UNIVERSITY OF SOUTHAMPTON

Automated Multi-Stage Geometry Parameterization of Internal Fluid Flow Applications

Nicola Hoyle

Thesis for the degree of Doctor of Philosophy

Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

September 2006

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Nicola Hoyle

The search for the most effective method for the geometric parameterization of many internal fluid flow applications is ongoing. This thesis focuses on providing a general-purpose automated parameterization strategy for use in design optimization. Commercial Computer-Aided Design (CAD) software, Computational Fluid Dynamics (CFD) software and optimizer tools are brought together to offer a generic and practical solution. A multi-stage parameterization technique for three-dimensional surface manipulation is proposed. The first stage in the process defines the geometry in a global sense, allowing large scale freedom to produce a wide variety of shapes using only a small set of design variables. Invariably, optimization using a simplified global parameterization does not provide small scale detail required for an optimal solution of a complex geometry. Therefore, a second stage is used subsequently to fine-tune the geometry with respect to the objective function being optimized. By using Kriging response surface methodology to support the optimization studies, two diverse applications, a Formula One airbox and a human carotid artery bifurcation, can be concisely represented through a global parameterization followed by a local parameterization.

Contents

List of Figures

List of Tables

Nomenclature

A_e	2D length of intake exit location measured at the engine filter location [m]
A_o	2D length of intake inlet location [m]
B	a basis
C_p	static pressure recovery [Pa]
$\bar{C}_p$	mean static pressure recovery [Pa]
E	two-dimensional diffuser total expansion ratio = A_e/A_{in}
$E(I)$	Expected Improvement
I_d	identity matrix
I	improvement
L	likelihood
$\mathbf{R}$	correlation matrix
R	correlation between two sample points
N	diffuser axial length [m]
T	dimensionless width factor of Hicks-Henne bump
U	density-weighted inlet velocity [ms^{-1}]
V	volume
a	Hicks-Henne bump amplitude [m]
b	Bernstein polynomial
d	degree of polynomial
f	objective function
h	bump height (m)
k	number of dimensions
n	number of observed responses
p_o	mass-averaged static pressure at filter [Pa]
p_u	mass-averaged static pressure at inlet [Pa]
q_u	dynamic pressure [kgm^{-1}s^{-2}]
q	under-relaxation value
$\mathbf{r}$	vector of correlations
s^2	mean square error
u	blood velocity [ms^{-1}]
w_A	area of negative shear region [m^2]

$\mathbf{x}^*$	unsampled point
x_p	normalized distance of Hicks-Henne bump peak along diffuser centerline from diffuser inlet, $x_p \in [0, 1]$
$\hat{y}$	the predictor

Greek	
Φ	radial basis function kernel
α	basis function
β	wall contouring parameter
δ_{ij}	Kronecker delta where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$
λ	Lagrange multiplier
ρ	fluid density [kgm^{-3}]
η	blood viscosity [kgm^{-1}s]
μ	mean
$\hat{\mu}$	maximum likelihood estimator of the mean
σ^2	variance
$\hat{\sigma}^2$	maximum likelihood estimator of the variance
θ, p, ζ	kriging hyperparameters
$\bar{\tau}_-$	negative time-averaged shear stress [Pa]
τ_w	wall shear stress [Pa]
ϕ	nodal value of transport variable
ψ_f	face value of transport variable

Abbreviations	
ALLF	Augmented Log-Likelihood Function
AMSSOD	Automated Multi-Stage Shape Optimization with Deformation
BLUP	Best Linear Unbiased Predictor
CAD	Computer Aided Design
CAM	Computer Aided Modelling
CCA	Common Carotid Artery
CFD	Computational Fluid Dynamics
CLLF	Concentrated Log-Likelihood Function
DHC	Dynamic Hill Climber
DoE	Design of Experiments
DNS	Direct Numerical Simulation
ECA	External Carotid Artery
F1	Formula One
FFD	Free-Form Deformation
GA	Genetic Algorithm
GEODISE	Grid-Enabled Optimization and Design Search for Engineering
ICA	Internal Carotid Artery

LLF	Log-Likelihood Function
MLE	Maximum Likelihood Estimate
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
NURBS	Non-Uniform Rational B-Spline
NURPS	Non-Uniform Rational Polynomial Spline
RANS	Reynolds-Averaged Navier Stokes
RBF	Radial Basis Function
RSM	Response Surface Model

Declaration of Authorship

I, Nicola Hoyle, declare that the thesis entitled

"Automated Multi-Stage Geometry Parameterization of Internal Fluid Flow Applications"

and the work presented in it are my own. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as Hoyle et al. (2005) and Hoyle et al. (2006).

Signed:

Date:

Acknowledgements

I am deeply grateful to my supervisors, Professor Andy Keane and Dr Neil Bressloff, for their guidance and support throughout my PhD. Both have been a constant source of intellectual stimulation and their research ethos has made these last three years thoroughly enjoyable.

I would also like to express my sincere gratitude to Valerie Hoyle, Matthew Hill, Alex Forrester, Andras Sobester, Tom Barrett and Vijay Bhaskar for their editorial support; Alex, Andras and Tom for their constant inspiration, encouragement and entertainment throughout my studies; Tony Scurr for his provision of milk and lunchtime conversation; and Jen Forrester for her kind hospitality.

The department that I have had the privilege to work in has been fantastic throughout the last three years. I wish to thank all who work within it for their friendliness, helpfulness and for making it such a great place to work.

On a different note, I must thank Matthew whose resolute support both morally and emotionally has been wonderful. His technical help has also been invaluable throughout my studies.

Last, but by no means least, I would like to thank my long-suffering family who have been there for me through every stage of this PhD and have supported me financially and morally throughout my education. In particular, I have to thank Mum for always being around during the stressful times, and for keeping me well fed; Christina for providing useful nutritional lessons and Dad for his encouragement and for keeping my fridge stocked with wine.

This work has been supported by the Engineering and Physical Sciences Research Council.

To mum and dad

Chapter 1

Introduction

In today's society, it is easy to forget how far the human race has progressed through increased use of technology over the last 100 years. Perhaps the most distinguishing manifestations of this historical era are the developments of the motor-car and the aeroplane. Both have brought a revolution in transport that has established a contemporary lifestyle entirely different from any that preceded it. The motor-car industry has grown to such an extent over the last century that its booms and slumps have the ability to unsettle governments, an economic theory endorsed by a former president of General Motors, Charles E. Wilson: *"What is good for the country is good for General Motors and vice-versa"*. The growth in transport by aeroplane has also been immense as many people now travel across the continents of the world by aeroplane, both on business and on holiday. As recently as ten years ago, most people would not have imagined that they would be able to travel from England to the south of France, albeit on a low-budget airline, for less than the cost of a train fare from Brighton to London.

The development of technology has spawned this growth, and most recently this has been accelerated by the increased availability and capability of the digital computer. From the mid-20th century onwards, not only were machines used to calculate performance data using analysis software, but also they slowly infiltrated design offices, with the development of Computer-Aided Design (CAD) software providing an efficient alternative to hand-drawn blueprints. Another important advance that occurred during this period was the development of software to allow computational modelling of fluid flow behaviour; that is Computational Fluid Dynamics (CFD).

The aerospace and the car industries primarily have been responsible for the development of these systems. In the present day, computers, along with in-house and commercially available CAD and CFD software, provide an indispensable support to the field of engineering. The impact of many diverse engineering applications on a broad spectrum of our everyday lives has provided a need to acquire a greater comprehension of flow physics particularly for the purposes of design. This can be accomplished with the use of optimizers in conjunction with CAD and CFD packages, but to be successful such an optimization must operate on an effective geometric description. What follows is a brief history of CAD and the use of CAD within CFD-based optimization studies. Therein, the motivation of the work documented in this thesis is discussed and this chapter concludes with an outline of the material covered in the following chapters.

1.1 Computer-Aided Geometric Design

1.1.1 A Brief History

The first recorded use of curves within a manufacturing environment was in the early Roman shipbuilding industry. A ship's ribs, or the wooden wireframe structure joined together at the keel defining the shape of the hull, were produced using templates which could be reused repeatedly. Any ship's hull could then be produced by modifying the geometry of the ribs.

Before the advent of computers, parametric curves were drawn with a high level of precision using a set of templates known as *French curves*: carefully designed wooden sections of conics and spirals. A curve is drawn by following the required sections of a French curve. Another tool used for the drawing of smooth parametric curves, mentioned in the work of du Monceau (1752), is known as a *spline*. This apparatus comprises a flexible piece of wood that is gently bent and held in place at discrete points with metal weights, known as *ducks*; see Figure 1.1. The curve is the shape created by the position and weight of the ducks. For large scale drawings produced at this time, attics (or lofts) of buildings were used to accommodate them - the word *lofting* has its origin here.

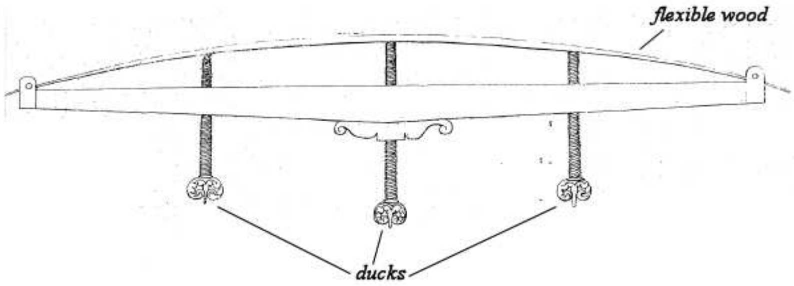


Figure 1.1: Mechanical spline tool (illustration given by Raalamb (1691))

It was not only the shipbuilding industry but also the aircraft industry that provided foundations to the field of Computer-Aided Design. Customarily, the construction template of an aircraft was defined by a series of conics which were drawn by draughtsmen and stored in the form of blueprints. A more efficient alternative was realised by Liming in 1944 (Liming, 1944). This involved storing a design in terms of a set of numerical variables instead of hand-drawn curves, and in doing so translated classical draughting techniques into numerical algorithms. However, the method to transform hand-drawn blueprints on the draughtsman's drawing board to mathematically defined curves and surfaces for computational representation was not clear.

In the 1950s, digital computers began their infiltration into design offices and Boeing developed and employed software based upon Liming's work in the design of fuselages. For the design of wings, however, a different kind of curve was developed by Boeing employees J. Ferguson and D. MacLaren. This was the origin of what we now know as *spline curves*, the mathematical counterpart of the mechanical spline. Although the first mathematical reference to splines was presented by Schoenberg (1946), Ferguson and MacLaren's idea was to piece cubic space curves together to form twice differentiable composite curves used in the geometrical definition of wings (MacLaren, 1958; Ferguson, 1964). Because of this, the curves were capable of interpolating easily through a set of points by minimizing a function similar to the physical properties of the mechanical spline tool. In modern parlance, the spline referred to in today's CAD world is instead thought of as the smoothest piecewise polynomial curve that passes through a set of fixed points.

During the '50s and '60s, many institutions and industries worked on constructing computational curves and surfaces mostly in isolation. However, although it had several

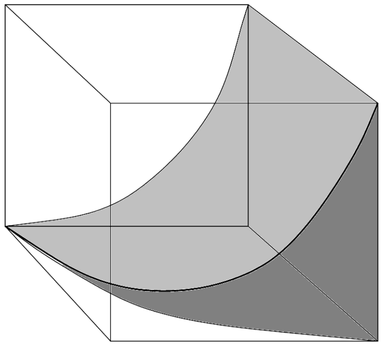


FIGURE 1.2: Bézier's basic curve

independent beginnings, the foundations of modern CAD engines subsequently became largely established in the French car industry. It was in 1959 that Citroën hired a young mathematician, Paul de Faget de Casteljau, in order to resolve some of the theoretical problems that had arisen in the transition from the physical to computational representation of parts. Unlike Liming's approach, de Casteljau initially built a system principally aimed at the *ab initio* design of curves and surfaces instead of concentrating on the computational duplication of their existing blueprints. From the start, he implemented the use of Bernstein polynomials with what is now known as the *de Casteljau algorithm*, and in doing so pioneered a new technique: *control polygons (courbes à pôles)*. Instead of defining a curve (or surface) through points *on* it, a control polygon utilizes points *near* it. This meant that the curve (or surface) was not changed directly; instead, the alteration of the control polygon itself instigated an intuitive change in the curve (or surface). This work was kept secret by Citroën until the algorithm was first published by Krautter and Parizot (1971).

Concurrently, Citroën's competitor Renault had also realized the need for computational representations of mechanical parts. Renault's design department was headed by Pierre Bézier who, although he was aware of similar developments at Citroën, proceeded to look at the theoretical problems of the transition independently. Using de Casteljau's algorithm, Bézier's initial idea was to characterize a "basic curve", defined as the intersection of two elliptic cylinders; see Figure 1.2. These two cylinders were defined inside a parallelepiped, affine transformations of which would result in affine transformations of the curve. Later, polynomial formulations were developed and subsequently extended to higher degrees.

It was not until the 1970s that there began a confluence of these different research approaches. In 1974, R. Barnhill and R. Riesenfeld used *the construction and representation of free-form curves, surfaces or volumes* to define a now very familiar term: Computer-Aided Geometrical Design or CAD. For further reading on the history of CAD, please refer to Farin (2002).

Bézier's work was widely published (Bézier, 1967, 1968, 1974, 1977) and Forrest's article (Forrest, 1972) on Bézier curves contributed greatly to the popularity of Bézier curves. UNISURF, the Renault CAD/CAM system, was developed solely to use Bézier curves and surfaces. Dassault followed in Renault's footsteps and built the system EVE, later evolving into CATIA (Computer Aided Three-Dimensional Interactive Application) (CATIA®, 2004). Today, CATIA, along with other commercial CAD packages, facilitates the use of both parametrically defined and "free-form" curves and surfaces. This ability to represent a design using an efficient mathematical model allows the CAD software to be coupled to an optimization process.

1.1.2 The Role of Geometry Parameterization in CFD-Based Design Optimization

With the progress of CAD and CFD software, automated optimization processes using these computational tools have proven popular, allowing hi-tech industries to produce numerous computational designs quickly and relatively inexpensively. These designs can be analysed with respect to an appropriate measure of merit, evaluated and modified, and thus updated in a cyclical manner until a final optimal design is reached. Providing efficient design optimization processes has created a hub of activity within engineering research (Siddall, 1982). The enhanced efficiency of optimization techniques has enabled the search of larger design spaces in which optimal designs can be found in numerous varied applications; see Keane and Robinson (1999).

Efficient optimization principally relies on concise sets of design parameters defining the geometry under examination. When a concise set of parameters is not readily available, designers may forego the potential to produce radical designs with a superior measure of merit. For previously tested and understood concepts and designs, design optimization is sometimes seen as a gradual evolution and improvement.

In cases where there is a limited understanding of the flow behaviour associated with the geometry under scrutiny, a reduced geometric control capability may prove detrimental in finding an optimal design. Ideally, the geometry parameterization would allow maximum control of the geometric shape whilst preserving a concise set of design parameters for the purpose of an efficient optimization process. However, the form of parameterization itself is often unclear. The literature on geometry parameterization techniques is substantial. Samareh (2001) surveyed a number of available techniques and assessed each on its suitability in dealing with complex models. It is clear that for design optimization studies, the particular geometry parameterization technique implemented can have an enormous impact on the final outcome. If a wing geometry is parameterized simply with planform and chord variation describing its shape, the optimal design may be localised to the area of the wing which these two particular features affect and so will not capture the true "global" nature of the wing shape that one may need to consider, including factors such as twist and sweep. A global geometry needs to be considered during a complete design process, and particularly in conceptual design. Local parameterization methods can be considered once a reasonably good global design has been reached, either through an initial optimization process or through a proven best pre-existing design.

One of the first studies of an optimal condition with its analysis described by the mathematical theory of fluid dynamics was performed by Glowinski and Pironneau (1975). Subsequently, aerofoils became a popular subject for CFD-based optimization problems characterized by Hicks et al. (1974) and Jameson (1988).

As a result of the extensive research performed on aerofoil and wing parameterization, it has become accepted to parameterize the overall shape of external wings through a set of well-known parameters: chord, span, planform, twist, sweep and shear. However, for internal fluid flow applications such as a diffuser, its parameterization cannot be defined using such well-known geometric quantities. In this case, manipulating the geometry to achieve a certain physical flow behaviour is a road less travelled.

Diffusers have been the subject of optimization since the late 1950s. Early experimental work classified the major flow regimes within straight diffusers (Kline et al., 1959; Fox and Kline, 1962). Relationships were deduced between these flow regimes and the diffuser characteristics (Reneau, 1967) whilst, concurrently, simple geometry parameterizations gave room for more efficient designs; see Carlson et al. (1967). Although still experimental at this stage, this was an important start in recognizing the impact of geometry parameterization on the optimization process using analysis codes instead of practical experiments to determine results. In the last few decades, maximum optimizer efficiency for aerodynamic utilization has been sought, implying that a geometry parameterization containing a concise set of design variables is desirable. Madsen (1998), Madsen et al. (1999) and Madsen et al. (2000) have highlighted the use of geometry parameterization in optimization studies for straight diffusers and Ghate et al. (2004) parameterized an S-shaped duct for optimization.

The development of parametric models within modern CAD engines is a key area of research. CAD, originally developed as an alternative to the drawing board, allowed for the improvement of design productivity and accuracy when first introduced. Common engineering shapes were soon parameterized and linked within the design part inside the CAD software to facilitate the automatic manipulation of the computational geometry to accommodate the required change. This eliminated the need for the designer to redraw the part in CAD and, in turn, made it practical to rapidly produce numerous design variations. The ability to differentiate between the large number of designs produced provided the link between CAD and analysis software such as CFD to optimize designs.

CAD-based parameterization has distinct advantages over other approaches, as it enables large alterations from the original shape without necessarily destroying the shape topology. In addition, the less complex shape configuration does not require more advanced tools to enable mesh deformation in concurrence with shape change. Almost all engineering firms use CAD as an integral part of their design process. It is because of this that the scope of this thesis is limited to CAD-based parameterization techniques, aiming to offer a generic, practical and industrially realistic solution.

1.2 Background and Objective of work

In this thesis, efficient and flexible methods of geometry parameterization are sought for use in an automated design optimization. Initially these are developed for use with a F1 airbox. This particular internal flow duct has been chosen due to the author's interest in the sport and the lack of agreed practice in this particular industrial sector. After a thorough study of techniques used on plane geometries, the best and simplest parameterization for obtaining a good global shape is selected. Following this, a geometry manipulation technique is developed and used in a general automated design optimization process for a three-dimensional geometry.

To illustrate the generic capability of the parameterization technique developed, a second case study has been tested: the shape optimization of a human carotid artery bifurcation. The choice of parameterization technique poses a similar problem but in this case the shape fit of the parametric model to that of a real artery geometry is optimized, aiming to achieve a close likeness.

Early motoring was seen as a new and somewhat dangerous form of outdoor sport which presented a new element - the ever-changing machine - for the sportsman to contend with. Not only has the domestic car improved on a huge scale, the sport of motor-racing has developed alongside, the pinnacle of which for both drivers and manufacturers is the Formula One (F1) World Championship.

The principles of aerodynamics developed from early aviation have been passed into the automotive industry and used to enhance the performance of racing cars. With the advent of the F1 World Championship, chassis design, engine technology, suspension technology and aerodynamic aids improved. F1 cars became faster, more agile and more spectacular to watch. The rapid development of computational power has permitted the feasibility of computational design, and latterly, shape optimization.

The design of engine air intakes, in particular those used in F1, has become a significant consideration as engines continually improve in sophistication and performance. Intake design seeks to maximize static pressure acting on the intake stroke of the engine cylinders. High static pressure over the cylinders increases the cylinder charge density and hence engine power. The design of the airbox geometry, including its bend through 90° and the position of the air filter element, all have an impact on both the static pressure recovery and cylinder-to-cylinder air distribution and thus engine performance. During the first few decades of the 20th century, it was known that diffusers could convert kinetic energy at the diffuser entry into static pressure at the exit, albeit with low efficiency. Improvement of the efficiency of this effect started in 1938 (Patterson, 1938). F1 aerodynamicists have studied engine air intakes since the 1950s. These began as small air vents in the engine cover bodywork over the cylinders in front of the driver. Ten years

Figure 1.3: Airbox inlet (2003 season)

later, the introduction of rear-engined cars left the engines exposed with no covering bodywork. By 1972, the teams had designed large scoop-like airboxes sitting above the driver's head. Safety, however, increasingly became an issue and roll bar structures were introduced. Two large scoops either side of the roll bar then became the norm, reducing in size through the early eighties until, in 1989, airboxes appeared akin to those seen today, see Figure 1.3.

F1 is a highly competitive sport and so time, cost and good results are critical. Careful design of individual components can often provide the necessary advantage to enhance performance. However, regulation constraints may limit the level of design improvement. For modern air intakes, expansion of the flow is required over a short distance while turning through 90° due to the roll bar specifications and engine layout configuration; see FIA (2005). During the time of the present study, the F1 teams place a 3-litre V10 engine behind the driver (see Figure 1.4); the position of the airbox thus takes advantage of the ramming effects of the oncoming air at high speeds. The engine filter is located over a trumpet tray, at the bottom of which sits an offset array of 10 engine inlet trumpets, one for each of the cylinders.

For the 2006 season, the engine specification has changed to 2.4-litre V8 engines. The airboxes are very similar in their positioning to the illustration in Figure 1.4, the only difference between the airboxes contained in this thesis and the ones that will be seen on the track in 2006 is the number of cylinders: 10 instead of 8. Interestingly, during the pre-2006 season testing when the V8 engines were not yet ready for the track, the cars were seen with an airbox inlet adaption, illustrated in Figure 1.5. By limiting the

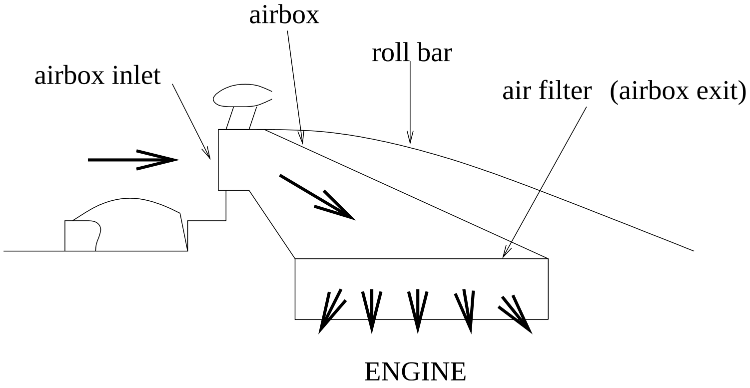


FIGURE 1.4: Airbox positioning within the F1 car

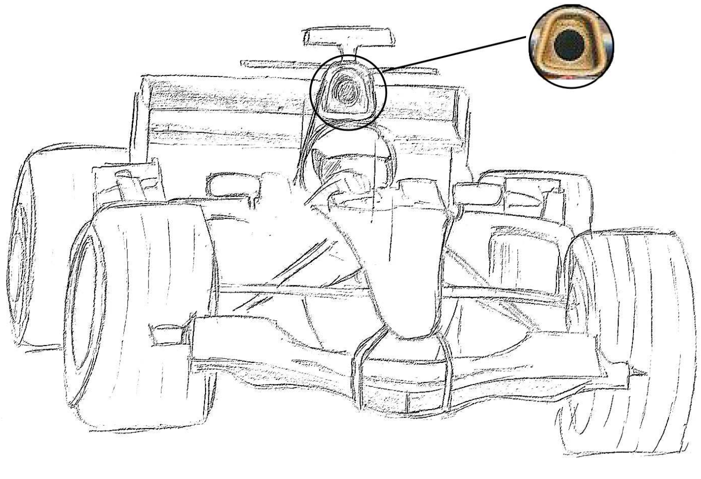


FIGURE 1.5: Restriction of entry area to limit engine power (see inset), seen to have been used on certain F1 cars during the pre-2006 season testing

air to the engine, the performance was reduced sufficiently to simulate the performance that would be experienced from the new, less powerful, engine.

In comparison to race car design, arterial geometry parameterization is a relatively new topic of research. However, the use of computational modelling has become a powerful research tool in aiding the understanding of arterial biomechanical behaviour and its relation to atherosclerosis, a common disease which can lead to stroke, heart attacks, eye and kidney problems. A detailed and informative review of computational techniques currently being used for research into patient-specific biomechanics for potential treatment decision making can be found in Steinman et al. (2003). Although surgeons have not yet accepted these techniques for routine clinical use, an improved understanding of local haemodynamics in a broad variety of different conditions, including the effect after surgical intervention, may lead ultimately to the possibility of a patient-specific predictive medicine approach to surgical intervention. Taylor et al. (1999) has adopted this approach for the planning of bypass surgery, Guadagni et al. (2001) discusses optimizing the treatment of congenital heart defects and Cebral et al. (2000) and Steinman et al. (2002a) uses the approach for cerebral aneurysms.

It is widely accepted that internal wall geometry is correlated with the sites of atherosclerosis. Early experimental and CAD models of the carotid artery bifurcation were highly idealised as “Y-shaped” models. Although better approximations are now used through “tuning fork” artery shapes, they can be applied only in a general sense. Examples of a Y-shaped artery model and a tuning fork artery model in comparison with the shape of a real artery model can be seen in Figure 1.6. Much more sophisticated image processing techniques have been developed for 3D geometry reconstruction of arteries based on Magnetic Resonance Imaging (MRI) (Steinman, 2002; Steinman et al., 2002b; Antiga and Steinman, 2004). This method captures the large variation in shape and dimensions of the arteries from patient to patient. For patient specific analysis it is important to capture an accurate computational representation of the artery for accurate flow simulations. Through accurate CFD modelling in realistic arteries, doctors are beginning to understand the link between the arterial haemodynamics, other physiology and the build up of disease. However, parametric models are also of importance to develop an understanding of how changes in arterial wall shape affect the haemodynamic behaviour. The industry, at present, although possessing powerful visualization tools to represent and reconstruct extremely accurate computational arteries, lacks a realistic parametric model for use in computational research. One key benefit of a realistic parametric geometry is to enable research to determine those patients for whom interventional medicine may be favourable. For example, by understanding the haemodynamics connected with key geometrical factors, a connection between arterial geometry and a pre-disposition of lower leg iscaemia in certain diabetics may be discovered. The techniques developed for studying F1 airboxes have been adopted here for building arterial models.

The main objective of this thesis is to develop a fully automated general-purpose parameterization strategy providing a global manipulation of the geometry surface followed by a local surface deformation for use in the design optimization of three-dimensional internal fluid flow applications.

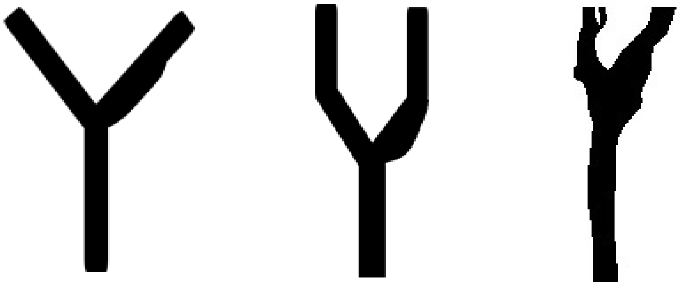


FIGURE 1.6: Y-shaped artery model (left), tuning fork artery model (centre) and real artery model (right)

1.3 Thesis Outline

In Chapter 2, the setup of an automated design optimization process is discussed. Each part of the process is described in detail and the geometry parameterization techniques available in commercial CAD software packages are surveyed. The implementation of a Design of Experiments (DoE) approach followed by the use of CFD is described and response surface methodology is proposed as the optimization approach used in this thesis. Response surface methodology is a means of quickening an exhaustive search process to find an optimal design.

Chapter 3 contains a mathematical description, in general terms, of response surface approximation theory. Kriging is the focus here, and a discussion of the various search techniques is given. Furthermore, for computationally expensive cases, a concentrated exploration in a reduced area of the response surface for additional improvement in design and convergence towards an optimum is described.

Chapter 4 implements a number of techniques surveyed in Chapter 2 in the design optimization of a plane straight expanding duct and a constant width elbow turning through 90°. The results of these studies are compared and each technique is analysed with respect to the number of design variables required and the amount of global shape control given. As the straight diffuser and elbow comprise both the features of an airbox in terms of expansion and turning of the flow, a suitable parameterization technique with good global shape control for a plane F1 airbox is then proposed and implemented in a 2D design optimization study.

The purpose of Chapter 5 is to propose a general automated multi-stage design optimization process for three-dimensional internal fluid flow applications with the capability of providing both a globally parametric representation of the geometry as well as a locally adaptable representation. The first stage is a global technique allowing the identification of a generally good geometry, and the second stage is a local technique providing features unavailable with the global technique for fine-tuning the geometry to gain improvements towards an optimal design representation. This chapter presents a survey of 3D surface representations, and from the conclusions drawn, the most appropriate surface manipulation techniques are chosen and a general automated process outlined.

In Chapter 6, the automated multi-stage optimization process built up in previous chapters is applied to a three-dimensional F1 airbox. A parametric geometry is constructed and a good global geometry found. The second stage of the process is implemented and local deformations of the airbox surface are optimized to achieve a high performance geometry.

Chapter 7 considers the multi-stage optimization process developed in Chapter 5, but now applied to the shape optimization of a realistic parametric human carotid artery bifurcation model. The initial artery geometry is determined from an automated analysis of scanned data of a real artery. The error between the parametric CAD model and the real artery is minimized through a local geometry manipulation stage. CFD analysis is performed on the resulting CAD model and compared with a CFD simulation through the ‘real’ artery.

Finally, in Chapter 8, the contributions made are noted and general conclusions drawn from the work presented along with suggestions for future research resulting from these investigations.

Chapter 2

Design Optimization Methodology

Optimization, in short, can be described as the action or process of rendering the most favourable outcome under a particular set of circumstances. Optimization as a concept is familiar to all as, instinctively, we search for the best solution given a set of circumstances in many everyday activities. This general concept of optimization is ubiquitous in countless applications, for example, in engineering design, biomechanics, weather prediction, econometrics and financial forecasting to name but a few.

In formulating a design optimization problem, we wish to find the best solution to a specific problem defined by a finite number of design variables, such that a desired performance criterion can be maximized (e.g. recovery time, fuel efficiency, profit, etc.) or minimized (e.g. aerodynamic drag, weight, loss, etc.). This criterion can be expressed explicitly in terms of an 'objective function'. In addition to this, limitations or 'constraints' may be imposed (e.g. physical size, manufacturing capability, economic). By systematically adjusting the values of all design variables, a 'good' (feasible) or 'best' (optimal) solution may be found.

In mechanical design an optimization framework commonly incorporates geometry construction, analysis and post-processing software, each of which has developed throughout the course of the last 40 years, often independently. As a result, difficulties often arise in determining an optimal solution when an efficient automation of the process cannot

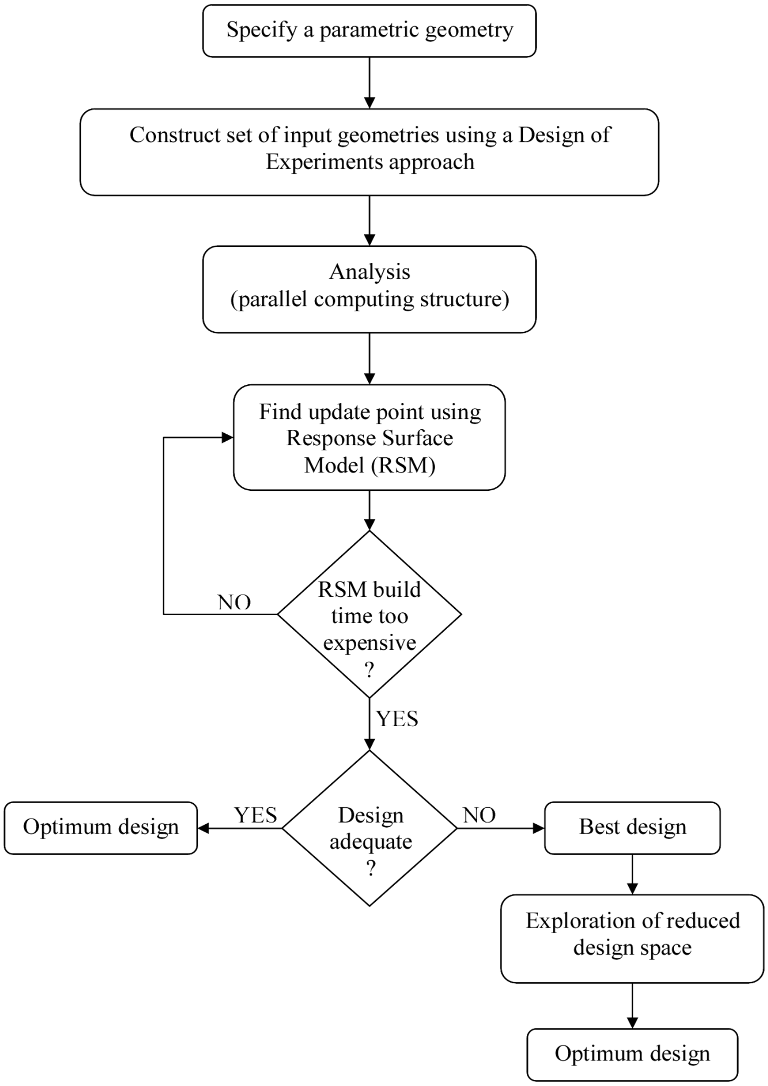


Figure 2.1: A generalised optimization strategy

be achieved due to software incompatibility. In an optimization process one must accept possible limitations generated from software integration difficulties.

A typical design optimization framework is illustrated in Figure 2.1, each step of which is discussed in further detail in the subsequent sections, giving a description of the automated system architecture employed throughout this thesis.

2.1 An Automated System Architecture

One of the primary issues faced by industries employing various software technologies is that of the incompatibility between the software input and output file requirements. To circumnavigate these issues, in-house codes may be written largely to provide a seamless integration of software. However, due to the nature of industry, these codes are rarely homogeneously shared across all business units resulting in wasted effort, time and money as different units produce their own ad hoc solutions.

To remain consistent with the use of only commercially available design and analysis software in this thesis, the design optimization process is incorporated within the Grid-Enabled Optimization and Design Search for Engineering (GEODISE) system (Geodise, 2002). The GEODISE system is implemented using the Matlab ® environment (Matlab ®, 2002) as an engineering portal giving remote access to the required CAD software CATIA V5 ® Dassault Systèmes, analysis software Fluent$^{\mathrm{TM}}$ (2003a) and optimization software OPTIONS (Keane, 2002). Matlab is adopted due to its prevalence in the engineering fraternity, making the toolkit flexible and easily extendible.

As the use of optimization in design is becoming more commonplace, and designers are demanding evermore accurate simulations, larger models are being tested requiring CFD computations many orders of magnitude larger than the optimization methods themselves. This in turn requires greater computational resources making this process well suited to the use of Grid computing (Foster, 2002; Eres et al., 2005). For many designers the integration of several heterogeneous environments and/or incompatible software on such a large scale would be a daunting undertaking. The development of Grid technologies with the Open Grid Services Architecture (OGSA) (Foster et al., 2002) and the Open Grid Services Infrastructure (OGSI, 2003) has allowed this type of service-orientated computation to become easily adopted.

The GEODISE toolkit is composed of a hierarchy of components which is depicted in Figure 2.2. Each box, from the scripting environment through computation to the applications, are exposed as Grid services and are connected appropriately in this service-based workflow; see Xue et al. (2004). Low level compute toolbox functions are available along with the input scripts for the OptionsMatlab toolbox. Users can then access remote compute resources such as the Condor Cluster shown in Figure 2.2, where the Globus

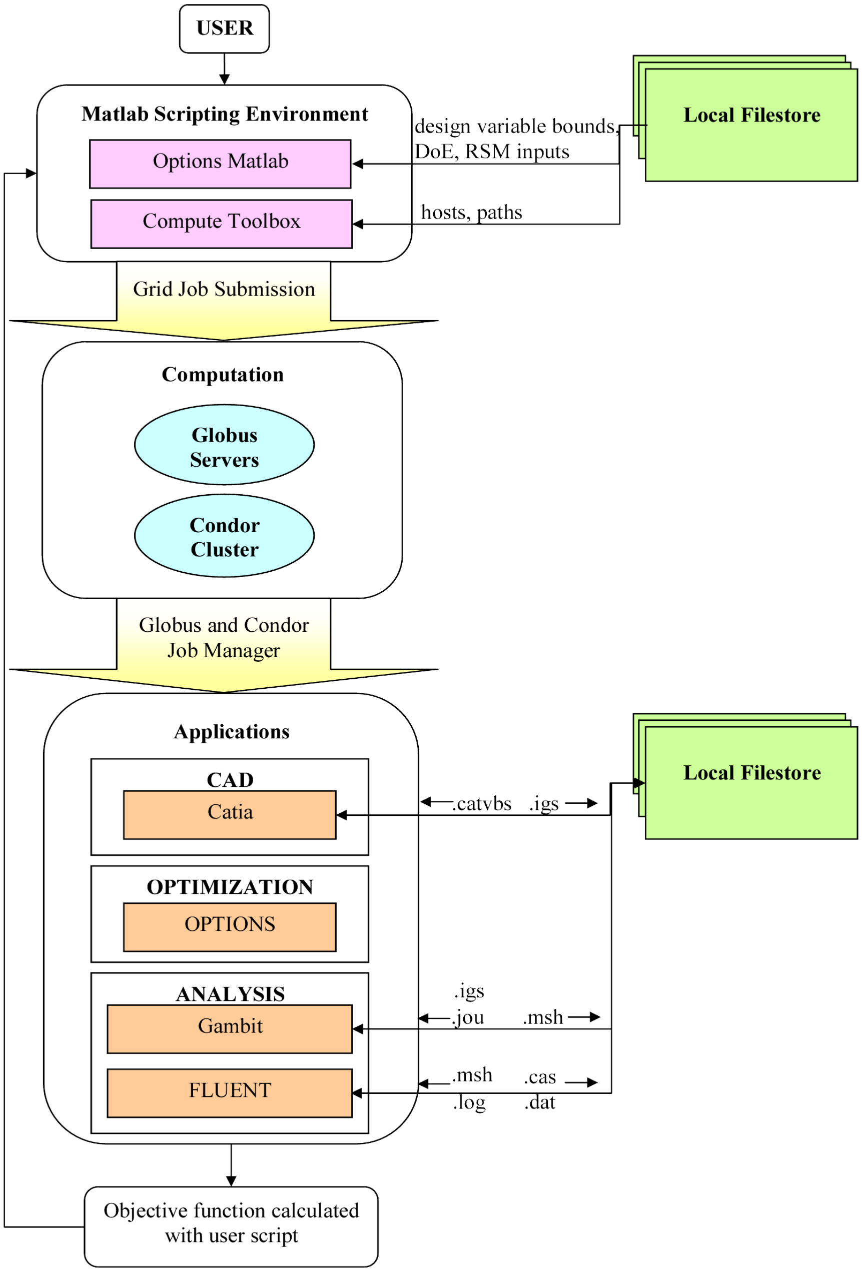


FIGURE 2.2: GEODISE system architecture

server (Globus, 2002) provides the middleware allowing the compilation of the remote resources. This server provides much of the functionality that the system requires including authentication, authorization, job submission, data transfer and resource monitoring. The applications box depicted in Figure 2.2 holds the array of higher level geometry, optimization, pre- and post-processing and CFD functions that the toolkit then calls with the appropriate input files from the user's filestore.

2.2 Geometric Parameterization

To begin the design search and optimization process shown in Figure 2.1, a parametric geometry must first be considered. Parameterization is here defined as the specification of a geometry by means of a finite number of parameters or design variables which are allowed to assume values in a given bounded range. The choice of method for geometric parameterization becomes crucial when used in conjunction with an optimizer. This choice is problematic, however, as in many situations one is faced with myriad feasible methods, sometimes with no real distinction between the advantages of one over another. The challenge is in selecting an appropriate set of design variables to allow a large amount of geometric variability or 'strong shape control' of the CAD created geometry, whilst retaining as compact a set as possible for the sake of efficient optimization. Strong shape control is important, allowing the optimizer to discover less intuitive designs with the potential to produce superior results. A proper choice of design variables usually requires a good understanding of the flow physics surrounding the geometry and the type of design variables likely to affect the objective function. In many internal fluid flow applications, however, the flow behaviour is not clearly understood. In particular, for cases where the designer is unable to predict likely changes in flow behaviour caused by certain changes in the geometry and whether these changes are likely to improve the design. In the absence of a clear understanding of the flow physics, strong shape control is essential in order to relate the design variables to the flow behaviour . A survey of the most common curve parameterization methods has been performed to determine the best choice of method to achieve strong shape control, retaining a set of only few design variables.

2.2.1 Polynomial Curves

In the field of numerical analysis, a spline is regarded as a special function defined in a piecewise manner by polynomials. In computer science, in particular the subfields of CAD and computer graphics, a spline is regarded as a piecewise parametric polynomial curve (Farin, 1990). Spline approaches to curve contouring have the advantages of providing a compact set of design variables and are naturally smooth. It is a popular representation, not only due to its inherent smoothness but also due to the simplicity of

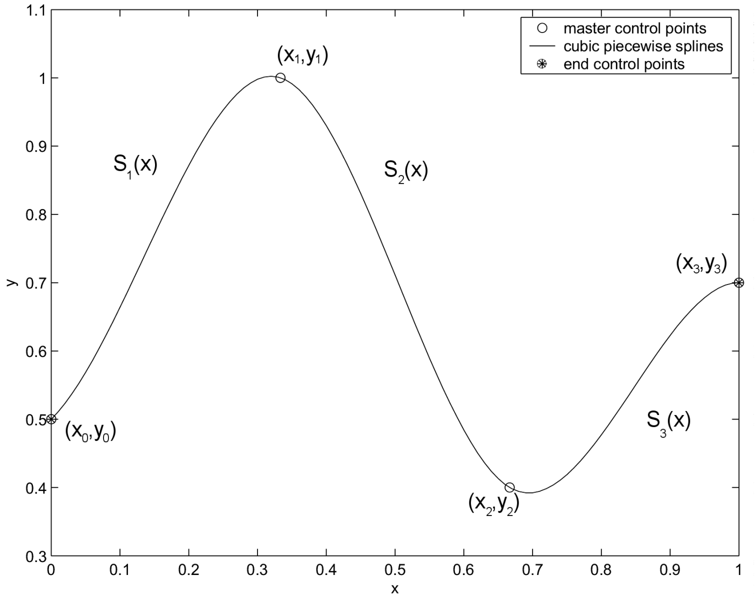


FIGURE 2.3: A piecewise cubic spline

its construction and evaluation and its capacity to create complex shapes. An example of a spline parameterization can be seen in Figure 2.3. Here, a piecewise spline comprising three polynomial (cubic) curves $S_1(x)$, $S_2(x)$ and $S_3(x)$ with end control points at (x_0,y_0) and (x_3,y_3) and two mid master points (x_1,y_1) and (x_2,y_2) is shown. At each of the mid master points, the piecewise splines join with first and second order continuity.

In its most general form, a univariate polynomial spline $S : [a, b] \rightarrow \mathbb{R}$ consists of polynomial pieces $S_i : [x_i, x_{i+1}] \rightarrow \mathbb{R}$, where

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b. \tag{2.1}$$

Hence,

$$\begin{aligned} S(x) &= S_1(x), \quad x_0 \leq x < x_1, \\ S(x) &= S_2(x), \quad x_1 \leq x < x_2, \\ &\vdots \\ S(x) &= S_n(x), \quad x_{n-1} \leq x < x_n. \end{aligned} \tag{2.2}$$

The vector $\mathbf{x} = (x_0, x_1, ..., x_n)$ is known as the knot vector and if the knots are equidistant it is a uniform spline. The smoothness is determined by the parametric continuity of one polynomial piece to the next. Continuity C^j implies that the two adjoining pieces S_i and S_{i+1} share common derivative values from order zero to order j.

The advantages of using this technique are the ease with which one can define parametric continuity, specify the curve tangency at the start and end of the curve and have interpolation through all the control points given. However, in cases where C^2 is imposed, one may see a tendency for the curve to oscillate. If this occurs, small changes could lead to dire ramifications on the geometry. By changing only one control point position, the entire curve is modified which would not be ideal if only a local change to the curve is desired. If a complex shape is modelled, the degree may be increased to allow for greater flexibility, however, this would incur a greater computational effort to evaluate the curve with a large number of points needed to describe the complex shape.

For the remainder of this thesis, the focus is mainly on the cubic spline, where each S_i is of degree 3, and the term "spline" in used in this restricted sense.

2.2.2 Bézier Curves

For simple geometries, Bézier curves (Bézier, 1970) are equally as effective as polynomial splines with, again, smooth and accurate properties represented concisely. Braibant and Fleury (1984) demonstrated that Bézier curves are well suited to geometric parameterization when used in optimization studies, while Farin (1990) describes some of the more useful properties of this particular technique.

In essence, a Bézier curve approximates a set of points, as opposed to the interpolation seen with polynomial splines, although the curve end points are interpolated. A Bézier curve can be calculated based on some $n + 1$ points to be interpolated. An example of a Bézier curve parameterization can be seen in Figure 2.4 with two master control points P_0 and P_3, through which the curve interpolates, and two knots P_1 and P_2 - four control points in total. Using this method one cannot prescribe the start and end tangency conditions. This is determined by the tangency calculated between the first two control points (P_0, P_1) and the last two control points (P_2, P_3) respectively. An additional advantage of this technique is that it is useful for curve collision detection, as the curve will lie within the convex closure of the control points. A Bézier curve is

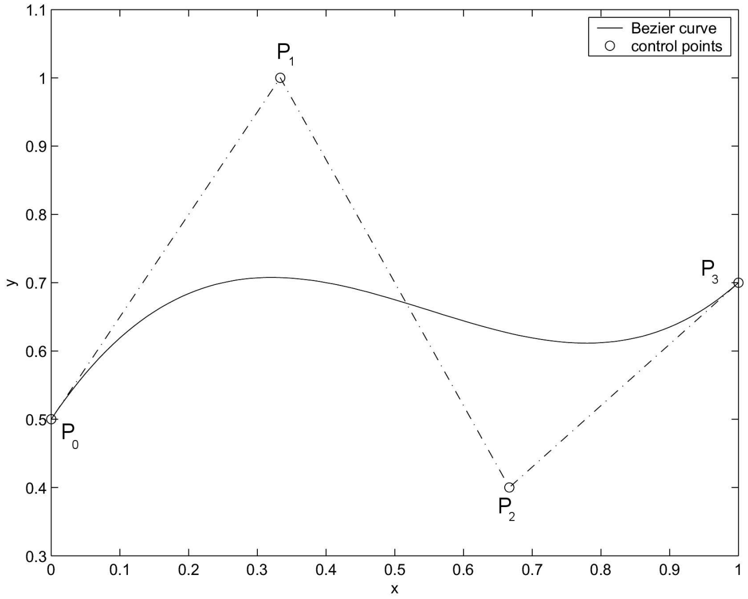


Figure 2.4: A Bézier curve

also invariant under affine transformations, i.e., rotation, scaling or translation of the control points result in the rotation, scaling or translation of the curve itself.

Mathematically, a Bézier curve can be defined as follows. Given a set of $n+1$ control points $P_0, \ldots, P_n$, the Bézier (or Bernstein-Bézier) curve is given by

$$S(x) = \sum_{i=0}^{n} P_i b_{i,n}(x) \tag{2.3}$$

where the Bernstein polynomial $b_{i,n}$ is defined as

$$b_{i,n} = \begin{pmatrix} n \\ i \end{pmatrix} x^i (1-x)^{n-i} \tag{2.4}$$

where $x \in [0,1]$ and i=0,...,n.

Similar to the polynomial splines, these curves do not offer strong shape control, nor do they offer an efficient way of describing complex shapes where a large number of control points are required or where a high order curve is needed. One way of improving this efficiency would be to divide the single Bézier curve into a number of lower order Bézier curves but this has its own disadvantage in that it becomes more difficult to ensure smoothness at the curve joins.

2.2.3 B-splines

B-splines refers to *basis splines.* Their useage as parametric curves was investigated by Schoenberg in the 1940s but did not become popular until the publications by de Boor and Cox in the 1970s (de Boor, 1972; Cox, 1972) where they discovered recurrence relations facilitating rapid evaluation of the basis functions. A generalized B-spline is defined as follows. Suppose that we have a knot vector $\mathbf{x}$ containing $m + 1$ knots $x_0 \leq x_1 \leq ... \leq x_m$, a B-spline is given by:

$$S(x) = \sum_{i=0}^{m} P_i \, \alpha_{i,n}(x) \qquad x \in [x_0, x_m], \tag{2.5}$$

where P_i are the control points and $\alpha_{i,n}(x)$ are the basis functions.

The m by n B-spline basis functions of degree n can be defined using the Cox-de Boor recursion formula,

$$\alpha_{j,0} = \begin{cases} 1 & \text{if } x_j \leq x < x_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{j,n}(x) = \frac{x - x_j}{x_{j+n} - x_j} \, \alpha_{j,n+1}(x) + \frac{x_{j+n+1} - x_j}{x_{j+n+1} - x_{j+1}} \, \alpha_{j+n,n-1}(x). \tag{2.6}$$

If the knots are equidistant then the B-spline is considered uniform. If $n = m$ then the B-spline degenerates into a Bézier curve. A B-spline has strong shape control and has all the advantages of the Bézier curves although it is stricter in the sense that a B-spline will lie within the union of the convex closures of all segments and also provides greater shape control as moving a control point does not affect the whole curve, unlike polynomial curves.

2.2.4 Hicks-Henne Bump Functions

Hicks and Henne (1978) developed a global shape function to efficiently modify aerofoil sections. These are analytical shape functions which allow for strong shape control and can be written in a general form (see Sóbester and Keane (2002)):

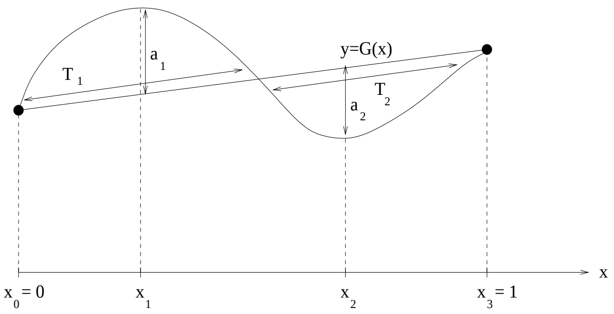


FIGURE 2.5: Two Hicks-Henne bump functions

$$f(x) = \sum_{i=1}^{n} a_i \left(\sin \left(\pi x^{\frac{-\log\ 2}{\log\ x_{p_i}}} \right) \right)^{T_i}, \tag{2.7}$$

for n bump functions and where $x, x_{p_i} \in [0, 1]$. These functions always guarantee smooth curves and can be described by three design variables per bump: the amplitude, a_i, the position of the bump peak, x_{p_i}, and the width of the bump, T_i.

Figure 2.5 shows how two bump functions can be added to the line $y = G(x)$ to create a smooth and continuous parameterized curve defined by

$$S(x) = G(x) + a_1 \left(\sin \left(\pi x^{\frac{-\log\ 2}{\log\ x_1}} \right) \right)^{T_1} + a_2 \left(\sin \left(\pi x^{\frac{-\log\ 2}{\log\ x_2}} \right) \right)^{T_2}. \tag{2.8}$$

2.3 Design of Experiments

Let us consider a k-dimensional design problem, i.e., a parametric model is employed which uses k design variables. The k-dimensional design space in which the problem is defined is commonly known as a hypercube. Response surface models (RSMs) (Myers, 1976; Box et al., 1978; Box and Draper, 1986; Khuri and Cornell, 1987; Myers and Montgomery, 1995) are built based on a set of initial points, the locations of which can be determined with an appropriate density according to a form of Design of Experiments (DoE) approach (Mead, 1988). This approach materialized from the need of the experimentalist to approximate the variation of the output or response over the whole hypercube in a systematic and efficient manner.

There are many methods for efficient space filling for which Montgomery (2000) provides an authoritative text, but the choice of method is largely dependent on the type of problem and the type of RSM selected. In a large proportion of methods, the sample points filling the space occur in a relatively even but not a constant manner, except for random space filling, where points could occur in batches with sparsely distributed points in other areas of the design space.

For problems where there is no previous or intuitive knowledge as to how the surface is likely to look, it is generally considered beneficial to use a technique which allows arbitrary designs to which more designs may be added should the construction of the response surface be poor. An LPτ method (Sobol, 1979), (Statnikov and Matusov, 1995) provides an attractive solution to this problem. This technique allows for the addition of new points into the design space without the need to reposition the existing points to retain the even distribution of points. This method provides a reasonably good coverage of the space although may have some limitations in coverage in comparison to a non-extensible approach such as the Latin-hypercube method (Keane and Nair, 2005). These DoE points supply the training data for the construction of a response surface. Typically, it is considered that for there to be a sufficient number of points in the initial data set to allow the initial response surface to be representative, approximately ten times the number of design variables is needed (Jones et al., 1998). For further discussions on this topic, refer to Sóbester et al. (2005).

2.4 Computational Fluid Dynamics

Research into fluid flow problems started with the work of Newton. Significant progress began in the mid 18th century with Leonhard Euler who, as legend has it, was invited by Frederick the Great to Potsdam in 1741, whereupon he was asked to engineer a water fountain. As a dedicated theorist, his route to success began with the need to understand the laws of fluid motion. From this Euler developed a set of equations for the motion of compressible, inviscid fluids in 1755. These equations are known today as the ‘Euler equations’ and are a set of partial differential equations (PDEs) expressing the conservation of mass, momentum and energy. In 1821, Claude-Louis Navier and George Stokes modified Euler’s equations to account for the forces between the molecules in an incompressible fluid, leading to the development of the equations of motion for viscous fluids, commonly known as the Navier-Stokes equations. In all cases studied in this thesis, incompressible flow is assumed.

Over the last half century, analysis codes approximating the solution of the Euler equations and the Navier-Stokes equations, or CFD software, have been developed. Today, these codes take a number of different forms. The least computationally expensive method, the panel method, is designed for irrotational flows.

The most computationally expensive method is a solution of the Navier-Stokes equations involving viscosity and thus boundary layer activity. The Navier-Stokes equations can be solved directly without any simplification via direct numerical simulation (DNS). However, this requires a mesh fine enough to capture the smallest levels of turbulence, which can be estimated by the turbulent microscale. Moreover, the higher the Reynolds number, the smaller the turbulent microscale and hence a finer mesh is required. At the Reynolds number of most practical engineering problems (such as those considered in this thesis), the capture of the turbulent microscale would require such a fine mesh that excessive computational resources would be needed. To overcome this, the Navier-Stokes equations can be simplified using a Reynolds averaging process. This yields a set of PDEs including Reynolds stress terms (which represent the unsteady, aperiodic motion in which transported quantities fluctuate in time and space over the main fluid flow) within the conservation of momentum equation. This method of finding a solution is known as a Reynolds-Averaged-Navier-Stokes (RANS) calculation and requires a turbulence model for closure, where the turbulence model provides a relation between the Reynolds stresses and the mean flowfield.

CFD software is complex in its working as the Navier-Stokes equations cannot be solved explicitly. What is needed is a scheme whereby a converged solution can be assumed to provide a good approximation to the exact solution of the original equations. This thesis uses the finite volume scheme.

The general idea for finite volume fluid flow simulations within a specified volume V is as follows:

- Mesh the volume – subdivide V into a finite number of adjoining control volumes or ‘cells’. The centroid of each cell is known as the node and it is here where the variable values are usually calculated.

- Discretization – integrate the PDEs over the control volumes to establish a set of algebraic equations directly stating the conservation laws as a balance of cell face fluxes and volume sources. In this thesis a finite-volume mesh is used and the implicit segregated equations solved.

- Interpolation – for the transport variable values ψ_f at the cell faces, which are needed to calculate the face fluxes, interpolation from the nodal values ψ is accomplished in this thesis using either a first or second order upwind scheme, i.e. ψ_f is derived from values in the cell upstream, or upwind, relative to the direction of the normal of the flow velocity vector.

For further detail on general CFD techniques, see Anderson (1995). Returning to Figure 2.1, having located design points in the hypercube determined through a DoE, these points are sampled by creating geometries defined by the DoE, meshed using an appropriate size mesh determined by mesh dependency studies, and then simulated by calling the CFD analysis code. In this thesis, the CFD code Fluent is used to perform steady RANS simulations to determine the objective function. The most appropriate turbulence model for the type of design problem is employed, more detail on which is given in Chapters 4, 6 and 7.

2.5 Response Surface Methodology

During the last 15 years, response surface modelling (also known as surrogate or metamodelling) has played an important role in the practice of global optimization. It is an area which is rapidly catching up with the sophistication of other commercial tools used in the design optimization process available to the industrial sector today, such as CAD and analysis software. Response surface models (RSMs) are created by the fitting of a curve or surface to an initial data set of sampled points within the design space, thereby inexpensively ‘mimicing’ the behaviour of the objective function and hence the solution of the problem. RSMs encompass a range of different curve fitting techniques including polynomials (Montgomery, 2000), radial basis functions (Broomhead and Loewe, 1988; Powell, 1987), Kriging (Matheron, 1963; Cressie, 1990) and support vector machines (Christianini and Shawe-Taylor, 1999).

Today, not only are we seeing continuous improvements of the polynomial and radial basis function approximation techniques and their usage inside an optimization process, there is also much research interest in the neural networking approach (Hajela and Berke, 1991; White et al., 1992), support vector machines and alternative kernel based methods (Christianini and Shawe-Taylor, 1999) used for pattern recognition within data sets. In very simple terms, neural networking involves a type of artificial intelligence which attaches connections between the sampled input data. It is the organization and weights of these connections that determine the output of an untried data point. These methods provide promising ways of approximating sample data but their useage is outside the scope of this thesis.

If only a DoE is implemented in order to locate an optimum solution to the design problem, an excessive number of calls to the CFD analysis code would have to be made. To evaluate the objective function through the solution of the Navier-Stokes equations, a large computational cost is incurred. Therefore, to improve efficiency and, moreover, the feasibility of utilizing optimization in industry, the objective function is approximated by means of a RSM. This method has the essential and desirable feature of requiring only a limited number of calls to the analysis code for the construction of the response surface, given the dimensional size of the problem. Alternative techniques to response surface methodology include direct genetic algorithm (GA) searches (Goldberg, 1989; Hajela, 1999) and multi-start Dynamic Hill Climbers (DHCs) (Yuret and de la Maza, 1993). These involve more exhaustive searching over the design space to reach an optimal value. This becomes impracticable for multi-dimensional problems such as those found in many industrial applications.

In the generalised optimization framework (Figure 2.1), an update point is found by building a response surface. Once the RSM is built, the accuracy of this surface approximating the objective function is determined by the surface fit, interpolating or regressing, and the number of sample points used in the DoE. This surface can be updated by placing further calls to the analysis software at appropriate points. The surface is evaluated and then searched in lieu of the real problem to find the next most likely best point. There are three different types of update strategy for the Kriging RSM approach to locate the next best update point: optimizing the predictor, minimizing the prediction error or maximizing the expected improvement. These are discussed in more detail in Chapter 3. The search of the surface can be performed either in serial or parallel using GAs or DHCs. Having located this new point, the analysis code is called to calculate the objective function at this point only. Once evaluated, these points are added to the set of sample points and the RSM can be rebuilt. Updating the response surface model in this way improves its accuracy and allows the update process to efficiently improve the design towards an optimal solution. It is important to note that no surrogate modelling scheme is guaranteed to find the true global optimum except when implementing a completely exhaustive search. The "Optimum design" featured in Figure 2.1 and the term "optimum design" discussed hereafter is used to define the best optimized design resulting from the optimization process and not the true global optimum of the objective function.

Alternative RSM approaches are discussed briefly in Chapter 3. The following chapter also provides a detailed mathematical description of Kriging and the process that is used to determine the most suitable location for the update point.

2.6 Exploration of Reduced Design Space

Following each update point, a decision is made as to whether time taken to build the response surface has exceeded a reasonable amount; see Figure 2.1. For high-dimensional design problems, once the number of update points reaches into the few hundreds, the time taken to build the response surface required to locate the next best candidate point to be analysed can become prohibitively expensive. For example, in high-dimensionality problems, a large initial DoE is performed but as the number of update points increases, it becomes impracticable to build and tune the RSM using Kriging as the surface construction time may exceed the time taken to analyse the update point itself.

At this point, a decision is also made as to whether the best design found is adequate with respect to the type of problem and its objective function. Should the objective function of the best design not be adequate, it is possible to further improve the design by performing a concentrated exploration of a reduced area of the design space. This process is similar to the theory of trust region searches (Alexandrov et al., 1998). Provided a reasonable number of update points have been calculated it is likely that the current best point is near a promising optimum value. The design space around this best point can be clipped and a concentrated exploration of this reduced design space performed. This involves performing a DoE within this reduced design space in order to improve the best design further. As this DoE can be run in parallel and no response surface construction is necessary, this provides an efficient method of converging upon an optimum design. Specific local search techniques are described in later chapters.

2.7 Summary

A description of a typical optimization process has been presented in this chapter. Invariably, this process is automated. The description of the automatic progression of procedures used in the work covered in this thesis is outlined.

Following the general structure of an optimization process, a parametric geometry must be constructed. A number of techniques which provide the building blocks for curve parameterization have been discussed. A DoE approach is then used to provide a set of geometries which fill the design space efficiently. Analysis is performed on these geometries providing the responses through which a response surface can be built. An update point is found using the RSM and its objective function is found. If the time to build the RSM is still reasonable, then the RSM is rebuilt including the recently analysed update point and searched for the next update point. Once the time taken to build the RSM becomes prohibitively expensive, the design may not have found an adequate objective function, especially for design problems of high-dimensionality. If the desired objective function has been satisfied then the optimum has been found. If not, then an exploration of a reduced area of the design space can be performed around this point to converge quickly to an optimal design. This description of a typical optimization strategy has not elaborated upon the details of response surface modelling, in particular Kriging, and how the most appropriate update point is located. Details of this can be found in the following chapter.

Chapter 3

Optimization using Response Surface Methodology

3.1 A Global Approximation

For the majority of engineering problems, the shape of the response surface is unknown. Many objective functions are deceptive and can lead to an optimization process based on a local approximation, which will only find a local maximum or minimum. In light of this, it is paramount that an approximation based optimization routine progresses towards the global optimum, or at least explores the design space reasonably widely.

RSMs can be one of two varieties: an interpolating model or a regression model. The difference is that the construction of the surface occurs *through* the set of points that have been sampled, in the case of interpolation, or *near*, in the case of regression. For functions whose objective function curve or surface is not known *a priori*, the decision as to whether the response surface should be allowed to feature regression is not obvious. For a completely regressing response surface, the addition of more points may not necessarily lead to a more accurate representation of the function, whereas the addition of more points in an interpolating surface involves more points of full accuracy. However, it is important to note that unless the function is smooth and continuous, the interpolating surface approximation may not represent the function very accurately in between the sampled points, since the requirement for interpolation can lead to the surface 'overshooting' or 'undershooting' the objective function values in such regions.

The definition of an RSM can differ widely. Montgomery (2000), for example, uses the term for polynomials, while this thesis uses a more general definition. Here, two sorts are considered: polynomial models and radial basis function (RBF) models.

Suppose a k-dimensional problem is sampled at n points $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where each $\mathbf{x}_i$ $(i = 1, \ldots, n)$ is a vector containing k design variables $x_{i1}, x_{i2}, \ldots, x_{ik}$. Each sample point has a corresponding objective function value y_i, collectively giving the vector $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$.

Polynomial models can be generalized by

$$\hat{y}(\mathbf{x}^*) = \sum_{l=1}^{n} a_l \alpha_l, \tag{3.1}$$

where $\mathbf{x}^*$ is the untried point, a_l are coefficients and α_l is a basis function where a set of all polynomials in $\mathbf{x}$ with degree d can be generated by a basis $B = \{\alpha_l\left(\mathbf{x}\right) | l = 1, \ldots, n\}$.

For example, if $d = 1$, a basis will contain a combination of the constant 1 and any first order term of x. If $d = 2$, a basis will contain a combination of the constant 1 and any second order term of x and so on.

These are the simplest of global RSMs but as the objective function landscape becomes more complex, higher order polynomials are required which increases the number of points needed to build the response surface. Hence, low order polynomials are generally considered for this type of approximation and because of this, there is a concern that it may lead to large inaccuracies in the model. Radial basis functions differ from polynomial models only in their choice of basis function. RBFs can be generalized by

$$\hat{y}\left(\mathbf{x}^*\right) = \sum_{t=1}^{n} a_t \Phi\left(\mathbf{x}^* - \mathbf{x}_t\right), \tag{3.2}$$

where the new untried point $\mathbf{x}^*$ is related to all of the sample points, a_t are coefficients, and Φ is commonly known as the kernel. There are a variety of different forms that Φ

can take to produce different basis function RSMs. A number of different examples can be defined as follows.

$$\begin{aligned} \Phi\left(\mathbf{x}^* - \mathbf{x}_t\right) &= \|\mathbf{x}^* - \mathbf{x}_t\| && \text{linear spline} \\ &= \|\mathbf{x}^* - \mathbf{x}_t\|^2 \log\left(\|\mathbf{x}^* - \mathbf{x}_t\|\right) && \text{thin-plate spline} \\ &= \|\mathbf{x}^* - \mathbf{x}_t\|^3 && \text{cubic spline} \\ &= \exp\left(-\frac{\|\mathbf{x}^* - \mathbf{x}_t\|}{\sigma^2}\right) && \text{Gaussian} \end{aligned} \tag{3.3}$$

All of the models mentioned above can be solved in such a way to form interpolating or regression models depending on the number of bases (Keane, 2004). Fourier analysis methods and least-squares methods are just two of the many techniques available to solve the models mentioned above.

Kriging (Matheron, 1963; Cressie, 1990) is a technique in the RBF category and is generalized as

$$\Phi = \exp\left(-\sum_{s=1}^{k} \theta_s |\mathbf{x}_s^* - \mathbf{x}_{ts}|^{p_s}\right), \tag{3.4}$$

where θ_s and p_s are unknown coefficients, commonly known as the hyperparameters. These hyperparameters provide statistical information on the quality of the surface being built and, once tuned, they can be used to rank the design variables in accordance to their relative dominance, see Jones (2001) and Keane (2003). The hyperparameters and their tuning will be discussed further in section 3.2. In most practical applications, the above Gaussian function is used and is based on that given by Sacks et al. (1989).

Low order polynomial approximations have been shown to provide a poor global approximation to some problems. However, Jones (2001) discusses the relative merits of the Kriging technique's robust capability in finding the true global optimum given potentially deceptive functions. Due to the versatility of Kriging in its capability of approximating complex objective function landscapes and its provision of additional statistical information of the surface, all cases studied here use a Kriging RSM for optimization purposes.

3.1.1 Kriging

Kriging is a technique which provides a statistical interpretation so that, in addition to the interpolator (or 'predictor'), a measure of the possible errors in the model is ascertained, which in turn may be used to position any further design points more prudently. The response surface can represent an approximation of the objective function, the error of the approximated objective function values, or the expected improvement in the objective function value that can hypothetically be attained over the design space. A balance between global exploration of the design space and local exploitation of promising regions of the design space is sought. Searching over the approximated objective function, the error of the approximation or the expected improvement of the approximation provide different balances between exploration and exploitation. These three methods are discussed in further detail in section 3.2.1 in order to determine the best method for finding a global optimum design.

Although Kriging is a versatile and robust method of approximating an objective function, it should be noted, however, that Kriging is not suited to all practical applications. Its efficiency is dependent largely upon the number of design variables defining the problem. In section 2.2 it has already been mentioned that strong shape control is important and in many cases, this is achieved by increasing the number of design variables. Therefore, one has to decide whether to trade-off the geometric complexity of the model for the computational time required to obtain an adequate solution. Typically, Kriging is computationally practicable for up to approximately 20 design variables (Keane, 2003).

3.2 The Update Process

The construction of the Kriging RSM can occur once the initial training data set, or DoE, has been evaluated by the analysis code. Figure 3.1 presents the procedure for finding an update point, showing the complete Kriging update process using the RSM (adapted from Figure 2.1). Both the DoE method and the complete update process applied in the optimization studies performed in this thesis are implemented using OPTIONS.

The DoE corresponds to a vector of n initial sample points. Let us consider the relation between these n sampled points. Each of the n sample points has a response y determined

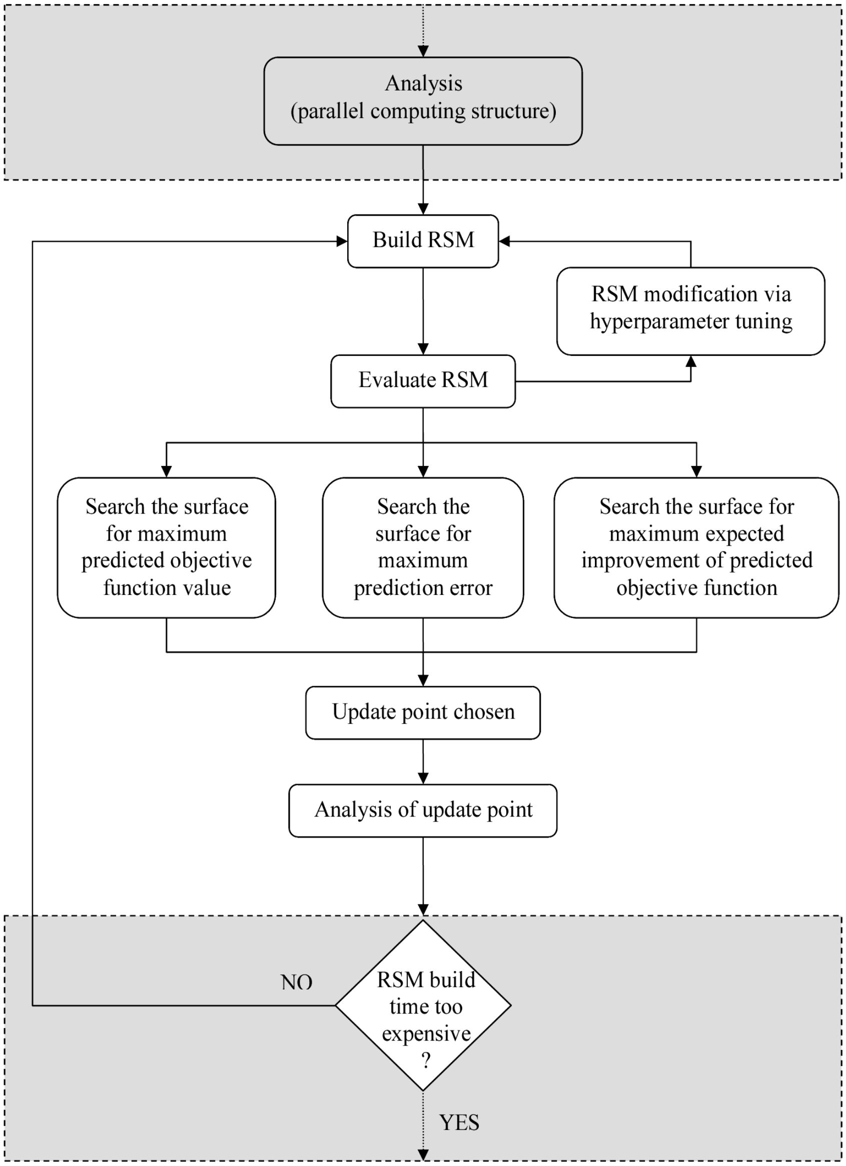


Figure 3.1: Kriging update process

by the objective function $y = f(\mathbf{x})$. Intuitively, assuming continuity of the objective function f, the difference between the responses $y(\mathbf{x}_i)$ and $y(\mathbf{x}_j)$ will be small if the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is small. In statistical parlance, $y(\mathbf{x}_i)$ and $y(\mathbf{x}_j)$ are highly correlated if $\|\mathbf{x}_i - \mathbf{x}_j\|$ is small, while if $\|\mathbf{x}_i - \mathbf{x}_j\|$ is large, the correlation will be small. This correlation is expressed as

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{s=1}^{k} 10^{\theta_s} |x_{is} - x_{js}|^{p_s}\right) + 10^{\zeta} \delta_{ij} \tag{3.5}$$

satisfying $R = 1 + 10^{\zeta}$ if $\mathbf{x}_i = \mathbf{x}_j$ and $\mathbf{R} \to 0 + 10^{\zeta}$ as $\|\mathbf{x}_i - \mathbf{x}_j\| \to \infty$. The hyperparameters θ_s and p_s represent the variable dominance and response surface smoothness respectively and ζ is a third hyperparameter measuring the magnitude of regression allowed. The regression term is included in all calculations performed in this thesis due to the uncertainty of the smoothness and continuity of the objective function, whether this is due to the nature of the problem or due to the inaccuracies encountered with the discretization error of CFD solutions.

The correlation matrix is a square $n \times n$ matrix $\mathbf{R}$ in the form

$$\mathbf{R} = \begin{pmatrix} 1+10^{\zeta} & R(\mathbf{x}_1,\mathbf{x}_2) & \cdots & R(\mathbf{x}_1,\mathbf{x}_n) \\ R(\mathbf{x}_2,\mathbf{x}_1) & 1+10^{\zeta} & & R(\mathbf{x}_2,\mathbf{x}_n) \\ \vdots & & \ddots & \vdots \\ R(\mathbf{x}_n,\mathbf{x}_1) & \cdots & \cdots & 1+10^{\zeta} \end{pmatrix}. \tag{3.6}$$

For any random variable $\mathbf{Y} = \{Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n)\}^T$, $\mathbf{Y}$ has a mean of $\mathbf{1}\mu$ and a covariance $\mathrm{Cov}(\mathbf{Y}) = \sigma^2\mathbf{R}$. $\mathbf{Y}$ now depends on the parameters μ, σ^2, θ_s, ζ and p_s $(s = 1, \ldots, k)$. To estimate the values of μ, σ^2, θ_s, ζ and p_s, a search is performed for the values of these parameters which maximize the likelihood of the responses.

Supposing we have a set of responses

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \tag{3.7}$$

then the likelihood L, with unknown μ and σ^2, can be defined as

$$L = \frac{1}{(\sigma^2)^{\frac{n}{2}}(2\pi)^{\frac{n}{2}} \det \mathbf{R}^{\frac{1}{2}}} \exp\left(-\frac{(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right). \tag{3.8}$$

In practice, however, it is more convenient to choose the parameters μ and σ^2 to maximize the log-likelihood function

$$\mathrm{LLF} = -\frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log(|\mathbf{R}|) - \frac{(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2} + \text{constant terms.} \tag{3.9}$$

The maximum likelihood estimators (MLEs) $\hat{\mu}$ and $\hat{\sigma}^2$ (circumflex denotes a MLE) of this are then defined by

$$\hat{\mu} = \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}, \tag{3.10}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y}-\mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu})}{n}. \tag{3.11}$$

By substituting the values for $\hat{\mu}$ and $\hat{\sigma}^2$ into equation 3.9 the concentrated log-likelihood function is obtained

$$\mathrm{CLLF} = -\frac{n}{2}\log(\hat{\sigma}^2) - \frac{1}{2}\log(|\mathbf{R}|) + \text{constant terms.} \tag{3.12}$$

This concentrated log-likelihood function is maximized to find the MLEs for ζ, θ_s and p_s, $s = 1, \ldots, k$. This is known as 'tuning' the hyperparameters (see Figure 3.1) and by doing this the accuracy of the RSM between the sample points is improved. Additionally, the relative dominance of the design variables can be readily assessed using the hyperparameters, $\log\theta_s > 0$, $0 < p \leq 2$. For example, a large $\log\theta_s$ value indicates a function where the objective value can change significantly over a small distance. p determines the smoothness of the function; the smoother the function, the closer p is to the value 2. For an interpolating model $\zeta = -\infty$, or in practice -6 or less to prevent ill-conditioning of the $\mathbf{R}$ matrix.

Note, however, that performing this hyperparameter tuning for each update can be computationally expensive and so in this thesis the hyperparameters are tuned for every 10 update points, under the assumption that the character of the surface will remain largely unchanged at intermediate updates.

3.2.1 Searching the RSM

When exploiting the RSM, a search method is chosen to find the best candidate point on the model that either

1. optimizes the predicted objective function value,
2. maximizes the prediction error of the predicted objective function value, or
3. maximizes the expected improvement one could achieve over the predicted objective function values.

Using the values of $\hat{\theta}_s$, $\hat{p}_s$ $(s = 1, \ldots, k)$, $\hat{\zeta}$, $\hat{\mu}$ and $\hat{\sigma}^2$ found from building the RSM, evaluating the RSM, and possibly tuning the hyperparameters, the following sections 3.2.1.1, 3.2.1.2 and 3.2.1.3 define the searches mathematically and provide an insight into the best methods for locating a global optimum design.

Search techniques are many and varied. Genetic algorithms and DHCs have already been mentioned. Others include simulated annealing and other evolutionary strategies. Keane and Nair (2005) offer detailed insight into techniques available to exploit RSMs. In all optimization studies in this thesis, a DHC method is used.

3.2.1.1 The Predictor

To arrive at a prediction for the objective function at some untried point $\mathbf{x}^*$, an objective function value is estimated and augmented to the initial n-dimensional data set provided by the DoE. It is ascertained, in the following steps, how consistent the estimated value of the objective function at the untried point is with the already observed pattern of variation between data points $\mathbf{x}$ and their corresponding responses $\mathbf{y}$. Estimates of the mean and variance of $\mathbf{y}$ are chosen to maximize the log-likelihood function

$$\text{LLF} = -\frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log(|\mathbf{R}|) - \frac{(\mathbf{y} - \mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2} + \text{constant terms.} \quad (3.13)$$

Suppose $\mathbf{x}^*$ is an unsampled point and the prediction of the objective function at this point is $y^* = y(\mathbf{x}^*)$. An augmented set of responses is now $\tilde{\mathbf{y}} = (\mathbf{y}^T \quad y_{n+1}(\mathbf{x}^*))^T$ with

$$\mathbf{r} = \begin{pmatrix} R(\mathbf{x}^*, \mathbf{x}_1) \\ \vdots \\ R(\mathbf{x}^*, \mathbf{x}_n) \end{pmatrix}$$

and

$$\tilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix}. \tag{3.14}$$

With the augemented sets of data, and referring to equation 3.13, it is only the third term that depends on $\tilde{\mathbf{y}}$, and so the quantity to be maximized is the augmented log-likelihood function

$$\text{ALLF} = -\frac{(\tilde{\mathbf{y}} - \mathbf{1}\hat{\mu})^T \tilde{\mathbf{R}}^{-1} (\tilde{\mathbf{y}} - \mathbf{1}\hat{\mu})}{2\sigma^2} + \text{terms independent of } y^*$$

$$= \frac{-\begin{pmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ y^* - \hat{\mu} \end{pmatrix}^T \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ y^* - \hat{\mu} \end{pmatrix}}{2\hat{\sigma^2}} + \text{terms independent of } y^*.$$

$$= \left[\frac{-1}{2\hat{\sigma^2}(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})}\right](y^* - \hat{\mu})^2 + \left[\frac{\mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})}\right](y^* - \hat{\mu}) + \text{terms independent of } y^* \,, \tag{3.15}$$

for which the maximum of this is the Kriging predictor:

$$\hat{y}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}). \tag{3.16}$$

For a full derivation, please refer to Appendix A.1.

It is $\hat{y}$ that is searched to find the value of $\mathbf{x}$ which optimizes the predicted objective function value, thereby providing the location of the next update point. As the DoE is augmented with values of the *predicted* objective function, this method tends towards an exploitation of the RSM and so for poor approximations of the objective function, this method could readily become trapped in a basin around a local optimum value. Jones (2001) shows that the augmented log-likelihood function may depend on the predicted value y^* in two ways: first, if the curvature of the augmented log-likelihood function is high, i.e., its value changes significantly as y^* varies, the confidence in accuracy of the Kriging predictor is high; and second, if the curvature of the augmented log-likelihood function is low, i.e., different values of y^* perform almost as well, the confidence in accuracy of the Kriging predictor is low. Hence, the potential error of the predictor is inversely related to the curvature of the augmented log-likelihood function

$$\text{curvature} = \frac{1}{\hat{\sigma}^2(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})}. \tag{3.17}$$

The measure of potential error is calculated in the following section.

3.2.1.2 Prediction Error

The mean square error, MSE, can be defined as

$$\text{MSE} = \hat{\sigma}^2 \left[1 - \begin{pmatrix} 1 & \mathbf{r}_x^T \end{pmatrix} \begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix} \right] = \hat{\sigma}^2 \left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} \right], \tag{3.18}$$

where the first two terms represent the reciprocal of equation 3.17. For a full derivation, please refer to Appendix A.2.

It is important to note that the the mean square error is conditional given that the correlation parameters are known. In reality, however, they are simply maximum likelihood estimators from equation 3.12. This means that it is likely that the value given by equation 3.18 underestimates the true uncertainty, especially when optimizing small dimensional problems. Bootstrapping methods are one technique of correcting the estimation error (den Hortag et al., 2004) but further work relating to this is outside the scope of this thesis. It is these values for the mean square error which are searched when maximizing the error of the predicted objective function values to locate the next best update point.

An advantage of this technique over updating using a search over the predictor to the objective function is that the error is equal to zero at the sample points. Additionally, the surface is explored with a much more global search. It must be noted, however, that becoming trapped in a basin of a local optimum is still possible and the surface will be exploited in these basins. A more robust method which is guaranteed to converge to an optimal design is the method of finding the value of the improvement expected at an untried point (Gutmann, 2001; Locatelli, 1997).

3.2.1.3 Expected Improvement

Expected improvement is the improvement expected to be achieved when sampling at an untried point, given that there is an error in our prediction at that point. Let a random variable $Y \sim N(\hat{y}(\mathbf{x}), s^2)$, where $\hat{y}$ is the Kriging predictor defined in equation 3.16 and s^2 is the mean square error. For a maximization problem, let f_{max} be the current best objective function value, then an improvement I will be achieved if $I = Y(\mathbf{x}) - f_{max} > 0$. The expectation of I can be defined as

$$E(I) = \int_{I=0}^{I=\infty} I \left\{ \frac{1}{\sqrt{2\pi} s(\mathbf{x})} \exp \left[-\frac{(I + f_{max} - \hat{y}(\mathbf{x}))^2}{2 s(\mathbf{x})^2} \right] \right\} \mathrm{d}I, \tag{3.19}$$

which can be integrated to give

$$E(I) = s \left(u \operatorname{cdf}(u) + \operatorname{pdf}(u) \right), \tag{3.20}$$

where $\operatorname{cdf}(u)$ is the normal cumulative distribution function and $\operatorname{pdf}(u)$ the normal probability density function and where $u = (\hat{y} - f_{max})/s$.

This is the function which is searched and maximized to find the next best update point. This provides a good balance between exploration and exploitation of the surface

(Sóbester et al., 2005) and is used for all optimization studies performed in this thesis. For the full derivation, again please see Appendix A.3.

3.2.1.4 Regression Term

Should a regression term be added into the correlation matrix such that

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{s=1}^{k} \theta_s |x_{is} - x_{js}|^{p_s}\right) + 10^{\zeta}\delta_{i,j}, \tag{3.21}$$

satisfying $R = 1 + 10^{\zeta}$ if $\mathbf{x}_i = \mathbf{x}_j$ and where $\delta_{i,j}$ is the Kronecker delta, the correlation matrix is of the form in equation 3.6. The augmented correlation matrix is in this case defined as

$$\tilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 + 10^{\zeta} \end{pmatrix}. \tag{3.22}$$

This would not affect the equations for $\hat{\mu}$, $\hat{\sigma}^2$ or the Kriging predictor $\hat{y}$. However, there is a change in the mean square error and so equation 3.18 can be written as

$$\mathrm{MSE} = \hat{\sigma}^2 \left[1 + 10^{\zeta} - \begin{pmatrix} 1 & \mathbf{r}_x^T \end{pmatrix} \begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix} \right]. \tag{3.23}$$

And by the same method given earlier, the mean square error including regression becomes

$$\mathrm{MSE} = \hat{\sigma}^2 \left[1 + 10^{\zeta} - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{1})^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right]. \tag{3.24}$$

Note: This regression term in the form 10^{ζ} is present in all calculations performed in this thesis.

Using equation 3.24 to find s leads to problems with the convergence of $E(I)$. To solve this, OPTIONS eliminates the ζ term and uses the value for the mean square error given in equation 3.18 and uses

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{R} + 10^{\zeta}\delta_{ij}) \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{n}. \tag{3.25}$$

This results in error estimates which assume the data follows a smooth trend ignoring the error due to the noise in the data. For further details see Forrester et al. (2006).

3.3 Summary

This chapter has introduced response surface methodology with a variety of approximation methods. Kriging has been chosen as the approximation method for use in global optimization due to its versatility and capability of representing complex objective functions. A number of steps have been outlined to determine the Kriging predictor and to find the next update point by searching the surface of the predicted objective function value, the prediction error or the expected improvement of the objective function value, including the correction required when regression of the response surface is allowed. Expected improvement is chosen as the most appropriate type of update strategy due to its robustness and capacity to guarantee eventually to find the location of the global optimum assuming adequate time and resources.

Chapter 4

An Automated Single Stage Shape Optimization Case Study

4.1 The Global Optimization of a Two-Dimensional Airbox

The process described in chapters 2 and 3 is applied in this chapter to the design of a two-dimensional airbox. To obtain an understanding of how the wall geometry and centreline bend affect the efficiency of expansion and the efficient turning of the flow, the two functions of the airbox are initially considered separately: that of expanding the flow through a straight diffuser and that of turning the flow through 90° with no expansion.

After a definition of the design objective and a discussion of the CFD model employed, the initial focus is on the straight diffuser in section 4.4. Research has been carried out on the optimal shape design of two-dimensional diffusers in turbulent flow using alternative methods to provide an accurate prediction of the flow separation by Zhang et al. (1995) and Lim and Choi (2004). These cases began from a widely studied optimal diffuser design rather than from scratch, and applied a simple technique to obtain strong shape control of the diffuser wall. Here, a variety of different geometry parameterization techniques are assessed with the aim of determining the most effective method of obtaining a geometry possessing a high degree of shape control using only a small number of design variables. A similar test is performed for a constant width elbow turning through 90° in section 4.5. The best techniques are then fused together for the final parameterization of a two-dimensional airbox in section 4.6.

4.2 Design Objective

The design problem in all cases studied in this chapter is the maximization of pressure recovery based on the internal flow through the diffuser. Steady, incompressible flow is assumed and so the pressure recovery coefficient, C_p may be defined by

$$C_p = \frac{p_o - p_u}{q_u}, \tag{4.1}$$

where p_o is the mass-averaged (or density-weighted) static pressure at the inlet, p_u is the mass-averaged static pressure at the filter and where

$$q_u = \frac{1}{2}\rho U^2 \tag{4.2}$$

denotes the dynamic pressure with ρ the fluid density and U the velocity at the inlet.

4.3 CFD Analysis and Optimization Strategy

All geometries are constructed using the CAD engine CATIA and imported into a meshing tool. Both meshing and flow simulations are executed using the commercial CFD package Gambit$^{\mathrm{TM}}$(Fluent$^{\mathrm{TM}}$, 2003a) and Fluent. To ensure that solutions yield sufficient accuracy within Fluent, a mesh dependency study has been performed on a straight walled diffuser prior to the optimization studies. An inlet mass flow rate of 10.8kgs^{-1} (Reynolds number, Re $= 6 \times 10^6$) is fixed, determined for a car travelling at approximately 70ms^{-1} or approximately 150mph, and a paved quadrilateral/triangle structure is used for the mesh. Structured quadrilateral cells of a fixed start size, growth rate and depth were grown from the wall to capture the boundary layer. A 'standard' wall function (the default setting in Fluent (Fluent$^{\mathrm{TM}}$, 2003b)) was used which is based upon the proposal of Launder and Spalding (1974). With this wall function, the logarithmic law for mean velocity is know to be valid for $y^+ > 30$, where y^+ is the wall-normal coordinate normalised by the local viscous length scale ν/u_τ. For this study $y^+ \approx 50$. Various mesh sizes were tested and solved with the same CFD model with a standard k-ϵ turbulence model at each attempt. Starting from a cell count of approximately 2000 cells, the mesh was progressively refined testing approximately 3, 6, 12, 24, 39, 75 and 120 thousand cell meshes. The profile of velocity U was set as uniform across the inlet and the dependent variable of turbulent dissipation rate was calculated from a turbulent intensity value of 1%, and the dependent variable of turbulent kinetic energy calculated from a hydraulic diameter of 0.2m. Using these values, the static pressure recovery value, C_p, was calculated for each of the different mesh sizes and the results shown in Figure 4.1. The percentage difference in C_p value between the meshes comprising of approximately 1500 cells and the finest mesh was 437%, whereas the percentage difference in C_p value between the meshes comprising of approximately 39000 cells and the finest mesh was 1.7%; this is considered to be a sufficiently accurate result. Hence a mesh with approximately 39000 cells was chosen for the optimization studies presented in this chapter. It is difficult to control the exact cell count when using an automated process to mesh each design discovered by the optimizer. However, the meshes used in this initial dependency study were created with a size function capability within the Gambit meshing tool which allows the same growth rate from the boundary layer to be developed regardless of the shape of the diffuser. Using this functionality, finer meshes are produced by decreasing the distance between the mesh nodes on the diffuser walls while maintaining a consistent boundary layer depth across meshes. The fixed boundary layer depth is such that it will capture all of the boundary layer as it develops over the streamwise distance x. The fixed boundary layer depth, fixed distance between wall nodes and a fixed cell growth rate mean that the meshes produced during the optimization processes later in this chapter will only vary by a small amount in terms of total cell count given varying wall shapes.

The CFD analysis for all the studies carried out in this chapter involves solving the incompressible two-dimensional steady-state Reynolds-averaged Navier-Stokes equations. The k-ϵ turbulence model (Launder and Spalding, 1974) is used for the straight diffuser study which is in keeping with the references studied, and the Spalart-Allmaras turbulence model (Spalart and Allmaras, 1992) is applied in the constant width turning

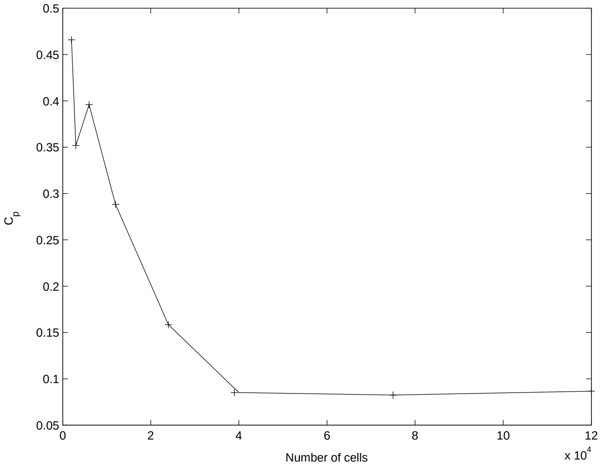


FIGURE 4.1: Graph illustrating the dependency of C_p value with mesh density for a 2D straight walled diffuser

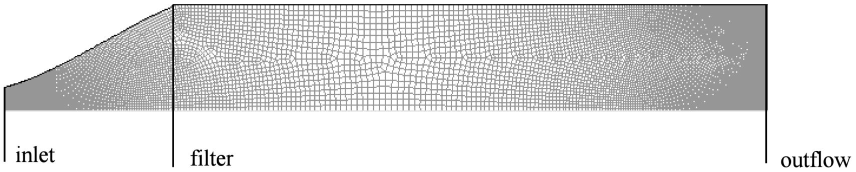


FIGURE 4.2: An example of a ∼39000 cell mesh with inlet, filter and outflow positions

elbow and the final study in section 4.6. The change to using a Spalart-Allmaras model is due to the fact that this model is more economic than the standard k-ϵ model and more accurate for wall-bounded flows and flows with mild separation and recirculation (Fluent$^{\mathrm{TM}}$, 2003b). The boundary conditions for studies conducted in sections 4.4 and 4.5 comprise a fixed mass flow rate of 10.8kgs^{-1} with a uniform profile at the inlet and a pressure outflow positioned at the exit of the duct. The exit of the duct is situated downstream of the engine filter. The position of the engine filter is chosen to be at the end of the diffuser expansion in the straight diffuser study, and at the end of the bend in the elbow study. This is to ensure that any separation arising within the diffuser or elbow does not pass through the outflow boundary. Mass-averaged static pressure values are taken at the inlet and at the position of the filter. Figure 4.2 illustrates a mesh of approximately 39000 cells and depicts the positions of the inlet, filter and outflow for the straight diffuser.

For the study carried out in section 4.6, the large expansion required over such a short distance coupled with the 90° bend means that a pressure outflow boundary condition at the filter is insufficient due to the unstabilized flow at this point caused by separation. The diffuser could be extended downstream of the filter creating a long constant width outflow duct as carried out in sections 4.4 and 4.5, but this is unrealistic in terms of the nature of the airbox setup within an F1 race car. To ensure that an accurate converged solution is obtained, the airbox model is extended to include the engine filter, represented in Fluent by a one-dimensional porous jump, and the trumpet tray, an area of fixed width situated between the filter and the engine trumpets which are positioned over the cylinders. The area represented by the engine trumpets is classified as a velocity inlet boundary condition.

A breathing engine sucks the air out of the airbox through the trumpets and so the velocity inlet condition at the trumpets is given a negative velocity value. It is assumed that the 4-stroke engine is at wide open throttle running at 18000 revolutions per minute (rpm), i.e., 9000 intake strokes per minute or 150 intake strokes of the pistons per second. Assuming 100% volumetric efficiency, the 3-litre engine requires 0.003 m^3 of air and hence the engine requires a volume flow rate $\dot{V} = 0.45\text{m}^3\text{s}^{-1}$. This flow rate is also equal to the product of the total cylinder area and the average piston velocity required by the engine. From this the piston velocity can be calculated given the total cylinder area.

For the two-dimensional model to be representative of the real 3D case, the total trumpet length is calculated by matching the 3D total cylinder area to filter area ratio. However, within the 3D airbox there is a second expansion ratio of total cylinder area to total inlet area to consider. For this 2D case, both cannot be considered. Thus, the length of the diffuser inlet and length of the filter are fixed. The first expansion ratio of cylinder area to filter area is considered more appropriate as the complete shape represents the centre plane of a 3D airbox and the geometry parameterization techniques for defining the walls can potentially be carried over into an airbox design strategy in three dimensions. Since the 2D diffuser is effectively doing half the work needed to expand the flow through a 3D airbox for the required breathing engine velocity, the velocity of the flow being sucked out of the 2D trumpets needs to be increased to represent sensible inlet speeds akin to those seen in the 3D case. The velocity through the engine trumpets is such that $\text{Re} = 2 \times 10^4$ giving a velocity of 58ms^{-1}. A pressure inlet is imposed at the diffuser inlet. The porous jump values used here are given by the thickness and permeability of the engine filters typically used by F1 teams and have the following values: face permeability = $1.3 \times 10^{-8}\text{m}^2$, filter thickness = 15mm and inertial resistance factor = 1000m^{-1}.

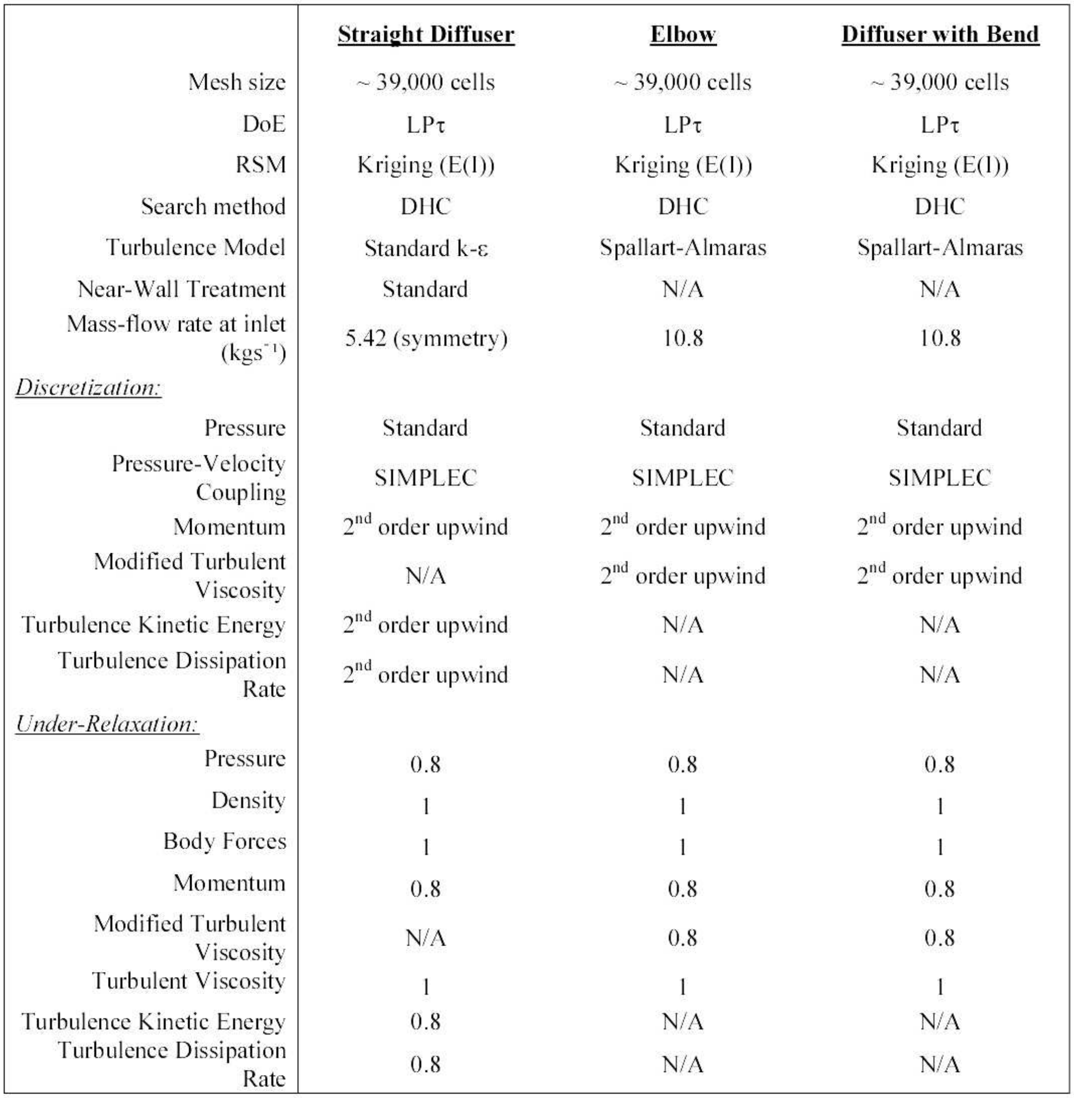
Table 4.1 provides the boundary conditions and turbulence model setup values for each of the three two-dimensional intake cases: the straight diffuser, the elbow, and the diffuser turning through 90°. The under-relaxation values shown in Table 4.1 are used to control the change of calculated nodal value during each CFD iteration. This occurs due to the non-linearity of the equation set solved by Fluent. So, for a variable ψ, the new variable value depends on the old value through the relation $\psi = \psi_{old} + q\Delta\psi$. The higher the under-relaxation values (q), the faster the convergence rate. Each simulation requires 1000 iterations to converge satisfactorily.

4.4 Straight Diffuser

In this section, three different parameterization techniques are considered for diffusers with straight centrelines. These diffusers have a total expansion ratio (diffuser exit length A_e : diffuser inlet length A_{in}) of $E = 4.5$ and aspect ratio $N/A_{in} = 1.6$ where N is the diffuser axial length.

The flow regimes classified by Fox and Kline (1962) established a correlation between diffuser performance and flow separation occurrence for varying expansion ratio and aspect ratio values at Reynolds number $\text{Re} = 1.6 \times 10^5$. Ideally, to obtain the best possible diffuser performance, the diffuser is designed in such a way so that no areas of separation are induced. This is because the separation regions obstruct the effective passage of flow and therefore the diffuser experiences increased losses. However, if a straight wall is imposed for the diffuser, the expansion and aspect ratios are such that, referring to Reneau et al. (1967)'s flow regimes, flow separation would be expected. Hence, the purpose of this parameterization exercise within the design optimization strategy is to allow contouring of the wall so that separation is reduced or even eliminated.

Wall contouring was first tested for two-dimensional straight diffusers using experimental means by Carlson et al. (1967). It was found that for $E \in [1.5, 4.5]$ and $N/A_{in} \in [3, 18]$, bell-shaped wall geometries returned the highest C_p value within their optimization study. Madsen et al. (1999) studied this problem using CFD and modern optimizer



	Straight Diffuser	**Elbow**	**Diffuser with Bend**
Mesh size	~ 39,000 cells	~ 39,000 cells	~ 39,000 cells
DoE	LPτ	LPτ	LPτ
RSM	Kriging (E(I))	Kriging (E(I))	Kriging (E(I))
Search method	DHC	DHC	DHC
Turbulence Model	Standard k-ε	Spallart-Almaras	Spallart-Almaras
Near-Wall Treatment	Standard	N/A	N/A
Mass-flow rate at inlet (kgs^{-1})	5.42 (symmetry)	10.8	10.8
Discretization:			
Pressure	Standard	Standard	Standard
Pressure-Velocity Coupling	SIMPLEC	SIMPLEC	SIMPLEC
Momentum	2nd order upwind	2nd order upwind	2nd order upwind
Modified Turbulent Viscosity	N/A	2nd order upwind	2nd order upwind
Turbulence Kinetic Energy	2nd order upwind	N/A	N/A
Turbulence Dissipation Rate	2nd order upwind	N/A	N/A
Under-Relaxation:			
Pressure	0.8	0.8	0.8
Density	1	1	1
Body Forces	1	1	1
Momentum	0.8	0.8	0.8
Modified Turbulent Viscosity	N/A	0.8	0.8
Turbulent Viscosity	1	1	1
Turbulence Kinetic Energy	0.8	N/A	N/A
Turbulence Dissipation Rate	0.8	N/A	N/A

TABLE 4.1: CFD and optimization setup values for Fluent (Fluent^{TM}, 2003b)

codes, maximizing the static pressure rise and adopting a B-spline parameterization with five master points along the wall. They compared computed results through the use of CFD with those obtained experimentally by Reneau (1967) by imposing a re-creation of the same inlet conditions for a straight walled diffuser. It was shown that for the straight walled benchmark diffuser, the CFD code consistently overestimated the pressure recovery values found via experiment. This was explained by the use of a stationary CFD model where flow separation occurs. However, it was confirmed that with the diffuser geometries falling within the bounds $E \in [1.5, 3]$ with a constant aspect ratio $N/A_{in} = 3$, Madsen et al's parameterization technique coupled with their

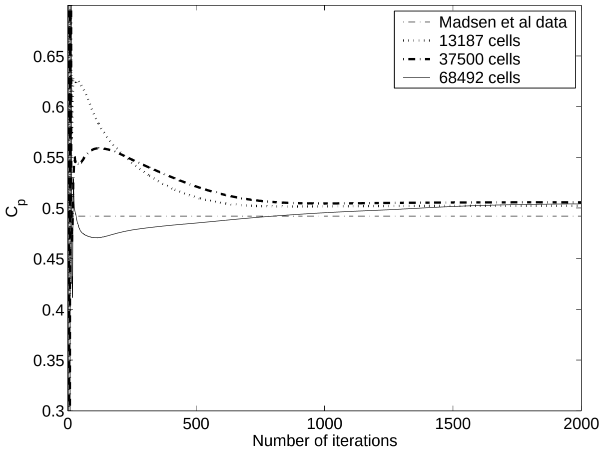


Figure 4.3: Validation between CFD model and Madsen et al. (1999)

optimization strategy produced bell-shaped diffusers which were found to return the highest static pressure recovery (C_p values), the same conclusion reached by Carlson et al.

Validation of the CFD analyses discussed here is depicted in Figure 4.3 by comparison with the results in Madsen et al for a straight walled diffuser with $E = 1.5$ and $N/A_{in} = 3$. As shown, the pressure recovery values converged to within 2% of those in Madsen et al. (1999). Madsen et al's diffuser did not contain any separation and nor did the model tested here. This was validated by examining the wall shear stress values which remained positive along the wall throughout the diffuser.

4.4.1 Geometry Parameterization

Parameterization One – Polynomial Splines

Here the model comprises four separate piecewise polynomial splines, referred to hereafter simply as splines, that pass through five master points along the wall. The advantage of using piecewise splines is the capability of governing the C^1 and C^2 continuity at each join. This observation means that more local control of the wall can be obtained by specifying the appropriate continuity at each master point. By demanding the continuity of the second order derivative at each master point, the rate of change of gradient is constant at these spline joins preventing an overshoot which can create a “rippling” effect along the curve – a situation that should be avoided.

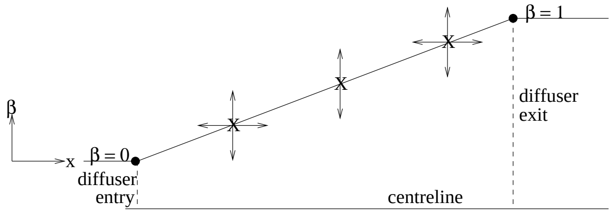


FIGURE 4.4: Spline parameterization of straight diffuser

The duct is defined to be symmetrical about its centerline with a cross-section area given by

$$A\left(x\right) = A_e + \left(A_o - A_e\right)\beta\left(x\right) \tag{4.3}$$

where $0 \leq \beta\left(x\right) \leq 1$. The parameter β is varied along the duct by treating it as a function of the x coordinate of the centerline at that particular cross-section. To avoid the problem of "rippling" through these points, the first and last splines are cubic and the middle two are quadratic curves. This approach means that the problem is not restricted by having too many constraints over the geometry to maintain continuity at the joints between the curves. Ghate et al. (2004) followed this approach to parameterize a duct using three piecewise cubics.

A five dimensional problem can then be set up as in Figure 4.4.

The piecewise variation of β is then prescribed as

$$\beta(x) = \begin{cases} c_{11} + c_{12}\,x + c_{13}\,x^2 + c_{14}\,x^3, & 0 \leq x \leq x_1 \\ c_{21} + c_{22}\,x + c_{23}\,x^2 \quad , & x_1 \leq x \leq x_2 \\ c_{31} + c_{32}\,x + c_{33}\,x^2 \quad , & x_2 \leq x \leq x_3 \\ c_{41} + c_{42}\,x + c_{43}\,x^2 + c_{44}\,x^3, & x_3 \leq x \leq x_4 \end{cases} \tag{4.4}$$

Using the boundary conditions, and the continuity conditions on β and setting the derivative at the end points to be zero, the coefficients c_{ij} can be expressed entirely in terms of $x_1, \cdots, x_5$ and $\beta_1, \cdots, \beta_3$, with the area of each cross-section being calculated using equation 4.3.

Parameterizations Two and Three – Hicks-Henne functions

A curve can also be modelled using the bump functions introduced by Hicks and Henne (1978). Their general form has been discussed previously in section 2.2. These functions always guarantee smooth curves and also have the appropriate end constraints, i.e., zero curvature. Here the function is superimposed onto a simple straight sided duct. Using one of these bump functions to describe the wall geometry provides three design variables: the amplitude, a, the distance along the duct centerline where the bump peak is situated, x_p, and the width of the bump, T.

Finally, the third parameterization involves two Hicks-Henne bump functions summed together giving a total of six design variables. This provides further geometries where stronger shape control of the curve is possible.

4.4.2 Results

In this first optimization study, a DoE was evaluated with a number of experiments equal to ten times the number of design variables, and twice this number of update points was computed, which is in keeping with standard practice (Sóbester and Keane, 2002). Due to the low number of design variables used, the RSM build time was not sufficiently limiting to lead to the need for an exploration of a reduced area of the design space.

After the optimal design was found, convergent-divergent diffusers presented competitive C_p values for all of the tested parameterization techniques. By converging the flow slightly at the inlet, the acceleration of the flow increases the local Reynolds number, the turbulence of the core flow and hence the turbulence of the boundary layer. This increase in turbulence is sufficient to prevent the onset of separation. The contours of velocity magnitude in a convergent-divergent diffuser, here found via the single Hicks-Henne approach, and in a straight walled diffuser are illustrated in Figure 4.5 for comparison. In the straight walled figure at the top, it can be seen that above the main core of flow there is an area of recirculation. Here, the flow has separated close to the diffuser entry as can be seen by the wall shear stress values along the wall in Figure 4.6. Because there is a region of high velocity flow and a region of very low velocity flow within the recirculation area at the filter point, the mass-averaged static pressure recovery returns a low overall value of $C_p = 0.21$. In contrast, the flow in the converging-diverging wall diffuser pictured at the bottom of Figure 4.5 expands with no separation, verified by positive wall shear stress values along the wall shown in Figure 4.7, and has a more even pressure uniformity across the filter. Because of this, the flow is generally at a relatively low velocity through the filter, even with the slight acceleration near the inlet caused by the converging wall, and hence returns a higher mass-averaged static pressure recovery value of $C_p = 0.36$. The optimum diffuser for both the double Hicks-Henne function and the spline technique also yielded a convergent-divergent diffuser almost exactly matching that of the one shown at the bottom of Figure 4.5. Hence, given an appropriate optimization strategy in terms of an accurately represented RSM, the different parameterization techniques tested are capable of producing results with the same geometrical features. In conclusion, given a straight diffusion of flow, the single Hicks-Henne approach uses the fewest design variables. However, for a turning expanding diffuser as studied in section 4.6, a polynomial spline parameterization or a double Hicks-Henne function would be the appropriate option to pursue because a single Hicks-Henne function applied upon a straight line connecting the entry to the filter of the airbox would only allow for one bump to contour the shape, whereas the other two techniques are capable of producing a convergence near the entry and a bulge near the exit, for example, giving stronger shape control of the wall.

With the inlet speed and expansion conditions, the point of separation occurring within the diffusers studied here are different to those studied by Carlson et al. (1967) and Madsen et al. (1999, 2000). This explains why none of the three parameterization techniques yielded a bell-shaped optimal design in contrast with the optimal bell-shaped designs found by them. The physical reasoning behind the bell-shaped designs is that provided the boundary layer is sufficiently turbulent at the inlet, they ensure that the boundary layer remains attached during the prime area enlargement. However, the studies in this chapter have a higher inlet speed than those studies of Carlson et al and Madsen et al and a larger expansion ratio for the boundary layer to remain attached. Hence, bell-shapes produced poor designs as the flow could not re-attach within the specified axial length, where the re-attachment occurred between 0.8 m and 1.2 m downstream of the filter, resulting in a low C_p value. Figure 4.8 illustrates an example of such a bell-shaped

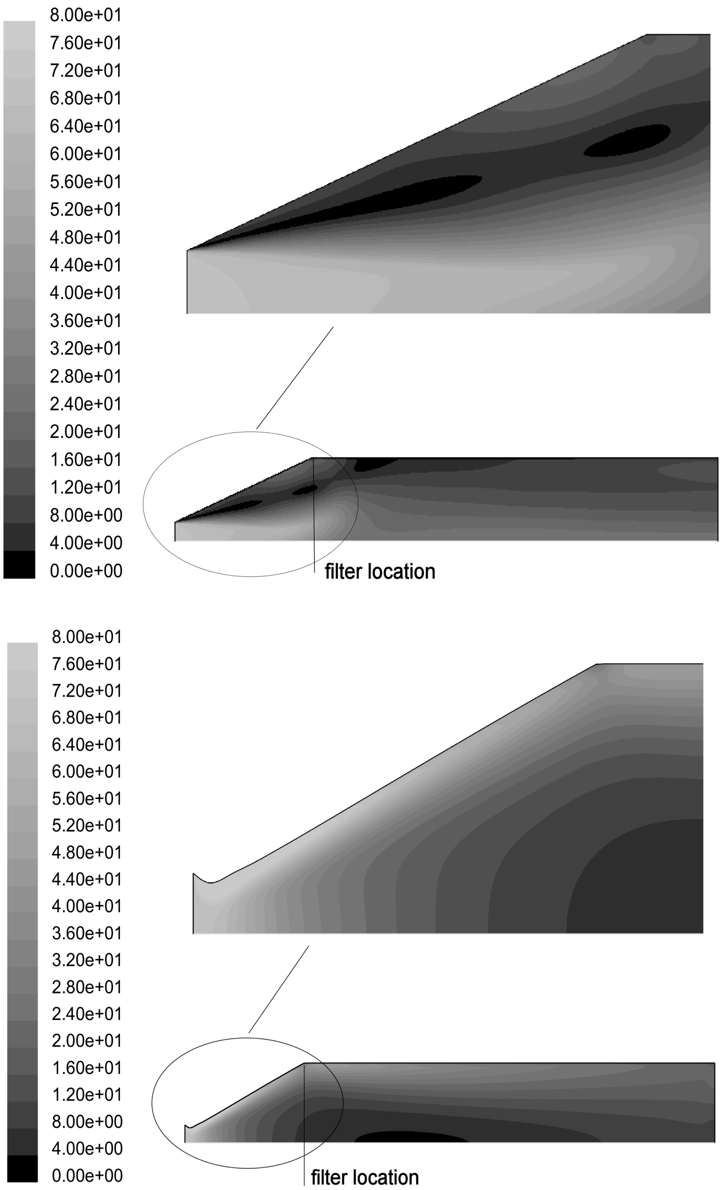


FIGURE 4.5: Filled contours of velocity magnitude in the whole computational domain of the symmetry half of a straight wall (top) and optimum convergent-divergent wall (bottom) diffuser (the diffuser section has been magnified to illustrate the difference in wall geometry)

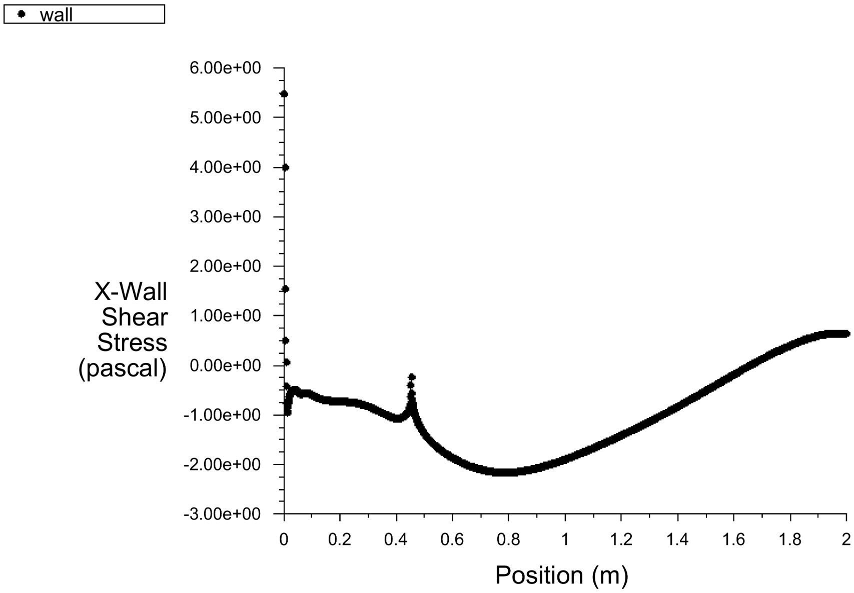


FIGURE 4.6: Wall shear stress values in the streamwise direction along the wall of a straight walled diffuser

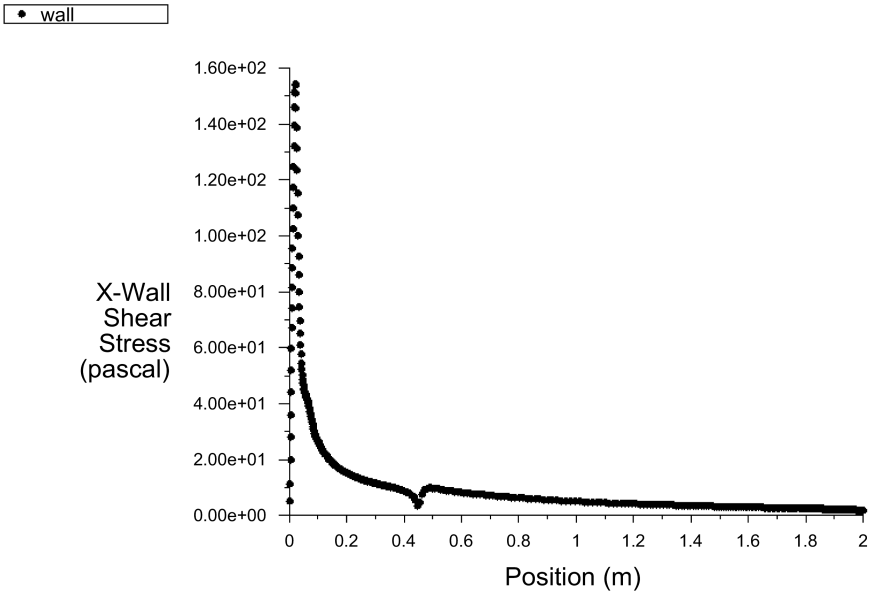


FIGURE 4.7: Wall shear stress values in the streamwise direction along the wall of the the convergent-divergent diffuser shown at the bottom of Figure 4.5.

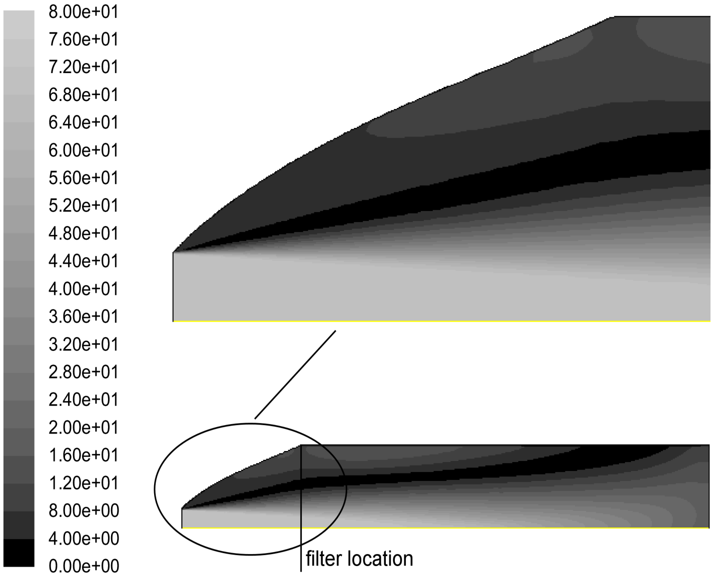


Figure 4.8: Filled contours of velocity magnitude in the whole computational domain of the symmetry half of a bell-shaped diffuser (the diffuser section has been magnified to show the wall shape more clearly)

design with $C_p = 0.12$. Due to the fixed engine position and strict overall car dimensions required by the F1 regulatory bodies, a longer diffuser to follow the flow streamlines in order for the flow to re-attach cannot be allowed.

Prior studies such as the five-dimensional case using B-splines tested by Madsen et al. (1999, 2000) constrained the geometries such that a positive wall slope was seen along the wall. However, considering the different required speed and expansion, this condition was not stipulated. By allowing the possibility of a constriction near the inlet, a prevention of separation would be possible. For the Hicks-Henne defined parameterizations the variable bounds were not limited so that a negative bump amplitude would be rejected for points placed near the entry point. Despite the spline technique limiting the wall height variable β to the interval $[0, 1]$, see Figure 4.4, the value of β and the position of the first control point coupled with an appropriate β value for the second control point could lead to a wall shape with a negative wall gradient, as seen near the inlet.

4.5 Elbow

Following the approach of the previous section two different parameterization methods for a duct of constant width turning through 90° are discussed next. The purpose of

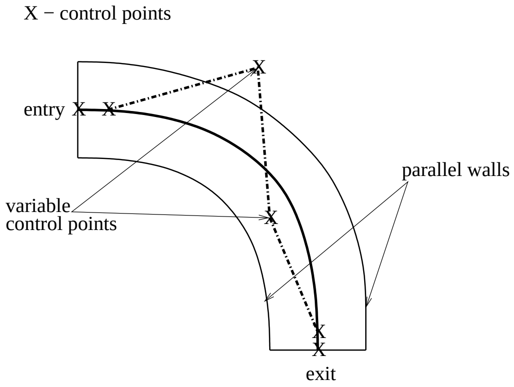


Figure 4.9: Geometry parameterization shown for a Bézier curve defined centreline

this study is to understand the flow physics surrounding the centreline bend and to determine the best method for use in a curved diffuser.

4.5.1 Geometry Parameterization

An optimization cycle is implemented on a non-diffusing elbow over the same distance in which an F1 airbox turns the flow. This returns similar C_p values no matter how the geometry is varied because the boundary layers on both walls remain attached. By shortening the distance over which the elbow bends through 90° separation can be forced, making the optimizer work harder to differentiate between alternative parameterizations. This shortened distance is used in the following optimization studies of this section.

Parameterization One – Bézier curves

The first method employs a Bézier curve with six overall control points defining the centreline, of which two are considered as variables (see Figure 4.9). Two parallel walls are constructed equidistant from the centerline on each side to define the constant width elbow. Horizontal and vertical tangency conditions are implemented at the entry and exit, respectively, by fixing control points along the required tangency. These conditions prevent designs harbouring a sharp point at the diffuser exit onto which the trumpet tray would join.

Parameterization Two – Polynomial splines

This parameterization utilises a spline passing through three points positioned along the centreline. Again, two parallel equidistant walls are placed either side of the centreline.

4.5.2 Results

Similarly to the studies of section 4.4, none of the techniques tested in this section are high-dimensional problems, and again an optimal design is found before an exploration of a reduced area of the design space is deemed necessary.

Here, all of the designs tested within the optimization process returned a negative pressure recovery. This is because there is no expansion and hence no pressure is being recovered due to the reduction of the flow's kinetic energy. Thus, the static pressure over the filter or exit of the bend is less than the static pressure at which the flow enters the bend. However, both parameterization methods returned similar optimum pressure recoveries of $C_p = -0.0413$ for the Bézier curve method and $C_p = -0.0415$ for the spline method, with similar geometries. These geometries turn the flow through 90° and at a gradual and even rate throughout the bend. This ensures no boundary layer separation.

Although both designs produce similar optimum C_p values, in looking forward towards the fusing of the expansion and the bend, the piecewise spline approach appears to be the most efficient when defining the centerline bend. Bézier curves have an inherent problem in that tangency at an end point of the curve cannot be set without requiring extra fixed control points next to the entry and exit points in the direction of the required tangency through which the Bézier curve must pass. The degree of tangency is then based upon the distance of these extra control points away from the points positioned on the entry and filter planes, which could become extra variables if necessary. This drawback is easily overcome by using polynomial splines. Given this extra complication within the geometry, the remaining study is pursued with the piecewise cubic spline defined centreline.

4.6 Diffuser with Bend

4.6.1 Combined Geometry Parameterization

The fusion of the expansion and bend parameterizations from the preceding two sections is next used to create a two-dimensional airbox model. It is important to note that to maximize the amount of shape control of the complete diffuser given to the optimizer, thus allowing the production of potentially radical results, the upper and lower walls need to be completely decoupled. Thus, the positioning of the control points through which the upper wall spline passes must be independent of the control points defining the lower wall spline. This is ensured by using a model in which there are no links between the centerline bend and spline control point positioning variables of the lower wall. Problems of wrinkles or loops forming on the lower wall spline leading to geometrically infeasible designs are then prevented.

The chosen parameterization involves the use of piecewise cubic splines for all three sections. As noted in section 4.4, the most efficient simple expansion materialised as a convergent-divergent diffuser. Hence, here the upper and lower wall variables are left free to produce convergent-divergent diffusers should high C_p values be returned for this combination of design variables. In section 4.5, it was shown that an efficient turning of the flow occurs through a regular and gradual bend and hence the ranges allowed for the three centerline control points are chosen with this in mind. The following parameterization technique for the airbox model features decoupled walls together with the capability of designs allowing converging walls or a wide range of features due to the strong shape control.

The technique adopted is illustrated in Figure 4.10. This requires a total of 16 design variables: six for the centerline control points CP1(x,y), CP2(x,y), CP3(x,y); five for the upper wall r_{u1}, r_{u2}, r_{u3} (these are the normal distances from the centerline) positioned by two variable ratios, D_1 and D_3, along the centerline with the middle ratio, D_2, fixed at 0.5; and five for the lower wall r_{l1}, r_{l2}, r_{l3} (these are the distances along the lines intersecting the centerline) defined by two variables ϕ_1 and ϕ_3 with ϕ_2 fixed at 45°.

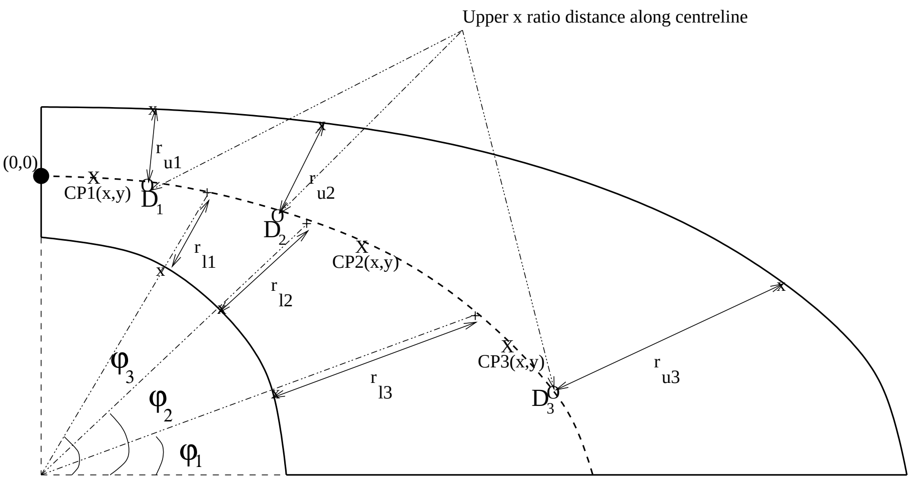


FIGURE 4.10: Geometry parameterization of 2D airbox model

4.6.2 Results

Although 16 design variables is practicable for the use of Kriging as the RSM approach, the limit of practical RSM building times is reached due to the high-dimensionality of this problem. The time taken to build the RSM with 300 points in 16 dimensions exceeded the time necessary to run the simulation for each update point itself. Hence extending the optimization strategy to include a simple, concentrated exploration of a reduced area of the design space is warranted (Figure 2.1).

For this study, the populating of the design space using a DoE and the search for the updates is performed within a 16-dimensional hypercube defined by the upper and lower bounds of the 16 design variables. The design space in which the concentrated exploration is performed after the global RSM based search is fixed in a reduced area of this design space. A small percentage of the design space for each variable is taken around the current best point found after the updates. This exploration region is 20% of each of the design variable ranges with the centre of this smaller hypercube at the best point. Within this exploration region a localised set of geometries are constructed via a further 50 point LPτ DoE and these points are evaluated using the CFD code to obtain their objective function values. No RSM is built using these values as the purpose of this search is to find potentially superior designs through the dense sampling of points within a small area without imposing any assumptions as to the nature of the design landscape, which would occur when fitting an RSM.

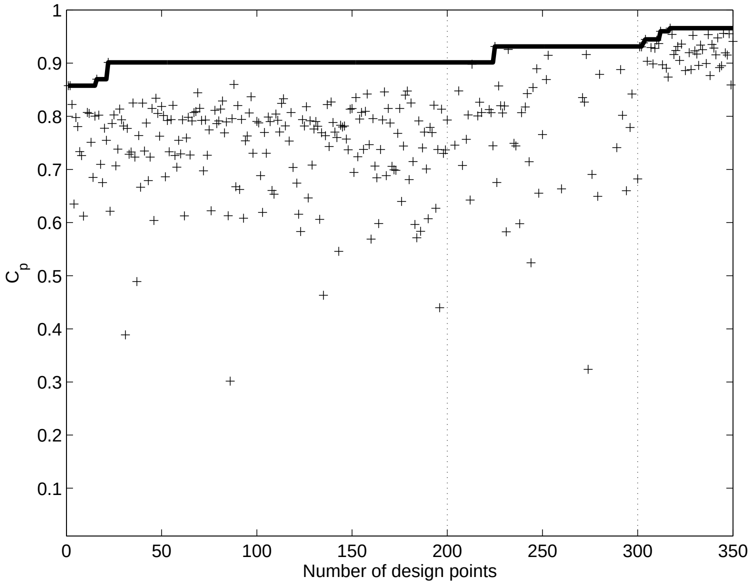


FIGURE 4.11: Optimization history: 200 DoE points followed by 100 update points, further followed by a 50 point exploration in 20% of the design space

Figure 4.11 shows the development of the optimization process, showing C_p values for design points 1 to 200 representing the initial 200 DoE points, design points 201 through 300 representing the subsequent RSM based update points and the final 301 to 350 design points from the 50 point concentrated exploration. The bold line indicates the current best optimum as each point is added.

A wide range of interestingly shaped diffusers have been produced within the first 300 design points, from convergent-divergent wall shapes to diffusers with bulges featured on the lower and upper walls. Many designs, such as the convergent-divergent types contain no areas of separation. However, due to the increase in kinetic energy of the flow through the slight convergence of the wall near the inlet, the pressure recoveries are lower than the best value seen; the best ranges between $C_p \sim= 0.7$ and $C_p \sim= 0.8$. Designs that are similar to those seen within the actual race cars are also found within the design space, with straight upper walls to fit within the present design of roll bar structure, i.e., they have no radical geometric features such as bulges seen on either wall. Velocity contours of one such airbox can be seen in Figure 4.12 where separation has been completely eliminated. However, with a pressure recovery of $C_p = 0.7805$ it is significantly lower than the best design found. The best design found in the first 300 calculations has a pressure recovery of $C_p = 0.9316$ and the velocity contours for this geometry are shown in Figure 4.13. Here, completely different geometric features have emerged. Two small

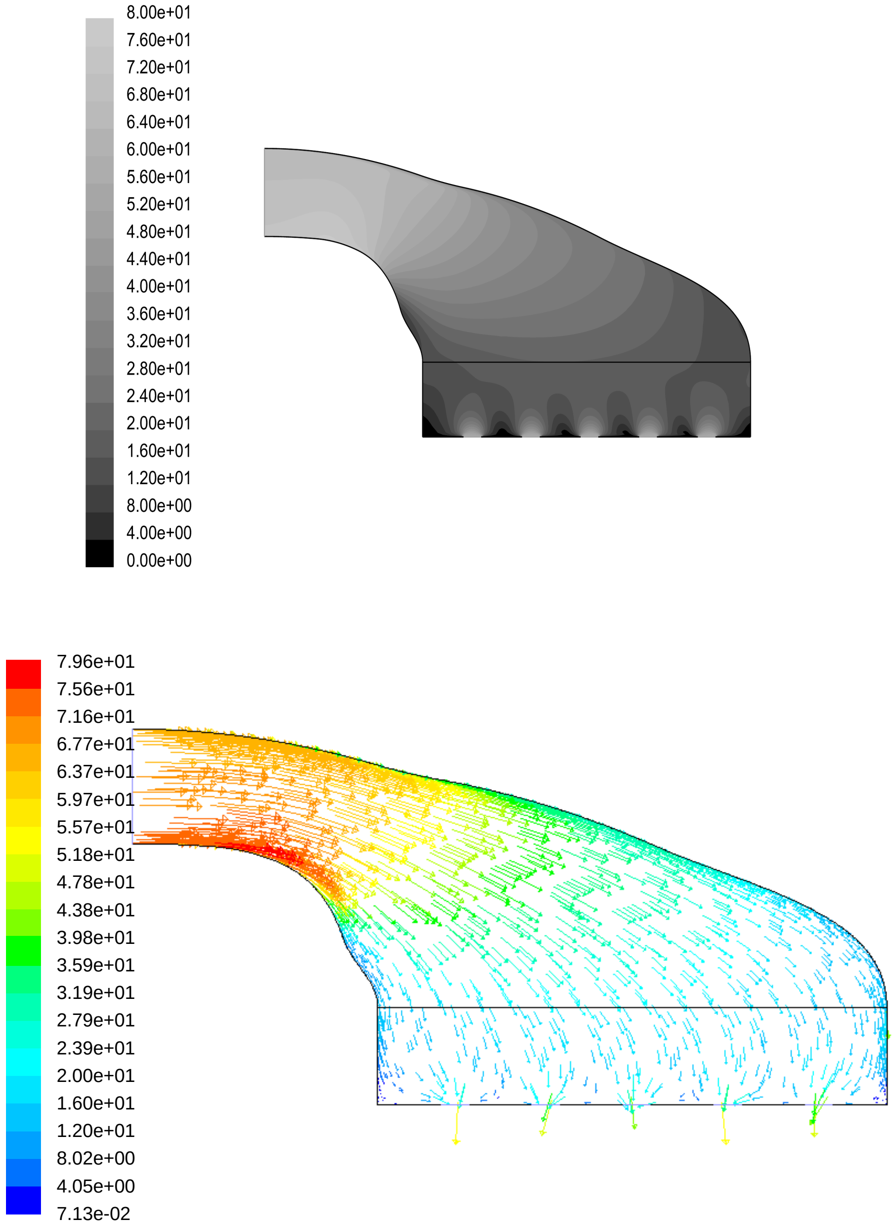


FIGURE 4.12: Filled contours of velocity magnitude (top) and velocity vectors (bottom) of velocity magnitude in a design which contains no flow separation, with $C_p = 0.7805$

bulges have been formed along the lower wall inducing separation within these bulges; the bubbles of separation are completely contained within the bulges. The upper wall also contains a bulge, and within this another bubble of separation forms. It is important to note here that these bulges are not due to the case of inadvertent "rippling", as described in section 4.4.1. All three control points along the lower spline, in this case, sit at the inflexion points of the curve and so the spline is not 'overshooting' the point, which is what leads to the "rippling" situation.

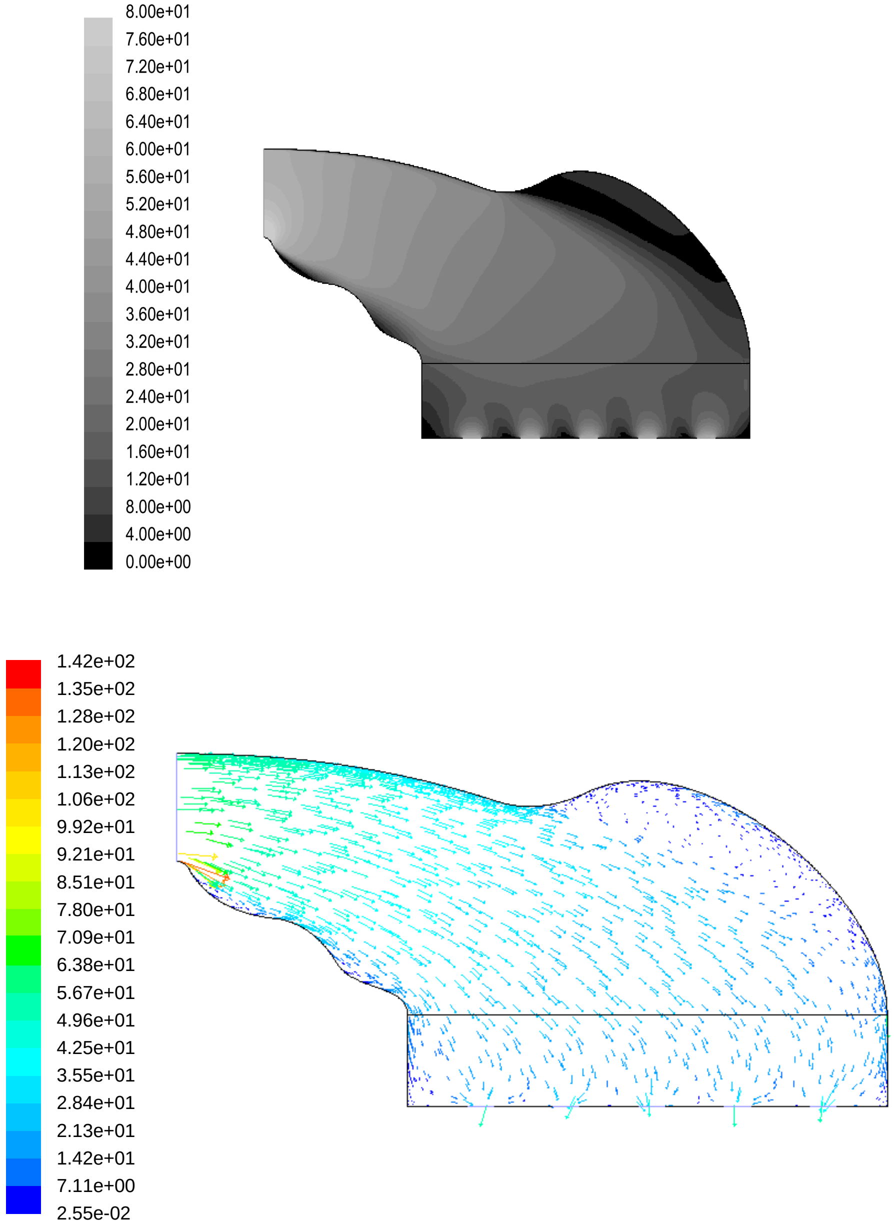


FIGURE 4.13: Filled contours of velocity magnitude (top) and velocity vectors (bottom) of velocity magnitude in the best design after 100 update points, with $C_p = 0.9316$

After 300 design points, the process continues with a concentrated exploration around the current best point found (with $C_p = 0.9316$). In Figure 4.11 these are shown as design points 301 to 350. It is clear that by focusing on a small region within such a large design space, and by performing a dense search in this area, the model is converged more quickly to a better design yielding a higher C_p value. The optimum design found returned a pressure recovery $C_p = 0.9658$ and the velocity contours for this geometry are illustrated in Figure 4.14. As can be seen here, the upper bulge has been reduced

Variable	Lower bound	Upper bound	Best value after updates	Best value after concentrated exploration
CP1x	0.1	0.175	0.1408	0.1399
CP1y	-0.015	0	-0.0122	-0.0120
CP2x	0.25	0.325	0.2501	0.2481
CP2y	-0.08	0	-0.0675	-0.0685
CP3x	0.375	0.55	0.4181	0.4290
CP3y	-0.202	-0.09	-0.2018	-0.1920
Upper x ratio 1	0.05	0.45	0.1853	0.1703
Upper x ratio 2	0.5	0.5	0.5	0.5
Upper x ratio 3	0.55	0.9	0.6162	0.5943
r_{u1}	0.055	0.08	0.0651	0.0673
r_{u2}	0.07	0.14	0.1359	0.1333
r_{u3}	0.145	0.28	0.2128	0.2111
ϕ_1	5	40	17.9152	18.3529
ϕ_2	45	45	45	45
ϕ_3	50	85	84.8928	83.5803
r_{l1}	-0.08	-0.055	-0.0734	-0.0712
r_{l2}	-0.14	-0.07	-0.1295	-0.1234
r_{l3}	-0.2	-0.15	-0.1506	-0.1462
Pressure recovery			0.9316	0.9658

Table 4.2: Design parameters and their corresponding bounds with the variable values for the best design found after 100 update points and again after a further 50 point exploration

slightly, eliminating separation on this wall. The lower bulges, however, remain present and capture small bubbles of separation within them. The upper and lower bounds of the whole design space, together with the design variable values for the best design after 300 design points and after the concentrated exploration are shown in Table 4.2.

An interesting observation is that after the evaluation of the update points and analysis of the best design obtained, one intuitively may think that a better design could be achieved by eliminating separation, creating a design with walls approximately following the flow streamlines of the airbox illustrated by the velocity contours in Figure 4.13. However, the subsequent concentrated exploration did produce a design without separation which returned a pressure recovery value $C_p = 0.9452$; the velocity contours for this geometry are shown in Figure 4.15. If the streamlines of the airbox in Figure 4.13 were followed, the upper wall would have been straight. What has been shown in this study is that by a small manipulation of the design variables, separation can be eliminated without following the streamlines of the best design after the update process to produce a design with a higher pressure recovery value. Another interesting design found within the concentrated exploration returning a high pressure recovery value of $C_p = 0.9555$ is shown in Figure 4.16. This airbox does feature a straight upper wall as well as two small bumps on the lower wall. This is of particular interest to the F1 aerodynamicist as this design would not require a redesign of the current roll bar structure in which the airbox sits.

It must be noted here that all the results discussed in this section are dependent on both the grid and the turbulence model chosen for the CFD simulation. It is likely that these same geometries simulated using a different turbulence model would produce different values of pressure recovery. However, the overall conclusions made here are to determine the capability of the parameterization techniques used to produce the geometries. Should a preferable CFD set up be found to produce more accurate pressure recovery results, the conclusions regarding the capability of the parameterization technique remains the same.

It can be seen that, for all the airboxes illustrated in Figures 4.14, 4.15 and 4.16, the elimination of separation on the upper wall is desired to return a high pressure recovery. However, it is also clear that the lower wall bulges are useful features in delivering an efficient expansion of the flow without experiencing losses from extensive separation. These bulges are allowed to exist due to the parameterization technique proposed, the key attribute of which is the geometric independence of the upper wall from the lower wall. Bulges containing small regions of separation have previously been seen to be beneficial in terms of reducing excessive non-uniformity of the flow at the diffuser exit when applied to a curved subsonic S-duct diffuser (Zhang et al., 2000). Increased losses in the airbox occur from separation. This is not the case here as the small separation bubbles are completely contained within the lower wall bumps. In this study, the two bulges containing small separation bubbles are beneficial, returning a higher pressure recovery over that of a design which eliminates separation entirely. The balance between the size of bump on the upper wall and size of bumps on the lower wall, which together control the expansion of the flow as it turns, is very fine. This balance has been explored computationally through the capability of the parameterization technique proposed as it has strong shape control. However, there is a need for further research into such features, including experimental testing.

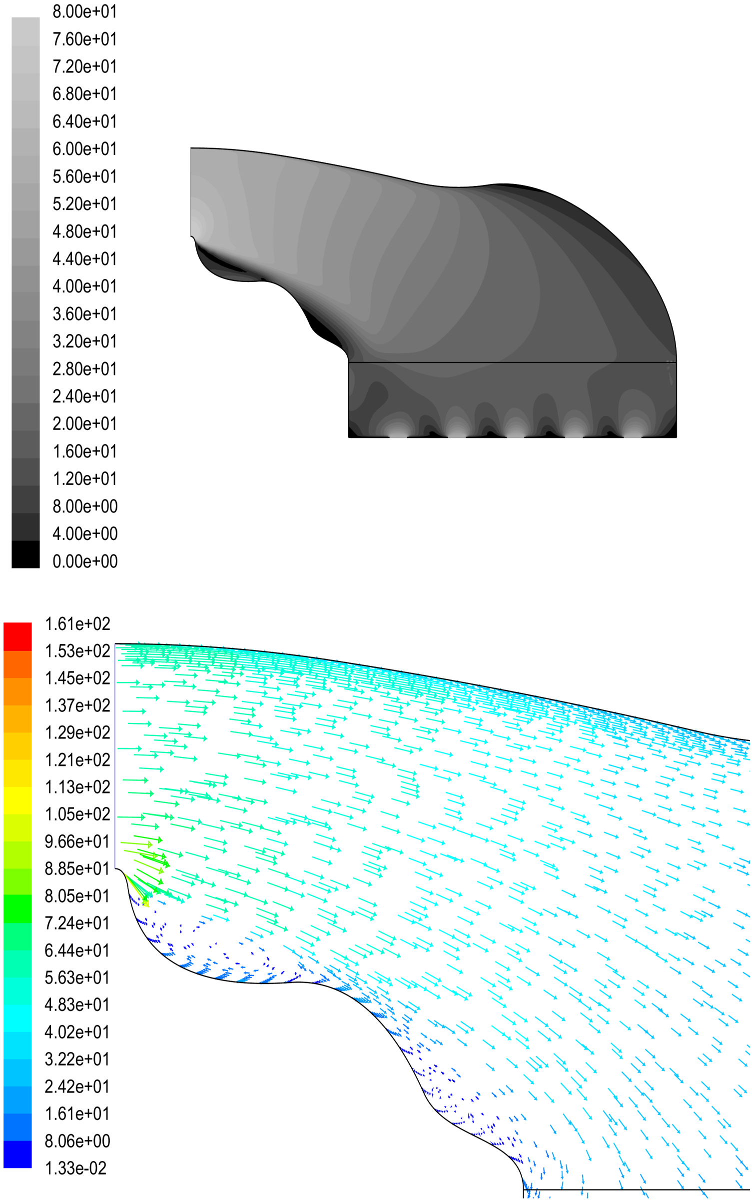


FIGURE 4.14: Filled contours (top) and velocity vectors (bottom) of velocity magnitude in the optimum design found having completed a 50 point concentrated exploration with $C_p = 0.9658$

4.7 Summary

This chapter has explored several curve parameterization techniques for the optimization of a straight diffuser and of a constant width elbow to develop an understanding of how to effectively parameterize a two-dimensional F1 airbox. The conclusions drawn from the initial studies aided the development of a novel parameterization technique for a 2D airbox allowing strong shape control of the wall.

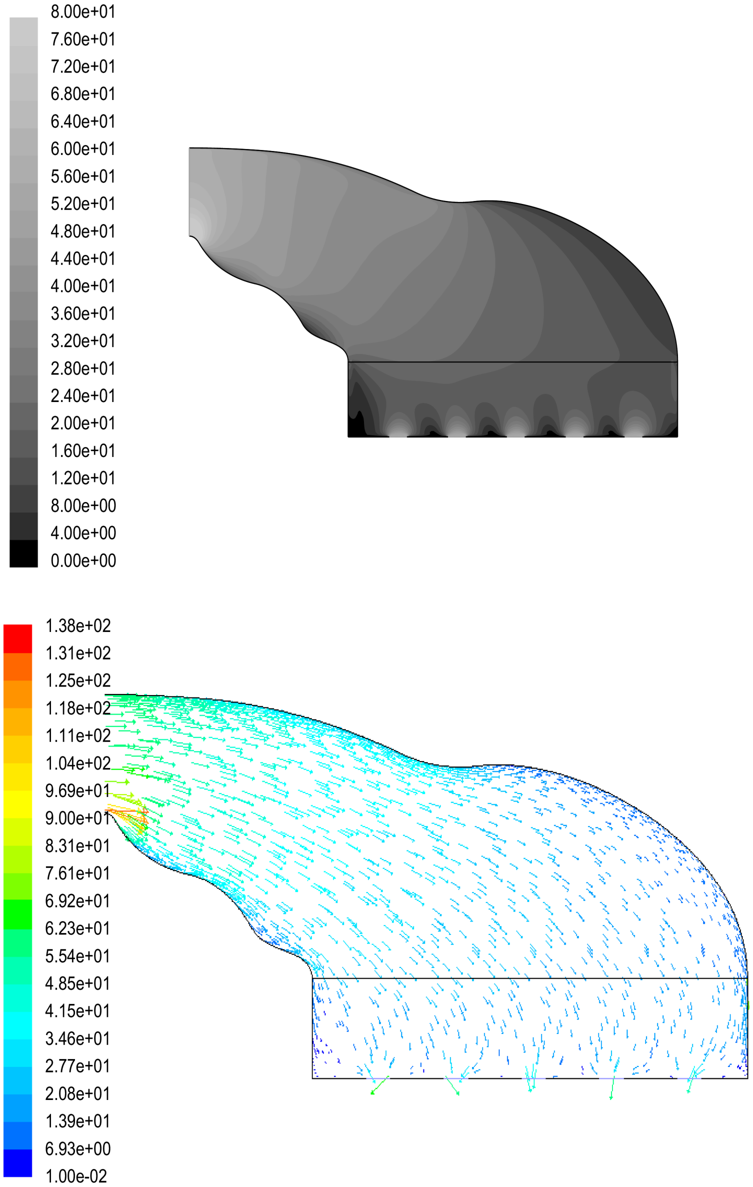


FIGURE 4.15: Filled contours (top) and velocity vectors (bottom) of velocity magnitude in a design which contains no separation with $C_p = 0.9452$ found during the 50 point concentrated exploration

The 2D airbox has been optimized with respect to pressure recovery. Due to the relatively high number of design variables needed to allow for strong shape control of the wall, a concentrated exploration of the design space was necessary to find an optimum design, as the time taken to build the RSM to locate update points became prohibitively expensive. Radical designs with high pressure recovery values were found to contain bulges, highlighting the importance of strong shape control in the construction of a parametric geometry.

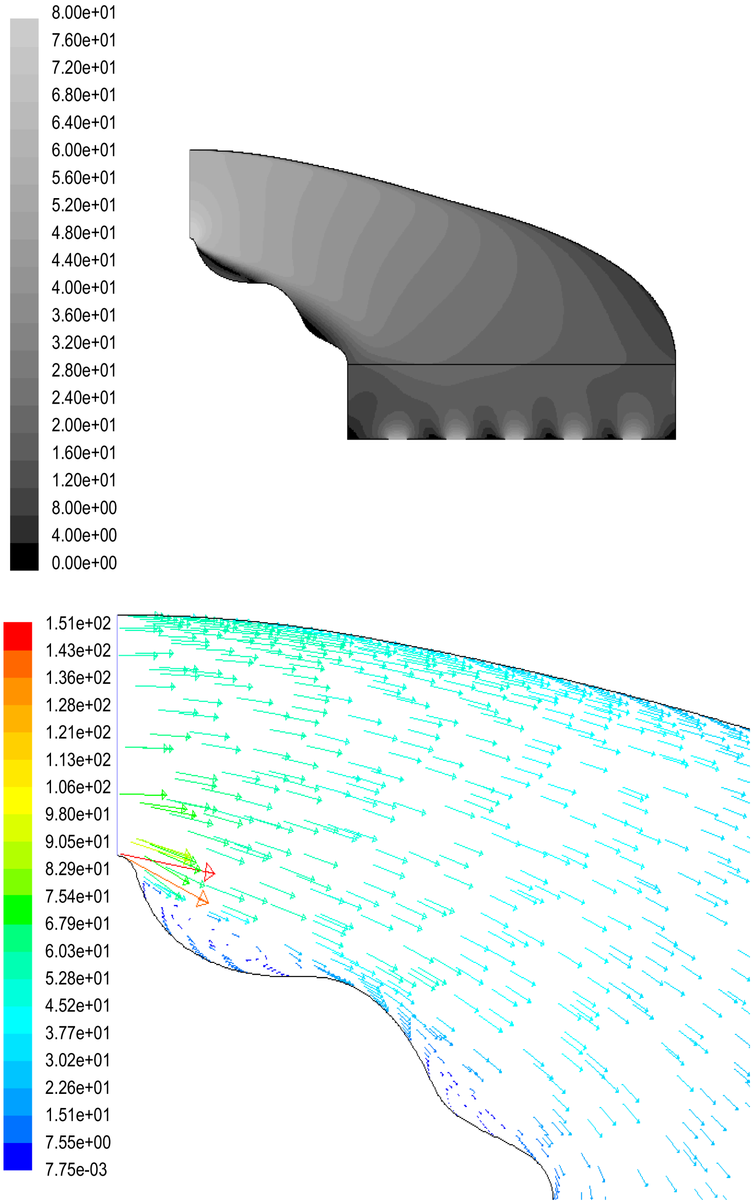


FIGURE 4.16: Filled contours (top) and velocity vectors (bottom) of velocity magnitude in a design with a straight upper wall and $C_p = 0.9555$ found during the 50 point concentrated exploration

The optimum geometry found after the concentrated exploration of the design space featured a large bulge on the upper wall and two small bulges on the lower wall, the bulges acting as areas where the separated flow could be contained. For the set of boundary conditions used in this chapter, the bulges were beneficial to the airbox's performance. This leads to a desire to perform a design optimization study on a three-dimensional airbox to determine whether such bulges may be beneficial for a more realistic flow simulation. Clearly, a three-dimensional parameterization technique that could provide strong shape control of the whole airbox wall and also allow for three-dimensional bulges needs to be developed. In light of this, a variety of three-dimensional parameterization techniques are discussed in the chapter that follows.

Chapter 5

Automated Multi-Stage Shape Optimization with Deformation (AMSSOD)

In Chapter 2 a number of parametric representations of curves were reviewed and in Chapter 4 these techniques were used in optimizing 2D diffusers. To progress into three dimensions, a comprehension of how CAD engines deal with surface representation is appropriate.

In three dimensions, the two main methods of geometry modelling that CAD engines use are solid modelling and surface modelling. Solid modelling allows a surface to be generated as a composite of primitive solids. Cubes, cylinders, spheres and cones, for example, are used with boolean operations to intersect, subtract or append each other to build up a model with a complex surface. Surface modelling, on the other hand, utilises parametric representations of surfaces to describe a complex object. The majority of engineering problems implement both of these methods successfully. Surface modelling is more general and requires perhaps a simpler and more intuitive construction via parametric curves; this is particularly useful where part of a complex surface is not clearly representable by a boolean collection of operations. For the studies that are represented later in this thesis, an important consideration is whether the method of describing the geometry may be fully automated and easy to manipulate. In this sense, drawbacks of representing a complex object may be overcome with the parametric method. This chapter focuses on the parametric representation of surfaces for use in the manipulation of three dimensional surfaces.

Parametric surface definition can be split into two main camps: surface patching and polygon meshing. A surface patch can be described as a single bivariate surface element $(u, v) \rightarrow \mathbb{R}^2$ where $u, v \in [0, 1]$ (see Figure 5.1). Complex surfaces are modelled using a number of these patches fitted together, much in the same way as piecewise curves are fitted together to obtain a higher degree of complexity as described in Chapter 2. A polygon mesh, on the other hand, is a much simpler way of representing a surface. It is favoured by communities who wish to avoid the additional time and computational cost of re-meshing after a small manipulation of the geometry. This type of surface is described simply as a discrete set of points making up an array of polygons fitted together; i.e., the collection of discrete points are joined together by straight lines to define a two-dimensional "meshed" curve. Although the polygon mesh is simpler and more flexible in many cases, a surface patch has a number of distinct advantages. First, the parametric representation is analytical and properties of the surface may be extracted, whereas the exactness and smoothness of a polygon mesh can only be improved by increasing the resolution of the polygons which can be expensive for complex surfaces. Second, as the deformation or manipulation of surfaces is a necessity when dealing with design optimization, manipulation of a parametric surface patch is achieved through altering the values of the control points which define it. The modified surface is just as well defined as its original counterpart and, as such, will retain its smoothness and topology. With a polygon mesh, however, if the change to the discrete set of mesh points is made to a geometry of low curvature, described by few polygons, and this change results in a surface of high curvature, the number of polygons defining the surface becomes too few and the accuracy and smoothness of the geometry is compromised. This problem can be corrected by individual polygon subdivision in the appropriate area but it is not a trivial procedure and will require greater memory for the storage of the increased number of mesh points. A crude example of this is shown in Figure 5.2. The wireframe of the teapot on the left is defined by a parametric non-uniform rational B-spline (NURBS) surface (a form of surface patch) and the wireframe of the teapot on the right is defined by polygons. It is clear that the NURBS surface captures the shape more accurately

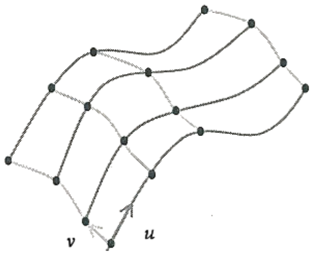


Figure 5.1: Bivariate surface

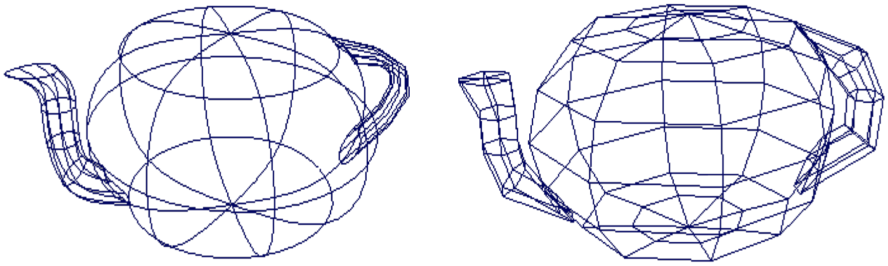


Figure 5.2: Wireframe model of teapot constructed with NURBS surfaces (left) and a polygon mesh (right)

and is inherently smooth. In contrast, the polygons need to be defined using a much finer resolution to capture the required curvature. It is for these reasons that all the three-dimensional surfaces in the subsequent chapters are defined as parametric surface patches, and each change in geometry is exported for re-meshing before analysis via CFD.

In many methods of surface patching with CAD engines, lofting tools are used to provide a surface between guiding curves or boundaries. Between these guiding curves, CAD engines commonly calculate the construction of a network of interpolating curves. This network of curves acts as the surface patch boundaries and each of these patches is 'filled in' to generate the surface geometry. It is important to understand how the automated lofting works in such CAD engines as it is this that may ultimately limit the capability of automated surface manipulation. The following section considers how the representation of an object impacts on the way in which it can be manipulated. The subsequent sections present a survey of a variety of commonly used surface representations, along with their advantages and disadvantages with respect to automated shape optimization. The chapter concludes by proposing an automated multi-stage parametrization technique for three-dimensional shape optimization to be implemented in the following chapters.

5.1 Global versus Local Geometry Manipulation

The parametric method used to represent an object's surface usually constrains the nature and extent to which the surface can be manipulated. The technique that is used to manipulate this surface becomes the deciding factor for whether the required manipulation can be achieved. In previous chapters the importance of maintaining strong shape control of the geometry in the method of parameterization has been mentioned. To recapitulate, strong shape control allows a large amount of geometric variability given a specified set of design parameters.

Due to the complex nature of many three-dimensional objects, it is important to consider whether the geometry change that is permitted through the method of parameterization has an impact on the geometry as a whole or on a small area of the geometry. These impacts can be thought of as acting in a global or a local sense. The advantage of a technique which allows one to manipulate a geometry in a global sense is that it can be used to perform design optimization studies for conceptual design, for which it is unclear to the designer what general shape should be chosen. Efficient optimization processes require a compact set of design variables, as discussed in Chapter 2. In defining a parametric geometry with a compact set of design variables, with each variable having a global affect, an optimal conceptual design in terms of any metric is rapidly achieved. However, if the required complexity of the shape is not present (because local surface control has been sacrificed in favour of low-dimensionality), then the use of a local parameterization technique, with both a compact set of design variables and strong shape control, would facilitate a more efficient optimization study. By using a local parametric deformation technique, independent of any previous global technique used, efficient optimization of local regions of such geometries is possible. A brief survey of techniques which may be used to parametrically define three-dimensional surfaces follows. Each technique has its own merits in terms of global and local manipulations of the surface and these are discussed.

5.1.1 Non-Uniform Rational Polynomial Spline Surfaces

Non-uniform rational polynomial spline (NURPS) surface representation is used for the lofting of multi-section surfaces and for filling an area defined by a closed boundary curve in CATIA V5. This type of surface interpolates the control points defining the patch boundaries. The surface is generally used when surface sections are built using boundary curves or guide splines, which could be defined using polynomial spline curves or Bézier curves. A baseline airbox geometry defined by multi-section surfaces with four guiding polynomial splines can be seen in Figure 5.3.

FIGURE 5.3: Baseline geometry defined using NURPS surfaces

Such a surface can be manipulated automatically via an external design table by changing the values of the control points defining the guiding curves. An increase in the number of control points and/or guiding curves will increase the extent of the allowed manipulation in a local sense. It is conceivable that one could maximize the possibility of local surface manipulation, after attaining a good global shape, by increasing the number of guiding curves. This would require a re-parameterization of the model and re-lofting so that the surface passes through the added guiding curves. However, this will provide a larger number of parameters and therefore reduces the computational efficiency of the design search. Thus, although simple and straightforward for attaining automated global surface manipulation, this method cannot be considered for a local manipulation technique.

5.1.2 Non-Uniform Rational B-spline Surfaces

Non-uniform rational B-spline (NURBS) surfaces are a generalisation of the B-spline surface patches; the key departure from NURPS is the weighting of the control points which makes NURBS surfaces rational. Let u and v define a bivariate patch ($u, v \in [0, 1]$), then a NURBS surface $S(u, v)$ of order (p, q) is defined parametrically with its shape determined by control points, $\mathbf{P}_{i,j}$, weights $w_{i,j}$ and the NURBS basis functions $N_{i,m}$ and $N_{j,n}$:

$$S(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} N_{i,p}(u)N_{j,q}(v)w_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{m}\sum_{j=0}^{n} N_{i,p}(u)N_{j,q}(v)w_{i,j}}. \tag{5.1}$$

For a more detailed review of NURBS curves and surfaces, please refer to Piegl and Tiller (1996).

The continuity of this type of surface depends only on the basis functions and not on the control points. NURBS have all the advantages of B-splines, while extending liberty of modelling. Rational cubic surfaces allow the construction of conic sections such as spheres and cylinders as well as free-form shapes. In contrast to the NURPS surface previously described, NURBS are simply rational polynomial patches of a fixed degree.

The construction of NURBS surfaces requires a quadrilateral mesh of $m \times n$ points; very few objects can be constructed appropriately with a single rectangular patch. A combination of patches to describe complex shapes can, however, lead to complications. These complications are often in the form of continuity at points where patches meet. For complex surfaces, deforming a NURBS patch can result in a number of problems:

- the patches can tear apart at the seams leaving a discontinuous geometry, or
- the continuity could be made to force the curvature to zero creating flat spots on the geometry in these regions.

In order to address these problems, high degree patches could replace the patches at these problem points. However, this adds more control points to the set of parameters, increasing the complexity of the model.

Another disadvantage of NURBS surfaces is that, if a local shape deformation is required, detail is added to the mesh of control points by adding more if necessary, but, unless the local patch can be trimmed and located to a specific position, extra control points are added where they are not necessary, thus increasing the parameter set without good cause. This is illustrated in Figure 5.4. If trimming of the surfaces is avoided, allowing the patch to span a section of the NURBS surface from inlet to filter, and if a small local

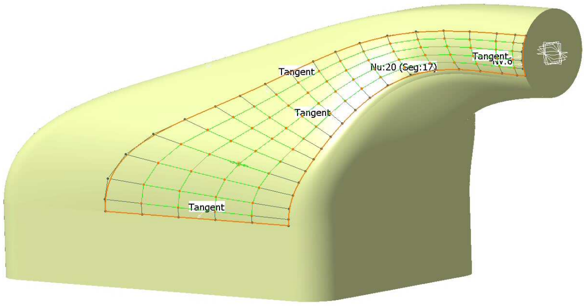


FIGURE 5.4: NURBS surface representation with a mesh of control points over a deformation patch

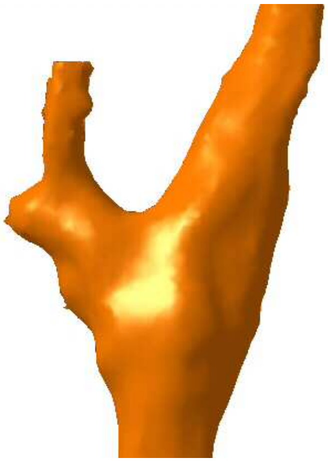


FIGURE 5.5: An example of the inner wall surface of a real artery

area within this patch is to be deformed, a complex deformation may require a much finer mesh of control points than those shown in Figure 5.4. This would add control points where no deformation is necessary and increase the complexity of the entire model, perhaps making this method unsuitable for a local geometry manipulation.

For a three-dimensional airbox study, these surfaces provide a good representation of the global shape and a surface can be defined with a coarse net of control points. The positions of these points can easily be modified to alter the shape of the airbox wall. The position of each control point requires only three design variables. If the modification of more than one control point at a time is required, then a compact set of design variables can still be retained for an efficient global optimization process for the first stage in the two stage process. Local manipulation, however, may increase the parameter count to beyond the practicable limit for use in optimization.

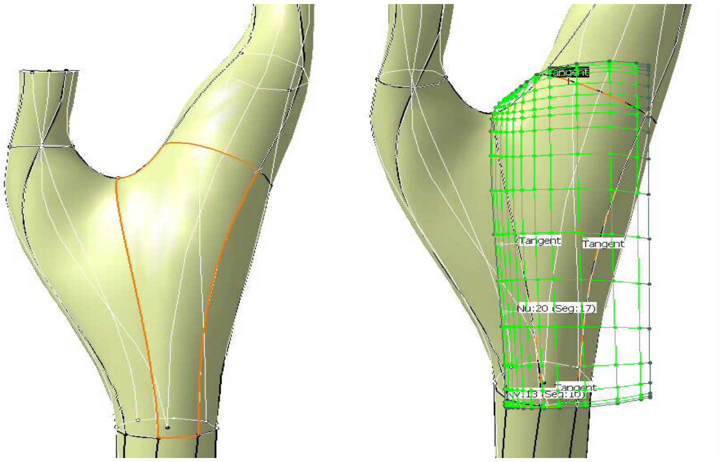


Figure 5.6: NURBS surface representation of artery (left) with the net of control points defining the lower right hand side of the bifurcation (right)

Figure 5.5 shows a surface fitted through a cloud of points collected via a scan of a patient artery cast. It is clear that to represent a realistic parametric carotid artery bifurcation model, a combination of NURBS surface patches is necessary to capture sufficient detail. For such a complex geometry, the general shape can be captured and an example of a CAD fit to the real artery shape can be seen on the left of Figure 5.6. The right of Figure 5.6 shows the net of control points controlling the NURBS patch defining the lower right hand side of the carotid artery bifurcation. To capture the global shape of the surface, a large number of patches is required and a fine net of control points to capture the complexity of the shape. The patch is not as straightforward as the deformation patch shown for the airbox. It is because of this that a global surface fit can be achieved but, even with the tangency conditions set at the boundary of the patch, when a control point near the boundary is displaced, resulting surface deformations become problematic with seam discontinuities.

Figure 5.7 illustrates the displacement of a control point near the patch boundary, its resulting surface deformation and the seam discontinuity at the join with its adjacent surface patch. It is because of this problem that this technique could not be considered for any local surface deformations.

5.1.3 Partial Differential Equations

Bloor and Wilson (1990) presented an efficient method of parameterizing surfaces by solving partial differential equations (PDEs). With this method, the surface is treated

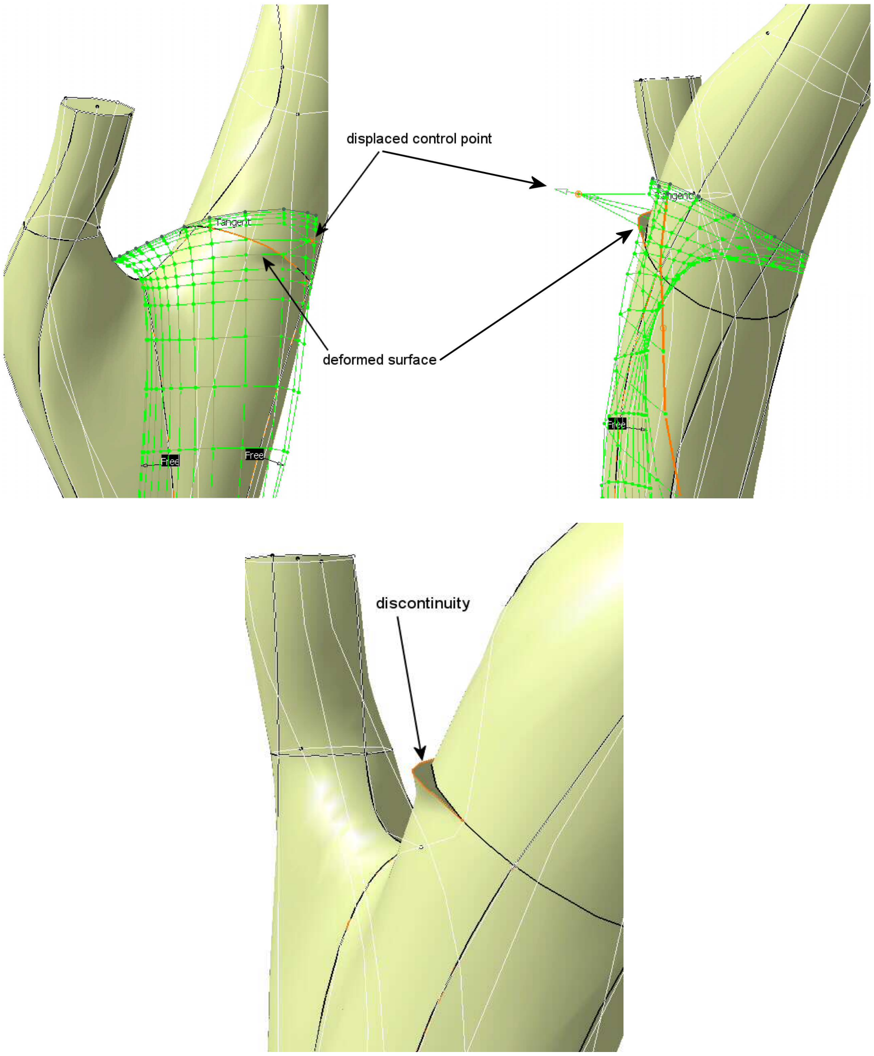


FIGURE 5.7: Seam discontinuity at patch boundary created by the displacement of a control point in the control net

as a boundary value problem and produces surfaces as the solutions to elliptic PDEs. This requires a small set of design variables and thus provides an efficient technique for use in conjunction with an optimization tool. A number of boundary curves representing positional and derivative conditions of the resulting surface are used. Each of these boundary curves is described by a set of parameters which define values of translation, rotation and dilation in three dimensions as well as a smoothing parameter. For complex objects, a number of PDE patches can be combined at common boundaries and the automation of this technique is straightforward (see Ugail and Wilson (2003)). A clear advantage of this form of surface representation is that it eliminates any discontinuities between surface patches that can occur with some of the spline-based surface patch techniques.

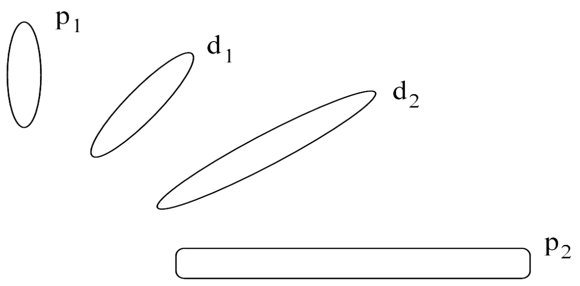


Figure 5.8: Illustration of positional and derivative boundary curves to define a 3D airbox

For a three-dimensional airbox, let us assume that two positional boundary curves would be used, one fixed at the inlet and the other fixed at the filter position, p_1 and p_2 respectively, as shown in Figure 5.8. To provide satisfactory geometric manipulation in a global sense just two derivative boundary curves d_1 and d_2 could be used. Discounting the need for rotation of d_1 and d_2, allowing the translation to occur along the airbox centreline and the dilation to occur along a direction normal to the centreline, the set of parameters will consist of just six transformation design parameters and a smoothing design parameter which can be determined and fixed prior to an optimization study. To manipulate both the boundary curves concurrently, at least 12 design parameters would be needed. Although this number falls within the practical limit of parameters for the efficient use of the Kriging response surface methodology discussed in Chapter 3, there are methods of manipulating the global shape of an airbox with even fewer parameters, for example using NURPS, and so this technique, although practical and straightforward to implement, may not be the most efficient approach for the subsequent studies.

For the representation of a realistic computational human carotid artery bifurcation, geometry data from a real artery is analysed and so the boundary curves can be defined with a high accuracy to give a globally good parametric representation. However, the combination of patches required to model the bifurcation could be problematic. Surfaces are needed to represent the common carotid artery, the internal carotid artery and the external carotid artery separately. Joining these patches requires common boundaries, of which none exist in this case.

Undoubtedly, this is a powerful global manipulation technique and certainly has many advantages for applications regarding automated design optimization. Local manipulation is indeed possible, but not easy and may require a dense collection of boundary curves to capture a particular geometric feature accurately, significantly increasing the parameter count.

5.1.4 Three-Dimensional Hicks-Henne Functions

Following the two-dimensional Hicks-Henne function described in section 2.2.4, a three-dimensional surface bump function is described in the general form by:

$$f(u,v) = h\left[\left(\sin \pi u^{-\frac{log\ 2}{log\ u_p}}\right)\left(\sin \pi v^{-\frac{log\ 2}{log\ v_p}}\right)\right]^T, \tag{5.2}$$

where u and v are the bivariate surface directions ($u, v \in [0,1]$), (u_p, v_p) is the position of the bump peak or the deformation centre, h is the bump amplitude and T is a measure of the bump width. Each sine function can be raised to a different power which would allow for two bump width parameters and result in asymmetrical patches. However, the simplest case of one bump width parameter is pursued here to keep the design variable count to a minimum. Figure 5.9 illustrates a surface patch with a Hicks-Henne bump function of this type. As the bump width parameter or curvature ratio (if the limit curve is normalised) decreases, the bump peak flattens and the deformation stretches further towards the limit curve. A surface representation of decreasing curvature ratio can be seen in Figure 5.10; each bump has a height $h = 1$.

This type of surface patch can be applied in a local sense to a NURPS surface in CATIA and the surface can be deformed automatically via CATIA's visual basic scripting language. A Hicks-Henne patch applied to the baseline airbox can be seen in Figure 5.11. The limit curve of this patch is normalised such that the bivariate surface patch varies in u and v between 0 and 1. For one bump, the limit curve can be drawn anywhere on the original surface. The surface then includes the patch within the area of the limit curve, so that when the patch is deformed, this deformation then becomes part of the surface. The only drawback is that when the surface has already been deformed with one patch, the limit curve has to remain the same and cannot be drawn on an already deformed surface. This is a limitation in CATIA not a limitation inherent to the technique. By retaining fixed limit curve patches, any number of deformations can be performed either in different areas or on top of one another, allowing the locality of the surface manipulation

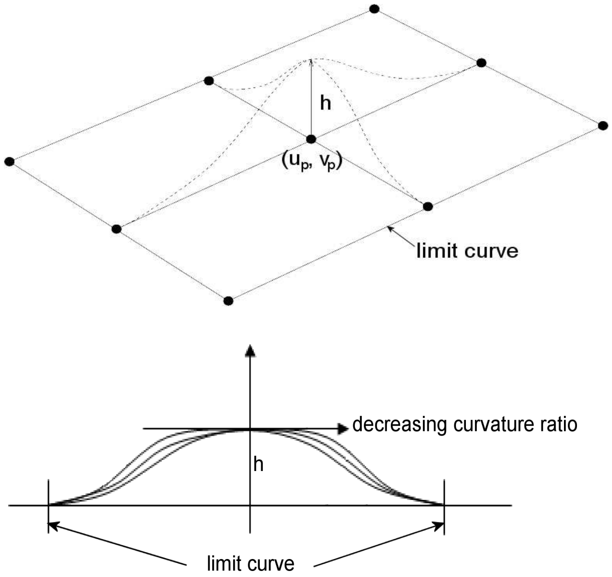


FIGURE 5.9: Hicks-Henne bump deformation patch

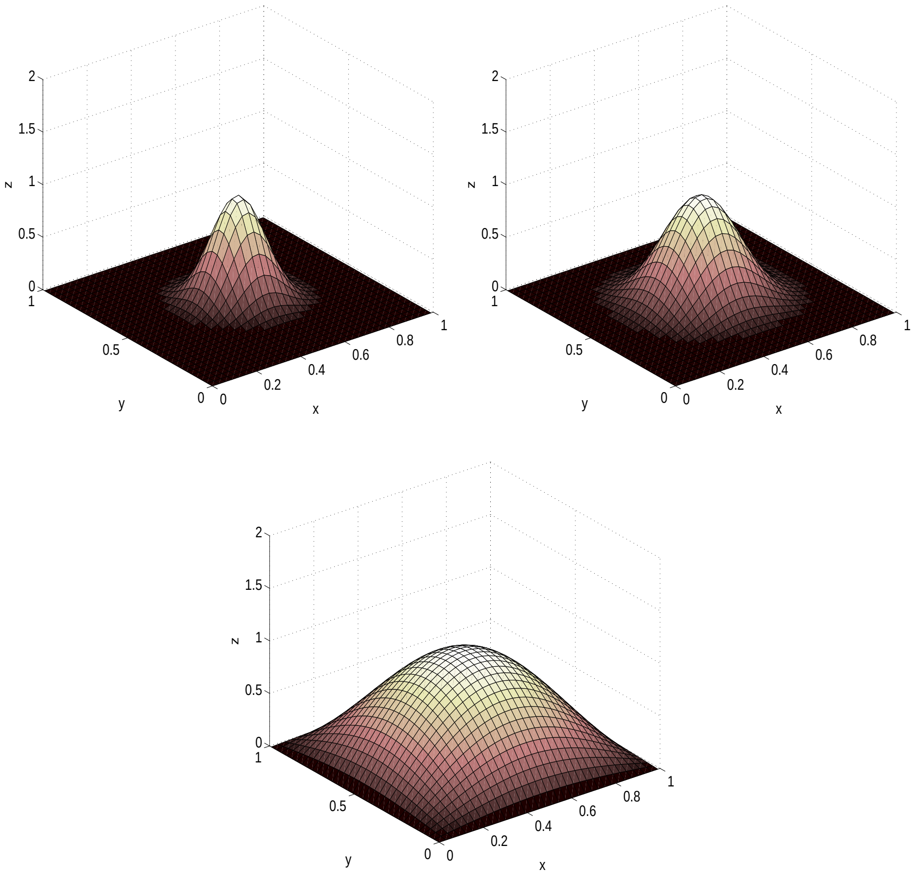


FIGURE 5.10: Hicks-Henne bump surface deformation patch with decreasing curvature ratio

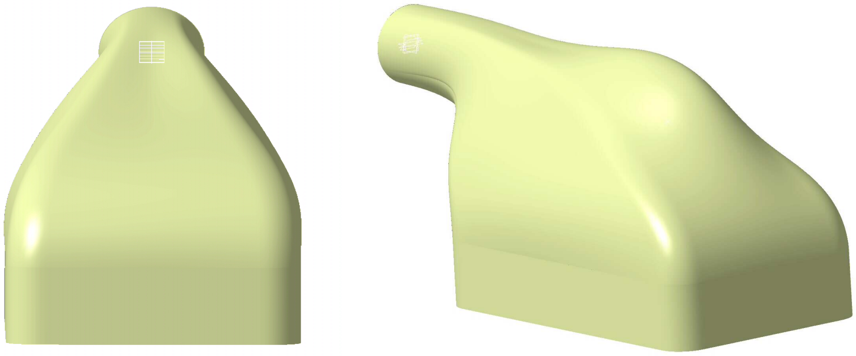


Figure 5.11: Baseline geometry (left) and geometry with Hicks-Henne surface patch (right)

in one direction to be explicitly controlled. The curvature variable controls the locality in the second dimension allowing the construction of ridges of deformations. Obviously this form of surface manipulation can only be applied in a local deformation sense, but it is easily automated and can be defined with only two design variables in addition to the patch boundaries: the bump height and the curvature of the deformation within the limit curve.

5.1.5 Free-Form Deformation

Free-form deformation (FFD) is a subset of the soft object animation (SOA) algorithms used in computer graphics and animation for morphing images and deforming models. This particular method can only achieve modest geometry changes (Barr, 1984; Sederberg and Parry, 1986; Watt and Watt, 1992) and may restrict the shape control of either a global or local manipulation. Sederberg and Parry (1986) presented a technique whereby an object is embedded in a space that is then deformed. This process is analogous to embedding an object in a parallelepiped of clear flexible plastic. In the analogy, the plastic is deformed and the object changes shape.

Lamousin and Waggenspack (1994) modified FFD to include a NURBS definition and multiple blocks to model complex shapes; an example of a NURBS block can be seen in Figure 5.12. The undeformed FFD block consists of a regular lattice of control points arranged along the block coordinate system (u, v, w) which is independent of, but initially parallel to, the object's coordinate system. A detailed description of deformation process is outside the scope of this work.

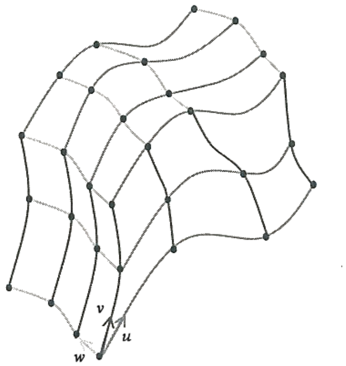


FIGURE 5.12: Trivariate NURBS volume

A disadvantage of these methods is in the indirect control of the deformation through adjusting control points or weights of the embedding volume. It is difficult to get the objects to pass through desired points precisely; for example, in the global manipulation of a CAD artery model to achieve a position of the control points determined by analysing a real artery geometry. Moreover, a large number of control points in complex models makes it impractical to determine the exact number of control points to be changed and how they must be changed to produce the desired deformation. Hsu et al. (1992), motivated by these deficiencies, investigated the direct manipulation of FFD. Given source and target points, their method automatically computes the necessary repositioning of the control points using a least-squares formulation. Although this is a start to providing more practical engineering applications for this method, its primary drawback is that it may fall down when finding the new positions of the control points. Complex, subtle, local deformations are then necessary as the number of FFD blocks required renders this technique unworkable.

5.1.6 Subdivision Surfaces

Geometric representation of three-dimensional models through subdivision surfaces is a relatively new technology. This technique defines a high-resolution polygon mesh that is generated automatically from a low-resolution polygon mesh hull via smoothing algorithms to closely approximate a NURBS surface. The subdivision surface itself can be described as the limit of an infinite number of refinements. They were introduced simultaneously by Edwin Catmull and Jim Clark (Catmull and Clark, 1978), and by Daniel Doo and Malcolm Sabin (Doo and Sabin, 1978). Little progress was made until

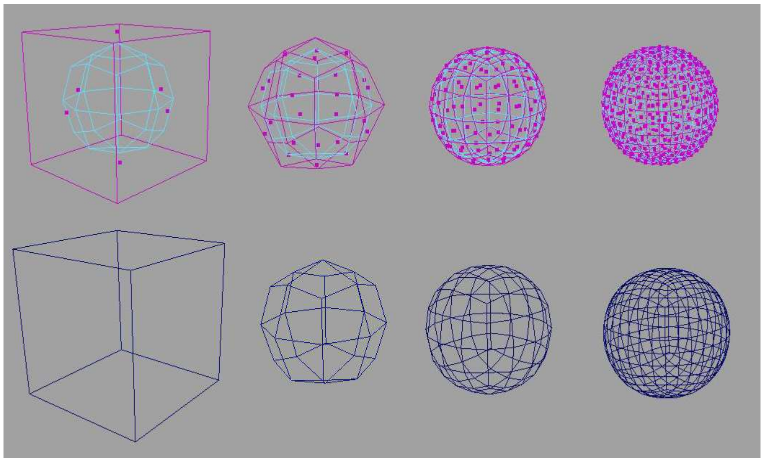


FIGURE 5.13: A cube with three levels of recursive subdivision

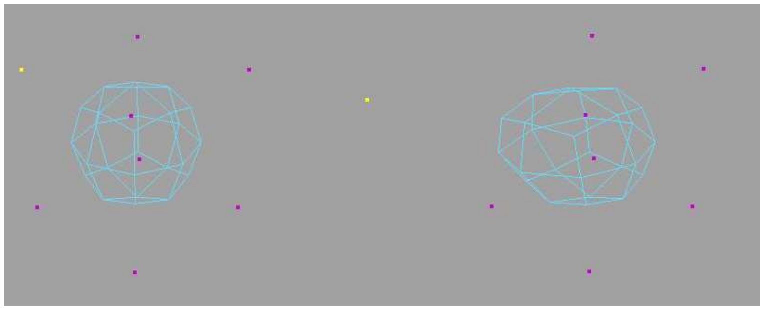


FIGURE 5.14: Control points of step zero representation; the highlighted (yellow) control point is moved from its original position (left) to a displaced position (right)

the 1990s when much research was performed in developing these types of surfaces for use in computer animation. They are now widely used in the computer graphics industry. For example, Pixar Animation used NURBS based software in films such as Toy Story in 1995 and A Bug's Life in 1998 until it was discarded in favour of subdivision schemes for Toy Story II in 1999, Monsters Inc. in 2001 and Finding Nemo in 2003.

An example of the first three steps of recursive subdivision is illustrated in Figure 5.13; in this case the subdivision surface actually represents a smooth sphere, this is shown as the far left image in Figure 5.15. These figures were produced using the animation software MAYA (2005).

Moving the control points of the step zero representation shown in Figure 5.14, where the highlighted control point is displaced, provides a global manipulation of the surface. The deformed surface is shown as the second image from the left in Figure 5.15 and shows the smooth sphere stretched in the direction in which the control point was moved.

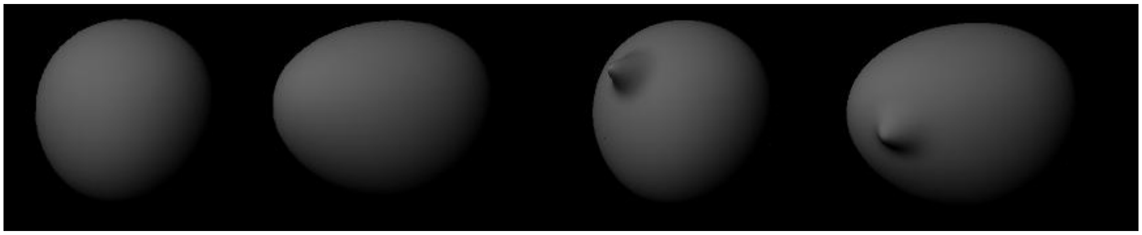


FIGURE 5.15: Rendered surface of the subdivision representation (far left), global deformation of surface from manipulation of step zero control points (second from left), local deformation of surface from manipulation of step three control points (third from left), combined global followed by local deformation of surface (far right)

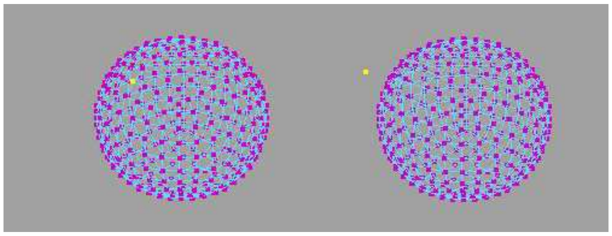


FIGURE 5.16: Control points of step three representation; the highlighted (yellow) control point is moved from its original position (left) to a displaced position (right)

Moving the control points of the step three representation, shown in Figure 5.16 where the highlighted control point is displaced, however, provides a very local manipulation of the surface. This local manipulation can be seen in the third image from the left in Figure 5.15 where only a very small area of the surface is deformed and the global shape remains spherical.

Edits of the control points in both a global sense and a local sense can be combined on the same surface, which can be seen in the far right image of Figure 5.15.

Interestingly, NURBS and subdivision surface representations share a common foundation, the uniform cubic B-spline (see equation 2.5 with n=3). Subdivision surfaces are uniform (usually cubic) B-splines in tensor product regions of the mesh, but they become non-polynomial at extra-ordinary vertices. At these vertices the surface continuity usually drops to first order. However, no explicit manipulation of the control points is necessary to achieve this smoothness and so eliminates the problems of NURBS surface patches highlighted in section 5.1.2.

The main advantages of subdivision surfaces over NURBS modelling are that:

- the refinement process is numerically stable,

- they are simpler to implement, and
- they can be used to describe highly complex topologies and have the ability to provide both global and local refinement.

Since they were developed specifically for the animation industry, subdivision surfaces introduce new problems when used in CAD software that are not present with a NURBS representation. The method is unsuitable for CAD in its present form but with new research, some authors believe that this technology will replace CAD NURBS modeling in the future.

5.2 Multi-Stage Geometry Parameterization

Having discussed some alternative parameterization techniques for global and local surface manipulation, a multi-stage geometry parameterization process for use in conjunction with an optimization process is now presented. The key advantage of a multi-stage parameterization process is to enable optimization of the geometry on both a global and a local level. If a single stage parameterization process could provide both global and local manipulation capabilities from the outset, on an unoptimized geometry using a small set of design variables, then only this stage would be needed to provide an efficient optimization process to determine the optimal geometry. This is not the case with many applications, especially those concerning internal fluid flows. It has already been mentioned that much research has been performed on external body geometries such as aerofoils, and their parameterization techniques are well understood in determining which local manipulations of the geometry will result in desired flow behaviour and performance metrics.

For internal fluid flow applications the designer is faced with myriad different forms of geometries for a wide variety of purposes, many of which have yet to be parameterized suitably for any form of optimization. There is as yet no ‘best practice’ means of parameterizing an internal fluid flow component, either for simple ducts such as those discussed in Chapter 4, or for three-dimensional models.

Against this background, a simple, yet effective, multi-stage parameterization technique has been devised which will cater for previously unparameterized geometries. The first stage parameterizes the geometry very simply, retaining a small set of design variables, and allowing the freedom of large geometry changes in a global sense. The second stage uses a local parametrization technique independent of the first stage, allowing a fine tuning of the global shape to occur. It is important to use parameterization techniques for both stages that are commonly accessible or can easily be implemented in as many of the modern commercial CAD engines as possible, so that industries may be able to implement this process with the software and computing power that is readily available. Having surveyed a number of commonly used techniques of surface representation, the methods best suited for this multi-stage geometry parameterization tool are chosen for use in the design optimization of internal fluid flow applications. These are presented in what follows.

5.2.1 Stage 1

For the first stage, polynomial splines are chosen to define a general global geometry shape. We can easily alter the number of guiding splines and control points through which these splines pass and so they can be applied to any internal fluid flow design problem. The capability of capturing distinctly localised features of the surface may be limited, but can be overlooked given that any local detail of the geometry will be picked up in Stage 2. The bounds of movement of the control points through which the guiding splines pass are straightforward to specify and can be fixed easily if geometric constraints are necessary. Although the PDE approach would perform in much the same way, fewer parameters are needed for the polynomial splines. NURBS surfaces also provide a useful global parameterization technique. For complex models, a large number of control points or a number of NURBS patches may be necessary, in which case any local deformations placed on these surfaces may disrupt the continuity between these patches, rendering this method unsuitable for a multi-stage process.

5.2.2 Stage 2

Stage 2 provides a geometry tuning tool for allowing local geometry features not available from the general modifications allowed through the global parametrization technique using polynomial splines described in Stage 1. This stage can also be described as

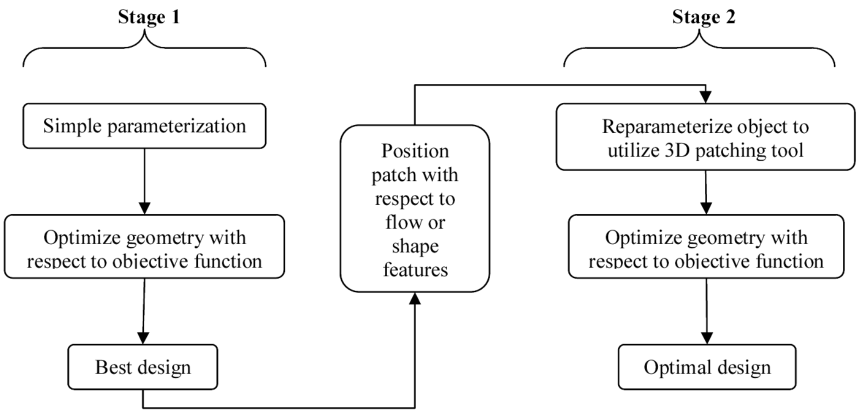


Figure 5.17: Procedure for implementing the multi-stage parameterization technique within an optimization process

providing a correction to the lofted areas generated by the CAD engine between the guiding splines used in Stage 1. The main two contenders capable of parameterizing small local geometry changes are the FFD technique, or the Hicks-Henne bump function. The Hicks-Henne bump patch is easily controlled automatically and uses just two variables once the position of the deformation centre and patch boundary has been determined. FFD, on the other hand, is neither intuitive with respect to its automated manipulation capabilities, nor is it obvious to an optimizer as to which control points of the embedding volume need to be adjusted in order to deform a specific local area of the geometry surface. With this in mind, the Hicks-Henne bump functions are chosen as the surface deformation technique.

5.3 Optimization using AMSSOD

To perform an automated multi-stage shape optimization with deformation (AMSSOD), the parameterization and optimization techniques need to be united to run entirely automatically. Figure 5.17 charts a general outline of the multi-stage parameterization process within a design optimization framework.

The Stage 1 process allows the designer to construct a simple parameterization using polynomial splines and CAD lofting in between these guiding splines to generate the surface. The parametric geometry is optimized with respect to the problem specific objective function, where this optimization process follows the format outlined in Figure 2.1. The best design with respect to the objective function is taken and Stage 2 begins with the reparameterization of the geometry to utilise the bump tool. It is here in the process where the position of the limit curves, which will remain fixed throughout the Stage 2 process, is decided. As these must be fixed, limit curves are chosen to allow any part of the surface to be deformed except along the limit curve itself. Appropriate limit curves are chosen such that they lie along the surface in a position where a deformation would be considered unlikely. For example, they can lie along the optimized splines defining the geometry from Stage 1.

The position of the first local deformation patch must next be determined. Positional determination of deformation patches is not obvious in many cases. If the flow through the application is well understood and certain desired geometric features are required, a simple mapping can be implemented. Knowing how the geometry can be manipulated to achieve the required flow features, the patch can be positioned accordingly and an optimization process including a DoE and updates of the bump height and curvature, outlined in Figure 2.1, performed to find the optimal shape of the deformation.

In many cases, the best positioning of a patch is simply not known. In this case, it is straightforward to automate a DoE of bump positions, heights and curvatures to determine promising positions for a deformation with respect to the problem objective function. Updates are then performed to determine the optimum position and size of deformation. In practice, given the representation of the Stage 1 surface, only the creation of a bump and the alteration of the height and curvature of the deformation via CATIA's visual basic scripting are necessary. The original parameters remain in place but are fixed at the best Stage 1 geometry. Stage 2 can be repeated as many times as necessary to achieve an optimal design.

5.4 Summary

This chapter has reviewed three-dimensional surface representations and manipulation techniques. The most appropriate techniques to offer a global parameterization and a local parameterization have been chosen. For a given geometry, the global parameterization is used to optimize a given metric to obtain a good general shape. Following this, a local parameterization technique is used to deform the generally good surface, fine-tuning the geometry to achieve an improved objective function. The local parameterization process optimizes just two design variables, making this a good tool to provide an efficient optimization process allowing any number of local deformations to be made to the geometry. The multi-stage process is wrapped as an automated process utilising polynomial splines for the Stage 1 global surface manipulation technique, and Hicks-Henne surface patches for the local deformations in Stage 2. The process is implemented for two very different three-dimensional case studies which are presented in the following two chapters.

Chapter 6

AMSSOD Implemented on a Three-Dimensional Airbox

In this chapter, the studies performed on the two-dimensional airbox in Chapter 4 are extended into three dimensions. The design optimization study in two dimensions revealed that, through a parameterization technique allowing strong shape control of the wall, interesting wall shape features are allowed to materialize as the design study converges to an optimum. The optimum shape featured bulges on the walls of the airbox, with the bulges providing regions for the flow to separate and reattach before passing through the filter and into the trumpet tray. These optimal shapes further emphasize the need for a parameterization technique providing strong shape control. This chapter focuses on a design optimization study for a three dimensional airbox and describes the parameterization technique used within the multi-stage process outlined in the previous chapter. The design problem is explained and the results are presented.

6.1 Geometry Parameterization

This study utilises the automated multi-stage shape optimization with deformation (AMSSOD) technique discussed in Chapter 5. The general outline of the process is specialised to suit this particular problem. This airbox specific multi-stage process can be seen in Figure 6.1.

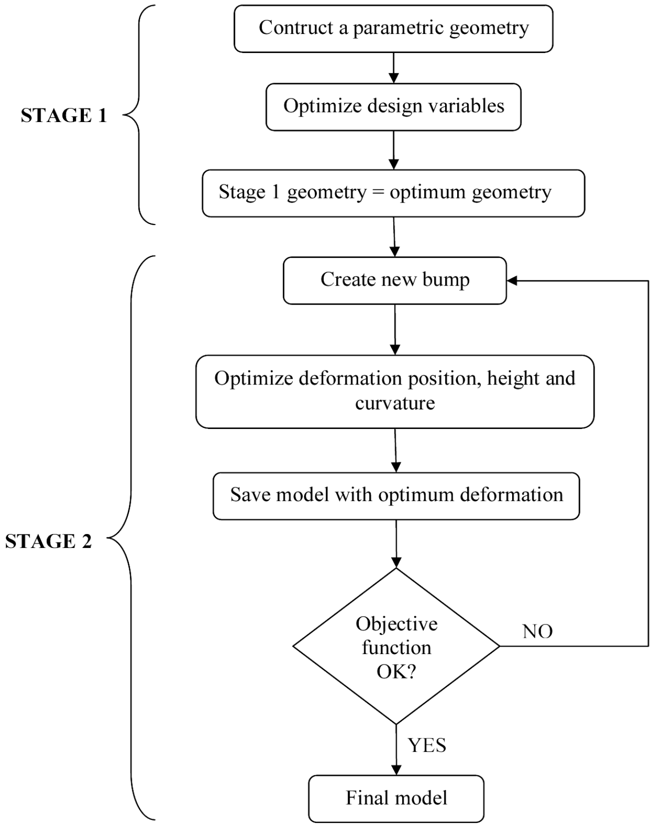


Figure 6.1: AMSSOD process for airbox

The objective function for the optimization processes described in Figure 6.1 is the pressure recovery calculated as the difference between the average static pressure at the inlet and at the airbox filter. The formulation of this objective function is found in section 4.2.

6.1.1 Stage 1

Using polynomial splines to define the Stage 1 geometry, Stage 1 is a simple process of constructing a parametric geometry and optimizing its design variables, as illustrated in Figure 6.1. As discussed in the previous chapter, the idea behind this initial step is to perform an optimization of the global shape to determine a generally 'good' geometry. The airbox is parameterized with four splines: one defining the backbone of the airbox on the upper surface, one defining the centre of the lower surface and two either side of the central lower spline. Each spline passes through two control points positioned between

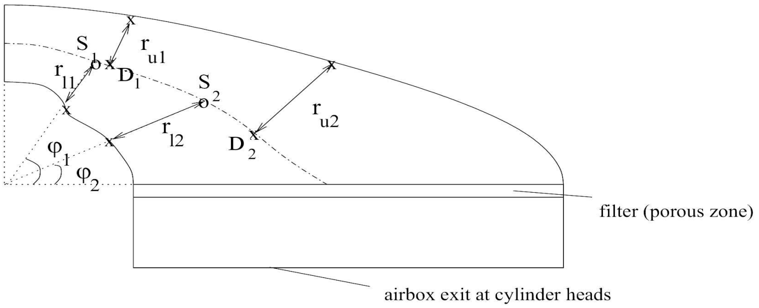


Figure 6.2: Stage 1 parameterization, side elevation (not to scale)

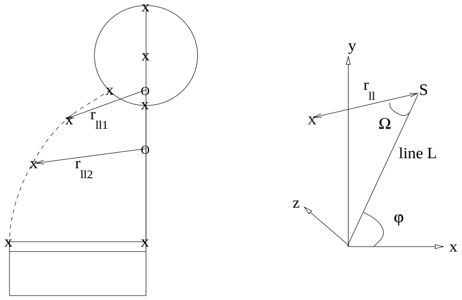


Figure 6.3: Stage 1 parameterization, front elevation (not to scale)

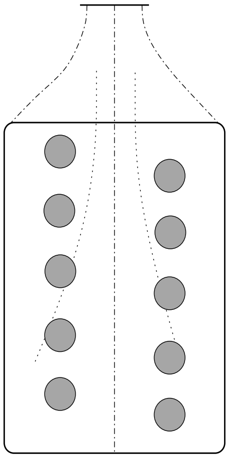


Figure 6.4: Stage 1 parameterization, planform (not to scale)

the entry and the filter. Three of these splines are chosen to define the lower section as, following the two-dimensional study, areas of separated flow forming in the lower section are expected. This allows for strong shape control in a global sense with only eight design variables. To retain maximum control of the overall airbox wall shape, the three splines defining the lower section of the airbox wall are kept completely decoupled from that of the backbone spline. This parameterization is illustrated in Figures 6.2, 6.3 and a planform of the geometry can be seen in Figure 6.4.

Two fixed points, D_1 and D_2, are placed at $\frac{1}{3}$ ratio and $\frac{2}{3}$ ratio, respectively, along the centreline bend. The two points through which the upper spline passes are defined by the distances r_{u1} and r_{u2} measured along a line normal to the centreline bend at D_1 and D_2, respectively. Points S_1 and S_2 along the centreline bend are fixed by the intersection of lines drawn by fixed ϕ_1 and ϕ_2 in the xy-plane at $\frac{\pi}{6}$ and $\frac{\pi}{3}$ respectively. The two points through which the central lower spline passes are defined by the distances r_{l1} and r_{l2}, measured along the lines drawn between S_1, S_2 and the origin respectively, as can be seen in Figure 6.2. From S_1 and S_2, an angle Ω of $\frac{\pi}{6}$ defines the two lines, one on the plane made by the z-axis and the line defined by ϕ_1, and the second on the plane made by the z-axis and the line defined by ϕ_2. This can be seen more clearly in Figure 6.3, where the plane on which Ω is measured also passes through the z-axis and the line L. Along these lines defined by Ω, the distances r_{ll1} and r_{ll2} define the distances to the points through which the lower left spline passes (see the left illustration of Figure 6.3). A similar process is carried out with Ω equal to $-\frac{\pi}{6}$ along which the distances r_{lr1} and r_{lr2} define the points through which the lower right spline passes. These four splines defined by r_{u1}, r_{u2}, r_{l1}, r_{l2}, r_{ll1}, r_{ll2}, r_{lr1}, r_{lr2} make up the eight design variable parameterization of Stage 1.

To simplify this set of parameters further, symmetry in the xy-plane is initially assumed, i.e. $r_{ll1} = r_{lr1}$ and $r_{ll2} = r_{lr2}$, thus reducing the parameter count to six. This requirement is by no means mandatory but, after careful consideration, the choice was made in favour of the smallest set of parameters possible to facilitate a fast optimization convergence towards a good global geometry. This then allows for the Stage 2 process to provide any asymmetrical features that may allow a superior pressure recovery performance. An asymmetry is exhibited in the airbox flow due to the staggered arrangement of the cylinders; an asymmetrical optimal design can therefore be expected.

6.1.2 Stage 2

The Stage 2 process begins by determining the position of appropriate limit curves. Imagine the deformation patch as a piece of cloth and the limit curve the edge of that cloth. This cloth can then be wrapped around the airbox surface with one edge fixed at the inlet and the opposite edge fixed at the filter position. The cloth could be wrapped entirely around the airbox, joining edges along the central lower spline, for example. As

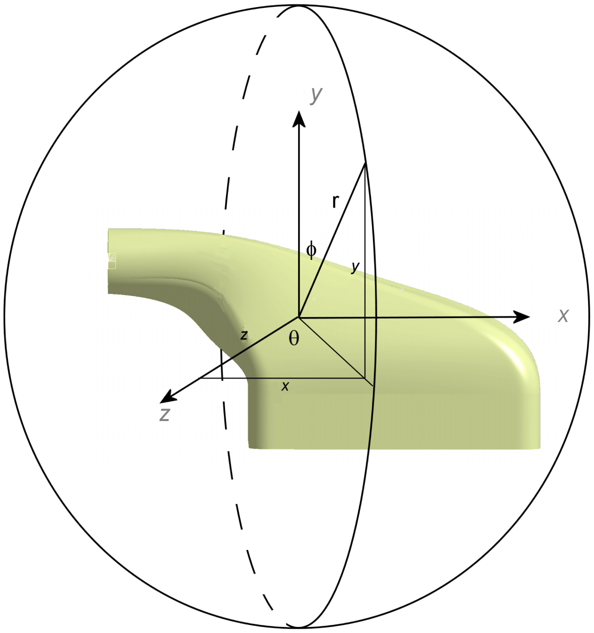


FIGURE 6.5: Spherical polar coordinates defined for airbox

the 2D study presented in Chapter 4 found bulges on the lower wall to be beneficial, the cloth is wrapped around the airbox fixing one of its edges along the lower left spline and the other along the lower right spline. This then determines the limit curve of the deformation patch. A second deformation patch can similarly be wrapped around the remaining lower surface of the airbox. The most appropriate limit curve is employed when the bump position is determined.

A new bump deformation can then be created on the airbox wall. This is achieved by automatically writing a suitable macro which is then run through CATIA. The macro creates a point at the midpoint of the airbox centreline. From here, a point in spherical polar coordinates (r, θ, ϕ) is chosen, where

$$\begin{aligned} x &= r\cos\theta\sin\phi \\ y &= r\sin\theta\sin\phi \\ z &= r\cos\phi, \end{aligned} \tag{6.1}$$

thus describing a point on a sphere, of fixed radius r, encompassing the airbox. $\theta \in [0, 2\pi]$ is the angle in the xz-plane and $\phi \in [0, 2\pi]$ refers to the angle coming out of that plane, as can be seen in Figure 6.5.

This point is projected onto the airbox surface along a direction normal to the surface. By encompassing the airbox within a sphere, the positioning of a point is allowed anywhere on the airbox surface even if deformations have already been placed on the surface. This projected point then serves as the centre for the Stage 2 geometry deformation.

It is not clear exactly where local deformations should be located to improve the pressure recovery. Ideally, a spray of individual local deformations on the surface could be used to develop a mapping between consequential flow features and the position of the deformation. The development of this kind of mapping would neither be trivial nor generic. Although the result may be of use in predicting the areas best suited for deformation, in order to rigorously define a true mapping function a large study of bump deformations may be required. This cannot be generalized to all internal flow ducts and nor could it be generalised to an airbox with varying flow conditions. As it is the aim of this thesis to present an efficient optimization scheme utilizing this two-stage process, the development of such a mapping would be expensive and perhaps superfluous to obtaining an optimum airbox geometry. Hence, to decide the position of the local deformation, a DoE is performed with the positional polars θ, ϕ, the bump height h and the curvature as variables. This spray of bumps over the airbox surface determines favourable locations of deformations. Updates are then performed to obtain the best position and shape of deformation. The model with the optimal deformation is then saved, and the process repeated accordingly.

6.2 CFD Analysis

To ensure that the solutions yield sufficient accuracy, a mesh dependency study on a baseline three-dimensional airbox geometry was performed. The geometry was tested with nine different mesh resolutions ranging from 100,000 cells to 5,500,000 cells (please see Appendix B.3). The comparison of the pressure recovery alone is inconclusive and so the velocity profiles along planar cuts at varying positions through the airbox are considered. This comparison shows that the 500,000 cell study would be an adequate mesh size.

Although this determines the accuracy of the solution output from Fluent, it is yet to be considered as to whether the accuracy of the prescribed CFD boundary conditions in comparison to a realistic flow situation is adequate. A uniform velocity profile fixed at the inlet does not adequately represent the inlet flow situation of a real airbox as there are many factors, in particular the boundary layer development at the inlet lip, which have been ignored. Hence, the flow external to the entire airbox was simulated as well as its internal flow. The baseline airbox geometry was used with a thick surface to simulate effectively the development of the boundary layer on the inlet lip which would affect the velocity profile at the airbox inlet. Using an internal airbox mesh of 1,250,000 cells, the total cell count of the entire domain both internally and externally was just under three million cells. This is considered far too expensive for the current study. Instead, two alternative options were tested. In the first, the velocity profile at the inlet of the simulation with the complete external domain was captured, exported and then imported into a simulation of the internal flow only. This was used instead of the uniform velocity profile originally prescribed for the internal airbox flow. In the second alternative a much smaller box which encased only the inlet was drawn. This gives a truer boundary layer development at the inlet lip of the airbox without a substantial increase in cell count.

For this case study, Stage 1 is performed with these two alternative CFD simulations, the first using the imported velocity profile from the external domain study as the inlet boundary condition, and the second using a reduced external domain around the airbox inlet only. The results of the multi-stage process are presented in the following section. Appendix B gives the details of the mesh dependency study, the results of the CFD simulations of the full external domain and the reduced external flow domain, as well as the simulation with a prescribed velocity inlet condition.

6.3 Results

For a converged solution, all the subsequent studies are performed by solving the RANS equations with a k-ϵ 'realizable' turbulence model, meaning that the model satisfies certain mathematical constraints on the normal stresses, consistent with the physics of turbulent flows. Please refer to the Fluent manual (Fluent$^{\text{TM}}$, 2003b) for further details. Here, non-equilibrium wall functions are used for the near wall treatment. These are different to the 'standard' Launder and Spalding (Launder and Spalding, 1974) wall functions used for the two-dimensional studies in Chapter 4. Its defining elements are that Launder and Spalding's log-law for mean velocity is sensitized to pressure-gradient effects and that it uses a two-layer-based concept to compute the budget of turbulent kinetic energy in the wall-neighbouring cells (Fluent^{TM}, 2003b). Even though an appropriate mesh size was chosen based upon the mesh dependency study performed in Appendix B, a different choice of mesh size or turbulence model could possibly provide different values for the pressure recovery values than those presented here. However, the overall conclusions regarding the geometry parameterization techniques would not be affected.

6.3.1 Stage 1 with Profiled Velocity Inlet Condition

The optimization history for the Stage 1 parameterization for the imported velocity inlet profile case is shown in Figure 6.6. This figure shows the C_p values for design points 1 to 75 representing the initial DoE followed by design points 76 to 175 representing the 100 update points. The bold line indicates the current best design C_p value as each update point is added. In this case the size of the initial DoE is selected based on the rule of thumb of using approximately ten times the number of design variables to produce a reasonably accurate response surface. 100 updates are then performed to converge to an optimum due to the limited computational resources and time allowed. Using just six design variables, each solution using a profiled velocity inlet takes approximately eight hours when run in parallel across two Xeon 2.8GHz compute nodes each with two processors. At this point, it is unnecessary to perform a concentrated exploration in a reduced area of the design space since the Stage 2 process should fine-tune the geometry sufficiently to converge to an optimal design.

The velocity contours for sections through the best design after the first 75 points can be seen in Figure 6.7 along with the wall shear stress showing separation in the y-direction. It is clear that the walls of the airbox contain no unusual bulges akin to those found in the two-dimensional study presented in Chapter 4. This design returns a pressure recovery of C_p=0.8399.

Figure 6.8 shows the velocity contours at each section cut through the airbox in sequence to show the flow developing from the inlet to the filter, through the filter and onto the

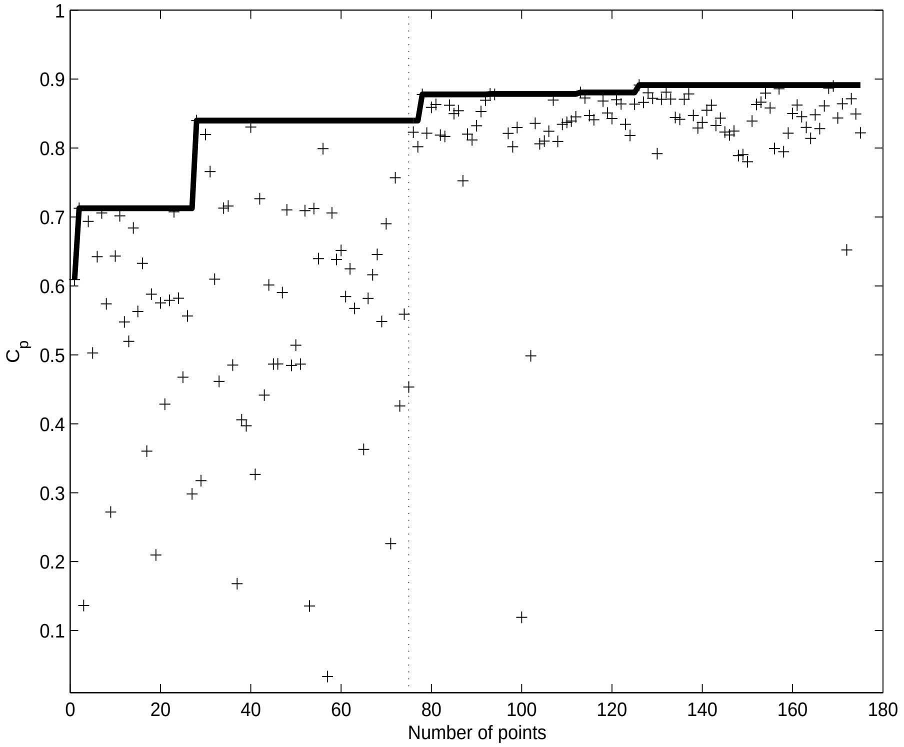


FIGURE 6.6: Optimization history for the airbox with a prescribed inlet profile

cylinders. The legend of contours of velocity magnitude are the same as those shown in Figure 6.7. It can be seen that the flow remains attached along the upper wall throughout the airbox but separates from the lower wall as the airbox begins to turn the flow. As the area of separation is large, a bulge along the lower wall could not be big enough to contain it entirely, and hence this may be the reason why bulges such as those seen on the lower wall of the optimum 2D airbox are not present here.

The optimum design found during the updates returned a pressure recovery $C_p = 0.8915$. Its velocity contours through sections of this airbox design can be seen in Figure 6.9 along with the wall shear stress showing separation in the y direction. Figure 6.10 depicts the velocity contours in each section. The comparison of geometry between the best design after the DoE study and the best design after the updates is shown in Figure 6.11. The upper and lower bounds of each design parameter and their values for the best geometry after the DoE and the final best geometry can be compared in Table 6.1.

Figure 6.11 shows clearly that the best design after the updates has changed the upper wall to accommodate a larger cross-sectional area near the inlet, and a depression towards the exit. These are the largest differences between the variables shown in Table

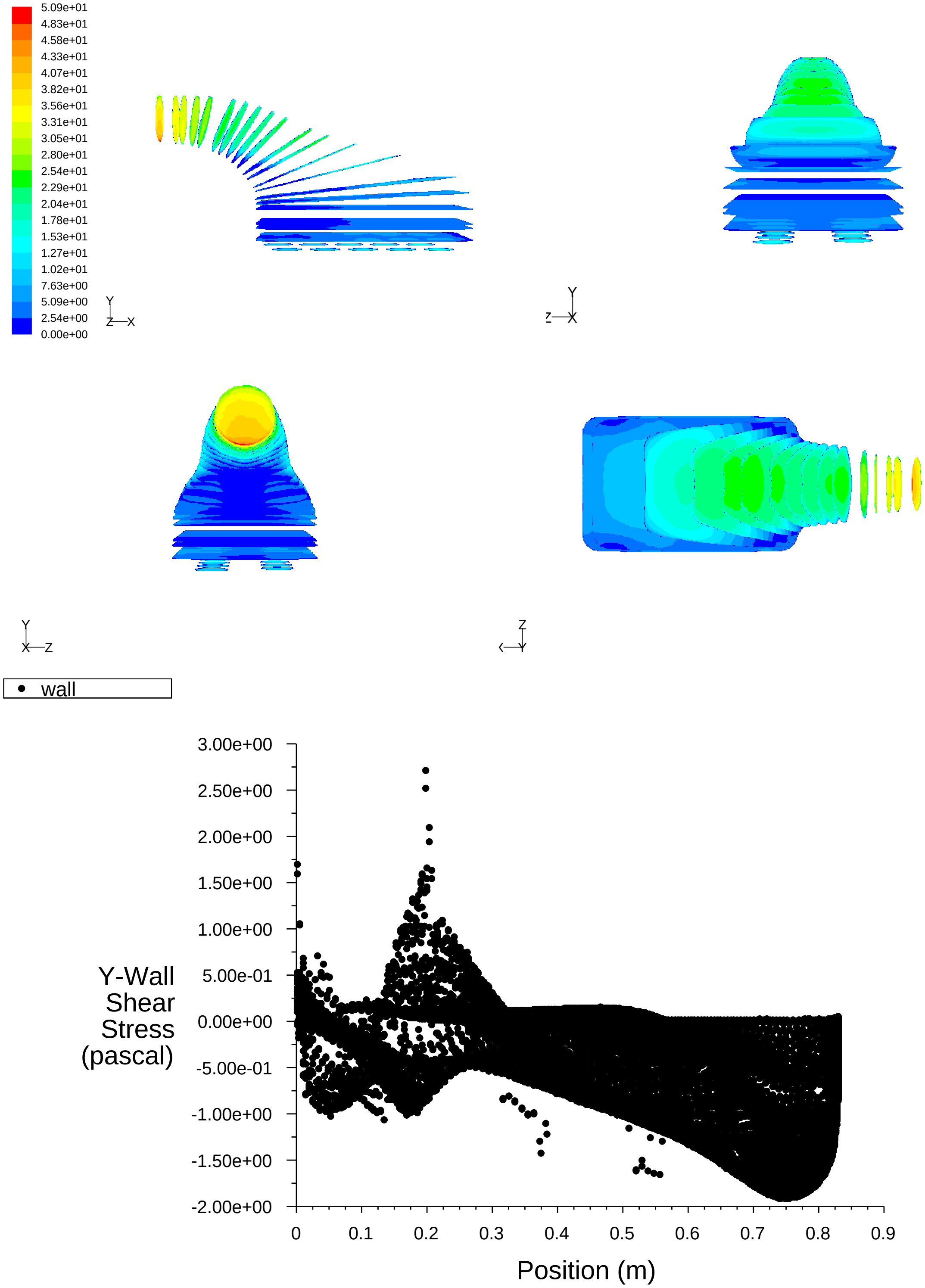


FIGURE 6.7: Contours of velocity magnitude through sections of the best geometry after DoE, C_p=0.8399, and its corresponding wall shear stress shown in the y direction

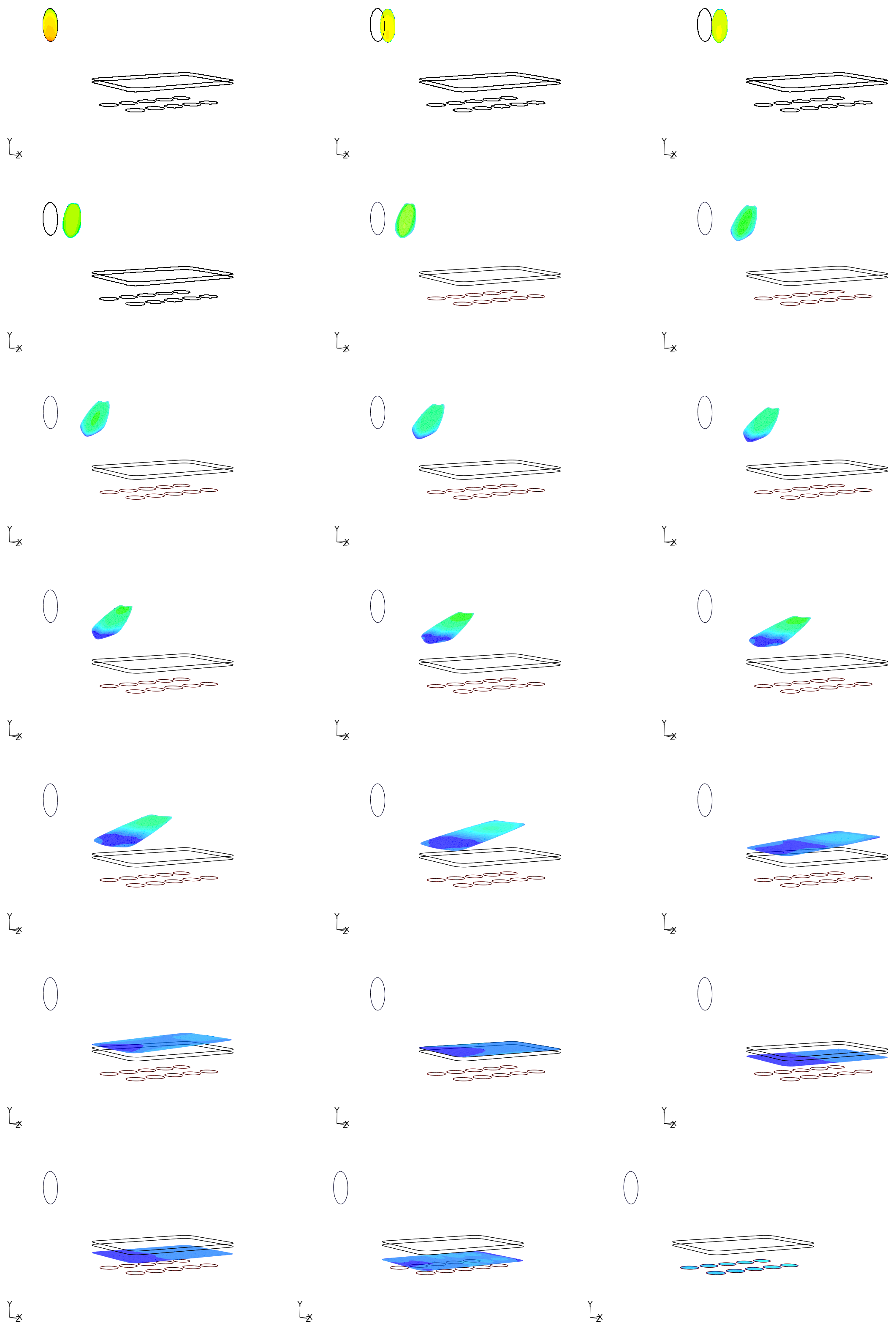


FIGURE 6.8: Contours of velocity magnitude through individual sections of the best geometry after DOE, C_p=0.8399

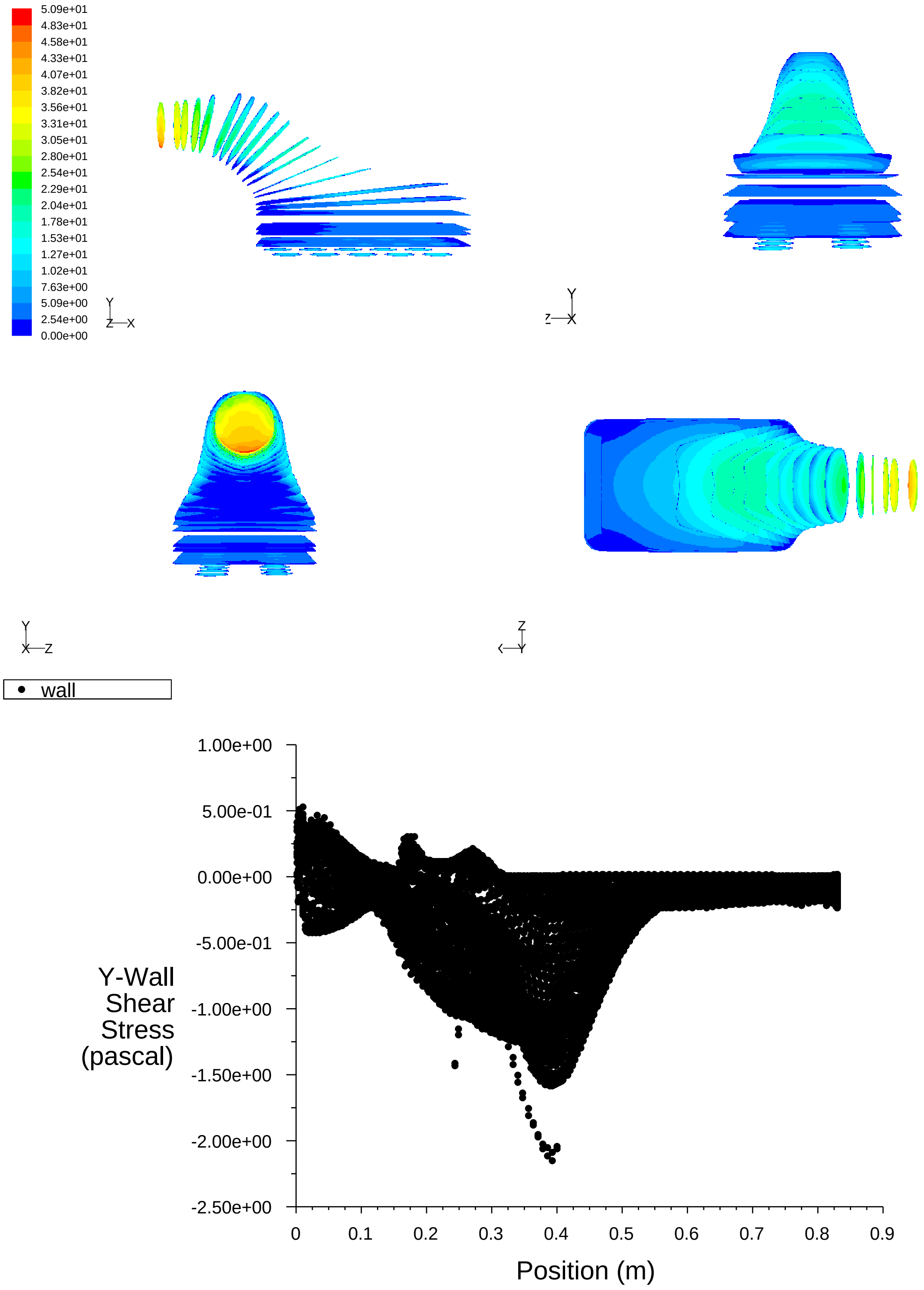


FIGURE 6.9: Contours of velocity magnitude through sections of the best geometry after updates, C_p=0.8915, and its corresponding wall shear stress shown in the y direction

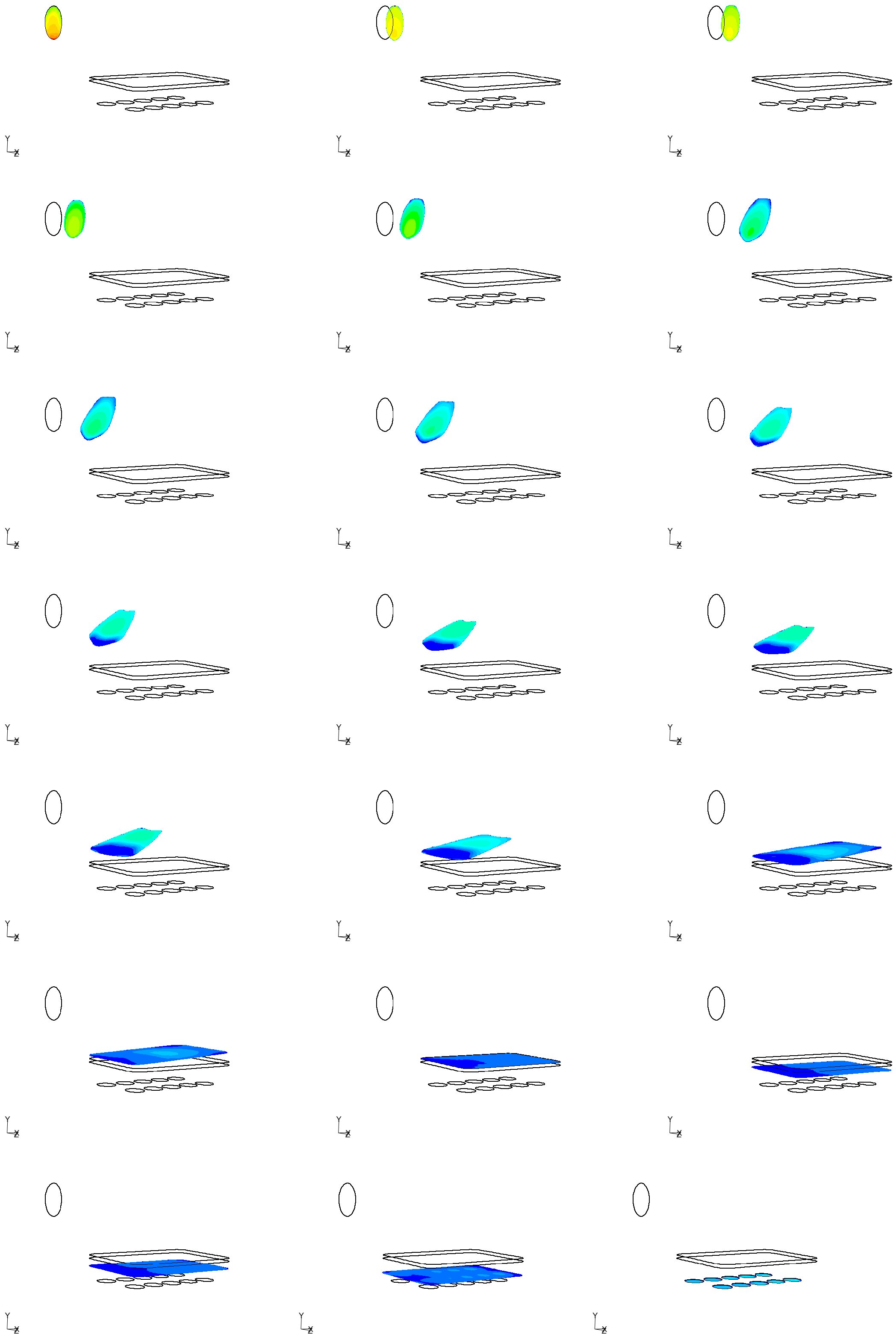


Figure 6.10: Contours of velocity magnitude through individual sections of the best geometry after updates, C_p=0.8915

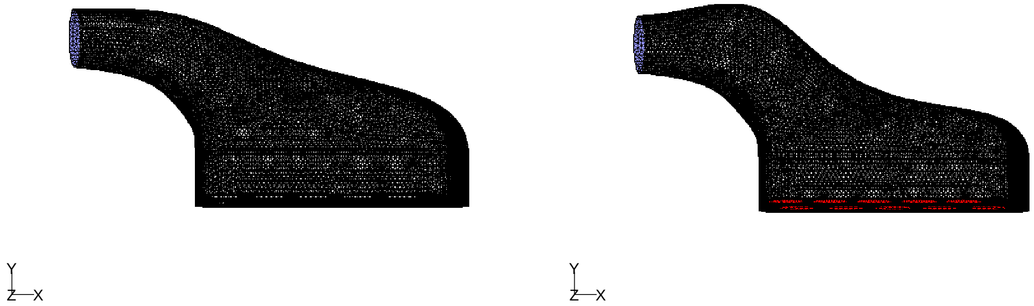


Figure 6.11: Geometry shown for best after DOE, $C_p = 0.8399$, (left) and updates, $C_p = 0.8915$, (right)

Variable	Lower bound (m)	Upper bound (m)	Best after DoE (m)	Best after updates (m)
r_{u1}	0.055	0.1	0.0592	0.0998
r_{u2}	0.1	0.28	0.1281	0.1001
r_{l1}	0.055	0.15	0.1055	0.1058
r_{l2}	0.11	0.2	0.1184	0.1217
r_{lr1}	0.055	0.12	0.1098	0.1069
r_{lr2}	0.12	0.17	0.1309	0.1401
Pressure recovery			0.8399	0.8915

Table 6.1: Design parameters and their corresponding bounds with a comparison of the parameter values for the designs of the best airbox with a profiled velocity inlet after the DoE points and after the completion of the updates

6.1. There is also a slight increase in the lower central wall distance of the second control point, producing a slightly straighter contour shape. Finally, there is an increase in the distance to the second control point along the side walls, also allowing for a slightly straighter wall contour shape. From the illustrations of velocity contours in Figures 6.7 and 6.9, the effect of the bump near the inlet is to create a region of slower flow along the upper wall of the airbox at this point. This coupled with a straighter lower wall shape result in a larger but a more uniform area of flow separation on the lower wall.

6.3.2 Stage 1 with External Domain

The optimization history for the case with a small external domain around the airbox inlet is shown in Figure 6.12. Here, the C_p values are plotted against the number of points, where the first 75 points represent the same DoE as used in section 6.3.1, and the remaining points representing the updates. Using the same six design variables, each

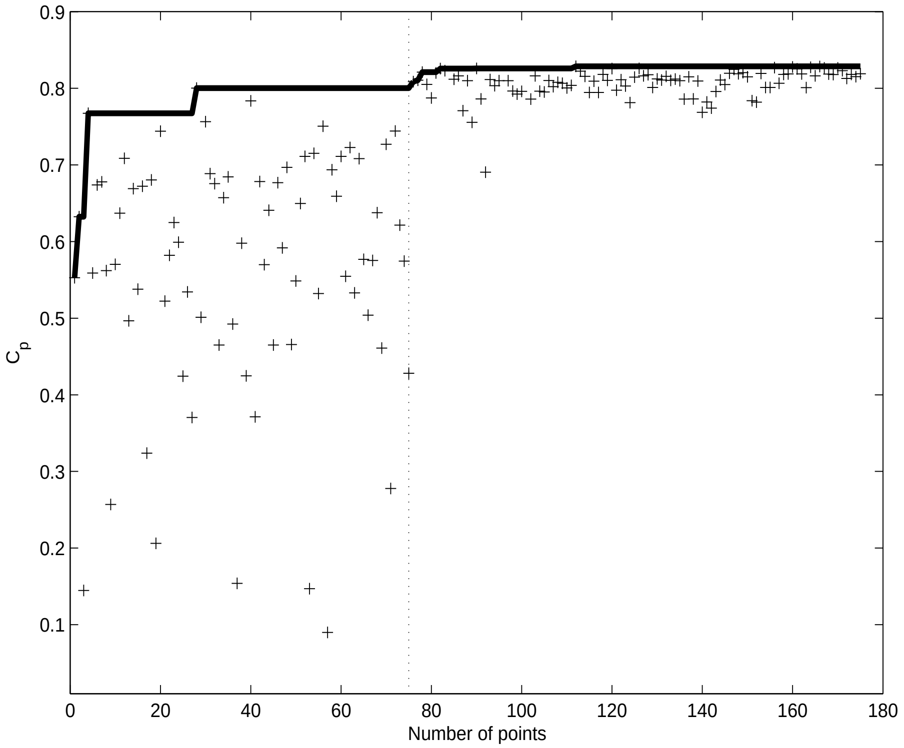


Figure 6.12: Optimization history for the airbox geometry with small external domain around inlet

solution of this airbox with external domain around the inlet lip takes approximately 18 hours when run in parallel across two Xeon 2.8GHz compute nodes each with two processors.

It is clear that there is a difference in solution between these inlet conditions. Interestingly, in the majority of cases, the cases solved with the small external domain around the inlet gave lower pressure recovery values. It can be seen that there are a few geometries returning poor pressure recovery values that performed significantly differently. In these cases, the geometries feature large bulges on either the upper or lower walls. This shows that a small alteration in inlet flow condition could have a large effect on the majority of the airbox flow. In this case, the best design after the initial DoE with the external domain is, in fact, the same as that found after the DoE with the imported velocity inlet profile, point number 28. Due to changes to the flow, the pressure recovery returned is lower; C_p=0.8003. The velocity contours of sections through this airbox along with the wall shear stress showing separation in the y direction are shown in Figure 6.13. There is a slight difference in flow from that seen in Figure 6.7. In particular the increased areas of flow separation along the upper wall by section 14 (counted from

Variable	Lower bound (m)	Upper bound (m)	Best after DoE (m)	Best after updates (m)
r_{u1}	0.055	0.1	0.0592	0.0800
r_{u2}	0.1	0.28	0.1281	0.1270
r_{l1}	0.055	0.15	0.1055	0.1126
r_{l2}	0.11	0.2	0.1184	0.1107
r_{lr1}	0.055	0.12	0.1098	0.0957
r_{lr2}	0.12	0.17	0.1309	0.1271
Pressure recovery			0.8003	0.8288

Table 6.2: Design parameters and their corresponding bounds with a comparison of the parameter values for the designs of the best airbox with a small external domain around the inlet after the DoE points and after the completion of the updates

the inlet), illustrated in Figure 6.14, accounts for the loss in pressure recovery. After the updates, the best geometry had a pressure recovery of C_p=0.8288. The velocity contours through this geometry can be seen in Figure 6.15 along with the wall shear stress in the y direction. The velocity contours in each section are depicted in 6.16, with the comparison of shapes before and after the updates depicted in Figure 6.17.

The design parameter values for the best geometries after the initial DoE and after the updates are be compared in Table 6.2.

From Table 6.2, the lower wall geometry has changed slightly, providing a larger cross-sectional area near the inlet reducing the area of separation on the lower wall. It is also clear that the upper wall shape has changed in a similar way to that of the best geometry found after the updates with the imported velocity inlet profile. Due to the bump on the upper wall, the onset of separation is delayed along the lower wall, and a reduced area of separation along the upper wall is seen near the filter. The right and left lower walls also experience a decrease in distance from the centerline, creating a straighter airbox wall shape near the inlet but still bulging out slightly towards the filter. The difference between the distances of the two control points along these lower side walls are roughly the same as those of the case with an imported velocity inlet profile.

From the Stage 1 studies, it can be concluded that although the presence of the external domain may provide more accurate solutions, here the optimization process returns the same best geometry after the DoE and the best geometries present similar features after the update process, i.e. the design trends, which are crucial to provide a meaningful

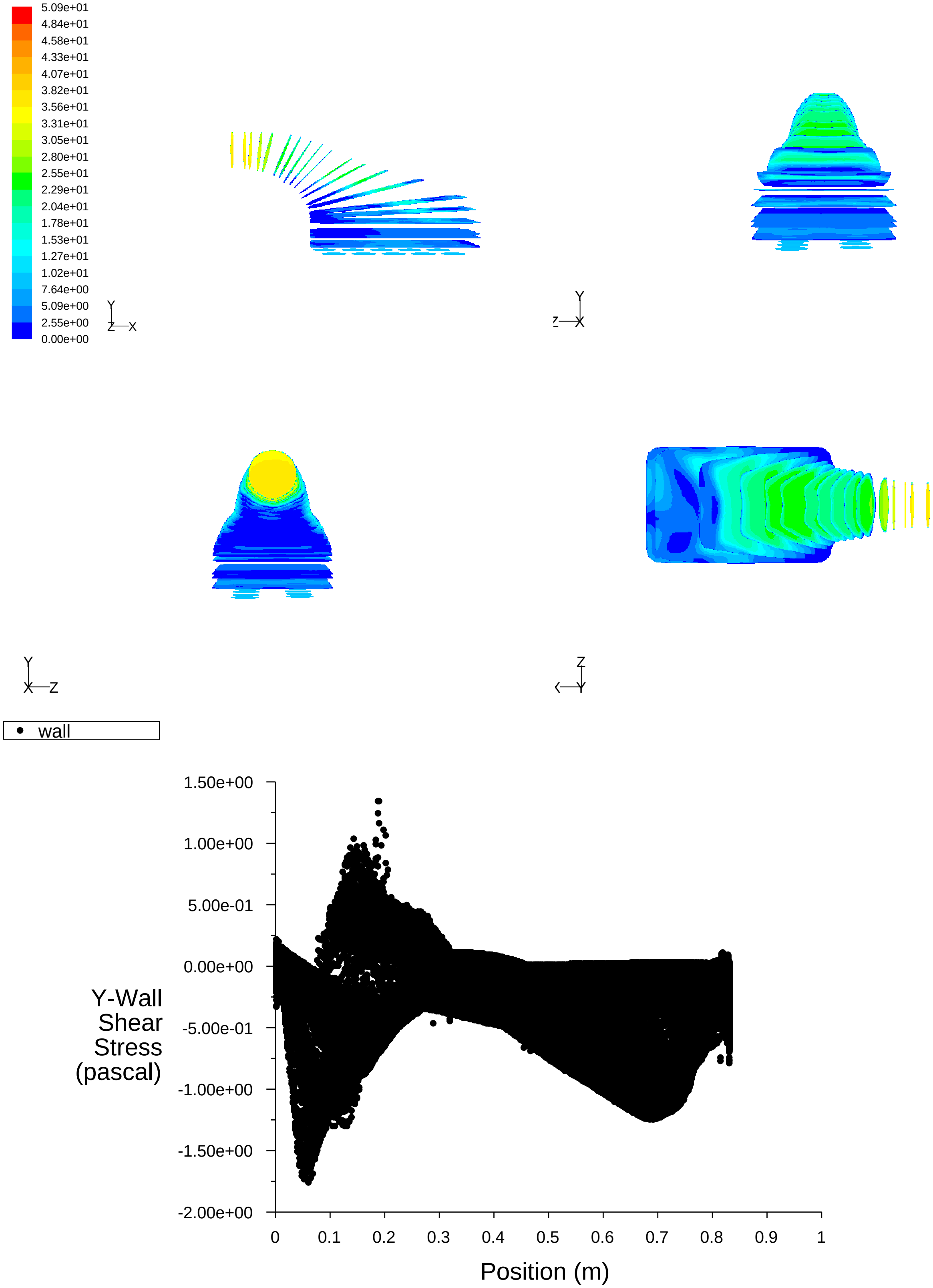


FIGURE 6.13: Contours of velocity magnitude through sections of the best geometry after DoE, C_p=0.8003, and its corresponding wall shear stress shown in the y direction

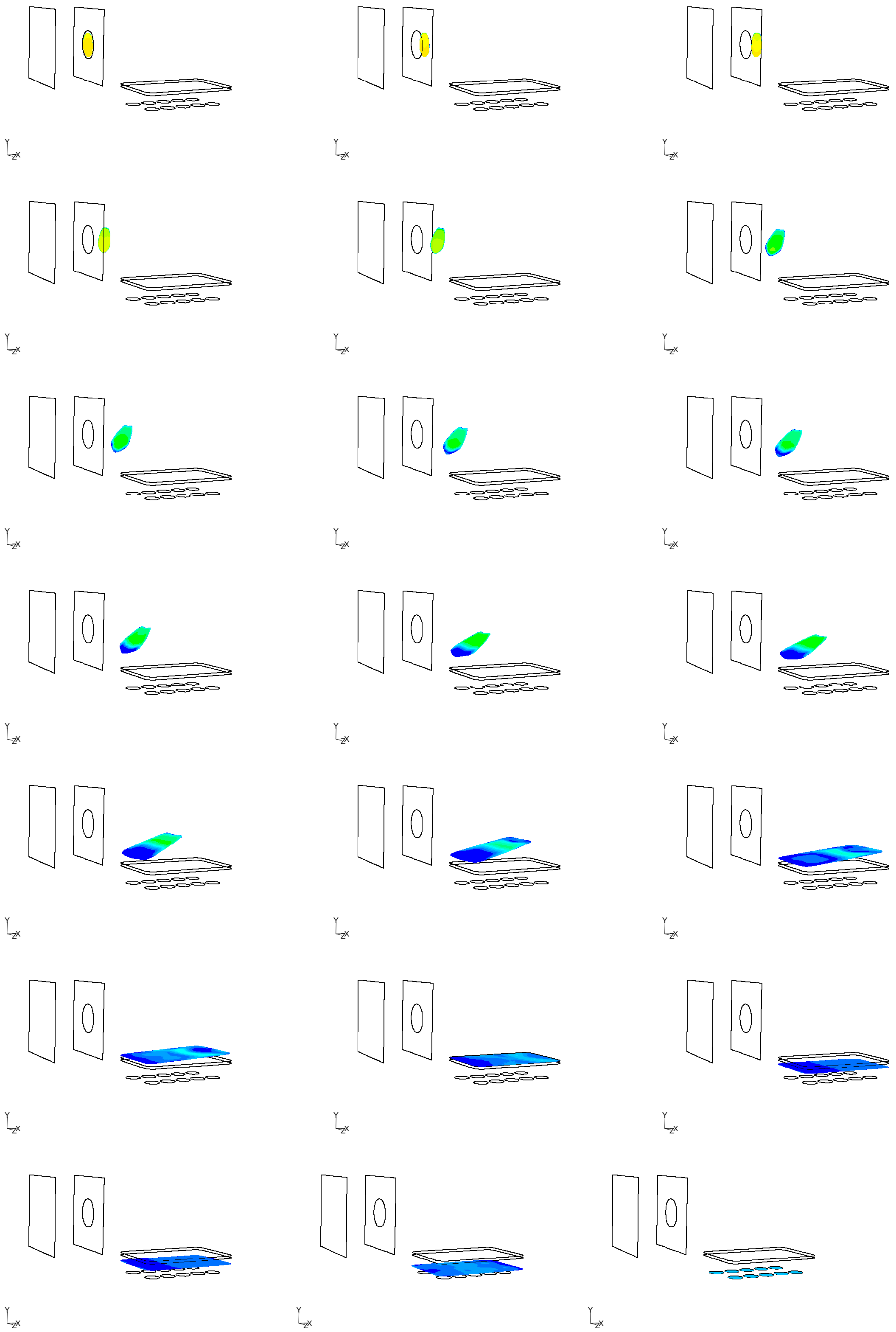


FIGURE 6.14: Contours of velocity magnitude through individual sections of the best geometry after DOE, C_p=0.8003

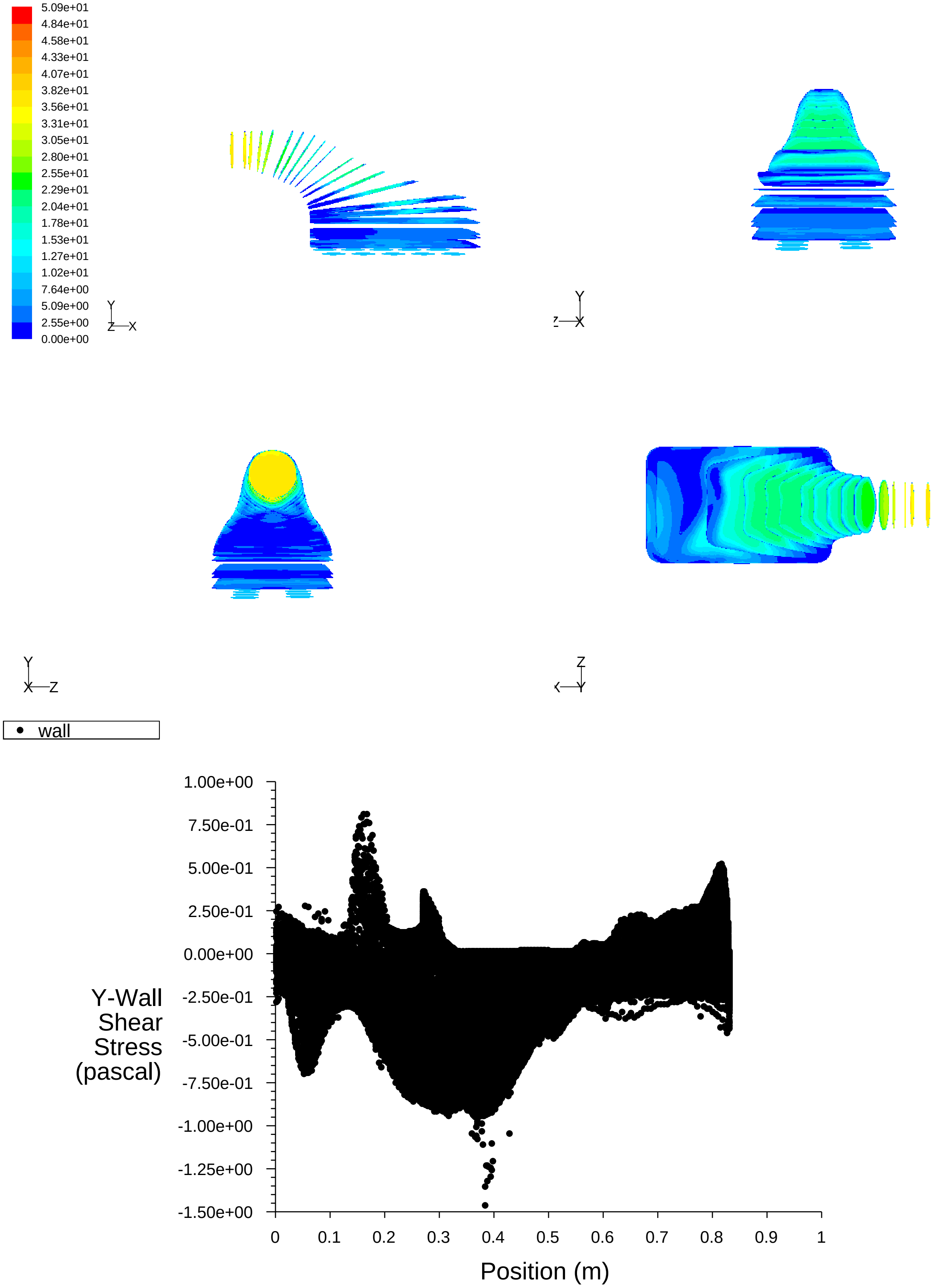


FIGURE 6.15: Contours of velocity magnitude through sections of the best geometry after updates, C_p=0.8288, and its corresponding wall shear stress shown in the y direction

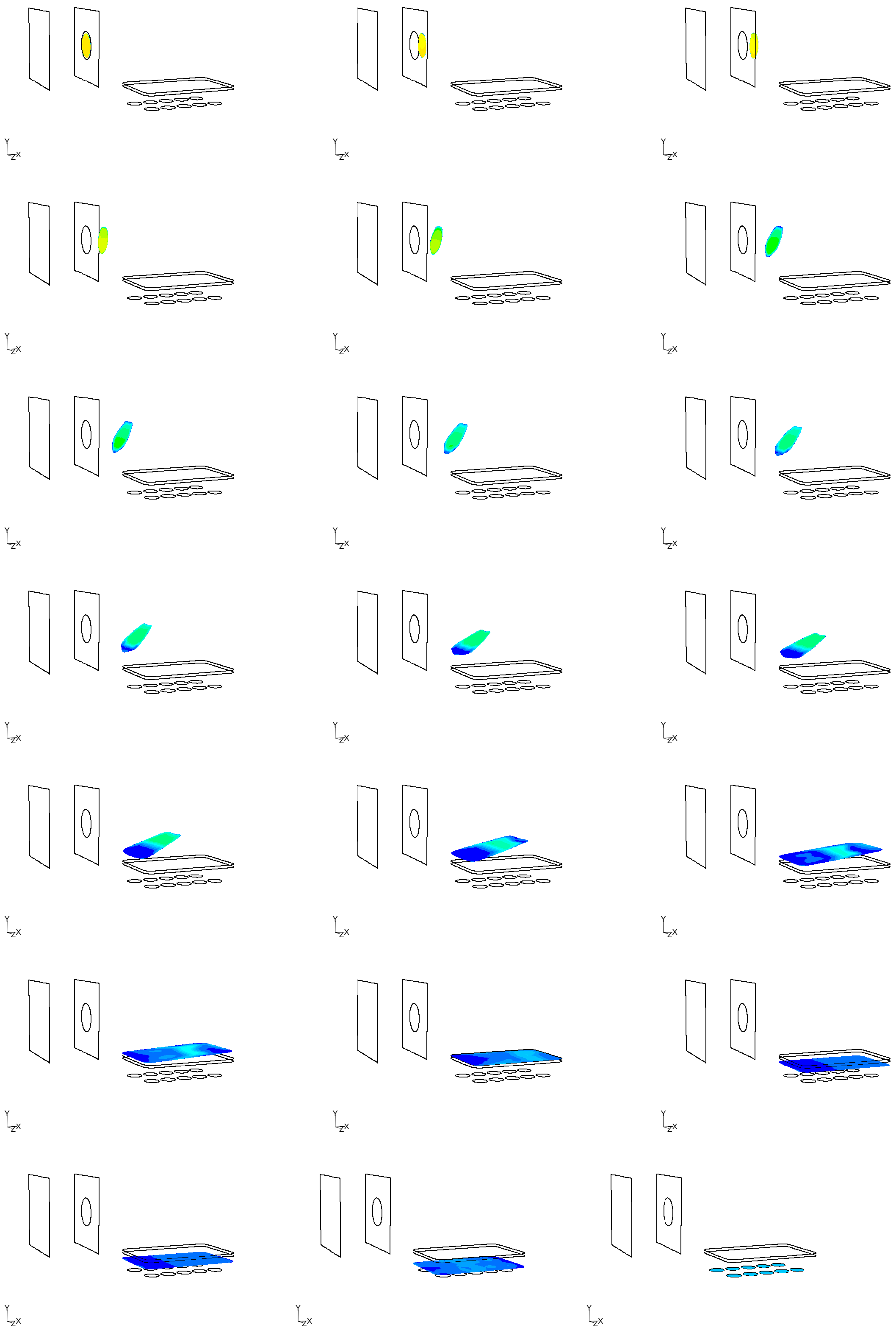


FIGURE 6.16: Contours of velocity magnitude through individual sections of the best geometry after updates, C_p=0.8288

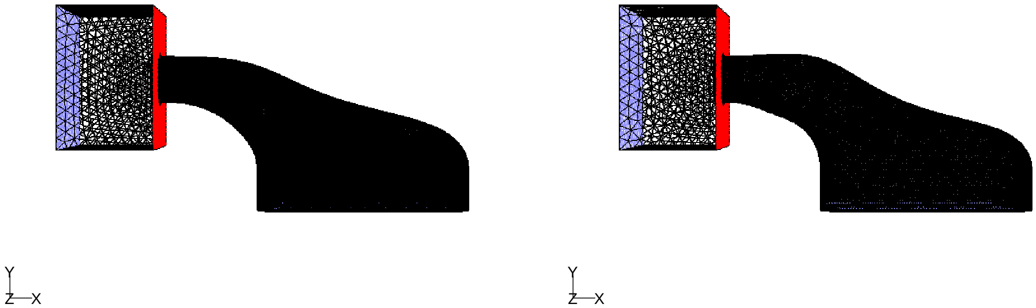


FIGURE 6.17: The best geometry after the initial DOE, $C_p = 0.8003$, (left) and updates, $C_p = 0.8288$, (right)

design search, are the same. In the Stage 2 studies, only the profiled velocity inlet case is used due to the faster run times.

Interestingly, during the search for a globally good geometry no good geometries featuring large bulges on either the upper or lower walls, akin to those seen in the two-dimensional case, are found in this study. Although the reduced number of design variables would allow for one bulge to be generated on either wall, the appearance of two bulges is not possible due to the reduced number of design variables and, thus, the reduced shape control.

6.3.3 Stage 2

The best geometry from Stage 1, using the profiled velocity inlet case, is used as the input to Stage 2, as shown in Figure 6.1. A DoE of bump position ($\theta, \phi \in [0, 2\pi]$), height ($h$(m) $\in [-0.03, 0.05]$) and curvature ($\in [0, 1]$) on the best design from Stage 1 is analysed using the same objective function. A 50 point DoE was followed by 30 update points and the optimization history can be seen in Figure 6.18 where these points are shown by $*$ following the Stage 1 process.

The best geometry returned a pressure recovery of $C_p = 0.8903$ with the positional variables $\theta = 33.75°$, $\phi = 168.75°$, height $h = 0.0375$m and curvature=0.41. This design can be seen in Figure 6.19 with the bump deformation encircled. Its velocity magnitude contours are also shown in Figure 6.20 along with the wall shear stress in the y direction.

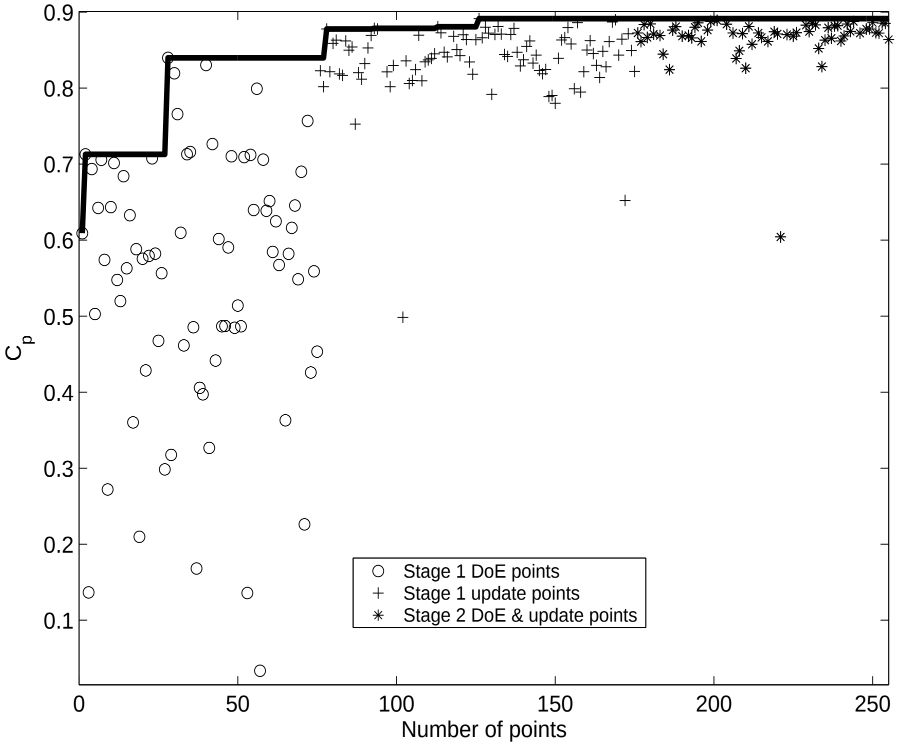


Figure 6.18: The optimization history of Stage 2 following the Stage 1 update points

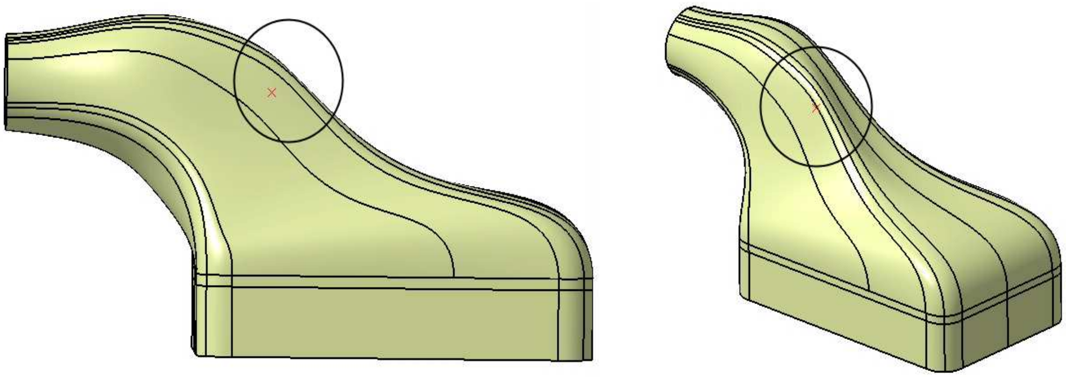


Figure 6.19: Best design after the Stage 2 updates with the bump deformation encircled, $C_p = 0.8903$

The velocity magnitude contours through each individual section are depicted in Figure 6.21.

Here the bump is positioned on the upper surface of the airbox wall just to one side of the spline defining the backbone of the airbox in the Stage 1 parameterization. Although the contours are very similar in comparison to the velocity contours shown for the best

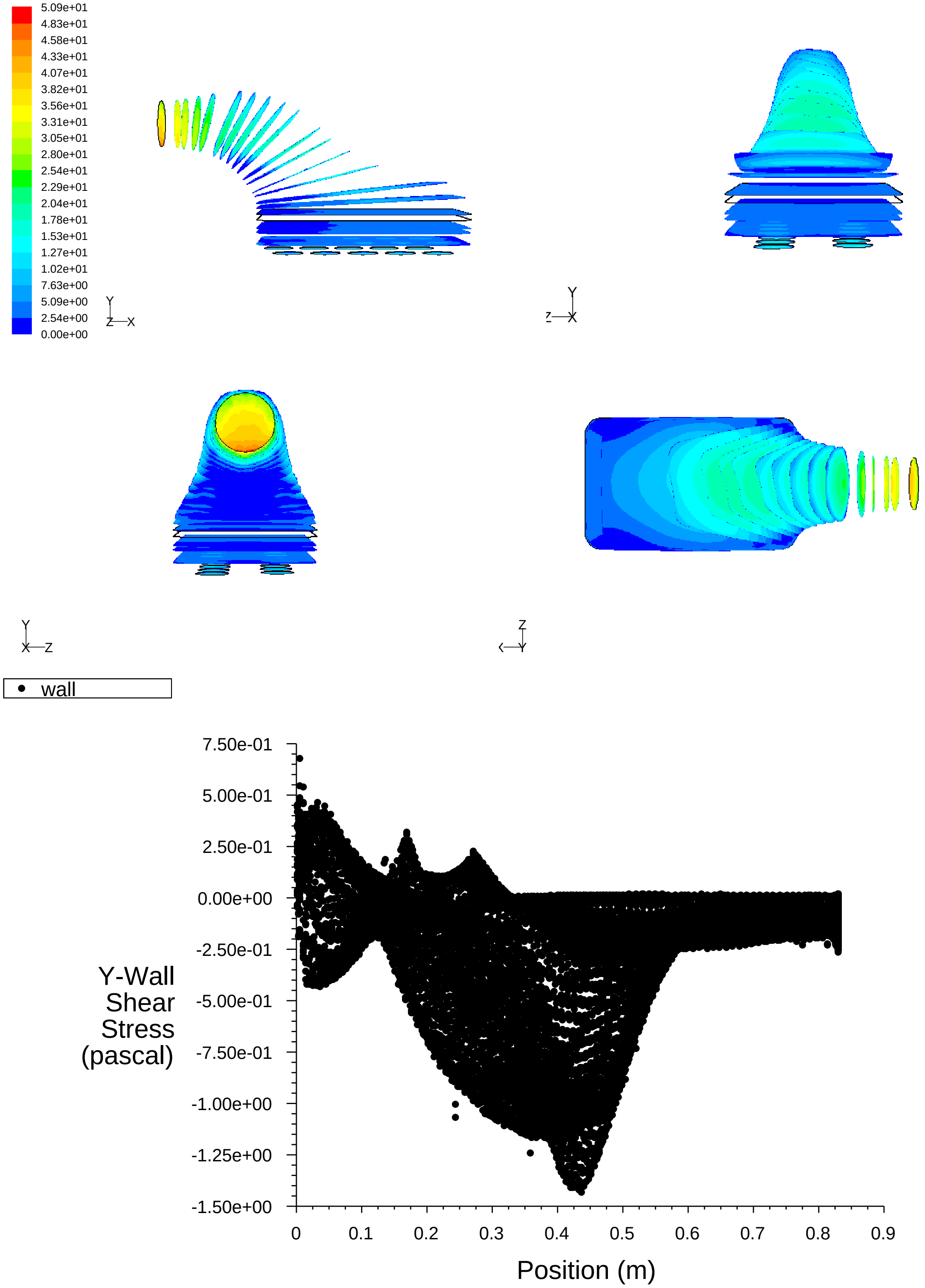


Figure 6.20: Contours of velocity magnitude through sections of the best geometry after Stage 2, $C_p = 0.8903$, and its corresponding wall shear stress shown in the y direction

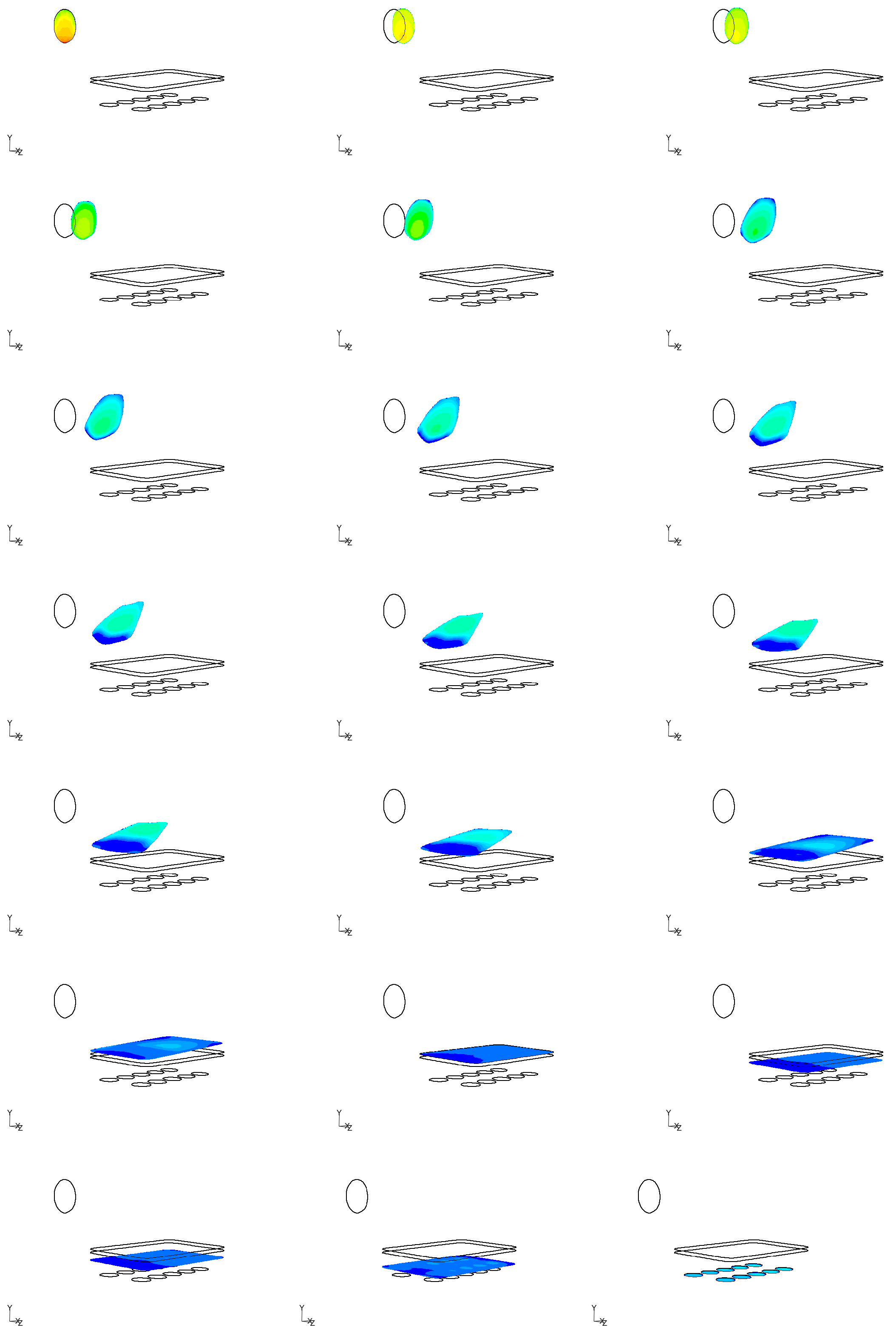


FIGURE 6.21: Contours of velocity magnitude through individual sections of the best geometry after Stage 2, $C_p = 0.8903$

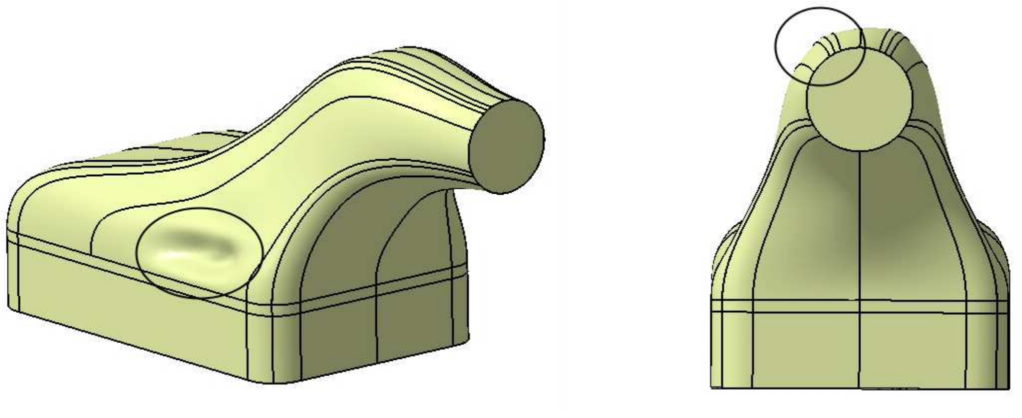


Figure 6.22: Designs found during Stage 2 process with $C_p = 0.8876$ (left) and $C_p = 0.8878$ where the bump deformations are encircled (right)

geometry after the Stage 1 updates (Figures 6.9 and 6.10), it can be seen that the bump causes a slight bias in the flow separation and from section 8 through to section 14 (counting from the inlet) a slightly larger area of separated flow is apparent.

The process tested here has allowed the position of the bump as well as the height and curvature of the deformation to be tested together. The initial DoE returns promising locations of bumps that may be favourable and the updates allow the bump to be moved around these points to find the optimum bump position, height and curvature. In this case, the DoE returned a number of good designs with bumps returning pressure recoveries greater than $C_p = 0.88$ and where the bumps were placed in quite different areas. Two such bumps can be seen in Figure 6.22 both with similar pressure recoveries of $C_p = 0.8876$ for the design shown on the left and $C_p = 0.8878$ for the design shown on the right.

The design on the left shows the bump deformation clearly as an indent on the left hand side of the airbox, whereas the design on the right shows the bump deformation as an indent on the upper wall left of the spline defining the backbone. Both of these designs show favourable areas as to where a deformation may improve the pressure recovery but in very different positions. Hence, the optimizer tested areas around both deformation positions determining whether the design is improved towards a local optimum or in fact the global optimum. Because of this, the Stage 2 updates could not locate a design with a bump which had a greater pressure recovery than that of the best geometry of Stage 1 with $C_p = 0.8915$.

This suggests that perhaps the Stage 1 parameterization had too much shape control and allowed the optimizer to find a near optimal design during the Stage 1 updates. It may also suggest that one particular favourable area of improvement, such as the position of the bump which returns the highest pressure recovery in the initial DoE, may need to be chosen to allow the search to concentrate in this area to find an improvement instead of over the whole surface. To locate other areas where bumps are favourable, the Stage 2 process can be repeated accordingly.

From analysing the variables for each of the 30 update points as they were added to the RSM, it is clear that the updates are focusing on a specific promising area where the height variable is being altered but is very close to zero. This can be seen in Figure 6.23 where the variables θ, ϕ, and h chosen for each update point are shown. This suggests that the optimization process may be driving the bump height down to zero as this is where the optimum geometry (the same geometry found in the Stage 1 updates) is to be found. If this is the case, it implies that the Stage 1 parameterization was too good, providing too much shape control, allowing the optimum geometry to be found. In hindsight, the Stage 1 process should have been performed on a geometry with even fewer design variables in its parameterization to allow the Stage 2 process to work to its best ability, i.e. by locating and fine-tuning the upper surface bump found here in the Stage 1 updates.

To test the Stage 2 process further and to allow for one bump to be placed and optimized in a favourable position, the location of the bump can be fixed after an initial DoE in Stage 2 with the positional and bump height variables to be tested. The bump height and curvature of the deformation can then be optimized at this point to improve the pressure recovery. Given that the geometry found after the Stage 1 updates may be a near optimum design, this modification to the Stage 2 process is tested on a known non-optimal geometry: the best design found after the initial DoE study in Stage 1 for the profiled inlet case with $C_p = 0.8399$.

The optimization history of this can be seen in Figure 6.24. Here, the red dashed line indicates the improvement in design after the initial DoE points of Stage 1 (shown as $\circ$).

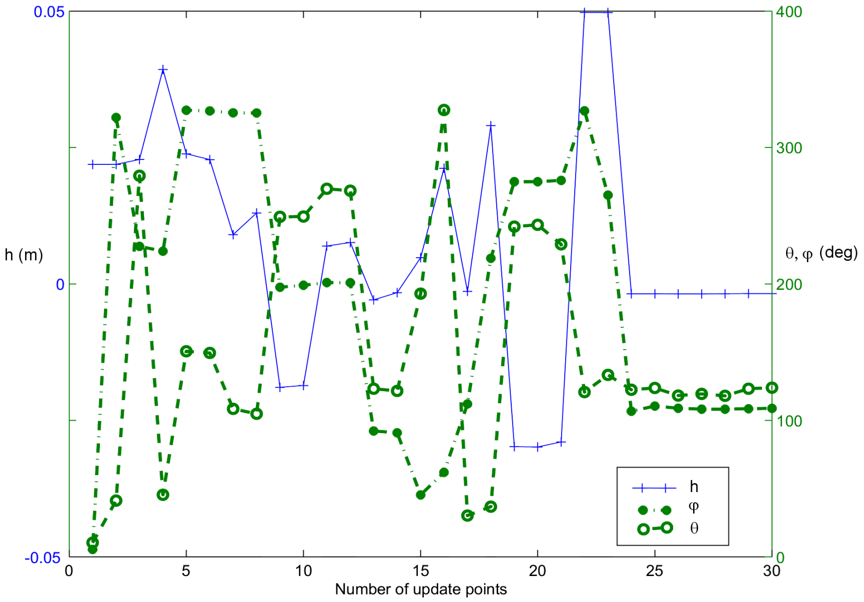


FIGURE 6.23: History of variables θ, ϕ(deg) and h(m) for each of the 30 update points

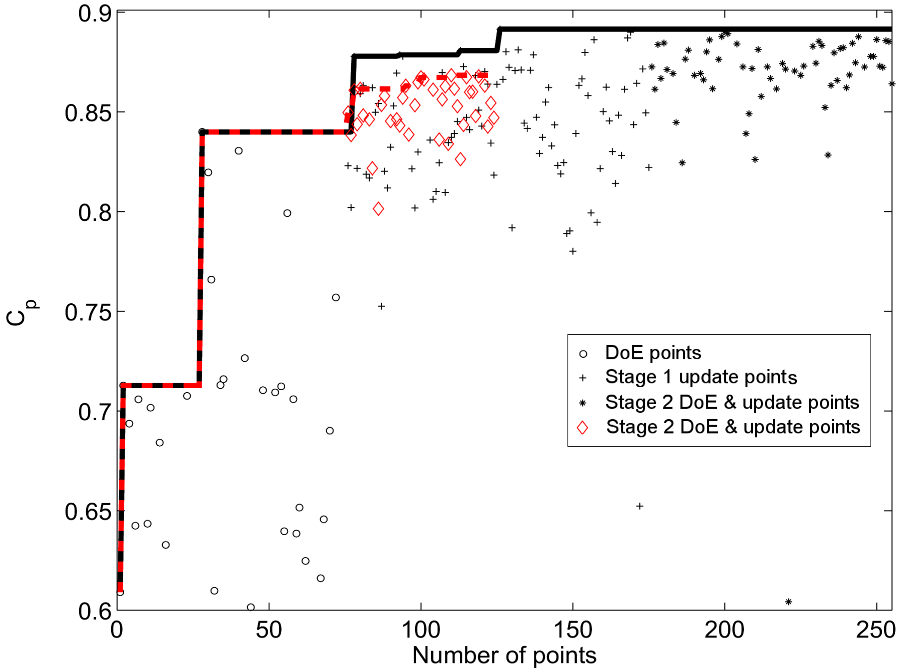


FIGURE 6.24: The optimization history of Stage 2 after the best geometry found during the Stage 1 DoE (♢) and after the best geometry found during the Stage 1 updates ($*$)

The black line in this figure indicates the progress of the Stage 1 update points (shown by $+$) followed by the Stage 2 process that was performed upon the best geometry found after the Stage 1 updates (shown by $*$).

The best improvement in pressure recovery in the initial 30 point DoE on the three

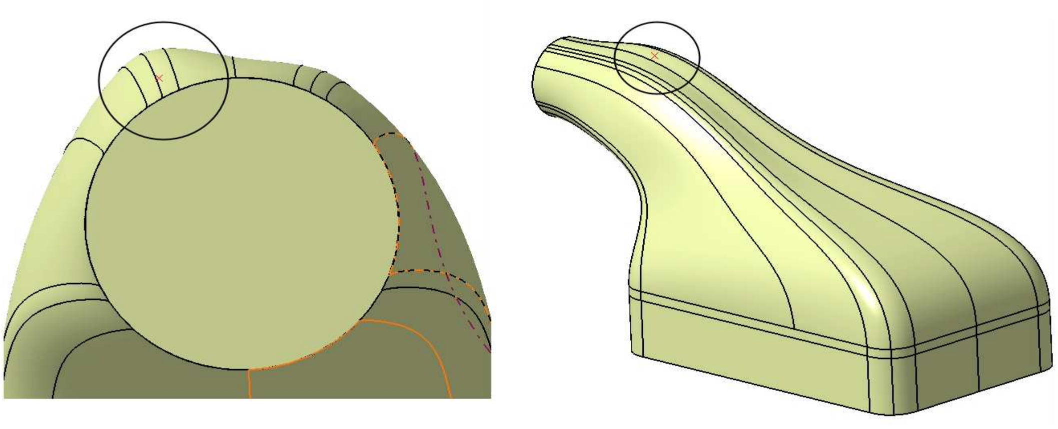


Figure 6.25: Best design after the Stage 2 updates performed following the Stage 1 DoE, with the bump deformation encircled, $C_p = 0.8684$

variables $\theta, \phi \in [0, 2\pi]$ and bump height $h(\mathrm{m}) \in [-0.05, 0.1]$, is found with a bump located at $\theta = 22.5°$, $\phi = 337.5°$ and height $h = 0.0531\mathrm{m}$. This gives an improvement from $C_p = 0.8399$ to $C_p = 0.8673$. Continuing the process shown in Figure 6.1, a DoE of bump height and curvature, using this position as the centre of the deformation, is then performed. After 10 DoE points and 10 update points, the optimal deformation size is found to be $h = 0.00125\mathrm{m}$ and curvature $= 0.63$. This gives a pressure recovery of $C_p = 0.8684$. This geometry is seen in Figure 6.25 with the velocity contours shown in Figure 6.26 along with the wall shear stress showing separation in the y direction. Figure 6.27 depicts the velocity magnitude contours at varying sections through the airbox.

The velocity contours show a reduced area of separation from section 7 (counted from the inlet) in comparison with the best geometry after the Stage 1 DoE (the velocity contours through which are shown in Figures 6.7 and 6.8).

From Figure 6.24, it is clear that the Stage 1 updates perform better in terms of finding a geometry with a higher pressure recovery more quickly. The best geometry after this Stage 2 process can be seen in Figure 6.25 and the best geometry found after the same number of points during the Stage 1 update process can be seen in Figure 6.28. In comparison, although the initial Stage 2 DoE has found a favourable position close to the backbone spline, as the centre of deformation has been fixed, the centre of deformation could not move towards the backbone spline to return a better pressure recovery value.

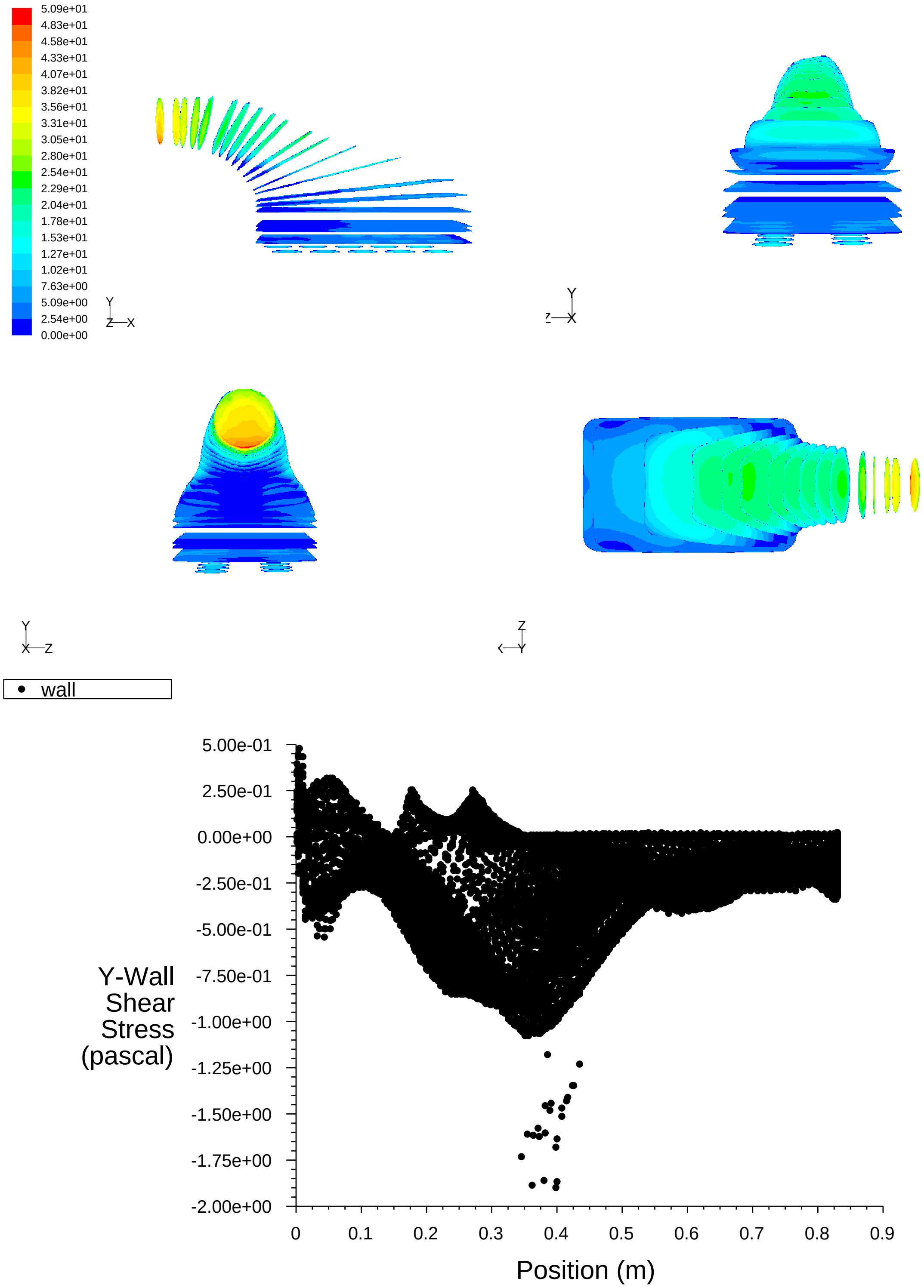


Figure 6.26: Contours of velocity magnitude through sections of the best geometry after one deformation, C_p=0.8684, and its corresponding wall shear stress shown in the y direction

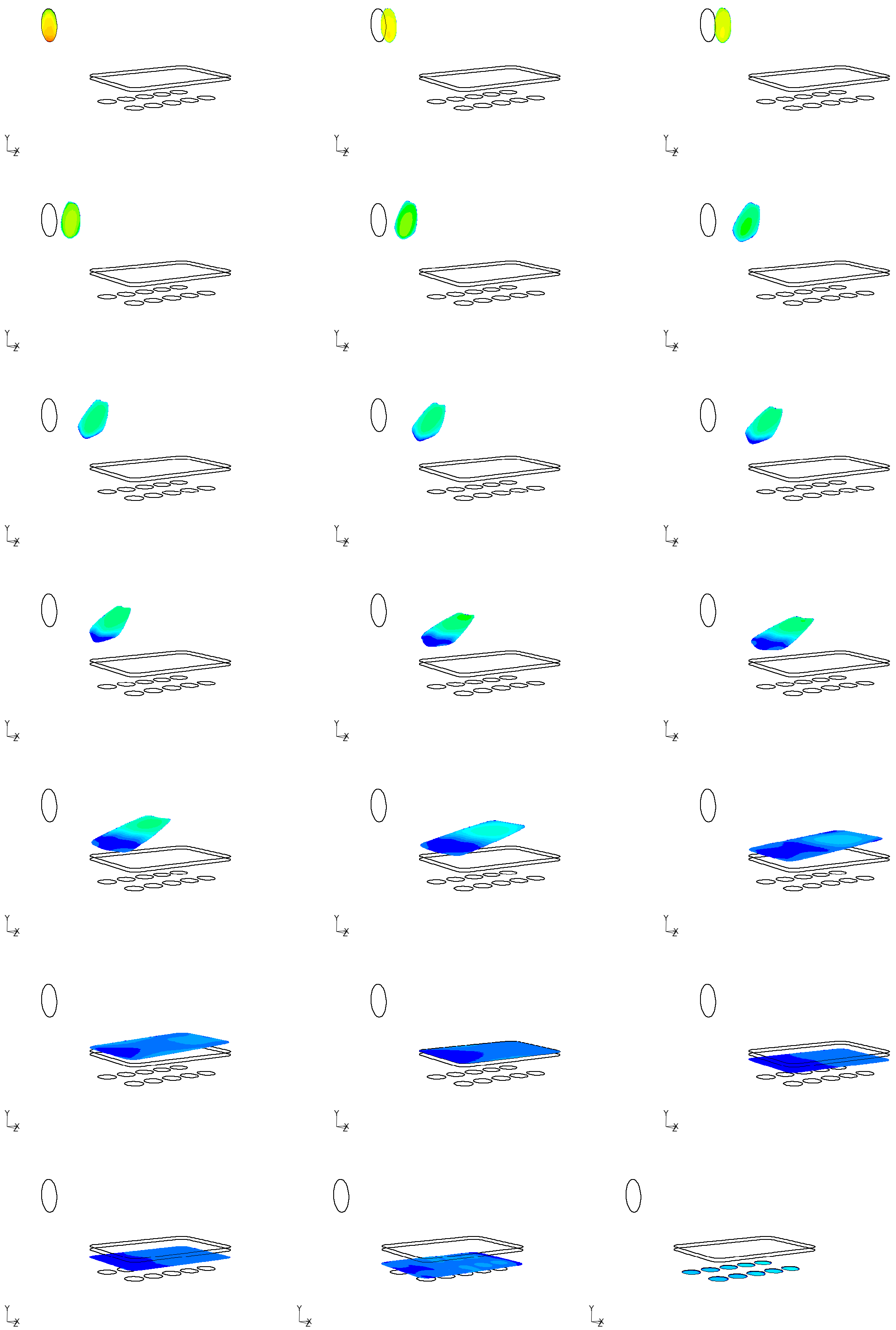


FIGURE 6.27: Contours of velocity magnitude through individual sections of the best geometry after one deformation, C_p=0.8684

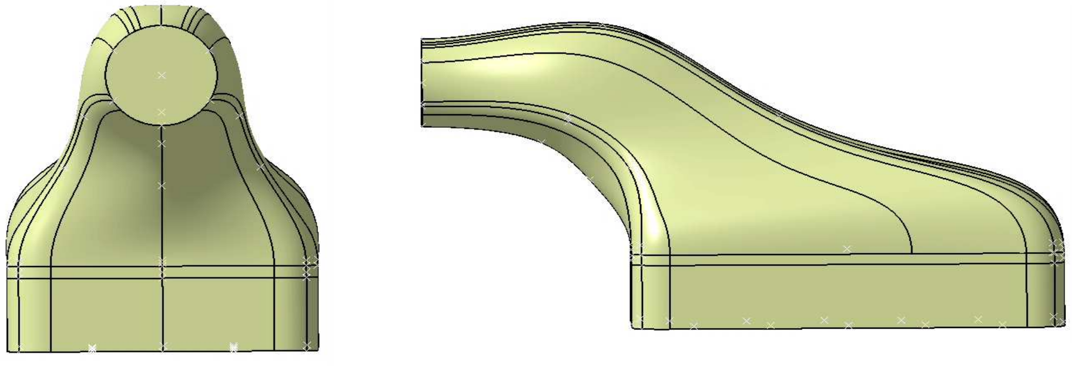


FIGURE 6.28: Best design after 50 Stage 1 update points with $C_p = 0.8808$

From this, it can be concluded that given a suitably non-optimal design, Stage 2 may be performed effectively by choosing a favourable area of bump position from the initial DoE and then updating the position, height and curvature of the deformation in a small area of the design space in this favourable region. A careful balance has to be struck here since time will be saved by not searching in other promising areas, but these could then be tested by repeating the Stage 2 process accordingly.

6.4 Summary

This chapter has implemented the AMSSOD process described in Chapter 5, and a parameterization technique to define an optimal 3D airbox has also been presented. The parameterization is a continuation of the 2D parameterization presented in Chapter 4, therefore retaining strong shape control through a decoupling of any geometrical links between the upper and lower wall guiding splines. Stage 1 has been effectively tested, and the optimization process has been performed for two different inlet flow conditions. A significant improvement in pressure recovery during this stage is found.

Due to the strong shape control present in the Stage 1 parameterization, the Stage 2 process, performed on the best geometry found after the Stage 1 updates, could not improve upon the pressure recovery. The Stage 2 update points tested geometries in a number of favourable regions until the strategy became focused in one favourable region gradually reducing the bump height variable towards zero in order to improve the pressure recovery. This suggested that the geometry found in Stage 1 was indeed a near optimum design. In hindsight, an even simpler parameterization for the airbox in Stage 1 should have been chosen to allow the Stage 2 process to work to its best ability.

The Stage 2 process was tested further on a non-optimal design and the position of the deformation centre was fixed at a favourable position determined by the initial Stage 2 DoE so that the search for an improved design by altering the bump height and curvature was focused in one favourable position. This Stage 2 process yielded an improved design with better pressure recovery. The resulting geometry had a local deformation at a position near the backbone spline of the airbox. However, the design could have been further improved towards the shape of the near-optimal design found in the Stage 1 updates if the position of the deformation centre were allowed to vary in a reduced area of the design space around the favourable position chosen from the initial DoE to move closer towards the backbone spline. For the general case of internal fluid flow applications where the location of a local deformation is unknown, a Stage 2 search in a reduced area of the design space around a favourable position chosen from the initial Stage 2 DoE should be used. To allow deformations in other favourable locations, the Stage 2 process can be repeated accordingly.

Chapter 7

AMSSOD Implemented on a Human Carotid Artery Bifurcation

The human carotid arteries supply the head and neck with blood via the two common carotid arteries (CCAs); they ascend in the neck and each divides into two branches at their respective bifurcation points: the external carotid artery (ECA) supplies the exterior of the head, the face and the greater part of the neck; and the internal carotid artery (ICA) supplies to a great extent the parts within the brain and eye cavity (see Figure 7.1). The geometry of a carotid artery bifurcation is illustrated by a point cloud formed from a scan of a patient artery shown in Figure 7.2.

Carotid originates from the Greek word *karotides* meaning *heavy sleep.* Ancient Greek physicians believed that by pressing hard on these arteries heavy sleep and loss of consciousness was induced. We now know that the carotid is the key artery which carries oxygen from the heart to the brain and logically, its constriction would indeed deprive the brain of oxygen leading to loss of body functionality and consciousness. Today, a common cause for concern medically is the progressive narrowing or hardening of the arteries over time due to the build up of fatty deposits, *atherosclerosis.* Atherosclerosis occurs naturally with age but is also accelerated by other risk factors such as high blood pressure, the presence of diabetes, cigarette smoking, antecedent cardiovascular disease,

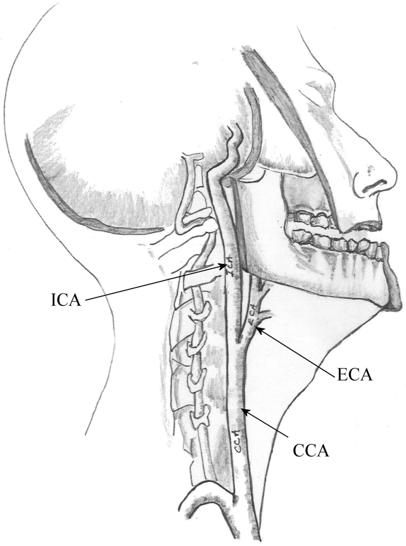


Figure 7.1: An illustration of the position of the carotid artery

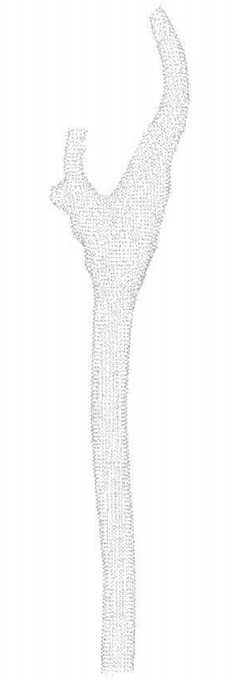


Figure 7.2: A point cloud of a carotid artery bifurcation

atrial fibrillation and electrocardiographic abnormalities (Wolf et al., 1991). The build up of fatty deposits within the intimal layer of the artery can cause a stricture around which a further build up on the inner lumen of the artery may be seen. This is known as a *stenosis*. Stenoses commonly occur at the bifurcation of the CCA into the ICA and ECA. At the point of bifurcation, a healthy ICA features a spacious bulb before the artery tapers downstream towards the brain. It is in the spacious region that a recirculation flow is present. Due to the nature of recirculatory flows, there exists an area of low shear stress, present on the inner wall, and it is here where a build up of plaque is seen in the inner wall of the artery.

In a healthy artery the flow is laminar (Ku, 1997). However, as the plaque builds up, the constriction caused by deposition of plaque in the artery results in a increase in flow velocity at this point. As the artery widens again downstream of the constriction, the flow velocity drops and the pressure increases which causes further build up downstream of the constriction. In areas such as the ICA sinus bulb, recirculatory flow is present around the plaque build up. As the stenosis becomes more severe, the flow becomes faster into the bifurcation strengthening the recirculatory flow. A danger materializes when there occurs a sudden rupturing of the inner wall plaque cap, due to the recirculatory flow and the elevated shear through the constriction, potentially leading to an embolism; where a clot or a mass of foreign material is carried by the bloodstream and becomes lodged in an artery, the blood flow is blocked.

It is this fragmentation of the plaque cap of the stenosis which is of interest in the current investigation. If the embolus is small and carried up the ICA, blurred vision is one of the symptoms that the patient may experience. If the embolus in the ICA is large, it lodges in the brain causing damage to nerve cells due to the interrupted blood flow. This is more commonly known as a stroke. A stroke can cause coma, paralysis, speech problems and dementia. Currently, stroke is the third primary cause of death in the UK, after heart disease and cancer, and is the leading cause of severe disability.

The postulation of a link between the arterial geometry, its corresponding blood flow patterns and the development of atherosclerosis first became evident in the 1960s (Fry, 1968). Since then, there has been a keen interest in understanding the detailed arterial fluid dynamics (Caro et al., 1971). However, the link between the geometry, the flow and the development of atherosclerosis remains to be identified (Davies, 2000).

Our interest in the carotid artery stems from this need for a fuller understanding of the local *haemodynamics* or blood flow patterns that occur in the carotid artery bifurcation. This notion that haemodynamics plays an important role in the development and progression of atherosclerosis means that, by extensive studies of arterial geometries and their related blood flow patterns, the medical community can progress towards improved diagnoses and treatment of arterial disease.

Advances in medical imaging have allowed doctors and researchers to visualise the extent of arterial disease. Images of the arteries can be obtained with high accuracy, but the process of measuring haemodynamic patterns becomes untenable due to the length of time needed to produce these images. Instead, CFD is used to model the pulsatile blood flow through the arteries. A number of studies have been performed on diseased arteries with idealised geometries or two-dimensional models (Milner et al., 1998) which can only be applied to the very general human case. Although the medical field has benefited from great insights that these studies have provided, it is apparent that the use of three-dimensional patient-realistic geometries coupled with suitable realistic flow conditions are vital in order to fully characterize the three-dimensional flow patterns. Much research has been performed more recently into three-dimensional vessel reconstruction via the translation of patient image scans into patient specific CAD based geometries compatible with meshing and CFD tools (Steinman et al., 2002b; Antiga and Steinman, 2004). With the recent advances in high resolution medical imaging, arterial geometries can be measured non-invasively. Magnetic Resonance Imaging (MRI), for example, can capture the outer and inner wall geometry of the carotid bifurcation in a number of planar scans which can be reconstructed into a computational model and so provide a precise and accurate representation of the patient's artery and disease build up. CFD models can then be run on these computational geometries and these CFD models give a quantification of the blood flow that is at least as accurate as regular invasive methods (Milner et al., 1998; Moore et al., 1999).

There has been much research into the possible connection between the development of atherosclerosis and an individual patient's artery geometry features. Thomas et al. (2005) concluded from their studies that the development of arterial disease could not conclusively be linked with the general arterial geometrical properties of young healthy individuals. Due to the similarity between young, healthy artery geometries in comparison to the enormous variations seen between the geometry of mildly diseased arteries, it is evident that the development of atherosclerosis becomes apparent in each individual with age, but early detection of those that will be adversely affected by atherosclerosis is challenging. It is therefore necessary to acquire a greater understanding of the haemodynamics of normal, mildly diseased arteries, as well as heavily stenosed arteries. Clearly, it would be difficult to undertake a large patient population study of patients diagnosed with carotid artery disease to try and assess possible geometric links with the build up of the disease. Instead, the focus of this study concentrates on the arterial geometry of an individual patient to construct a parameterized realistic computational representation of the artery. The key advantage in constructing a realistic parametric artery based on image reconstructions (Steinman et al., 2002b), is the ability to use parameters in optimization studies. For example, the benefit of certain treatments can be optimized given patient-realistic parametric artery models.

Carotid angioplasty followed by stenting is a common treatment for atherosclerosis. This procedure involves inserting a catheter which is guided towards the carotid artery. It carries a small balloon that inflates to flatten the plaque against the artery wall. The stent, which is made of a stainless steel wire mesh in the shape of a small tube is inserted into the artery to hold it open. This then restores normal flow in the carotid artery so that blood and oxygen can get to the brain. Carotid angioplasty and stenting is being used increasingly as a safer and more cost effective alternative to the carotid endarterectomy procedure which involves the physical removal of the plaque from the artery. To date, however, no conclusive long-term results can be drawn from the stenting procedure as it is a relatively recent treatment. However, problems can occur such as a re-stenosis occurring in less than six months after the implantation. This has been observed in a number of patients (Yadav et al., 1997; Wholey et al., 1998). Clearly, the effect on the local blood flow caused by a stent insertion is not fully understood. An alternative form of treatment is anastomosis, or bypassing, of the artery and, potentially, this discipline can make good use of a parametric representation of the carotid artery bifurcation. For fast and efficient optimization studies, it is desirable to employ a compact set of design variables. The AMSSOD parametrization techniques presented earlier in this thesis provides an effective geometry manipulation process. This would facilitate the future efficient optimization of the benefit of treatment, elucidating the change in characteristics of the blood flow caused by the treatment, potentially allowing surgeons to decide which patients would favour certain treatments. Not only may this provide a better understanding of the haemodynamical effects of various treatments but also provide some insight into the suitability of the procedures and whether of surgical intervention is favourable. This provides the motivation for the construction of a realistic parametric artery model.

7.1 Geometry Parameterization

The first and foremost difficulty which is faced when approaching the parametrization of a carotid artery bifurcation is the recognition of geometric features common to all types of arteries. Figure 7.3 illustrates the large diversity of arterial geometries (BioFluidMechanicsLab). These arteries are digitised silhouette images of postmortem specimen casts of the inner lumen boundaries on the arterial walls. Here, the patients are of different sexes and are aged between 49 and 90. This chapter utilises the AMSSOD technique described in Chapter 5 to recreate automatically a parametric model of one of these artery geometries with the realistic patient-specific features.

Some obvious common geometrical features are seen in Figure 7.3, each of which has substantial variations between one artery and the next. Examples of such variations are seen in the ICA bulb width, the ICA and ECA branch angles and the diameter of the ICA, ECA and CCA. An early generic synthetic geometry for experimental uses of the carotid artery bifurcation was developed by Bharadvaj et al. (1982) who represented the artery mainly by Y-shaped models. A relatively new geometry representation is the tuning fork model (Ding et al., 2001). A typical CAD geometry of this “tuning-fork” carotid artery bifurcation is seen in Figure 7.4. Although these models capture the general geometry and size of the ICA, ECA and CCA, they are not capable of capturing differences between geometries such as those seen in Figure 7.3, and so any specific internal blood flow characteristics resulting from these geometrical differences could not be captured.

The variation of inter- and intra-patient arterial geometries leads to a corresponding variability in the haemodynamic environment. It is therefore very important that a practical

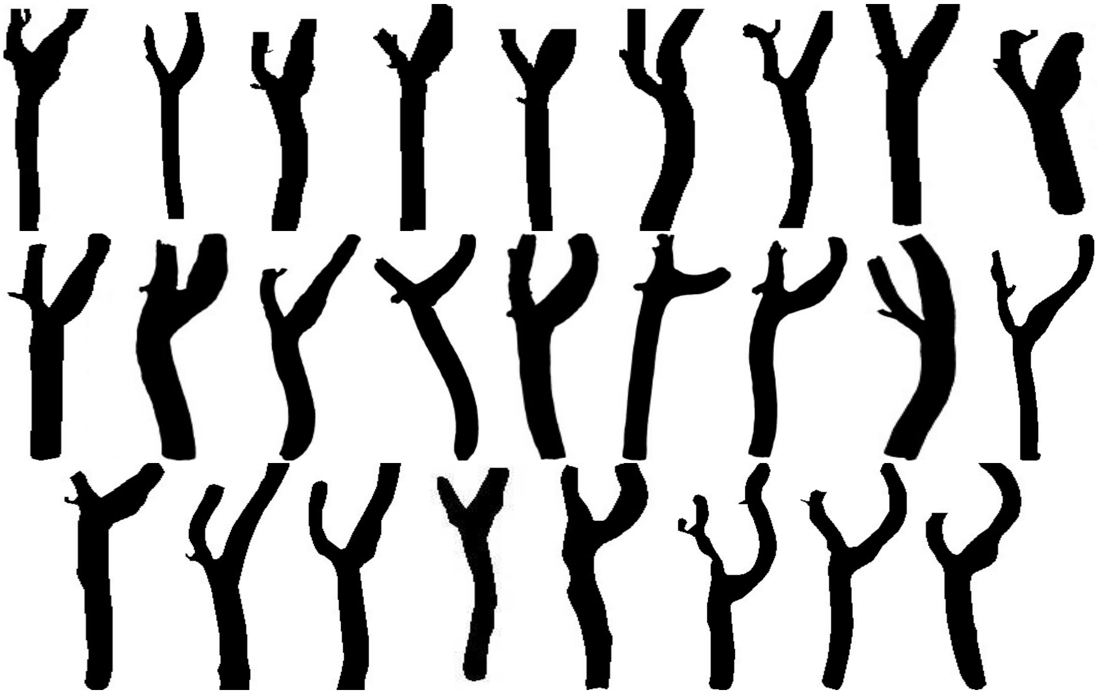


FIGURE 7.3: Silhouettes of digitised patient artery casts courtesy of BioFluidMechanicsLab

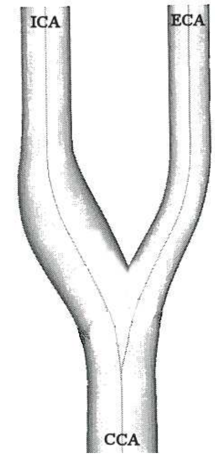


FIGURE 7.4: A typical CAD carotid artery bifurcation

parametric artery model should be capable of detailed and complex manipulations. The huge geometric variation between different patient geometries can be described by factors such as the branch angles, tortuosity (Brinkman et al., 1994), curvature (Smedby, 1998) and (non-)planarity (Friedman and Ding, 1998). The relative magnitude of these features could make the difference between a potentially fatal or non-fatal progression of atherosclerosis. In the severe cases of stenoses where surgical intervention is required, it is fairly routine to carry out an MRI scan of the patient's artery. This scanned data is used here to construct a base parametric model which, although idealised, captures the main geometrical features such as branch angle, diameter, tortuosity and non-planarity.

In many cases, assumptions and constraints inherent to CAD techniques are imposed on the geometric CAD models, thus limiting the overall flexibility to create patient-realistic geometries (Ding et al., 2001). However, the multi-stage technique proposed in this thesis overcomes a number of constraints imposed by the surface generation properties adopted by the CAD engine. Here CATIA is used to construct a realistic parametric CAD model, using the implementation of the process outlined in Figure 7.5.

Given scanned data for a real human artery, a baseline parametric model is constructed in Stage 1. Stage 2 then manipulates the base parametric CAD model so as to capture the localised features of the real artery geometry. The error between the base model and the real artery model is assessed and the CAD model is updated to reduce this error until a sufficiently accurate fit is found. The following sections outline the two stages in more detail.

7.1.1 Stage 1

The Stage 1 process outlined in Figure 7.5 begins by defining a parametric geometry. In other words, the choice of parameters necessary to construct the artery for future optimization work must be decided. It is important to have accurate artery diameters at varying sections along the artery. Determining the ICA branch angle will provide the orientation of the ICA at the bifurcation as well as the plane along which the bulb width is measured. With this in mind, it is possible to analyse data from any scan automatically to output these parameter values. For this study it was not possible to acquire MRI scan data directly but a point cloud, consisting of approximately 7000 points, taken from a scan of a postmortem specimen cast of the inner lumen boundaries on the arterial wall (BioFluidMechanicsLab) was used instead. An MRI would output a similar set of points but with a varying degree of resolution of the slices.

The analysis of the point cloud data begins by translating the geometry so that its origin is found near the exit of the CCA into the root of the bifurcation cavity. This can be achieved by roughly slicing the point cloud data into xy-planes at varying z values, where the z direction is aligned with the principal flow direction. Gaps in the data are indicative of the plane intersecting both the ICA and the ECA. The diameter of the

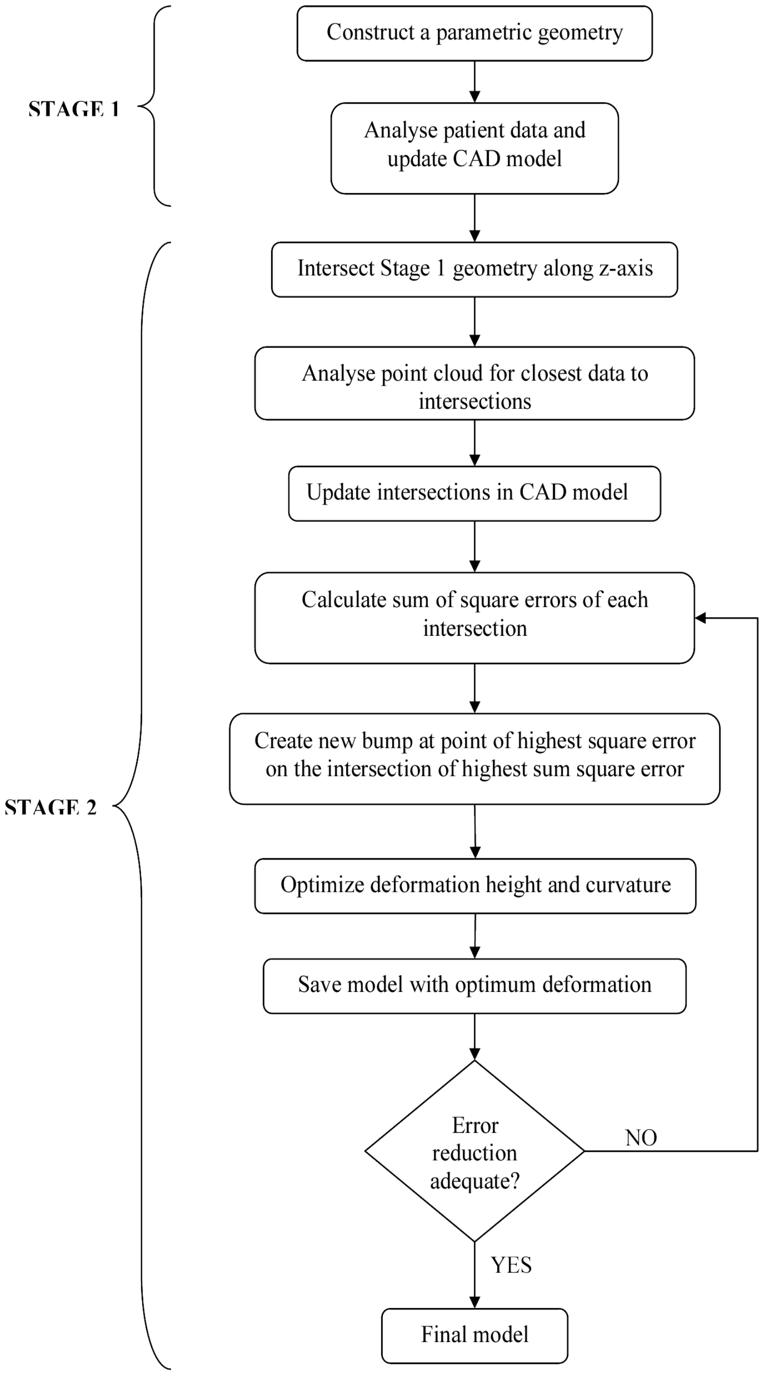


FIGURE 7.5: AMSSOD process for a carotid artery bifurcation

artery slices on these planes is determined. Where the planes intersect the CCA, the diameter should be reasonably constant and so the new origin is placed on the plane downstream of the plane where the diameter increases significantly.

Once the new origin has been found, the key to calculating the values of the parameters is locating the "divider" point. This is the point on the inner arterial wall between the ECA and the ICA at the bifurcation, shown by F in Figure 7.6. To do this, the x-axis is split into regular intervals of approximately 1mm in length. In each of these intervals, the z-distance between each adjacent point is calculated. If this value is below a specified tolerance the maximum z-value is found. Should the distance be above the specified tolerance, it indicates that in this particular interval of x-values, the ICA or the ECA may be crossing over the bifurcation. In this case, the maximum of the z values situated below the gap is found. The z coordinate of the divider point is then taken to be the minimum z value of the set of maximum z-values for each interval of x. The x and y coordinates of the divider point are then found accordingly. Having found this divider point, tortuosity of the artery must be considered. Given specified intervals of rotation about the z-axis θ ($\theta \in [0, 2\pi]$), the geometry is rotated through increments of θ and the process described above repeated for calculating the minimum value of the set of maximum z-values for each interval in x. If this new z coordinate for the divider point is found to be less than the original z coordinate of the divider point with no rotation, the divider point z coordinate is replaced. The x and y values are found accordingly and are rotated back through -θ to replace the original x and y coordinates. This process is repeated at increments of θ until the true divider point is found.

Now the coordinates of the divider point have been determined, the point cloud can be intersected at one plane midway between the new translated origin and the CCA inlet, at point K in Figure 7.6. At these three planes (J, K and O in Figure 7.6), an ellipse is fitted to the point cloud data and sampled at the CCA inlet and the midway plane with five points, and the plane at the origin is sampled at eight points. These eight points are used to start an appropriate number of splines to define the ICA and ECA (see Figure 7.6).

Next, a suitable number of planes are chosen to intersect the bifurcating artery above the origin on plane O. This has a greater resolution than the intersection of the CCA in order to capture the greater detail in geometry through the bifurcation and along the

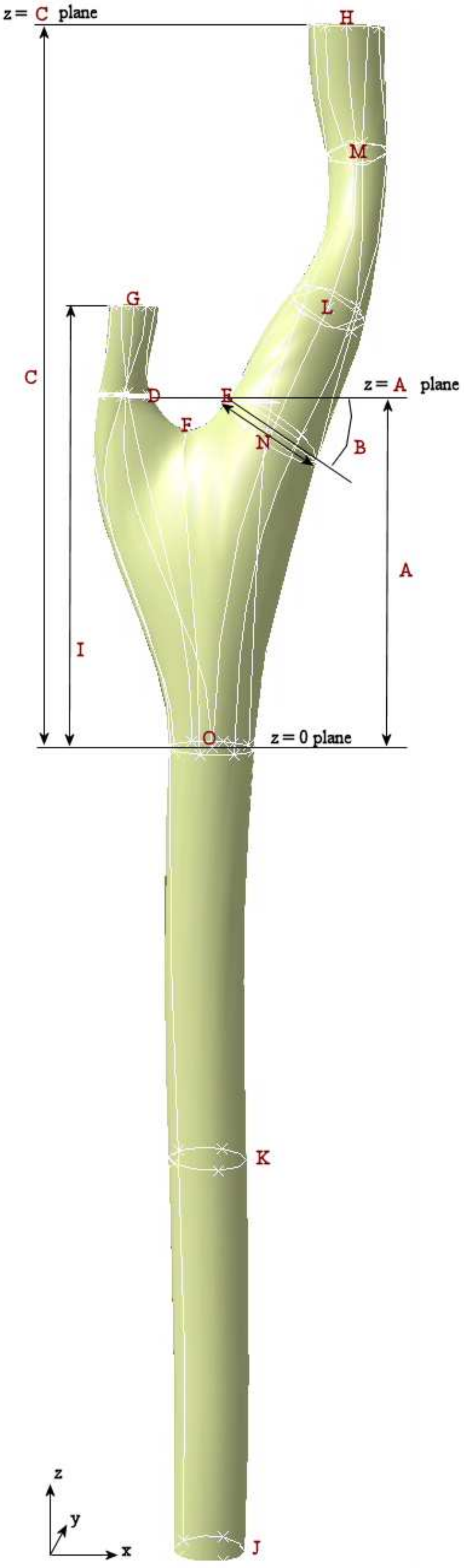


Figure 7.6: Carotid artery bifurcation parameterization for Stage 1

ICA and ECA. For this study six intersecting planes equally spaced from the origin at the root of the bifurcation to the ICA exit are chosen; see Figure 7.7. At each of these planes, ellipses are fitted to the data on each of the six intersections, as shown in Figure 7.7, and the diameter and the centroids of these ellipses extracted.

The root points of the ECA and the ICA, shown by points D and E respectively in Figure 7.6, are determined by finding the coordinates of closest points on the intersecting plane, directly downstream of the divider point, to the divider point at F.

The branch angle of the ICA is next found by calculating the angle between the divider point, at F, and the root ICA point, at E, on the zx-plane. This angle defines the plane in which the bulb width is measured.

The intersecting planes further downstream of the divider point (see Figure 7.7) allow for the centroid and diameter of the ICA and ECA to be input into the CATIA model to define the arteries on the z=A plane and surrounding the centroids at G, L, M and H depicted in Figure 7.6. These values give an accurate representation of the non-planarity and tortuosity downstream of the bifurcation. As there is only one intersecting plane through the ECA on the z=A plane, the ECA is extended to the next intersecting plane with constant diameter to allow for an ECA outflow sufficiently far downstream to aid the convergence of the CFD solution. The parameters which determine the Stage 1 model are defined in Table 7.1, and are illustrated in Figure 7.6. An external design table is used to control these parameter values. This allows the model to be altered automatically by changing the parameter values inside the design table. An outline of the code used to determine the parameter values automatically is provided in Appendix C.

7.1.2 Stage 2

With the Stage 1 CAD geometry complete, the purpose of Stage 2 is to minimize the error with respect to real artery point cloud data. To determine this error, the CAD model is sampled at a number of points. To do this, the Stage 1 model is intersected on xy-planes of varying z value from the origin through the bifurcation to the ICA exit. Here, the bifurcation root area is intersected five times including the plane upon which the divider point lies. Maintaining the same distance between intersections, the arteries

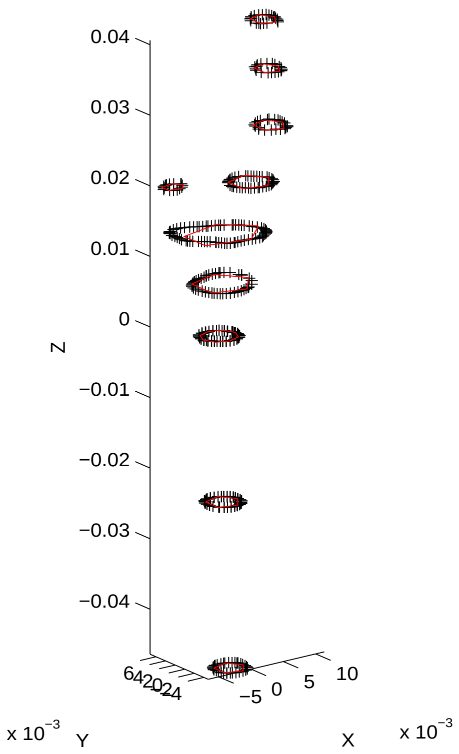


FIGURE 7.7: Ellipses fitted to the point cloud at the six intersections

Notations	Description
A	Height of bifurcation point plane above x, y-plane at origin
B	ICA branch angle
C	Height of ICA exit plane above x, y-plane at origin
D	Cartesian coordinates (ECArootX, ECArootY, A)
E	Cartesian coordinates (ICArootX, ICArootY, A)
F	Cartesian coordinates ($\frac{1}{2}(D(x) - E(x))$, $\frac{1}{2}(D(y) - E(y))$, A + BifurcationRoot)
G	Centroid coordinate of ECA exit (ECAcent(x), ECAcent(y), I)
H	Centroid coordinate of ICA exit (ICAcent(x), ICAcent(y), C)
I	Height of ECA exit plane above x, y-plane at origin
J	Sampled point cloud at z=min(z) plane at 5 points
K	Sampled point cloud at z=min(z)/2 plane at 5 points
L	Centroid of sampled point cloud at z=I describing circle of radius L_r
M	Centroid of sampled point cloud at z=R describing circle of radius M_r
N	Bulb width
O	Sampled point cloud on z=0 plane at 8 points

TABLE 7.1: Parameters and formulae for the parametric CAD bifurcation model

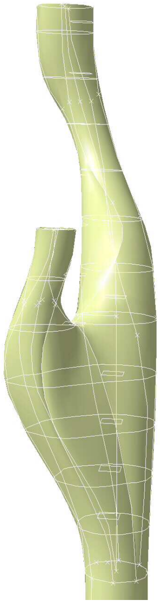


FIGURE 7.8: Positions of cross-sections at various z-axis values

downstream of the divider point are intersected six times, making 11 intersections in total. Note that the intersection planes are positioned to align with the dense regions of the point cloud. To ensure that the intersection is not placed within a gap between the slices making up the point cloud, a band width of $\frac{1}{10}$mm is positioned around the intersecting plane. This band is moved along the artery until a slice of points from the point cloud is captured within it and the new value of the intersecting plane is stored. This step is shown in the second box of Stage 2 in Figure 7.5. The revised values of the intersection planes are then used in a macro to automate the intersection of the Stage 1 model with 15 points sampling each intersection at regular intervals. Figure 7.8 shows the Stage 1 CAD geometry with the intersecting planes.

The CATIA macro exports the 15 points on each intersection to a file which is read by a Matlab code to extract the coordinates of these points. The nearest point in the point cloud to each sample point is found and the square error between each of the CAD sampled points and the point cloud is determined. Following the flow chart in Figure 7.5, the sum of the square errors of each intersection is calculated. This allows the intersection with the highest sum of square errors to be found. The worst point on this plane at which the square error is highest is found and a new point in the CAD model corresponding to the target point cloud point is created. This point is then projected onto the CAD artery surface along a direction normal to the surface, as the surface

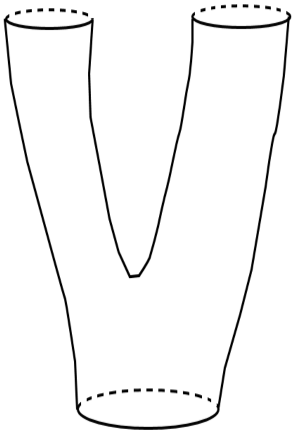


FIGURE 7.9: Limit curve (shown as the solid line) used for near side Stage 2 deformations

will be deformed along this direction. This eliminates the possibility of the deformation overshooting the target point cloud. The macro then creates a new bump with this projected point at its centre. Within the model, two limit curves are defined, one of which is shown in Figure 7.9. Either the near side or the far side of the bifurcation is deformed depending on the position of the projected point onto the surface. The limit curves are placed here as they must remain fixed throughout the whole Stage 2 process as, in CATIA V5, new limit curves cannot be drawn on an already deformed surface to allow for further deformations. Deformations can, however, be placed independently or on top of each other as required. Once the bump has been created and the surface is deformed, the macro then re-intersects the deformed model, exporting the new coordinates of the deformed sample points.

After deciding the location of the deformation, its size must then be optimized. The design objective for this problem is as follows. After each bump is added, the coordinates of the 15 points on each intersecting plane in the CAD model are exported. The following code is run to determine the closest point in the point cloud data to each of the 15 points on each intersection

Here, $t(i)$ is the position of the point in CloudPts which is closest to intersection_pt(i,:) and $d(i)$ the corresponding distance which is taken to be the error objective function which is minimized when optimizing the bump height.

The curvature of the bump, however, can only be found by minimizing a different objective, that of the sum of the errors on all points on all planes:

Algorithm 1

```
for each intersection do
  $t$ = zeros(size($xi$,1),1);
  $d$ = zeros(size($xi$,1),1);
  for $i$=1:15 do
    produce a matrix $yi$ consisting of tiled copies of intersection_pt($i$,:) of the same size as size(CloudPts,1);
    find minimum distance between intersection_pt($i$,:) and CloudPts;
    [$d(i)$,$t(i)$] = min(sum(($x$-$yi$)$^2$,2));
  end for
end for
```

$$\text{Sum}_{error} = \sum_{j=1}^{11} \sum_{i=1}^{15} d(i) \tag{7.1}$$

First, with the height of the bump as the design variable with a constant curvature ratio of 1, the optimization tool OPTIONS (Keane (2002)) is used to create a Design of Experiments (DoE) of five points. The DoE points are prescribed at heights of 0m, 0.00125m, 0.0025m, 0.00375m and 0.005m if the error ($d(i)$) is positive and 0m, -0.00125m, -0.0025m, -0.00375m and -0.005m if $d(i) < 0$. This is followed by five update points to find the optimum bump height. Second, a DoE of five points prescribed at fixed optimal bump height and curvature ratios of 0, 0.25, 0.5, 0.75 and 1 is used, followed by five update points. The model with optimum bump height and curvature is then saved. The errors are analysed to determine whether the fit between the CAD model and the real data is sufficient. This process is repeated adding more and more deformations until this condition is satisfied.

An outline of the code used to determine the best intersection planes, the macro for intersecting the Stage 1 geometry and the macro for creating deformations automatically is provided in Appendix C.2.

The purpose of the Stage 2 geometry manipulation is to correct the error which is inherent in the Stage 1 CAD model. This error is not due to the CAD engine itself but due to the simplicity with which the initial geometry is described. Here, an increase in the number of splines describing the Stage 1 geometry will not increase the number of design variables and thus has no impact on the efficiency of the optimization process. In this case, the greater the number of splines, the greater the complexity of the model. Although no optimization occurs based upon these parameters, the purpose of this study is to create a parametric model whose parameters can easily be manipulated within an efficient optimization study. For example, should an optimization study be performed to maximize the degree of stenosis before surgical intervention becomes necessary for a particular patient geometry, perhaps three parameters would be altered to change the diameter of the artery at the position of the stenosis. A more complex geometry requires more variables to implement the same change thereby reducing the optimization process' efficiency and practicality.

7.2 Results

At the end of Stage 1, the set of parameter values for the new geometry are stored in the design table. These are given in Table 7.2. By allowing an external design table to control the changes to the CAD model, a number of different patient geometries can be described automatically with this Stage 1 process, examples of which can be seen in Appendix C.1.

Figure 7.10 illustrates the fit between the real artery, shown in orange, and the Stage 1 CAD geometry, shown in beige. The fit is good in that it captures the non-planarity of the ICA and ECA, and the mild tortuosity experienced by the geometry downstream of the bifurcation root. However, there are features on the real geometry, such as the cavity seen at the root of the ECA which are not captured by the intersections. It is features such as these that should be captured in Stage 2. If this is not the case, then the resolution of the intersections performed at the start of Stage 2 must be refined to allow for sharp contained bumps which may lie between intersections. If this does not allow for appropriate deformations to capture the true features of the real artery, the resolution of the initial Stage 1 xy-plane intersections must be refined to more accurately describe the splines defining the initial Stage 1 geometry.

From Figure 7.10 it becomes apparent as to where the main error between these two geometries lies. Clearly, the area of the bifurcation root is underestimated in the Stage 1 geometry. This is due to the small number of splines guiding the shape of the ICA and the ECA. Ellipses were not fitted to the point cloud at intersections between the origin and the divider point because it would be unusual for the shape to be close to an ellipse in many cases. By providing sample points on an ellipse, the actual shape of this

Parameters	Value
A (m)	2.0318000e-02
B (deg)	3.1608137e+01
C (m)	4.0636000e-02
ECArootX (m)	-3.8040000e-03
ECArootY (m)	4.3480000e-03
ICArootX (m)	9.4400000e-04
ICArootY (m)	7.5000000e-04
BifurcationRoot (m)	-3.3760000e-03
ECAcentX(m)	-5.0975000e-03
ECAcentY (m)	4.4360000e-03
ICAcentX (m)	9.6195000e-03
ICAcentY (m)	5.3665000e-03
I (m)	2.7100000e-02
J(x,y) (m)	(2.5000000e-04,-2.9845000e-03)
	(-1.1772254e-03,-1.1151984e-03)
	(-3.4865246e-03,-1.8292081e-03)
	(-3.4865246e-03,-4.1397919e-03)
	(-1.1772254e-03,-4.8538016e-03)
J(z) (m)	-4.6374000e-02
K(x,y) (m)	(2.8300000e-04,-1.7240000e-03)
	(-1.2496003e-03,3.8544335e-04)
	(-3.7293997e-03,-4.2029231e-04)
	(-3.7293997e-03,-3.0277077e-03)
	(-1.2496003e-03,-3.8334434e-03)
K(z) (m)	-2.3187000e-02
L centroid(x,y) (m)	(3.8025000e-03,3.6400000e-04)
L_r (m)	5.7630000e-03/2
M centroid(x,y) (m)	(9.7050000e-03, 4.6960000e-03)
R (m)	3.3863333e-02
M_r (m)	3.2380000e-03/2
N (m)	6.89188e-03
O(x,y) (m)	(6.4900000e-04,-7.7650000e-04)
	(-8.2061474e-05,8.3605701e-04)
	(-1.8470000e-03,1.5040000e-03)
	(-3.6119385e-03,8.3605701e-04)
	(-4.3430000e-03,-7.7650000e-04)
	(-3.6119385e-03,-2.3890570e-03)
	(-1.8470000e-03,-3.0570000e-03)
	(-8.2061474e-05,-2.3890570e-03)
O centroid	(-1.8470000e-03,-7.7650000e-04)
ECA diameter at exit	3.1050000e-03

TABLE 7.2: Parameter values used in the design table for the Stage 1 model

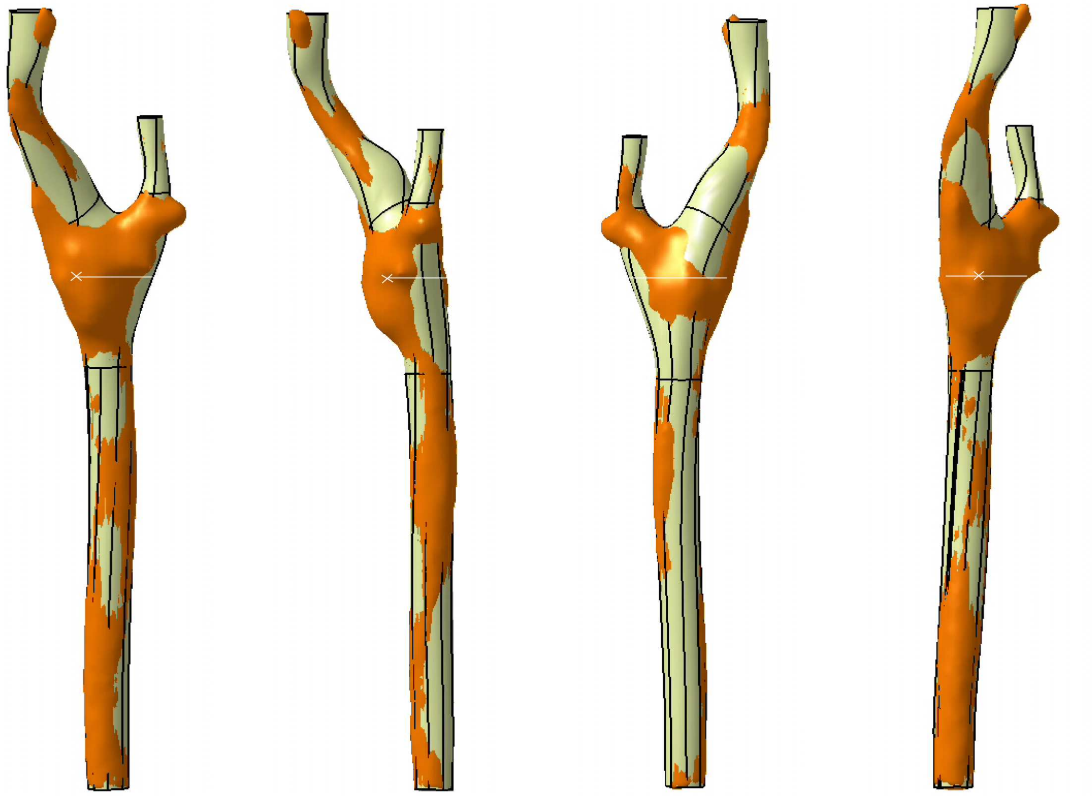


FIGURE 7.10: Difference in geometry between the patient artery (orange) and the parametric CAD geometry (beige) after Stage 1 shown from four different angles (plane 3 and point 15 on plane 3 are also illustrated)

section of the artery surface could be grossly misjudged and result in an unnecessary increase in parameters. Here, the surface is a loft between the splines guiding its shape.

As would be expected given the explanation above, the largest sum of point errors is found initially on plane 3, with the highest square error at sample point 15 on this plane, shown in Figure 7.10. A bump is created on the artery surface and the errors are minimized to find the optimal height and curvature of the deformation for each bump. In this study, six deformations are performed and the optimal bump height and curvature values can be seen in Table 7.3.

The square errors at each point, interpolated over the surface of the artery, along with the sum of the square errors on the intersecting planes for the initial geometry constructed in Stage 1 are illustrated in Figure 7.11. Plane 3 has the largest error of $2.318 \times 10^{-5}\text{m}^2$ and the worst region on this plane is shown to be in red on the far side of the artery as pictured and on the ICA side, at sample point 15. The optimization of Bump 1 applied

Bump number	Worst point	Plane, Point	Point error before bump (m^2)	Point error after bump (m^2)	Bump height (m)	Total error before bump (m^2)	Total error after bump (m^2)	Curvature
1	193	3,15	6.46×10^{-6}	2.97×10^{-8}	2.50×10^{-3}	9.53×10^{-5}	7.43×10^{-5}	0.01
2	187	3,9	2.89×10^{-6}	4.40×10^{-8}	-2.55×10^{-3}	7.43×10^{-5}	6.98×10^{-5}	0.37
3	377	2,3	2.29×10^{-6}	1.87×10^{-7}	1.25×10^{-3}	6.98×10^{-5}	6.41×10^{-5}	0.01
4	638	5,8	1.87×10^{-6}	1.66×10^{-8}	1.56×10^{-3}	6.41×10^{-5}	6.24×10^{-5}	0.93
5	813	3,2	2.23×10^{-6}	1.17×10^{-7}	2.50×10^{-3}	6.24×10^{-5}	5.83×10^{-5}	0.24
6	1066	5,14	1.34×10^{-6}	4.09×10^{-7}	1.25×10^{-3}	5.83×10^{-5}	5.63×10^{-5}	1

Table 7.3: Table showing the optimal values of bump height and curvature for the worst point found after Stage 1 and each subsequent bump

to the Stage 1 artery shows the reduction of this error in Figure 7.12. Here, the error on plane 3 has been reduced, although it still remains the highest on this plane. The highest point of error is now on the near ECA side of the artery on the same plane. After the optimization of Bump 2, the results are seen in Figure 7.13. Figures 7.14-7.17 show the results after Bumps 3-6 respectively. With each new bump, the maximum sum of square errors is reduced, as shown on the right hand side of each of these figures. The percentage difference in the total sum of square errors after Stage 1 and after Bump 6 is 41%, the reduction after each bump of this error can be seen in Figure 7.18. The greatest planar sum of square errors reduces from 2.318×10^{-5} after Stage 1, to 7.026×10^{-6}, a percentage difference of 70%.

Although the error reductions are significant, the process comes to a halt after six deformations. This is because the search for the next worst point, and therefore the projected bump position, lies along the limit curve. As the limit curve itself cannot be deformed, the process is concluded. This is a consequence of using CATIA V5 to model the geometry rather than of the overall multi-stage approach. The limit curves lie along the guiding splines defined with control points taken from the analysis of the real artery data at a small number of intersections. To obtain a greater reduction in error, the process would return to Stage 1 and intersect the point cloud data with a

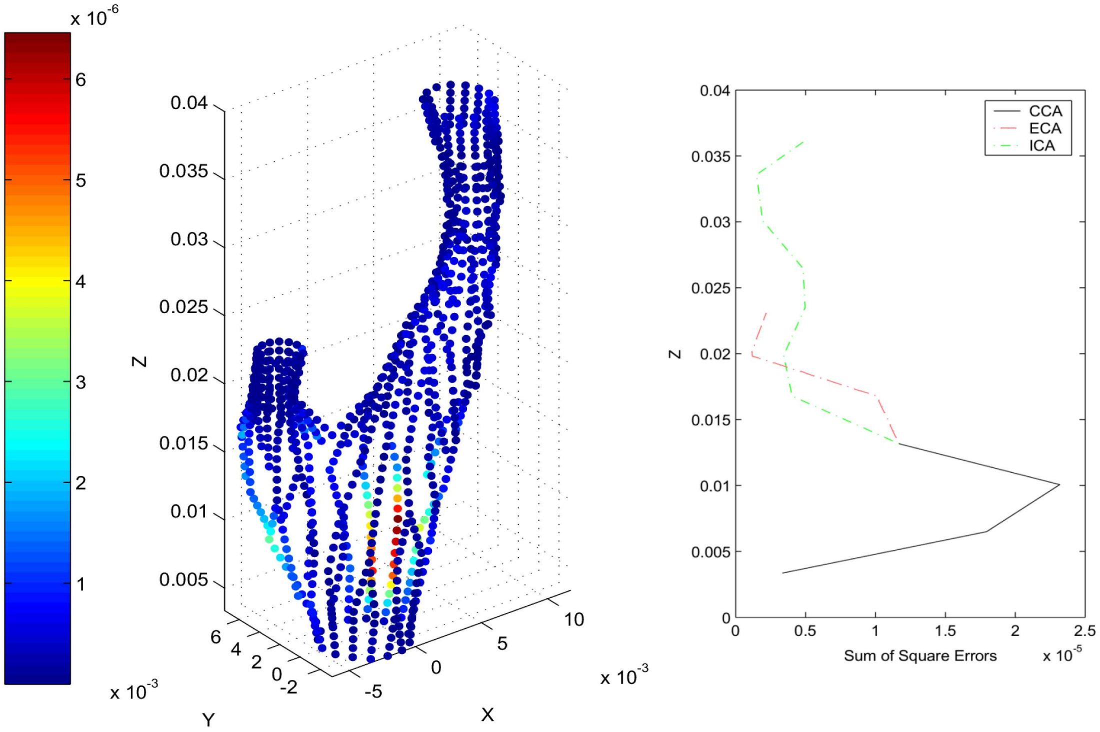


FIGURE 7.11: Square error between CAD and real geometry shown on the left along with the sum of square errors on the intersecting xy-planes shown on the right, after Stage 1

finer resolution and translate this into the Stage 1 geometry. The Stage 2 process could then be repeated and more deformations may be allowed before the error falls along the limit curve. As it is a comparison between the error between discrete points rather than between surfaces, total accuracy or 100% reduction in error will never be achieved.

An outline of the macro construction for the automatic placing of deformations is given in Appendix C.2.

7.2.1 CFD Comparison of Patient and CAD Carotid Artery Bifurcation Model

To obtain an idea as to how the local manipulation of the artery surface in Stage 2 has improved the accuracy of the artery shape, the flow simulations through the Stage 1 CAD model and the Stage 2 CAD model can be compared with the original target artery geometry.

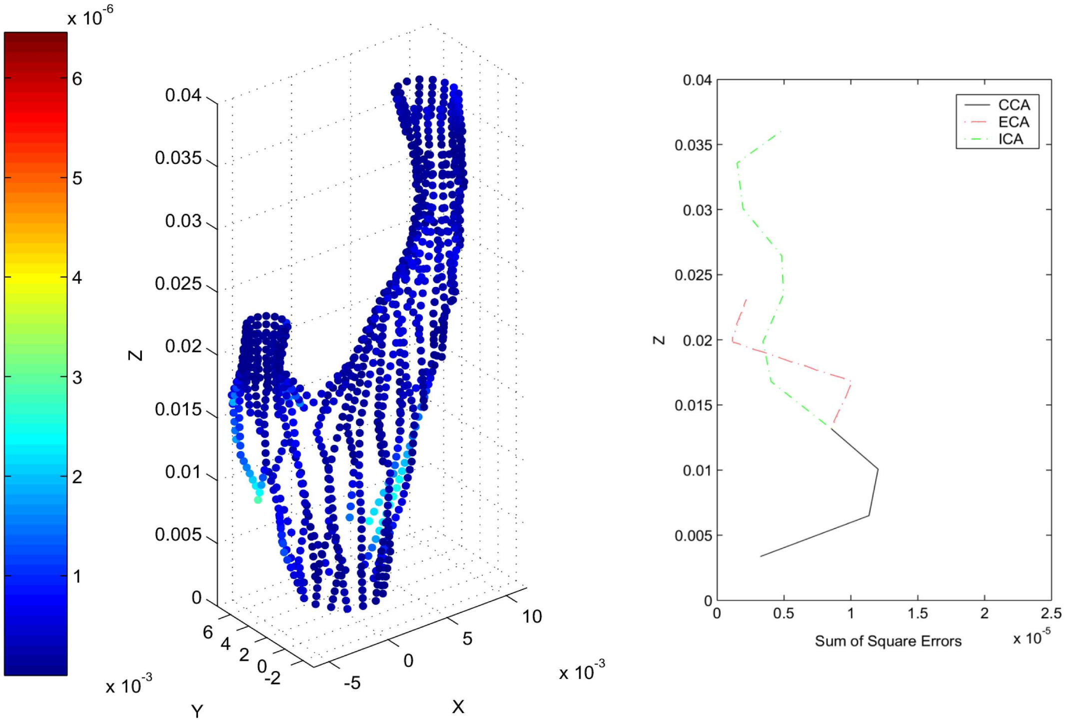


Figure 7.12: Square error between CAD and real geometry shown on the left along with the sum of square errors on the intersecting xy-planes shown on the right, after Bump 1

All models are meshed using a mesh model kindly provided by Dr Neil W. Bressloff using a hex-core hybrid volume mesh with an interval size of 0.48mm. This was chosen in light of the mesh dependency study performed by Bressloff et al. (2004). For this study, this interval size generated a volume mesh containing approximately 65,000 cells. An example of this mesh is shown in Figures 7.19 and 7.20, where the mesh is cut through the yz-plane to reveal the hex-core. This mesh comprises structural hexahedral cells in the centre of the artery and tetrahedral cells on the outer edges of the volume. The interval size used is the length of each side of the cell in the hex-core. The tetrahedral cells toward the edges of the volume have an interval size of this and smaller. Using this interval size, the boundary layer along the artery wall will be captured.

The flexibility of the artery walls are neglected in the following studies. The entry to the artery birfurcation is set as a user-defined velocity inlet and the exits of the artery through the ICA and ECA are set as outflows.

The pulsatile velocity inflow waveform, set as the boundary condition at the CCA inlet, is

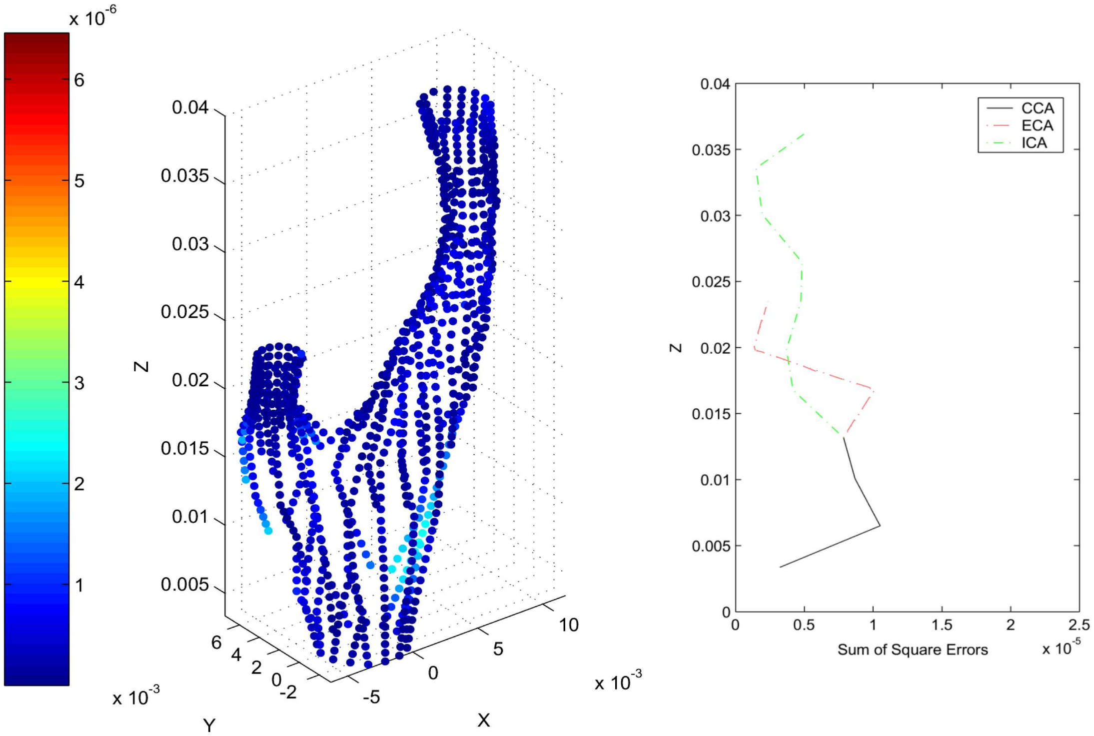


FIGURE 7.13: Square error between CAD and real geometry shown on the left along with the sum of square errors on the intersecting xy-planes shown on the right, after Bump 2

shown in Figure 7.21. This inlet velocity profile is based on the pulse used by Holdsworth et al. (1999) and has a time-averaged Reynolds number of Re = 271. The density of blood is assumed to be $\rho_{blood} = 1035\text{kgm}^{-3}$, the viscosity is $\eta = 0.0035\text{kgm}^{-1}\text{s}$, the CCA diameter at the inlet is $d_{CCA} = 0.008\text{m}$ and the mass flow split between the ECA and the ICA is of the ratio 30:70. As the peak Reynolds number falls below the critical value of approximately 2300 for steady fully developed pipe flow, laminar flow through the artery is assumed. Steady solves have been performed for a carotid artery bifurcation (Bressloff et al., 2004) and although these solves capture to a mild extent the recirculatory flow inside the ICA sinus bulb, an unsteady solve captures this recirculatory region more accurately.

Hence, all cases are solved using an unsteady non-Newtonian pulsatile flow coupled with the Pressure-Implicit with Splitting of Operators (PISO) pressure-velocity coupling scheme. A time step of 0.0001s is employed and thus 9170 time steps were required to simulate one complete pulse at the inflow. The full details of this are outlined in Appendix C.3.

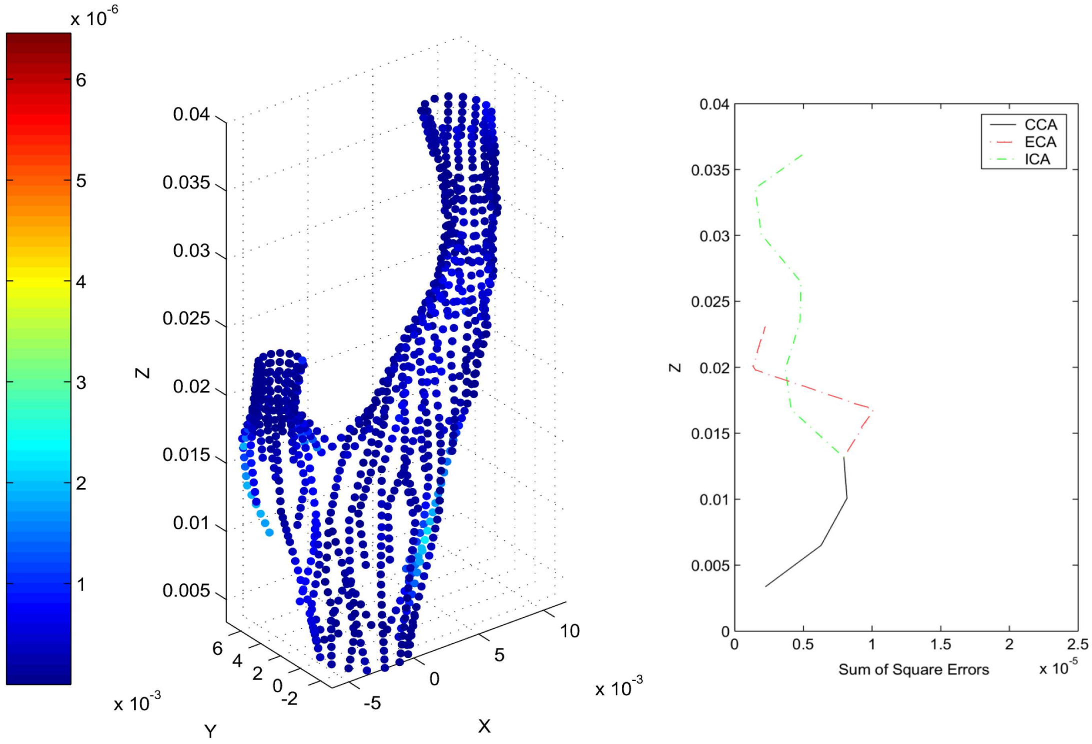


FIGURE 7.14: Square error between CAD and real geometry shown on the left along with the sum of square errors on the intersecting xy-planes shown on the right, after Bump 3

Two metrics are considered for comparison. First,

$$w_A = \int_S \mathrm{d}A_- \tag{7.2}$$

where w_A is the area of the negative wall shear stress, $\mathrm{d}A_-$ is the area vector of a cell for all values of negative time-averaged shear stress $\bar{\tau}(\mathrm{Pa})$ and S represents the artery inner wall surface. And second,

$$\tilde{\tau} = \frac{|\sum_i \gamma_i (\bar{\tau}_-)_i \mathrm{d}A_i|)}{\sum_i \gamma_i \mathrm{d}A_i} \tag{7.3}$$

integrated across all wall mesh faces with incremental area $\mathrm{d}A_i$, where for the ith element

$$(\bar{\tau}_-)_i = \frac{\sum_N |\tau_w|(-1)^j}{N} \tag{7.4}$$

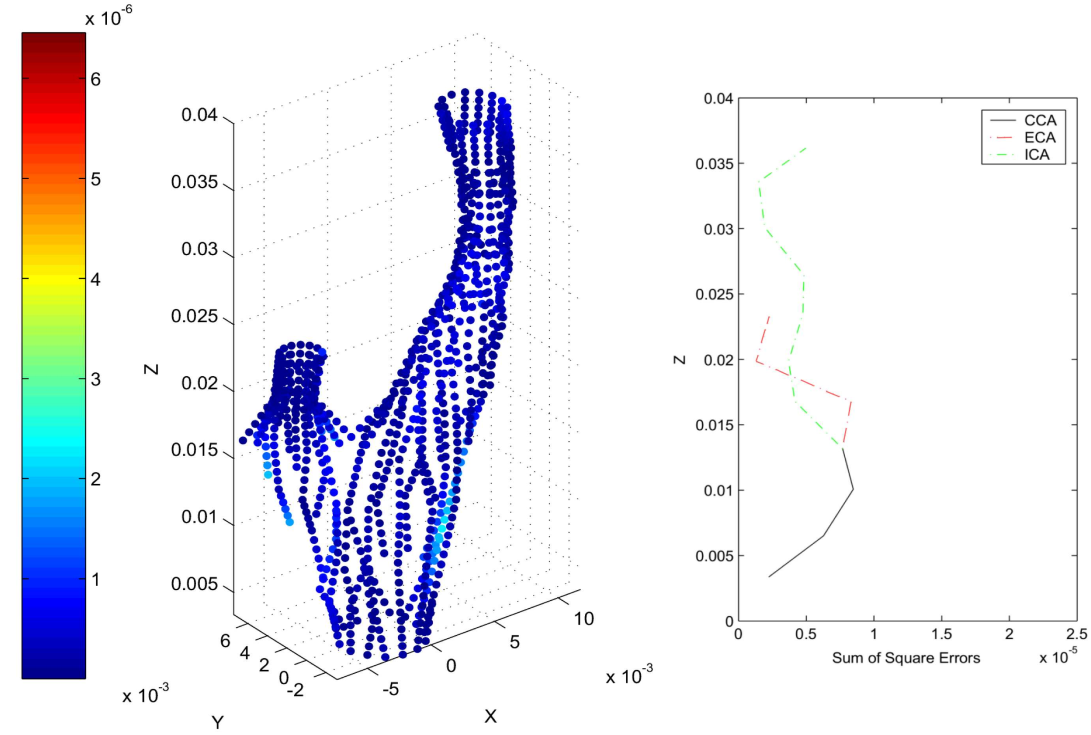


Figure 7.15: Square error between CAD and real geometry shown on the left along with the sum of square errors on the intersecting xy-planes shown on the right, after Bump 4

signifies the average value of the wall shear stress magnitude, $|\tau_w|$, across N time steps. If the y-component of $|\tau_w|$ is negative, $j = 1$, and if the y-component of $|\tau_w|$ is positive, $j = 0$. Furthermore, γ_i is defined as

$$\gamma_i = \begin{cases} 1 & \text{if } (\bar{\tau}_{-})_i < 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

τ_w is the wall shear stress defined by

$$\tau_w = \eta \frac{\partial u}{\partial \hat{\mathbf{n}}} . \hat{\mathbf{t}} \tag{7.6}$$

where u is the blood velocity, η the blood viscosity and $\hat{\mathbf{n}}$, $\hat{\mathbf{t}}$ normal and tangential unit vectors on the inner artery wall respectively.

The comparison of results from the CFD simulations can be seen in Table 7.4. The area of negative shear regions, w_A, has improved from a 53% difference between the

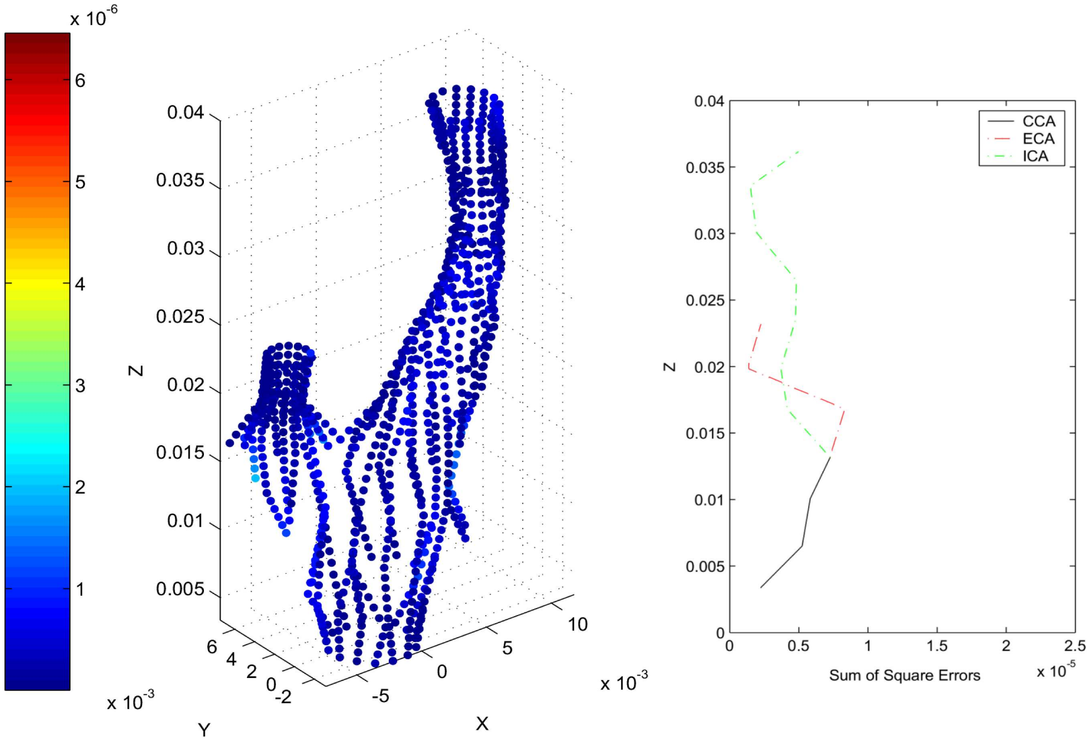


Figure 7.16: Square error between CAD and real geometry shown on the left along with the sum of square errors on the intersecting xy-planes shown on the right, after Bump 5

	w_A (m^2)	$\tilde{\tau}$ (Pa)
After Stage 1	71.95	3.41
After Bump 6 of Stage 2	123.96	7.85
Real	153.69	7.20

Table 7.4: Comparison of w_A and $\tilde{\tau}$ values after the CFD simulations through the artery after Stage 1, Stage 2 and the real geometry

Stage 1 geometry and the real geometry to a 19% difference between the final Stage 2 geometry after 6 bumps and the real geometry. Meanwhile $\tilde{\tau}$, indicating the extent of the recirculation region averaged over one pulse, has improved from a 52% difference to a 9% difference after Bump 6 of Stage 2. Contour plots of the time-averaged wall shear stress, $\tilde{\tau}$, for each of the Stage 1, Stage 2 after Bump 6 and real geometries can be seen in Figures 7.22, 7.23 and 7.24 respectively.

To show the negative regions of $\tilde{\tau}$, a magnification of the bifurcation region shows the improvement of the match between the CAD geometries after Stage 1 and after Stage 2 in Figures 7.25 and 7.26 respectively, and the real geometry, in Figure 7.27. For the geometry after Stage 2, the negative wall shear stress experienced is much more similar to

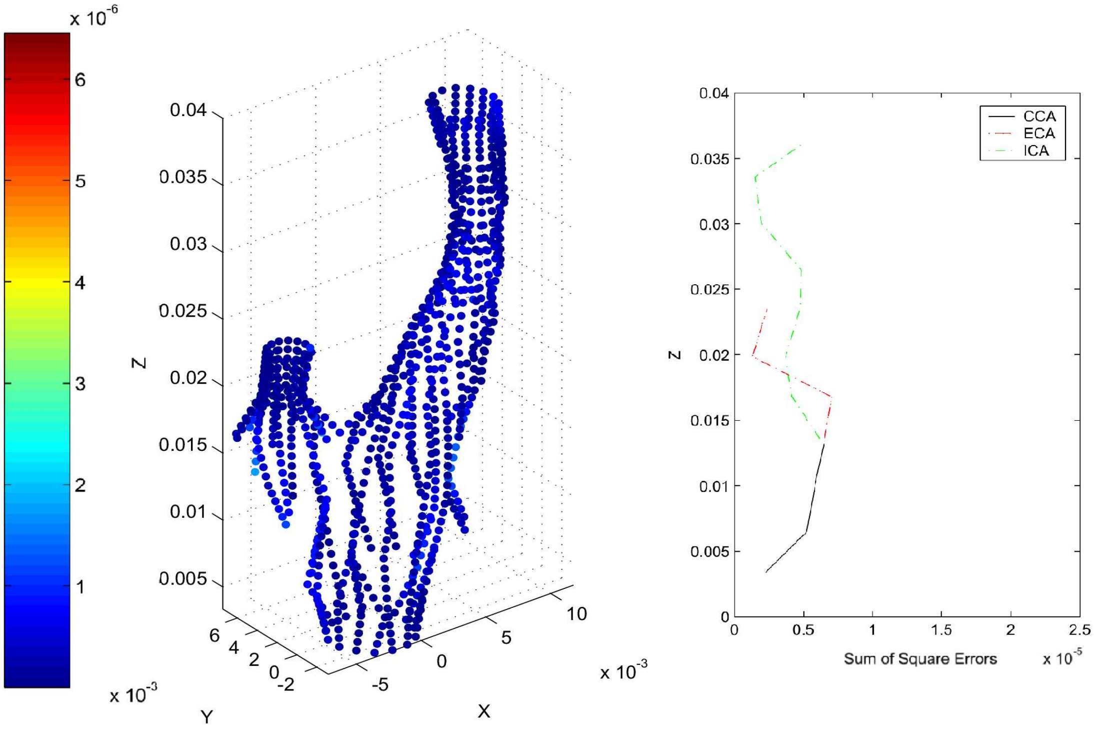


FIGURE 7.17: Square error between CAD and real geometry shown on the left along with the sum of square errors on the intersecting xy-planes shown on the right, after Bump 6

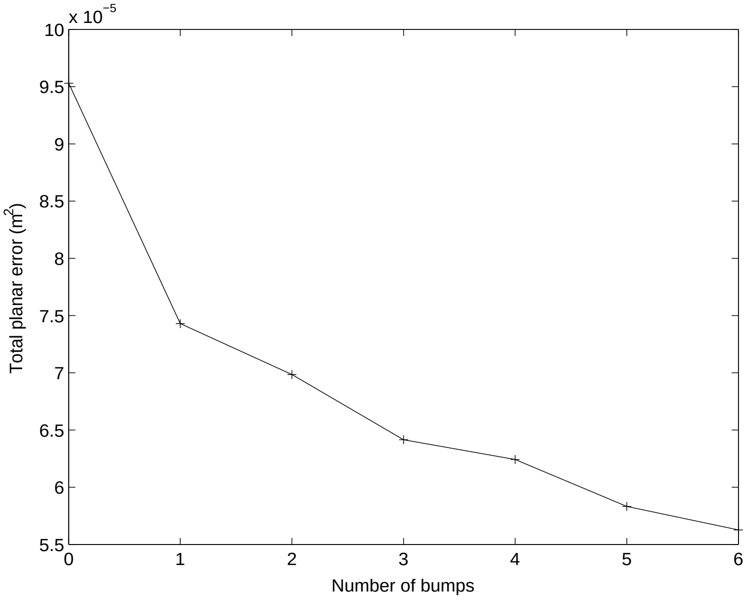


FIGURE 7.18: Progression of the total error after each bump

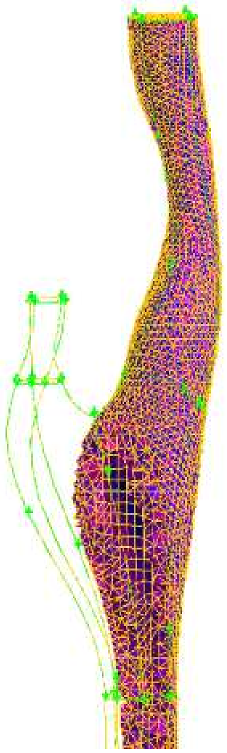


FIGURE 7.19: Slice of $\sim$ 65000 cell mesh, cut through the yz-plane to reveal the hex core

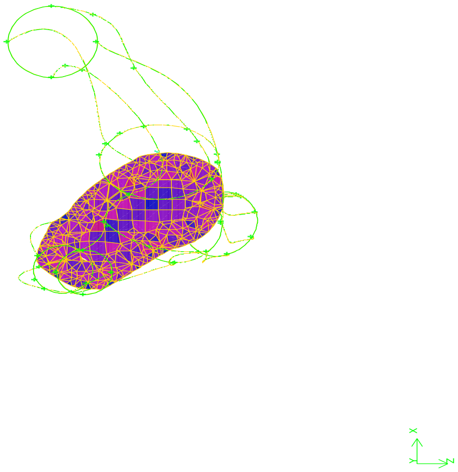


FIGURE 7.20: Slice of $\sim$ 65000 cell mesh, cut through the xy-plane to reveal the hex core

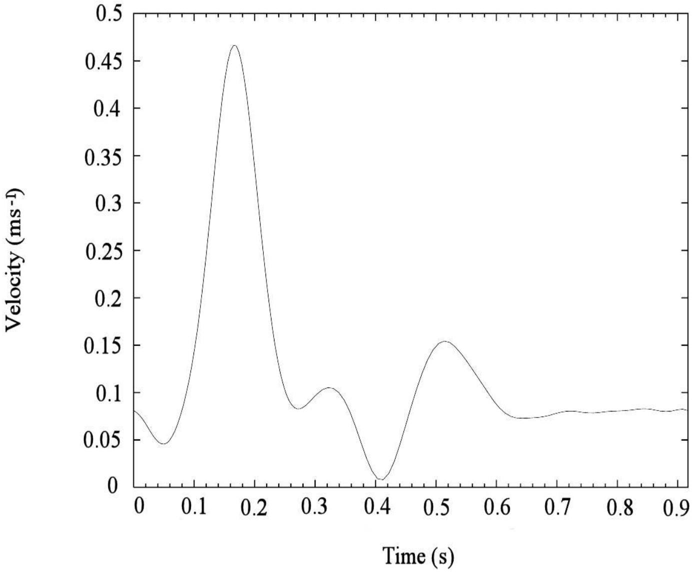


FIGURE 7.21: Velocity inflow waveform at inlet to the CCA to simulate human pulsatile blood flow

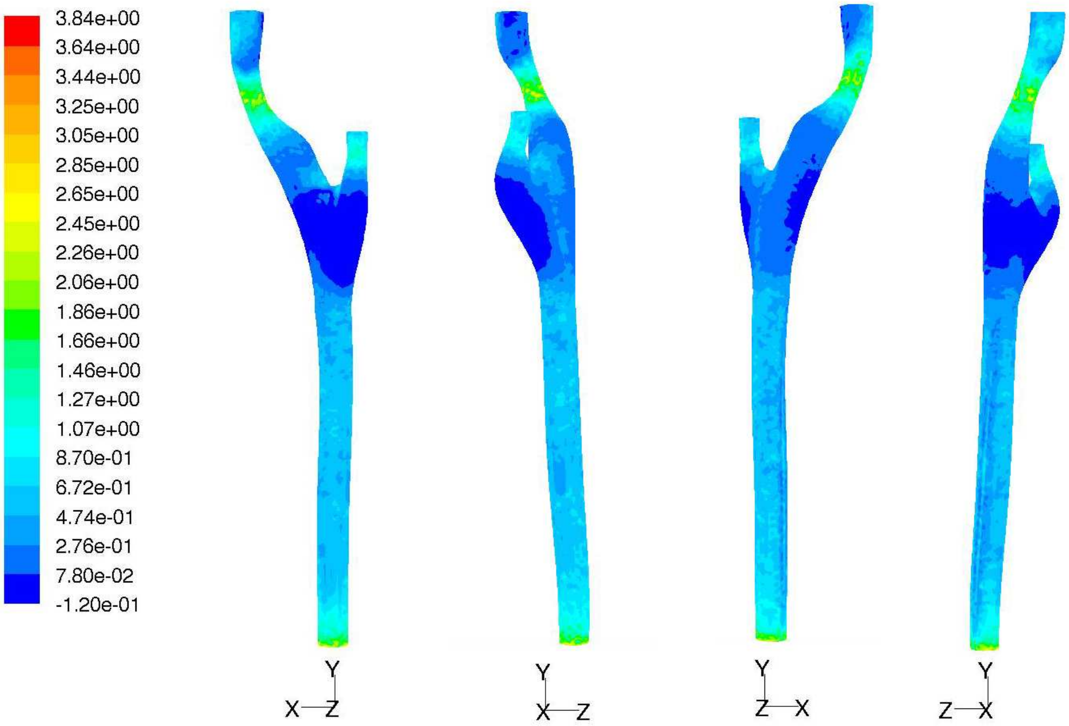


FIGURE 7.22: $\tilde{\tau}$ shown on the CAD geometry after Stage 1

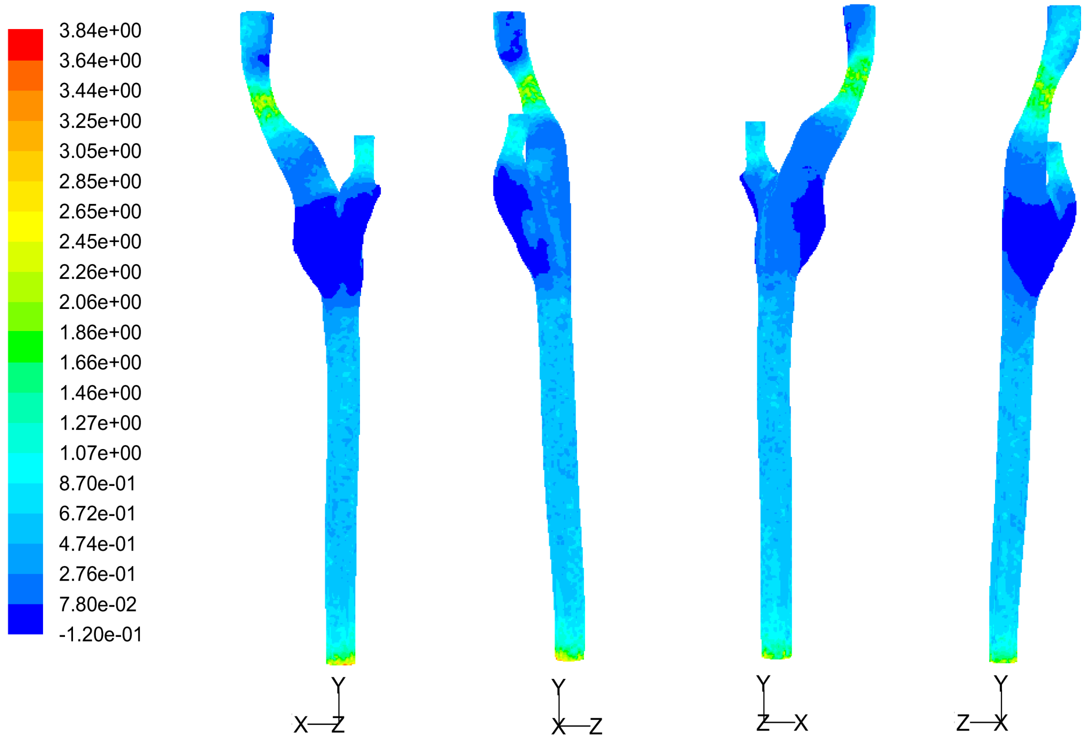


FIGURE 7.23: $\tilde{\tau}$ shown on the CAD geometry after Stage 2

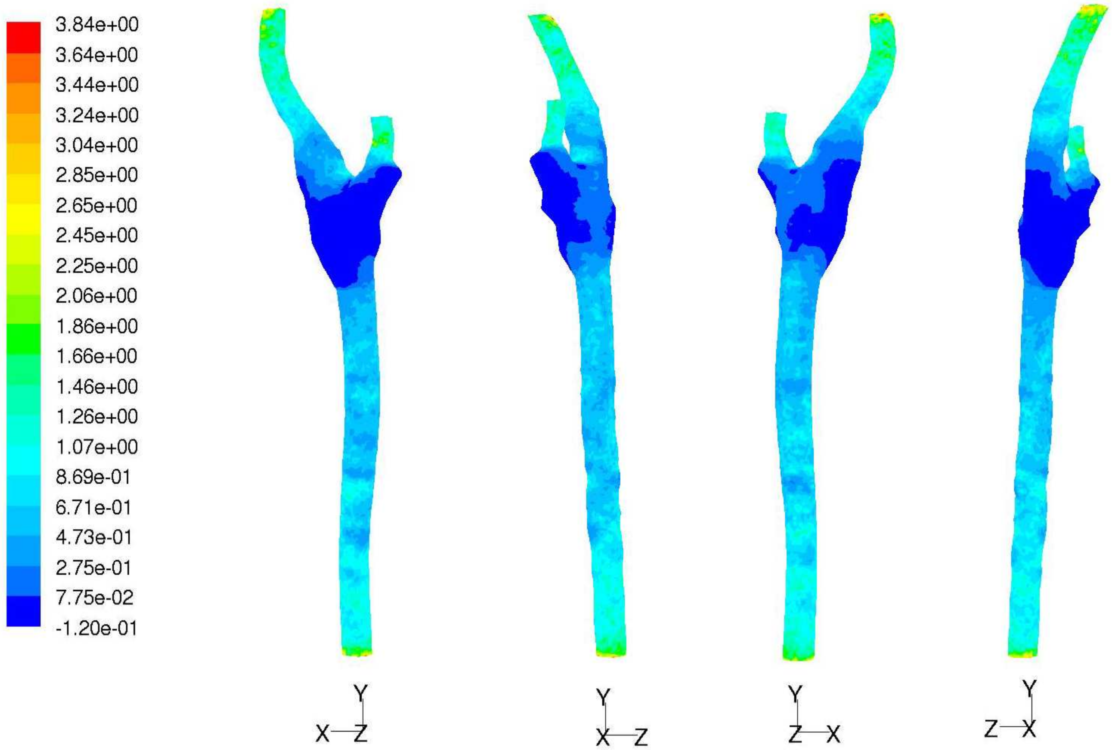


FIGURE 7.24: $\tilde{\tau}$ shown on the real artery geometry

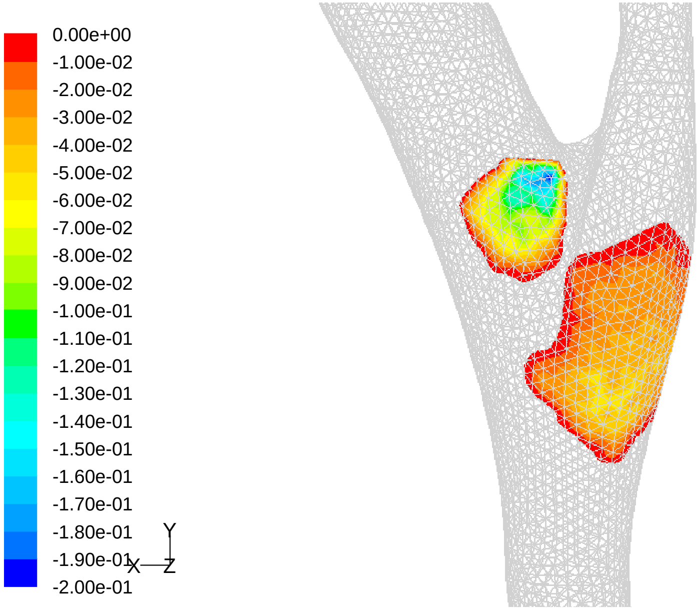


FIGURE 7.25: $\tilde{\tau} < 0$ shown on the CAD geometry after Stage 1

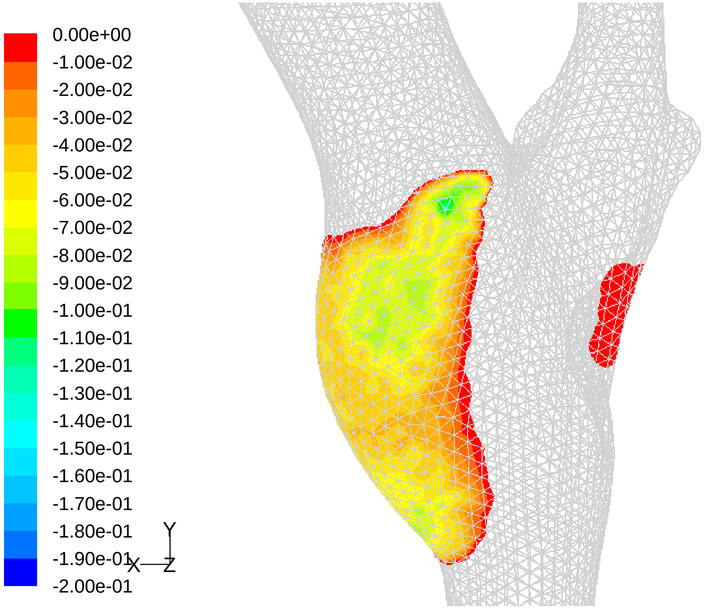


FIGURE 7.26: $\tilde{\tau} < 0$ shown on the CAD geometry after Stage 2

that of the real geometry than the region of negative wall shear stress experienced by the smooth idealised CAD model after Stage 1. Although not exact, due to the limitations of the CAD engine requiring the designer to keep the limit curve fixed throughout Stage 2, it is a much more accurate representation of a parametric artery in comparison to the idealised parametric carotid artery bifurcation models available in the field to date.

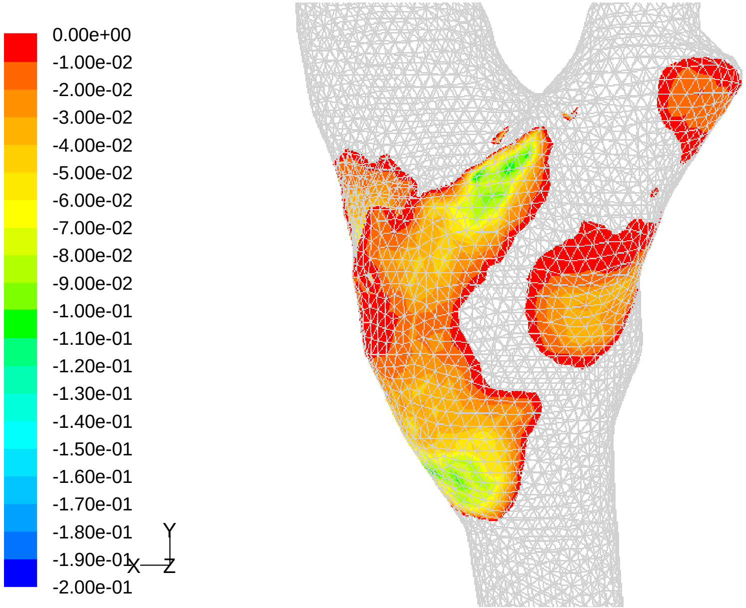


FIGURE 7.27: $\tilde{\tau} < 0$ shown on the real artery geometry

7.3 Summary

In this chapter the AMSSOD process has been implemented to provide a patient-realistic parametric representation of the human carotid bifurcation. It has shown to be beneficial and to work effectively given a suitable problem. An initial parametric geometry was constructed with diameter values, branch angles and values describing the non-planarity and tortuosity taken from an automated analysis of real artery data. This geometry served as the Stage 1 geometry. From here, the error between this initial CAD geometry and the real artery geometry was calculated. Deformations to the surface were made in order to improve the accuracy of the CAD model at the regions of greatest error. In total, six deformations in Stage 2 were performed and a significant improvement in accuracy has been highlighted both in terms of geometrical error and with respect to CFD results for the parametric model compared to the target artery data.

Chapter 8

Discussion and Conclusion

In modern engineering design search and optimization, geometry parameterization plays a key role in determining the design capability of the optimization process, and has a significant impact on computational efficiency. Early primitive forms of optimization processes were developed for the aerospace industry by Orville and Wilbur Wright. The improvements to the design of their aeroplane based upon experimental testing led to the first powered flight at Kitty Hawk in December 1903. In terms of internal fluid flow applications, curve contouring techniques were put in place to begin the improvement of diffuser efficiency in the early 1900s, but these relied on analytical flow solutions (Patterson, 1938). With the invention and widespread uptake of digital computers, various parametric techniques to efficiently define aerodynamic bodies for optimization were developed; for example, Hicks and Henne presented a method based on surface patches in 1978. Providing an efficient optimization process through parametric design soon caught on and has provided a field of research that has grown enormously to the present day and can now be seen across many diverse engineering communities (Siddall, 1982).

Despite the plethora of research into parameterization methods, the most appropriate construction of a parametric geometry to allow for efficient optimization studies has proved to be an enigma for many internal fluid flow applications. The intention of the research documented in this thesis has been to investigate techniques which provide a large amount of freedom to generate radical shapes for a particular internal flow topology while retaining an efficient optimization process. All research has been performed within commercially available software using practicable computational resources in order to offer industries a realistic solution. This has led to a general-purpose process comprising a multi-stage parameterization and optimization framework, providing the ability to perform strong shape control in tandem with an efficient convergence to an optimal design. The remainder of this chapter summarizes the progress of the research presented in this thesis, highlighting the contributions to the field that this research has made and relating any shortcomings. Areas of further work highlighted by this thesis are also discussed.

8.1 Hitherto...

The research presented in this thesis has, at its core, the use of commercial software tools. It aims to confirm the practicalities of using commercial software in the engineering field and to develop its uses further in design optimization. Optimization frameworks embedded in automated architectures are commonly used for engineering problems. An outline of the GEODISE automated architecture used throughout this thesis is described in Chapter 2, followed by a description of a typical optimization process. A study of curve parametrization techniques is undertaken as these, more often than not, provide the building blocks for almost all parametric geometry constructions across the industry. Design of Experiment approaches, CFD theory and an introduction to optimization using response surface methodology and convergence to an optimal design using concentrated exploration in a reduced area of the design space for high-dimensional problems are described.

Chapter 3 provides a detailed mathematical description of the response surface modelling approach of Kriging. Kriging is chosen for all optimization studies presented in this thesis due to its versatility in representing complex objective function landscapes. For many internal fluid flow applications, the objective function landscape is not known and so a global optimizer which has the ability to cope with a highly non-linear objective function is favourable. One shortcoming of this method is that Kriging is only workable if the design variable count is low, typically fewer than 20. Compute resources required to build the response surface for problems with greater dimensionality increase substantially, often beyond the compute resources available in a typical industrial environment.

The optimization process is first implemented by testing several parameterization techniques on a straight diffuser and an elbow. This study generated an understanding of the type of optimal designs that materialised for expanding flow and for flow turning through a bend. Although the best designs produced from each individual study presented very similar geometric features, it is clear that a range of parameterization techniques provided good designs with varying design variables. Chapter 4 highlighted the importance of the parameterization technique implemented and its impact on the resulting designs produced. A key factor of any parameterization method is whether it is allowed the freedom to produce a wide variety of geometrical shapes. Significant freedom in this sense suggests the ability to perform strong shape control. A parameterization approach that possesses this ability will be able to produce intuitive shapes and also radical shapes, resulting in a potentially superior objective function value. From the insight gained with these studies, a parametric geometry of a two-dimensional F1 airbox was constructed and optimized with respect to its performance. Radical shapes were produced with high performance values, indicating that the chosen parameterization technique offers strong shape control.

From this, a progression into three-dimensional studies required a parameterization technique capable of producing radical shapes akin to those seen in Chapter 4. As the geometry construction is performed using a commercial CAD package, surface representation must be understood before a survey of surface manipulation techniques is undertaken. Chapter 5 reviews a variety of surface manipulation techniques that may be used in automated optimization processes. Each approach is considered in terms of whether it controls the shape manipulation in a global or a local sense. The idea of providing a multi-stage parameterization technique that can perform both global and then local surface manipulations, optimizing the geometry as a whole, is deemed advantageous so that fine-tuned and perhaps asymmetrical designs can be produced via an optimization process using only a small set of design parameters. Polynomial splines are chosen for the global manipulation approach and three-dimensional Hicks-Henne bump functions are chosen for local surface manipulation. These two approaches are combined together to form a general-purpose automated multi-stage parameterization and optimization framework. In Chapter 5, this automated multi-stage shape optimization with deformation (AMSSOD) process is outlined, and this approach was used in the subsequent optimization studies.

In Chapter 6, the AMSSOD process was performed on a three-dimensional F1 airbox. Stage 1 was carried out effectively and a parametric geometry was constructed using polynomial splines. To test the realistic nature of the flow conditions, the geometry in Stage 1 was optimized with two different flow conditions at the airbox inlet. Stage 2 was implemented starting with the best geometry found after the Stage 1 optimization. The pressure recovery of the best geometry found in Stage 1 could not be improved upon during the Stage 2 process indicating that the geometry found in Stage 1 was a near optimal design. Simplifying the Stage 1 parameterization to reduce the shape control may allow the Stage 2 process to work to its best ability in finding an optimal design through local deformation. The Stage 2 process was further tested on a non-optimal geometry from Stage 1 and, after this Stage 2 process, a design with deformation was produced returning an improved pressure recovery. From this study, it was also concluded that for internal fluid flow applications where the best location of a deformation is unknown, the Stage 2 process should focus on optimizing the deformation position, height and curvature in a reduced area of the design space around a favourable area chosen from the initial Stage 2 DoE.

Following this, a completely different application was chosen to illustrate the generic capability of the AMSSOD process: the human carotid artery bifurcation. In this case, one of the criticisms of computational research in this field is the idealised nature of parametrically defined computational arteries used in which many of the important geometrical details are not captured. Here, instead of optimizing the arterial shape for a performance metric, the error between a parametric CAD model and a real artery geometry was minimized using the AMSSOD process to provide a realistic parametric artery which may be used for further research. The Stage 1 geometry was found via an automated analysis of real artery data to extract key geometrical features. Stage 2 then deformed the artery in the regions of highest error, the majority of which were found in the region of the lofts between the guiding splines defining the ICA and the ECA in the bifurcation root. This process worked effectively for six deformations of the CAD artery model. A limitation of the specific CAD tool being used was deemed responsible in not allowing the arbitrary placement of deformation limit curves once the surface has already been deformed. Thus, the Stage 2 process came to a halt as the regions of highest error lay along this limit curve. The Stage 2 process, however, reduced the error significantly. The improvement was demonstrated by running CFD studies on the geometry produced after each of the two stages, comparing the results with the real artery flow simulation. A dramatic reduction in the difference between the time-averaged negative wall shear stresses was seen as well as a reduction in the difference between the area of the negative wall shear stress, indicating that this process has been successful. The AMSSOD process has been shown to allow complex objects to be parameterized with only a small number of design variables, and can be used with many more optimization studies performed on internal fluid flow applications.

In summary, the main contribution that this thesis makes is in the development of a multi-stage parameterization process utilising a geometric tool for local surface deformation for use in the optimization of 3D shape applications.

Additionally, the work of this thesis has:

- provided an in-depth analysis of parameterization techniques for two-dimensional and three-dimensional shape optimization studies;
- highlighted the current capabilities and limitations for local parametric deformations available in commercial CAD software packages;
- investigated the use of curve parameterization techniques to provide a novel geometry parameterization for the design optimization of a 2D airbox;
- developed an automated multi-stage process for use in design optimization studies that works wholly within existing commercial CAD software, facilitating the file exportation for meshing and CFD analysis;
- demonstrated the capability of the multi-stage parameterization process by developing a parametric model of a 3D F1 airbox, the successful design optimization of which resulted in an improvement in pressure recovery;
- demonstrated the versatility of the multi-stage parameterization process by developing a parametric model of a 3D human carotid bifurcation geometry. The effective use of this parametric model followed by local surface deformations allowed for a successful shape optimization, improving the match between a CAD produced model and a real carotid artery bifurcation.

8.2 Thereafter

Although the multi-stage shape optimization tool has been shown to provide a powerful means to facilitate the optimization of global geometric shape followed by local geometric shape in two optimization studies, the current deficiencies of the CAD engine in relation to the local deformations of surface geometries has restrained the ability to repeatedly apply patches as desired. Therefore, a further investigation into arbitrarily placed limit curves on already deformed geometries could lead to a significant improvement in the results of Stage 2 in the AMSSOD framework.

A simpler Stage 1 parameterization of the three-dimensional F1 airbox discussed in Chapter 6 could lead to an improved performance of the Stage 2 process, where the Stage 2 search is in an area of one favourable deformation location only.

Further to the artery problem studied in Chapter 7, a repeatable process through Stage 1 to sample real artery data at more locations, to increase the number of control points along the splines which define the ICA and ECA, may reduce the error further. The Stage 2 process can then be repeated to provide the correction of the lofting between these guiding splines as previously demonstrated.

Directly following this work, the results presented in Chapter 7 may help to identify patients for whom treatment through interventional medicine is likely to be favourable. Metrics can be developed to further understand the role of elevated shear stress regions and reversed flow in connection with arterial disease sites in a large number of different geometries and corresponding haemodynamic environments.

It is likely that a parametric study, using the technique for geometry construction illustrated in Chapter 7, would be beneficial in exploring the impact of surgical intervention, as well as attempting to further understand the effects of geometrical differences. One example of this is the question of how the geometrical difference between normal arteries and arteries of certain diabetic patients affects the pre-disposition of lower leg iscaemia.

Furthermore, the parametric definition of patient specific arteries could be used to research the effect of applying anastomosis to a diseased artery. Although stent insertion is a popular form of treatment of the carotid artery and other arteries particularly around the heart, anastomosis or bypassing of the artery is also common. Providing research on the haemodynamic effects of a bypass would benefit surgeons in helping to optimize the benefit of treatment.

Appendix A

Kriging Theory

Kriging is what is known by statisticians as a Gaussian stochastic process (Doob, 1934). The word 'stochastic' is of Greek origin meaning "pertaining to chance" and is synonymous with 'random'. A stochastic process can then be defined as a process to approximate unsampled points using random variables (Wolfram, 2002), in this case corresponding to the set of responses. This approximation is defined by a response surface model.

Kriging, named after its inventor Krige (1951), is a technique first developed for use in geology. Its original purpose was to use prior knowledge about the spatial distribution of a mineral within a given sample space to predict the level of mineral concentration at unsampled points.

What follows are the mathematical derivations of three different types of updating the RSM; the first updates the RSM at an unsampled points where the maximum likelihood of the predicted objective function value is highest (section A.1), the second updates the RSM at an unsampled point where the error of the prediction is highest (section A.2), and the third updates the RSM at an unsampled point where the expectation of the improvement of the objective function is highest (section A.3).

A.1 Maximum Likelihood

A vector of n initial sample points is found using a Design of Experiments. A sample at an untried point $\mathbf{x}^*$ is then needed at which the objective function value or response $y(\mathbf{x}^*) = f(\mathbf{x}^*)$, where f is the objective function is uncertain. This uncertainty can be represented by associating $y(\mathbf{x}^*)$ with a Normal distribution $N(\mu, \sigma^2)$. Before the analysis code is called and the untried $\mathbf{x}^*$ sampled, the relation of the untried point to the n previously sampled points is considered. Assuming continuity of f, the difference between the responses $y(\mathbf{x}_i)$ and $y(\mathbf{x}_j)$ will be small if the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is small. This has a statistical interpretation being that $y(\mathbf{x}_i)$ and $y(\mathbf{x}_j)$ are highly correlated if $\|\mathbf{x}_i - \mathbf{x}_j\|$ is small. This correlation is expressed as

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{s=1}^{k} \theta_l |x_{is} - x_{js}|^{p_s} \right), \tag{A.1}$$

satisfying $R = 1$ if $\mathbf{x}_i = \mathbf{x}_j$.

The correlation matrix is a square $n \times n$ matrix $\mathbf{R}$ in the form

$$\mathbf{R} = \begin{pmatrix} 1 & R(\mathbf{x}_1, \mathbf{x}_2) & \cdots & R(\mathbf{x}_1, \mathbf{x}_n) \\ R(\mathbf{x}_2, \mathbf{x}_1) & 1 & & R(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & & \ddots & \vdots \\ R(\mathbf{x}_n, \mathbf{x}_1) & \cdots & \cdots & 1 \end{pmatrix}. \tag{A.2}$$

To summarize, for any random variable $Y = \{Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n)\}^T$, Y has a mean of $\mathbf{1}\mu$ and a covariance $Cov(Y) = \sigma^2\mathbf{R}$. Y now depends on the parameters μ, σ^2, θ_s and p_s ($s = 1, \ldots, k$). To estimate the values of μ, σ^2, θ_s and p_s values for these parameters are chosen which will maximize the likelihood of the responses.

Suppose there are a set of responses

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \tag{A.3}$$

then the likelihood can be defined as the hypothetical probability that an event which has already occurred would yield a specific outcome (Fisher, 1912, 1921, 1922; Edwards, 1997). Each y_i $(i = 1, \ldots, n)$ has a probability density function

$$\mathrm{pdf}(\mathbf{y};\mathbf{V}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right), \tag{A.4}$$

where the vector $\mathbf{V}$ contains the two unknown parameters μ and σ^2. The likelihood, L, may therefore be written as

$$\begin{aligned} L(\mathbf{V}) &= \prod_1^n \mathrm{pdf}(\mathbf{y};\mathbf{V}) \\ &= \prod_{i=1}^{n} \left[\tfrac{1}{(\sigma^2)^{\frac{1}{2}}(2\pi)^{\frac{1}{2}}} \exp\left(-\tfrac{(y_i-\mu)^2}{2\sigma^2}\right)\right] \\ &= \tfrac{1}{(\sigma^2)^{\frac{n}{2}}(2\pi)^{\frac{n}{2}}} \prod_{i=1}^{n} \exp\left(-\tfrac{(y_i-\mu)^2}{2\sigma^2}\right) \\ &= \tfrac{1}{(\sigma^2)^{\frac{n}{2}}(2\pi)^{\frac{n}{2}}} \exp\left(-\textstyle\sum_{i=1}^{n} \tfrac{(y_i-\mu)^2}{2\sigma^2}\right), \end{aligned}$$

and assuming the y_is are independent,

$$= \tfrac{1}{(\sigma^2)^{\frac{n}{2}}(2\pi)^{\frac{n}{2}} \det \mathbf{R}^{\frac{1}{2}}} \exp\left(-\tfrac{(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right). \tag{A.5}$$

In practice, however, it is more convenient to choose the parameters to maximize the log-likelihood function $\ln L(\mathbf{V})$ where

$$\ln L(\mathbf{V}) = -\frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log(|\mathbf{R}|) - \frac{(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2} + \text{constant terms.} \tag{A.6}$$

The maximum likelihood estimators $\hat{\mu}$ and $\hat{\sigma}^2$ of the respective parameters μ and σ^2 are those functions of $y_1, \ldots, y_n$ which maximize the log-likelihood function. Mathematically, $\hat{\mathbf{V}} = \hat{\mathbf{V}}(y_1, \ldots, y_n)$ is such that

$$L(\hat{\mathbf{V}}) = \sup_{\mathbf{V}} L(\mathbf{V}) \quad \text{for all responses } \mathbf{y}. \tag{A.7}$$

Usually the maximum occurs at a unique point in the parameter space and does not lie on the boundary. Then

$$\frac{\partial(\ln L(\mathbf{V}))}{\partial V_i} = 0, \quad i = 1, 2 \ , \tag{A.8}$$

is solved where $V_1 = \mu$ and $V_2 = \sigma^2$. It must also be assumed that the domain in which pdf is non-zero does not depend on either μ or σ^2.

Hence, the maximum likelihood estimators $\hat{\mu}$ and $\hat{\sigma}^2$ can be derived as follows:

$$\frac{\partial(\ln L(\mathbf{V}))}{\partial \mu} = 0 = \frac{\mathbf{1}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}$$

$$\Rightarrow \qquad \mathbf{1}^T\mathbf{R}^{-1}\mathbf{y} - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}\mu = 0$$

$$\Rightarrow \qquad \hat{\mu} = \sup_{\mu} \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}$$

$$\Rightarrow \qquad \hat{\mu} = \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} \tag{A.9}$$

$$\frac{\partial(\ln L(\mathbf{V}))}{\partial \sigma^2} = 0 = -\frac{n}{2\sigma^2} + \frac{(\mathbf{y} - \mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^4}$$

$$\Rightarrow \qquad -n + \frac{(\mathbf{y} - \mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{\sigma^2} = 0$$

$$\Rightarrow \qquad \hat{\sigma}^2 = \sup_{\sigma^2} \frac{(\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{n}$$

$$\Rightarrow \qquad \hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{n}. \tag{A.10}$$

By substituting equations A.9 and A.10 into equation A.6 the concentrated log-likelihood function

$$\text{CLLF} = -\frac{n}{2} \log(\hat{\sigma}^2) - \frac{1}{2} \log(|\mathbf{R}|) + \text{constant terms}. \tag{A.11}$$

is obtained. This concentrated log-likelihood function can be maximized to find the estimators and $\hat{\theta}_s$ and $\hat{p}_s$ $(s = 1, \ldots, k)$ by equating the derivative of equation A.11 with respect to θ_s to zero and similarly equating the derivative of equation A.11 with respect to p_s to zero. These estimates can then be used to compute $\mathbf{R}$ and subsequently, to compute the values of $\hat{\mu}$ and $\hat{\sigma}$.

To arrive at a prediction for the objective function at some untried point $\mathbf{x}^*$ an objective function value is estimated and augmented to the initial n-dimensional data set. It is ascertained, in the following steps, how consistent the guestimated value of the objective function at the untried point is with the already observed pattern of variation between data points and their responses. An intuitive predictor of $y^* = y(\mathbf{x}^*)$ would be the value which maximizes the augmented log-likelihood function.

Let $\tilde{\mathbf{y}} = (\mathbf{y}^T \quad y_{n+1}(\mathbf{x}^*))^T$ be the augmented $(n+1)$-dimensional vector of responses and $\mathbf{r}$ be the vector of correlations

$$\mathbf{r} = \begin{pmatrix} R(\mathbf{x}^*, \mathbf{x}_1) \\ \vdots \\ R(\mathbf{x}^*, \mathbf{x}_n) \end{pmatrix}. \tag{A.12}$$

The new correlation matrix now becomes

$$\tilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix}, \tag{A.13}$$

and the augmented log-likelihood function becomes

$$L(y^*) = -\frac{(\tilde{\mathbf{y}} - \mathbf{1}\hat{\mu})^T \tilde{\mathbf{R}}^{-1} (\tilde{\mathbf{y}} - \mathbf{1}\hat{\mu})}{2\sigma^2} + \text{terms independent of } y^* \tag{A.14}$$

$$= \frac{-\begin{pmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ y^* - \hat{\mu} \end{pmatrix}^T \begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{y} - \mathbf{1}\hat{\mu} \\ y^* - \hat{\mu} \end{pmatrix}}{2\hat{\sigma}^2} + \text{terms independent of } y^*. \tag{A.15}$$

Using Theil's partitioned inverse formula

$$\begin{pmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{r}^T & 1 \end{pmatrix}^{-1} = \left(\begin{array}{c|c} \mathbf{R}^{-1} + \mathbf{R}^{-1}\mathbf{r}(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})^{-1}\mathbf{r}^T\mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{r}(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})^{-1} \\ \hline -(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})^{-1}\mathbf{r}^T\mathbf{R}^{-1} & (1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})^{-1} \end{array} \right). \tag{A.16}$$

The augmented log-likelihood therefore becomes

$$L(y^*) = \left[\frac{-1}{2\hat{\sigma}^2(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})}\right](y^* - \hat{\mu})^2 + \left[\frac{\mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})}\right](y^* - \hat{\mu}) + \text{terms independent of } y^*. \tag{A.17}$$

By taking the derivative with respect to y^* and equating to zero

$$\left[\frac{-1}{\hat{\sigma}^2(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})}\right](y^* - \hat{\mu})^{+} \left[\frac{\mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r})}\right] = 0. \tag{A.18}$$

Solving this for y^* then gives the Kriging predictor

$$\hat{y}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}). \tag{A.19}$$

A.2 Prediction Error – A Gaussian Approach

In this section, the mean square prediction error for an interpolating model is derived fully from first principles. The adjustment made for a regressing model is shown in Section 3.2.1.4.

Given a set of designs $\mathbf{S} = \{s_1, \ldots, s_n\}$ and data (responses) $\mathbf{y_s} = \{y(s_1), \ldots, y(s_n)\}$ one can consider the linear predictor $\hat{y}(\mathbf{x}) = \mathbf{c}(\mathbf{x})^T \mathbf{y}_s$ of $y(\mathbf{x})$ at a arbitrary untried $\mathbf{x}$. For an interpolating model the correlation matrix is defined by

$$\mathbf{R} = \begin{pmatrix} 1 & R(\mathbf{x}_1, \mathbf{x}_2) & \cdots & R(\mathbf{x}_1, \mathbf{x}_n) \\ R(\mathbf{x}_2, \mathbf{x}_1) & 1 & & R(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & & \ddots & \vdots \\ R(\mathbf{x}_n, \mathbf{x}_1) & \cdots & \cdots & 1 \end{pmatrix}. \tag{A.20}$$

Replace $\mathbf{y}_s$ by corresponding random quantity $\mathbf{Y} = \{Y(s_1), \ldots, Y(s_n)\}^T$. Treat $\hat{y}(\mathbf{x})$ as random and compute the mean square error (MSE) of the predictor averaged over a random process. The best linear unbiased predictor (BLUP) is obtained by choosing an $n \times 1$ vector $\mathbf{c}(\mathbf{x})$ to minimize the MSE.

Let

$$\mathbf{F} = \begin{Bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_n) \end{Bmatrix}, \tag{A.21}$$

and let $\mathbf{r}_x = [R(\mathbf{x}_1, \mathbf{x}), \ldots, R(\mathbf{x}_n, \mathbf{x})]^T$ for untried $\mathbf{x}$ as described earlier. The stochastic model can be written as

$$\mathbf{Y} = \mathbf{F}\mu + \mathbf{z} \tag{A.22}$$

where $\mathbf{z} = (Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_n))^T$ is the error function in the stochastic process and μ is the mean of the observed responses.

Also define $\mathbf{A} = (A_1, \ldots, A_{n+1})$ so that the linear combinations $\bar{Y}$ and $\bar{Z}$ can be defined by $\bar{Y} = \sum_i A_i Y(\mathbf{x}_i)$ and $\bar{Z} = \sum_i A_i Z(\mathbf{x}_i)$ where each A_i represents a weight. Define $\bar{f} = \sum_i A_i f_{x_i}$ and $\bar{\mathbf{r}} = \sum_i A_i \mathbf{r}_{x_i}$.

Then, for any linear predictor $\mathbf{c}(\mathbf{x})^T\mathbf{Y}$ of $\bar{Y}(\mathbf{x})$, the MSE of the predictor is:

$$\begin{aligned}
MSE &= E\left(\mathbf{c}(\mathbf{x})^T\mathbf{Y} - \bar{Y}(\mathbf{x})\right)^2 \\
&= E\left[\mathbf{c}(\mathbf{x})^T\mathbf{Y}\mathbf{Y}^T\mathbf{c}(\mathbf{x}) + \bar{Y}(\mathbf{x})^2 - 2\mathbf{c}(\mathbf{x})^T\mathbf{Y}\bar{Y}(\mathbf{x})\right] \\
&= E\left[\mathbf{c}(\mathbf{x})^T(\mathbf{F}\mu + \mathbf{z})(\mathbf{F}\mu + \mathbf{z})^T\mathbf{c}(\mathbf{x}) + (\bar{f}\mu + \bar{Z})^2 - 2\mathbf{c}(\mathbf{x})^T(\mathbf{F}\mu + \mathbf{z})(\bar{f}\mu + \bar{Z}(\mathbf{x}))\right] \\
&= \mathbf{c}(\mathbf{x})^T\mathbf{F}\mu\mathbf{c}(\mathbf{x})^T\mathbf{F}\mu - 2\mathbf{c}(\mathbf{x})^T\mathbf{F}\mu\bar{f}\mu + \bar{f}\mu\bar{f}\mu + \mathrm{cov}(\mathbf{c}(\mathbf{x})^T\mathbf{Y}, \mathbf{c}(\mathbf{x})^T\mathbf{Y}) \\
&\qquad + \mathrm{cov}(\bar{Y}, \bar{Y}) - 2\mathrm{cov}(\mathbf{c}(\mathbf{x})^T\mathbf{Y}, \bar{Y}) \\
&= (\mathbf{c}(\mathbf{x})^T\mathbf{F}\mu - \bar{f}^T\mu)^2 + \mathbf{c}(\mathbf{x})^T\hat{\sigma}^2\mathbf{R}\mathbf{c}(\mathbf{x}) + \mathbf{A}^T\mathbf{R}\mathbf{A}\hat{\sigma}^2 - 2\mathbf{c}(\mathbf{x})^T\hat{\sigma}^2\bar{\mathbf{r}} \\
&= \hat{\sigma}^2\left[\mathbf{A}^T\mathbf{R}\mathbf{A} + \mathbf{c}(\mathbf{x})^T\mathbf{R}\mathbf{c}(\mathbf{x}) - 2\mathbf{c}(\mathbf{x})^T\bar{\mathbf{r}}\right],
\end{aligned} \tag{A.23}$$

(following the proof given by Schonlau (1997) for the error estimation of a linear combination of Y_i's), subject to the constraint $\mathbf{F}^T\mathbf{c}(\mathbf{x}) = \bar{f}$, which itself follows from $E(\mathbf{c}(\mathbf{x})^T\mathbf{Y}) = \mathbf{c}(\mathbf{x})^T\mathbf{F}\mu$ and $E(Y(\mathbf{x})) = \bar{f}^T\mu \ \forall\ \mu$.

In this case, however, the focus is on one Y_i rather than a linear combination and hence $\mathbf{A}^T\mathbf{R}\mathbf{A} = 1$, $\bar{f} = f_x$ and $\bar{\mathbf{r}} = \mathbf{r}_x$.

Now let us introduce Lagrange multipliers for the unbiased constraint equation $\mathbf{F}^T\mathbf{c}(\mathbf{x}) = f(x)$.

LEMMA: To minimize a function $\varphi(\mathbf{x})$ subject to a constraint $\psi(\mathbf{x}) =$ constant, one looks for the solutions of $\nabla\varphi(\mathbf{x}) = \lambda\psi(\mathbf{x})$ where ∇ is the gradient with respect to the variable chosen to minimize $\varphi(\mathbf{x})$.

In this case

$$\varphi(\mathbf{x}) = \hat{\sigma}^2\left[1 + \mathbf{c}(\mathbf{x})^T\mathbf{R}\mathbf{c}(\mathbf{x}) - 2\mathbf{c}(\mathbf{x})^T\mathbf{r}_x\right] \tag{A.24}$$

and

$$\psi(\mathbf{x}) = \mathbf{F}^T\mathbf{c}(\mathbf{x}) = f_x. \tag{A.25}$$

For $\nabla = \frac{\partial}{\partial \mathbf{c}(\mathbf{x})}$, the equation $\nabla\varphi(\mathbf{x}) = \lambda\psi(\mathbf{x})$ becomes

$$\hat{\sigma}^2\mathbf{R}\mathbf{c}(\mathbf{x}) - \hat{\sigma}^2\mathbf{r}_x = \lambda\mathbf{F} \tag{A.26}$$

$\Rightarrow$

$$\hat{\sigma}^2\mathbf{R}\mathbf{c}(\mathbf{x}) - \hat{\sigma}^2\mathbf{r}_x - \lambda\mathbf{F} = 0, \tag{A.27}$$

which together with the constraints there are a system of equations:

$$\begin{aligned} \hat{\sigma}^2\mathbf{R}\mathbf{c}(\mathbf{x}) - \lambda\mathbf{F} &= \hat{\sigma}^2\mathbf{r}_x \\ \mathbf{F}^T\mathbf{c}(\mathbf{x}) &= f_x. \end{aligned} \tag{A.28}$$

Hence the BLUP must satisfy

$$\begin{pmatrix} 0 & \mathbf{F}^T \\ \mathbf{F} & \hat{\sigma}^2\mathbf{R} \end{pmatrix} \begin{pmatrix} -\lambda \\ \mathbf{c}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} f_x \\ \hat{\sigma}^2\mathbf{r}_x. \end{pmatrix} \tag{A.29}$$

Recall, the BLUP

$$\hat{y}(\mathbf{x}) = \mathbf{c}(\mathbf{x})^T\mathbf{y}_s. \tag{A.30}$$

Now, $\mathbf{c}(\mathbf{x})^T\mathbf{y}_s$ can be written as

$$\mathbf{c}(\mathbf{x})^T\mathbf{y}_s = \begin{pmatrix} -\lambda \\ \mathbf{c}(\mathbf{x}) \end{pmatrix}^T \begin{pmatrix} 0 \\ \mathbf{y}_s \end{pmatrix} = \left[\begin{pmatrix} 0 & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} f_x \\ \mathbf{r}_x \end{pmatrix}\right]^T \begin{pmatrix} 0 \\ \mathbf{y}_s \end{pmatrix}, \tag{A.31}$$

by subsituting appropriately from equation A.29.

The inverse partition matrix formula is now derived to solve equation A.31.

As $\mathbf{R}$ is symmetric, then

$$\begin{pmatrix} 0 & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{pmatrix}^{-1} = \begin{pmatrix} \alpha & \gamma^T \\ \gamma & \mathbf{\Gamma} \end{pmatrix}, \tag{A.32}$$

for some $\alpha \in \mathbb{R}$, $\gamma \in \mathbb{R}^n$ and symmetric $\mathbf{\Gamma} \in \mathbb{R}^{n\times n}$, where $\mathbf{R} \in \mathbb{R}^{n\times n}$.

The relation

$$\begin{pmatrix} 0 & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \alpha & \gamma^T \\ \gamma & \mathbf{\Gamma} \end{pmatrix} = I_d \tag{A.33}$$

yields

$$\gamma^T \mathbf{F} = 1, \tag{A.34}$$

$$\alpha \mathbf{F}^T + \gamma^T \mathbf{R} = 0, \tag{A.35}$$

$$\gamma \mathbf{F}^T + \mathbf{\Gamma} \mathbf{R} = I_d. \tag{A.36}$$

Equation A.35 implies

$$\gamma = -\alpha \mathbf{R}^{-1} \mathbf{F}, \tag{A.37}$$

and substituting this into equation A.34 one obtains

$$\alpha = -\frac{1}{\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F}}. \tag{A.38}$$

Substituting A.38 into A.37, one finds

$$\gamma = \frac{\mathbf{R}^{-1} \mathbf{F}}{\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F}}. \tag{A.39}$$

Finally, it follows from A.36 that

$$\mathbf{\Gamma} = (I_d - \gamma \mathbf{F}^T)\mathbf{R}^{-1} = \mathbf{R}^{-1} - \frac{\mathbf{R}^{-1}\mathbf{F}\mathbf{F}^T\mathbf{R}^{-1}}{\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F}}. \tag{A.40}$$

Therefore,

$$\begin{pmatrix} 0 & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{pmatrix}^{-1} = \left(\begin{array}{c|c} -(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1} & (\mathbf{R}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1})^T \\ \hline \mathbf{R}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1} & \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{R}^{-1} \end{array} \right) \tag{A.41}$$

and so,

$$\begin{aligned} \hat{y}(\mathbf{x}) &= \left(f_x^T, \mathbf{r}_x^T\right) \begin{pmatrix} 0 & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \mathbf{y}_s \end{pmatrix} \\ &= f_x^T \frac{\mathbf{F}^T\mathbf{R}^{-1}\mathbf{y}_s}{\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F}} + \mathbf{r}_x^T\mathbf{R}^{-1}\left(\mathbf{y}_s - \mathbf{F}\frac{\mathbf{F}^T\mathbf{R}^{-1}\mathbf{y}_s}{\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F}}\right) \\ &= f_x\hat{\mu} + \mathbf{r}_x^T\mathbf{R}^{-1}\left(\mathbf{y}_s - \mathbf{F}\hat{\mu}\right). \end{aligned} \tag{A.42}$$

In Kriging, $\mathbf{F} = \mathbf{1}$ and $f_x = 1$ and it can be seen that by substitution into A.42 the Kriging predictor 3.16 is obtained.

Upon substituting $\mathbf{F} = \mathbf{1}$ and $f_x = 1$ into equation A.28 and substituting into equation A.23 given that $\mathbf{A}^T\mathbf{R}\mathbf{A} = 1$,

$$MSE = \hat{\sigma}^2[1 + \mathbf{c}^T\mathbf{R}\mathbf{c} - 2\mathbf{c}^T\mathbf{r}_x] = \hat{\sigma}^2[1 + \mathbf{c}^T(\mathbf{r}_x + \lambda^T) - 2\mathbf{c}^T\mathbf{r}_x]$$

$$= \hat{\sigma}^2[1 - \mathbf{c}^T\mathbf{r}_x] + \lambda.$$

Now

$$-(1 \quad \mathbf{r}_x^T)\begin{pmatrix} -\frac{\lambda}{\sigma^2} \\ \mathbf{c}(\mathbf{x}) \end{pmatrix} = \frac{\lambda}{\sigma^2} - \mathbf{r}_x^T\mathbf{c}$$

$$\Rightarrow \hat{\sigma}^2[1 + \mathbf{c}^T\mathbf{R}\mathbf{c} - 2\mathbf{c}^T\mathbf{r}_x] = \hat{\sigma}^2\left[1 - (1 \quad \mathbf{r}_x^T)\begin{pmatrix} \frac{\lambda}{\sigma^2} \\ \mathbf{c} \end{pmatrix}\right],$$

hence the mean square error A.23 becomes

$$\mathrm{MSE} = \hat{\sigma}^2\left[1 - \begin{pmatrix}1 & \mathbf{r}_x^T\end{pmatrix}\begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1}\begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix}\right] \tag{A.43}$$

Using equation A.41 with $\mathbf{F} = \mathbf{1}$,

$$\begin{aligned}
&\begin{pmatrix}1 & \mathbf{r}_x^T\end{pmatrix}\begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1}\begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix} \\
= \quad & (-1(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1} + \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}, \\
& \quad 1(\mathbf{R}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1})^T + \mathbf{r}^T\mathbf{R}^{-1} - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\mathbf{R}^{-1})\begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix} \\
= \quad & -1(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}1 + \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}1 \\
& \quad +1\mathbf{R}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}\mathbf{r} + \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{r} \\
= \quad & \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} - \frac{11}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} + \frac{\mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}1}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} + \frac{1(\mathbf{R}^{-1}\mathbf{1})^T\mathbf{r}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} - \frac{\mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{r}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}. \qquad \text{(A.44)}
\end{aligned}$$

Thus

$$\hat{\sigma}^2\left[1 - \begin{pmatrix}1 & \mathbf{r}_x^T\end{pmatrix}\begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1}\begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix}\right]$$

$$= 1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{11 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}1 - 1\mathbf{r}^T\mathbf{R}^{-1}\mathbf{1} + \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{r}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}$$

$$= 1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1} - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1} + \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{r})}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}$$

$$= 1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}. \tag{A.45}$$

The MSE is

$$\hat{\sigma}^2 \left[1 - \begin{pmatrix} 1 & \mathbf{r}_x^T \end{pmatrix} \begin{pmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix} \right] = \hat{\sigma}^2 \left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{1})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} \right]. \tag{A.46}$$

For further details see Cressie (1993).

A.3 Expected Improvement

A more sophisticated and most robust form of response surface, in terms of the convergence to a global optimum, is that of the expected improvement. This involves the computation of how much improvement one can expect to achieve when sampled at an untried point. Let the random variable $Y(\mathbf{x}) = (Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n))^T \sim N\left(\hat{y}(\mathbf{x}), s^2(\mathbf{x})\right)$ where $\hat{y}$ is the kriging predictor as defined in equation A.19 and s^2 is the mean square error without the regression term as defined in equation A.46.

Let us assume that one wishes to find an optimum minimum solution to the objective function and the current best objective function value is f_{min}. An improvement of I will be achieved if $I = f_{min} - Y(\mathbf{x}) > 0$. The probability density function of achieving this improvement is given by

$$\text{pdf}(Y(\mathbf{x}), \hat{y}(\mathbf{x})) = \frac{1}{\sqrt{2\pi}s(\mathbf{x})} \exp\left[-\frac{(f_{min} - I - \hat{y}(\mathbf{x}))^2}{2s(\mathbf{x})^2} \right]. \tag{A.47}$$

The expectation of I can be defined as

$$E(I(Y)) = \int_{I=0}^{I=\infty} I(Y) \ \text{pdf}(Y, \hat{y}) \ \mathrm{d}I \tag{A.48}$$

So,

$$E(I) = \int_{I=0}^{I=\infty} I \left\{ \frac{1}{\sqrt{2\pi} s(\mathbf{x})} \exp\left[- \frac{(f_{min} - I - \hat{y}(\mathbf{x}))^2}{2s(\mathbf{x})^2} \right] \right\} \mathrm{d}I. \tag{A.49}$$

Let us make a change of variables

$$Z = \frac{Y - \hat{y}}{s} \quad , \quad u = \frac{f_{min} - \hat{y}}{s} \tag{A.50}$$

which implies that $s \ \mathrm{d}Z = \mathrm{d}Y$.

It is known that $Y = f_{min} - I$ so

$$I = f_{min} - Y = su + \hat{y} - Y = su - sZ = s(u - Z). \tag{A.51}$$

As there would be a negative improvement value if $Z > u$,

$$I = \begin{cases} s(u - Z) & , \quad Z < u \\ 0 & , \quad \text{otherwise} \end{cases} \tag{A.52}$$

Now,

$$E(I) = \int_{-\infty}^{\infty} I(Y)\text{pdf}(Y) \ \mathrm{d}Y = \int_{-\infty}^{u} I(Z)\text{pdf}(Z) \ s \ \mathrm{d}Z$$

$$\Rightarrow \quad E(I) = \int_{-\infty}^{u} s(u - Z). \frac{1}{s\sqrt{2\pi}} \ e^{-Z^2/2} s \ \mathrm{d}Z$$

$$\Rightarrow \quad E(I) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} u \ e^{-Z^2/2} s \ \mathrm{d}Z - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} Z \ e^{-Z^2/2} s \ \mathrm{d}Z$$

$$\Rightarrow \quad E(I) = \frac{s\,u}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{-Z^2/2}\,\mathrm{d}Z + \left[\frac{s}{\sqrt{2\pi}}\,e^{-Z^2/2}\right]_{-\infty}^{u}$$

$$\Rightarrow \quad E(I) = s\left(u.\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{-Z^2/2}\,\mathrm{d}Z + \frac{1}{\sqrt{2\pi}}\,e^{-u^2/2}\right) \tag{A.53}$$

$$\Rightarrow \quad E(I) = s\left(u\,\mathrm{cdf}(u) + \mathrm{pdf}(u)\right), \tag{A.54}$$

where cdf(u) is the standard normal cumulative distribution function and pdf(u) is the standard normal probability density function.

When finding an optimum maximum solution to the problem with the current best objective function value as f_{max}, an improvement I will be achieved if $Y = I - f_{max}$. Equation A.54 is achieved but in this case with $u = (\hat{y} - f_{max})/s$.

Appendix B

3D Airbox Analysis Setup

B.1 Mesh Generation

B.1.1 Internal Airbox

For the design optimization process to run entirely automatically, the CAD geometry must be imported into a meshing tool using an executable script, in these studies GAMBIT$^{\text{TM}}$(Fluent$^{\text{TM}}$, 2003a) is used. The majority of meshing tools available commercially run in batch mode via reading a journal file. The set up of this journal file is important as, more often than not, one is faced with compatibility issues between the output file from the CAD engine and the meshing software.

The geometry created by CATIA for the internal walls of the 3D airbox can be seen in Figure B.1. The front section of the trumpet tray has been hidden to illustrate the location of the cylinders.

B.1.2 External Flow around Airbox

To date, the representation of the airbox has been based solely on its internal flow given a fixed uniform mass flow rate at the inlet. However, the flow simulation must be realistic enough to draw conclusions about the geometry manipulation. To do this an airbox positioned at the centre of a large box is tested and the flow external to the airbox is simulated as well as the internal flow. This requires a thick-surfaced geometry

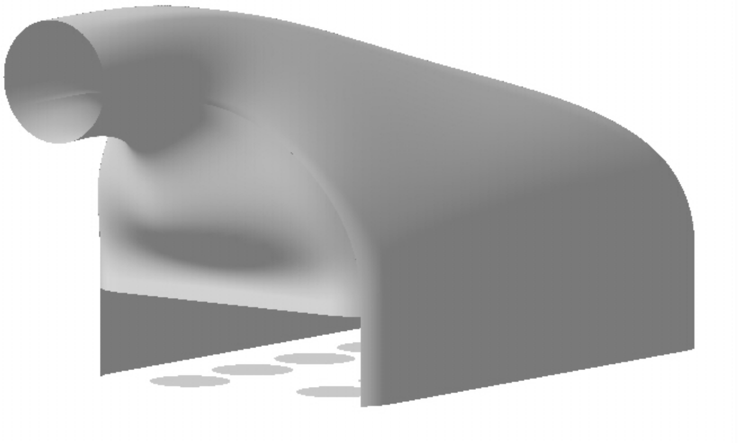


Figure B.1: CATIA baseline geometry of walls of internal airbox without the front trumpet tray section to illustrate the position of the cylinders

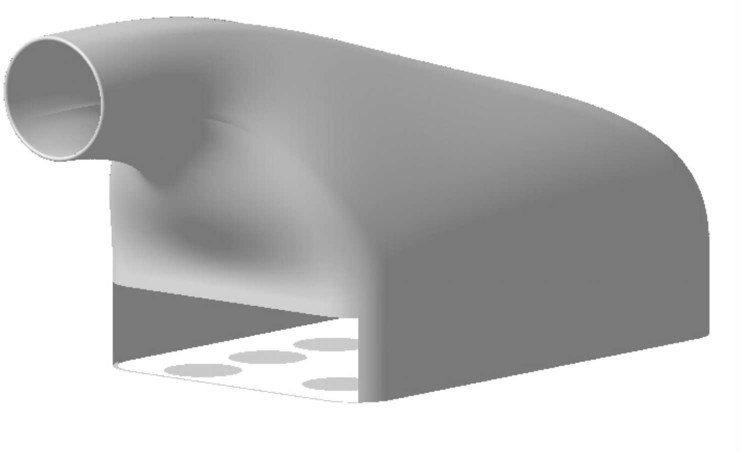


Figure B.2: CATIA geometry of thick-surfaced airbox with the front trumpet tray section hidden to illustrate the location of the cylinders

to allow the boundary layer development entering the airbox around the inlet lip to be captured, rather than a prescribed boundary condition of a uniform mass flow rate. The thick-surfaced baseline geometry can be seen in Figure B.2. Only one simulation to capture the developed inlet profile is required and the mesh can be constructed manually if necessary.

An illustration of the mesh generated can be seen in Figure B.3. This includes a close up view of how the mesh is treated at the inlet lip faces to ensure an accurate representation of the stagnation points, which occur at the inlet, along with the boundary layer development into the internal part of the airbox.

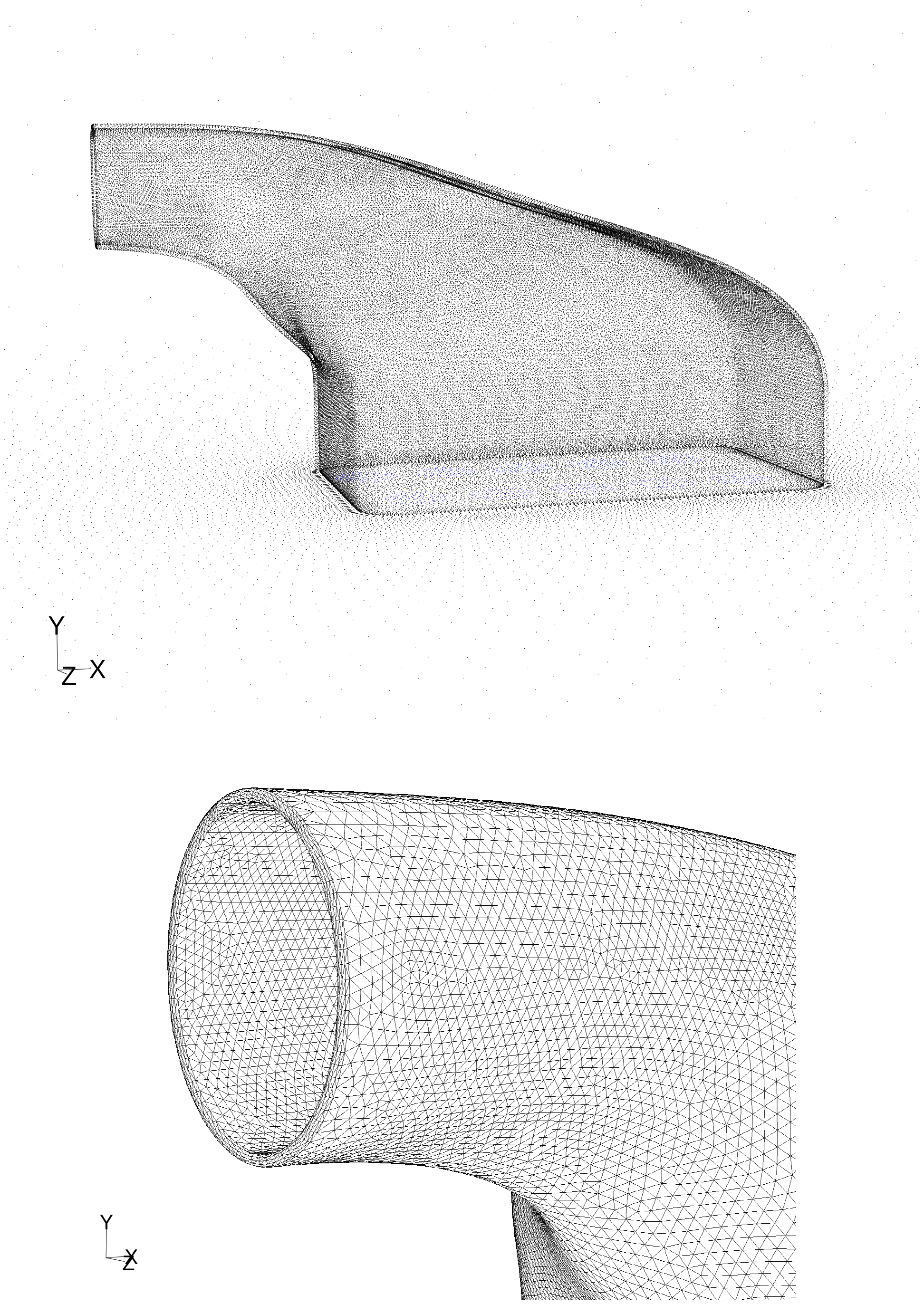


Figure B.3: Unstructured mesh of a thick surfaced 3D airbox

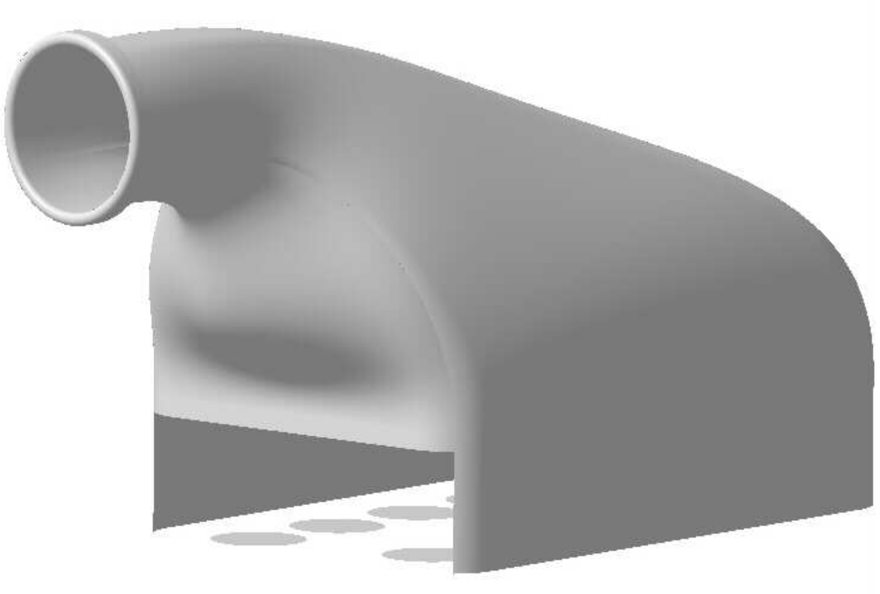


FIGURE B.4: CATIA geometry of thick-lipped airbox with the front trumpet tray section hidden to illustrate the location of the cylinders

B.1.3 Reduced External Flow around Airbox

Ideally, all simulations performed should include the external box for a realistic solution. However, this is considered far too expensive for a design study with a few hundred runs. The addition of the external box described in the previous section adds approximately one and a half million cells to the mesh count. An alternative option is to draw a much smaller box around just the inlet of the airbox to simulate the boundary layer development provided by the thick surface around the airbox inlet lip. The geometry produced by CATIA is now designed to have a fixed inlet lip with the inlet of the internal airbox in the same position as for the internal airbox case. This allows the geometry to change freely with the change of design parameter values without affecting the lip. Such a geometry can be seen in figure B.4 with a lip thickness of 15mm.

The mesh to simulate a smaller airbox can be constructed automatically. An illustration of the mesh generated is seen in Figure B.5.

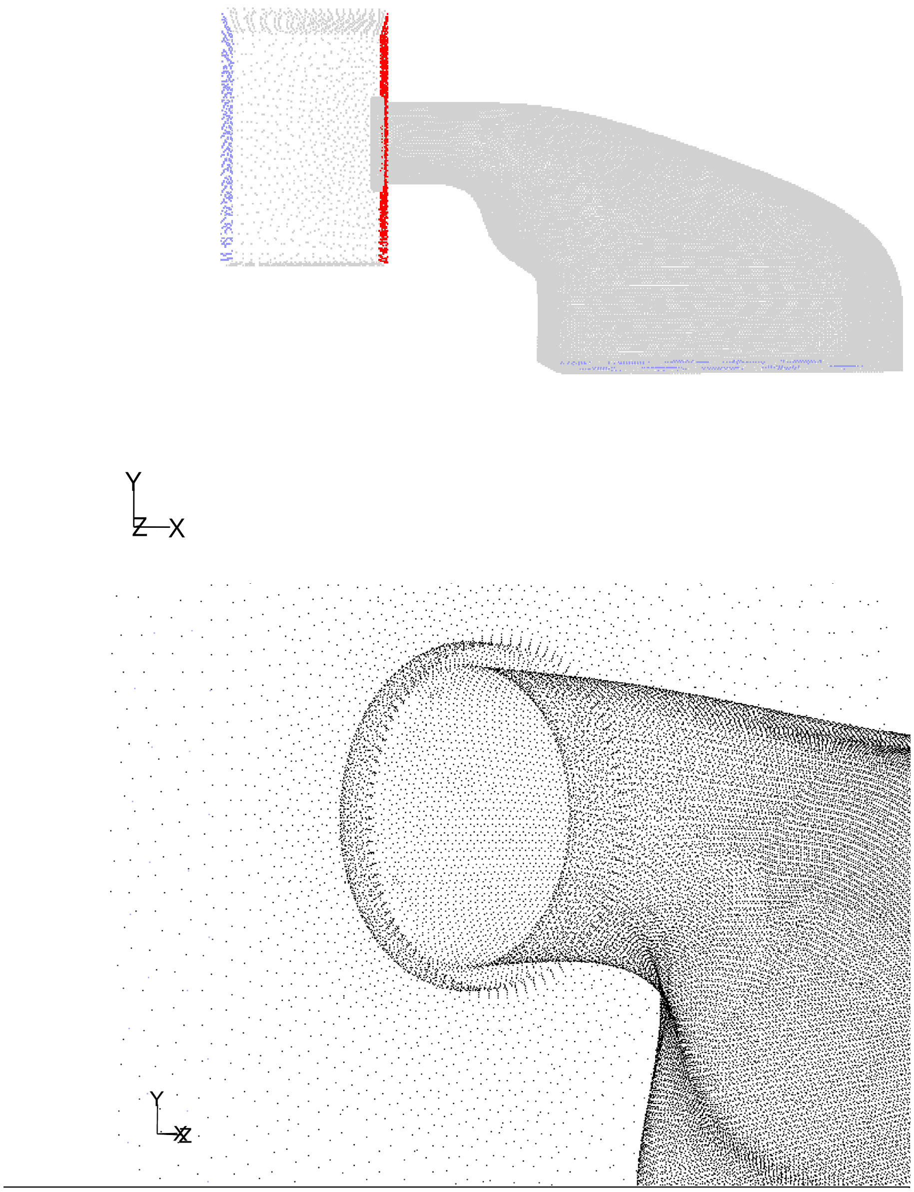


FIGURE B.5: Unstructured mesh of a thick surfaced 3D airbox

B.2 Flow Simulation

To run a flow simulation automatically, a journal file is read into Fluent in a similar way to that of the meshing journal described above.

B.2.1 Internal Airbox Simulation with Uniform Inlet Profile

An annotated and abridged journal file for the internal airbox flow simulation with a fixed uniform mass flow rate at the inlet is given below. In all cases, the mesh file is first loaded into the serial solver Fluent. Here, the grid domain is reordered and the domain partitioned appropriately given the number of parallel processors to be used. This is then written out as a case (.cas) file.

The case file is then read in by the parallel Fluent solver and the boundary conditions set as follows:

```
file read-case "filename.cas"
define models viscous ke-standard y
define boundary-conditions mass-flow-inlet , y 0.5894 n 1 1 n 0 y y y n 1 n 0 n 0 n n n y
5 0.41
define boundary-conditions pressure-outlet , n 0 n n y n n y 5 5 n
define boundary-conditions fluid fluid.3 n n n y 0 0 0 0 0 1 n n y n 1 0 0 0 1 0 n 75000000
n 75000000 n 75000000 n 1000 n 1000 n 1000 0 0 n 1
```

This means that the inlet mass flow rate is set at 0.5894kgs^{-1} with the turbulence specification method as intensity of 5% and a hydraulic diameter of 0.41m. The pressure outlet condition also specifies the turbulence specification method with an intensity of 5% and a hydraulic diameter of 5m. The boundary conditions set for the porous zone acting as the engine filter, fluid.3, has a viscous resistance of 75000000m^{-2} in all directions and an inertial resistance of 1000m^{-1} in all directions.

Under-relaxation factors are set and the solution is iterated with first order upwind:

```
solve set under-relaxation pressure 0.6 mom 0.7
solve initialize initialize-flow
solve iterate 1500
```

The turbulence model is now changed to the kϵ-realizable with non-equilibrium wall functions enabled. The pressure velocity coupling is changed from SIMPLE to SIMPLEC, the under-relaxation factors raised and the upwind scheme for momentum changed to second order:

```
define models viscous ke-realizable y near-wall-treatment
non-equilibrium-wall-fn y
solve set p-v-coupling 21
solve set under-relaxation pressure=0.8 mom=0.8
solve set discretization-scheme mom 1
solve iterate 2000
```

Finally the upwind schemes for the turbulent kinetic energy and dissipation rate are both changed to second order:

```
solve set discretization-scheme k 1 epsilon 1
solve iterate 1500
```

The case and data files are then written out in compressed form:

```
file write-case-data "filename.cas.gz"
```

B.2.2 Simulation Including Complete External Domain

The journal file for the flow simulation including the external flow around the airbox is similar and is as follows.

Read in appropriately reordered and partitioned .cas file. Initially define the turbulence model to be the standard kϵ model. Define the velocity boundary conditions as follows:

```
define boundary-conditions velocity-inlet velocity_inlet.1 n n y y n 50 y n n y 3 10
define boundary-conditions velocity-inlet velocity_inlet.2 n n y y n -9 y n n y 5 5
```

This means that the front face of the external box has a velocity set to 50ms^{-1} in the X direction (perpendicular to the face) with a turbulence specification method of intensity at 3% and hydraulic diameter of 10m. The velocity condition set at the cylinder faces is -9ms^{-1} and so the air is being sucked out of the airbox at a velocity required by the engine at 18000rpm. At the cylinders the turbulence specification method is of intensity and hydraulic diameter of 5% and 5m respectively.

The boundary conditions for the velocity inlet for the external box, the pressure outlet, and the porous zone, fluid.5, are the same as for the internal flow simulation run file above. The initial under-relaxation factors and iteration number are set as the following:

```
solve set under-relaxation pressure 0.6 mom 0.7 k 0.6 epsilon 0.8
solve initialize initialize-flow
solve iterate 1500
```

The turbulence model is then changed to the realizable kϵ, the pressure-velocity coupling set to SIMPLEC and the upwind scheme for momentum is changed to second order as in the run file for the internal simulation. After 2000 iterations, the upwind schemes for the turbulence kinetic energy and dissipation rate are both set to second order and 1500 iterations are performed. This concludes the external flow simulation and the case and data files are written out. The velocity profile at the airbox inlet is captured and written to a file.

Note: The reduced external domain simulation uses the same instructions as above with the exportation of the inlet velocity profile.

B.2.3 Simulation with Prescribed Inlet Velocity Profile

The velocity profile captured in the previous section is imported into the journal file for the internal flow and is used instead of the uniform mass flow rate at the entry. The journal file for the simulation remains the same as that for section B.2.1 except the boundary condition for the entry is replaced with:

```
file read-profile "InletProfile.prof"
define boundary-conditions velocity-inlet , n y y y y n "inlet_face" "x-velocity" y n "inlet_face"
"y-velocity" y n "inlet_face" "z-velocity" n n n y 5 0.41
```

B.2.4 Comparison of all Simulations

The results of the flow simulations on the centerplane for the baseline geometries of that with a uniform inlet profile can be seen in Figure B.6 and for the baseline geometry with the imported velocity inlet profile can be seen in Figure B.7. The inlet profile is exported from the single simulation including a full external domain, the centerplane of

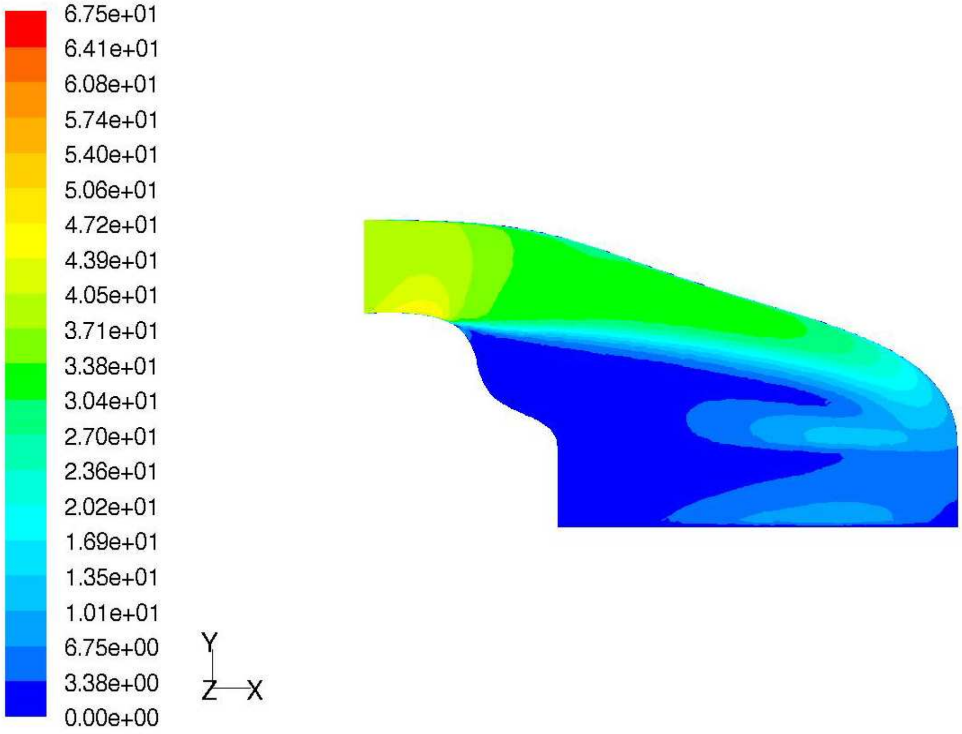


Figure B.6: Filled contours of velocity magnitude on the centreplane of the 3D airbox of a uniform mass flow rate inlet profile

velocity contours can be seen in Figure B.8. The centerplane of velocity contours shown for the reduced external domain can be seen in Figure B.9.

B.3 Mesh Dependency Study

To determine the most appropriate mesh size to use for the design studies in Chapters 6 and 7, a number of different mesh sizes, from 100,000 cells to 5,500,000 cells for the internal flow domain, were analysed using Fluent and the C_p values calculated. The results can be seen in Figure B.10.

It is clear that there is a downward trend from the coarsest mesh to the finest mesh but there is only a percentage difference of less than 2% between the 500,000 cell mesh and the 5,500,000 cell mesh rendering this comparison inconclusive.

Instead, the velocity profiles taken at various stages along the flow path through the airbox are examined. These stages are shown as lines A, B and C in Figure B.11.

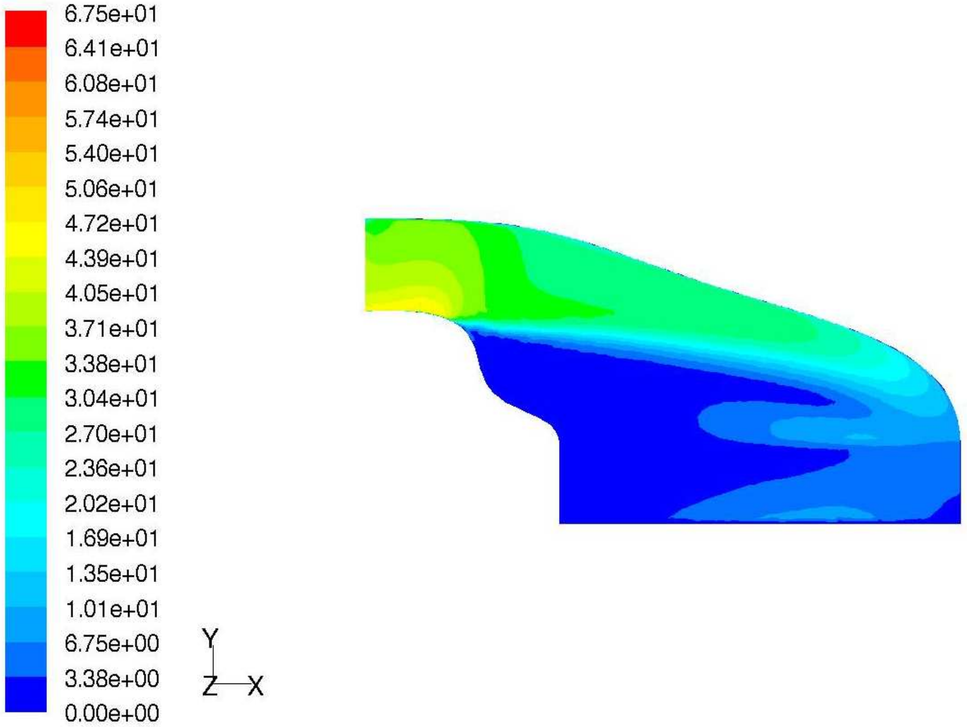


FIGURE B.7: Filled contours of velocity magnitude on the centreplane of the 3D airbox of a prescribed velocity inlet profile

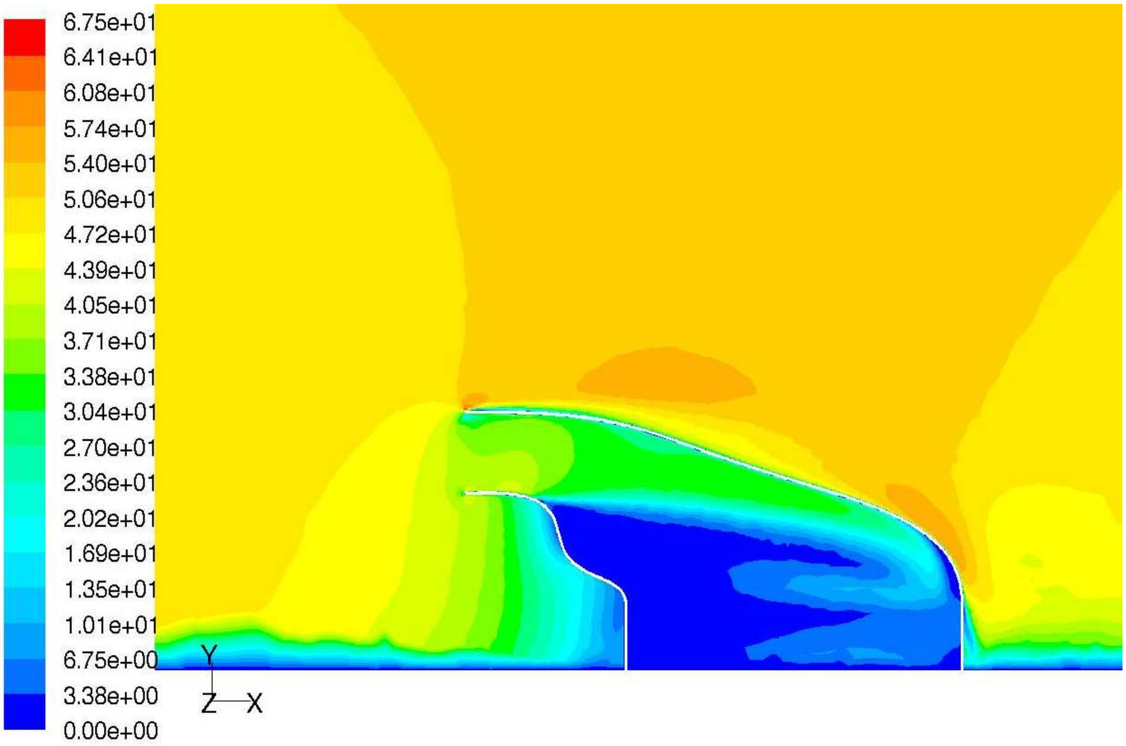


FIGURE B.8: Filled contours of velocity magnitude on the centreplane of the 3D airbox with full external domain

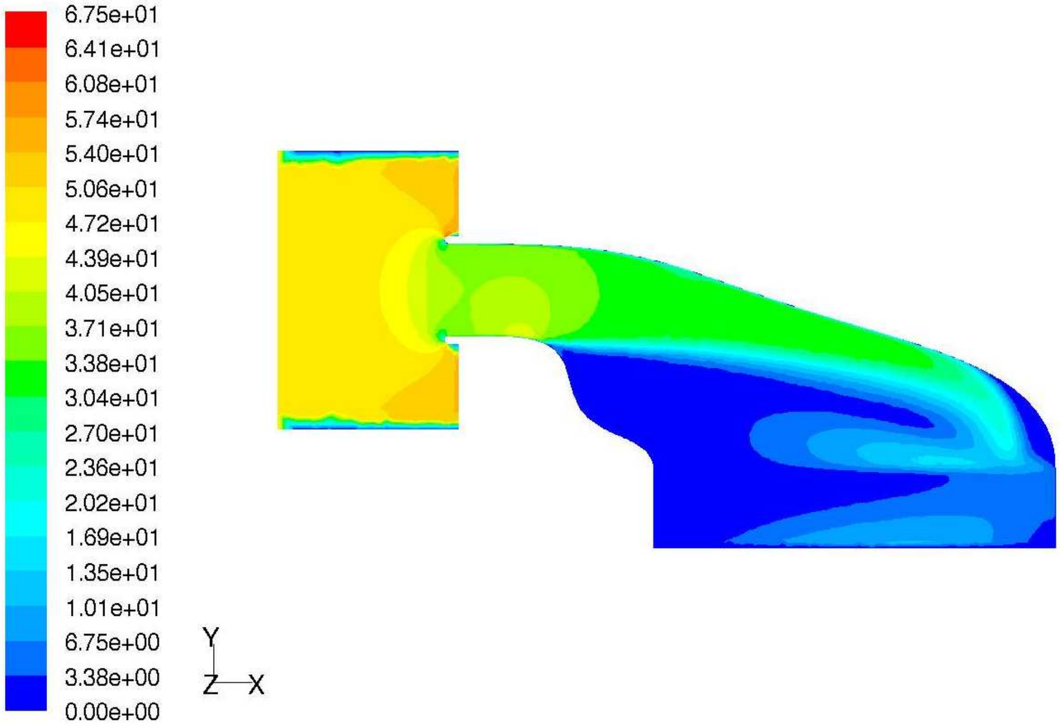


Figure B.9: Filled contours of velocity magnitude on the centreplane of the 3D airbox with a small external box around inlet lip

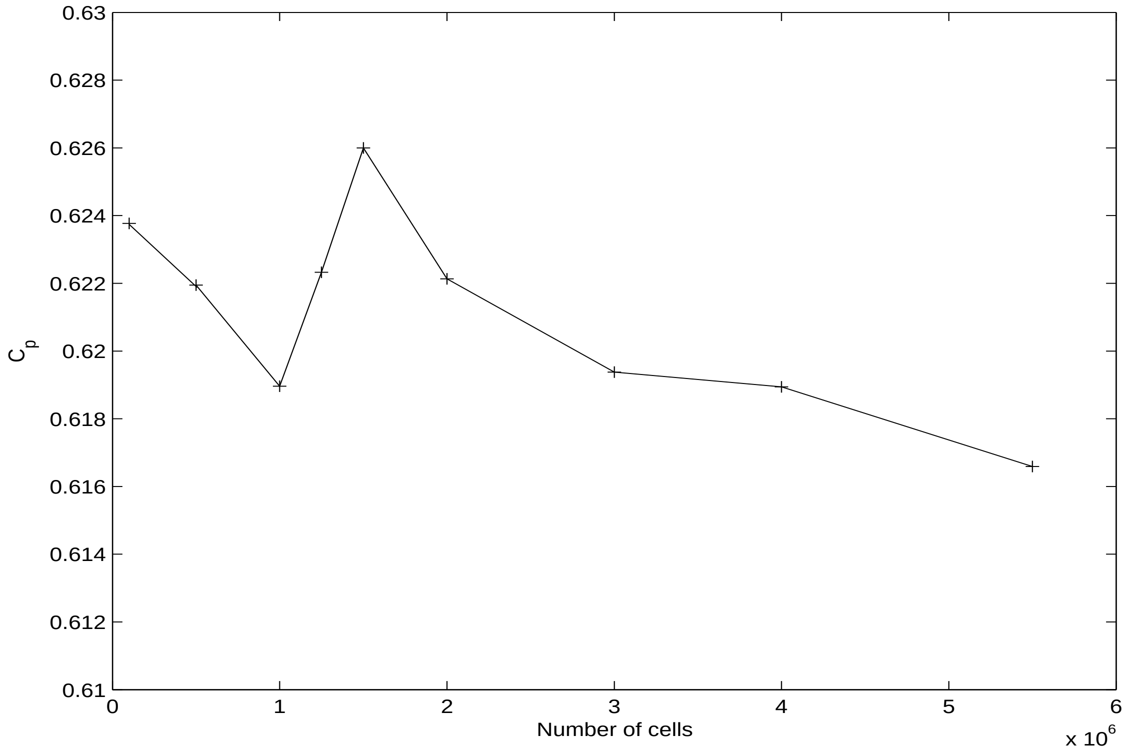


Figure B.10: Graph illustrating the dependency of the C_p value with the mesh density

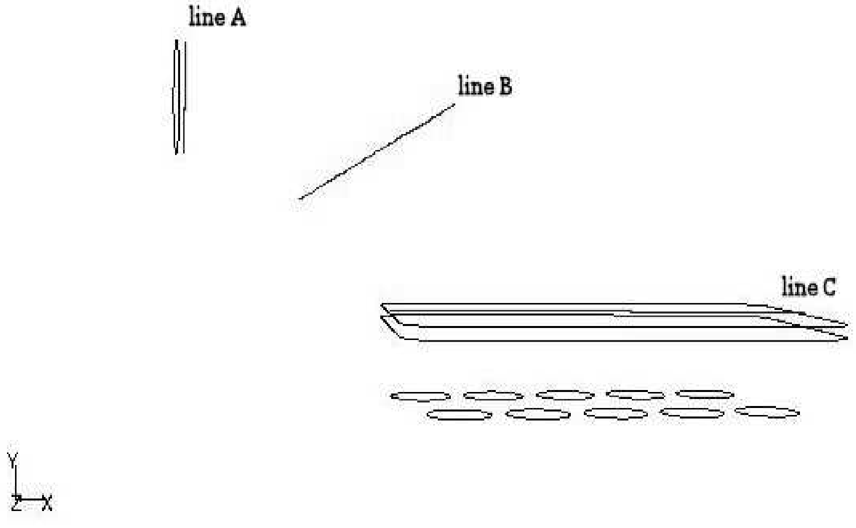


Figure B.11: Position of lines at which velocity profiles are taken

The X velocity profiles at line A are shown in Figure B.12, the X and Y components of the velocity profile at line B are illustrated in Figure B.13 and the Y velocity profile at line C is shown in Figure B.14. The profiles at all mesh sizes are shown along with the profiles taken from the simulation run with the imported velocity inlet profile containing a developed boundary layer.

It can be seen from these profile comparisons that the coarsest mesh which has the closest match to the profile of the finest mesh would be that of the 1,250,000 cell mesh, although the 500,000 cell mesh would be considered adequate. Velocity contour illustrations of the 500,000 cell case are shown in Figure B.6.

The profiles shown of the simulation with the developed boundary layer condition at the inlet is not significantly different with it returning a pressure recovery value of $C_p = 0.65$. This is only a 4.5% percentage difference to the C_p value of the 1,250,000 cell simulation. Hence, although the fixed uniform mass flow inlet boundary condition using a 1,250,000 cell mesh would be ideal, it is considered that a 500,000 cell mesh is a realistic enough model to use for the purpose of geometric shape control within the design process studied in Chapter 6.

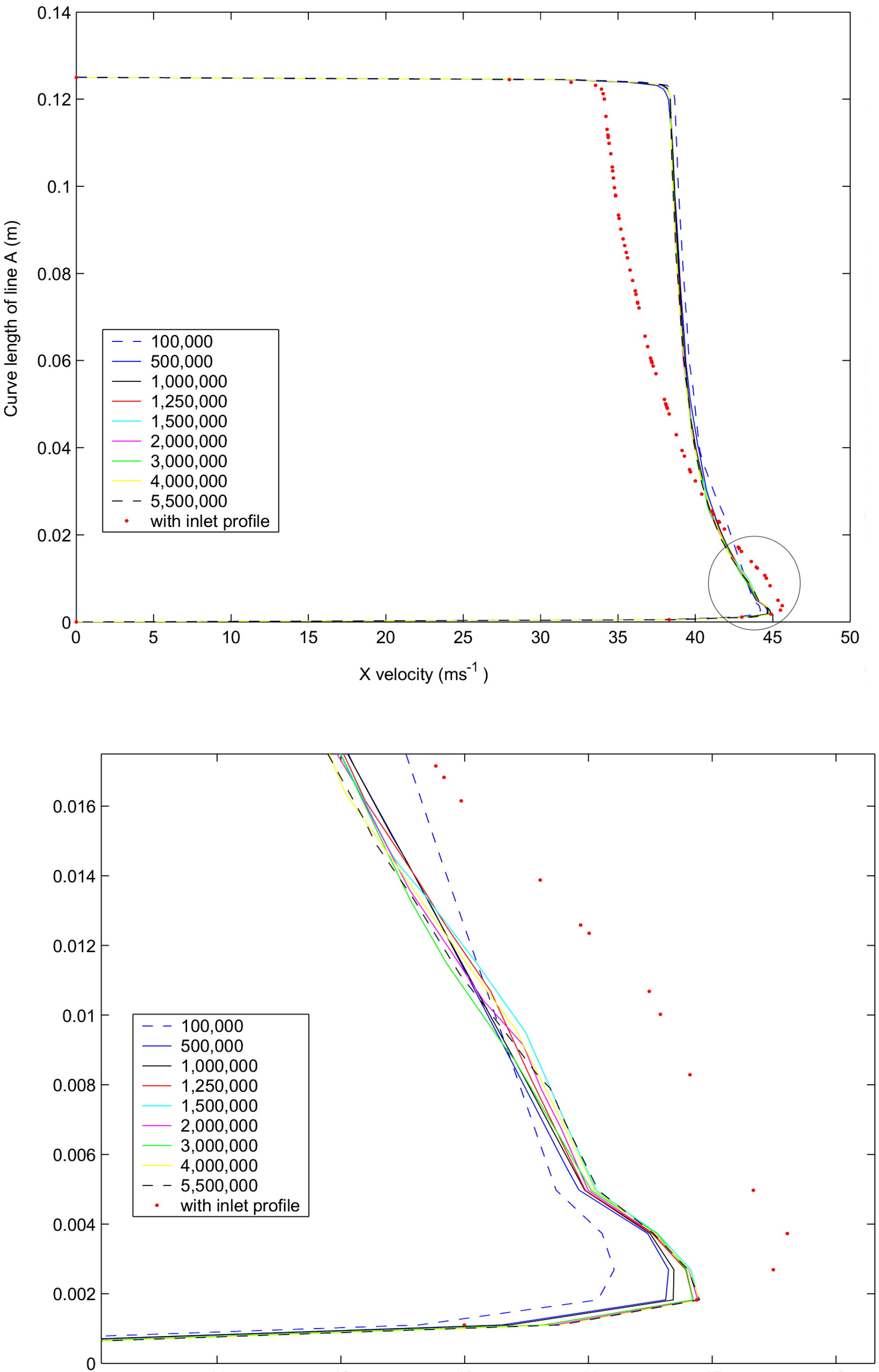


FIGURE B.12: Velocity Profile at Line A. The lower figure is a magnification of the area within the circle shown in the upper figure.

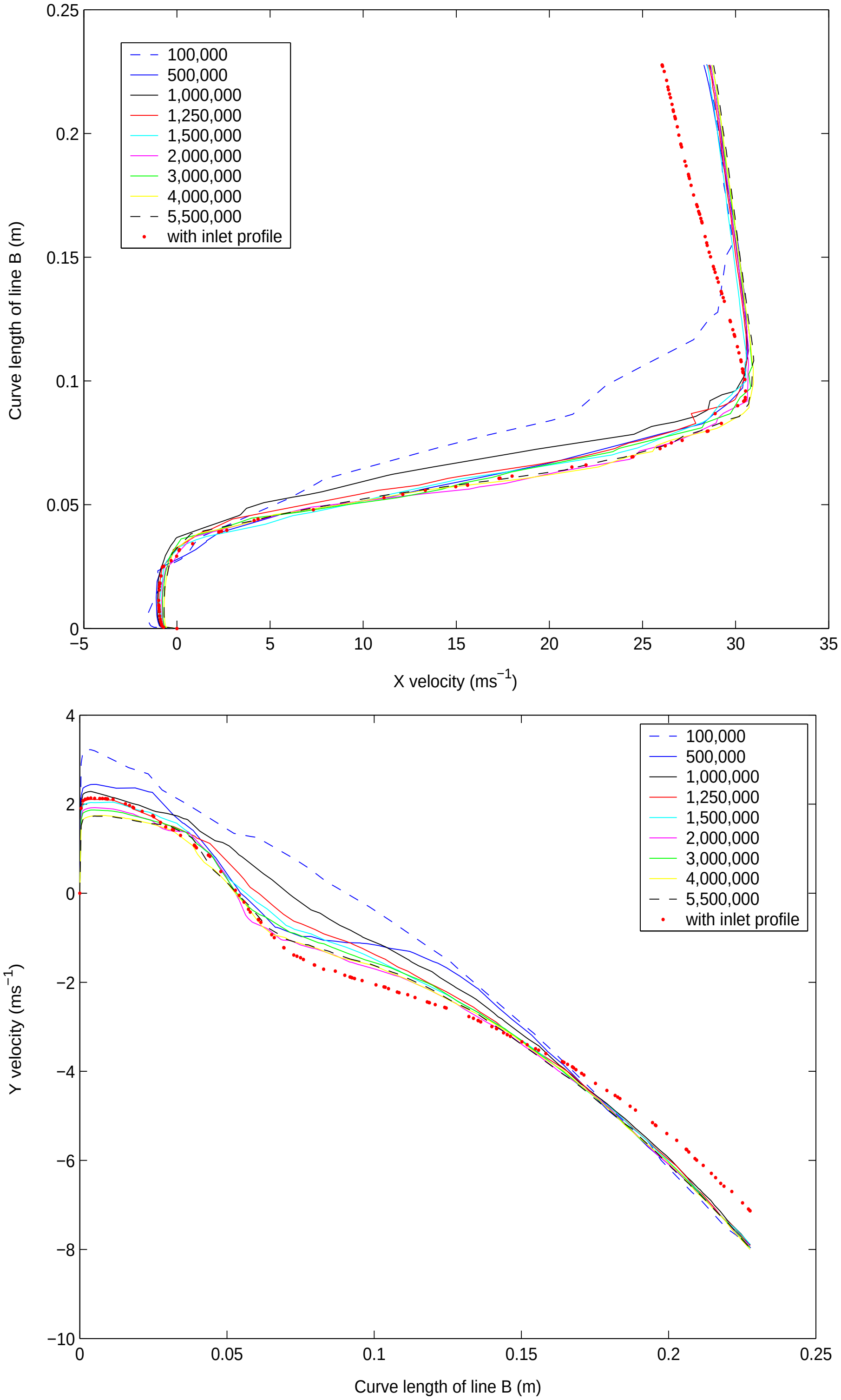


FIGURE B.13: Velocity Profile at Line B. The upper figure shows the X velocity component profile and the lower figure shows the Y velocity component profile.

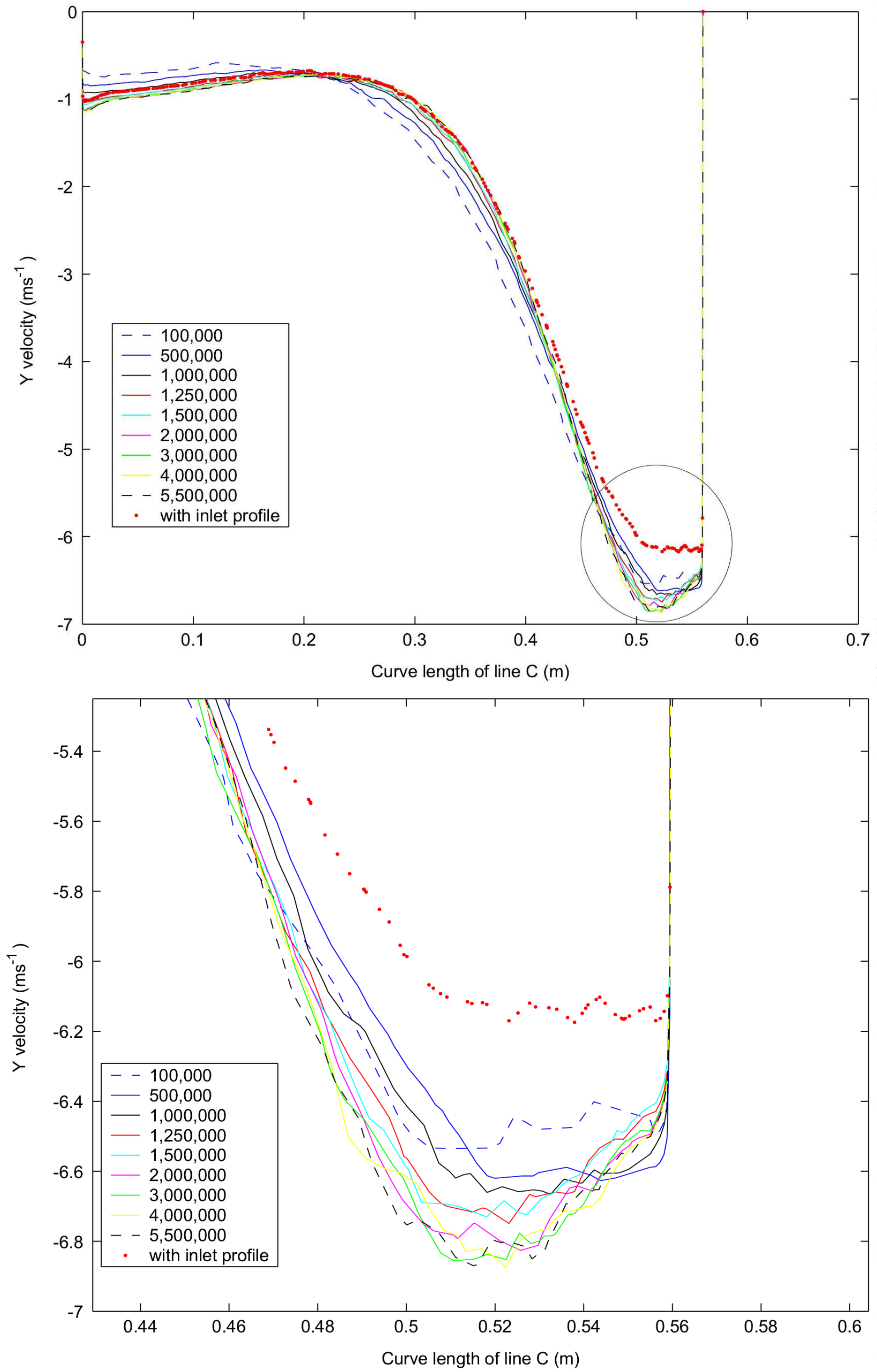


FIGURE B.14: Velocity Profile at Line C. The lower figure is a magnification of the area within the circle shown in the upper figure.

Appendix C

Carotid Artery Analysis Setup

C.1 Automated Artery Point Cloud Analysis

The following section describes the automated point cloud analysis to extract the key geometrical features described by design variables in the CATIA design table. A synopsis of the code used is given at appropriate points.

The geometry is rotated to align the blood flow with the z-axis. The geometry is translated such that the origin of the xy-plane lies approximately near the root of the bifurcation on the CCA.

```
load PointCloud_x_coords.dat
load PointCloud_y_coords.dat
load PointCloud_z_coords.dat
Xlength=max(PointCloud_x_coords)-min(PointCloud_x_coords);
Ylength=max(PointCloud_y_coords)-min(PointCloud_y_coords);
Zlength=max(PointCloud_z_coords)-min(PointCloud_z_coords);
if Xlength==max(Xlength, Ylength, Zlength);
    PointCloud_x_coords=points_z_coords;
   PointCloud_y_coords=points_y_coords;
   PointCloud_z_coords=points_x_coords;
else if Ylength==max(Xlength, Ylength, Zlength);
   PointCloud_x_coords=points_x_coords;
   PointCloud_y_coords=points_z_coords;
   PointCloud_z_coords=points_y_coords;
else
```

```
   PointCloud_x_coords=points_x_coords;
   PointCloud_y_coords=points_y_coords;
   PointCloud_z_coords=points_z_coords;
end
```

The point cloud analysis is performed to enable the extraction of key geometrical features of the patient's artery with which a reasonably accurate CAD model is constructed. The key issue is finding the dividing point of the artery at the bifurcation from a spray of points, typically totalling approximately 7000 points. This is not trivial due to the possibility of a patient artery featuring extreme tortuosity and non-planarity. By slicing the artery finely, with an appropriate resolution, in the x-direction, one finds the position of the maximum z value of each slice, checking for the possibility of the ICA or ECA crossing above the divider point further downstream within the same slice of x values. The minimum value of the array of maximum z values found is taken to be the divider.

```
MinX=min(points_x_coords);
MaxX=max(points_x_coords);
Resolution=30;
Dist=(MaxX-MinX)/Resolution;
for k=1:Resolution
    j=1;
    for i=1:size(points_x_coords)
       if points_x_coords(i) < MaxX-(Dist*(k-1)) & points_x_coords(i) > Max_X-(Dist*k)
       x_slice(j)=points_x_coords(i);
       y_slice(j)=points_y_coords(i);
       z_slice(j)=points_z_coords(i);
       j=j+1;
       end
    end
save ([‘x_slice’ num2str(k) ‘.dat’], ‘x_slice’,’-ascii’)
save ([‘y_slice’ num2str(k) ‘.dat’], ‘y_slice’,’-ascii’)
save ([‘z_slice’ num2str(k) ‘.dat’], ‘z_slice’,’-ascii’)
clear x_slice
clear y_slice
clear z_slice
end for k=1:Resolution
A{k}=load([‘z_slice’ num2str(k) ’.dat’]);
end
for k=1:Resolution
```

```
if max(A{k})>0
    for i=1:size(A{k},2)-1
       if abs(A{k}(i)-A{k}(i+1))<0.005
          B(k)=max(A{k});
       else
          B(k)=A{k}(i);
       end
      else
          B(k)=1;
      end
    end
end
[Divider_Z,position]=min(B);
[Divider_Z,Divider_position]=max(Aposition);
C=load([‘y_slice’ num2str(position) ‘.dat’]);
D=load([‘x_slice’ num2str(position) ‘.dat’]);
Divider_Y=C(Divider_position);
Divider_X=D(Divider_position);
```

From the appropriate z-slice the x coordinate is determined. For the y coordinate, the first few points at the maximum z value must be considered to ensure that the divider point does not lie on the wall of the ICA or ECA. By analysing the y values of these few points, it can be determined whether the dividing point represents a peak or perhaps a flatter divide.

Interpolate appropriate x and y values in the appropriate x-slice given a small cluster of maximum z values, say five maximum values, to find the coordinates of the divider point:

```
Divider_coords=[DIV_X, DIV_Y, DIV_Z];
```

So far, the possibility of artery tortuosity has been ignored. To achieve the correct coordinates of the dividing point taking account of potential tortuosity, the artery is sliced in the x-direction at varying rotations of the point cloud. After a full rotation through 2π the true z value of the divider point is chosen.

```
choose the increments of theta_incr through which the artery is rotated

for theta = 0:theta_incr:2*pi
    R = [cos(theta), -sin(theta); sin(theta), cos(theta)];
NewX=R*points_x_coords;
```

```
NewY=R*points_y_coords;
NewZ=points_z_coords;
Re-slice as above to find the minimum maximum z value from all the x-slices and find the divider
point coordinates.
end
```

The (x,y,z) coordinates of the dividing point at the artery bifurcation has now been found. If the artery has been rotated through angle θ, the (x',y',z') coordinates found must be translated using the inverse rotation matrix to find the true (x,y,z).

```
R_inv = [cos(theta), sin(theta); -sin(theta), cos(theta)];
[DIV_X, DIV_Y, DIV_Z] = R_inv*[DIV_X,DIV_Y,DIV_Z];
```

The point cloud is then be split into intersections at any given resolution below and above the dividing point. The diameter of the artery at each of these z-planes is determined and likewise the artery centroid points. The diameters of the separate ICA and ECA arteries on the same plane are determined from the points lying either side of the dividing point in the xz-plane.

An ellipse is fitted to the point cloud at these intersections and sampled at the required number of points, `N`. The ellipse has a major axis of radius `R1` (the x diameter of the artery at each intersection), a minor axis of radius `R2` (the y diameter of the artery at each intersection), each ellipse is not rotated through any angle and the ellipse is centred at `x0, y0` (the x, y centroid of the artery at each intersection). The coordinates of the points are given by `ellipseX` and `ellipseY` .

```
for each intersection
if length(R1) =length(x0)
    A=length(R1)*length(x0);
else
    A=length(R1);
end;
angle=0;

for k=1:A
    if length(x0)==1
       centreX=x0;
       centreY=y0;
       radiusMaj=R1(k);
       radiusMin=R2(k);
```

```
    if length(angle)==1
       ang=angle;
    else
       ang=angle(k);
    end;
    elseif length(R1)==1
       centreX=x0(k);
       centreY=y0(k);
       radiusMaj=R1;
       radiusMin=R2;
       ang=angle;
    elseif length(x0)==length(R1)
       centreX=x0(k);
       centreY=y0(k);
       radiusMaj=R1(k);
       radiusMin=R2(k);
       ang=angle(k);
    end;
    theta=linspace(0,2*pi,N(rem(k-1,size(N,1))+1,:)+1);
    ellipseX=radiusMaj*cos(theta)*cos(ang)-sin(ang)*radiusMin*sin(theta)+centreX;
    ellipseY=radiusMaj*cos(theta)*sin(ang)+cos(ang)*radiusMin*sin(theta)+centreY;
end
```

The fitted ellipses to the point cloud are seen for the artery discussed are shown in Chapter 7 can be seen in Figure C.1.

The ICA branch angle also must be determined for the Stage 1 CAD model representation. This is achieved by finding the closest point on the plane immediately above the dividing point of the ICA and of the ECA. The angle to the horizontal is determined.

All the components for the design table have now been found and the parametric CAD model can be updated.

To test this process, a number of other arteries were analysed and the CAD table updated. The results of which can be seen in Tables C.1, C.2, C.3, C.4 with their corresponding ellipses fits in Figures C.2, C.4, C.6, C.8 respectively, and their corresponding fits to the arteries they represent in Figures C.3, C.5, C.7 and C.9 respectively.

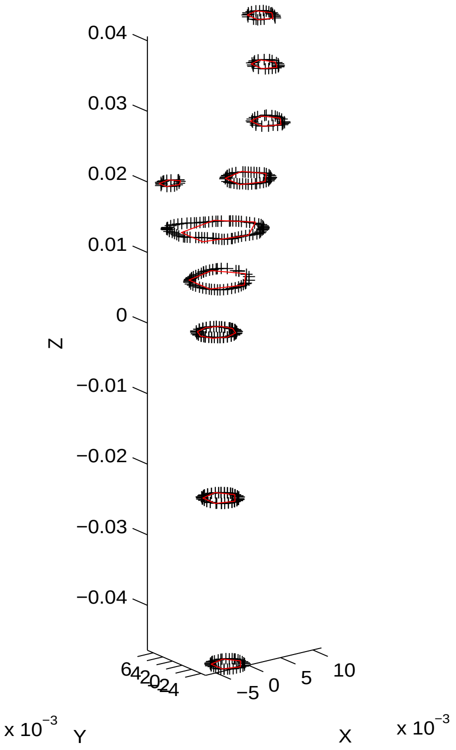


FIGURE C.1: Ellipses fitted to the point cloud at specified z-plane intersections

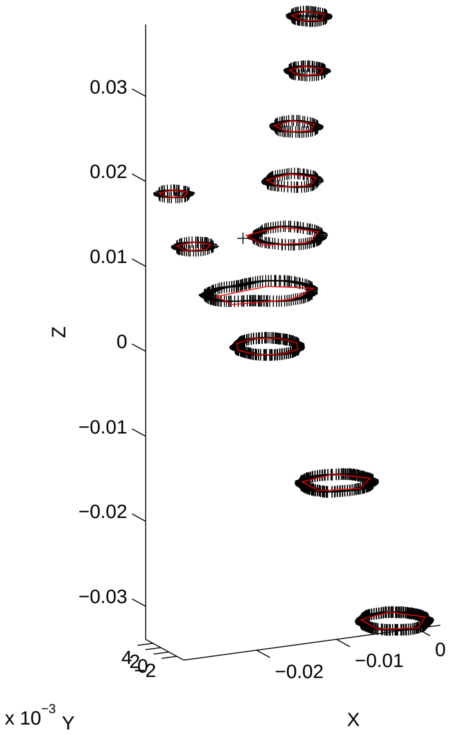


FIGURE C.2: Ellipses fitted to the point cloud of artery 1 at specified z-plane intersections

Parameters	Value
A (m)	0.012818
B (deg)	23.6436
C (m)	0.038453
ECArootX (m)	0.023061
ECArootY (m)	0.00106
ICArootX (m)	0.017483
ICArootY (m)	0.000421
BifurcationRoot (m)	-0.0053472
ECAcentX(m)	0.027300
ECAcentY (m)	0.0022
ICAcentX (m)	0.019227
ICAcentY (m)	0.011244
I (m)	0.0006755
J(x,y) (m)	(0.004733,6.3e-05)
	(0.0020755,0.0036352)
	(-0.0022245,0.0022707)
	(-0.0022245,-0.0021447)
	(0.0020755,-0.0035092)
J(z) (m)	-0.033882
K(x,y) (m)	(0.011749,0.000929)
	(0.0089077,0.0042396)
	(0.0043103,0.0029751)
	(0.0043103,-0.0011171)
	(0.0089077,-0.0023816)
K(z) (m)	-0.016941
L centroid(x,y) (m)	(0.013100,0.001100)
L_r (m)	0.005973/2
M centroid(x,y) (m)	(0.011494,0.000704)
R (m)	0.032044
M_r (m)	0.004021/2
N (m)	0.006800
O(x,y) (m)	(0.019655,0.0015535)
	(0.018589,0.0040874)
	(0.016015,0.005137)
	(0.01344,0.0040874)
	(0.012374,0.0015535)
	(0.01344,-0.00098042)
	(0.016015,-0.00203)
	(0.018589,-0.00098042)
O centroid	(0.016015,0.0015535)
ECA diameter at exit	0.003468

Table C.1: Parameter values for Stage 1 model of artery 1

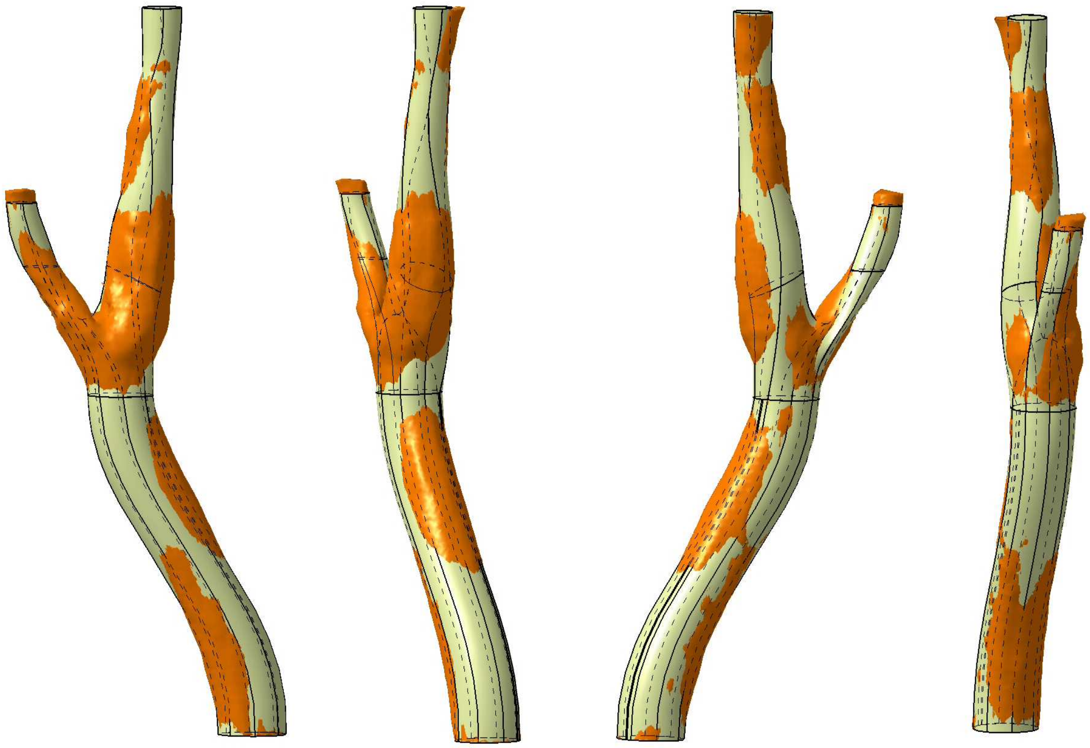


Figure C.3: Difference in geometry between the patient artery (orange) and the parametric CAD geometry (beige) of artery 1

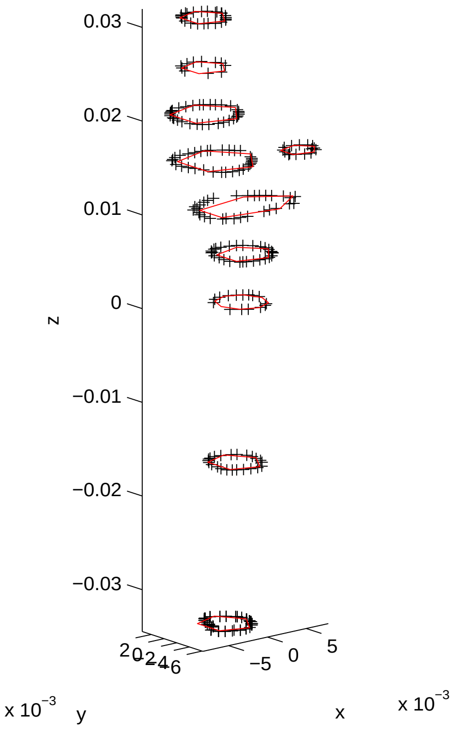


Figure C.4: Ellipses fitted to the point cloud of artery 2 at specified z-plane intersections

Parameters	Value
A (m)	0.015984
B (deg)	23.1581
C (m)	0.031967
ECArootX (m)	0.004547
ECArootY (m)	-0.002995
ICArootX (m)	0.00115
ICArootY (m)	-0.000475
BifurcationRoot (m)	-0.002565
ECAcentX(m)	0.005878
ECAcentY (m)	-0.00394
ICAcentX (m)	0.015984
ICAcentY (m)	-0.005576
I (m)	-0.002932
J(x,y) (m)	(0.002693,0.000478) (0.00082596,0.003256) (-0.002195, 0.0021949) (-0.002195,-0.0012389) (0.00082596,-0.0023)
J(z) (m)	-0.034471
K(x,y) (m)	(0.003346,-0.00025) (0.0015159,0.0025699) (-0.0014452,0.0014928) (-0.0014452,-0.0019928) (0.0015159,-0.0030699)
K(z) (m)	-0.017236
L centroid(x,y) (m)	(-0.002978,-0.0020595)
L_r (m)	0.008256/2
M centroid(x,y) (m)	(-0.0055865,-0.0030725)
R (m)	0.026639
M_r (m)	0.004673/2
N (m)	-0.010053
O(x,y) (m)	(0.003494,-0.001002) (0.0027219,0.00094396) (0.000858,0.00175) (-0.0010059,0.00094396) (-0.001778,-0.001002) (-0.0010059,-0.002948) (0.000858,-0.003754) (0.0027219,-0.002948)
O centroid	(0.000858,-0.001002)
ECA diameter at exit	0.00395

TABLE C.2: Parameter values for Stage 1 model of artery 2

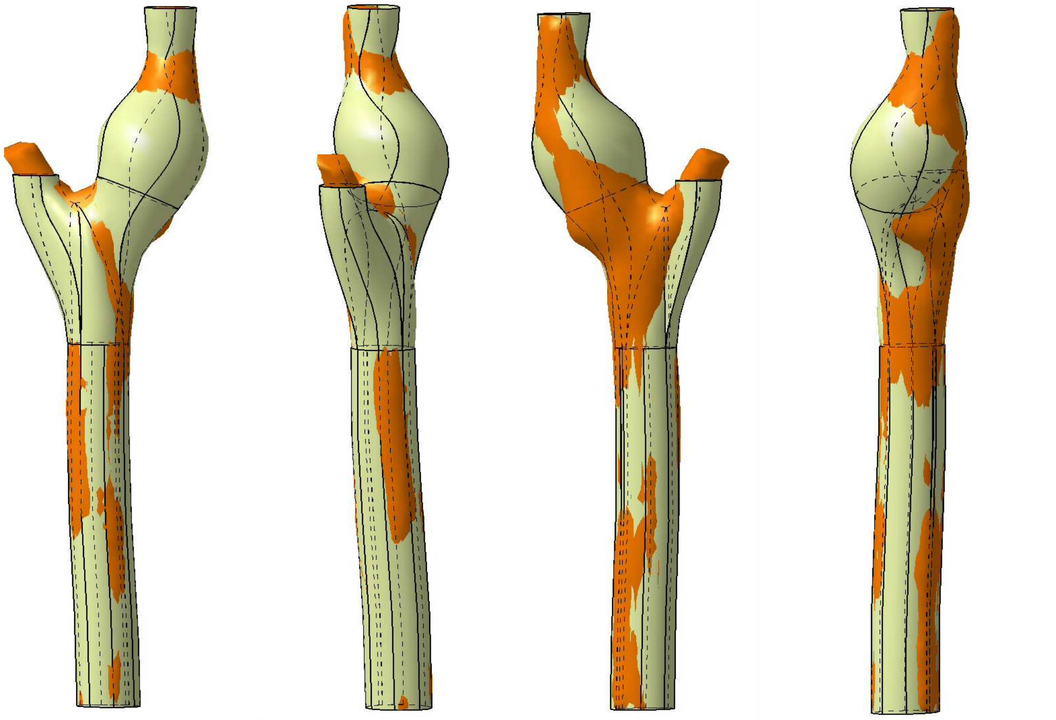


FIGURE C.5: Difference in geometry between the patient artery (orange) and the parametric CAD geometry (beige) of artery 2

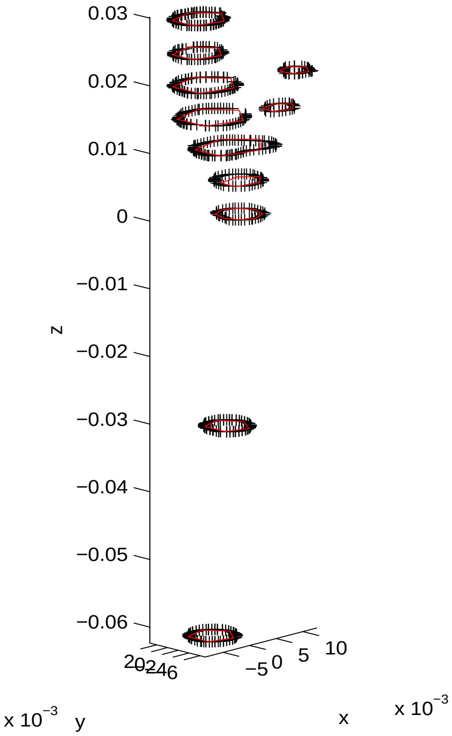


FIGURE C.6: Ellipses fitted to the point cloud of artery 3 at specified z-plane intersections

Parameters	Value
A (m)	0.015095
B (deg)	61.9275
C (m)	0.03019
ECArootX (m)	0.005315
ECArootY (m)	-0.004703
ICArootX (m)	0.0024
ICArootY (m)	-0.003843
BifurcationRoot (m)	-0.0005015
ECAcentX(m)	0.010551
ECAcentY (m)	-0.004972
ICAcentX (m)	0.020127
ICAcentY (m)	-0.0043475
I (m)	-0.002117
J(x,y) (m)	(0.003422,0.000127) (0.0011849,0.0031143) (-0.0024347, 0.0019732) (-0.0024347,-0.0017192) (0.0011849,-0.0028603)
J(z) (m)	-0.062363
K(x,y) (m)	(0.004208,-0.0017455) (0.0021295,0.0012594) (-0.0012335,0.00011161) (-0.0012335,-0.0036026) (0.0021295,-0.0047504)
K(z) (m)	-0.031182
L centroid(x,y) (m)	(-0.0031095,-0.001932)
L_r (m)	0.009353/2
M centroid(x,y) (m)	(-0.004621,-0.0020985)
R (m)	0.025158
M_r (m)	0.007438/2
N (m)	0.0082869
O(x,y) (m)	(0.005592,-0.0025125) (0.0047003,-0.00030244) (0.0025475,0.000613) (0.00039471,-0.00030244) (-0.000497,-0.0025125) (0.00039471,-0.0047226) (0.0025475,-0.005638) (0.0047003,-0.0047226)
O centroid	(0.0025475,-0.0025125)
ECA diameter at exit	0.004269

Table C.3: Parameter values for Stage 1 model of artery 3

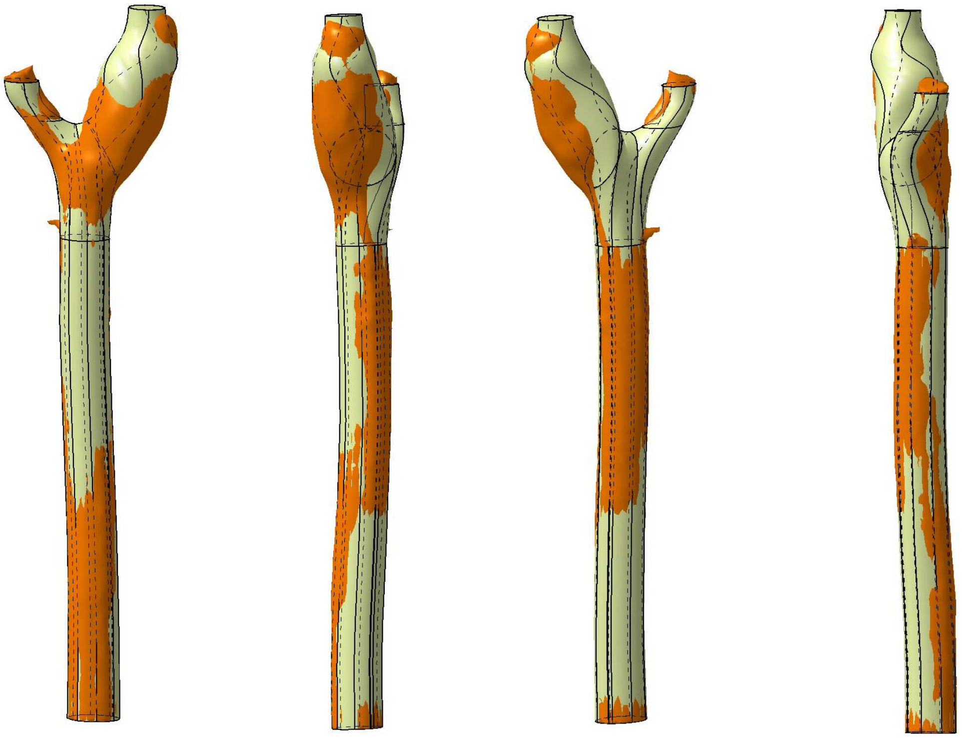


Figure C.7: Difference in geometry between the patient artery (orange) and the parametric CAD geometry (beige) of artery 3

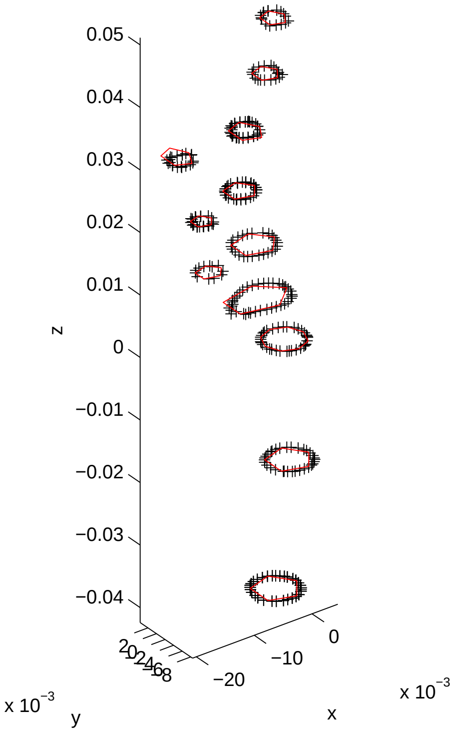


Figure C.8: Ellipses fitted to the point cloud of artery 4 at specified z-plane intersections

Parameters	Value
A (m)	0.017051
B (deg)	4.9843
C (m)	0.051152
ECArootX (m)	-0.014344
ECArootY (m)	-0.00593
ICArootX (m)	-0.010294
ICArootY (m)	-0.005297
BifurcationRoot (m)	-0.0045
ECAcentX(m)	-0.017615
ECAcentY (m)	-0.0065
ICAcentX (m)	0.00836
ICAcentY (m)	-0.00325
I (m)	-0.00256
J(x,y) (m)	(0.003766,0.000238)
	(0.0012802,0.0036181)
	(-0.0027419, 0.002327)
	(-0.0027419,-0.001851)
	(0.0012802,-0.0031421)
J(z) (m)	-0.042408
K(x,y) (m)	(0.004429,-0.0016815)
	(0.0020658,0.0014974)
	(-0.0017578,0.00028317)
	(-0.0017578,-0.0036462)
	(0.0020658,-0.0048604)
K(z) (m)	-0.021204
L centroid(x,y) (m)	(-0.0067095,-0.0021235)
L_r (m)	0.004669/2
M centroid(x,y) (m)	(-0.0037275,-0.0026835)
R (m)	0.042627
M_r (m)	0.005/2
N (m)	0.006943
O(x,y) (m)	(0.001798,-0.004008)
	(0.00082721,-0.0016251)
	(-0.0015165,-0.000638)
	(-0.0038602,-0.0016251)
	(-0.004831,-0.004008)
	(-0.0038602,-0.0063909)
	(-0.0015165,-0.007378)
	(0.00082721,-0.0063909)
O centroid	(-0.0015165,-0.004008)
ECA diameter at exit	0.004669

TABLE C.4: Parameter values for Stage 1 model of artery 4

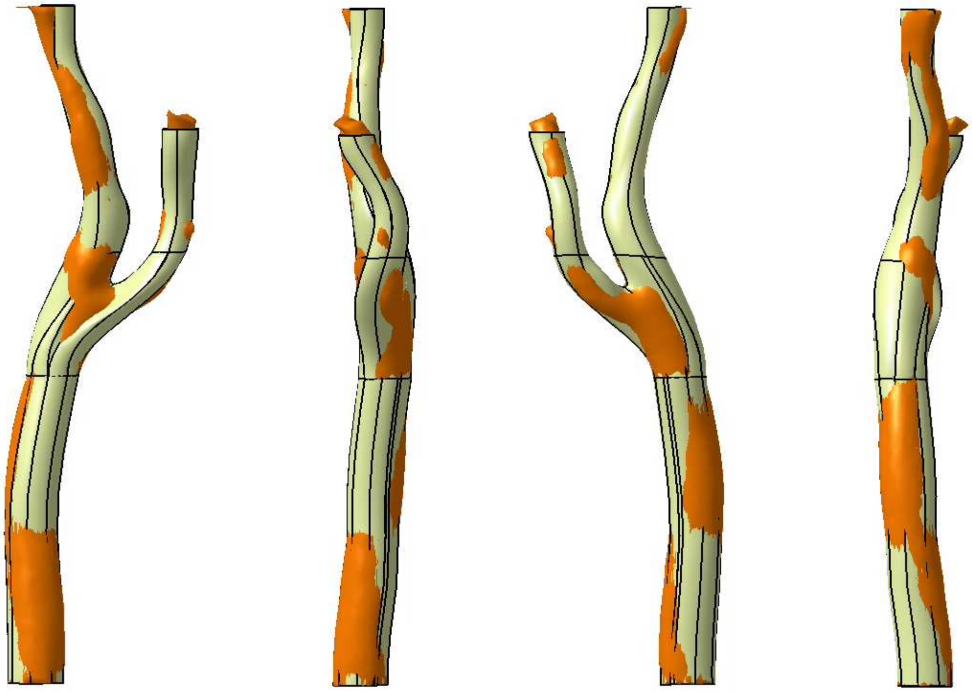


FIGURE C.9: Difference in geometry between the patient artery (orange) and the parametric CAD geometry (beige) of artery 4

C.2 Automated Creation of New Bump Deformation

The matlab function outlined below writes out a visual basic script macro which can be run in batch using CATIA. `BCOUNT` describes the current number of deformations applied to the model, `bump_h` is the bump height, `Pt_coords` are the coordinates of the point cloud point whose nearest CAD model point has the greatest error. `Join` is the limit curve used, `curvature` is the curvature of the deformation and `Intersect_Pt_start` is the point number at which the sample points of the intersections from the previous deformation starts.

```
function write_catvbs(BCOUNT,bump_h,Pt_coords,Join,curvature)

Intervals=[1,1/15,2/15,3/15,4/15,5/15,6/15,7/15,8/15,9/15,10/15,11/15,12/15,13/15,14/15];
NEW_CATVBS=fopen(‘Stage2_bumpN.catvbs’,’w’);
fprintf(NEW_CATVBS,‘Language="VBSCRIPT"\n\n’);
fprintf(NEW_CATVBS,‘Sub CATMain()\n\n’);
fprintf(NEW_CATVBS,‘Set documents1 = CATIA.Documents\n’);
fprintf(NEW_CATVBS,[‘Set partDocument1=documents1.Open("CAD_bump‘num2str(BCOUNT-1)’.CATPart")\n’]);
```

A new point at the position of the cloud point whose closest CAD point has the largest error is created:

```
fprintf(NEW_CATVBS,‘Set part1=partDocument1.Part\n’);
fprintf(NEW_CATVBS,‘Set hybridShapeFactory1=part1.HybridShapeFactory\n’);
fprintf(NEW_CATVBS,[‘Set hybridShapePointCoord1=hybridShapeFactory1.AddNewPointCoord(‘num2str(
Pt_coords(1))’,‘num2str(Pt_coords(2))’,‘num2str(Pt_coords(3))’)\n’]);
fprintf(NEW_CATVBS,‘Set hybridBodies1=part1.HybridBodies\n’);
fprintf(NEW_CATVBS,‘Set hybridBody1=hybridBodies1.Item("ICA_BULB")\n’);
fprintf(NEW_CATVBS,‘hybridBody1.AppendHybridShape hybridShapePointCoord1\n’);
fprintf(NEW_CATVBS,‘part1.InWorkObject=hybridShapePointCoord1\n’);
fprintf(NEW_CATVBS,‘part1.Update\n’);
```

This new point is projected onto the artery surface to act as the centre of the deformation:

```
fprintf(NEW_CATVBS,‘Set reference1=part1.CreateReferenceFromObject(hybridShapePointCoord1)\n’);
fprintf(NEW_CATVBS,‘Set hybridShapes1=hybridBody1.HybridShapes\n’);
fprintf(NEW_CATVBS,[‘Set hybridShapeBump1=hybridShapes1.Item("Bump.‘num2str(BCOUNT-1)’")\n’]);
fprintf(NEW_CATVBS,‘Set reference2=part1.CreateReferenceFromObject(hybridShapeBump1)\n’);
fprintf(NEW_CATVBS,‘Set hybridShapeProject1=hybridShapeFactory1.AddNewProject(reference1,reference2)
\n’);
fprintf(NEW_CATVBS,‘hybridShapeProject1.SolutionType=0\n’);
fprintf(NEW_CATVBS,‘hybridShapeProject1.Normal=True\n’);
fprintf(NEW_CATVBS,‘hybridShapeProject1.SmoothingType=0\n’);
fprintf(NEW_CATVBS,‘hybridBody1.AppendHybridShape hybridShapeProject1\n’);
fprintf(NEW_CATVBS,‘part1.InWorkObject=hybridShapeProject1\n’);
fprintf(NEW_CATVBS,‘part1.Update\n’);
```

A line is created normal to the surface at the projected point to describe the direction along which the deformation will be made:

```
fprintf(NEW_CATVBS,‘Set reference3=part1.CreateReferenceFromObject(hybridShapeBump1)\n’);
fprintf(NEW_CATVBS,‘Set reference4=part1.CreateReferenceFromObject(hybridShapeProject1)\n’);
fprintf(NEW_CATVBS,‘Set hybridShapeLineNormal1=hybridShapeFactory1.AddNewLineNormal(reference3,reference4,
0.000000,3.000000,False)\n’);
fprintf(NEW_CATVBS,‘hybridBody1.AppendHybridShape hybridShapeLineNormal1\n’);
fprintf(NEW_CATVBS,‘part1.InWorkObject=hybridShapeLineNormal1\n’);
fprintf(NEW_CATVBS,‘part1.Update\n’);
```

Create the new deformation:

```
fprintf(NEW_CATVBS,‘Set reference5=part1.CreateReferenceFromObject(hybridShapeBump1)\n’);
fprintf(NEW_CATVBS,[‘Set hybridShapeAssemble1=hybridShapes1.Item("Join.‘num2str(Join)’")\n’]);
```

```
fprintf(NEW_CATVBS,‘Set reference6=part1.CreateReferenceFromObject(hybridShapeAssemble1)\n’);
fprintf(NEW_CATVBS,‘Set reference7=part1.CreateReferenceFromObject(hybridShapeProject1)\n’);
fprintf(NEW_CATVBS,‘Set reference8=part1.CreateReferenceFromObject(hybridShapeLineNormal1)\n’);
fprintf(NEW_CATVBS,‘Set hybridShapeBump2=hybridShapeFactory1.AddNewBump(reference5)\n’);
fprintf(NEW_CATVBS,‘hybridShapeBump2.LimitCurve=reference6\n’);
fprintf(NEW_CATVBS,‘hybridShapeBump2.DeformationCenter=reference7\n’);
fprintf(NEW_CATVBS,‘hybridShapeBump2.DeformationDir=reference8\n’);
fprintf(NEW_CATVBS,[‘hybridShapeBump2.DeformationDistValue=‘num2str(bump_h)’\n’]);
fprintf(NEW_CATVBS,‘Set parameters1=part1.Parameters\n’);
fprintf(NEW_CATVBS,[‘Set realParam1=parameters1.Item("Part2\ICA_BULB\Bump.‘num2str(BCOUNT)
’\Center curvature ratio")\n’]);
fprintf(NEW_CATVBS,‘realParam.value=0.01\n’);
fprintf(NEW_CATVBS,‘hybridShapeBump2.CenterTension=realParam1\n’);
fprintf(NEW_CATVBS,‘hybridBody1.AppendHybridShape hybridShapeBump2\n’);
fprintf(NEW_CATVBS,‘part1.InWorkObject=hybridShapeBump2\n’);
fprintf(NEW_CATVBS,‘part1.Update\n’);
```

The model is now re-intersected with the new deformation:

```
for each plane
   fprintf(NEW_CATVBS,[‘Set hybridShapePlaneOffset‘
   num2str(i)’=hybridShapes1.Item("Plane.‘num2str(Plane(i))’")\n’]);
   fprintf(NEW_CATVBS,[‘Set reference‘num2str(j(i))’=part1.
   CreateReferenceFromObject(hybridShapePlaneOffset‘num2str(i)’)\n’]);
   fprintf(NEW_CATVBS,[‘Set reference‘num2str(j(i)+1)’=part1.
   CreateReferenceFromObject(hybridShapeBump2)\n’]);
   fprintf(NEW_CATVBS,[‘Set hybridShapeIntersection‘num2str(i)’=hybridShapeFactory1.
   AddNewIntersection(reference‘num2str(j(i))’,reference‘num2str(j(i)+1)’)\n’]);
   fprintf(NEW_CATVBS,[‘hybridShapeIntersection‘num2str(i)’.PointType=0\n’]);
   fprintf(NEW_CATVBS,[‘hybridBody1.AppendHybridShape hybridShapeIntersection‘num2str(i)’
   \n’]);
   fprintf(NEW_CATVBS,[‘part1.InWorkObject=hybridShapeIntersection‘num2str(i)’\n’]);
   fprintf(NEW_CATVBS,‘part1.Update\n’);
end
```

On the intersecting planes where the intersection passes through both the ICA and the ECA, a geometric entity nearest to either the ICA or the ECA is chosen to define separate intersections for each of these arteries on the same plane.

Finally, each of the intersections is sampled with 15 equally spaced points.

```
for each intersection
k=1;
l=1;
   for m=1:15
      fprintf(NEW_CATVBS,[‘Set hybridShapeIntersection‘num2str(i)’=hybridShapes1.Item("
      Intersect.‘num2str(INTERSECTION)’")\n’]);
       fprintf(NEW_CATVBS,[‘Set reference‘num2str(l)’=part1.CreateReferenceFromObject(
      hybridShapeIntersection‘num2str(i)’)\n’]);
       fprintf(NEW_CATVBS,[‘Set hybridShapePointOnCurve‘num2str(k)’=hybridShapes1.Item("
      Point.‘num2str(Intersect_Pt_start+(15*(i-1)))’")\n’]);
       fprintf(NEW_CATVBS,[‘Set reference’num2str(l+1)’=part1.CreateReferenceFromObject(
      hybridShapePointOnCurve‘num2str(k)’)\n’]);
       fprintf(NEW_CATVBS,[‘Set hybridShapePointOnCurve‘num2str(k+1)’=hybridShapeFactory1.
      AddNewPointOnCurveWithReferenceFromPercent(reference‘num2str(l)’,reference‘num2str(l+1)’,
      ‘num2str(Intervals(m))’,False)\n’]);
       fprintf(NEW_CATVBS,[‘hybridBody1.AppendHybridShape hybridShapePointOnCurve‘num2str(k+1)’\n’]);
       fprintf(NEW_CATVBS,[‘part1.InWorkObject=hybridShapePointOnCurve‘num2str(k+1)’\n’]);
       fprintf(NEW_CATVBS,‘part1.Update\n’);
       k=k+2;
       l=l+2;
   end
end
```

This geometry is exported as a step file. In matlab, a simple code is run to extract the cartesian coordinates of the points sampling the intersections before calculating the new error between these deformed points and the real artery point cloud.

```
fprintf(NEW_CATVBS,‘Set fs=CATIA.FileSystem\n’);
fprintf(NEW_CATVBS,‘Set f=fs.GetFile("DesignPointInfo.txt")\n’);
fprintf(NEW_CATVBS,‘Set ts=f.OpenAsTextStream("ForReading")\n’);
fprintf(NEW_CATVBS,‘dataline=ts.ReadLine\n’);
fprintf(NEW_CATVBS,‘DesPointNum=CDbl(dataline)\n’);
fprintf(NEW_CATVBS,‘StpFile=ts.ReadLine\n’);
fprintf(NEW_CATVBS,‘ts.Close\n’);
fprintf(NEW_CATVBS,‘partDocument1.ExportData StpFile,"stp"\n’);
fprintf(NEW_CATVBS,‘End Sub\n’);
fclose(NEW_CATVBS);
```

C.3 Flow Simulation

A user-defined function is used to impart the pulsatile waveform onto the velocity inlet at the entry to the CCA of the arteries. The Fourier series representation of the inflow velocity is given by

$$v = \frac{1}{16\pi}\left[a_0 + \sum_{n=1}^{13}\left(a_n \cos\left(\frac{2n\pi t}{0.917}\right) + b_n \sin\left(\frac{2n\pi t}{0.917}\right)\right)\right]. \tag{C.1}$$

The Fourier coefficients are given by

$$a = [6.000, 1.076, -2.315, -2.705, -0.639, 1.775, 1.168, -0.202, -0.267, -0.152, 0.146, 0.118, 0.056, 0.010]$$

$$b = [0.000, 2.989, 3.071, -1.979, -1.583, -1.903, 1.065, 0.578, 0.152, -0.202, -0.133, 0.022, 0.050, 0.072]$$

The Fluent log file used to run the simulation was kindly supplied by Dr Neil W. Bressloff and the outline of which is as follows.

```
file read-case filename.msh
define user-defined u-d-m 2
define user-defined use-contributed-cpp yes
define user-defined interpreted-functions inlet_pulse.c , , no
define models solver segregated yes
define models viscous laminar yes
define materials change-create air blood yes , 1035.0 no no yes constant 0.0035 no no no no
no no yes
define operating-conditions operating-pressure ,
define models unsteady-2nd-order yes
define boundary-conditions velocity-inlet inflow n n y y y y "udf" "inlet_y_velocity" yes
define boundary-conditions outflow eca_out 0.3
define boundary-conditions outflow ica_out 0.7
solve set discretization-scheme mom 1
solve set p-v-coupling 22
solve set under-relaxation pressure 0.9 mom 0.9
solve initialize compute-defaults velocity-inlet inflow
solve initialize initialize-flow
solve monitors residual check-convergence y n n n
solve monitors residual convergence-criteria 1e-8 plot yes print yes
solve monitors residual n-save 10000 n-disp 10000
```

```
solve monitors surface set-monitor , wall-shear wall , n no no no ,

solve set time-step 0.0001 solve dual-time-iterate 9170 1
```

Bibliography

N Alexandrov, J E Dennis, R M Lewis, and V Torczon. A Trust Region Framework for Managing the Use of Approximation Models in Optimization. *Structural Optimization*, 15(1):16–23, 1998.

J D Anderson. *Computational Fluid Dynamics; the Basics with Applications*. Mechanical Engineering Series. McGraw-Hill, Singapore, 1995.

L Antiga and D A Steinman. Robust and Objective Decomposition and Mapping of Bifurcating Models. *IEEE Transactions on Medical Imaging*, 23(6):704–713, June 2004.

A H Barr. Global and Local Deformation of Solid Primitives. *Computer Graphics*, 18 (3):21–30, 1984.

P Bézier. Définition Numérique des Courbes et Surfaces II. *Automatisme*, XII:17–21, 1967.

P Bézier. Procédé de Définition Numérique des Courbes et Surfâces non Mathématiques. *Automatisme*, XIII(5):189–196, 1968.

P Bézier. *Mathematical and Practical Possibilities of UNISURF*. Academic Press, in R Barnhill and R Riesenfeld, editors, Computer-Aided Geometric Design edition, 1974.

P Bézier. *Essay de Définition Numérique des Courbes et des Surfaces Expérimentales*. PhD thesis, University of Paris, 1977.

P E Bézier. *Emploi des Machines a Commande Numerique*. Masson et Cie., Paris, 1970.

B K Bharadvaj, R F Mabon, and D P Giddens. Steady Flow in a Model of the Human Carotid Bifurcation. Part I: Flow Visualization. *Journal of Biomechanics*, 15:349–362, 1982.

BioFluidMechanicsLab. Charité - Univerisity of Berlin. Digitized Carotid Bifurcations. http://www.charite.de/biofluidmechanik/gb/forschung/athero_data_GB_0.html.

M I G Bloor and M J Wilson. Using Partial Differential Equations to Generate Freeform Surfaces. *Computer-Aided Design*, 22:202–212, 1990.

G E P Box and N R Draper. *Empirical Model-Building and Response Surfaces.* John Wiley & Sons, Inc, 1986.

G E P Box, W G Hunter, and J S Hunter. *Statistics for Experimenters.* John Wiley & Sons, Inc, 1978.

V Braibant and C Fleury. Shape Optimal Design Using B-Splines. *Computer Methods in Applied Mechanics and Engineering*, 44:247–267, August 1984.

N W Bressloff, J Banks, and K V Bhaskar. Parametric Geometry Definition and Analysis of the Human Carotid Artery Bifurcation. *Proceedings of the IMECE2004: ASME International Mechanical Engineering Congress and RD&D Expo*, November 2004. Anaheim, Calafornia.

A M Brinkman, P B Baker, W P Newman, R Vigorito, and M H Friedman. Variability of Human Coronary Artery Geometry: An Angiographic Study of the Left Anterior Descending Arteries of 30 Autopsy Hearts. *Annals of Biomedical Engineering*, 22: 34–44, 1994.

D S Broomhead and D Loewe. Multivariate Functional Interpolation and Adaptive Networks. *Computer Systems*, 2:321–355, 1988.

J J Carlson, J P Johnston, and C J Sagi. Effects of Wall Shape on Flow Regimes and Performance in Straight Two-Dimensional Diffusers. *Journal of Basic Engineering*, pages 151–160, March 1967.

C G Caro, J M Fitz-Gerald, and R C Schroter. Atheroma and Arterial Wall Shear: Observation, Correlation and Proposal of a Shear Dependent Mass Transfer Mechanism for Atherogenesis. *Proceedings of the Royal Society London [Biology]*, 177:109–159, 1971.

CATIA®. *Catia User Guide.* Dassault Systèmes, Suresnes Cedex, France, 2004.

E Catmull and J Clark. Recursively Generated B-Spline Surfaces on Arbitrarily Topological Meshes. *Computer-Aided Design*, 10:350–355, 1978.

J R Cebral, R Lohner, and J E Burgess. Computer Simulation of Cerebral Artery Clipping: Relevance to Aneurysm Neuro-Surgery Planning. *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering*, September, Barcelona 2000.

N Christianini and J Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel based learning methods).* Cambridge University Press, 1999.

M A Cox. The Numerical Evaluation of B-Splines. *Journal of Institute of Mathematics and its Application*, 10:134–149, 1972.

N Cressie. The Origins of Kriging. *Mathematical Geology*, 22:239–252, 1990.

N A C Cressie. *Statistics for Spatial Data.* John Wiley & Sons Inc, revised edition, 1993.

P F Davies. Spatial Hemodynamics, the Endothelium, and Focal Athergenesis: a Cell Cycle Link? [editorial;comment]. *Circulation Research*, 86:114–116, 2000.

C de Boor. On Calculating with B-Splines. *Journal of Approximation Theory*, 6:50–62, 1972.

C den Hortag, J P C Kleijnen, and A Y D Siem. The Correct Kriging Variance Estimated by Bootstrapping. *Journal of the Operational Research Society*, (2004-46), 2004. ISSN 0924-7815.

Z Ding, K Wang, J Li, and X Cong. Flow Field and Oscillatory Shear Stress in a Tuning-Fork-Shaped Model of the Average Human Carotid Bifurcation. *Journal of Biomechanics*, 34:1555–1562, 2001.

D Doo and M Sabin. Behaviour of Recursive Division Surfaces near Extraordinary Points. *Computer-Aided Design*, 10(6):356–360, 1978.

J L Doob. Stochastic Processes and Statistics. *Proceedings of the National Academy of Sciences*, 20:376–379, 1934.

H L Duhamel du Monceau. *Eleménts de l'Architecture Navale ou Traité Pratique de la Construction des Vaissaux.* Paris, 1752.

A W F Edwards. What Did Fisher Mean by ‘Inverse Probability’ in 1912-1922? *Statistical Science*, 12(3):177–184, 1997.

M H Eres, G E Pound, Z Jiao, J L Wason, F Xu, A J Keane, and S J Cox. Implementation and Utilisation of a Grid-Enabled Problem Solving Environment in Matlab. *Future Generation Computer Systems*, 2005. in press.

G Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, New York, 1990.

G Farin. *In: Handbook of CAGD*. Elsevier, 2002.

J Ferguson. Multivariate Curve Interpolation. *Journal of the Association for Computing Machinery*, 11(2):221–228, 1964.

FIA. *2005 Formula One Technical Regulations*. Federation Internationale de L’Automobile, 2005.

R A Fisher. On An Absolute Criterion for Fitting Frequency Curves. *Messenger of Mathematics*, 41:155–160, 1912.

R A Fisher. On the ‘Probable Error’ of a Coefficient of Correlation Deduced from a Small Sample. *Metron*, 1:3–32, 1921.

R A Fisher. On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society A*, 222:309–368, 1922.

Fluent$^{\text{TM}}$. CFD Flow Modeling Software. www.fluent.com, 2003a.

Fluent$^{\text{TM}}$. *Fluent User Guide*, 2003b.

A R Forrest. Interactive Interpolation and Approximation by Bézier Polynomials. *The Computer Journal*, 15(1):71–79, 1972. Reprinted in CAD 22(9):527-537, 1990.

A I J Forrester, A J Keane, and N W Bressloff. Design and Analysis of ‘Noisy’ Computer Experiments. *AIAA Journal*, 2006. in press.

I Foster. The Grid: A New Infrastructure for 21st Century Science. *Physics Today*, 55 (2), 2002.

I Foster, C Kesselman, J Nick, and S Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002.

R W Fox and S J Kline. Flow Regimes in Curved Subsonic Diffusers. *Journal of Basic Engineering*, 84:303–316, September 1962.

M H Friedman and Z Ding. Variability of the Planarity of the Human Aortic Bifurcation. *Medical Engineering and Physics*, 20:469–472, September 1998.

D L Fry. Acute Vascular Endothelial Changes Associated with Increased Blood Velocity Gradients. *Circulation Research*, 22:165–197, 1968.

Geodise. The Geodise Project. http://www.geodise.org, 2002.

D Ghate, A Isaacs, K Sudhakar, P M Mujumdar, and A G Marathe. 3D-Duct Design Using Variable Fidelity Method. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York*, September 2004.

Globus. Globus Project. http://www.globus.org, 2002.

R Glowinski and O Pironneau. On the Numerical Computation of the Minimum-Drag Profile in Laminar Flow. *Journal of Fluid Mechanics*, 72(2):385–389, 1975.

D E Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, 1989.

G Guadagni, F Migliavacca, G Dubini, and E L Bove. Simulations of Surgical Planning for Fontan Procedures. *Proceedings of the ASME Bioengineering Conference*, 50:911–912, 2001.

H M Gutmann. A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*, 19(3):201–227, 2001.

P Hajela. Nongradient Methods in Multidisciplinary Design Optimization. *Journal of Aircraft*, 36(1):255–265, 1999.

P Hajela and L Berke. Neurobiological Computational Models in Structural Analysis and Design. *Computers and Structures*, 41(4):657–667, 1991.

R M Hicks and P A Henne. Wing Design by Numerical Optimization. *Journal of Aircraft*, 15(7):407–412, July 1978.

R M Hicks, E M Murman, and G N Vanderplaats. An Assessment of Airfoil Design by Numerical Optimization. *NASA*, TM X-3092, 1974.

D W Holdsworth, C J D Norley, R Frayne, D A Steinman, and B K Rutt. Characterization of Common Carotid Artery Blood-Flow Waveforms in Normal Human Subjects. *Physiological Measurement*, 20:219–240, 1999.

N Hoyle, N W Bressloff, and A J Keane. Design Optimization of an Engine Air Intake. *Proceedings of the 3rd MIT Conference on Computational Fluid and Solid Mechanics*, June 2005.

N Hoyle, N W Bressloff, and A J Keane. Design Optimization of a Two-Dimensional Subsonic Engine Air Intake. *AIAA Journal*, June 2006. accepted for publication.

W M Hsu, J F Hughes, and H Kaufman. Direct Manipulation of Free-Form Deformation. *Computer Graphics*, 26(2):177–184, 1992.

A Jameson. Aerodynamic Design Through Control Theory. *Journal of Scientific Computing*, 3:233–260, 1988.

D R Jones. A Taxonomy of Global Optimization Based on Response Surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

D R Jones, M Schonlau, and W J Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

A Keane and P Nair. *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. Wiley, 2005.

A J Keane. The Options Design Exploration System Reference Manual and User Guide - version b3.1. http://www.soton.ac.uk/∼ajk/options.ps, 2002.

A J Keane. Wing Optimization Using Design of Experiment, Response Surface and Data Fusion Methods. *Journal of Aircraft*, 40(4):741–750, 2003.

A J Keane. Design Search and Optimisation using Radial Basis Functions with Regression Capabilities. *Adaptive Computing in Design and Manufacture (Ed: I C Parmee)*, VI:39–49, 2004.

A J Keane and G M Robinson. Experiences with Optimizers in Design. *Proceedings of the 14th International Symposium on Air Breathing Engines (ISABE), AIAA, Florence*, ISABE 99-7176:A99–34177, 1999.

A I Khuri and J A Cornell. *Response Surfaces, Designs and Analyses.* Marcel Dekker Inc, 1987.

S J Kline, D E Abbott, and R W Fox. Optimum Design of Straight-Walled Diffusers. *Journal of Basic Engineering*, 81(3):321–331, 1959.

J Krautter and S Parizot. Système d'Aide à la Définition et à l'Usinage des Surfâces de Carosserie. *Journal de la SIA*, 44:581–586, 1971. Special issue: La Commande Numérique, P Bézier, guest editor.

D G Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139, 1951.

D N Ku. Blood Flow in Arteries. *Annual Reviews of Fluid Mechanics*, 29:399–434, 1997.

H J Lamousin and W N Waggenspack. NURBS-Based Free-Form Deformation. *IEEE Computer Graphics and Applications*, 14(6):1–9, 1994.

B E Launder and D B Spalding. The Numerical Computation of Turbulent Flows. *Computer Methods in Applied Mechanics and Engineering*, 3(2):269–289, 1974.

S Lim and H Choi. Optimal Shape Design of a Two-Dimensional Asymmetric Diffuser in Turbulent Flow. *AIAA Journal*, 42(6):1154–1169, 2004.

R Liming. *Practical Analytical Geometry with Applications to Aircraft.* Macmillan, 1944.

M Locatelli. Bayesian Algorithms for One-Dimensional Global Optimization. *Journal of Global Optimization*, 10(1):57–76, 1997.

D MacLaren. Formulas for Fitting a Spline Curve through a Set of Points. Appl. Math. Rpt. 2, Boeing, 1958.

J I Madsen. *Design Optimization of Internal Flow Devices.* PhD thesis, Aalborg University, 1998.

J I Madsen, N Olhoff, and T J Condra. Optimization of Straight, Two-Dimensional Diffusers by Wall Contouring and Guide Vane Insertion. *Proceedings of the 3rd World Congress on Structural and Multidisciplinary Optimization*, Buffalo, NY, USA, May 17-21 1999.

J I Madsen, W Shyy, and R T Haftka. Response Surface Techniques for Diffuser Shape Optimization. *AIAA Journal*, 39(9):1512–1518, 2000.

G Matheron. Principles of Geostatics. *Economic Geology*, 58:1246–1266, 1963.

Matlab ®. Matlab 6.5. http://www.mathworks.com, 2002.

MAYA. *Maya ® 7 Unlimited*. Autodesk, 2005.

R Mead. *The Design of Experiments*. Cambridge University Press, Cambridge, England, UK, 1988.

J S Milner, J A Moore, B K Rutt, and D A Steinman. Hemodynamics of Human Carotid Artery Bifurcations: Computational Studies in Models Reconstructed from Magnetic Resonance Imaging of Normal Subjects. *Journal of Vascular Surgery*, 27(1):143–156, July 1998.

D C Montgomery. *Design and Analysis of Experiments*. Wiley, fifth edition, 2000.

J A Moore, D A Steinman, D W Holdsworth, and C R Ethier. Accuracy of Computational Hemodynamics in Complex Arterial Geometries Reconstructed from Magnetic Resonance Imaging. *Annals of Biomedical Engineering*, 27:32–41, 1999.

R H Myers. *Response Surface Methodology*. Allyn and Bacon, 1976.

R H Myers and D C Montgomery. *Response Surface Methodology: Process and Produce Optimization Using Design of Experiments*. Wiley, New York, 1995.

OGSI. Open Grid Services Infrastructure. 2003.

G N Patterson. Modern Diffuser Design. *Aircraft Engineering*, 10:267–273, September 1938.

L Piegl and W Tiller. *The NURBS Book*. Springer-Verlag, 2nd edition, 1996. ISBN 3540615458.

M J D Powell. *Radial Basis Functions for Multivariate Interpolation: A Review*. Clarendon Press Institute Of Mathematics And Its Applications Conference Series: Algorithms for Approximation. Clarendon Press, Oxford, UK, 1987. pp 143-167.

A C Raalamb. *Skepps Bygerriet eller Adelig Öfning*. Tionde Tom, Stockholm, 1691.

L R Reneau. Performance and Design of Straight, Two-Dimensional Diffusers. *Journal of Basic Engineering*, pages 141–150, March 1967.

L R Reneau, J P Johnston, and S J Kline. Performance and design of straight, two-dimensional diffusers. *Journal of Basic Engineering*, pages 141–150, March 1967.

J Sacks, W J Welch, T J Mitchell, and H P Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–435, 1989.

J A Samareh. Survey of Shape Parameterization Techniques for High Fidelity Multidisciplinary Shape Optimization. *AIAA Journal*, 39(5):877–884, 2001.

I J Schoenberg. Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions. *Quarterly of Applied Mathematics*, 4:45–99 and 112–141, 1946.

M Schonlau. *Computer Experiments and Global Optimization.* PhD thesis, University of Waterloo, Ontario, Canada, 1997.

T W Sederberg and S R Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics*, 20(4):151–160, 1986.

J N Siddall. *Optimal Engineering Design: Principles and Applications.* Marcel Dekker Inc, New York, 1982.

O Smedby. Geometrical Risk Factors for Atherosclerosis in the Femoral Artery: A Longitudinal Angigraphic Study. *Annals of Biomedical Engineering*, 26(3):391–397, 1998.

A Sóbester and A J Keane. Empirical Comparison of Gradient-Based Methods on an Engine-Inlet Shape Optimization Problem. *AIAA 2002-5507*, 2002.

A Sóbester, S J Leary, and A J Keane. On the Design of Optimization Strategies Based on Global Approximation Models. *Journal of Global Optimization*, 33(1), 2005.

I M Sobol. On the Systematic Search in a Hypercube. *SIAM Journal of Numerical Analysis*, 16(5):790–793, 1979.

P R Spalart and S R Allmaras. A One-Equation Turbulence Model for Aerodynamic Flows. *30th Aerospace Sciences Meeting and Exhibit, Reno, NV*, January 1992.

R B Statnikov and J B Matusov. *Multicriteria Optimization and Engineering.* Chapman and Hall, New York, 1995.

D A Steinman. Image-Based CFD Modelling in Realistic Geometries. *Annals of Biomedical Engineering*, 30:483–497, 2002.

D A Steinman, J S Milner, C J Noreley, S P Lownie, and D W Holdsworth. Computed Blood Flow Dynamics in an Anatomically Realistic Cerebral Aneurysm. *Proceedings of the Fourth World Congress of Biomechanics*, August, Calgary 2002a.

D A Steinman, J B Thomas, H M Ladak, J S Milner, B K Rutt, and J D Spence. Reconstruction of Carotid Bifurcation Hemodynamics and Wall Thickness using CFD and MRI. *Magnetic Resonance in Medicine*, 47(1):149–159, Jan 2002b.

D A Steinman, D A Vorp, and C R Ethier. Computational Modelling of Arterial Biomechanics: Insights into Pathogenesis and Treatment of Vascular Disease. *Journal of Vascular Surgery*, 37:1118–1128, May 2003.

C A Taylor, M T Draney, J P Ku, D Parker, B N Steele, and K Wang et al. Predictive Medicine: Computational Techniques in Therapeutic Decision Making. *Computer Aided Surgery*, 4:231–247, 1999.

J B Thomas, L Antiga, S Che, J S Milner, D A Steinman, J D Spence, B K Rutt, and D A Steinman. Variation in the Carotid Bifurcation Geometry of Young Versus Older Adults: Implications for Geometric Risk of Athersclerosis. *Stroke*, 36(11):2450–2456, Nov 2005.

H Ugail and M J Wilson. Efficient Shape Parameterisation for Automatic Design Optimisation using a Partial Differential Equation Formulation. *Computers and Structures*, 81:2601–2609, 2003.

A Watt and M Watt. *Advanced Animation and Rendering Techniques.* Addison-Wesley, 1992.

H White, A R Gallant, K Kornik, M Stinchcombe, and J Wooldridge. *Artificial Neural Networks: Approximation and Learning Theory.* Blackwell Publishers, 1992.

M H Wholey, M Wholey, P Bergeron, E B Diethrich, M Henry, J C Laborde, K Mathias, S Myla, G S Roubin, F Shawl, J G Theron, J S Yadav, G Dorros, J Guimaraens, R Higashida, V Kumar, M Leon, M Lim, H Londero, J Mesa, S Ramee, A Rodriguez, K Rosenfield, G Teitelbaum, and C Vozzi. Current Global Status of Carotid Artery Stent Placement. *Catheterization and Cardiovascular Diagnosis*, 44(1):1–6, 1998.

P A Wolf, R B D'Agnostino, A J Belanger, and W B Kannel. Probability of Stroke: A Risk Profile from the Framingham Study. *Stroke*, 22(3):312–318, 1991.

S Wolfram. *A New Kind of Science.* Wolfram Media, 2002.

G Xue, W Song, S J Cox, and A J Keane. Numerical Optimisation as Grid Services for Engineering Design. *Journal of Grid Computing*, 2(3):223–238, 2004.

J S Yadav, G S Roubin, S Iyer, J Vitek, P King, and W D Jordan. Elective Stenting of the Extracranial Carotid Arteries. *Circulation*, 95(3):76–81, 1997.

D Yuret and M de la Maza. Dynamic Hill Climbing: Overcoming the Limitations of Optimization Techniques. *Procedings of the 2nd Turkish Symposium on AI and ANN*, pages 254–260, 1993.

J Zhang, C K Chu, and V Modi. Design of Plane Diffusers in Turbulent Flow. *Inverse Problems in Engineering*, 2:85–102, 1995.

W-L Zhang, D D Knight, and D Smith. Automated Design of a Three-Dimensional Subsonic Diffuser. *Journal of Propulsion and Power*, 16(6):1132–1140, 2000.