragm type, artificial heart, with outlet and inlet ports, and having the physical dimensions shown in Figure 4.1. The physical domain is divided into 868 nine-noded quadrilateral elements with 3621 nodes. The flow is caused by a boundary velocity applied on the control boundary Γ_m. In order to obtain volume flow rates comparable to those in a three-dimensional geometry, we assumed that the depth of the present two-dimensional flow was equal to the height, d_s, of the control boundary. Using data from Jin and Clark (1993), it was assumed that an average flow rate of Q_s=4.05 l/min was supplied by the system. With an average heart rate of 70 pulses per minute, one cycle of operation would be completed in 0.857 s with 0.300 s for the systole and 0.557 s for the diastole, respectively. This information, together with the height of the control boundary, d_s=8.0 cm, result in an average velocity of u_s=1.055 cm/s of the fluid at control boundary Γ_m during systole. The reference quantities, u_s and d_s, were used to non-dimensionalize all equations. Assuming a viscosity of $\nu=3.5\times10^{-2} cm^2/s$ for blood, the Reynolds number for this flow would be $\mathrm{Re}=\frac{u_s d_s}{\nu}$=241. The time scale was formed as the ratio of velocity and length scales with a value of t_s=7.584 s. The time domain was divided into 80 equal time intervals, which resulted in non-dimensionalized time steps of Δt=0.0014125. The problem was solved for one cycle of operation, equal to 0.857 s. It is recognized that the proper simulation of a periodic system, such as the operating artificial heart, would require the solution to extend over several cycles. Unfortunately,

example	α_c	α_g	α_d	α_v	CPU min.
1	0.0	0.0	1.0	0.0	1800
2	0.001	0.001	1.0	0.0	3100
3	0.004	0.004	1.0	0.0	2700

Table 4.1: Weight coefficients in the cost functional in examples 1-3 in two-dimensional boundary velocity control.

this was not practical, because of the enormous computational time required for this solution to be obtained (see Tables 4.1 and the results section 4.2 for typical CPU times for single-cycle solutions in two- and three-dimensional domains, respectively). It is assumed that the chamber is initially filled with fluid. During systole the outlet port is open while the inlet is closed and conversely for the diastole. The desired flow rate, Q^d, is assumed to have the time-dependent profile shown in Figure 4.2 (also nondimensionalized using the desired "average" flow rate, Q_s). The peak values and the profile were chosen to be close to those measured in a sac-type ventricular assist device (Jin and Clark, 1993).

The cost functional in these examples was the one described by equation (2.7), expressing an attempt to approach a desired flow rate while maintaining relatively low mean-squared shear stress and vorticity. The term corresponding to the boundary velocity has been neglected, as power consumption is not of concern within the present approximation. The weight coefficients in the cost functional in these examples as well as the required computational times are presented in Table 4.1.

The non-dimensionalized form of the Navier-Stokes and adjoint equations were solved by the finite element method. The penalty function formulation of the Navier-Stokes equations with $\mathbf{u}$ and p as unknowns was used (see section 1.8.2 and Girault and Raviart, 1986). Zero initial conditions for the Navier-Stokes equations were used to start the first control iteration. For each subsequent control iteration, the flow solution from the previous iteration was used as the initial condition for the Navier-Stokes equations

at that iteration. In view of the nonlinearity of the Navier-Stokes equations, the convergence of the flow solution was checked at each time step by evaluating the relative error. After solving the Navier-Stokes equations, the adjoint equations were solved for all time steps. The simulation was performed using a finite element package developed for this purpose, which uses the Galerkin approach in discretizing both the Navier-Stokes and adjoint equations. The discretized finite element equations were solved using a slightly modified version of an out-of-core equation solver (see Hasbani and Engelman, 1979). All computations were performed on an IBM AIX 3.2 workstation.

The first numerical example was performed with the objective to achieve a desired flow rate, regardless of the amount of shear stress and vorticity in the field. In the second example, shear stress and vorticity were also taken into account. The third example was performed with a higher importance given to shear stress and vorticity. The value of γ (convergence tolerance) was 0.01 in all examples. The descent parameter, λ, was chosen as 0.001 in example 1 and 0.0001 in examples 2 and 3. This value seems to be near optimal for examples 2 and 3 because the control iteration did not converge for higher values of λ, while the convergence was too slow for smaller values.

The flow rates achieved in these examples and the corresponding desired flow rate are plotted in Figure 4.2. The convergence rates and the value of $(Q - Q^d)$ are plotted in Figure 4.3 versus the number of the iterations. In Figure 4.4, the velocity vector and the streamline contours in example 1 are plotted for time steps 16 and 61, corresponding to peak systole and diastole, respectively. Similar field plots are presented in Figure 4.5 for example 3 . The optimal boundary velocity profiles (optimal controls) are shown in Figure 4.6 for a few time steps. Finally, the evolution of a particular control velocity profile (at time step 16) as a function of control iteration is presented in Figure 4.7 .

Our numerical experiments have indicated that the value of the descent parameter, λ, in equation (2.25) should be chosen carefully. In fact, for large values of λ, the Navier-Stokes equations do not converge. In linear-quadratic problems, an optimal descent parameter can be computed. Since our system is nonlinear with a quadratic cost

functional, it is not possible to choose a universal optimal descent parameter, but a value must be chosen optimally at every step. Because this proved to be time consuming, it seemed better to start the computation using as large a λ as possible and to reduce it whenever numerical divergence occurred.

The speed of convergence of the design scheme was found to be sensitive to the initial guess for the control $\mathbf{v}(\mathbf{x})$. Because the selection of a suitable non-zero initial guess could only be done by trial-and-error and would require an a priori knowledge of the solution, we found it more suitable to choose a zero initial guess for the control in all the examples.

Convergence of the design algorithm suffers from the presence of the second-order derivatives on the right-hand side of the adjoint equations in (2.10). The relative importance of these second-order terms is set by the values of the weight coefficients α_c and α_g in equation (2.7). Indeed, for large values of these coefficients the convergence of the algorithm became more difficult (both for the Navier-Stokes equations and the control iteration). Figure 4.3(a) shows that the solution in example 1 converged in 20 iterations, while this value increased to 36 in example 2 and to 31 in example 3.

In the first example, the goal was to match a realistic flow rate regardless of the amount of shear stress and vorticity produced by the resultant flow pattern. Figure 4.2 shows that, in this case, the differences between the desired and obtained flow rates were relatively small, with the obtained average flow rate being equal, within 10%, to the desired value. In order to understand the second and third examples, one has to consider that, in the absence of any constraint or desired flow rate specification, the optimal velocity field would be the trivial one, which obviously produces zero mean-squared stress and vorticity. When the cost functional accounts for the desired flow rate, shear stress and vorticity (examples 2 and 3), the optimal flow rate was expected to be lower than the desired one, because the algorithm tends to reduce the stress level by rearranging the flow distribution in the domain as well as by reducing the velocity amplitude. This expectation is confirmed by the results shown in Figure 4.2 for examples 2 and 3. The

average flow rates were about 45% and 65% lower than the specified desired flow rate in examples 2 and 3, respectively. This discrepancy poses no problem because it can be easily accommodated by specifying a higher desired flow rate. Figure 4.3(b) shows that, throughout the iterative process, the value of $(Q - Q^d)$ did not decrease in examples 2 and 3 as much as it did in example 1. It is interesting to note the difference in the shapes of the control velocity profiles in these examples (Figure 4.6). In the first example, the velocity profiles are nearly flat with sharp edges close to the walls, while in examples 2 and 3 these profiles are smoother. One might also note that the profiles in example 3 are smoother than the ones in example 2, which is attributed to the larger values of the weight coefficients α_c and α_g in example 3. This confirms our physical expectation that a smoother profile would produce lower velocity gradients in the regions close to the walls. Another interesting observation is that, when the stress level is minimized, the control velocity is highly skewed (Figures 4.6(b) and 4.6(c)), so that higher velocity amplitudes correspond to locations facing an open port.

Finally, concerning the evolution of the boundary control profiles in Figure 4.7, we observe that in all the examples the first profile suggested by the algorithm is flat. As the control iteration advances, the profile in example 1 reaches higher magnitudes but remains flat. On the other hand, in examples 2 and 3 the algorithm increases the magnitude of the control profiles while at the same time decreasing the values near the boundary and skewing them towards the open port.

4.2.2 Optimal Control Using Initial Velocity Distribution

In this section some numerical results for the control problem described in section 2.4 (model No. 4) are presented. The geometry used in the following examples and its discretization are the same as presented in the previous section in Figure 4.1. Also, the same average velocity of $u_s = 1.055\ cm/s$ and the same Reynolds number of 241 were assumed for the boundary Γ_m. The time scale of 7.584 s and the non-dimensionalized time steps of Δt=0.0014125 were used in accordance with their values in the previous

section. As discussed in section 2.1, this model is a rough approximation to the control problem and it is presented here only for the purpose of illustration of the method. In this model, it is assumed that the forcing function is the initial velocity distribution in the entire domain. In the context of the optimal control of this problem, this initial velocity distribution is taken as the control variable (see equation 2.27), and the optimization problem for this case consists of finding an initial velocity distribution, which causes the fluid to flow out of the domain with a velocity distribution as closely as possible to a specified, "desired" one and at the same time minimizes the mean-squared vorticity and shear stress. Due to the nature of the control variable in this problem, the flow starts from an initial velocity distribution and develops to a steady-state solution. We have assumed that this process happens while Γ_i is closed and Γ_o is open, with Γ_m being part of the solution domain, for the entire time interval. Our objective in this problem was to demonstrate the applicability of the optimization algorithm as presented in section 2.2. This objective can be illustrated by solving the optimization problem for any number of time steps; hence, we decided to solve the problem for only 5 time steps, to avoid the requirement of excessive CPU time.

The cost functional was chosen to be the same as described in section 2.2 by equation (2.6). As the optimization algorithm in section 2.4.3 and the cost functional in equation (2.6) suggest, a time-dependent desired velocity field, $\mathbf{u}^d(t)$, and an initial control, $\mathbf{v}(\mathbf{x})$, should be specified. In order to compare the results, all the following examples have been provided with the same desired velocity field. This time-dependent desired velocity field does not correspond to any real situation and is introduced only for the purpose of illustration of the optimization algorithm. It was obtained from the solution of the Navier-Stokes equation (2.27) with $\mathbf{u}|_{\Gamma_m} = u_s$ (a constant velocity profile corresponding to the average velocity during systole in the boundary control problem of the previous section) and $\mathbf{u}(\mathbf{x}, 0)$=0.0, over 5 time steps. The weight coefficients in the cost functional in these examples as well as the required computational times are presented in Table 4.2.

In example 1, two different initial controls, $\mathbf{v}_1(\mathbf{x})$ and $\mathbf{v}_2(\mathbf{x})$, were used. The initial velocity distribution $\mathbf{v}_1(\mathbf{x})$ corresponds to zero initial field, and $\mathbf{v}_2(\mathbf{x})$ was obtained from

example	α_c	α_y	α_d	α_v	CPU min.
1	0.0	0.0	1.0	0.0	550 with $\mathbf{v}_1$
2	0.01	0.01	1.0	0.0	310
3	0.05	0.05	1.0	0.0	380

Table 4.2: Weight coefficients in the cost functional in examples 1-3 in two-dimensional initial velocity control.

the solution of the Navier-Stokes equations and corresponds to the velocity field in the first step of the solution of these equations with $\mathbf{u}|_{\Gamma_m} = 0.5u_s$. Examples 2 and 3 were obtained with the same initial control $\mathbf{v}_2(\mathbf{x})$. One may note that the zero initial control is admissible because it satisfies the continuity equation. In all examples, the convergence criterion in the optimization algorithm was chosen to be $||\mathbf{v}_{n+1} - \mathbf{v}_n||$ as described in section 2.4 with γ=0.01 (convergence tolerance). Regarding the value of the descent parameter, λ, in equation (2.33), a few values in the range of 0.1 to 6 were tried for each example, among which, a value of γ=4.0 was chosen for example 1 and γ=1.0 for examples 2 and 3; these values seemed to be near optimal for the corresponding examples. In fact, as in the case of the boundary control problem in section 4.2.1, the control iteration did not converge for relatively large values of λ, while the convergence was too slow for relatively small values.

In Figure 4.8 the convergence rates and the value of $\mathbf{u} - \mathbf{u}^d$ in examples 1-3 are plotted versus the number of iterations. The resultant optimal velocity field in example 1 is compared with the desired velocity profiles in Figure 4.9 for both controls $\mathbf{v}_1(\mathbf{x})$ and $\mathbf{v}_2(\mathbf{x})$. All results correspond to the solution at time step 5. Similar results are presented for examples 2 and 3 with the control $\mathbf{v}_2(\mathbf{x})$ in Figures 4.10 and 4.11, respectively. In Figure 4.12, the velocity vector in examples 1, 2 and 3 are plotted for time step 5.

The algorithm converged in 22 iterations in example 1 for the control $\mathbf{v}_2(\mathbf{x})$, while this number increased to 34 and 42 in examples 2 and 3, respectively, which were provided

with the same initial control as example 1. This increase in the number of required iterations can be attributed to the lower value of descent parameter, λ, which is in turn due to the presence of the second order derivatives in the right-hand side of the adjoint equations (see equation (2.29). In example 1, with α_d=1.0 and $\alpha_c = \alpha_g$=0.0, the goal was to demonstrate the applicability of the optimization algorithm by matching a desired velocity distribution. The above values for the weight coefficients imply that our only concern was to obtain a velocity field as close to the desired velocity distribution as possible, regardless of the amount of shear stress and vorticity produced by the resultant flow pattern. As expected, the desired and optimal velocity field in this example are quite close, within 80% and 90 % for the controls $v_1(x)$ and $v_2(x)$, respectively (see Figure 4.9).

In examples 2 and 3, the optimal velocity field is not as close to the desired velocity as it was in example 1 (see Figures 4.10 and 4.11, respectively). This is due to the presence of the curl and gradient of the velocity field in examples 2 and 3, respectively. In other words, the algorithm in examples 2 and 3 tries to compromise between minimizing the difference of the desired and optimal velocities and minimizing the mean-square vorticity and shear stresses. Because the desired velocity field does not correspond to a minimum vorticity/shear stress field, the resultant optimal velocity field has a different profile, containing respectively lower levels of vorticity or shear stresses.

In the first example, the velocity profiles have sharp edges close to the walls (Figure 4.9), while in examples 2 and 3 these profiles are smoother. This is attributed to the non-zero values of the weight coefficients α_c and α_g in examples 2 and 3. Similarly to the case of the boundary velocity control, this confirms our physical expectation that a smoother profile would produce lower velocity gradients in the regions close to the walls. This result can also be observed in Figure 4.12 by comparing the flow patterns in examples 1, 2 and 3, especially at the outlet port of the domain.

4.3 Three-Dimensional Systems

In this section we present a numerical study of the three-dimensional boundary control problem corresponding to model No. 3. The discussion of the necessary conditions of optimality and the optimization algorithm in section 2.3 is quite general and is valid both for two- and three-dimensional domains.

Similarly to the two-dimensional case of this problem in section 4.2.1, three numerical examples were computed in this case. The physical domain, shown in Figure 4.13, is an approximate three-dimensional model of a sac-type artificial heart studied experimentally by Jin and Clark (1993). The physical dimensions, as well as the outlet and inlet ports (Γ_o and Γ_i, respectively) and the control boundary, Γ_m, are shown in Figure 4.13(b). In the finite element discretization of the problem, 27-noded bricks (see Figure B.2) were used, and the physical domain was divided into 3353 nodes with 326 27-noded elements. In this three-dimensional case, a boundary velocity is applied on the control boundary, Γ_m, which causes the blood flow through the domain. It was assumed that an average flow rate of Q_s=4.05 *l/min* was supplied by the system as reported by Jin and Clark (1993). The width of the control boundary, d_s=6.5 *cm*, was taken as the reference length for the domain. With an average heart rate of 70 pulses per minute, one cycle of operation would be completed in 0.857 *s* with 0.300 *s* for the systole and 0.557 *s* for the diastole, respectively. Therefore, the average flow rate of Q_s=4.05 *l/min* results in an average velocity of u_s=4.068 *cm/s* of the fluid at the control boundary Γ_m during the systole. The reference quantities, u_s and d_s, were used to non-dimensionalize all equations. Assuming a viscosity of $\nu=3.5\times10^{-2} cm^2/s$ for blood, the Reynolds number for this flow would be $\mathrm{Re}=\frac{u_s d_s}{\nu}$=756. The time scale was formed as the ratio of velocity and length scales with a value of t_s= 3.20 *s*. The time domain was divided into 20 equal time intervals, which resulted in non-dimensionalized time steps of Δt=0.0134. The problem was solved for one cycle of operation, equal to 0.857 *s*. As in the case of the two-dimensional problems in the previous section, this restriction was imposed in order to reduce the required computational time. It is assumed that the chamber is initially

filled with fluid. During systole the outlet port is open while the inlet is closed and conversely for the diastole. The desired flow rate, Q^d, is assumed to have the time-dependent profile shown in Figure 4.14 (also nondimensionalized using the desired "average" flow rate, Q_s). The peak values and the profile were chosen to be close to those measured in a sac-type ventricular assist device (Jin and Clark, 1993).

Similarly to the two-dimensional case in section 4.2.1, the non-dimensionalized form of the Navier-Stokes and adjoint equations were solved by the finite element method, with the use of the penalty function formulation for the Navier-Stokes equations with $\mathbf{u}$ and p as unknowns (see section 1.8.2 and Girault and Raviart, 1986). Zero initial conditions for the Navier-Stokes equations were used to start the first control iteration. For each subsequent control iteration, the flow solution from the previous iteration was used as the initial condition for the Navier-Stokes equations at that iteration. In view of the nonlinearity of the Navier-Stokes equations, the convergence of the flow solution was checked at each time step by evaluating the relative error as described in section C.7. After solving the Navier-Stokes equations, the adjoint equations were solved for all time steps. The simulation was performed using the three-dimensional finite element package FLOPTIM developed for this purpose, which uses the Galerkin approach in discretizing both the Navier-Stokes and the adjoint equations. The discretized finite element equations were solved using a slightly modified version of an out-of-core equation solver (see Hasbani and Engelman, 1979). All computations were performed on an IBM AIX 3.2 workstation.

Regarding the cost functional in these examples, the one described by equation (2.7) was used. The weight coefficients for the cost functional as well as the required computational times are presented in Table 4.3.

As illustrated in Table 4.3, the weight coefficients in these examples are the same as those in their counterparts in the two-dimensional case in section 4.2.1. The objective of the first example was to achieve a desired flow rate, regardless of the amount of shear stress and vorticity in the field. In the second example, shear stress and vorticity were

example	α_c	α_g	α_d	α_v	CPU min.
1	0.0	0.0	1.0	0.0	8800
2	0.001	0.001	1.0	0.0	9500
3	0.004	0.004	1.0	0.0	9900

Table 4.3: Weight coefficients in the cost functional in examples 1-3 in three-dimensional boundary velocity control.

also taken into account. The third example was performed with a higher importance given to shear stress and vorticity. The value of γ (convergence tolerance) was 0.01 in all examples. The descent parameter, λ, was chosen as 0.0001 in all examples, among a few other values in the range of 0.00005 to 0.0005. This value seems to be near optimal for these examples because the control iteration did not converge for higher values of λ, while the convergence was too slow for smaller values.

The flow rates achieved in these examples and the corresponding desired flow rate are plotted in Figure 4.14(a). The value of $(Q - Q^d)$ and the convergence rates are plotted in Figures 4.14(b) and 4.14(c), respectively, versus the number of iterations. The velocity vector and the streamline contours in example 1 are plotted in Figures 4.15 and 4.16 respectively, for time step 4 corresponding to peak systole. In Figures 4.17 and 4.18, the velocity vector and the streamline contours in example 1 are plotted respectively, for time step 16 corresponding to peak diastole. Similar field plots are presented in Figure 4.19-4.22 for example 3 . Finally, the evolution of a particular control velocity profile (at time step 16) as a function of control iteration is presented in Figures 4.23-4.25, for examples 1-3 respectively.

As shown in Figure 4.13(a), a relatively rough mesh was employed for this three-dimensional problem. This was due to the practical restrictions in the amount of available memory size and CPU time in our computer. The maximum Reynolds number we could achieve for this mesh was 277, which corresponds to a flow rate of 1.484 *lit/min*. For

higher values of Reynolds number, the solution of the Navier-Stokes equations did not converge. Our numerical experiments with the two-dimensional case of this problem (section 4.2.1) show that flows at higher Reynolds numbers can be solved by using a finer mesh with more elements and nodal points and also by using smaller time increments for the time interval. Despite the fact that the calculated Reynolds number of 756 was not achieved in the present three-dimensional case, the application of the optimization algorithm to three-dimensional problems is well demonstrated by the achieved results.

Concerning the sensitivity of the design scheme to the descent parameter, λ, and to the initial guess for the control, $\mathbf{v}(x)$, the discussion of section 4.2.1 applies equally to this three-dimensional case. In all the examples, a zero initial guess for the control was chosen, which avoids the complicated trial and error in providing a suitable non-zero velocity profile for the control.

In the first example, the only term in the cost functional (2.7) was $(Q-Q^d)$, implying that our goal was to reach a realistic flow rate, Q^d, regardless of the amount of shear stress and vorticity produced by the resultant flow pattern. According to Figure 4.14(a), the differences between the desired and optimal flow rates were relatively small, with the obtained average flow rate being equal, within 12%, to the desired one. On the other hand, in examples 2 and 3, shear stress and vorticity are present in the cost functional and, as discussed in section 4.2.1, the algorithm tends to reduce the stress level by reducing the velocity amplitude as well as by rearranging the flow distribution. Therefore, the optimal flow rate would be expected to be lower than the desired one. This is demonstrated in Figure 4.14(a) for examples 2 and 3, which shows that the average flow rates were about 50% and 75% lower than the specified desired flow rate in examples 2 and 3, respectively. This discrepancy poses no problem because it can be easily accommodated by specifying a higher desired flow rate. Figure 4.14(b) shows that, throughout the iterative process, the value of $(Q-Q^d)$ did not decrease in examples 2 and 3 as much as it did in example 1. One may note the oscillations in the convergence of example 1 in Figure 4.14(c). These oscillations can be attributed to the higher magnitudes of the velocity field in this example compared with those in example 2 and 3. In fact, further increase in

Reynolds number will result in divergence of the solution in Example 1. Regarding the rearrangement of the flow distribution by introducing the shear stress and vorticity in the cost functional, one may compare the velocity vector plots in Figures 4.17(b) and 4.21(b). In the first one, corresponding to example 1, sharp gradients in the velocity field around the middle of the plane can be observed. Also, a vortex seems to form under the control boundary facing the closed port. These two observations are no longer present in example 3 (Figure 4.21(b)), showing the success of the optimization algorithm in rearranging the flow distribution favoring the reduction of shear stress and vorticity.

It is interesting to note the difference in the shapes of the control velocity profiles in these examples (profiles corresponding to the last iteration in Figures 4.23-4.25). In the first example, the velocity profiles are nearly flat with sharp edges close to the walls, while in examples 2 and 3 these profiles are smoother. One might also note that the profiles in example 3 are smoother than the ones in example 2, which is attributed to the larger values of the weight coefficients α_c and α_g in example 3. This confirms our physical expectation that a smoother profile would produce lower velocity gradients in the regions close to the walls. Another interesting observation in these plots is that, when the stress level is minimized, the control velocity is highly skewed (Figures 4.24 and 4.25), so that higher velocity amplitudes correspond to locations facing an open port.

Finally, concerning the evolution of the boundary control profiles in Figures 4.23-4.25, we observe that, in all examples, the first profile suggested by the algorithm is flat. As the control iteration advances, the profile in example 1 reaches higher magnitudes but remains flat. On the other hand, in examples 2 and 3 the algorithm increases the magnitude of the control profiles while at the same time decreasing the values near the boundary and skewing them towards the open port.

Chapter 5

Conclusions and Suggestions for Further Studies

5.1 Conclusions

In the present work we have applied optimal control theory to the time-dependent Navier-Sokes equations in some models of artificial hearts. The main results of this study are as follows.

i) A few simplified mathematical models for artificial hearts were presented.

ii) The optimal control theory was applied to the governing equations, and the necessary conditions of optimality were derived in a manner suitable for numerical treatment of the theory.

iii) A cost functional, approximately relevant to artificial heart operation was formulated; it accounts for differences from a desired flow rate and the mean-squared shear stresses and vorticity.

iv) The numerical solution of the governing equations and the optimality conditions in

two- and three-dimensional domains were achieved.

v) Three models were studied numerically, which consist of a two-dimensional model with an initial velocity distribution as the control, a two-dimensional model with a time-dependent boundary velocity as the control and a three-dimensional model with a time-dependent boundary velocity as the control. In all cases, it was found that,

- in the absence of shear stress and vorticity in the cost functional, the optimization algorithm produces, within about 10%, the desired flow rate,

- when shear stress and vorticity are included in the cost functional, a decreased optimal flow rate is obtained,

- minimization of the stress and vorticity levels results in smoother velocity distributions, and

- the speed of convergence of the optimization algorithm is sensitive to the initial guess for the control variable and to the value of the descent parameter.

vi) In the case of two- and three-dimensional boundary velocity control, it was found that minimizing the stress and vorticity levels results in a highly skewed profile for the optimal boundary velocity control, so that higher velocity amplitudes correspond to locations facing an open port. These profiles are also smoother close to the walls, as opposed to the flat profiles with sharp edges close to the walls when stress and vorticity are not included in the cost functional.

vii) In the case of three-dimensional boundary control problem, it was found that, by including the stress and vorticity in the cost functional, the optimization algorithm successfully rearranges the flow distribution, it smoothens the sharp gradients and removes a vortex that would occur otherwise in the velocity field.

Because our objective was to demonstrate the validity of the application of optimal control theory in the area of fluid dynamics, and in order to avoid unnecessary com-

plications, we introduced certain idealizations and simplifications into our model of the physical system and the cost functional. Nevertheless, the same procedure could be applied to more realistic models of a given device without any conceptual difficulty, provided that increased computational resources are available. This study has demonstrated that optimal control of flow systems is practical and may be used as a design tool. In addition to artificial hearts, boundary control can be applied to the optimal design of a variety of other engineering systems involving fluids, such as hydraulic and pneumatic systems and flow generating apparatus, including wind tunnels and water channels. Generalization of optimal control may lead to the optimization not only of operational characteristics but also of geometrical features of such systems.

5.2 Suggestions for Further Research

5.2.1 Introducing a Moving Boundary as the Control

An extension of the present research will consist of studying model No. 2, as presented in section 2.1. This model represents a more realistic approximation of artificial hearts, in which the control is the motion of a boundary with a specified velocity profile. This boundary, in fact, represents a membrane or pusher plate, whose dynamics is assumed to be decoupled from the motion of the fluid as discussed in section 2.1. To implement this model, the following two distinct and demanding tasks must be undertaken:

(a) Developing the optimization algorithm. Due to the presence of a moving boundary, the derivation of the necessary conditions of optimality for this model would be quite complicated. As far as the author is informed, no optimization algorithm on moving boundary problems has been reported in the literature.

(b) Developing the necessary computer code. Having derived the necessary conditions of optimality, one should devise the necessary code to perform the numerical computations.

This consists of solving two different sets of equations, i.e. the Navier-Stokes equations and the corresponding adjoint equations. The solution procedure of the Navier-Stokes equations for moving boundary problems is not well established in the literature, although some results on moving boundary problems have been reported (e.g. see the FIDAP Manuals and the paper by Floryan and Rasmussen, 1989, for more references). Substantial efforts should be spent to modify our present Navier-Stokes program to account for moving boundaries in this model. The implementation of this algorithm would first be applied to two-dimensional problems and then extended to three-dimensional ones.

5.2.2 Implementation of More Realistic Models

A more realistic model of artificial hearts was formulated as model No. 1 in section 2.1. A possible continuation of this research, would be to develop the complete mathematical formulation of this model and to derive the necessary conditions of optimality (see Ahmed, 1995). It should be emphasized, however, that even proving the existence of an optimal control for this coupled moving boundary problem is not a trivial task from a mathematical point of view. The computational implementation of this model would also be the most difficult one among the alternative models presented in section 2.1.

5.2.3 Optimization with Different Cost Functionals

The optimal control problem described in previous chapters requires a cost functional to be specified. The choice of the cost functional requires a good understanding of the physical and physiological functions of artificial hearts. Extensive experimental studies on complications and problems in artificial hearts and heart valves have been reported in the literature. A brief review of these results was presented in section 1.4, which could be further continued in the future with the aim of determining the most realistic criteria to be included in the cost functional. Further numerical simulations of the problems

described in the previous sections may be carried out with these criteria included in the cost functional.

5.2.4 Introducing a Flow Constraint in the Adjoint Equations

A glance at the formulation of the optimal control problems discussed in sections 2.3 and 2.4 shows that the constraint of a desired flow rate has been taken into account by introducing the squared norm of $(Q - Q^d)$ in the cost functional. This formulation results in a discrepancy between the desired and optimal flow rates, especially when shear stress and/or vorticity are present in the cost functional (see the numerical results in sections 4.2 and 4.3). An alternative approach in implementing the constraint of flow rate would consist of introducing this constraint in the adjoint equations with a Lagrange Multiplier technique. This would require revising the mathematical formulation and deriving the new conditions of optimality and then incorporating them in the present numerical code. By doing so the above mentioned discrepancy will be removed.

5.2.5 Towards an Optimal Artificial Heart Configuration

The present optimization study has so far been concerned with minimizing a cost functional, consisting of fluid mechanical parameters, in a specified geometrical configuration. It is obvious that a general optimization of an artificial heart should also deal with optimization of design parameters, such as shape, overall size, location of ports and surgical convenience. In the past, extensive multidisciplinary research has improved the design of artificial hearts. Blood chambers have been redesigned to eliminate stagnation areas, while the traditional diaphragm-type blood chamber has, in some designs, been replaced by a flexible blood sac (Jin and Clark, 1993). The present optimization process could be applied to various acceptable designs, and the design producing the lowest minimum value for the cost functional would be considered as the optimal one.

References

[1] Abergel F. and Temam R., *On Some Control Problems in Fluid Mechanics*, Theoret. Comput. Fluid Dynamics, vol. 1, 303-325, 1990.

[2] Ahmed N. U., *Optimal Control of Stochastic Systems*, Probabilistic Analysis and Related Topics, Bharucha-Reid A. T. (ed.), Academic Press, New York, 1979.

[3] Ahmed N. U., *Elements of Finite Dimensional Systems and Control Theory*, Longman, John Wiley & Sons, New York, 1988.

[4] Ahmed N. U., *Mathematical Challenges of the Artificial Heart*, 10th Canadian Symposium on Fluid Dynamics, University of New Brunswick, Fredericton, June 2-4, 1992a.

[5] Ahmed N. U., *Optimal Control of Hydrodynamic Flow with Possible Application to Artificial Heart*, Dynamic Systems and Applications, vol. 1, 103-120, 1992b.

[6] Ahmed N. U., *Mathematical Problems in Modeling Artificial Heart*, J. Mathematical Problems in Engineering, vol. 1, no. 2, 1-10, 1995.

[7] Ahmed N. U. and Teo K. L., *Optimal Control of Distributed Parameter Systems*, North Holland, New York, 1981.

[8] Akay H. U., *UCODE3 User's Manual - A Two- and Three-Dimensional Finite Element Program for the Solution of Steady and Unsteady Field Equations - Version 90.1*, Purdue University at Indianapolis, Indianapolis, Indiana, 1990.

[9] Atsumi M. D., Imachi K., Fujimasa I. and Sezai Y., *Research and Development of Artificial Heart in Japan*, J. Biomaterials Applications, vol. 4, 161-224, 1990.

[10] Baker A. J., *A Finite Element Computational Theory for the Mechanics and Thermodynamics of a Viscous Compressible Multi-Species Fluid*, Bell Aerospace Research Rept. 9500-920200, 1971.

[11] Baker A. J., *Finite Element Solution for Viscous Incompressible Fluid Dynamics*, Int. J. Numer. Meth. Eng., vol. 6, 89-101, 1973.

[12] Baker A. J., *A Finite Element Algorithm for the Navier-Stokes Equations*, NASA CR-2391, 1974.

[13] Baker A. J., *Finite Element Computational Fluid Mechanics*, Hemisphere, Washington, 1983.

[14] Becker E. B., Carey G. J. and Oden J. T., *Finite Elements: An Introduction*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[15] Bernhard W. F., *A Fibrillar Blood-Prosthetic Interface for both Temporary and Permanent Ventricular Assist Devices: Experimental and Clinical Observations*, Artificial Organs, vol. 13, 255-271, 1989.

[16] Bird R. B., Armstrong R. C. and Hassager O., *Dynamics of Polymeric Liquids*, John Wiley & Sons, New York, vol. 1, 1987.

[17] Bird R. B., Dai G. C. and Yarusso B. J., *The Rheology and Flow of Viscoplastic Materials*, Reviews in Chem. Eng., vol. 1, no. 1, 1-70, 1983.

[18] Blackshear P. L., *Hemolysis and Prosthetic Surfaces*, Chemistry of Biosurfaces, Hair M. L. (ed.), Marcel Dekker, New York, vol. 2, 523-561, 1972.

[19] Brebbia C. A. and Connor J. J., *Fundamentals of Finite Element Techniques for Structural Engineers*, Butterworths, London, 1975.

[20] Butkovskiy A. G., *Distributed Control Systems*, American Elsevier Publishing Co. Inc., New York, 1969.

[21] Campion R. and Crochet M. J., *On the Streamfunction-Vorticity Finite Element Solutions of Navier-Stokes Equations*, Int. J. Num. Meth. Eng., vol. 12, 1809-1818, 1978.

[22] Carey G. F. and Oden J. T., *Finite Elements: A Second Course*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[23] Cebeci T. and Smith A. M. O., *A Finite Difference Solution of the Incompressible Turbulent Boundary Layer Equations by an Eddy Viscosity Concept*, McDonnell Douglas Aircraft Company Rept. DAC-67130, 1968.

[24] Chein S., Usami S., Taylor M., Lundberg J. L. and Gregersen M. I., J. Appl. Physiol., vol. 21, 81-87, 1966.

[25] Cheon G. J., Kim H. C., Min B. G. and Han D. C., *A New Type of Motor-Driven Blood Pump for Artificial Heart*, J. Biomech. Eng., vol. 112, 473-475, 1990.

[26] Coulantuoni G., Hellums J. D. and Moake J. L., *The Response of Human Platelets to Shear Stress at Short Exposure Times*, American Society of Artificial Internal Organs, vol. 23, 626-631, 1977.

[27] Cuvelier C., *Optimal Control of a System Governed by the Navier-Stokes Equations Coupled with the Heat Equations*, New Developments in Differential Equations, Eckhaus W. (ed.), North Holland, Amsterdam, 81-98, 1978.

[28] Cuvelier C., Segal A. and van Steenhoven A., *Finite Element Methods and Navier-Stokes Equations*, Reidel Publishing Co., Dordrecht, 1986.

[29] Davis A. J., *The Finite Element Method: A First Approach*, Clarendon, Oxford, 1980.

[30] Dhatt G., Fomo B. K. and Bourque C. A., *A $\psi - \omega$ Finite Element Formulation for the Navier-Stokes Equations*, Int. J. Num. Meth. Eng., vol. 17, 199-212, 1981.

[31] Dhatt G. and Touzot G., *The Finite Element Displayed*, John Wiley & Sons, Chichester, 1984.

[32] Fattorini H. O. and Sritharan S. S., *Existence of Optimal Controls for Viscous Flow Problems*, Proc. Roy. Soc. Lond. A., vol. 439, 81-102, 1992.

[33] FIDAP7.0 Manuals, Fluid Dynamics International Inc., 1993.

[34] Fleming W. H. and Rishel R. W., *Deterministic and Stochastic Optimal Control*, Springer-Verlag, Berlin, 1975.

[35] Floryan J. M. and Rasmussen H., *Numerical Methods for Viscous Flows with Moving Boundaries*, Appl. Mech. Rev., vol. 42, no. 12, 323-341, 1989.

[36] Forsikov A., *Control Problems and Theorems Concerning the Unique Solvability of a Mixed Boundary Value Problem for the Three Dimensional Navier-Stokes and Euler Systems*, Math. USSR Sbornik., vol. 43, 251-273, 1982.

[37] Forsikov A., *On Some Control Problems and Results Concerning the Unique Solvability of a Mixed Boundary Value Problem for the Three-Dimensional Navier-Stokes and Euler Systems*, Soviet Math. Dokl., vol. 21, 889-893, 1983a.

[38] Forsythe G. E. and Wasow W. R., *Finite Difference Methods for Partial Differential Equations*, John Wiley & Sons, New York, 1960.

[39] Fung Y. C., *Biomechanics: Mechanical Properties of Living Tissues*, Springer-Verlag, New York, 1981.

[40] Fung Y. C., *Biomechanics: Motion, Flow, Stress and Growth*, Springer-Verlag, New York, 1990.

[41] Gabe I. T. and Zazt L., *Biorheology*, vol. 5, 86, 1968.

[42] Galanga F. L. and Lloyd J. R., *An Experimental Study of the Flow-Induced Mass Transfer Distribution in the Vicinity of Prosthetic Heart Valves*, J. Biomech. Eng., vol. 103, 1-10, 1981.

[43] Gamkrelidze R. V., *Principles of Optimal Control Theory*, Plenum Press, New York, 1978.

[44] Gartling D. K., *NACHOS II - A Finite Element Computer Program for Incompressible Flow Problems, Part I - Theoretical Background*, Sandia National Lab., Albuquerque, New Mexico, 1987.

[45] Girault V. and Raviart P. A., *Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag, Berlin, 1986.

[46] Glass L., Hunter P. and McCulloch A. (eds.), *Theory of Heart*, Springer-Verlag, New York, 1991.

[47] Glowinski R., *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, 1984.

[48] Glowinski R. and Pironneau O., *Finite Element Methods for Navier-Stokes Equations*, Ann. Rev. Fluid Mech., vol. 24, 167-204, 1992.

[49] Goldsmith H. L. and Skalak R., *Hemodynamics*, Ann. Rev. Fluid Mech., vol. 7, 213-247, 1975.

[50] Gunzburger M. D., *Finite Element Methods for Viscous Incompressible Flows*, Academic Press, Boston, 1989.

[51] Gunzburger M. D., Hou L. and Svobodny T., *Boundary Velocity Control of Incompressible Flow with an Application to Viscous Drag Reduction*, SIAM J. Control and Optim., vol. 30, 167-181, 1992.

[52] Gunzburger M. D. and Peterson J., *Issues in Shape Optimization for the Navier-Stokes Equations*, Proceedings of the 31st Conference on Decision and Control, Tucson, Arizona, 3390-3392, 1992.

[53] Hasbani Y. and Engelman M., *Out-of-Core Solution of Linear Equations with Non-Symmetric Coefficient Matrix*, Computers and Fluids, vol. 7, 13-31, 1979.

[54] Hinton E. and Owen D. R. J., *An Introduction to Finite Element Computations*, Pineridge Press, Swansea, 1979.

[55] Huebner K. H., *The Finite Element Method for Engineers*, John Wiley & Sons, New York, 1975.

[56] Huebner K. H. and Thornton E. A., *The Finite Element Method for Engineers*, 2nd ed., John Wiley & Sons, New York, 1982.

[57] Hughes T. J. R., Liu W. K. and Brooks A., *Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation*, J. Comput. Physics, vol. 30, 1-60, 1979.

[58] Jin W. and Clark C., *Experimental Investigation of Unsteady Flow Behaviour within a Sac-Type Ventricular Assist Device (VAD)*, J. Biomechanics, vol. 26, no. 6, 697-707, 1993.

[59] Joyce L., Johnson K. E., Pierce W. S., DeVries W. C., Semb B. K. H., Copeland J. G., Griffith B. P., Cooley D. A., Frazier O. H., Cabrol C., Keon W. J., Unger F., Bucherl E. S. and Wolner E., *Summary of the World Experience With Clinical Use of Total Artificial Hearts as Heart Support Devices*, J. Heart Transplant, vol. 5, 229-235, 1986.

[60] Kolff W. J., *Experiences and Practical Considerations for the Future of Artificial Hearts and of Mankind*, Artificial Organs, vol. 12, 89-11, 1988.

[61] Krylov N. V., *Controlled Diffusion Processes*, Springer-Verlag, New York, 1980.

[62] Ladyzhenskaya O. A., *The Mathematical Theory of Viscous Incompressible Flow*, 2nd ed., Gordon and Breach, New York, 1969.

[63] Ladyzhenskaya O. A., *New Equations for the Description of the Viscous Incompressible Fluids and Solvability in the Light of the Boundary Value Problems for Them*, Boundary Value Problems of Mathematical Physics, AMS, Providence, vol. 5, 95-118, 1970a.

[64] Ladyzhenskaya O. A., *Modification of the Navier-Stokes Equations for Large Velocity Gradients*, Boundary Value Problems of Mathematical Physics and Related Aspects of Function Theory, New York, vol. 2, 57-69, 1970b.

[65] Landau L. D. and Lifshitz E. M., *Fluid Mechanics*, Pergamon Press, London, 1959.

[66] Lelkes P. I. and Samet M. M., *Endothelialization of the Luminal Sac in Artificial Cardiac Prostheses: A Challenge for Both Biologists and Engineers*, J. Biomech. Eng., vol. 113, 132-142, 1991.

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

[68] Mann K. A., Deutsch S., Tarbel J. M., Geselowitz D. B., Rosenburg G. and Pierce W. S., *An Experimental Study of Newtonian and Non-Newtonian Flow Dynamics in a Ventricular Assist Device*, J. Biomech. Eng., vol. 109, 139-147, 1987.

[69] Merrill E. W., Benis A. M., Gilliland E. R., Sherwood T. K. and Salzman E. W., J. Appl. Physiol., vol. 20, 954-967, 1965.

[70] Moritz A., Wolner E. and Nose Y., *Clinical Use of the Artificial Heart, Indications and Results*, Wien Klin Wochenscher, vol. 18, 161-167, 1988.

[71] Mussivand T., Navarro R., Chen J. F., Braun J., Harasaki H., Kiraly R., Batur C., McMillin C. and Nose Y., *Flow Visualization in an Artificial Heart Using Diffuse and Laser Lighting*, Amer. Soc. Advanced Instrum. Org. vol. 34, no. 3, 317-321, 1988.

[72] Mussivand T., Masters R. G., Hum A., Holmes K. S., Rajagopalan K., Hendry P. J. and Keon W. J., *Pulsatile Flow Artificial Heart*, Proceedings of 15th Canadian Congress of Applied Mechanics, 1995.

[73] Nakamura M., *Numerical Study on the Unsteady Flow of Non-Newtonian Fluid*, J. Biomech. Eng., vol. 112, no. 1, 100-103, 1990.

[74] Nose Y., *My Life with the National Institutes of Health Artificial Programs*, Artificial Organs, vol. 14, 174-190, 1990.

[75] Oden J. T., *Finite Elements of Nonlinear Continua*, McGraw-Hill, New York, 1972.

[76] Oden J. T. and Reddy J. N., *An Introduction to the Mathematical Theory of Finite Elements*, Wiley, New York, 1976.

[77] Owens D. H., *Multivariable and Optimal Systems*, Academic Press. London, 1981.

[78] Pedley T. J., *The Fluid Mechanics of Large Blood Vessels*, Cambridge University Press, Cambridge, 1980.

[79] Perktold K., *Pulsatile Non-Newtonian Blood Flow in Three-Dimensional Carotid Bifurcation Models*, J. Biomed. Eng., vol. 13, no. 6, 507-515, 1991.

[80] Perktold K., *Flow and Stress Characteristics in Rigid Walled and Compliant Carotid Artery Bifurcation Models*, J. Med. & Biolo. Eng. & Computing, vol. 32, no. 1, 19-16, 1994.

[81] Peskin C. S., *Flow Patterns Around Heart Valves: A Numerical Method*, J. Comput. Physics, vol. 10, 252-271, 1972.

[82] Peskin C. S., *The Fluid Dynamics of Heart Valves: Experimental, Theoretical and Computational Methods*, Ann. Rev. Fluid Mech., vol. 14, 235-259, 1982.

[83] Pham T. V. and Mitsoulis E., *Entry and Exit Flows of Casson Fluids*, Canadian J. Chem. Eng., vol. 72, 1080-1084, 1994.

[84] Polya B. T., *The Convergence Rate of the Penalty Function Method*, USSR Computational Mathematics and Mathematical Physics, vol. 11, 1-12, 1971.

[85] Pontryagin L. S., Boltyanskii V. G., Gamkrelidze R. V. and Mishchenko E. F., *The Mathematical Theory of Optimal Processes*, John Wiley & Sons, New York, 1962.

[86] Rao S. S., *The Finite Element Method in Engineering*, Pergamon Press, Oxford, 1982.

[87] Reddy J. N., *On the Finite Element Method with Penalty for Incompressible Fluid Flow Problems*, The Mathematics of Finite Elements and Applications vol. 3, Academic Press, London, 227-235, 1979.

[88] Reddy J. N., *On Penalty Function Methods in the Finite Element Analysis of Flow Problem*, Int. J. Numer. Meth. in Fluids, vol. 2, 151-171, 1982.

[89] Reddy J. N., *An Introduction to the Finite Element Method*, McGraw-Hill, New York, 1984.

[90] Sahrapour A. Ahmed N.U. and Tavoularis S., *Optimal Control of the Navier-Stokes Equations*, Proceedings of the First Conference of the CFD Society of Canada, Montreal, 437-448, 1993.

[91] Sahrapour A., Ahmed N.U. and Tavoularis S., *Boundary Control of the Navier-Stokes Equations with Potential Application to Artificial Hearts*, Proceedings of the 2nd Conferenc of the CFD Society of Canada, Gottlieb J. and Ethier C. (eds.), Toronto, 387-394, 1994.

[92] Sakakibara N., Novoa R., Davies C. R., Chen J. F., Jacobs G. B., Takatani S., Mussivand T., Golding L. R., Nose Y. and Loop F. D., *New Crisscross-Shaped Port Design for Universal Serial Pumps*, Amer. Soc. Advanced Instrum. Org., vol. 35, 713-435, 1989.

[93] Sallam A. M. and Hwang N. H. C., *Red Blood Cell Hemolysis in a Turbulent Jet*, ASME 1983 Biomechanics Symposium, Huston,Texas, 83-86, 1983.

[94] Sani R. L., Gresho P., Lee R. L. and Griffiths D. F., *The Cause and Cure of the Spurious Pressure Generated by Certain GFEM Solutions of the Navier-Stokes Equations*, Int. J. Num. Meth. Fluids, vol. 1, 17, 1981.

[95] Schwarz A. C., Tiederman W. G. and Phillips W. M., *Influence of Cardiac Flow Rate on Turbulent Shear Stress from a Prosthetic Heart Valve*, J. Biomech. Eng., vol. 110, 123-128, 1988.

[96] Shah V. L., *Blood Flow*, Advances in Transport Processes, Mujumdar A. S. and Mashelkar R. A. (eds.), Halsted Press, New Delhi, vol. 1, 1-57, 1980.

[97] Sritharan S. S., *An Optimal Control Problem in Exterior Hydrodynamics*, Proc. Roy. Soc. Edinb. A., 1992.

[98] Stein P. D. and Sabbah H. N., *Measured Turbulence and Its Effects Upon Thrombus Formation*, Circulation Research, vol. 35, 608-614, 1974.

[99] Tanner R. I., *Die-Swell Reconsidered: Some Numerical Solutions Using a Finite Element Program*, Appl. Polym. Symp., vol. 20, 201-208, 1973.

[100] Tanner R. I., *Engineering Reology*, Oxford University Press, Oxford, 1985.

[101] Tarbel J. M., Gunshinan J. P., Geselowitz D. B., Rosenburg G., Shung K. K. and Pierce W. S., *Pulsed Ultrasonic Doppler Velocity Measurements Inside a Left Ventricular Assist Device*, J. Biomech. Eng., vol. 108, 232-238, 1986.

[102] Taylor C. and Hood P., *A Numerical Solution of the Navier-Stokes Equations Using the Finite Element Technique*, Comput. Fluids, vol. 1, 73-100, 1973.

[103] Taylor D. E. M. and Wade J. D., *Pattern of Blood Flow within the Heart: a Stable System*, Cardiovasc. Res., vol. 7, 14-21, 1973.

[104] TECPLOT Manual, Amtec Engineering Inc., 1993.

[105] Temam R., *Navier-Stokes Equations*, North Holland, Amsterdam, 1979.

[106] Temam R., *Navier-Stokes Equations and Nonlinear Functional Analysis*, SIAM, Philadelphia, 1983.

[107] Thomasset F., *Implementation of Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag, New York, 1981.

[108] Tiederman W. G., Steinle M. J. and Phillips W. M., *Two-Component Laser Velocimeter Measurements Downstream of Heart Valve Prostheses in Pulsatile Flow*, J. Biomech. Eng., vol. 108, 59-64, 1986.

[109] Warga J., *Optimal Control of Differential and Functional Equations*, Academic Press, New York, 1972.

[110] Williams A. R., *The Induction of Intervascular Thrombi In Vivo by Means of Localized Hydrodynamic Shear Stress*, Basic Elements of Blood Trauma, Martinus Nijhoff, 63-73, 1978.

[111] Woo Y. R. and Yoganathan A. P., *In Vitro Pulsatile Flow Velocity and Shear Stress Measurements in the Vicinity of Mechanical Mitral Heart Valve Prostheses*, J. Biomech., vol. 19, no. 1, 39-51, 1986.

[112] Wonham W. M., *Random Differential Equations in Control Theory*, Probabilistic Methods in Applied Mathematics, Bharucha-Ried A. T. (ed.), Academic Press, New York, vol. 2, 1970.

[113] Yoganathan A. P., Corcoran W. H. and Harrison E. C., *In Vitro Measurements in the Vicinity of Aortic Prosthesis*, J. Biomech., vol. 12, 135-152, 1979.

[114] Yoganathan A. P., Woo Y. R. and Williams F. P., *In Vitro Hydrodynamic Characteristics of the St. Jude Bileaflet Aortic Prosthesis*, Advances in Cardiac Valves, DeBokey M. E. (ed.), Yorke Med. Books, 1982.

[115] Zienkiewicz O. C., *The Finite Element Method*, McGraw-Hill, London, 1979.

[116] Zienkiewicz O. C. and Cheung Y. K., *Finite Elements in the Solution of Field Problems*, The Engineer, 507-510, 1965.

[117] Zienkiewicz O. C. and Morgan K., *Finite Elements and Approximation*, John Wiley & Sons, New York, 1983.

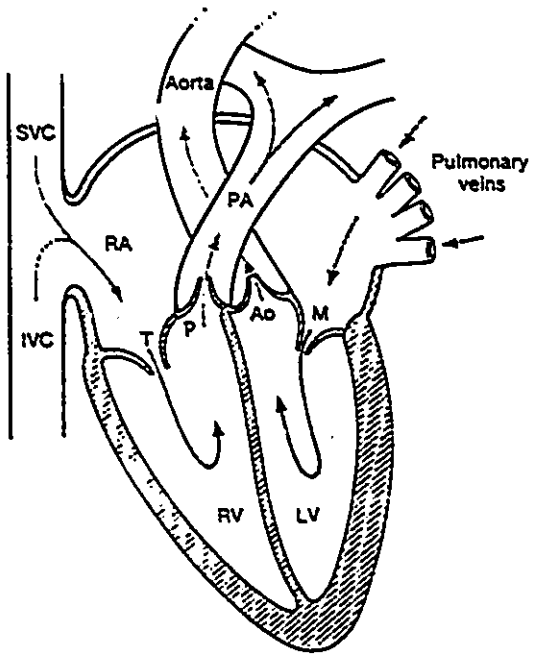


Figure 1.1: Blood flow through the heart. The arrows show the direction of blood flow. The symbols are: SVC, superior vena cava; IVC, interior vena cava; RA, right atrium; RV, right ventricles; PA, pulmonary artery; LV, left ventricle. The valves are T, tricuspid, P, pulmonary, AO, aortic, M, mitral. From Fung (1990).

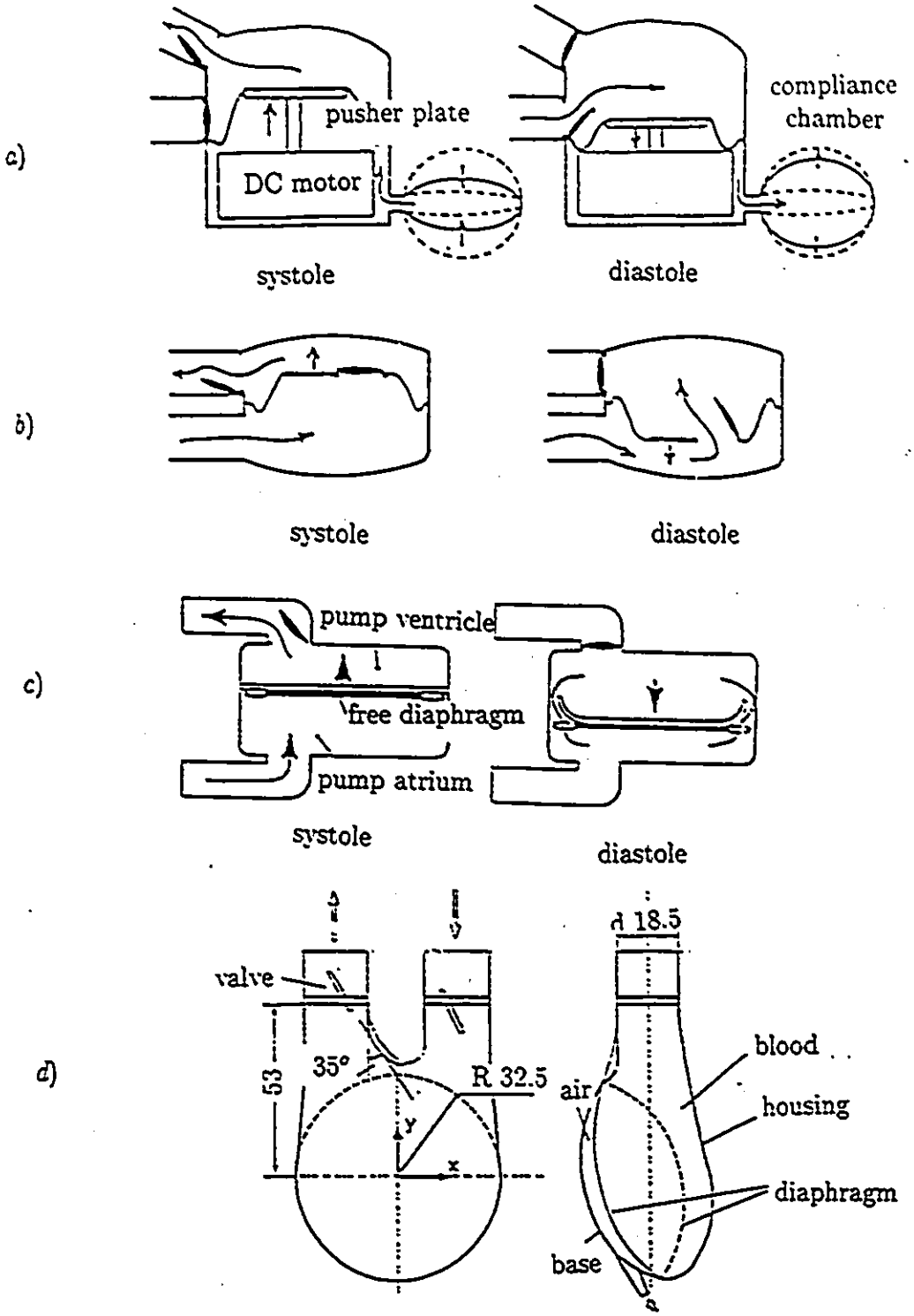


Figure 1.2: Some designs for artificial hearts, *a*) pump with a compliance chamber, *b*) pump without a compliance chamber, *c*) free diaphragm blood pump (from Atsumi *et al.*, 1990), *d*) a sac-type ventricular assist device (from Jin and Clark, 1993).

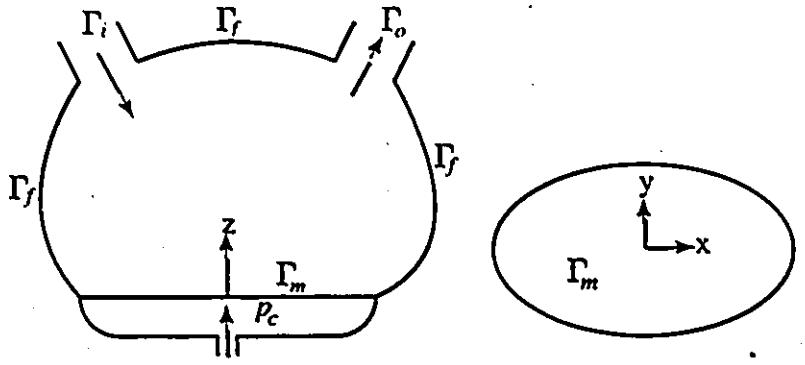


Figure 2.1: Artificial heart with membrane.

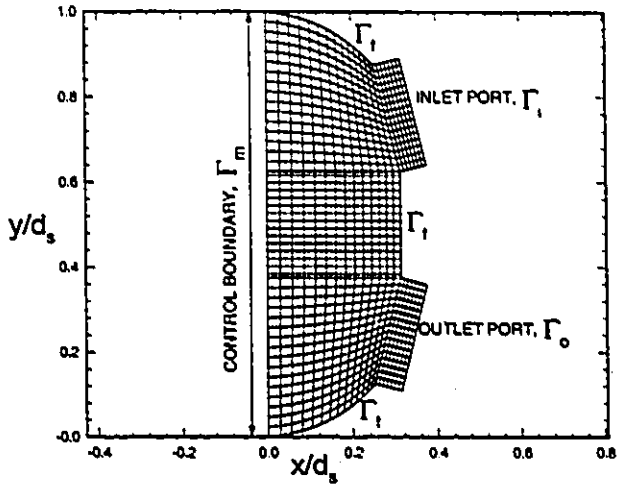


Figure 4.1: Computational grid system.

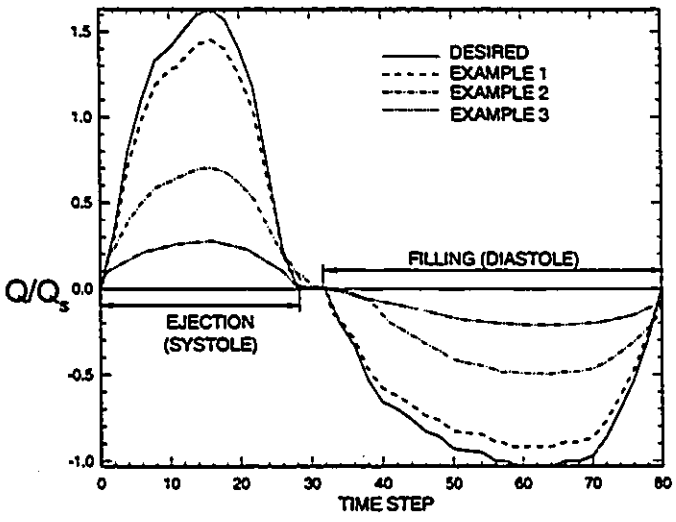
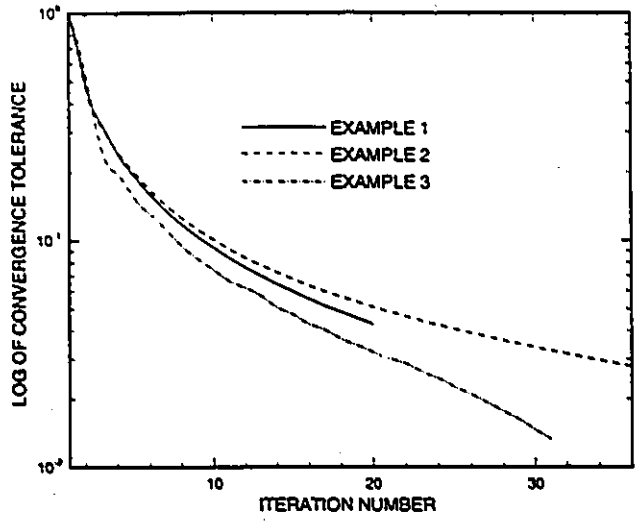
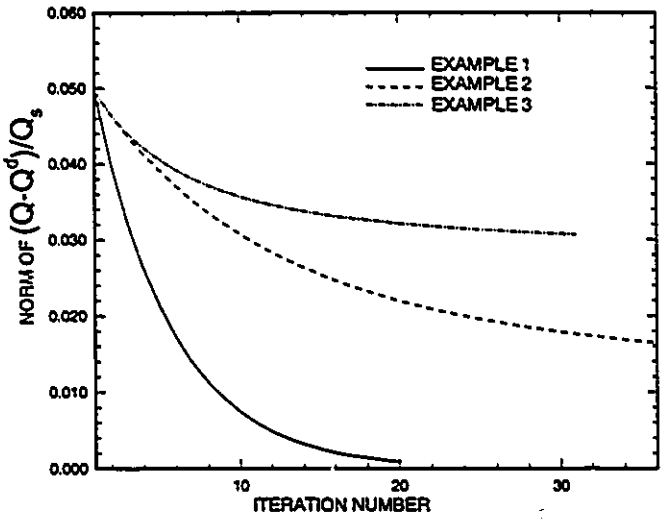


Figure 4.2: Desired and optimal flow rates in examples 1-3.



(a)



(b)

Figure 4.3: (a) Convergence rates and (b) variation of $\bar{Q} - Q^d$ in examples 1-3.

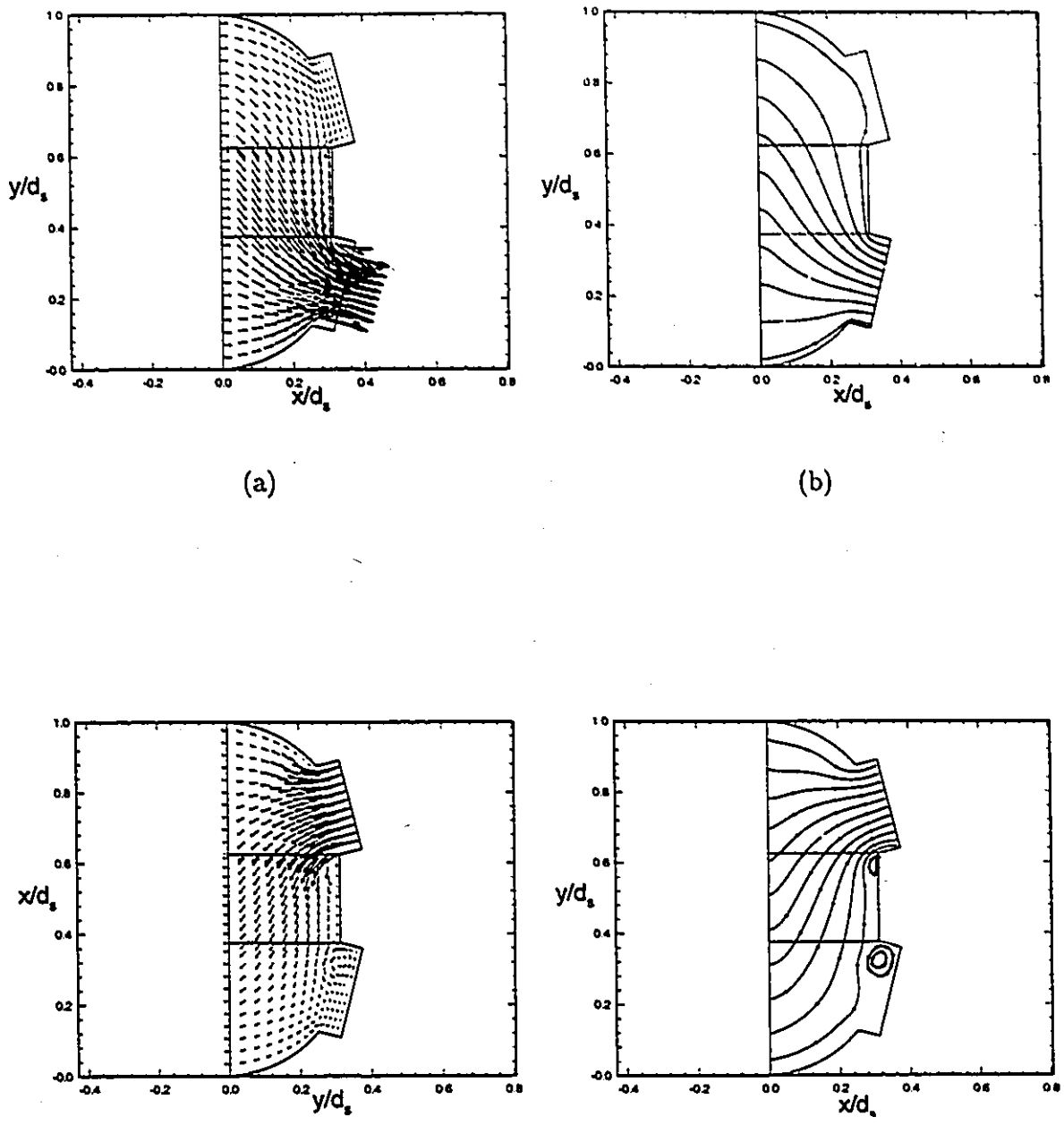


Figure 4.4: (a) Velocity vector plot and (b) streamlines at systolic peak; (c) velocity vector plot and (d) streamlines at diastolic peak in example 1.

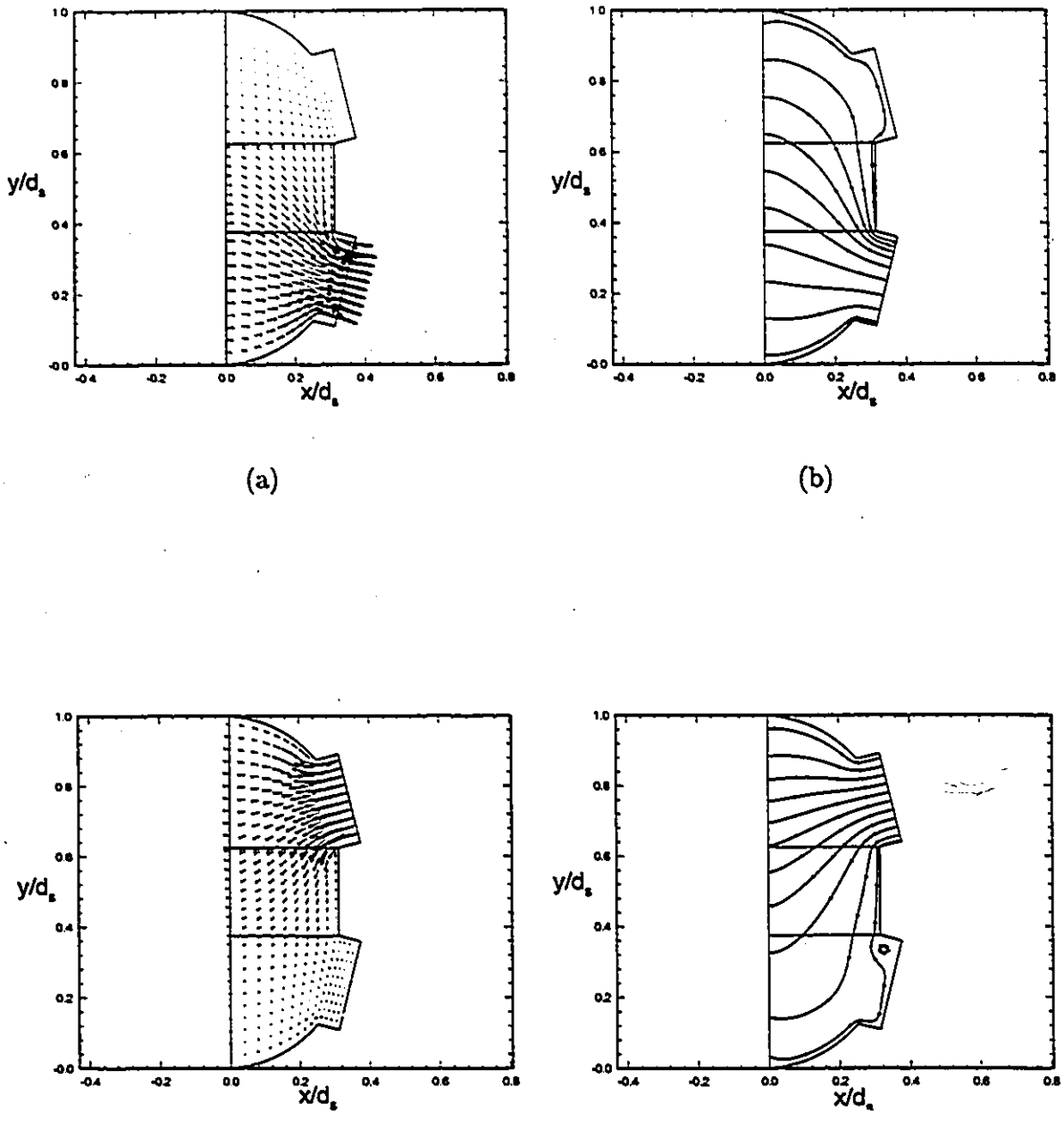


Figure 4.5: (a) Velocity vector plot and (b) streamlines at systolic peak; (c) velocity vector plot and (d) streamlines at diastolic peak in example 3.

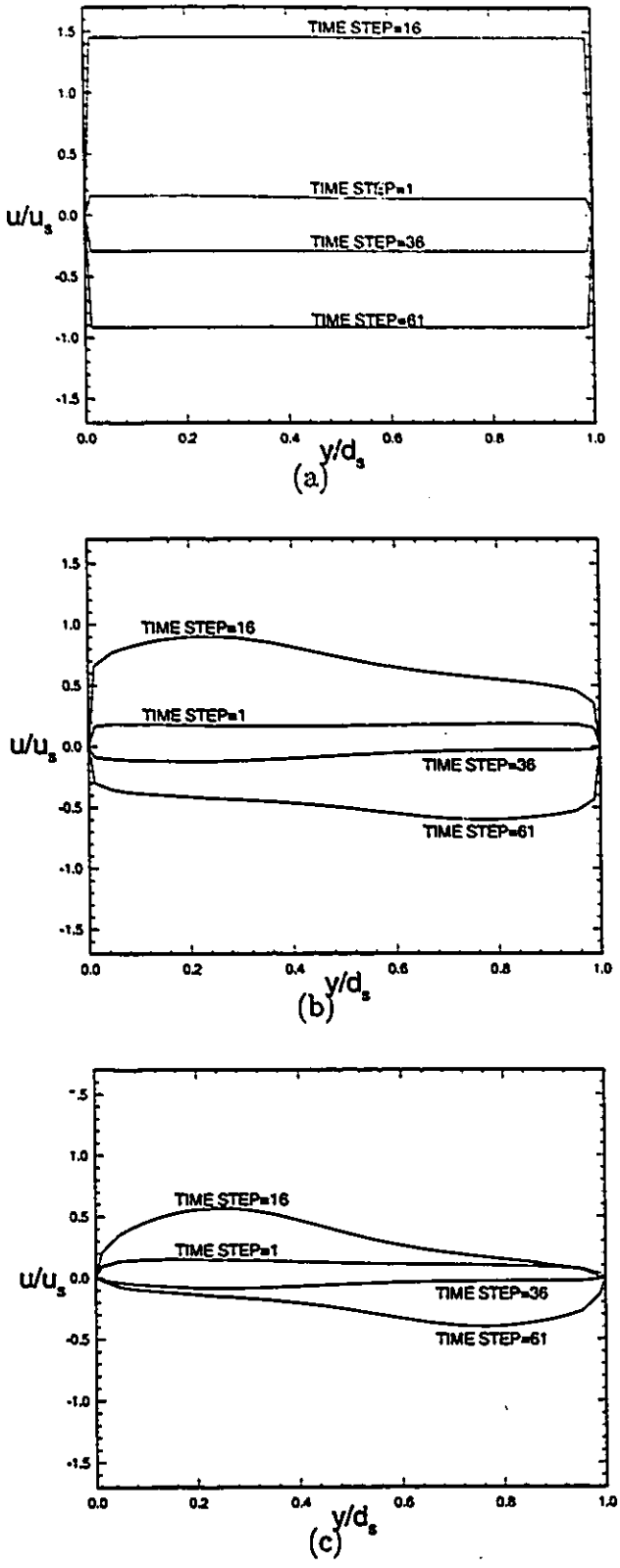


Figure 4.6: Optimal boundary control profiles in examples 1-3.

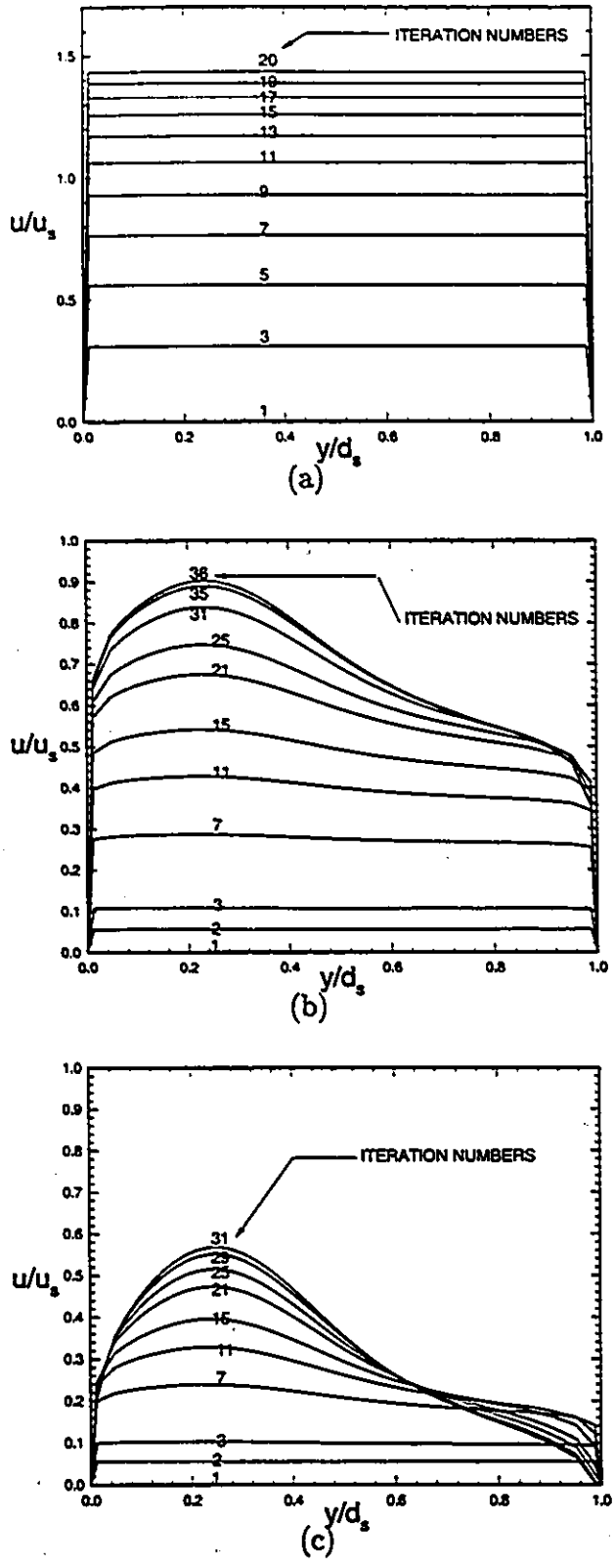
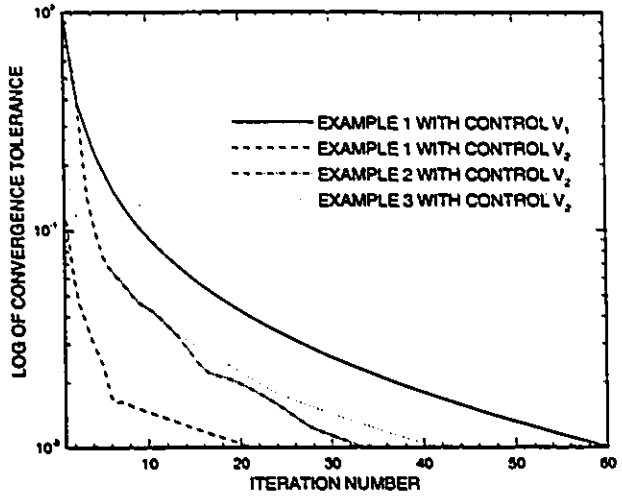
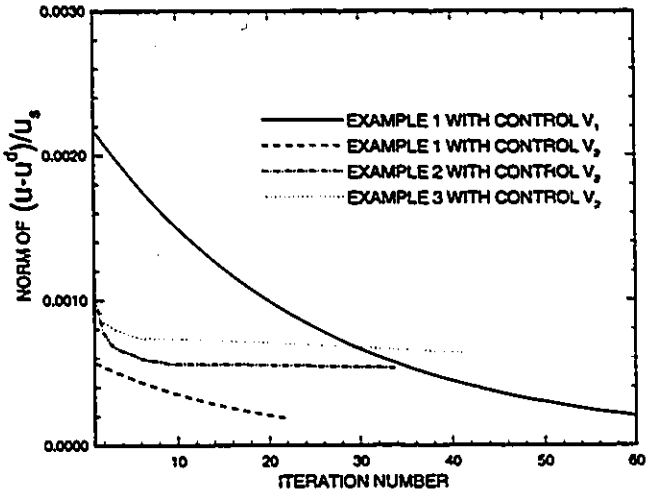


Figure 4.7: Evolution of the boundary control profile at systolic peak in examples 1-3.

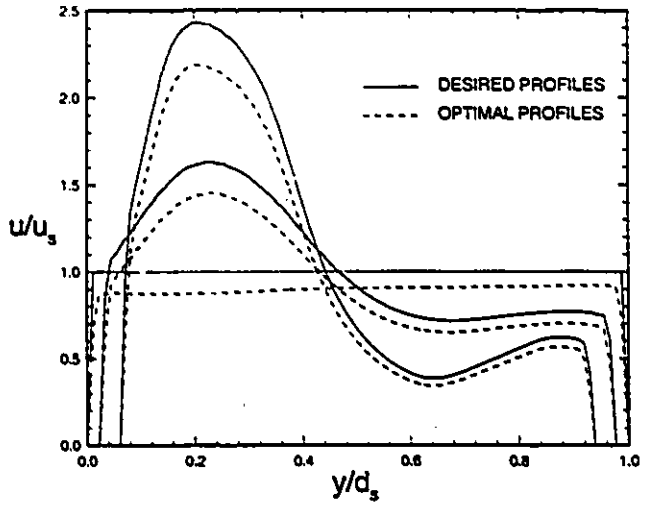


(a)

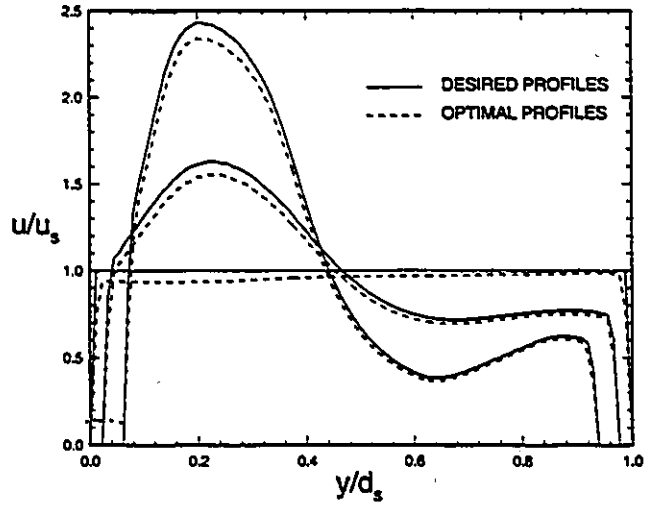


(b)

Figure 4.8: a) Convergence rates and b) variation of $u-u^d$ in examples 1-3 in initial velocity control.



(a)



(b)

Figure 4.9: Desired and optimal velocity profiles for 3 cross sections (at x/d_s=0.0, 0.10 and 0.17) in example 1 with a) initial control $\mathbf{v}_1$ and b) initial control $\mathbf{v}_2$.

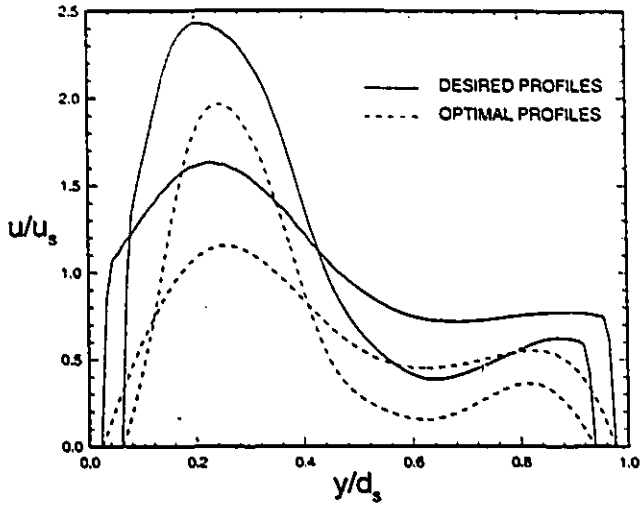


Figure 4.10: Desired and optimal velocity profiles for 2 cross sections (at x/d_s=0.10 and 0.17) in example 2 with initial control v_2.

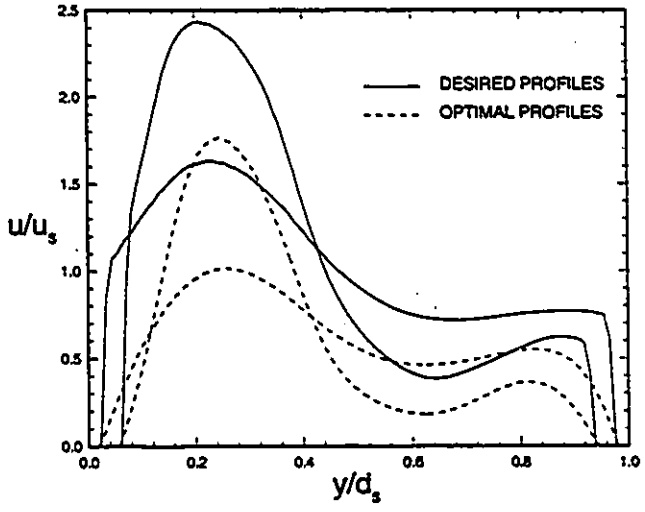
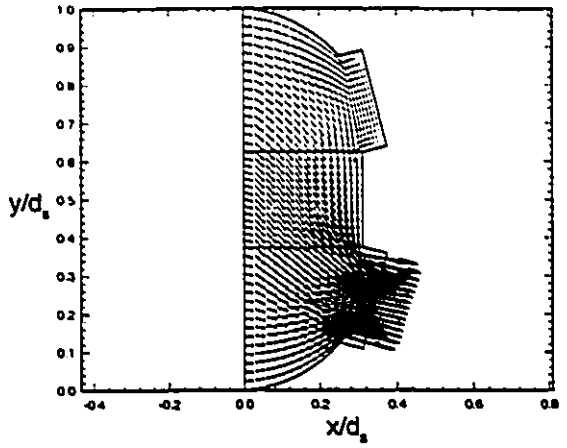
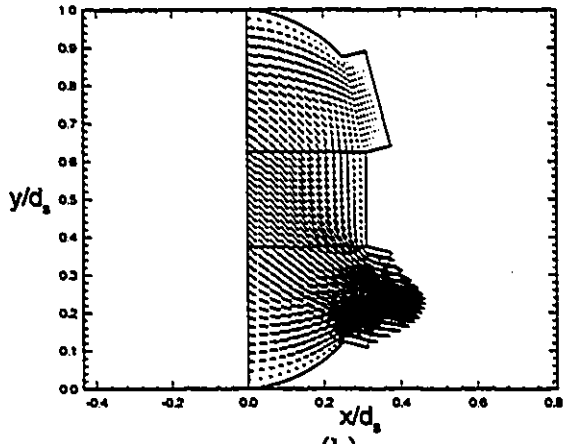


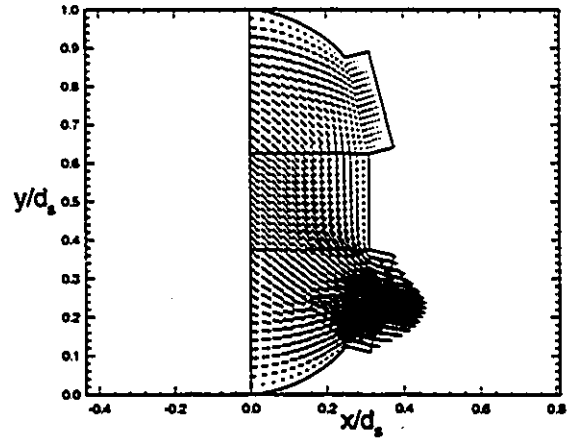
Figure 4.11: Desired and optimal velocity profiles for 2 cross sections (at x/d_s=0.10 and 0.17) in example 3 with initial control v_2.



(a)

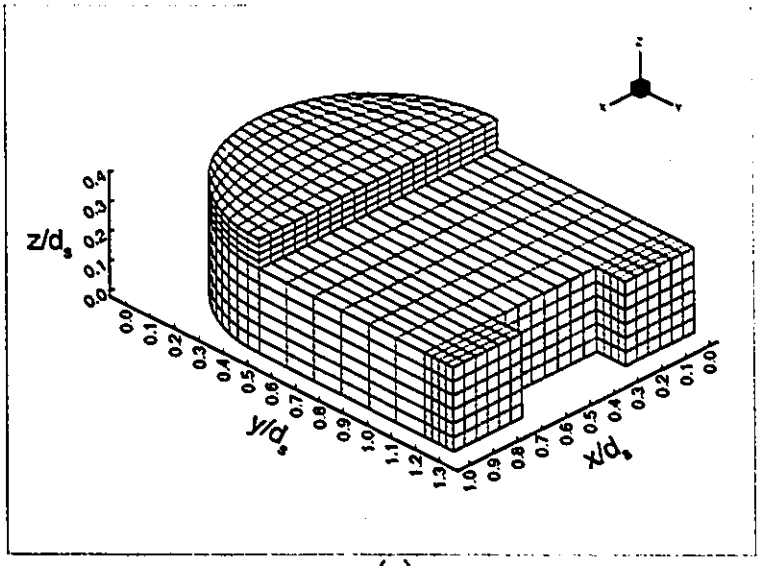


(b)

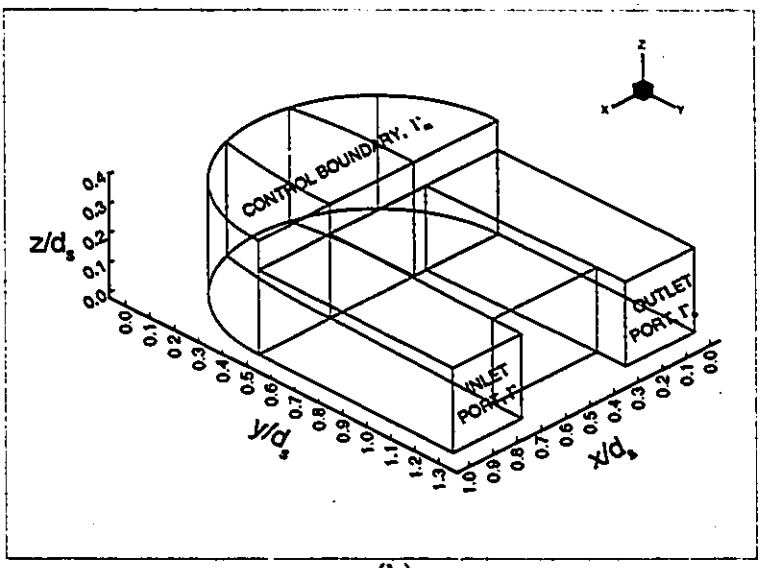


(c)

Figure 4.12: Velocity vector plots of the optimal field at time step 5 in (a) example 1; (b) example 2 and (c) example 3.

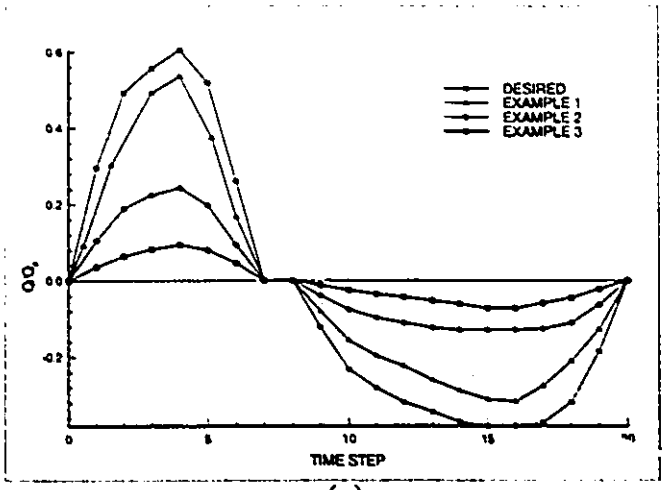


(a)

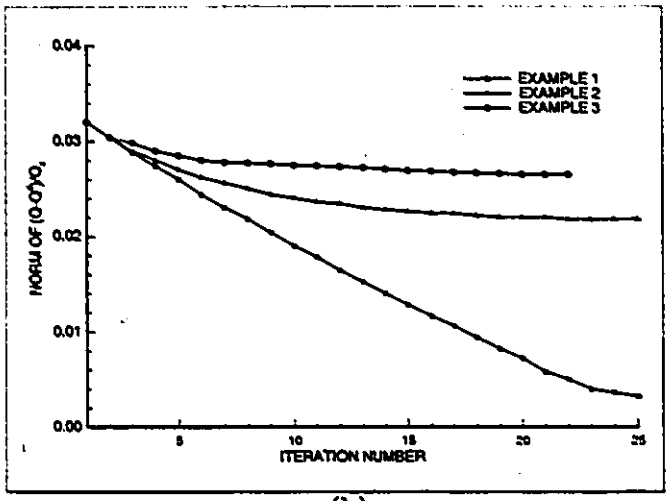


(b)

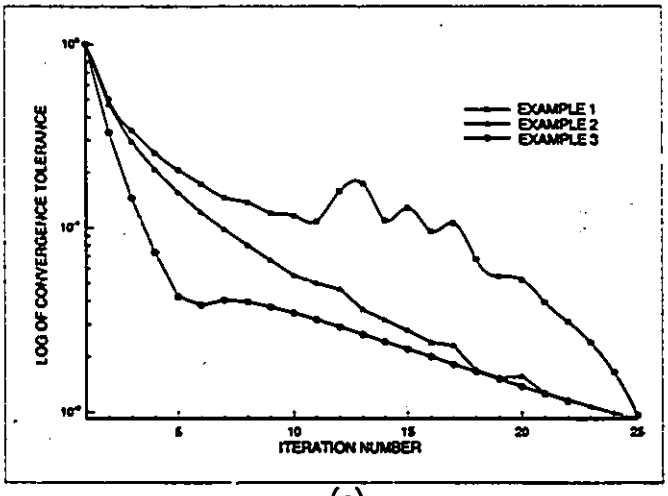
Figure 4.13: (a) Computational grid system with 326 27-node elements (mid-side nodes are also connected); (b) Control, inlet and outlet boundaries in the mesh.



(a)

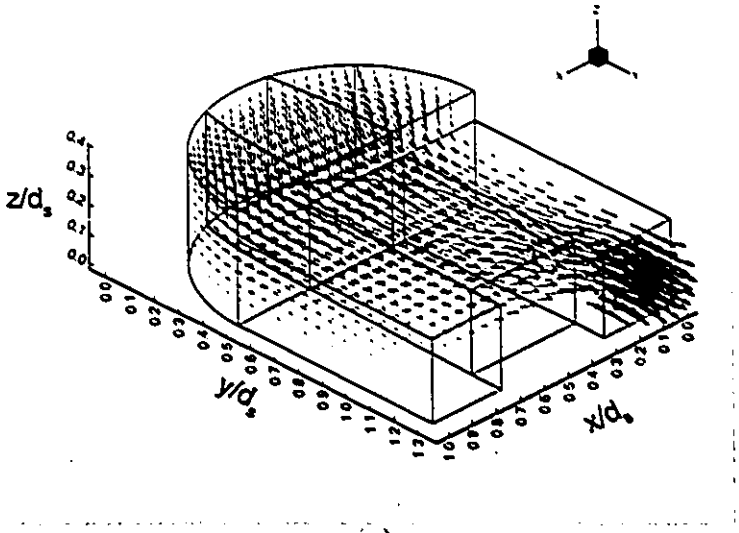


(b)

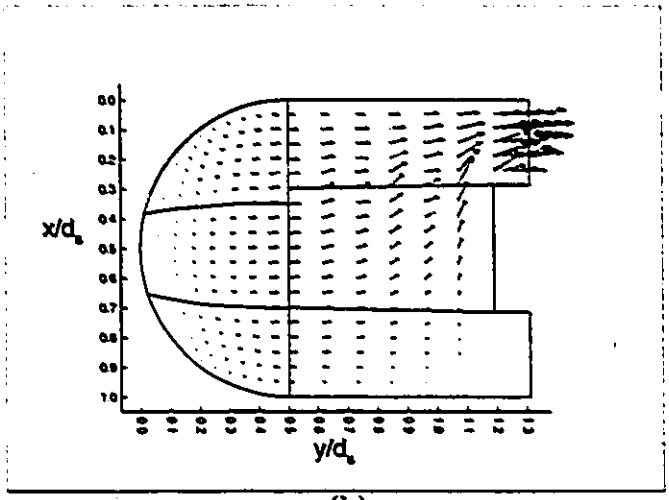


(c)

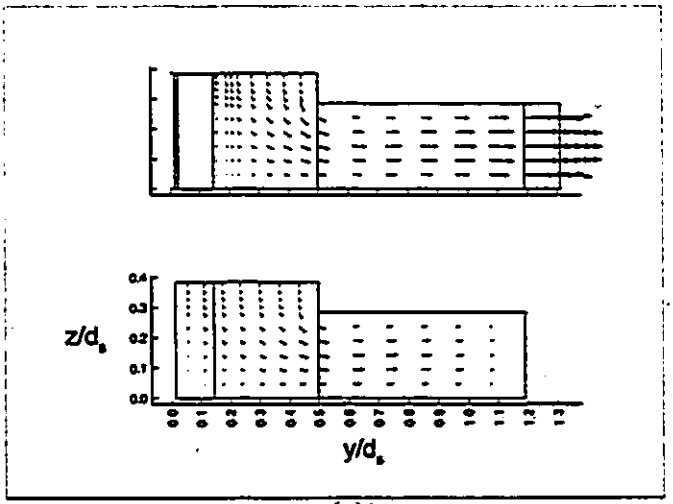
Figure 4.14: (a) Desired and optimal flow rates; (b) variation of $Q - Q^d$ and (c) convergence rates in examples 1-3.



(a)

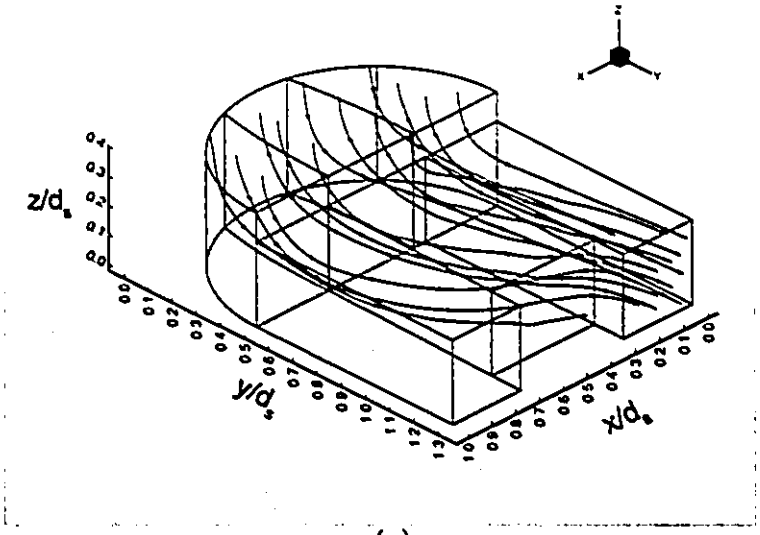


(b)

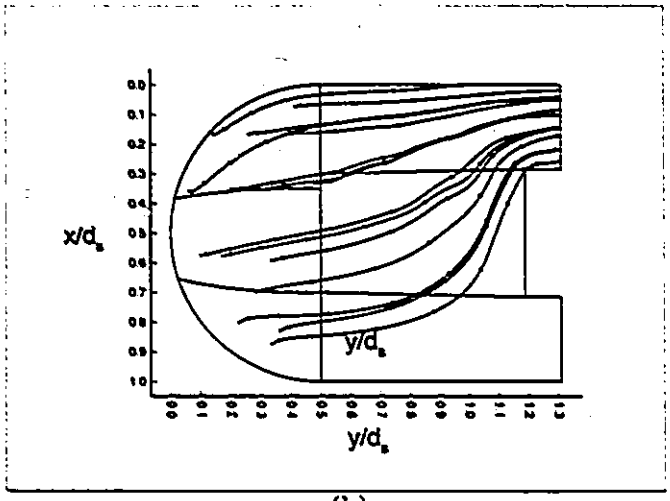


(c)

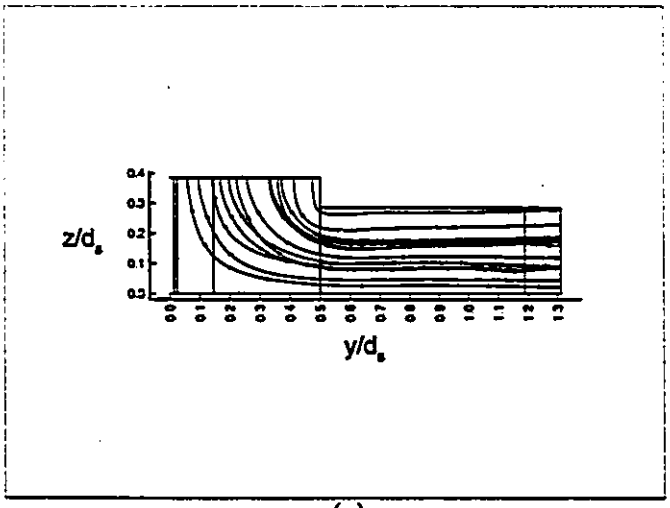
Figure 4.15: Velocity vector plot in systolic peak in example 1; (a) three-dimensional view; (b) z=0.15 plane; (c) x=0.15 and x=0.5 planes.



(a)

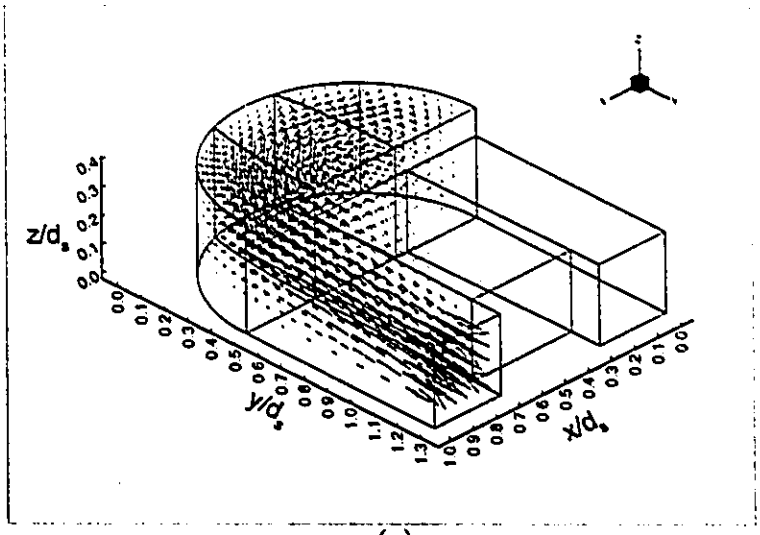


(b)

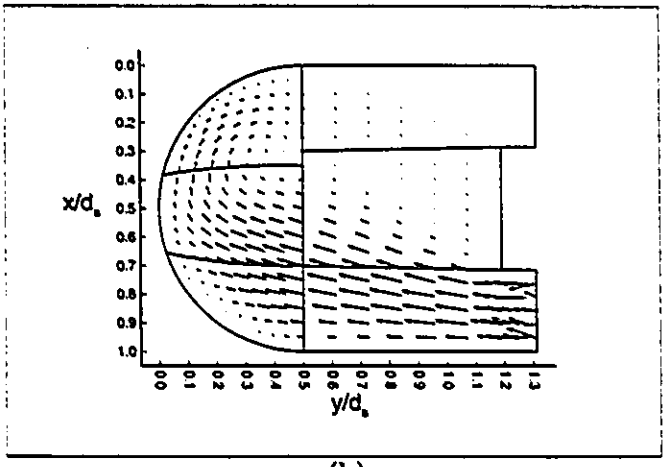


(c)

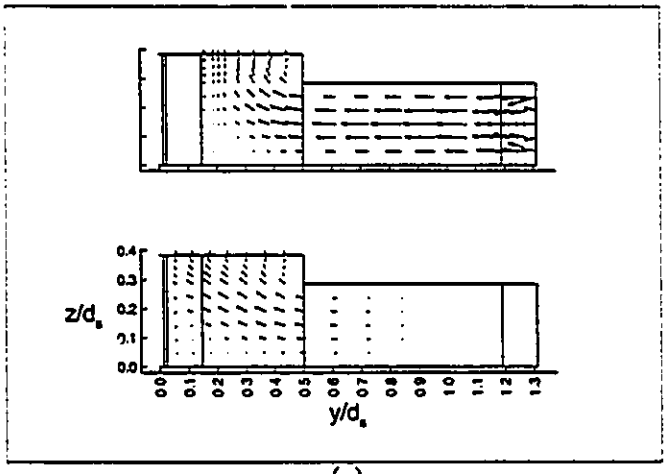
Figure 4.16: Streamlines in systolic peak in example 1; (a) three-dimensional view; (b) top view; (c) side view.



(a)

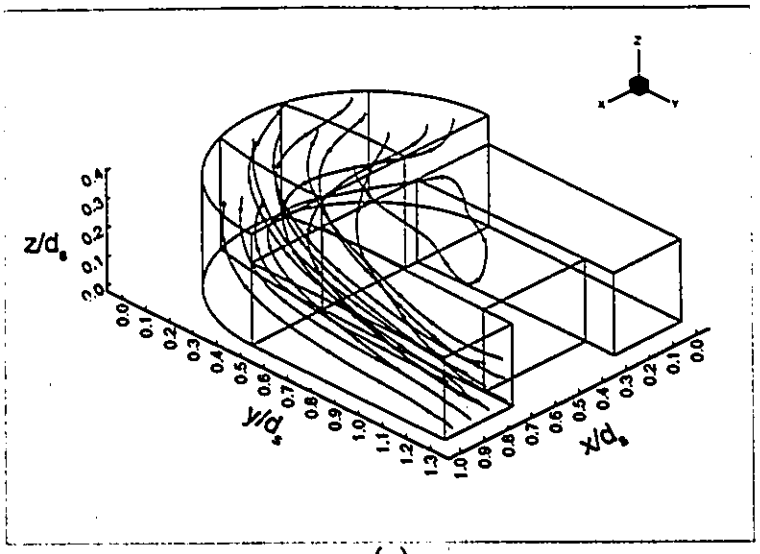


(b)

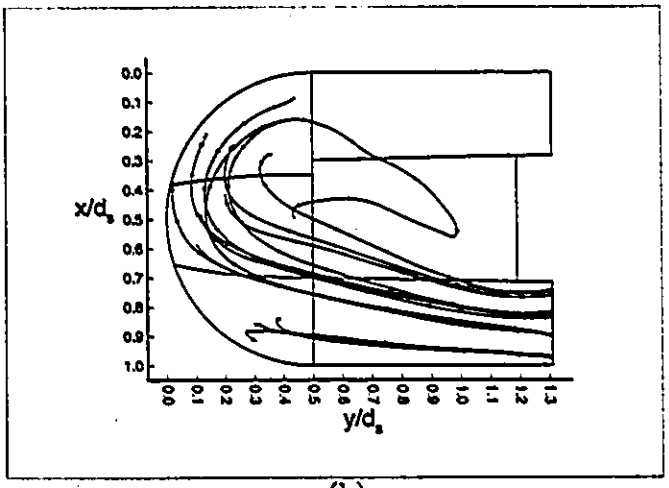


(c)

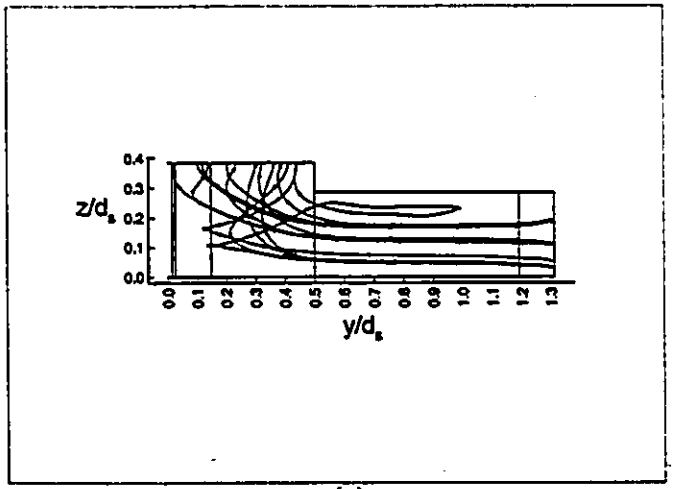
Figure 4.17: Velocity vector plot in diastolic peak in example 1; (a) three-dimensional view; (b) z=0.15 plane; (c) x=0.5 and x=0.15 planes.



(a)

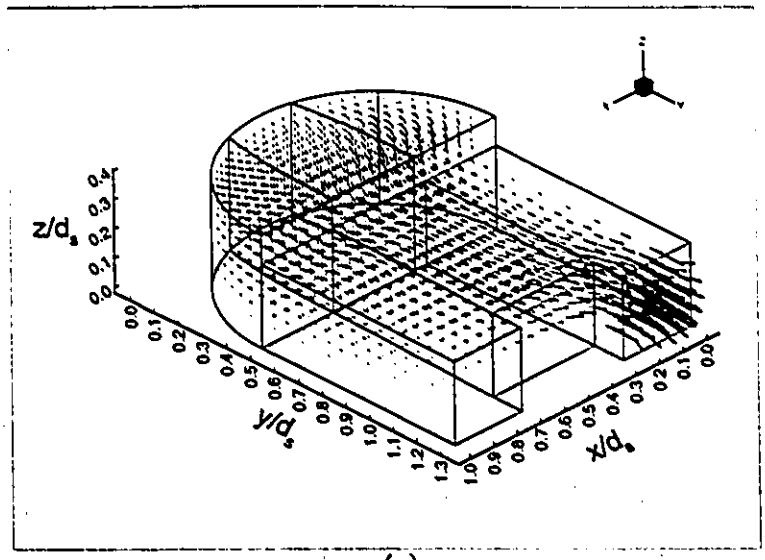


(b)

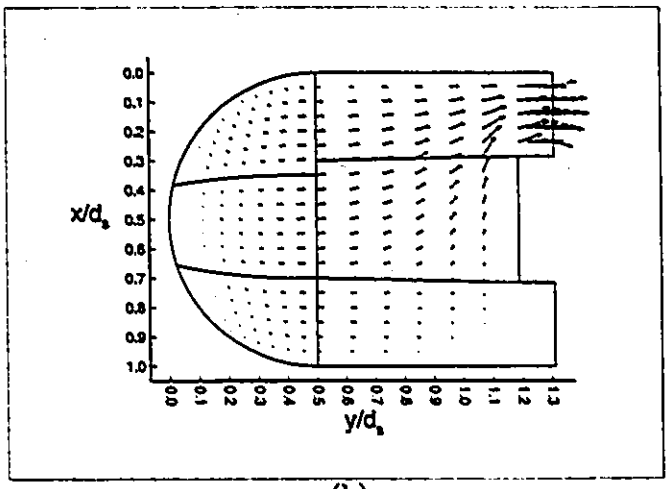


(c)

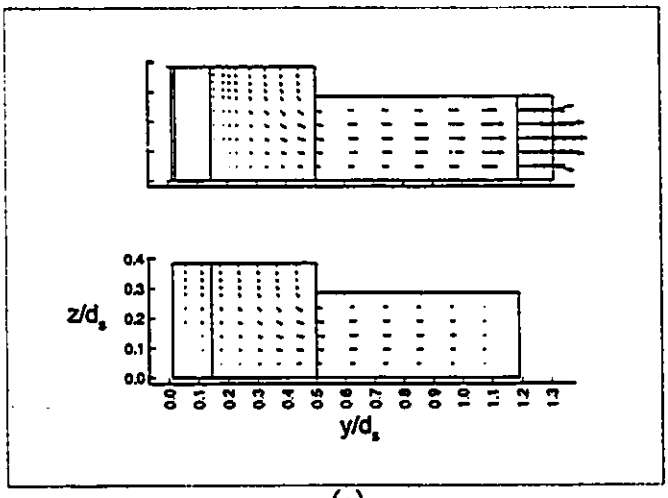
Figure 4.18: Streamlines in diastolic peak in example 1; (a) three-dimensional view; (b) top view; (c) side view.



(a)

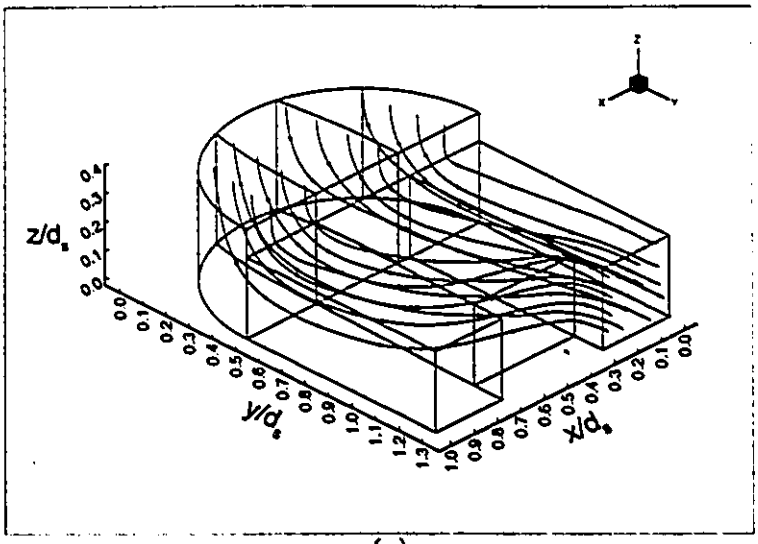


(b)

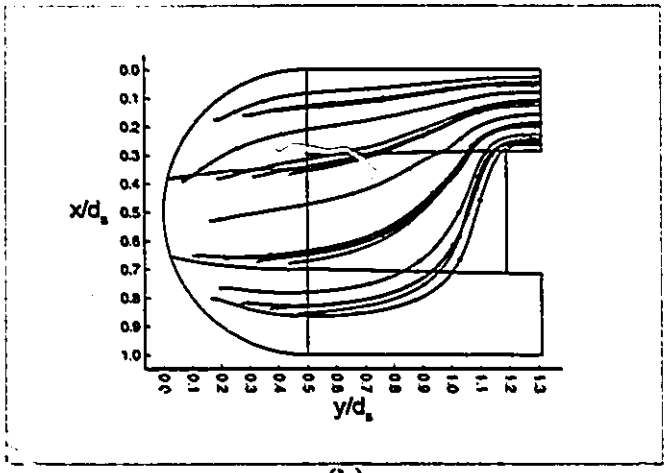


(c)

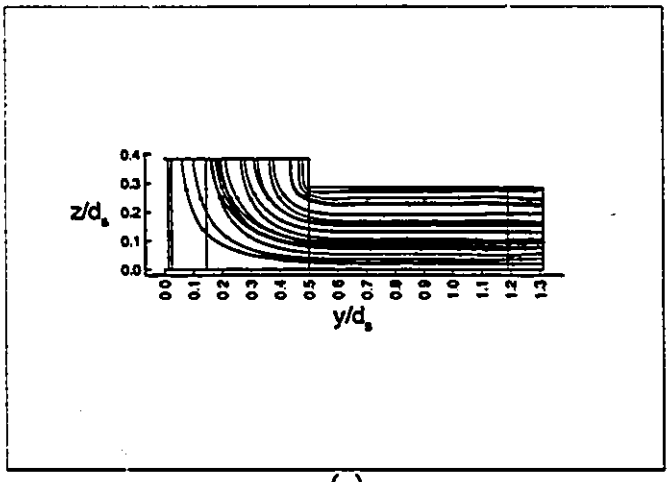
Figure 4.19: Velocity vector plot in systolic peak in example 3; (a) three-dimensional view; (b) z=0.15 plane; (c) x=0.15 and x=0.5 planes.



(a)

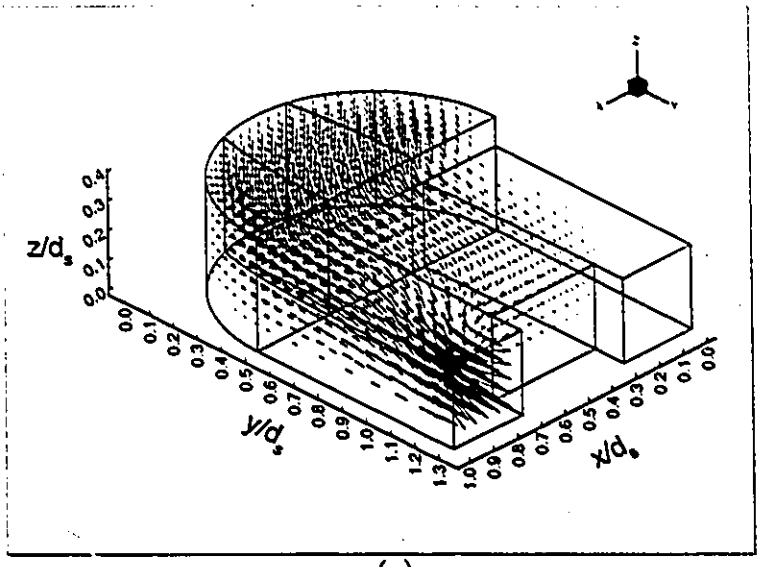


(b)

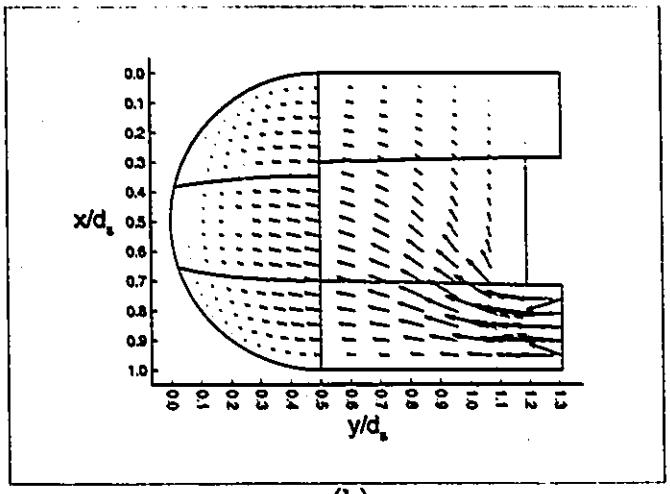


(c)

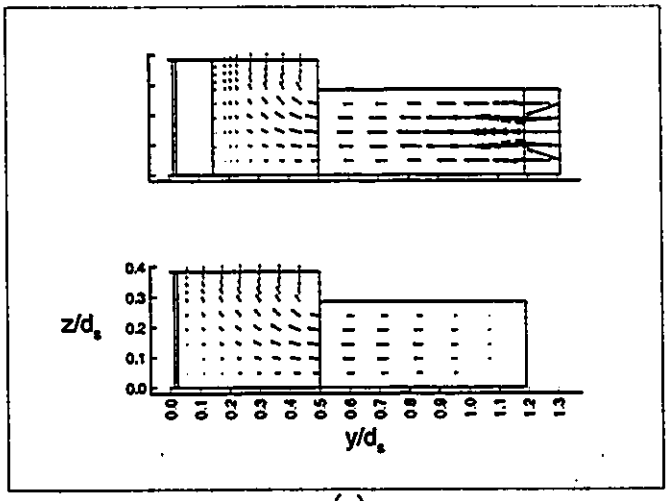
Figure 4.20: Streamlines in systolic peak in example 3; (a) three-dimensional view; (b) top view; (c) side view.



(a)

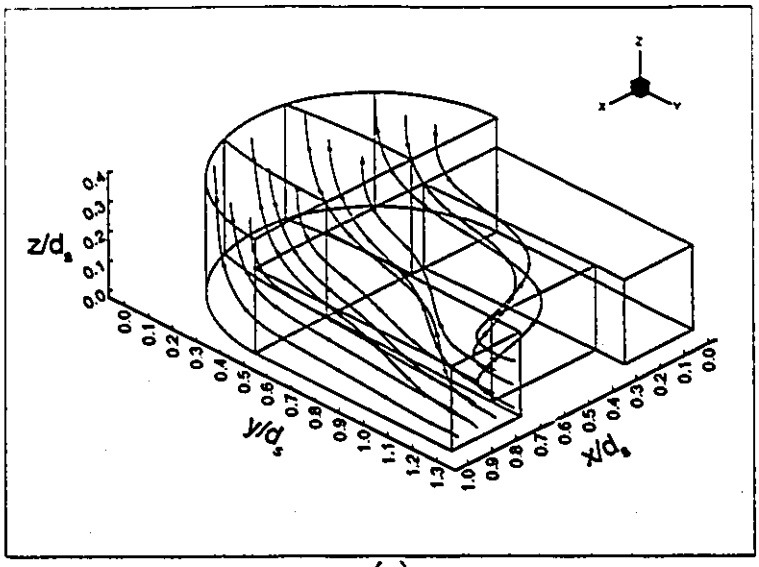


(b)

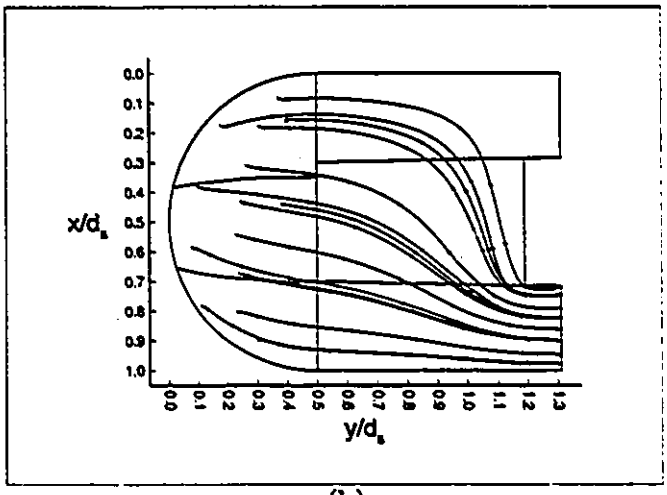


(c)

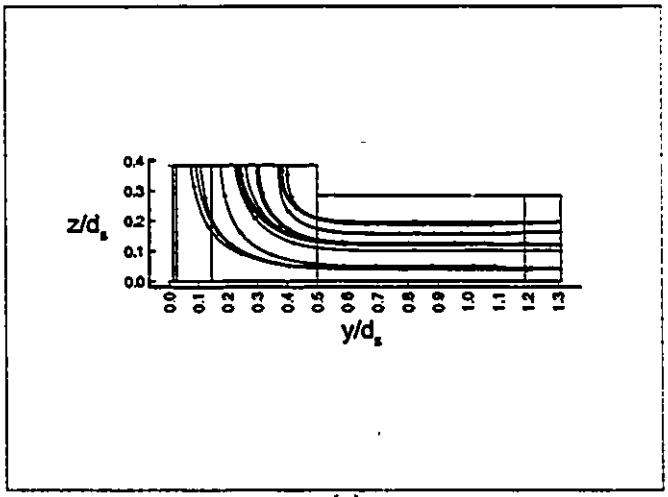
Figure 4.21: Velocity vector plot in diastolic peak in example 3; (a) three-dimensional view; (b) z=0.15 plane; (c) x=0.5 and x=0.15 planes.



(a)



(b)



(c)

Figure 4.22: Streamlines in diastolic peak in example 3; (a) three-dimensional view; (b) top view; (c) side view.

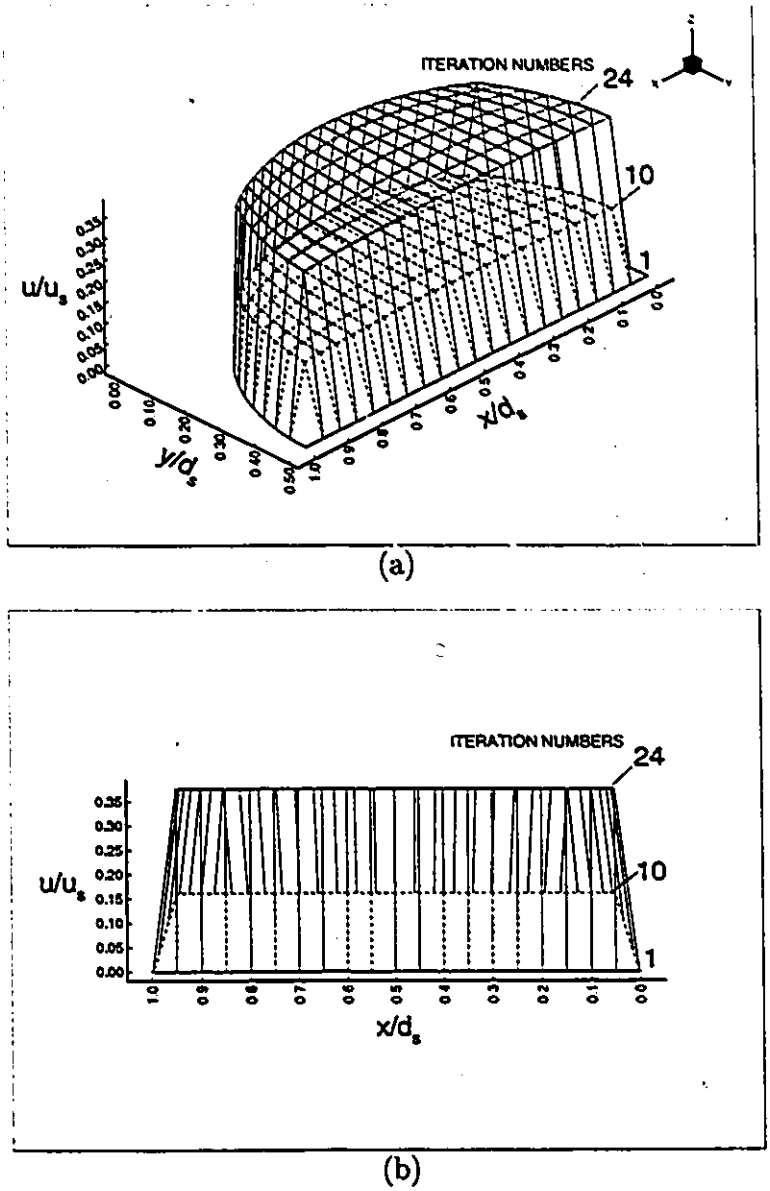


Figure 4.23: Evolution of the boundary control profile at diastolic peak with design iteration in example 1; (a) three-dimensional view; (b) side view.

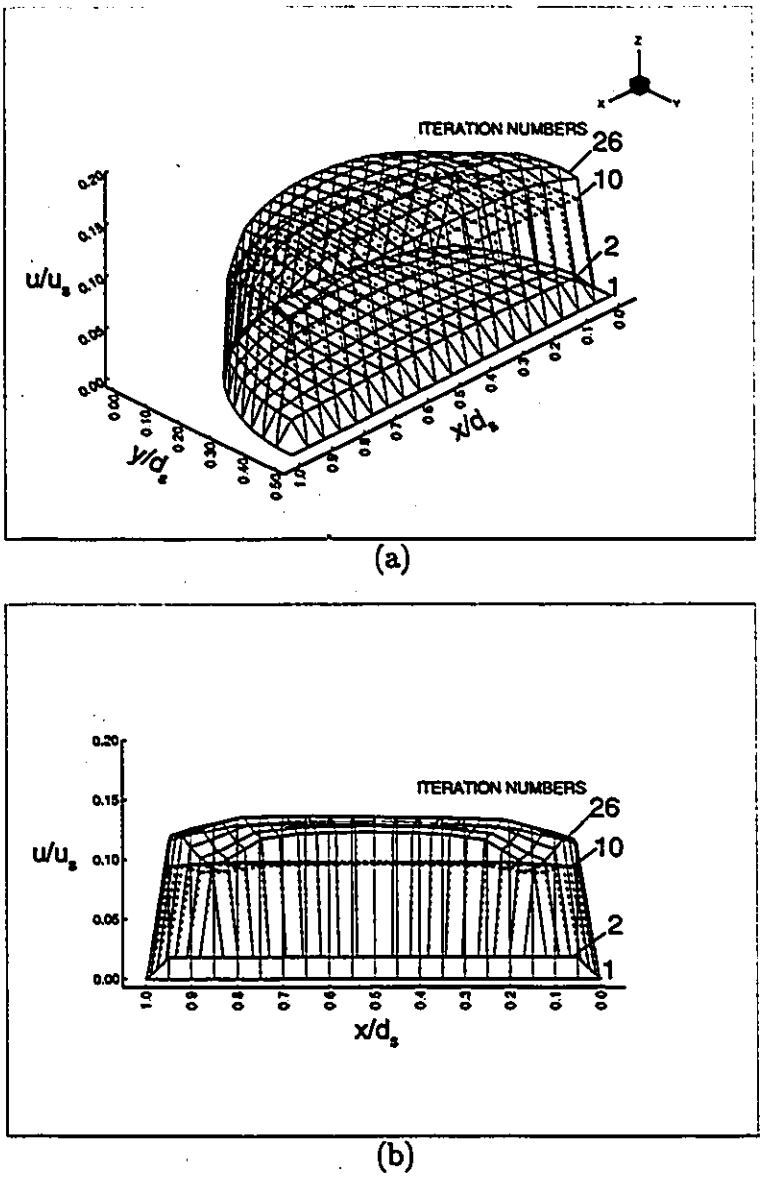


Figure 4.24: Evolution of the boundary control profile at diastolic peak with design iteration in example 2; (a) three-dimensional view; (b) side view.

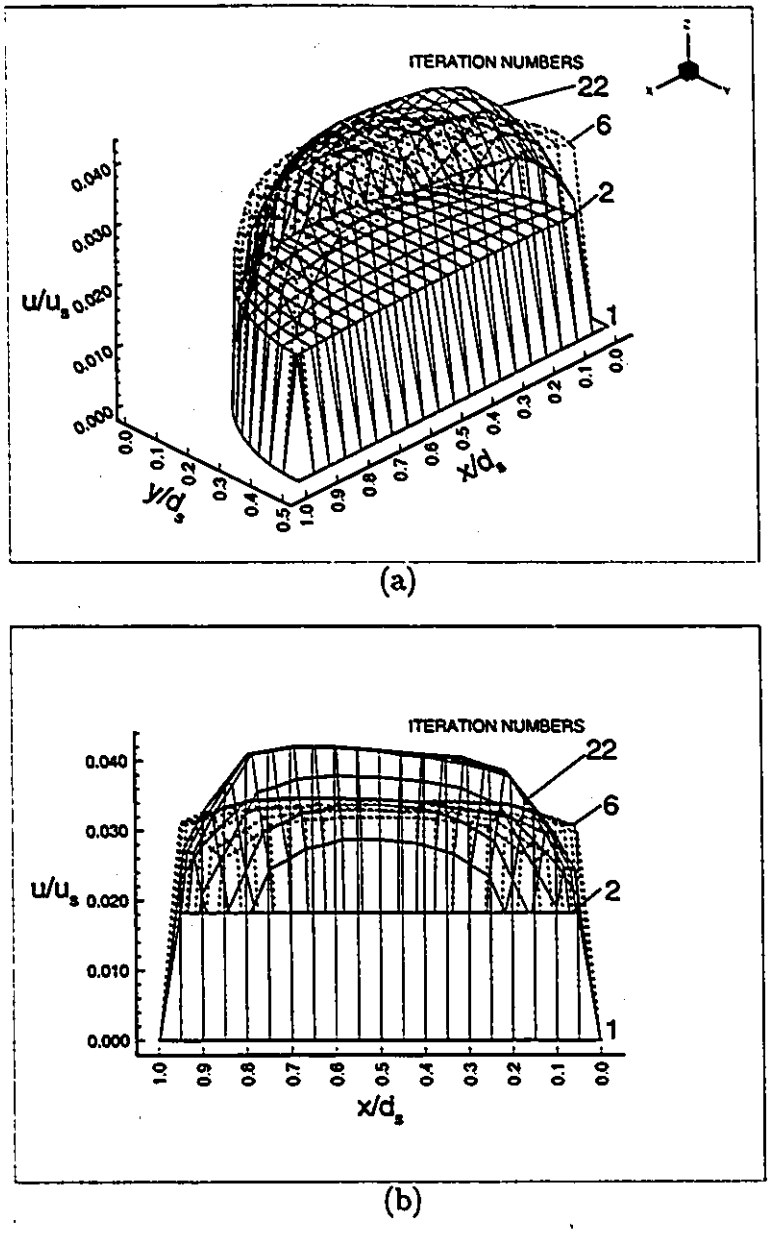


Figure 4.25: Evolution of the boundary control profile at diastolic peak with design iteration in example 3; (a) three-dimensional view; (b) side view.

Appendix A

Derivations in the Optimization Algorithm

A.1 Derivation of the Gradient of the Cost Functional

We start from the expression for the Gateaux derivative of the cost functional in equation (2.21) as

$$J'(\mathbf{v}, \mathbf{v}-\mathbf{v}^o) = \int_I \int_\Omega \{-\frac{\partial \mathbf{z}}{\partial t} - (\mathbf{u}\cdot\nabla)\mathbf{z} + (\nabla\mathbf{u})^T\mathbf{z} - \nu\nabla^2\mathbf{z} + \frac{1}{\rho}\nabla q_z\} \cdot \mathbf{w}\, d\Omega dt + \int_I \int_\Omega \mathbf{G}^2 \cdot (\mathbf{v}-\mathbf{v}^o)\, d\Omega dt. \tag{A.1}$$

in which the superscript ^{o} for $\mathbf{z}$ is dropped for convenience. The first term in the first integral of this equation will take the following form

$$\begin{aligned} \int_I \int_\Omega -\frac{\partial \mathbf{z}}{\partial t} \cdot \mathbf{w}\, d\Omega dt &= \int_\Omega -\mathbf{z}\cdot\mathbf{w}\, d\Omega|_0^T + \int_I \int_\Omega \frac{\partial \mathbf{w}}{\partial t} \cdot \mathbf{z}\, d\Omega dt \\ &= \int_I \int_\Omega \frac{\partial \mathbf{w}}{\partial t} \cdot \mathbf{z}\, d\Omega dt \end{aligned} \tag{A.2}$$

(we note that $\mathbf{z}(T) = \mathbf{w}(0) = 0$ from equations (2.10) and (2.18), respectively).

For the second term of the first integral we have

$$\begin{aligned}
\int_I\int_\Omega -(\mathbf{u}\cdot\nabla)\mathbf{z}\cdot\mathbf{w}\,d\Omega dt &= -\int_I\{\int_\Omega(u_1\frac{\partial z_1}{\partial x_1}w_1+u_2\frac{\partial z_1}{\partial x_2}w_1+u_3\frac{\partial z_1}{\partial x_3}w_1\\
&\quad+u_1\frac{\partial z_2}{\partial x_1}w_2+u_2\frac{\partial z_2}{\partial x_2}w_2+u_3\frac{\partial z_2}{\partial x_3}w_2\\
&\quad+u_1\frac{\partial z_3}{\partial x_1}w_3+u_2\frac{\partial z_3}{\partial x_2}w_3+u_3\frac{\partial z_3}{\partial x_3}w_3\}\,d\Omega dt\\
&= \int_I\{\int_\Omega(u_1\frac{\partial w_1}{\partial x_1}z_1+u_2\frac{\partial w_1}{\partial x_2}z_1+u_3\frac{\partial w_1}{\partial x_3}z_1\\
&\quad+u_1\frac{\partial w_2}{\partial x_1}z_2+u_2\frac{\partial w_2}{\partial x_2}z_2+u_3\frac{\partial w_2}{\partial x_3}z_2\\
&\quad+u_1\frac{\partial w_3}{\partial x_1}z_3+u_2\frac{\partial w_3}{\partial x_2}z_3+u_3\frac{\partial w_3}{\partial x_3}z_3)\,d\Omega\\
&\quad+\int_\Omega(w_1\frac{\partial u_1}{\partial x_1}z_1+w_1\frac{\partial u_2}{\partial x_2}z_1+w_1\frac{\partial u_3}{\partial x_3}z_1\\
&\quad+w_2\frac{\partial u_1}{\partial x_1}z_2+w_2\frac{\partial u_2}{\partial x_2}z_2+w_2\frac{\partial u_3}{\partial x_3}z_2\\
&\quad+w_3\frac{\partial u_1}{\partial x_1}z_3+w_3\frac{\partial u_2}{\partial x_2}z_3+w_3\frac{\partial u_3}{\partial x_3}z_3)d\Omega\\
&\quad-\int_\Gamma(u_1n_1w_1z_1+u_2n_2w_1z_1+u_3n_3w_1z_1\\
&\quad+u_1n_1w_2z_2+u_2n_2w_2z_2+u_3n_3w_2z_2\\
&\quad+u_1n_1w_3z_3+u_2n_2w_3z_3+u_3n_3w_3z_3)ds\}dt\\
&= \int_I\{\int_\Omega(\mathbf{u}\cdot\nabla)\mathbf{w}\cdot\mathbf{z}\,d\Omega-\int_\Gamma(\mathbf{u}\cdot\mathbf{n})\mathbf{w}\cdot\mathbf{z}\,ds\\
&\quad+\int_\Omega(\nabla\cdot\mathbf{u})\mathbf{w}\cdot\mathbf{z}\,d\Omega\}dt\\
&= \int_I\{\int_\Omega(\mathbf{u}\cdot\nabla)\mathbf{w}\cdot\mathbf{z}\,d\Omega-\int_\Gamma(\mathbf{u}\cdot\mathbf{n})\mathbf{w}\cdot\mathbf{z}\,ds\}dt. \quad \text{(A.3)}
\end{aligned}$$

The third term of the first integral would take the following form

$$\begin{aligned}
\int_I\int_\Omega(\nabla\mathbf{u})^T\mathbf{z}\cdot\mathbf{w}\,d\Omega dt &= \int_I\int_\Omega\{(z_1\frac{\partial u_1}{\partial x_1}+z_2\frac{\partial u_2}{\partial x_1}+z_3\frac{\partial u_3}{\partial x_1})w_1\\
&\quad+(z_1\frac{\partial u_1}{\partial x_2}+z_2\frac{\partial u_2}{\partial x_2}+z_3\frac{\partial u_3}{\partial x_2})w_2\\
&\quad+(z_1\frac{\partial u_1}{\partial x_3}+z_2\frac{\partial u_2}{\partial x_3}+z_3\frac{\partial u_3}{\partial x_3})w_3\}\,d\Omega dt\\
&= \int_I\int_\Omega\{(w_1\frac{\partial u_1}{\partial x_1}+w_2\frac{\partial u_1}{\partial x_2}+w_3\frac{\partial u_1}{\partial x_3})z_1\\
&\quad+(w_1\frac{\partial u_2}{\partial x_1}+w_2\frac{\partial u_2}{\partial x_2}+w_3\frac{\partial u_2}{\partial x_3})z_2\\
&\quad+(w_1\frac{\partial u_3}{\partial x_1}+w_2\frac{\partial u_3}{\partial x_2}+w_3\frac{\partial u_3}{\partial x_3})z_3\}\,d\Omega dt
\end{aligned}$$

$$= \int_I \int_\Omega (\mathbf{w} \cdot \nabla)\mathbf{u} \cdot \mathbf{z} \; d\Omega dt. \tag{A.4}$$

For the fourth term of the first integral we have

$$\begin{aligned}
\int_I \int_\Omega -\nu \Delta \mathbf{z} \cdot \mathbf{w} \; d\Omega dt &= \int_I \int_\Omega -\nu \nabla^2 \mathbf{z} \cdot \mathbf{w} \; d\Omega dt \\
&= -\nu \int_I \int_\Omega \{(\frac{\partial^2 z_1}{\partial x_1^2} w_1 + \frac{\partial^2 z_1}{\partial x_2^2} w_1 + \frac{\partial^2 z_1}{\partial x_3^2} w_1) \\
&\quad + (\frac{\partial^2 z_2}{\partial x_1^2} w_2 + \frac{\partial^2 z_2}{\partial x_2^2} w_2 + \frac{\partial^2 z_2}{\partial x_3^2} w_2) \\
&\quad + (\frac{\partial^2 z_3}{\partial x_1^2} w_3 + \frac{\partial^2 z_3}{\partial x_2^2} w_3 + \frac{\partial^2 z_3}{\partial x_3^2} w_3)\} \; d\Omega dt \\
&= -\nu \int_I \{ - \int_\Omega \frac{\partial z_1}{\partial x_1} \frac{\partial w_1}{\partial x_1} d\Omega - \int_\Omega \frac{\partial z_1}{\partial x_2} \frac{\partial w_1}{\partial x_2} d\Omega - \int_\Omega \frac{\partial z_1}{\partial x_3} \frac{\partial w_1}{\partial x_3} d\Omega \\
&\quad + \int_\Gamma n_1 w_1 \frac{\partial z_1}{\partial x_1} ds + \int_\Gamma n_2 w_1 \frac{\partial z_1}{\partial x_2} ds + \int_\Gamma n_3 w_1 \frac{\partial z_1}{\partial x_3} ds \\
&\quad - \int_\Omega \frac{\partial z_2}{\partial x_1} \frac{\partial w_2}{\partial x_1} d\Omega - \int_\Omega \frac{\partial z_2}{\partial x_2} \frac{\partial w_2}{\partial x_2} d\Omega - \int_\Omega \frac{\partial z_2}{\partial x_3} \frac{\partial w_2}{\partial x_3} d\Omega \\
&\quad + \int_\Gamma n_1 w_2 \frac{\partial z_2}{\partial x_1} ds + \int_\Gamma n_2 w_2 \frac{\partial z_2}{\partial x_2} ds + \int_\Gamma n_3 w_2 \frac{\partial z_2}{\partial x_3} ds \\
&\quad - \int_\Omega \frac{\partial z_3}{\partial x_1} \frac{\partial w_3}{\partial x_1} d\Omega - \int_\Omega \frac{\partial z_3}{\partial x_2} \frac{\partial w_3}{\partial x_2} d\Omega - \int_\Omega \frac{\partial z_3}{\partial x_3} \frac{\partial w_3}{\partial x_3} d\Omega \\
&\quad + \int_\Gamma n_1 w_3 \frac{\partial z_3}{\partial x_1} ds + \int_\Gamma n_2 w_3 \frac{\partial z_3}{\partial x_2} ds + \int_\Gamma n_3 w_3 \frac{\partial z_3}{\partial x_3} ds\} dt \\
&= -\nu \int_I \{ \int_\Omega z_1 \frac{\partial^2 w_1}{\partial x_1^2} d\Omega - \int_\Gamma n_1 \frac{\partial w_1}{\partial x_1} z_1 ds + \int_\Gamma n_1 w_1 \frac{\partial z_1}{\partial x_1} ds \\
&\quad + \int_\Omega z_1 \frac{\partial^2 w_1}{\partial x_2^2} d\Omega - \int_\Gamma n_2 \frac{\partial w_1}{\partial x_2} z_1 ds + \int_\Gamma n_2 w_1 \frac{\partial z_1}{\partial x_2} ds \\
&\quad + \int_\Omega z_1 \frac{\partial^2 w_1}{\partial x_3^2} d\Omega - \int_\Gamma n_3 \frac{\partial w_1}{\partial x_3} z_1 ds + \int_\Gamma n_3 w_1 \frac{\partial z_1}{\partial x_3} ds \\
&\quad + \int_\Omega z_2 \frac{\partial^2 w_2}{\partial x_1^2} d\Omega - \int_\Gamma n_1 \frac{\partial w_2}{\partial x_1} z_2 ds + \int_\Gamma n_1 w_2 \frac{\partial z_2}{\partial x_1} ds \\
&\quad + \int_\Omega z_2 \frac{\partial^2 w_2}{\partial x_2^2} d\Omega - \int_\Gamma n_2 \frac{\partial w_2}{\partial x_2} z_2 ds + \int_\Gamma n_2 w_2 \frac{\partial z_2}{\partial x_2} ds \\
&\quad + \int_\Omega z_2 \frac{\partial^2 w_2}{\partial x_3^2} d\Omega - \int_\Gamma n_3 \frac{\partial w_2}{\partial x_3} z_2 ds + \int_\Gamma n_3 w_2 \frac{\partial z_2}{\partial x_3} ds \\
&\quad + \int_\Omega z_3 \frac{\partial^2 w_3}{\partial x_1^2} d\Omega - \int_\Gamma n_1 \frac{\partial w_3}{\partial x_1} z_3 ds + \int_\Gamma n_1 w_3 \frac{\partial z_3}{\partial x_1} ds \\
&\quad + \int_\Omega z_3 \frac{\partial^2 w_3}{\partial x_2^2} d\Omega - \int_\Gamma n_2 \frac{\partial w_3}{\partial x_2} z_3 ds + \int_\Gamma n_2 w_3 \frac{\partial z_3}{\partial x_2} ds \\
&\quad + \int_\Omega z_3 \frac{\partial^2 w_3}{\partial x_3^2} d\Omega - \int_\Gamma n_3 \frac{\partial w_3}{\partial x_3} z_3 ds + \int_\Gamma n_3 w_3 \frac{\partial z_3}{\partial x_3} ds\} dt
\end{aligned}$$

$$
\begin{aligned}
&= -\nu \int_I \{ \int_\Omega \Delta \mathbf{w} \cdot \mathbf{z} \, d\Omega + \int_\Gamma [\, w_1 (n_1 \frac{\partial z_1}{\partial x_1} + n_2 \frac{\partial z_1}{\partial x_2} + n_3 \frac{\partial z_1}{\partial x_3}) \\
&\quad + w_2 (n_1 \frac{\partial z_2}{\partial x_1} + n_2 \frac{\partial z_2}{\partial x_2} + n_3 \frac{\partial z_2}{\partial x_3}) \\
&\quad + w_3 (n_1 \frac{\partial z_3}{\partial x_1} + n_2 \frac{\partial z_3}{\partial x_2} + n_3 \frac{\partial z_3}{\partial x_3}) \,] ds \\
&\quad - \int_\Gamma [\, z_1 (n_1 \frac{\partial w_1}{\partial x_1} + n_2 \frac{\partial w_1}{\partial x_2} + n_3 \frac{\partial w_1}{\partial x_3}) \\
&\quad + z_2 (n_1 \frac{\partial w_2}{\partial x_1} + n_2 \frac{\partial w_2}{\partial x_2} + n_3 \frac{\partial w_2}{\partial x_3}) \\
&\quad + z_3 (n_1 \frac{\partial w_3}{\partial x_1} + n_2 \frac{\partial w_3}{\partial x_2} + n_3 \frac{\partial w_3}{\partial x_3}) \} ds] dt \\
&= -\nu \int_I \{ \int_\Omega \Delta \mathbf{w} \cdot \mathbf{z} \, d\Omega + \int_\Gamma \mathbf{w} \cdot (\mathbf{n} \cdot \nabla) \mathbf{z} ds \\
&\quad - \int_\Gamma \mathbf{z} \cdot (\mathbf{n} \cdot \nabla) \mathbf{w} \, ds \} dt. \qquad \text{(A.5)}
\end{aligned}
$$

Finally, for the last term of the first integral we have

$$
\begin{aligned}
\int_I \int_\Omega \nabla q_z \cdot \mathbf{w} \, d\Omega dt &= \int_I \int_\Omega (w_1 \frac{\partial q_z}{\partial x_1} + w_2 \frac{\partial q_z}{\partial x_2} + w_3 \frac{\partial q_z}{\partial x_3}) d\Omega dt \\
&= -\int_I \int_\Omega q_z (\frac{\partial w_1}{\partial x_1} + \frac{\partial w_2}{\partial x_2} + \frac{\partial w_3}{\partial x_3}) d\Omega dt \\
&\quad + \int_I \int_\Gamma q_z (n_1 w_1 + n_2 w_2 + n_3 w_3) ds dt \\
&= -\int_I \int_\Omega (\nabla \cdot \mathbf{w}) q_z \, d\Omega dt + \int_I \int_\Gamma q_z \mathbf{n} \cdot \mathbf{w} \, ds dt. \qquad \text{(A.6)}
\end{aligned}
$$

Since $\nabla \cdot \mathbf{w} = 0$ (see equation (2.18)), then

$$
\int_I \int_\Omega \nabla q_z \cdot \mathbf{w} \, d\Omega dt = \int_I \int_\Gamma q_z \mathbf{n} \cdot \mathbf{w} ds dt. \qquad \text{(A.7)}
$$

Combining the above results for the various terms would result in

$$
\begin{aligned}
J'(\mathbf{v}, \mathbf{v} - \mathbf{v}^o) &= \\
&\int_I \int_\Omega \{ \frac{\partial \mathbf{w}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{w} + (\mathbf{w} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{w} \} \cdot \mathbf{z} \, d\Omega dt \\
&+ \int_I \int_\Gamma \{ -(\mathbf{u} \cdot \mathbf{n}) \mathbf{w} \cdot \mathbf{z} + \nu (\mathbf{n} \cdot \nabla) \mathbf{w} \cdot \mathbf{z} - \nu (\mathbf{n} \cdot \nabla) \mathbf{z} \cdot \mathbf{w} + q_z \mathbf{n} \cdot \mathbf{w} \} ds dt \\
&+ \int_I \int_\Omega \mathbf{G}^2 . (\mathbf{v} - \mathbf{v}^o) d\Omega dt. \qquad \text{(A.8)}
\end{aligned}
$$

Comparing the first integral in the above equations with the first equation in (2.18), would result in the following form for this integral

$$
\int_I \int_\Omega \{ \frac{\partial \mathbf{w}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{w} + (\mathbf{w} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{w} \} \cdot \mathbf{z} \, d\Omega dt = \frac{1}{\rho} \int_I \int_\Omega -(\nabla q_w) \cdot \mathbf{z} \, d\Omega dt
$$

$$
\begin{aligned}
&= \frac{1}{\rho}\int_I\int_\Omega -(z_1\frac{\partial q_w}{\partial x_1} + z_2\frac{\partial q_w}{\partial x_2} + z_3\frac{\partial q_w}{\partial x_3})d\Omega dt \\
&= \frac{1}{\rho}\int_I\int_\Omega (q_w\frac{\partial z_1}{\partial x_1} + q_w\frac{\partial z_2}{\partial x_2} + q_w\frac{\partial z_3}{\partial x_3})d\Omega dt \\
&\quad - \frac{1}{\rho}\int_I\int_\Gamma (n_1 q_w z_1 + n_2 q_w z_2 + n_3 q_w z_3)dsdt \\
&= \frac{1}{\rho}\int_I\int_\Omega q_w(\nabla\cdot\mathbf{z})d\Omega dt - \frac{1}{\rho}\int_I\int_\Gamma q_w(\mathbf{n}\cdot\mathbf{z})dsdt. \qquad \text{(A.9)}
\end{aligned}
$$

Since $\nabla\cdot\mathbf{z} = 0$ and $\mathbf{z}|_\Gamma = 0$ (see equations (2.10)), then

$$
\int_I\int_\Omega \{\frac{\partial \mathbf{w}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{w} + (\mathbf{w}\cdot\nabla)\mathbf{u} - \nu\Delta\mathbf{w}\}\cdot\mathbf{z}\; d\Omega dt \;=\; 0. \qquad \text{(A.10)}
$$

Now consider the second integral in equation (A.8). The first and the second terms in this integral are zero because of the condition $\mathbf{z}|_\Gamma = 0$ in equation (2.10). For the third term of this integral, we have

$$
\int_I\int_\Gamma -\nu(\mathbf{n}\cdot\nabla)\mathbf{z}\cdot\mathbf{w}\; dsdt = \int_I\int_{\Gamma_m} -\nu(\mathbf{n}\cdot\nabla)\mathbf{z}\cdot(\mathbf{v}-\mathbf{v}^o)dsdt \qquad \text{(A.11)}
$$

because of the condition $\mathbf{w}|_{\Gamma_f} = \mathbf{w}|_{\Gamma_s} = \mathbf{w}|_{\Gamma_o}$ =0 and $\mathbf{w}|_{\Gamma_m} = \mathbf{v}-\mathbf{v}^o$ (see equation (2.18)). Similarly, because of these boundary conditions for $\mathbf{w}$, the last term of the second integral in equation (A.8) will have the following form

$$
\int_I\int_\Gamma q_z\mathbf{n}\cdot\mathbf{w}dsdt = \int_I\int_{\Gamma_m} q_z\mathbf{n}\cdot(\mathbf{v}-\mathbf{v}^o)dsdt. \qquad \text{(A.12)}
$$

Regarding the last integral in equation (A.8), we use the result in the next section that $\mathbf{G}^2 = \mathbf{v}$, where the control $\mathbf{v}$ is defined only on the boundary Γ_m. This results in restricting this volume integral to the boundary Γ_m. Using the above results and noting that $J'(\mathbf{v}-\mathbf{v}^o) \geq 0$ would result in an inequality for the Gateaux derivative of the cost functional as

$$
J'(\mathbf{v}, \mathbf{v}-\mathbf{v}^o) = \int_I\int_{\Gamma_m} \left\{\left(-\nu(\mathbf{n}\cdot\nabla)\mathbf{z}^o + q_z\cdot\mathbf{n} + \mathbf{G}^2\right)\cdot(\mathbf{v}-\mathbf{v}^o)\right\}dsdt \geq 0; \quad \forall\, \mathbf{v}\in U_{ad}. \qquad \text{(A.13)}
$$

A.2 Derivation of G^1 and G^2

A.2.1 Derivation of G^1

We consider the cost functional in its general form (equation (2.6) or (2.7)) as

$$\begin{aligned} J(\mathbf{u},\mathbf{v}) &= \frac{1}{2}\int_I\int_\Omega\{R_1(\nabla\times\mathbf{u})+R_2(\nabla\mathbf{u})+R_3(\mathbf{u}-\mathbf{u}^d)+R_4(\mathbf{v})\}d\Omega dt \\ &= \int_I g(\mathbf{u},\mathbf{v})dt \\ &= \int_I\{g_1(\mathbf{u},\mathbf{v})+g_2(\mathbf{u},\mathbf{v})+g_3(\mathbf{u},\mathbf{v})+g_4(\mathbf{u},\mathbf{v})\}dt \end{aligned} \tag{A.14}$$

in which R is a quadratic function and $g = g_1 + g_2 + g_3 + g_4$ is the integrand of the cost functional. In the following, we shall restrict our discussion to the case of boundary control problem with the cost functional (2.7). Then, the integrands $g_1 - g_4$ will have the following form

$$g_1(\mathbf{u},\mathbf{v}) = \frac{1}{2}\int_\Omega R_1(\nabla\times\mathbf{u})d\Omega, \tag{A.15}$$

$$g_2(\mathbf{u},\mathbf{v}) = \frac{1}{2}\int_\Omega R_2(\nabla\mathbf{u})d\Omega, \tag{A.16}$$

$$g_3(\mathbf{u},\mathbf{v}) = \frac{1}{2}R_3(\int_{\Gamma_m}\mathbf{u}ds - Q^d) \tag{A.17}$$

and

$$g_4(\mathbf{u},\mathbf{v}) = \frac{1}{2}\int_\Omega R_4(\mathbf{v})d\Omega. \tag{A.18}$$

We recall the following definitions

$\mathbf{G^1} = \mathbf{G^1_{g_1}} + \mathbf{G^1_{g_2}} + \mathbf{G^1_{g_3}} + \mathbf{G^1_{g_4}}$: Gateaux derivative of the integrand of the cost functional, g, with respect to $\mathbf{u}$ and

$\mathbf{G^2} = \mathbf{G^2_{g_1}} + \mathbf{G^2_{g_2}} + \mathbf{G^2_{g_3}} + \mathbf{G^2_{g_4}}$: Gateaux derivative of the integrand of the cost functional, g, with respect to $\mathbf{v}$.

In the following, we outline the derivation of these derivatives. For brevity, we shall use the following notations

$$\partial_i = \frac{\partial}{\partial x_i}; \quad \partial_{ij} = \frac{\partial^2}{\partial x_i \partial x_j}; \quad i,j = 1,2,3. \tag{A.19}$$

We have the following definition for $\mathbf{G}^1_{g_1} = g_1'(\mathbf{u},\mathbf{v})$

$$(g_1'(\mathbf{u},\mathbf{v}),\mathbf{w}) = \lim_{\epsilon \to 0} \frac{g_1(\mathbf{u}+\epsilon\mathbf{w},\mathbf{v}) - g_1(\mathbf{u},\mathbf{v})}{\epsilon}. \tag{A.20}$$

In general R_1 has the following form

$$R_1(\nabla \times \mathbf{u}) = \sum_{i,j=1}^{3} R_{ij}(\nabla \times \mathbf{u})_i(\nabla \times \mathbf{u})_j. \tag{A.21}$$

Then

$$g_1(\mathbf{u},\mathbf{v}) = \frac{1}{2}\int_\Omega \sum_{i,j=1}^{3} R_{ij}(\nabla \times \mathbf{u})_i(\nabla \times \mathbf{u})_j d\Omega \tag{A.22}$$

and

$$g_1(\mathbf{u}+\epsilon\mathbf{w},\mathbf{v}) = \frac{1}{2}\int_\Omega \sum_{i,j=1}^{3} R_{ij}(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_i(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_j d\Omega. \tag{A.23}$$

We ignore the cross components of R_{ij}. Then

$$\begin{aligned}
&(g_1'(\mathbf{u},\mathbf{v}),\mathbf{w}) \\
&= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \{ \frac{1}{2}\int_\Omega [\, R_{11}(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_1(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_1 \\
&\quad + R_{22}(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_2(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_2 \\
&\quad + R_{33}(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_3(\nabla \times (\mathbf{u}+\epsilon\mathbf{w}))_3 \,]\, d\Omega \\
&\quad - \frac{1}{2}\int_\Omega [\, R_{11}(\nabla \times \mathbf{u})_1(\nabla \times \mathbf{u})_1 + R_{22}(\nabla \times \mathbf{u})_2(\nabla \times \mathbf{u})_2 \\
&\quad + R_{33}(\nabla \times \mathbf{u})_3(\nabla \times \mathbf{u})_3 \,]\, d\Omega \,\} \\
&= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \{ \frac{1}{2}\int_\Omega [\, R_{11}(\partial_2(u_3+\epsilon w_3) - \partial_3(u_2+\epsilon w_2))(\partial_2(u_3+\epsilon w_3) - \partial_3(u_2+\epsilon w_2)) \\
&\quad + R_{22}(\partial_3(u_1+\epsilon w_1) - \partial_1(u_3+\epsilon w_3))(\partial_3(u_1+\epsilon w_1) - \partial_1(u_3+\epsilon w_3)) \\
&\quad - R_{33}(\partial_1(u_2+\epsilon w_2) - \partial_2(u_1+\epsilon w_1))(\partial_1(u_2+\epsilon w_2) - \partial_2(u_1+\epsilon w_1)) \,]\, d\Omega \\
&\quad - \int_\Omega [\, R_{11}(\partial_2 u_3 - \partial_3 u_2)(\partial_2 u_3 - \partial_3 u_2) + R_{22}(\partial_3 u_1 - \partial_1 u_3)(\partial_3 u_1 - \partial_1 u_3) \\
&\quad + R_{33}(\partial_1 u_2 - \partial_2 u_1)(\partial_2 u_3 - \partial_3 u_2) \,]\, d\Omega \,\}.
\end{aligned} \tag{A.24}$$

Skipping the intermediate steps, we will have

$$\begin{aligned}
(g_1'(\mathbf{u},\mathbf{v}),\mathbf{w}) \;=\; & \frac{1}{2}\int_\Omega \{\, R_{11}(\partial_2 u_3 \cdot \partial_2 w_3 - \partial_2 u_3 \cdot \partial_3 w_2 + \partial_2 w_3 \cdot \partial_2 u_3 - \partial_2 w_3 \cdot \partial_3 u_2 \\
& - \partial_3 u_2 \cdot \partial_2 w_3 + \partial_3 u_2 \cdot \partial_3 w_2 - \partial_3 w_2 \cdot \partial_2 u_3 + \partial_3 w_2 \cdot \partial_3 u_2)
\end{aligned}$$

$$
\begin{aligned}
&+ R_{22}(\partial_3 u_1 \cdot \partial_3 w_1 - \partial_3 u_1 \cdot \partial_1 w_3 + \partial_3 w_1 \cdot \partial_3 u_1 - \partial_3 w_1 \cdot \partial_1 u_3 \quad \text{(A.25)}\\
&- \partial_1 u_3 \cdot \partial_3 w_1 + \partial_1 u_3 \cdot \partial_1 w_3 - \partial_1 w_3 \cdot \partial_3 u_1 + \partial_1 w_3 \cdot \partial_1 u_3)\\
&+ R_{33}(\partial_1 u_1 \cdot \partial_1 w_2 - \partial_1 u_2 \cdot \partial_2 w_1 + \partial_1 w_2 \cdot \partial_2 u_1 - \partial_1 w_2 \cdot \partial_2 u_1\\
&- \partial_2 u_1 \cdot \partial_1 w_2 + \partial_2 u_1 \cdot \partial_2 w_1 - \partial_2 w_1 \cdot \partial_1 u_2 + \partial_2 w_1 \cdot \partial_2 u_1) \ \} \, d\Omega.
\end{aligned}
$$

Using integration by parts and some rearrangements we shall have

$$
\begin{aligned}
(g_1'(\mathbf{u},\mathbf{v}),\mathbf{w}) \;=\; & \frac{1}{2}\int_\Omega \{\ R_{11}(\partial_{23} u_3 \cdot w_2 + \partial_{32} u_2 \cdot w_3 - \partial_{22} u_3 \cdot w_3 - \partial_{33} u_2 \cdot w_2)\\
& + R_{22}(\partial_{31} u_1 \cdot w_3 + \partial_{13} u_3 \cdot w_1 - \partial_{33} u_1 \cdot w_1 - \partial_{11} u_3 \cdot w_3)\\
& + R_{33}(\partial_{12} u_2 \cdot w_1 + \partial_{21} u_1 \cdot w_2 - \partial_{11} u_2 \cdot w_2 - \partial_{22} u_1 \cdot w_1) \ \} \, d\Omega \quad \text{(A.26)}\\
& + \int_\Gamma \{R_{11}(\partial_2 u_3 \cdot n_2 w_3 - \partial_2 u_3 \cdot n_3 w_2 - \partial_3 u_2 \cdot n_2 w_3 + \partial_3 u_2 \cdot n_3 w_2)\\
& + R_{22}(\partial_3 u_1 \cdot n_3 w_1 - \partial_3 u_1 \cdot n_1 w_3 - \partial_1 u_3 \cdot n_3 w_1 + \partial_1 u_3 \cdot n_3 w_3)\\
& + R_{33}(\partial_1 u_2 \cdot n_1 w_2 - \partial_1 u_2 \cdot n_2 w_1 - \partial_2 u_1 \cdot n_1 w_2 + \partial_2 u_1 \cdot n_2 w_1) \ \} \, d\Omega.
\end{aligned}
$$

For simplicity we take $R_{11} = R_{22} = R_{33}$. Then after some steps we will have the following

$$
\begin{aligned}
&(g_1'(\mathbf{u},\mathbf{v}),\mathbf{w})\\
&= \int_\Omega \{\ R_{11}\ \nabla \times \ [\ (\partial_2 u_3 - \partial_3 u_2)\tilde{i} + (\partial_3 u_1 - \partial_1 u_3)\tilde{j} + (\partial_1 u_2 - \partial_2 u_1)\tilde{k}\ ]\ \cdot \mathbf{w}\ \} \, d\Omega\\
&\quad - \int_\Gamma \{R_{11}[(n_2(\partial_1 u_2 - \partial_2 u_1) - n_3(\partial_3 u_1 - \partial_1 u_3))\tilde{i} \cdot w_1 \tilde{i}\\
&\quad + (n_3(\partial_2 u_3 - \partial_3 u_2) - n_1(\partial_2 u_3 - \partial_3 u_2))\tilde{j} \cdot w_1 \tilde{j} \qquad \text{(A.27)}\\
&\quad + (n_1(\partial_3 u_1 - \partial_1 u_3) - n_2(\partial_2 u_3 - \partial_3 u_2))\tilde{k} \cdot w_1 \tilde{k}]\} d\Gamma\\
&= \int_\Omega R_{11}\ [\ \nabla \times (\nabla \times \mathbf{u})\ ]\ \cdot \mathbf{w}\ d\Omega - \int_\Gamma R_{11}\ [\ \mathbf{n} \times (\nabla \times \mathbf{u})\ ]\ \cdot \mathbf{w}\ d\Gamma.
\end{aligned}
$$

Then the final expression for $\mathbf{G}^1_{g_1}$ will be of the form

$$
\mathbf{G}^1_{g_1} = R_{11}\{\ [\nabla \times (\nabla \times \mathbf{u})]|_\Omega - [\mathbf{n} \times (\nabla \times \mathbf{u})]|_\Gamma. \qquad \text{(A.28)}
$$

Now we consider the integrand of the second term in the cost functional as

$$
\begin{aligned}
g_2(\mathbf{u},\mathbf{v}) \;&=\; \frac{1}{2}\int_\Omega R_2(\nabla \mathbf{u}) d\Omega \qquad \text{(A.29)}\\
&=\; \frac{1}{2}\int_\Omega \sum_{i,j=1}^{3} R_{ij}(\nabla \mathbf{u})_i (\nabla \mathbf{u})_j \, d\Omega
\end{aligned}
$$

We are interested in quadratic form for shear stresses as

$$(\nabla \cdot \overline{\overline{R}} \nabla \mathbf{u}) = \begin{bmatrix} u_{1,1} & u_{2,1} & u_{3,1} \\ u_{1,2} & u_{2,2} & u_{3,2} \\ u_{1,3} & u_{2,3} & u_{3,3} \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ u_{3,1} & u_{3,2} & u_{3,3} \end{bmatrix} \tag{A.30}$$

where R_{ij} are the elements of the coefficient matrix R and $u_{i,j} = \frac{\partial u_i}{\partial x_j}$. In order to have the quadratic cross terms, we consider the trace of the above equation. Then

$$g_2(\mathbf{u}, \mathbf{v}) = \frac{1}{2} \int_\Omega \left\{ R_{22} u_{2,1}^2 + R_{33} u_{3,1}^2 + R_{33} u_{3,2}^2 + R_{11} u_{1,2}^2 + R_{11} u_{1,3}^2 + R_{22} u_{2,3}^2 \right\} \, d\Omega. \tag{A.31}$$

$$\begin{aligned} g_2(\mathbf{u} + \epsilon \mathbf{w}, \mathbf{v}) \;=\; & \frac{1}{2} \int_\Omega \left\{ R_{22} \left[\partial_1 (u_2 + \epsilon w_2) \right]^2 + R_{33} \left[\partial_1 (u_3 + \epsilon w_3) \right]^2 \right. \\ & + R_{33} \left[\partial_2 (u_3 + \epsilon w_3) \right]^2 + R_{11} \left[\partial_2 (u_1 + \epsilon w_1) \right]^2 \\ & \left. + R_{11} \left[\partial_3 (u_1 + \epsilon w_1) \right]^2 + R_{22} \left[\partial_3 (u_2 + \epsilon w_2) \right]^2 \right\} \, d\Omega. \end{aligned}$$

Since $\mathbf{G}_{g_2}^1 = g_2'(\mathbf{u}, \mathbf{v})$ and

$$(g_2'(\mathbf{u}, \mathbf{v}), \mathbf{w}) = \lim_{\epsilon \to 0} \frac{g_2(\mathbf{u} + \epsilon \mathbf{w}, \mathbf{v}) - g_2(\mathbf{u}, \mathbf{v})}{\epsilon} \tag{A.32}$$

then, after some algebraic manipulations, we will have

$$\begin{aligned} (g_2'(\mathbf{u}, \mathbf{v}), \mathbf{w}) \;=\; & \int_\Omega \{ R_{22} (\partial_1 u_2 \partial_1 w_2) + R_{33} (\partial_1 u_3 \partial_1 w_3) \\ & + R_{33} (\partial_2 u_3 \partial_2 w_3) + R_{11} (\partial_2 u_1 \partial_2 w_1) \\ & + R_{11} (\partial_3 u_1 \partial_3 w_1) + R_{22} (\partial_3 u_2 \partial_3 w_2) \} \, d\Omega. \end{aligned} \tag{A.33}$$

Integrating by parts, would result in

$$\begin{aligned} (g_2'(\mathbf{u}, \mathbf{v}), \mathbf{w}) \;=\; & - \int_\Omega \{ R_{11} (\partial_{33} u_1 + \partial_{22} u_1) w_1 + R_{22} (\partial_{11} u_2 + \partial_{33} u_2) w_2 \\ & + R_{33} (\partial_{11} u_3 + \partial_{22} u_3) w_3 \} \, d\Omega \\ & + \int_\Gamma \{ R_{11} (\partial_2 u_1 n_2 + \partial_3 u_1 n_3) w_1 + R_{22} (\partial_1 u_2 n_1 + \partial_3 u_2 n_3) w_2 \\ & + R_{33} (\partial_1 u_3 n_1 + \partial_2 u_3 n_2) w_3 \} \, d\Gamma \end{aligned} \tag{A.34}$$

For the choice of $R_{11} = R_{22} = R_{33}$, we will have

$$(g_2'(\mathbf{u}, \mathbf{v}), \mathbf{w}) =$$

$$- \int_\Omega R_{11}\left((\frac{\partial^2 u_1}{\partial x_3^2}+\frac{\partial^2 u_1}{\partial x_2^2})\vec{i}\ \ (\frac{\partial^2 u_2}{\partial x_1^2}+\frac{\partial^2 u_2}{\partial x_3^2})\vec{j}\ \ (\frac{\partial^2 u_3}{\partial x_1^2}+\frac{\partial^2 u_3}{\partial x_2^2})\vec{k}\right)\cdot \mathbf{w}\, d\Omega$$
$$+ \int_\Gamma R_{11}\left((\frac{\partial u_1}{\partial x_2}n_2+\frac{\partial u_1}{\partial x_3}n_3)\vec{i}\ \ (\frac{\partial u_2}{\partial x_1}n_1+\frac{\partial u_2}{\partial x_3}n_3)\vec{j}\ \ (\frac{\partial u_3}{\partial x_1}n_1+\frac{\partial u_3}{\partial x_2}n_2)\vec{k}\right)\cdot \mathbf{w}\, d\Gamma. \tag{A.35}$$

Finally

$$\begin{aligned}\mathbf{G}^1_{g_2} &= -R_{11}\left(\frac{\partial^2 u_1}{\partial x_3^2}+\frac{\partial^2 u_1}{\partial x_2^2}\quad \frac{\partial^2 u_2}{\partial x_1^2}+\frac{\partial^2 u_2}{\partial x_3^2}\quad \frac{\partial^2 u_3}{\partial x_1^2}+\frac{\partial^2 u_3}{\partial x_2^2}\right)|_\Omega \\ &\quad + R_{11}\left(\frac{\partial u_1}{\partial x_2}n_2+\frac{\partial u_1}{\partial x_3}n_3\quad \frac{\partial u_2}{\partial x_1}n_1+\frac{\partial u_2}{\partial x_3}n_3\quad \frac{\partial u_3}{\partial x_1}n_1+\frac{\partial u_3}{\partial x_2}n_2\right)|_\Gamma.\end{aligned} \tag{A.36}$$

For the third term in the cost functional, a similar procedure should be performed. For this case we have $\mathbf{G}^1_{g_3} = g_3'(\mathbf{u},\mathbf{v})$ and

$$(g_3'(\mathbf{u},\mathbf{v}),\mathbf{w}) = \lim_{\epsilon\to 0}\frac{g_3(\mathbf{u}+\epsilon\mathbf{w},\mathbf{v})-g_3(\mathbf{u},\mathbf{v})}{\epsilon} \tag{A.37}$$

Also

$$g_3(\mathbf{u},\mathbf{v}) = \frac{1}{2}\sum_{i,j=1}^{3} R_{ij}(\int_{\Gamma_m}\mathbf{u}ds - Q^d)_i(\int_{\Gamma_m}\mathbf{u}ds - Q^d)_j \tag{A.38}$$

and

$$g_3(\mathbf{u}+\epsilon\mathbf{w},\mathbf{v}) = \frac{1}{2}\sum_{i,j=1}^{3} R_{ij}(\int_{\Gamma_m}(\mathbf{u}+\epsilon\mathbf{w})ds - Q^d)_i(\int_{\Gamma_m}(\mathbf{u}+\epsilon\mathbf{w})ds - Q^d)_j\ . \tag{A.39}$$

Substituting equations (A.38) and (A.39) in equation (A.37), ignoring the cross components of the coefficient matrix R and skipping some intermediate steps, we will have

$$G^1_{g_2} = R_{11}(\int_{\Gamma_m} uds - Q^d)_1 w_1 + R_{22}(\int_{\Gamma_m} uds - Q^d)_2 w_2 + R_{33}(\int_{\Gamma_m} uds - Q^d)_3 w_3 \tag{A.40}$$

Similar to the previous cases, we take $R_{11} = R_{22} = R_{33}$. Then

$$\mathbf{G}^1_{g_3} = R_{11}(\int_{\Gamma_m} uds - Q^d) = R_{11}(Q - Q^d) \tag{A.41}$$

Finally we consider the fourth term in the cost functional. Since there in no $\mathbf{u}$ in this term,

$$\mathbf{G}^1_{g_4} = 0. \tag{A.42}$$

Combining the above results, the final expression for G^1 is obtained as

$$\begin{aligned}
\mathbf{G}^1 &= \alpha_1\{\,[\nabla\times(\nabla\times\mathbf{u})]|_\Omega - [\mathbf{n}\times(\nabla\times\mathbf{u})]|_\Gamma\} \\
&+ \alpha_2\{-[\frac{\partial^2 u_1}{\partial x_3^2}+\frac{\partial^2 u_1}{\partial x_2^2},\quad \frac{\partial^2 u_2}{\partial x_1^2}+\frac{\partial^2 u_2}{\partial x_3^2},\quad \frac{\partial^2 u_3}{\partial x_1^2}+\frac{\partial^2 u_3}{\partial x_2^2}]|_\Omega \\
&\qquad +[\frac{\partial u_1}{\partial x_2}n_2+\frac{\partial u_1}{\partial x_3}n_3,\quad \frac{\partial u_2}{\partial x_1}n_1+\frac{\partial u_2}{\partial x_3}n_3,\quad \frac{\partial u_3}{\partial x_1}n_1+\frac{\partial u_3}{\partial x_2}n_2]|_\Gamma\} \\
&+ \alpha_3(Q-Q^d)
\end{aligned} \tag{A.43}$$

A.2.2 Derivation of $\mathbf{G}^2$

In the first three terms of the cost functional, there is no $\mathbf{v}$. Then for these terms the Gateaux derivative with respect to the control will be zero, namely

$$\mathbf{G}^2_{g_1} = \mathbf{G}^2_{g_2} = \mathbf{G}^2_{g_3} = 0. \tag{A.44}$$

Regarding the fourth term in the cost functional, we have $\mathbf{G}^2_{g_4} = g_4'(\mathbf{u},\mathbf{v})$ and

$$(g_4'(\mathbf{u},\mathbf{v}),\mathbf{v}-\mathbf{v}^o)) = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\{g_4(\mathbf{v}+\epsilon(\mathbf{v}-\mathbf{v}^o)) - g_4(\mathbf{v})\} \tag{A.45}$$

in which

$$g_4(\mathbf{v}) = \frac{1}{2}\int_\Omega \sum_{i,j=1}^{3} R_{ij}(\mathbf{v})_i(\mathbf{v})_j\, d\Omega \tag{A.46}$$

and

$$g_4(\mathbf{v}+\epsilon(\mathbf{v}-\mathbf{v}^o)) = \frac{1}{2}\int_\Omega \sum_{i,j=1}^{3} R_{ij}(\mathbf{v}+\epsilon(\mathbf{v}-\mathbf{v}^o))_i(\mathbf{v}+\epsilon(\mathbf{v}-\mathbf{v}^o))_j\, d\Omega. \tag{A.47}$$

Substituting equations (A.46) and (A.47) in equation (A.45) and ignoring the cross terms results in

$$\begin{aligned}
(g_4'(\mathbf{v}),(\mathbf{v}-\mathbf{v}^o)) &= \frac{1}{2}\lim_{\epsilon\to 0}\frac{1}{\epsilon}\int_\Omega \big\{R_{11}(v+\epsilon(v-v^0))_1(v+\epsilon(v-v^0))_1 \\
&\quad + R_{22}(v+\epsilon(v-v^0))_2(v+\epsilon(v-v^0))_2 \\
&\quad + R_{33}(v+\epsilon(v-v^0))_3(v+\epsilon(v-v^0))_3 \\
&\quad - R_{11}(v_1v_1) - R_{22}(v_2v_2) - R_{33}(v_3v_3)\,\big\}\, d\Omega
\end{aligned} \tag{A.48}$$

or

$$(g_4'(\mathbf{v}), (\mathbf{v} - \mathbf{v}^o)) = \int_\Omega R_{11}(\mathbf{v}) \cdot (\mathbf{v} - \mathbf{v}^o)\, d\Omega \tag{A.49}$$

which results in

$$\mathbf{G}^2 = \mathbf{G}_{g_4}^2 = R_{11}(\mathbf{v}). \tag{A.50}$$

Appendix B

Some Details on the Finite Element Formulation of the Problem

B.1 Admissible Class of Approximating Functions Φ and Ψ

A brief examination of equations (3.9) will readily reveal the following facts:

1. First-order derivatives of the velocity shape function Φ occur, which implies that Φ and its first derivatives must be continuous in each element ($\Phi \in C^1(\Omega)$). They should also be continuous in the entire physical domain.

2. There is no differentiation with respect to Ψ in the Galerkin equations (3.9). Hence $\Psi \in C^0(\Omega)$ within each element but it may be discontinuous in the entire domain. Both Φ and Ψ should be integrable.

These considerations show that the choices of the approximation functions for velocity and pressure cannot be arbitrary and that conditions for the approximating functions for the pressure are weaker than those for the velocity approximating functions. In general, the pressure may be approximated by polynomials that are at least one degree less

than the polynomials for the velocity.

In the following, we summarize one type of admissible two- and three-dimensional elements. For other admissible elements in two- and three-dimensional domains, one may consult Cuvelier *et al.* (1986), Reddy (1984) or any standard finite element text.

9-node quadrilateral element

In this two-dimensional element the velocity components are approximated by the following biquadratic approximating function

$$\Phi = \left\{ \begin{array}{c} \frac{1}{4}rs(1-r)(1-s) \\ -\frac{1}{4}rs(1+r)(1-s) \\ \frac{1}{4}rs(1+r)(1+s) \\ -\frac{1}{4}rs(1-r)(1+s) \\ -\frac{1}{2}s(1-s)(1-r^2) \\ \frac{1}{2}r(1+r)(1-s^2) \\ \frac{1}{2}s(1+s)(1-r^2) \\ -\frac{1}{2}r(1-r)(1-s^2) \\ (1-r^2)(1-s^2) \end{array} \right\}. \tag{B.1}$$

This element and the ordering of the various interpolation functions are shown in Figure B.1, while r and s refer to the normalized (local) coordinates of the element, which vary from -1 to +1, as shown in the figure.

Two different pressure approximations are possible for this element. One is a continuous approximating function, in which the degrees of freedom are located at four corner nodes of the element and the approximating function is given by

$$\Psi = \left\{ \begin{array}{c} \frac{1}{4}(1-r)(1-s) \\ \frac{1}{4}(1+r)(1-s) \\ \frac{1}{4}(1+r)(1+s) \\ \frac{1}{4}(1-r)(1+s) \end{array} \right\}. \tag{B.2}$$

The second possible approximation for pressure in this element is a discontinuous linear

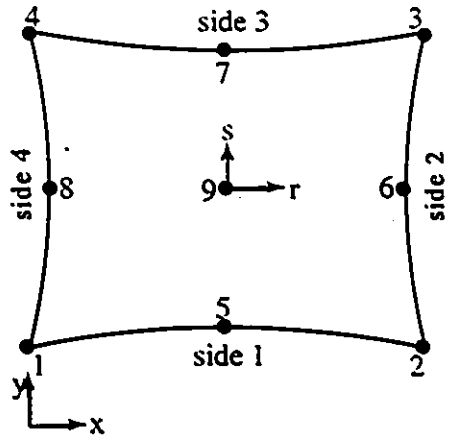


Figure B.1: 9-node two-dimensional quadrilateral element.

approximation described by

$$\Psi = \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} \tag{B.3}$$

in which the pressure is approximated by $P = a + bx + cy$ within the element. Note that in this case the unknowns are the coefficients in the above expression and not the values of the pressure at the nodal points.

One more relation is required to relate the global coordinate system, (x, y), to the local coordinate system, (r, s). This is obtained using the parametric concept, as

$$x = \Lambda^T X \quad , \quad y = \Lambda^T Y. \tag{B.4}$$

If $\Lambda = \Phi$, then the element is called isoparametric, which means that the interpolation functions used to define the dependent variables are of the same order as the functions defining the element geometry.

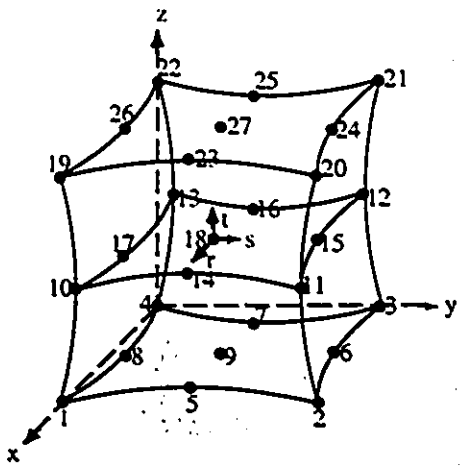
27-node three-dimensional element

In this element, the velocity components are approximated by the following tri-

quadratic approximating function.

$$\Phi = \left\{ \begin{array}{c} -\frac{1}{8}rst(1-r)(1-s)(1-t) \\ \frac{1}{8}rst(1+r)(1-s)(1-t) \\ -\frac{1}{8}rst(1+r)(1+s)(1-t) \\ \frac{1}{8}rst(1-r)(1+s)(1-t) \\ \frac{1}{4}st(1-r^2)(1-s)(1-t) \\ -\frac{1}{4}rt(1+r)(1-s^2)(1-t) \\ -\frac{1}{4}st(1-r^2)(1+s)(1-t) \\ \frac{1}{4}rt(1-r)(1-s^2)(1-t) \\ -\frac{1}{2}t(1-r^2)(1-s^2)(1-t) \\ \frac{1}{4}rs(1-r)(1-s)(1-t^2) \\ -\frac{1}{4}rs(1+r)(1-s)(1-t^2) \\ \frac{1}{4}rs(1+r)(1+s)(1-t^2) \\ -\frac{1}{4}rs(1-r)(1+s)(1-t^2) \\ -\frac{1}{2}s(1-r^2)(1-s)(1-t^2) \\ \frac{1}{2}r(1+r)(1-s^2)(1-t^2) \\ \frac{1}{2}s(1-r^2)(1+s)(1-t^2) \\ -\frac{1}{2}r(1-r)(1-s^2)(1-t^2) \\ (1-r^2)(1-s^2)(1-t^2) \\ \frac{1}{8}rst(1-r)(1-s)(1+t) \\ -\frac{1}{8}rst(1+r)(1-s)(1+t) \\ \frac{1}{8}rst(1+r)(1+s)(1+t) \\ -\frac{1}{8}rst(1-r)(1+s)(1+t) \\ -\frac{1}{4}st(1-r^2)(1-s)(1+t) \\ \frac{1}{4}rt(1+r)(1-s^2)(1+t) \\ \frac{1}{4}st(1-r^2)(1+s)(1+t) \\ -\frac{1}{4}rt(1-r)(1-s^2)(1+t) \\ \frac{1}{2}t(1-r^2)(1-s^2)(1+t) \end{array} \right\} \qquad \text{(B.5)}$$

The ordering of the various interpolation functions is shown in Figure B.2, while r, s and t refer to the normalized (local) coordinates of the element, which vary from -1 to +1,



surface number	nodes
1	4,7,3,13,16,12,22,25,21
2	1,5,2,10,14,11,19,23,20
3	4,8,1,13,17,10,22,26,19
4	3,6,2,12,15,11,21,24,20
5	4,8,1,7,9,5,3,6,2
6	22,26,19,25,27,23,21,24,20

Figure B.2: 27-node three-dimensional element.

as shown in the figure.

Three different pressure approximations are possible for this element. One is a continuous trilinear approximating function, in which the degrees of freedom are located at eight corner nodes of the element and the approximating function is given by

$$\Psi = \left\{ \begin{array}{l} \frac{1}{8}(1-r)(1-s)(1-t) \\ \frac{1}{8}(1+r)(1-s)(1-t) \\ \frac{1}{8}(1+r)(1+s)(1-t) \\ \frac{1}{8}(1-r)(1+s)(1-t) \\ \frac{1}{8}(1-r)(1-s)(1+t) \\ \frac{1}{8}(1+r)(1-s)(1+t) \\ \frac{1}{8}(1+r)(1+s)(1+t) \\ \frac{1}{8}(1-r)(1+s)(1+t) \end{array} \right\}. \tag{B.6}$$

The second possible approximation for pressure in this element is a discontinuous trilinear

approximation described by

$$\Psi = \left\{ \begin{array}{l} \frac{1}{8}(g-r)(g-s)(g-t)/g^3 \\ \frac{1}{8}(g+r)(g-s)(g-t)/g^3 \\ \frac{1}{8}(g+r)(g+s)(g-t)/g^3 \\ \frac{1}{8}(g-r)(g+s)(g-t)/g^3 \\ \frac{1}{8}(g-r)(g-s)(g+t)/g^3 \\ \frac{1}{8}(g+r)(g-s)(g+t)/g^3 \\ \frac{1}{8}(g+r)(g+s)(g+t)/g^3 \\ \frac{1}{8}(g-r)(g+s)(g+t)/g^3 \end{array} \right\} \tag{B.7}$$

in which $g = 2\sqrt{1/3}$ and the pressure degrees of freedom are located at the eight points of $2\times2\times2$ Gaussian integration points and are the actual pressure values at these points.

The third possible approximation for pressure in this element is a discontinuous linear approximation described by

$$\Psi = \left\{ \begin{array}{l} 1 \\ x \\ y \\ z \end{array} \right\} \tag{B.8}$$

in which the pressure is approximated by $P = a+bx+cy+dz$ within the element. Similar to two-dimensional elements, in this approximation the unknowns are the coefficients in the above expression and not the values of the pressure at the nodal points.

As in two-dimensional cases, one more relation is required to relate the global coordinate system, (x, y, z), to the local coordinate system, (r, s, t). This is obtained using the parametric concept, as

$$x = \Lambda^T X \quad , \quad y = \Lambda^T Y \quad , \quad z = \Lambda^T Z. \tag{B.9}$$

If $\Lambda = \Phi$, then the element is called isoparametric, which means that the interpolation functions used to define the dependent variables are of the same order as the functions defining the element geometry.

B.2 Evaluation of the Derivatives and Integrals

In this section we consider three-dimensional problems. The results may, however, be easily simplified to describe the corresponding two-dimensional cases.

A glance at the coefficients of the matrix equations (3.11) and (3.21) shows that, in order to construct the coefficient matrices, some integration and derivation of the approximating functions Φ and Ψ with respect to the global coordinates, (x, y, z), is needed. One may also note that these interpolating functions are described in local coordinates, (r, s, t), while their integrals and derivatives are needed in global coordinates, (x, y, z). Since these integrations and derivations are most easily carried out in local coordinates, the following transformations are introduced

$$\begin{bmatrix} \frac{\partial\Phi}{\partial r} \\ \frac{\partial\Phi}{\partial s} \\ \frac{\partial\Phi}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \begin{bmatrix} \frac{\partial\Phi}{\partial x} \\ \frac{\partial\Phi}{\partial y} \\ \frac{\partial\Phi}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial\Lambda^T}{\partial r}X & \frac{\partial\Lambda^T}{\partial r}Y & \frac{\partial\Lambda^T}{\partial r}Z \\ \frac{\partial\Lambda^T}{\partial s}X & \frac{\partial\Lambda^T}{\partial s}Y & \frac{\partial\Lambda^T}{\partial s}Z \\ \frac{\partial\Lambda^T}{\partial t}X & \frac{\partial\Lambda^T}{\partial t}Y & \frac{\partial\Lambda^T}{\partial t}Z \end{bmatrix} \begin{bmatrix} \frac{\partial\Phi}{\partial x} \\ \frac{\partial\Phi}{\partial y} \\ \frac{\partial\Phi}{\partial z} \end{bmatrix} = J \begin{bmatrix} \frac{\partial\Phi}{\partial x} \\ \frac{\partial\Phi}{\partial y} \\ \frac{\partial\Phi}{\partial z} \end{bmatrix} \quad \text{(B.10)}$$

where J is the Jacobian of the transformation from global coordinates, (x, y, z), to the local element coordinates, (r, s, t). Using isoparametric elements, i.e. $\Lambda = \Phi$, and inverting equation (B.10) one gets the required transformation, as

$$\begin{bmatrix} \frac{\partial\Phi}{\partial x} \\ \frac{\partial\Phi}{\partial y} \\ \frac{\partial\Phi}{\partial z} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial\Phi}{\partial r} \\ \frac{\partial\Phi}{\partial s} \\ \frac{\partial\Phi}{\partial t} \end{bmatrix}. \quad \text{(B.11)}$$

The transformation of the element area is given by the following relation

$$d\Omega = dxdydz = |J|drdsdt \quad \text{(B.12)}$$

and the new integration limits would be the limits of the local coordinates, which are from -1 to +1.

After transforming all the integrals in coefficient matrices, (3.12)-(3.16) and (3.23)-(3.24), they will be of the general form

$$I = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} f(r, s, t)drdsdt. \quad \text{(B.13)}$$

These integrals are evaluated using a numerical quadrature formulae, as

$$I = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} f(r_i, s_j, t_k) W_i W_j W_k \tag{B.14}$$

where the weighting coefficients W_i, W_j and W_k depend on the specific quadrature used. r_i, s_j and t_k are quadrature points. One may consult Zienkiewicz and Morgan (1983) for the location of the quadrature points and the values of the weight coefficients in two- and three-dimensional integrations.

B.3 Solution Methods

In general, the application of the Galerkin finite element method as described in section 3.2 to the steady Navier-Stokes equations or to the transient Navier-Stokes equations after time discretization would result in a system of algebraic nonlinear equations, which in matrix form can be written as

$$K(U)U = F \tag{B.15}$$

In this equation, K is the global matrix which has been obtained after assemblage of the element matrices and is generally large, non-symmetric and sparse; U is the global vector of unknowns, namely the nodal values of velocity components and pressure; and F is a known vector containing the effects of body forces and boundary conditions. Due to the presence of the convective term in the Navier-Stokes equations, the above system of equations is nonlinear. For high Reynolds numbers, the nonlinear character will be dominant, in which case the rate of convergence and the cost efficiency of the solution method would be important issues. In general, the most time consuming part of a finite element program is the solution of the final algebraic equations (B.15).

In the following, some iterative methods for solving equations (B.15) are briefly discussed. Two major points should be considered in choosing one specific solution method: the radius of convergence and the rate of convergence. The algorithm should be convergent for a wide range of problems with different material properties and geometries.

The rate of convergence should also be sufficiently high for computational cost efficiency. Unfortunately no solution method satisfies all these requirements.

B.3.1 Successive Substitution Method

This method, which is also called Picard or functional iteration, is relatively simple and has a large radius of convergence. In the $(n+1)$-th iteration this method has the form

$$K(U^n)U^{n+1} = F(U^n). \tag{B.16}$$

The rate of convergence of this method, especially for highly nonlinear problems, is slow because it is only a first order method. The rate of convergence can be improved by using an accelerating factor as

$$K(U^n)U^* = F(U^n) \tag{B.17}$$

$$U^{n+1} = \alpha U^n + (1-\alpha)U^* \quad , \quad 0 \leq \alpha \leq 1. \tag{B.18}$$

B.3.2 Newton's Method

This method, which is a second order method, has a higher rate of convergence. Let U^* be an approximate solution of equation (B.15). Then

$$\begin{aligned} U &= U^* + \Delta U \\ R(U) &= K(U)U - F(U) = 0. \end{aligned} \tag{B.19}$$

The Taylor series expansion of $R(U)$ in the neibourhood of U may be written as

$$R(U) = R(U^*) + \frac{\partial R}{\partial U}(U^*)\Delta U + O(\Delta U^2). \tag{B.20}$$

Truncating the above expression and using (B.19) results in the approximation

$$R(U^*) = -\frac{\partial R}{\partial U}(U^*)\Delta U \tag{B.21}$$

which suggests the following iteration algorithm

$$R(U^n) = -\frac{\partial R}{\partial U}(U^n)(U^{n+1} - U^n) = -J(U^n)(U^{n+1} - U^n) \tag{B.22}$$

or

$$U^{n+1} = Un - J^{-1}(U^n)R(U^n) \tag{B.23}$$

where J is the Jacobian matrix. This algorithm has the advantage of having a better rate of convergence than the Picard method but its radius of convergence is small, i.e. it is more sensitive to the initial solution vector U^0.

B.3.3 Modified Newton-Raphson Method

The main drawback in Newton's method is that the Jacobian matrix should be computed at each iteration, which is computationally expensive. In the modified Newton-Raphson method, this Jacobian matrix is not computed at each iteration but instead a fixed Jacobian is used, which is computed once at some initial estimate of the solution vector U^0, as

$$U^{n+1} = U^n - J^{-1}(U^0)R(U^n). \tag{B.24}$$

This method also suffers from a relatively small rate of convergence.

B.4 Initial and Boundary Conditions

The solution of the transient Navier-Stokes equations like that of any other time dependent partial differential equation, requires initial conditions to be specified. The initial condition in the Navier-Stokes equations can be applied by prescribing the velocity field at time $t = 0$. In specifying this, one should note that the initial velocity field must satisfy the continuity equation. An initial velocity field that is zero everywhere obviously satisfies this equation.

Two kinds of boundary conditions are possible: Dirichlet and Neumann boundary conditions, also known as essential and natural boundary conditions, respectively. Dirichlet boundary conditions (namely, specified values for the velocity components, $\bar{U}$, along the boundary) are explicitly imposed as $u = \bar{U}$ on Γ, while Neumann boundary conditions (namely, specified values for the derivatives of the velocity components) are applied through boundary integrals on the right hand sides of equations (3.11).

In the case of incompressible flows, no boundary conditions for pressure are necessary (see e.g. Cuvelier *et al.*, 1986). Boundary conditions for pressure are actually given implicitly through normal stress components (Neumann boundary conditions) in boundary integrals, as mentioned earlier. It is important to note that, at each node, the velocity and the stresses can not be specified in the same direction. As a general rule, only one of the pairs $(u_1, u_2), (u_1, t_2), (t_1, u_2)$ and (t_1, t_2) can be specified at each boundary node of a two-dimensional domain.

Both initial and boundary conditions are applied after all element matrices are assembled in the single global matrix.

B.5 Time Discretization

After assembling the element equations (3.11), the following global equations for the whole domain are obtained

$$[M]\dot{U} + [K(U)]U = [f] \qquad \text{in } I = [0, T]. \tag{B.25}$$

This equation represents the original Navier-Stokes equation, it is discrete in space and continuous in time and may be solved in time by a direct time integration method, which in a finite difference manner approximates the continuous time derivative by the solution in discrete time steps. In this approximation, the time interval, $I = [0, T]$, is divided into N discrete time intervals of length $\Delta t = \frac{T}{N}$ (in general, the time step may not necessarily be fixed), resulting in the set of discrete points $t_0, t_1, ..., t_N$. Then the solution is sought

at these discrete points.

This kind of discretization of continuous time derivatives can be further divided in one-step and multi-step methods and in explicit and implicit methods. In one-step method approximations for the computation of the solution at time step t_n, only the solution at the preceding time step t_{n-1} is required, whereas multi-step methods require that the solution be available at some intermediate steps as well. In this section, only one-step time integration methods will be considered.

Explicit methods are characterized by the fact that the solution at time t_n can be computed by simple algebraic manipulations from the solution at the preceding time step t_{n-1}. In implicit methods a set of simultaneous equations must be solved. In constructing a time integration scheme the problem of stability and accuracy of the results should be considered. Explicit methods, although simpler to implement and less expensive to run, require smaller time steps for their stability. Implicit methods are generally preferable for the solution of the Navier-Stokes equations, due to their superior stability characteristics and their consistent treatment of the pressure. In this section, the so-called one-step θ-method will be discussed for the solution of the ordinary differential equations (B.25). In the θ method, the solution, U, and the forcing function, f, between two successive time steps t_n and t_{n+1} are assumed to vary linearly. Then, at the time $t_{n+\theta} = t_n + \theta\Delta t$, between time steps t_n and t_{n+1}, the solution is approximated as

$$U(t_{n+\theta}) = U_{n+\theta} = (1-\theta)U_n + \theta U_{n+1} \tag{B.26}$$

in which $\theta \in [0,1]$ and Δt may be constant or variable. Similarly f at time $t_{n+\theta}$ is approximated as

$$f_{n+\theta} = (1-\theta)f_n + \theta f_{n+1}. \tag{B.27}$$

From (B.26), the time derivative of U at time $t_{n+\theta}$ becomes

$$\dot{U}_{n+\theta} = \frac{1}{\Delta t}(U_{n+1} - U_n). \tag{B.28}$$

Now consider the equation (B.25) at time $t_{n+\theta}$

$$[M]\dot{U}_{n+\theta} + [K]U_{n+\theta} = [f]_{n+\theta}. \tag{B.29}$$

Substituting (B.26), (B.27) and (B.28) in (B.29) results in the following recursive relationship

$$[\hat{M}]U_{n+1} = [\hat{f}] \tag{B.30}$$

in which

$$\begin{aligned}
[\hat{M}] &= [M] + \theta\Delta t[K] \\
[\hat{f}] &= \{[M] - (1-\theta)\Delta t[K]\}\, U_n + \Delta t\,\{(1-\theta)[f_n] + \theta[f_{n+1}]\}\,.
\end{aligned} \tag{B.31}$$

Thus, by using the system of algebraic equations (B.30), the solution at time step $n+1$ is obtained from the known solution at time step n. Some initial velocity, U_0, is specified to initiate the procedure. From a computational point of view, the above discretization is performed on the elements rather than on the global equations, which would result in significant savings in the computer memory size required for solving the problem. The following are some well known methods, corresponding to special stability conditions for θ

a) θ=0 : explicit Euler scheme; conditionally stable (small Δt),

b) $\theta = \frac{1}{2}$: Crank-Nicolson scheme; unconditionally stable,

c) θ=1 : implicit Euler scheme; unconditionally stable.

Regarding the accuracy of the θ-scheme, it is found that (see e.g., Dhatt and Touzot, 1984) the global error for the case $\theta = \frac{1}{2}$ is $O(\Delta t^2)$ and for the other cases it is $O(\Delta t)$.

It should be noted that, besides the Navier-Stokes equations, the continuity equation should also be satisfied at each time step. Also note that in the case of velocity-pressure formulation of the Navier-Stokes equations (see equations (3.11)), the pressure is included in the solution vector, U, in (B.25).

In conclusion, the use of an explicit procedure for the solution of time dependent Navier-Stokes equations would result in substantial difficulties and hence implicit methods are the widely used ones.

Appendix C

Description of the Computer Code

C.1 General Description

The computer code FLOPTIM has been developed to perform the numerical simulation of the optimization algorithms described in previous chapters. This program may also be used to solve only the Navier-Stokes equations. The program is coded in FORTRAN and is based on the Finite Element Method for the solution of the governing partial differential equations. Double-precision real arithmetic is used throughout the program, which is mandatory for the present analysis and most finite element applications.

C.2 Program Capabilities

The program is capable of solving the Navier-Stokes and the adjoint equations as described in Chapter 2. For the case of the Navier-Stokes equations, the simulations are possible for two-dimensional or three-dimensional, linear or non-linear, steady-state or transient flows. Both essential and natural boundary conditions and initial conditions can be specified. Also, time-dependent essential boundary conditions are allowed. The

flow is restricted to be laminar and Newtonian. Regarding the optimization algorithm, the program can be used for two- or three-dimensional initial velocity control and two- or three-dimensional boundary velocity control.

The program has the capability of solving the equations both in dimensional and nondimensional forms. If the dimensional form is chosen, the user is required to input data in a consistent dimensional system, and output results will be in the same system of units. On the other hand, if the nondimensional form is chosen, then the input data should be nondimensionalized according to the discussion in section 3.5.

C.3 Boundary and Initial Conditions

The boundary conditions that can be applied to the fluid include specified nodal values for velocity components (essential boundary conditions) and specified total stress for element sides (natural boundary conditions). Also a combination of these boundary conditions may be specified provided that the theoretical restrictions described in section B.4 are respected.

For time-dependent problems, initial conditions can be specified for velocity components, which prescribe the state of the flow at the initial time. It should be noted that the specified initial field for the velocity should satisfy the continuity equations.

C.4 Element Library

In the finite element discretization of the Navier-Stokes equations, the elements are characterized by their geometric shapes and by the interpolation functions used in them to approximate the velocity and the pressure as described in section B.1. In the program FLOPTIM, for two-dimensional problems, 9-node quadrilateral elements are used. This

element and the node numbering convention adopted in the program are shown in Figure B.1. The approximating function for the velocity is described in equation (B.1) and, for the pressure, the discontinuous linear approximation (equation (B.3)) is used. In the case of three-dimensional problems, a 27-node quadratic brick is used. This isoparametric element has the tri-quadratic velocity approximation as described by equation (B.5) and the discontinuous linear pressure approximation given by equation (B.8). The ordering and surface numbers for this element are shown in Figure B.2; those are needed in identifying the element boundaries in case of natural boundary conditions.

C.5 Mesh Generation

The physical coordinates of the nodal points and the node numbers associated with the elements should be provided to the program. Any commercial mesh generator can be used to generate this data. We have used the mesh generator of the FIDAP code for this purpose. We have also used the same package to identify the nodal points with specified boundary conditions.

C.6 Physical Properties

The numerical values of the physical quantities, density and viscosity should be provided to the program. If the non-dimensionalized version of the Navier-Stokes equations, as described in Chapter 3 is solved, which is strongly recommended, then the value of 1 should be specified for viscosity and the value of density would be the Reynolds number of the flow being studied. These quantities should be constant over the time and domain.

The volumetric force that can be applied to the fluid is the body force and, similarly to the density and viscosity, it can only have a constant value over time and domain.

C.7 Solution Techniques

In the present version of FLOPTIM, the penalty function formulation of the Navier-Stokes equations, as described in sections 1.8.2 and 3.4, is used. Application of the Galerkin Finite Element Method to this formulation results in the system of equations (3.29). As discussed in Chapter 3, in obtaining the finite element formulation of the governing partial differential equations, a weak variational approach is used, in which the spatial approximations are considered first, resulting in a set of ordinary differential equations. These equations are further approximated (using finite difference in time) to obtain a set of algebraic equations. The element matrices and vectors in these equations (see equations (3.12) -(3.16)) are calculated using an isoparametric formulation, in which the same element interpolation functions are used to describe both the geometry of the element and the variation of the dependent variable. The isoparametric transformations, the derivatives and integrals and the Jacobian matrix required to evaluate the element matrices and vectors are performed as described in section B.2.

Regarding the time integration in the above element equations, a θ-family of integration schemes, as described in section B.3, is used with a value of $\theta = 1.0$, which leads to a fully implicit approximation in time.

The above finite element discretization of the differential equations gives rise to the final algebraic equations in the form

$$\mathbf{A}X = \mathbf{b} \tag{C.1}$$

In the present formulation, the matrix A is large, asymmetric, sparse and banded. There are a few different schemes to store the global matrix A and to solve equation (C.1). We have used the so-called out-of-core skyline storage method for A (see Dhatt and Touzot, 1984). In this method, due to the sparseness of matrix A, only the elements between the leftmost and rightmost non-zero elements in each row are stored. The matrix is usually so large that, even with this storage scheme, it cannot be stored in core and it must be segmented in blocks and stored on low-speed storage. We have adapted and slightly

modified a solver developed by Hasbani and Engelman (1979), which uses the above storage technique and the direct Gauss elimination method to solve the set of equations (C.1). The details of this method can be found in the above reference.

The system of algebraic equations (C.1) should be solved at each time step. This equation is nonlinear due to the convective term in the Navier-Stokes equations and should be solved iteratively. We have used the successive substitution (or Picard) method as discussed in section B.3.1 to perform this iteration. As a convergence criterion, we have specified that the relative error must obey

$$\frac{||\Delta \mathbf{u}_n||}{||\mathbf{u}_n||} \leq \epsilon$$

in this nonlinear iteration, in which $||\cdot||$ is the Euclidean norm, $\mathbf{u}_n$ is the solution at iteration n and $\Delta \mathbf{u}_n = \mathbf{u}_n - \mathbf{u}_{n-1}$.

After calculating the velocity vector, $\mathbf{u}$, at each time step, the pressure is calculated using equation (3.28).

C.8 Post-Processing

After the successful solution of the Navier-Stokes equations or the optimization algorithm, the results were written on output files which were further exported to a graphics package to produce plots. We have used the TECPLOT graphics package (TECPLOT, 1993) for our two- and three-dimensional graphics.

C.9 Flow Chart

In this section, we present a flow chart of the computer code FLOPTIM and describe the function of the various subroutines. The subroutines identified by the superscript * are not called if the program is used to solve only the Navier-Stokes equations.

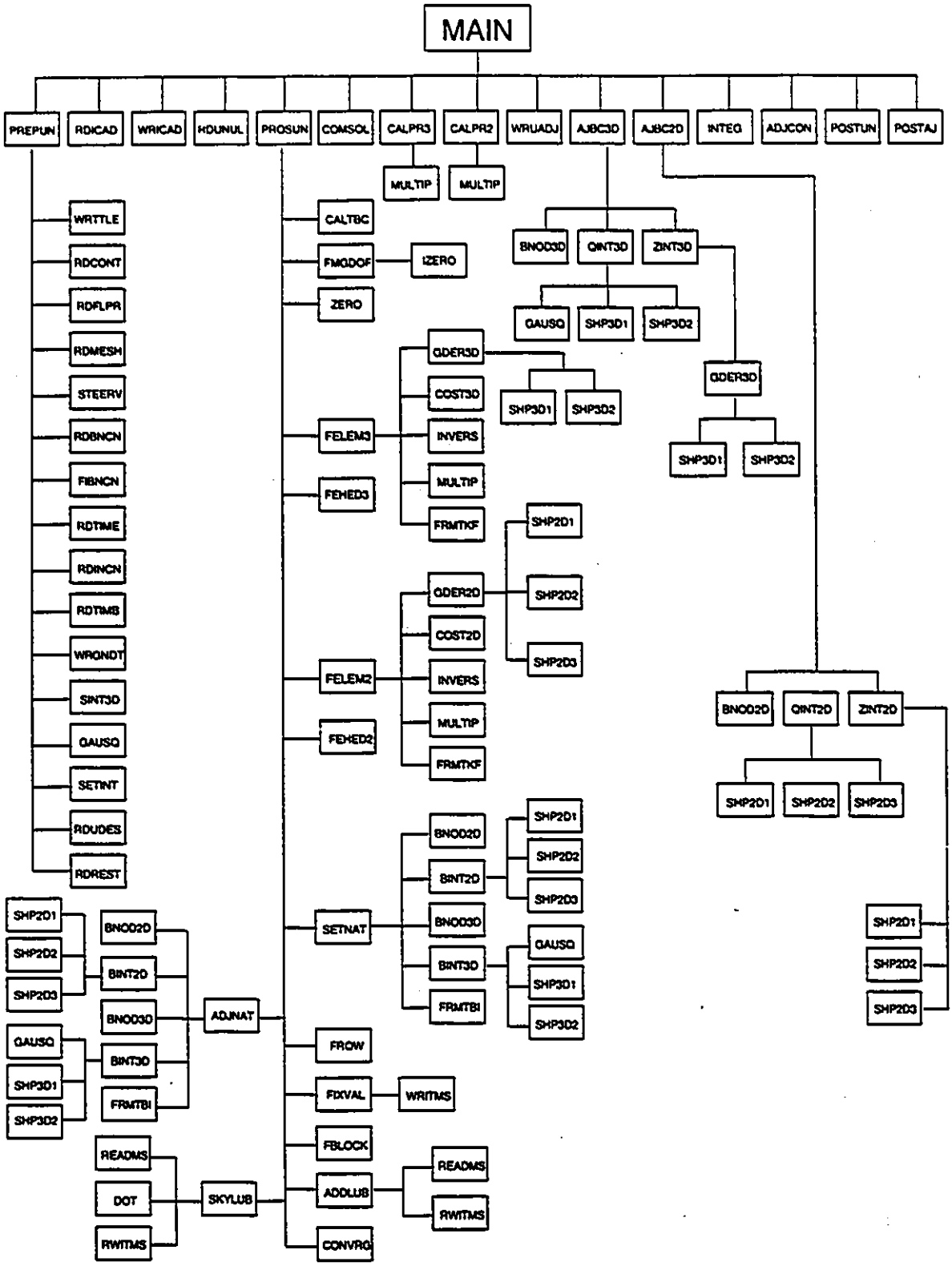


Figure C.1: Flow chart of the program FLOPTIM.

The program consists of 72 subroutines, which have been independently developed excluding subroutines SETINT, GAUSQ, SINT3D, SHP2D1, SHP2D2, SHP2D3 which were taken from Akay (1990). Also subroutines FROW, FIXVAL, FBLOCK, ADDLUB, SKYLUB and DOT were taken from Hasbani and Engelman (1979) and were slightly modified.

• MAIN

Directs the execution of the program by calling subroutines to read the input data, to form and to solve the finite element equations and to print the results.

• ADDLUB

Called to assemble the coefficient matrix by blocks. Each block is of maximum size memory. Each block after assembly contains, in addition to appropriate section of the coefficient matrix, all the necessary information for the solution process.

• ADJCON*

Called in two- and three-dimensional initial control problems, at the end of each optimization iteration (after solving the adjoint equations), to calculate the gradient of the cost functional (equation (2.32)) and the new control (equation (2.33)). This subroutine is called only at the end of the last time step.

• ADJNAT*

Called in two- and three-dimensional elements to calculate the contribution to load vector from right-hand side of the adjoint equations.

• AJBC2D*

Same as AJBC3D for two-dimensional boundary control problems.

• AJBC3D*

Called in three-dimensional boundary control problems, at the end of each optimization iteration (after solving the adjoint equations), to calculate the gradient of the cost functional (equation (2.24)) and the new control (equation (2.25)). This subroutine is called at the end of each time step.

• BINT2D

Called to calculate the line integrals in two-dimensional elements.

• BINT3D

Called to calculate the surface integrals in three-dimensional elements.

• BNOD2D

Called to find the local nodes on a specific side of two-dimensional elements.

• BNOD3D

Called to find the local nodes on a specific surface of three-dimensional elements.

• CALPR3

Called in three-dimensional problems to calculate the pressure field from the total velocity field formed by COMSOL. This calculation is performed using equation (3.28). In the case of adjoint equations, it calculates the adjoint pressure field.

• CALPRS

Same as CALPR3 for two-dimensional problems.

• CALTBC

Called in time-dependent problems at the beginning of each time step to calculate the time-dependent essential boundary conditions for the specified nodal points.

• COMSOL

Called to combine the solution vector (containing the nodal velocity unknowns) with the input boundary conditions to form the total velocity vector after solving the finite element equations. In the case of the adjoint equations, it forms the adjoint velocity vector.

• CONVRG

Called to check the convergence of the Navier-Stokes equations at each time step.

• COST2D*

Called to calculate the numerical value of the cost functional in optimization algorithm for two-dimensional domains.

• COST3D*

Called to calculate the numerical value of the cost functional in optimization algorithm for three-dimensional domains.

• DOT

Called to calculate the scalar product of two vectors.

• FBLOCK

Called to determine the block structure for a skyline stored non-symmetric coefficient

matrix according to the core storage available.

• FEHED2

Same as FEHED3 for two-dimensional problems.

• FEHED3

Called to form vectors LM and CONSTR for three-dimensional elements as needed in subroutines FROW and FIXVAL. For the definition of these vectors, see Hasbani and Engelman (1979).

• FELEM2

Called to form the finite element equations at element level for two-dimensional elements.

• FELEM3

Called to form the finite element equations at element level for three-dimensional elements.

• FIBNCN

Called to read nodal points with specified essential boundary conditions and their values.

• FIXVAL

Called to update the load vector BE so as to eliminate those degrees of freedom which have a constrained value. Such values are indicated by a negative entry in vector LM.

• FMGDOF

Called to specify the global degrees of freedom of non-constrained nodal points (nodes with no boundary conditions specified).

• FRMTBI

Called in time dependent problems to set up step-by-step contribution of boundary integrals to load vector, in two- or three-dimensional elements.

• FRMTKF

Called in time-dependent, two- and three-dimensional elements to perform the time discretization of the finite element equations.

• FROW

Called to calculate the difference between the leftmost (uppermost) non-zero coefficient and the diagonal coefficient in those rows (columns) of the element stiffness matrix for which the vector LM(I) $>$ 0, and the first and last equation to which the element con-

tributes to in the full system of equations.

• GAUSQ

Called to set up Gauss quadratures for two-dimensional surfaces of three-dimensional elements. This information is needed in calculating the surface integrals in three-dimensional problems.

• GDER2D

Called to evaluate global derivatives of the approximation functions in two-dimensional elements.

• GDER3D

Called to evaluate global derivatives of the approximation functions in three-dimensional elements.

• INTEGR*

Called at the end of the last time step, at each optimization iteration to perform the time integration of the cost functional.

• INVERS

Called to calculate the inverse of a square matrix.

• IZERO

Called to initialize an integer vector.

• IZERO2

Called to initialize an integer matrix.

• MULTIP

Called to compute the product of a $(l \times m)$ and a $(m \times n)$ matrix.

• POSTAJ*

Called at the end of the last time step to write the optimal velocity and pressure after convergence of the optimization algorithm.

• POSTUN

Called at the end of each time step to write the solution field (velocity and pressure) after solving the Navier-Stokes equations.

• PREPUN

Calls subroutines to read the information about the problem type, fluid properties, mesh

data and constant and time-dependent boundary conditions.

• PROSUN

Calls subroutines to form and to solve the finite element equations both for the Navier-Stokes equations and the adjoint equations.

• QINT2D*

Same as QINT3D for two-dimensional problems.

• QINT3D*

Called in three-dimensional boundary control problems to compute the surface integral of the adjoint pressure (see equation (2.24)).

• RDBNCN

Called to read nodal points with specified essential or natural boundary conditions and their values.

• RDCONT

Called to read from input file the data on problem type, number of nodes, number of elements, the value of the convergence tolerance in Navier-Stokes equations, maximum number of iterations allowed, weight coefficients of cost functional and the necessary flags to direct the execution of the problem.

• RDFLPR

Called to read from input file the fluid properties ρ, μ, two or three components of the body force (in two- and three-dimensional problems, respectively) and the penalty parameter (see equation 3.26).

• RDICAD*

Called at the beginning of each optimization iteration in initial control problems to assign the new control variable as the initial velocity field. It is not called in boundary control problems.

• RDINCN

Called to read the initial velocity field if it is specified.

• RDMESH

Called to read nodal coordinates and element connectivity matrix from input file for two- and three-dimensional problems.

• RDREST

Called to read the initial conditions from restart file. This subroutine is called only if the problem is restart of a previously converged solution of the Navier-Stokes equations (i.e. with a lower Reynolds number).

• RDTIMB

Called in time-dependent problems to read time-dependent essential boundary conditions.

• RDTIME

Called in time-dependent problems to read data related to time integration, such as initial time, number of time steps, time increment, θ, and the time step at which the inlet valve is closed and outlet valve is opened (in the case of boundary control optimization).

• RDUDES*

Called in initial control problems to read the desired velocity distribution.

• RDUNUL*

Called at the beginning of the solution of the adjoint equations to initialize the adjoint velocity field. It is called both in initial control and in boundary control problems. In the latter case, it is also called at the beginning of the solution of the Navier-Stokes equations to give initial conditions to these equations from the previous iteration.

• READMS

Called to read information stored on a random access file.

• SETINT

Called to set up quadrature rules of orders 1, 2 and 3 for two-dimensional elements. This information is needed in calculating the surface integrals in two-dimensional problems.

• SETNAT

Called in two- and three-dimensional elements to calculate the contribution to load vector from natural boundary conditions in Navier-Stokes equations.

• SHP2D1

Called to set up the approximation function and its local derivatives for 4-node two-dimensional elements. This subroutine is not presently used.

• SHP2D2

Called to set up the approximation function and its local derivatives for 8-node two-dimensional elements. This subroutine is not presently used.

• SHP2D3

Called to set up the approximation function and its local derivatives for 9-node two-dimensional elements.

• SHP3D1

Called to set up the approximation function and its local derivatives for 8-node three-dimensional elements. This subroutine is not presently used.

• SHP3D2

Called to set up the approximation function and its local derivatives for 27-node three-dimensional elements.

• SINT3D

Called to set up quadrature rules of orders 1, 2 and 3 for three-dimensional elements. This information is needed in calculating the volume integrals in three-dimensional problems.

• SKYLUB

Called to solve the linear system of equations $\mathbf{AX} = \mathbf{b}$, where $\mathbf{A}$ has been stored in blocks on low-speed storage.

• STEERV

Called to form the steering vectors MLNODE and MLEAD. MLNODE contains the number of degrees of freedom at each nodal point and MLEAD contains the first position in the global matrix of unknowns that this node appears.

• WRGNDT

Called to print out the generated input data.

• WRITMS

Called to write information to a random access file.

• WRICAD

Called only once at the beginning of the first optimization iteration in initial control problems to assign the initial velocity field as the first control variable. It is not called in boundary control problems.

- WRTTLE

Called to print the title page in the output file.

- WRUADJ*

Called in optimization algorithm, after solving the Navier-Stokes equations, to store the solution field (velocity and pressure) to be used in solving the adjoint equations.

- ZERO

Called to initialize a real vector.

- ZERO2D

Called to initialize a real matrix.

- ZINT2D*

Same as ZINT3D for two-dimensional problems.

- ZINT3D*

Called in three-dimensional boundary control problems to compute the surface integral of the adjoint velocity (see equation (2.24)).

Appendix D

Validation of the Computer Code

Because of lack of experimental and numerical results on the optimal control problems described in Chapter 4, it is not possible to validate our numerical results. However, it is possible to test the results of our Navier-Stokes solver against other available results. Among the available sources, we decided to compare the results of our code FLOPTIM with those of the code FIDAP (FIDAP7.07, 1993) for a few geometries.

In all the following examples, 9-noded quadrilateral elements have been used and the steady-state Navier-Stokes equations have been solved by the penalty function method with a value of $\epsilon = 1.0 \times 10^{-7}$ for the penalty parameter, for both the present and the FIDAP codes.

D.1 Axisymmetric Step Flow

The geometry, the mesh and the boundary conditions are depicted in Figure D.1. A parabolic velocity profile with a maximum magnitude equal to 1.0 has been imposed on the inflow boundary. In Figure D.2, velocity vectors, streamlines, and pressure contours from FLOPTIM are presented for Re=60. The corresponding results from FIDAP are

shown in Figure D.3. The same problem was repeated with Re=200 and the results are presented in Figures D.4 and D.5 for FLOPTIM and FIDAP, respectively.

In the case of Re=60, both codes are in very good agreement in predicting the velocity profiles, streamlines and pressure contours. However, In the case of Re=200, the velocity profiles immediately after the circulation zone are slightly different. The above problems have also been numerically reported by Hughes *et al.* (1979).

D.2 Driven Cavity Flow

A problem description is shown in Figure D.6. The upper plate of the square cavity is assumed to have a dimensionless velocity of magnitude 1.0. The problem was solved for two different Reynolds numbers of Re=100 and Re=400. The results for Re=100 are shown in Figure D.7 and D.8 for FLOPTIM and FIDAP, respectively. In Figures D.9 and D.10 the corresponding results are presented for Re=400.

The results predicted by both codes are in very good agreement for both Re=100 and Re=400. In the case of Re=100, the center of vortex is predicted at (0.597, 0.736) by FLOPTIM and at (0.606, 0.732) by FIDAP. In the case of Re=400, these values are (0.569, 0.604) and (0.563, 0.598) for FLOPTIM and FIDAP, respectively.

D.3 Two-Dimensional Model in the Boundary Control Problem

In this section, the two-dimensional model of the boundary control problem in section 4.2.1 is studied. The geometry and the mesh are the same as shown in Figure 4.1. The inlet port is closed and the outlet port is open and a constant velocity of magnitude one is imposed on Γ_m. The Reynolds number was Re=241, the same value used in section

4.2.1. The velocity vectors and streamline contours are shown in Figures D.11 and D.12 for FLOPTIM and FIDAP, respectively.

A low velocity region at the top of the open port is similarly predicted by both codes. Streamlines and pressure contours are also in good agreement in both cases.

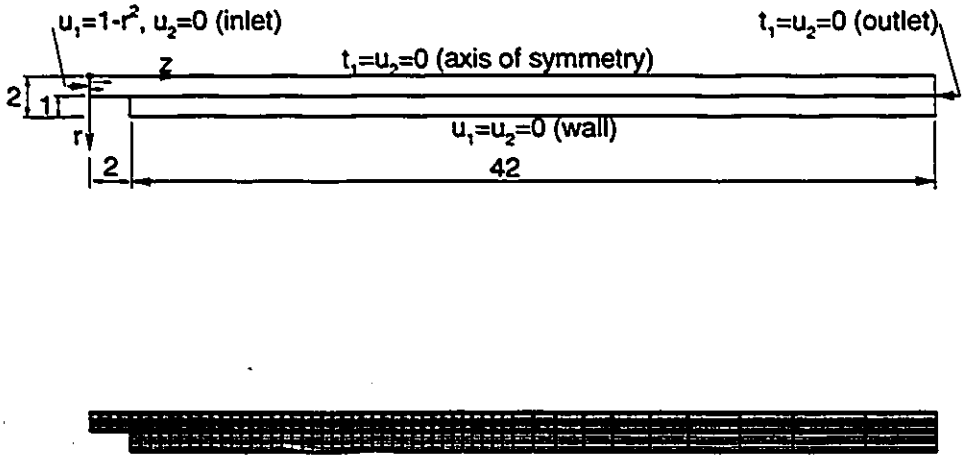


Figure D.1: Top: Problem description, and bottom: the mesh in axisymmetric step flow.

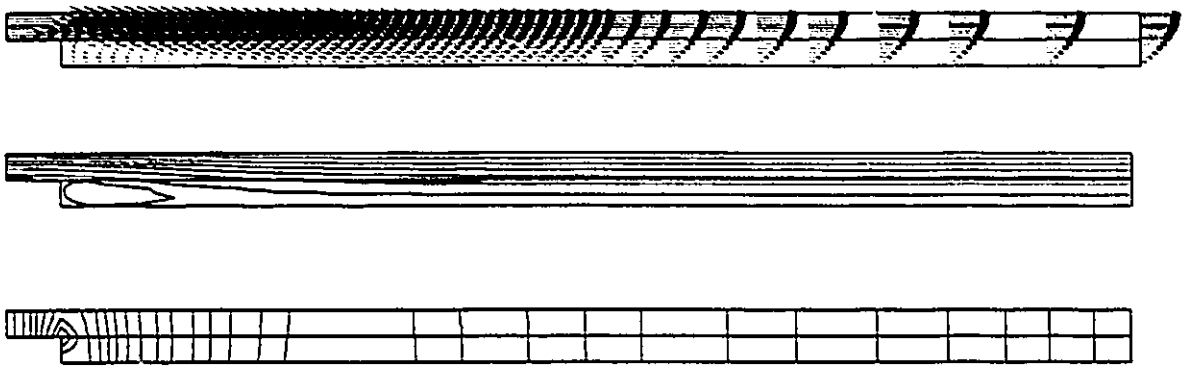


Figure D.2: Top: velocity vectors; middle: streamlines, and bottom: pressure contours from FLOPTIM (Re=60).

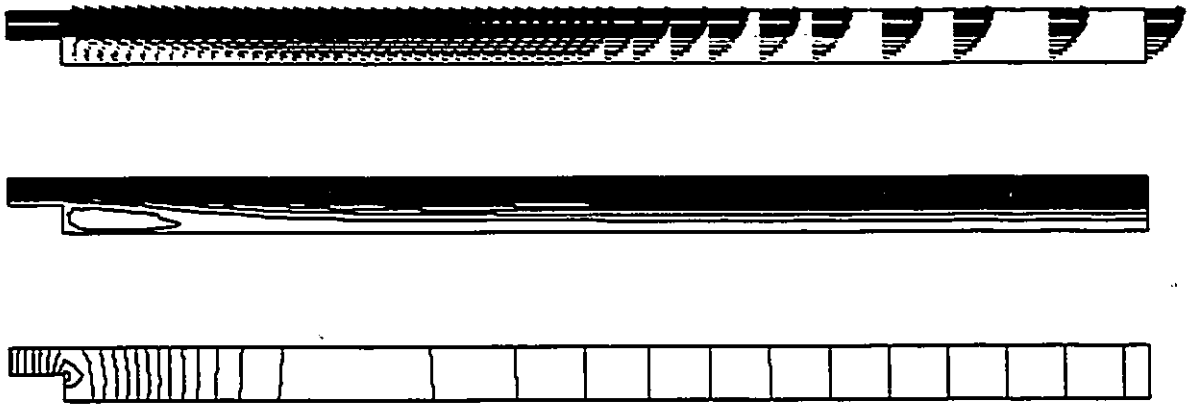


Figure D.3: Top: velocity vectors; middle: streamlines, and bottom: pressure contours from FIDAP (Re=60).

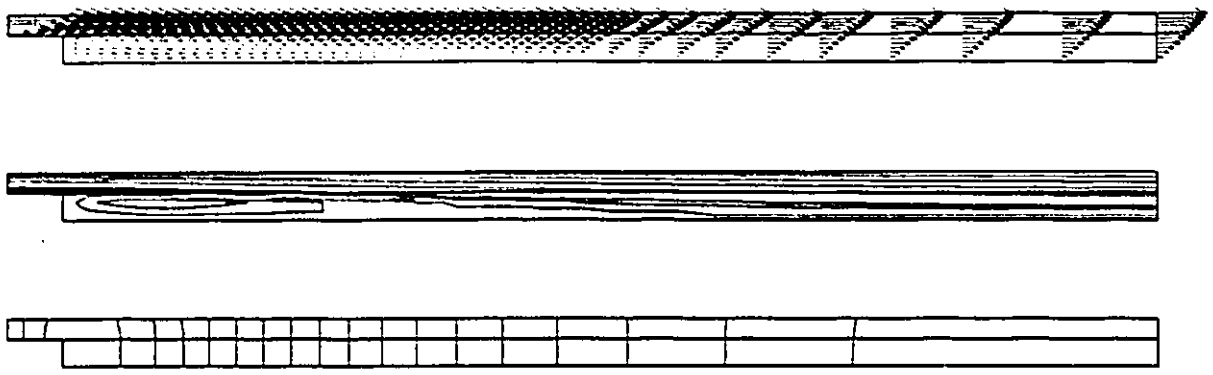


Figure D.4: Top: velocity vectors; middle: streamlines, and bottom: pressure contours from FLOPTIM (Re=200).

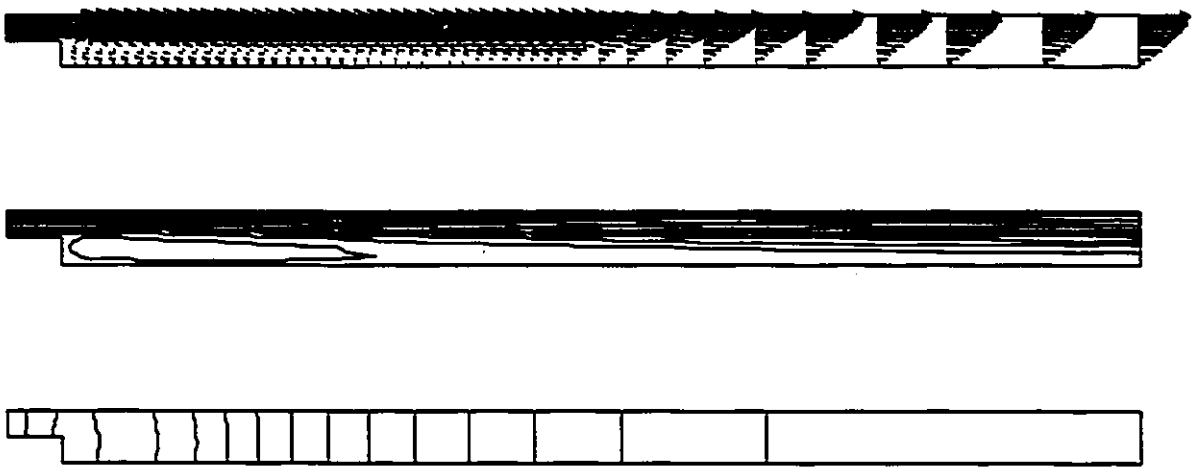


Figure D.5: Top: velocity vectors; middle: streamlines, and bottom: pressure contours from FIDAP (Re=200).

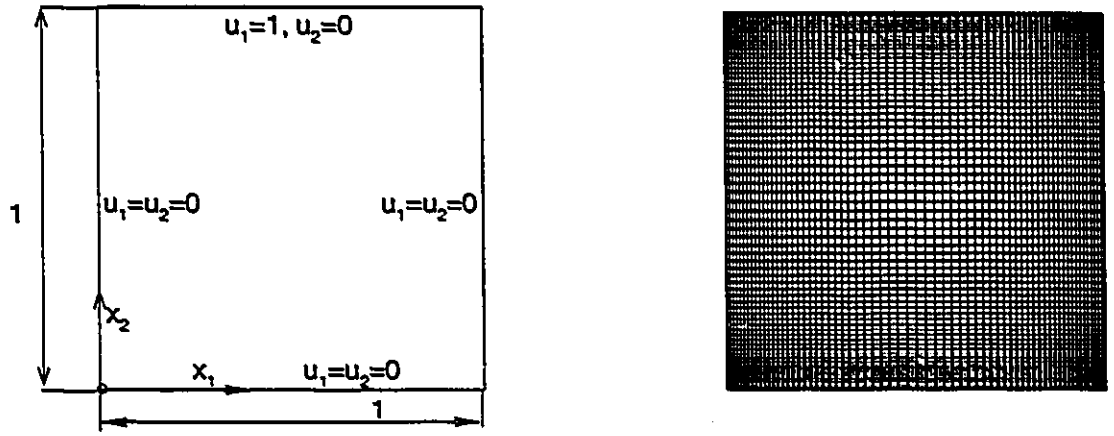


Figure D.6: Left: problem description, and right: the mesh in driven cavity flow.

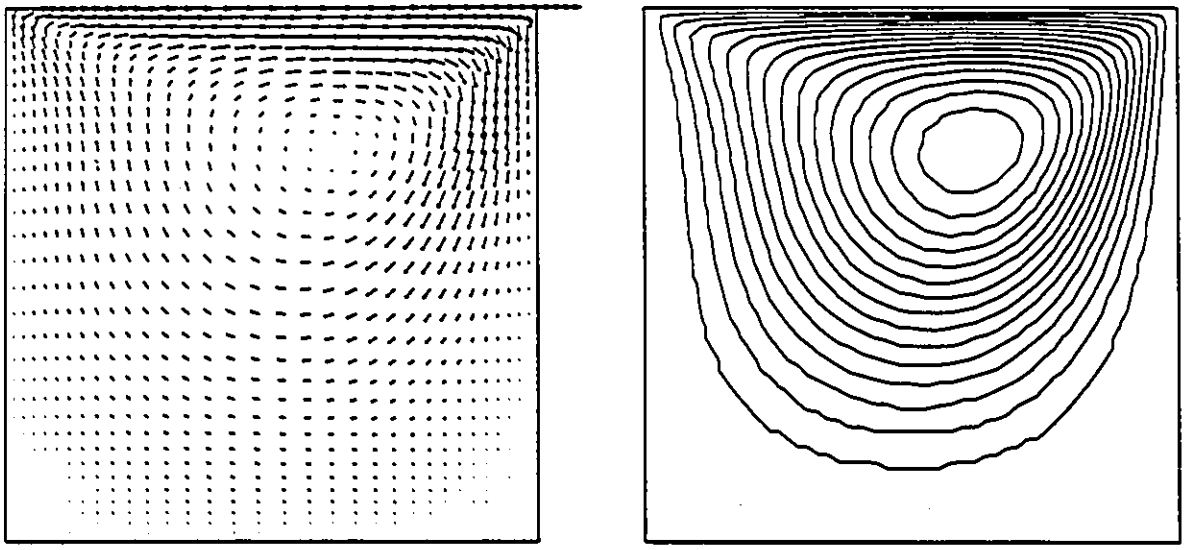


Figure D.7: Left: velocity vectors, and right: streamlines from FLOPTIM (Re=100).

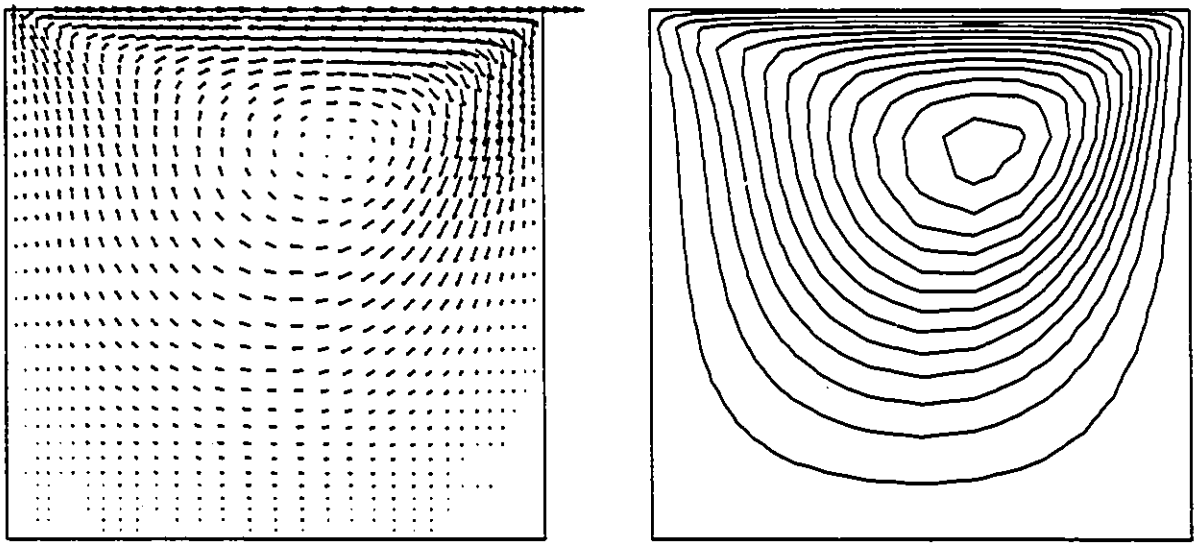


Figure D.8: Left: velocity vectors, and right: streamlines from FIDAP (Re=100).

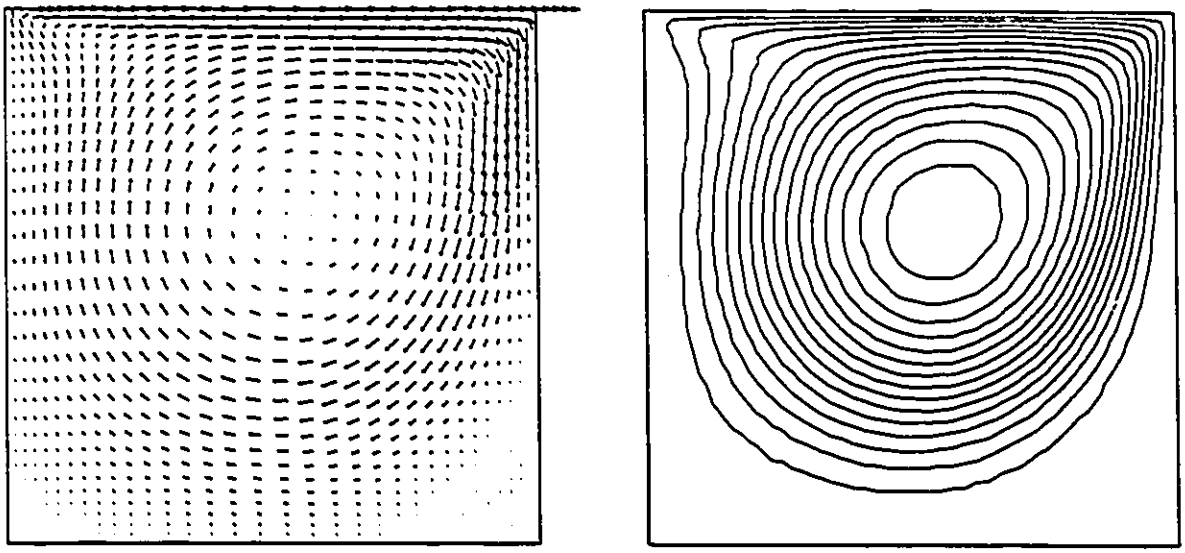


Figure D.9: Left: velocity vectors, and right: streamlines from FLOPTIM (Re=400).

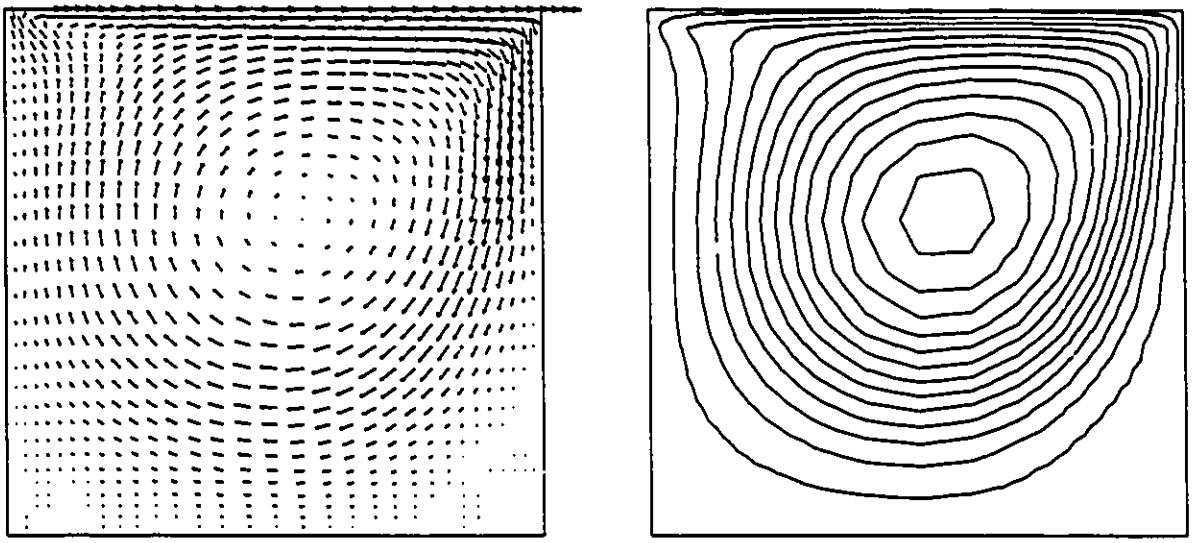


Figure D.10: Left: velocity vectors, and right: streamlines from FIDAP (Re=400).

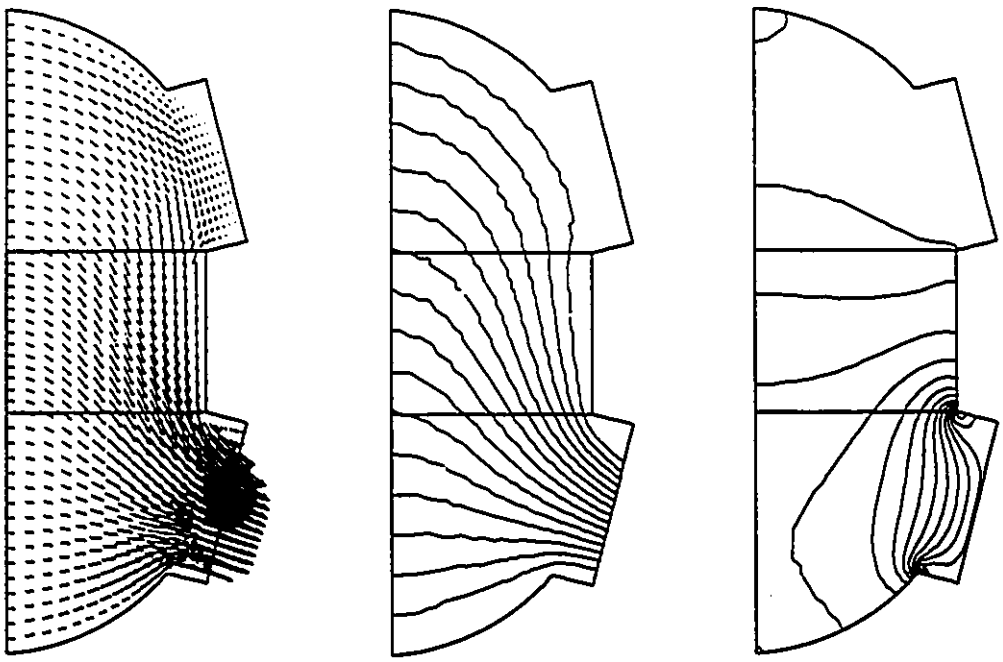


Figure D.11: Left: velocity vectors; middle: streamlines, and right: pressure contours from FLOPTIM (Re=241).

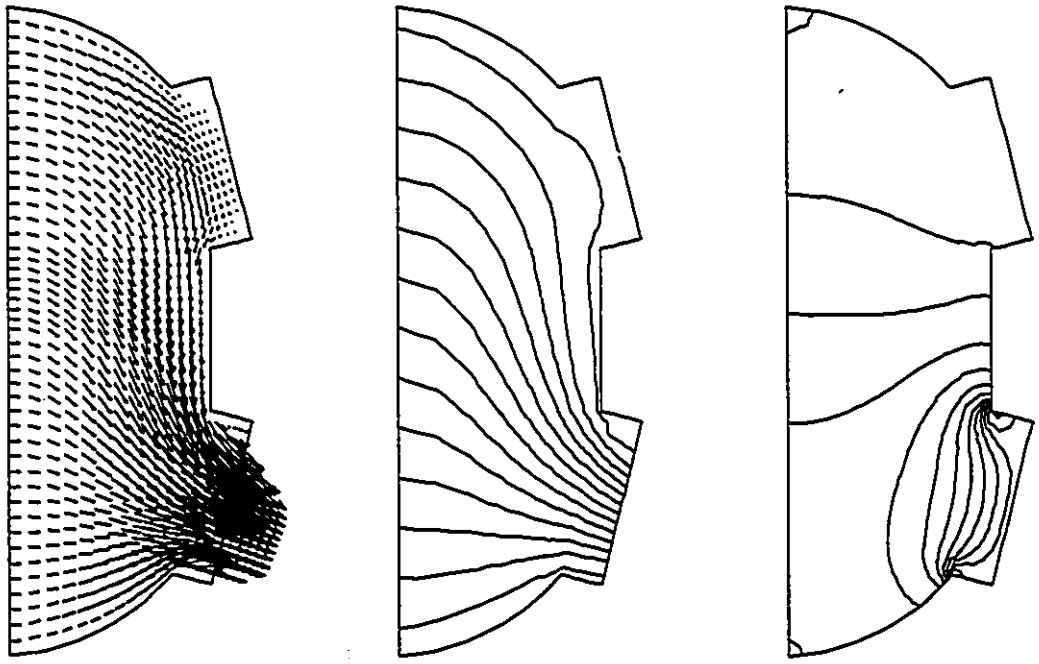


Figure D.12: Left: velocity vectors; middle: streamlines, and right: pressure contours from FIDAP (Re=241).