University of Southampton

Faculty of Engineering, Science and Mathematics

School of Engineering Sciences

Aerodynamic Design Optimization Using Flow Feature Parameterization

by

Thomas Robin Barrett

Thesis for the degree of Doctor of Philosophy

November, 2007

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE and MATHEMATICS
SCHOOL OF ENGINEERING SCIENCES

Doctor of Philosophy

AERODYNAMIC DESIGN OPTIMIZATION USING FLOW FEATURE PARAMETERIZATION

By Thomas Robin Barrett

Design optimization methods using high-fidelity computational fluid dynamics simulations are becoming increasingly popular in the area of aerodynamic design, sustaining the desire to make these methods more computationally efficient. Such design strategies typically define the aerodynamic product using a parametric model of the geometry, but this can often require a large number of design variables, increasing the computational cost. This thesis proposes that a parametric model of aerodynamic flow features, rather than geometry, can be a parsimonious method of representing designs, giving a reduction in the number of design parameters required for optimization. The parameterization of flow features is coupled with inverse design, in order to recover the corresponding geometry. While an expensive analysis code is used in evaluating design performance, computational cost is reduced by using a low-fidelity code in the inverse design process. This newly presented method is demonstrated using four case studies in 2-D airfoil design, in which the parameterized flow feature is the surface pressure distribution, and two case studies for 3-D wing design, in which the spanwise loading distribution is parameterized. These strategies are consistently compared against a benchmark design search method which uses a conventional parameterization of the geometry. The two methods are described in detail, and their relative performance is analysed and discussed. The newly presented method is found to converge towards the optimum design significantly more quickly than the benchmark method, providing designs with greater performance for a given computational expense. A parameterization of flow features can generate designs with higher quality and detail than a geometry-based method of the same dimensionality.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I am sincerely grateful to many people for their guidance and support throughout my three years of study in the Computational Engineering and Design research group. The work in this thesis has been funded by a studentship from the University of Southampton School of Engineering Sciences, which is gratefully received. The support and guidance from the project supervisors, Dr. Neil Bressloff and Professor Andy Keane, has been fabulous, and I could not have wished for a better supervisory team. In addition, the initial leadership from Robert Lewis at Advantage CFD is greatly appreciated. My deepest thanks go to many people in the research group for their emotional, technical and editorial support, as well as their friendship; in particular: Alex and Jen Forrester, Nici Hoyle, Tony Scurr, András Sóbester, Praveen Thokala, David Toal and Narcis Ursache. To generate the results presented in this thesis has required over 24000 hours of run time on the Microsoft Compute Cluster; I am enormously grateful to Ivan Voutchkov for his hard work in maintaining the cluster, and for the countless occasions when I have gone to him for help. Following my internship at the Institute of Fluid Science at Tohoku University, I am grateful to Professor Shigeru Obayashi for kindly accommodating me in his research group and Nao Konohara for her helpful support during my stay. Finally, but not least, I want to thank my family. In particular, I thank my Dad for offering so much guidance throughout my education, and I thank Sarah for her unending encouragement.

Declaration of Authorship

I, Thomas Robin Barrett, declare that the thesis entitled

"Aerodynamic Design Optimization Using Flow Feature Parameterization"

and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly 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: Barrett et al. [2006a], Barrett et al. [2006b], Barrett et al. [2006c].

In addition, the thesis conforms, where possible, to British standard BS 4821:1990.

Signed:

Date: November, 2007

Nomenclature

Listed below are the definitions commonly used in this thesis.

x	=	geometrical ordinate in the streamwise direction, for two-dimensional flow
z	=	geometrical ordinate in the vertical direction, for two-dimensional flow
X	=	geometrical ordinate in the streamwise direction, for three-dimensional flow
Y	=	geometrical ordinate in the spanwise direction, for three-dimensional flow
Z	=	geometrical ordinate in the vertical direction, for three-dimensional flow
M	=	flow speed Mach number
Re	=	flow Reynolds number
α	=	angle of attack
C_p	=	pressure coefficient
c_d	=	airfoil drag coefficient, normalized with respect to chord
c_l	=	airfoil lift coefficient, normalized with respect to chord
c	=	airfoil chord
$z_{t\,max}$	=	airfoil maximum thickness
$z_{c\,max}$	=	airfoil maximum mean thickness (maximum camber)
r_{LE}	=	airfoil leading edge radius
C_D	=	wing drag coefficient, normalized with respect to wing projected area
C_L	=	wing lift coefficient, normalized with respect to wing projected area

Chapter 1. Introduction

"*As we have moved from the great pioneers, such as Lanchester, to the modern age of sophisticated computational methods and integrated ways of working, so we have moved from the 'art of compromise' to the 'science of optimisation'*."

The above quotation is taken from a lecture given by Jeff Jupp of Airbus (Jupp [2001]). It portrays succinctly that the process of design is one of compromise. In a modern aircraft design project, these compromises can be vastly complex, but they are not beyond reasoning when modern computational methods are employed in the design process. This design process, and indeed this thesis, is multi-faceted, and concerns aerodynamics analysis, parametric modelling techniques and optimization.

1.1 The Role of Aerodynamics in Design

Historically, the study of aerodynamics has been motivated to a large extent by the dream of achieving and perfecting the act of manned powered flight. As recently as the late 19th century, the flight of birds and insects was thought by some to rely on a mythical "*vital force*", and fierce debate raged amongst the scientific community as to whether such motion could be achieved by an inanimate object. Wilbur Wright commented similarly in 1901 "*nobody will fly for a thousand years*", but two years later thanks to their persistence the Wright brothers achieved their dream. Aerodynamics concerns the prediction of forces and moments acting on a body, when the body is immersed in a fluid (usually air) with relative velocity. The motion of a fluid over a solid body gives rise to two basic flow phenomena: the pressure distribution acting normal to the surface and the shear stress distribution acting tangential to the surface due to the

viscosity of the fluid. Knowledge of these flow phenomena permits the prediction of the net forces and moments on the body, and this is the key interest of an engineer. Engineers strive to use their knowledge of aerodynamics in order to design improved products. However, design decisions are rarely based on experience alone, and rely additionally on the use of some form of analysis. The Wright brothers built their own wind tunnel in their bicycle shop, performing a series of methodical experiments with airfoil and planform geometries in the quest for a more efficient wing design. The scale and complexity of modern aircraft design projects and analysis techniques far exceeds the efforts of the Wrights, but after more than 100 years the same principles of engineering design practice still apply.

Classical analysis of finite wings and airfoils (*infinite* wings) began with the solution of potential flow equations, i.e., the *closed form* solution for inviscid, irrotational, incompressible flow. This was performed with hand calculations until the arrival of the modern digital computer, which allowed large calculations to be rapidly performed. By the 1960's, computational fluid dynamics (CFD) approaches such as the source panel method were standard tools of the aerospace industry. Further development of CFD solution schemes allowed the iterative solution of transonic potential flow, the Euler equations, and subsequently the Reynolds averaged Navier-Stokes (RANS) equations. Modern research into design oriented CFD focuses on turbulence simulation and accurate drag prediction, as well as reducing computational expense.

CFD simulations are relied upon heavily in modern aircraft design projects. Because this is a relatively inexpensive task compared to experimental wind-tunnel testing, CFD can be performed on a large scale and can be easily accessed by all the designers. Typically, varying levels of CFD fidelity and capability are used at different stages in the design process. At the concept design stage, the objective is to assess the technical and economic feasibility of the potential product as a whole, and this consideration should encompass all aspects of the design and its impact on the user. This study is often based on previous designs, and so empirical and calibrated CFD analyses are commonly employed. At the preliminary and detailed design stages the product is broken down into the design of its component parts; higher fidelity analysis methods are used in order to model the relevant flow features in more detail, and obtain a more accurate figure for the predicted drag. The use of more expensive and complex flow simulations in preliminary design means that much of the engineering time is spent in pre-processing and post-processing the analysis. The decision of what modifications should be made to the design is typically a *manual* one, and this is not always obvious based on the results of CFD. There are invariably compromises to be made with each design decision; there could be conflicting aerodynamic load requirements, and structural issues often lead to further compromises in the aerodynamic performance.

Thus, the need to perform a more systematic exploration of engineering compromises, and accelerate the design process, has lead to the increasing use of automated *optimization* methods.

Optimization as a subject in mathematics is very old, but its application in aerodynamic design problems only began in earnest following the widespread use of the modern digital computer. At its most fundamental, optimization is the search for a set of inputs to a function, known as the *objective function*, that result in that function taking its minimum possible value, or, conversely, its maximum possible value for a maximization problem. Despite the optimization techniques being carried over into an aerospace design context, such functions are rarely optimized *per se*. Rather, the non-linear nature of CFD analysis and the requirement for a large and multi-dimensional search space means that this is an exercise in design improvement, hence the term *design search and optimization* (DSO) is used. This line of reasoning is shared by van Egmond [1990]: "*Expectations of achieving the absolute best design invariably lead to maximum disappointment*". In addition, the inputs to an objective function and the computational model can never be all-encompassing, i.e., there will always be real life factors not taken into account in the design search process. Therefore, automated optimization processes are used in industrial situations to complement and accelerate the work of the engineer. A fundamental requirement for performing optimization is a parametric description of the design; for aerodynamic design this parameterization typically involves inputs relating to the external geometry. The selection of an appropriate parameterization is a key factor in the successful application of DSO methods, and this is the focus of this thesis.

1.2 The Role of Parameterization in Design

The basic process of design has been described as *the making of decisions that change the product definition* (Keane and Nair [2005]). In aerodynamic design, these decisions are made based on the results of the aforementioned aerodynamic analysis. The product definition, in its most traditional form, is an engineering drawing communicating the physical dimensions and geometrical features of the product. However, such a primitive description does not readily allow measured and reproducible changes, and certainly prevents automated changes using an optimization algorithm. The need for an efficient and systematic approach to aerodynamic design was recognized in the 1930's by the designers of the NACA[1]

[1] National Advisory Committee for Aeronautics, which subsequently became the National Aeronautics and Space Administration (NASA).

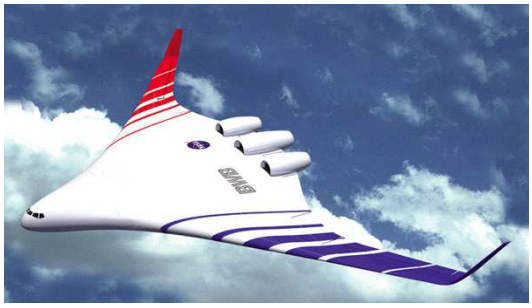
4-digit airfoils (Jacobs et al. [1933]). The designers used a series of successful airfoil shapes to generate one of the earliest examples of a parametric model, i.e., a mathematical description which allows a design to be defined using one or more design variables. Among these design variables are the airfoil thickness and camber quantities. This model facilitates intuitive and precisely measured changes to the shape, and leads to a methodical design process.

In many fields of modern engineering design, the required product definition is becoming increasingly more complex, and designers are forced to adhere to ever more demanding time and budget constraints. The former is particularly true in aerodynamic shape design, where detailed and subtle design changes are often necessary to minimize drag or control a separation point, for example. Thus, a significant research effort has gone into investigating more sophisticated techniques for representing and manipulating designs. At present, the parameterization techniques used in aerospace design can be split into two broad classes: computer aided design (CAD) based methods, and analytical techniques. Modern CAD software is becoming increasingly sophisticated, and is typically capable of quite complex parametric modelling tasks using non-uniform rational B-spline (NURBS) curves and surfaces. CAD is also very accessible to the designer, allowing the input of design data from external software and the export of geometry and mesh data to a CFD pre-processor. There are also many powerful analytical methods which have not yet been adopted in CAD packages. These include basis function methods, partial differential equation methods and free-form deformation. An introduction to the most commonly used parameterization techniques follows in Chapter 2.

A parameterization scheme can be set up to perform global changes in shape or local modifications, or in some cases, both (Hoyle [2006]). In the design process, a *global* method is typically used in the initial concept stage, while increasingly *local* techniques are employed at subsequent stages (Keane and Nair [2005]).

An example of a global technique is a NURBS representation of a full aircraft wing-body configuration; this can typically perform large modifications to the design and is not constrained with respect to its overall form. The ability to perform global shape modifications is a key attribute for parameterization schemes employed for conceptual design tasks. An ability to consider radically different designs is becoming increasingly important, as designers are forced to consider new concepts in a bid to reduce the environmental impact of passenger aircraft. The main aircraft emissions targets set out by the European commission to be achieved by 2020 are a 50% reduction in carbon dioxide emissions per passenger kilometre, and an 80% cut in nitrous oxide emissions (Reneaux [2004]). Drag reduction can contribute significantly to this effort. Aircraft manufacturers therefore have a responsibility to pursue revolutionary

concepts such as the blended wing-body and the supersonic biplane (Yamashita et al. [2007]), which are illustrated in Figure 1-1 (a) and (b), respectively. Both of these concepts promise to dramatically reduce drag; the blended wing body could greatly improve the lift to drag ratio, while the supersonic biplane features very little wave drag at cruise conditions. These examples emphasize the importance of considering a global range of designs at the conceptual stage.



(a)

(b)

Figure 1-1 Future aircraft concepts. (a) A blended wing-body aircraft. (b) The supersonic biplane.

In addition to providing global shape manipulation, a parameterization can be used to perform more local surface modifications. *Local* techniques are able to apply detailed modifications to a specific area of the product. Examples include the use of NURBS control points and bump functions. Provided the constraints on a design problem allow it, local modifications can also result in radically different designs. However, some techniques, such as the above NACA airfoil definition, are based on a set of existing designs, and thus the generated designs are generally evolutionary relative to the input set. Parameterization and shape control techniques have undergone extensive development, and an impressive level of local control can be achieved. However, increasing the degree of surface control usually entails an increase in the number of design variables. This increase in dimensionality increases the complexity of the design task, and when automatic optimization is employed this equates to an increase in the computational cost of a design search. Thus, there is a need for parsimonious parameterization techniques in order to minimize this cost.

1.3 The Need for Efficiency in Design

Despite advances in CFD simulations, computing power and optimization strategies, the computational expense of high-fidelity CFD means that more efficient design optimization methods are still sought after for use in aerodynamic design. A reduction in the number of input parameters as a result of improved parametric modelling is a common contributor to this efficiency. As previously mentioned, this number of input parameters can be large when manipulating geometry, in order to obtain the detail and smoothness required for high-fidelity flow analysis. Additionally, many large-budget, state-of-the-art aerospace design projects result in highly complex and intricate geometries. However, an increase in model complexity usually comes with an increased cost in performing the design search. Thus, the setup of a parameterization scheme is an interesting compromise between achieving a sufficient level of detail and local control, and minimizing the complexity of the design task.

Figure 1-2 illustrates this compromise, by considering the problem of minimizing the drag of an airfoil. A design search is set up in which the airfoil is parameterized using a spline curve method, where the design variables are the positions of control points on the airfoil surface. The drag is determined using a low-fidelity potential flow solver and a genetic algorithm (GA) is employed to search for low-drag designs; note that any optimization algorithm could be used in this example problem. This design search is run ten times, using a different number of variables to define the airfoil shape at each attempt and using the same number of GA iterations, representing a fixed computational budget. Figure 1-2 plots the number of variables used versus the best drag result obtained during the search, also showing some of the optimized airfoil geometries. It can be seen that when 2 design variables are employed the poor degree of local control severely limits the ability to generate low drag designs, but the airfoil shape is regular and smooth. In contrast, when the airfoil is defined using 20 variables, it is clearly possible to achieve very good local control, but the immense complexity of this design search has resulted in a best design which lacks smoothness, and therefore it too has a rather high drag. Many thousands of design iterations, and a large computational budget, would be required for this design search to converge onto a truly optimal design. In this design example, the best compromise is achieved when 10 variables are employed for optimization, since this provides sufficient local control but also converges sufficiently quickly to reach a low-drag airfoil shape. This simple example demonstrates the need for efficient parameterization schemes, which are able to generate detailed and complex changes in shape, but which use a relatively small number of design variables to minimize the cost of a design search.

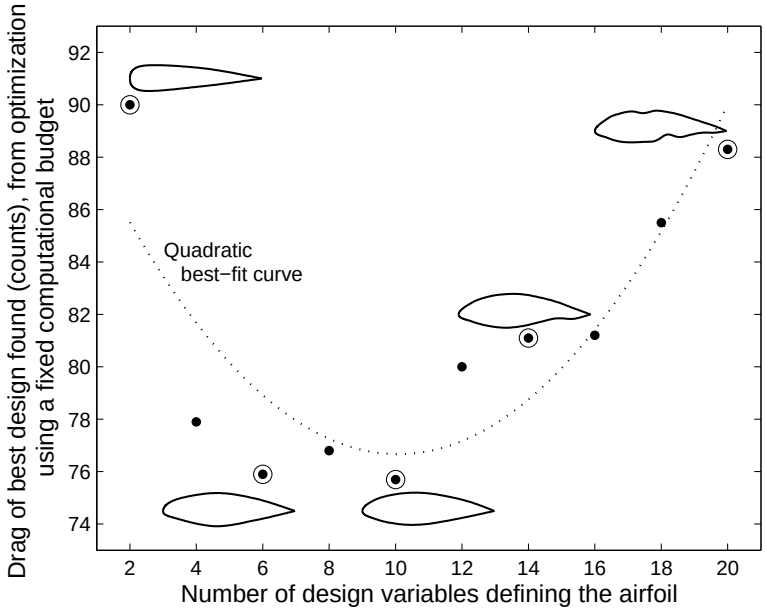


Figure 1-2 An airfoil design parameterization example. As the number of design variables is increased, there is an increase in both the level of local control and the complexity of the optimization task.

A great advantage of CFD is the ability to calculate pressure and velocity data at any point in the discretized domain, be this on the body surface or in the off-surface flow. This data can be used to extract information relating to individual *flow features* such as induced vortices, separation, or the variation of surface pressure or aerodynamic forces. These flow features can be implicitly linked to the analysed geometry. However, while the definition of geometry can be very complex, one can imagine that when subjected to a flow field the resulting flow features are not necessarily so complex. For example, in minimizing the induced drag of a wing one might aim for a simple elliptical lift distribution, while the corresponding shape, for a given flow field, could turn out to be rather more complicated. In such situations, one can postulate that the flow features surrounding the component are potentially simpler to represent parametrically than the geometry. Also, since varying the flow features is likely to have an effect on the entire geometry under analysis, a simple parameterization of flow features may be able to produce quite complex geometrical modifications. Further, such a parameterization will perceivably have an intuitive effect on aerodynamic forces, such as lift and drag.

Of course, by specifying flow features, the designer is then tasked with determining the geometry which realizes these flow features for the given flow conditions. The specification of flow features and subsequent realization of the required geometry is not a new idea. So-called *inverse design* methods have been used widely, particularly in the context of designing an airfoil which generates a prescribed surface pressure distribution; see for example, Dulikravich [1990] or Drela [1989]. The *design* of flow features is

not an intuitive concept, perhaps because aerodynamic effects are invisible whereas engineers are more familiar with geometry manipulation. However, the design of flow features is in some senses more logical; after all, it is the flow features which uniquely establish the forces on a body. The geometry is simply the means of achieving the required flow features. Despite the increasing importance of aesthetics, engineers are concerned rather less by what their product looks like; instead their efforts are focused on improving its performance. Inverse design has been used for various flow feature specifications (for example Qin et al. [2005] used the spanwise lift profile), but not principally as a means of reducing the dimensionality of design optimization problems.

The aim of the work described in this thesis is to investigate the use of flow feature parameterization as a means of improving the efficiency of the design process. It is proposed that this technique can generate detailed and localized geometrical modifications while reducing the total number of defining design variables. The research does not focus on optimization algorithms or CFD techniques, but rather a method in which shape control, inverse design and optimization methods are combined in an attempt to accelerate the process of design. In this work, the application of such methods is to the aerodynamic design of 3-D aircraft wings and 2-D wing sections. Consistently, a comparison is made between two design strategies. The first is treated as a benchmark in aerodynamic shape optimization, in which the geometry is defined parametrically using a representative number of input parameters, and each design selected by the optimization process is analysed using high-fidelity CFD to give a measure of performance. The alternative approach uses a parameterization of flow features, since they can potentially be described using fewer inputs, combined with an inverse design step to recover the required geometry. Following inverse design, each design is evaluated identically to those in the benchmark process. This work is therefore a comparison between these two approaches to parameterization, and investigates the design performance of these methods given a fixed computational budget.

1.4 Thesis Outline

The purpose of thesis is to compare two different parameterization approaches for aerodynamic design, and to demonstrate their relative performance using practical examples. Therefore, the work makes frequent references to the disciplines of parametric modelling, CFD analysis, optimization algorithms and design strategies including inverse design. A background to these items is given in Chapter 2.

In Chapter 3, the concept of flow feature parameterization is introduced, and areas of related work are identified. The proposed parameterization technique is first applied to the design of 2-D airfoils, and is evaluated relative to the benchmark process. The parameterized flow feature for this application is the airfoil surface pressure distribution. The setup of a comparison between the two design methods is described, detailing the parameterization techniques, optimization strategy, CFD analysis setup and inverse design. Chapter 4 reports the results of four case studies for 2-D airfoil design. The objective of the design searches is to minimize the total drag of the airfoil at a single operating point. Drag is calculated using RANS analyses; in the first two case studies a subsonic flow regime is specified, and in two further case studies a transonic flow regime is used. The results from these case studies are analysed in detail and conclusions are drawn.

In Chapter 5, the proposed parameterization method is implemented in a 3-D design scenario. The task is to design a wing-tip device with the objective of minimizing drag. A background to the use of wing-tip devices is documented. Following this, a study is described which investigates an appropriate flow feature to parameterize for this 3-D design problem. The chosen geometry description is the trailing edge chord distribution, and the parameterized flow feature is the spanwise lift distribution. The setup of a comparison between design searches using the flow feature based parameterization and the geometry-based parameterization is described. Chapter 6 reports two case studies for this 3-D wing-tip design task; in one the drag is calculated using Euler simulations, and the other uses RANS simulations. The results from these design searches are analysed and conclusions are drawn.

In Chapter 7, the findings reported in Chapters 4 and 6 are scrutinized in a general sense. Key conclusions and contributions are listed. To finish, recommendations for future work are given describing how the work in this thesis could be taken further.

Chapter 2. Current Practices in Aerodynamic Design

The purpose of this thesis is to present a new approach to the design of components subjected to aerodynamic flows. The work exploits many other computational techniques which are well established and used routinely in design exercises. Before any alternative concept is presented, these current practices are discussed, forming a background to the methods used in later chapters. One of the key themes of this work is parameterization techniques; a number of examples are given below and their relative advantages and disadvantages are discussed. This chapter also outlines the key areas of CFD analysis and optimization algorithms, and introduces the concept of inverse design.

2.1 Parameterization Techniques

Parameterization is the representation of the chosen physical characteristics of a design in terms of one or more numerical parameters, known as *design variables*. These design variables can be either continuously varying or discrete. Typically, such a parameterization is applied to geometry, describing changes to all or part of the design under scrutiny. Using a parametric description of a design, the job of the designer, or indeed, an optimization algorithm, is to select the values of the design variables which give an improved design performance. In engineering design, this selection process is based on the results of analysis, be this computational or experimental. Each variable has a range associated with it; collectively these ranges form the *design space*, with each design taking up a point in this space. At its lowest level, the NACA definition (described below) allows an airfoil to be described using only its camber and thickness quantities, allowing rapid design studies to be performed. In this case the use of only two variables permits a thorough search of the design space, but may not be able to manipulate the airfoil shape in sufficient detail to give the required performance gains. Conversely, a more detailed parameterization may yield improved performance but result in a more lengthy design search due to the higher dimensional search space. Thus, as demonstrated in Chapter 1, the choice of parameterization technique is often a trade-off between the detail and complexity required for a design and the budget of analysis calls

afforded. This has been the subject of extensive research, seeking for representations which reduce the number of design variables while retaining the ability to capture a global range of designs. A survey of many techniques used in the aerospace sector is given by Samareh [1999]. Below is an outline of a number of geometric parameterization techniques relevant to the current work.

2.1.1 NACA Airfoils

During the 1930's, NACA (National Advisory Committee for Aeronautics, which later became NASA) developed one of the earliest examples of geometric parameterization. The experimentally developed definition gives smooth and efficient airfoil shapes, and forms a family known famously as the NACA 4-digit series. These airfoils have been heavily used in the aircraft industry, but are rarely used today having been replaced by more advanced CFD developed shapes. The 4-digit airfoil definition is described in the landmark NACA Report 460 (Jacobs et al. [1933]), and is summarized here.

In this definition, the airfoil is specified using an expression for the camber line plus a thickness distribution either side of this line, forming the upper and lower surfaces in two-dimensional (x,z) co-ordinates. The camber line, z_c, consists of one parabola from the leading edge to the point of maximum camber, and another parabola extending from this point to the trailing edge:

$$z_c = z_{c\max}\left(\frac{1}{(x_m)^2}\right)(2x_m\, x/c-(x/c)^2) \quad \text{for} \quad 0 \le x/c \le x_m\,, \quad \text{and}$$

$$z_c = z_{c\max}\left(\frac{1}{(1-x_m)^2}\right)(1-2x_m+2x_m\, x/c-(x/c)^2) \ \text{ for } \quad x_m \le x/c \le 1\,. \tag{2.1}$$

Here, $z_{c\,max}$ is the maximum camber and x_m is the position of maximum camber as a fraction of the chord, c. The thickness distribution, z_t, is a simple irrational polynomial function, the coefficients of which were found by fitting to a number of popular airfoils of the time:

$$z_t = 5z_{t\max}(0.2969\sqrt{x/c}-0.1260\,x/c-0.3537(x/c)^2+0.2843(x/c)^3-0.1015(x/c)^4)\,, \tag{2.2}$$

where $z_{t\max}$ is the airfoil maximum thickness. The airfoil co-ordinates are given by

$$\begin{aligned} x_U &= x - z_t\sin(\theta) & z_U &= z_c + z_t\cos(\theta) \\ x_L &= x + z_t\sin(\theta) & z_L &= z_c - z_t\cos(\theta) \end{aligned}\,, \tag{2.3}$$

where $\tan\theta = \mathrm{d}z_c/\mathrm{d}x$. Using the NACA 4-digit formulation, there are three design variables: the airfoil maximum camber (first digit, as a percentage of chord), position of maximum camber (second digit in tenths of chord) and maximum thickness (last two digits, as a percent). This simplicity means that the resulting shapes are inherently smooth and allow significant flexibility while always giving *sensible* shapes. However, the definition is unable to provide detailed control over the shape and surface curvature, and so is unsuitable for modern airfoil design. The capability of this parameterization could be improved by using more complex expressions for the camber and thickness; for example, the camber line could be determined by the position of the maximum camber point, and one or more other interpolated points.

2.1.2 CAD Based Techniques

The use of computer aided design (CAD) software is now commonplace in all engineering disciplines. Initially, this was the tool of a draughtsman in the design office, but it is being used increasingly in the conceptual and preliminary design phase as a tool for parametric design and as an input deck for computational analysis. Using a fully parametrically coupled CAD model of a component, a change can be made to the dimensions of a certain geometrical feature, and the changes to the whole component are updated automatically. A key tool within modern CAD software is the generation of complex parametric curves and surfaces, and in line with contemporary design needs these items can be controlled very precisely in terms of shape and curvature. Typically, the designer specifies a curve by requiring that it interpolates a number of points in space. Two popular curves of this type are now summarized: the polynomial spline and the B-spline.

Polynomial curves have often been used for data representation due to their ease of computation, and their behavior is simple and well understood. However, in order to pass through n data points a curve of order $(n-1)$ is required, and if n is high this can lead to numerical instability. An alternative and well behaved approach is to use a segmented curve, i.e., a spline, where a polynomial curve is placed between two adjacent data points to be interpolated. A single segment of the polynomial spline of order (n-1) describes the variation of z with x in the form:

$$z(x) = \sum_{i=1}^{n} a_i x^{i-1}, \quad x_1 \le x \le x_2 \tag{2.4}$$

The constants x_1 and x_2 are the extents of the spline segment in x, and a_i are coefficients to be determined by specifying boundary conditions for the segment. These boundary conditions arise by requiring adjacent

spline segments to have the same value of first and second derivatives at the point at which they join. The gradient is specified at the start point of the first segment and the end point of the last segment of the complete spline; this completes the curve. An advantage of this polynomial curve is the ability to readily control the gradient of the curve at the boundary. By specifying the co-ordinates of the segment joins, or *data points*, the coefficients of the polynomials are found and thus the curve is defined.

The polynomial spline is simple to implement, but as a result it is somewhat rudimentary. A more complex representation, rather more tailored to geometric shape design, is the B-spline curve. This is a generalizataion of the Bézier curve, which was originally developed for use in the automobile industry. The Bézier is a single-segment parametric curve, for a degree n curve it is defined by

$$\mathbf{C}(u) = \sum_{i=0}^{n} B_{i,n}(u)\mathbf{P}_i \qquad 0 \le u \le 1, \tag{2.5}$$

where $\mathbf{C}(u)$ is a vector-valued function of the independent variable u. This is similar in form to the polynomial segment in Eq. 2.4; in this case the basis functions $B_{i,n}$ are the nth degree Bernstein polynomials, and the coefficients $\mathbf{P}_i$ are called *control points*. The control points are such that they define the shape of a *control polygon*. Figure 2-1 shows an example of a third order (cubic) Bézier curve.

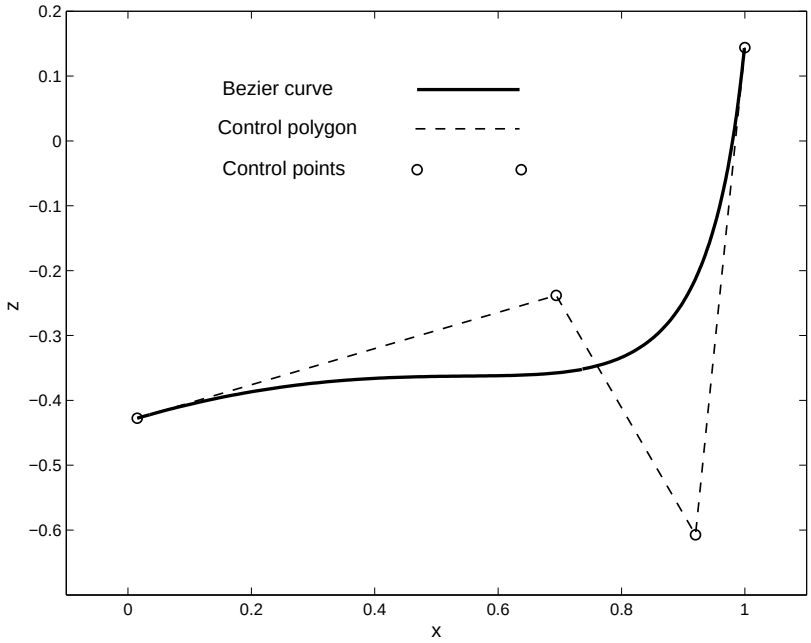


Figure 2-1 Example of a cubic Bézier curve.

The Bézier curve has useful properties, but, as with any polynomial, it consists of a single segment and as such it lacks detail and local control. Development of the Bézier representation resulted in the segmented B-spline curve. For a vector curve **C** in the variable u, a p-th order B-spline is defined by

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i \qquad a \le u \le b\,, \tag{2.6}$$

for a curve on the interval $[a,b]$. $\mathbf{P}_i$ are the $(n+1)$ control points, as in a Bézier curve, and $N_{i,p}(u)$ are the p-th degree B-spline basis functions. The *knot vector,* **U**, containing $(m+1)$ *knots*, is defined as

$$\mathbf{U} = \{\underbrace{a,\ldots,a}_{p+1}, u_{p+1},\ldots,u_{m-p-1}, \underbrace{b,\ldots,b}_{p+1}\}\,, \tag{2.7}$$

in which $m=n+p+1$. Now the i-th B-spline basis function is defined recursively as

$$N_{i,0}(u) = \begin{cases} 1 & \text{if} \quad u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)\,. \tag{2.8}$$

Eq. (2.8) is referred to as the *Cox-de Boor* recursion formula (detailed in de Boor [1968] and de Boor [1972]). When evaluating the curve $\mathbf{C}(u)$, for each u one finds the knot span in which u lies, computes the relevant basis functions, and multiplies these by the corresponding control points using Eq. 2.6.

What remains is to devise a method of calculating the control point locations that cause the B-spline curve to interpolate the data points in the given order, as this is the feature so useful in CAD software. If the $(n+1)$ data points to be interpolated are $\mathbf{D}_k$, these correspond to values of u, or *parameters*, t_k. Then for the data points, Eq. 2.6 becomes

$$\mathbf{D}_k = \mathbf{C}(t_k) = \sum_{i=0}^{n} N_{i,p}(t_k)\mathbf{P}_i \qquad \text{for} \qquad 0 \le k \le n\,. \tag{2.9}$$

Here, the basis functions $N_{i,p}(t_k)$ collectively form a $(n+1)\times(n+1)$ matrix, **N**. $\mathbf{D}_k$ and $\mathbf{P}_i$ are both vectors in s-dimensional space, and are rows of the $(n+1)\times s$ matrices **D** and **P**. The above relation can therefore be written as the linear system

$$\mathbf{D} = \mathbf{N} \cdot \mathbf{P} \quad , \tag{2.10}$$

which can be solved for the matrix **P**. Calculation of the corresponding curve then proceeds as normal using Eq. 2.6. An example of such an interpolating B-spline is shown in Figure 2-2.

B-spline curves are more complex to implement and require more information than simple polynomial splines, but they permit finer shape control. They also have the useful property that changing the position of control point $\mathbf{P}_i$ only affects the curve on the interval $[u_i, u_{i+p+1}]$, allowing highly local modifications. An excellent description and background to B-spline curves is given by Piegl and Tiller [1997].

From a geometric parameterization viewpoint, spline curves give excellent shape control and facilitate a large design space incorporating detailed and radical designs; this, however, gives a correspondingly large number of design variables. Also, the curves described here are inherently polynomials and therefore cannot represent some simple shapes such as circles and ellipses, and may lack the complex curvature control required for the detailed manipulation of shock waves on an airfoil surface, for example. For such cases, the more generalized non-uniform rational B-spline (NURBS) can be used; NURBS are described in detail by Piegl and Tiller [1997].

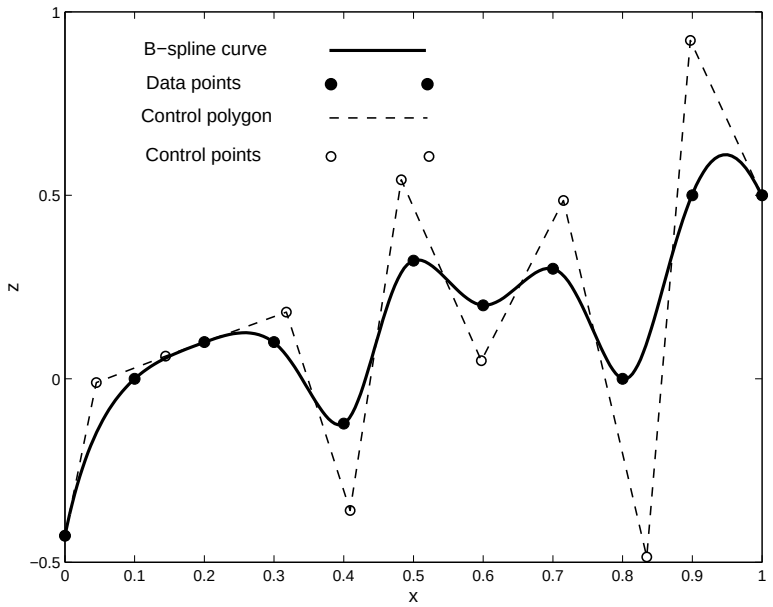


Figure 2-2 Example of an interpolating B-spline of degree three.

2.1.3 Analytical Methods

Outside of the domain of CAD software, various analytical functions and methods have been applied to the geometry manipulation problem. The use of linear combinations of functions is popular, including straight line, polynomial, ellipse and the more advanced Wagner functions (Xing and Damodaran [2005]), for example. Shape modifying functions are also commonly used, such as Hicks-Henne functions (Hicks and Henne [1978]). These are essentially local *bumps*, and since their curvature decays to zero at the extremities they can readily be patched onto an existing shape. A single Hicks-Henne bump takes the form

$$z = A\left[\sin(\pi x^{\ln 2/\ln x_p})\right]^w \qquad \text{for} \qquad 0 \le x \le 1, \tag{2.11}$$

where A is the bump height, x_p is the location of the peak and w varies its width. There are therefore three design variables associated with each bump. Typically a number of bumps are manipulated simultaneously during a design search procedure.

An alternative analytical approach for geometry parameterization is to define it as a solution to a set of partial differential equations (PDEs). This method was first used by Bloor and Wilson in order to provide smooth blending between surfaces, but has been applied to conceptual parameterization of complete aircraft configurations; see for example, the works by Bloor and Wilson [1995] and Smith et al. [1995]. By using an existing body, a number of boundary conditions are derived from the position and slopes at the boundary. This requires relatively few variables; solution of the PDEs gives the new surface. This description results in excellent smoothness and global control, but cannot perform the local variations in shape that are necessary for preliminary and detailed design tasks.

In order to incorporate more detailed shape changes into an analytical description, some methods utilize combinations of existing shapes. For airfoil design, existing shapes have been used in linear combinations, but a more efficient technique uses them to derive a series of orthogonal shape functions, as described in the work by Robinson and Keane [2001]. In this work, a family of nine NASA supercritical airfoils were analyzed and decomposed into six orthogonal functions, which, when appropriately combined, are able to re-capture the original airfoils. These orthogonal functions benefit from an intuitive description, similar to the NACA 4-digit series; i.e., the first function controls airfoil thickness, the second is camber and third is a form of twist, etc. This means that a model of varying complexity can be used, depending on the desired control and model dimensionality. However, while this

description is very concise, it is fundamentally derived from existing airfoil shapes and therefore gives a design space leading to evolution rather than innovation.

2.2 Computational Fluid Dynamics

Computational aerodynamics analysis of components can be performed using a wide variety of approaches of varying fidelity and capability. Each has advantages and disadvantages depending on the computational budget afforded and the level of accuracy and detailed modelling required. The capability of computational analysis has increased dramatically in recent years, but the latest RANS solvers require significant computational effort, and so are not suitable for conceptual design, for example. In this case the classical empirical and potential flow based analyses are more useful. The key types of flow analysis codes are summarized below.

2.2.1 Panel Methods

Panel methods are so-called because the geometry surface, be this in two or three dimensions, is discretized into a series or rectangular *panels*. The analysis proceeds by solving the linearized potential equation for inviscid, irrotational, incompressible flow for each of the geometry panels. To a limited extent, compressible flows can be modelled if a compressibility correction is employed, such as the famous Prandtl-Glauert model. Such approximations break down, however, in the transonic regime, meaning that panel methods cannot model flows for a free-stream Mach number greater than around 0.7 or for flows with even weak shocks. However, it is possible to estimate the wave drag of a wing by using prior knowledge of the wave drag of individual sections coupled with simple sweep theory (Petruzzelli and Keane [2001]); such a method can be used in conjunction with a panel CFD solution. The great advantage of panel methods is their rapid solution time. Additionally, the requirement for only a surface mesh (and sometimes a wake mesh) means that pre-processing for panel methods is often minimal.

2.2.2 Full Potential Methods

Full potential methods provide solutions for inviscid, irrotational, compressible flows. This full solution requires a computational mesh to be generated for the entire flow domain, but the inclusion of

compressibility terms means that solutions can be obtained for high Mach number flows. This means that the solver can to some extent model shocks, but because irrotational conditions are imposed the solution does not apply across strong shocks. The position of the shock can therefore be predicted incorrectly, leading to overestimations for wave drag and lift. However, a designer will typically tolerate only a weak shock in an airfoil design, for example; in this case the full potential equation can provide acceptable accuracy. Since a potential model is used, these solvers provide very rapid solution convergence. It is possible to incorporate quite accurate viscous models, but these do not model separated flows.

2.2.3 Euler Methods

The Euler equations represent inviscid, rotational, compressible flow. Since rotational flow is considered, a potential flow regime can no longer be assumed. The following succinct description of the governing equations is given by Jameson [1989]. For a three-dimensional flow, the variables x, y, z, u, v, w, ρ, E and p are the Cartesian co-ordinates, Cartesian velocity components, density, total energy and pressure, respectively. The Euler equation can be written in vector form as

$$\frac{\partial \mathbf{w}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{g}}{\partial y}+\frac{\partial \mathbf{h}}{\partial z}=0\,, \tag{2.12}$$

where

$$\mathbf{w}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix},\qquad \mathbf{f}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uH\end{bmatrix},\qquad \mathbf{g}=\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ \rho vH\end{bmatrix}\quad\text{and}\quad \mathbf{h}=\begin{bmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ \rho wH\end{bmatrix}. \tag{2.13}$$

In Eq. (2.13), the total enthalpy, H, is given by

$$H=E+\frac{p}{\rho}\,, \tag{2.14}$$

and p is obtained from the equation of state,

$$p=(\gamma-1)\rho\left(E-\frac{1}{2}(u^2+v^2)\right), \tag{2.15}$$

where γ is the ratio of specific heats. The entire flow domain is discretized, reducing the Euler calculations to the solution of a large system of coupled linear equations. These equations are then solved iteratively, using a variety of different schemes. Since an iterative scheme is used on such a large set of equations, the solution is computationally more expensive than for the potential equation solvers. The advantage of this solver over a full potential code is the ability to more accurately predict the strength and position of shocks, since rotational flows are calculated. Relative to potential methods, reasonably accurate drag predictions can be obtained if additional far-field momentum thickness calculations, for example, are implemented.

2.2.4 Reynolds-Averaged Navier-Stokes (RANS) Methods

RANS solvers attempt to solve the complete viscous, rotational, compressible equations of fluid flow, i.e., the Navier-Stokes equations. The Euler equation, Eq. (2.12), is extended to include the viscous stresses, τ, forming the matrix system

$$\frac{\partial \mathbf{w}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{g}}{\partial y}+\frac{\partial \mathbf{h}}{\partial z}=\frac{\partial \mathbf{R}}{\partial x}+\frac{\partial \mathbf{S}}{\partial y}+\frac{\partial \mathbf{T}}{\partial z}, \tag{2.16}$$

where

$$\mathbf{R}=\begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx}+v\tau_{xy}+w\tau_{xz} \end{bmatrix}, \quad \mathbf{S}=\begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx}+v\tau_{yy}+w\tau_{yz} \end{bmatrix} \quad \text{and} \quad \mathbf{T}=\begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx}+v\tau_{zy}+w\tau_{zz} \end{bmatrix}. \tag{2.17}$$

In this form, the Navier-Stokes equations describe the general turbulent flow of a fluid in thermodynamic equilibrium. However, Jameson [1996] reports that, to resolve all turbulence scales in three dimensions, a computational mesh size of the order $Re^{9/4}$ is required, which is obviously too large for a design situation. Thus, an approximation is used to provide time-averaging of the six rapidly fluctuating viscous stresses, τ. This results in the Reynolds equations (hence the name Reynolds-averaged), which are non-linear in nature and require some form of turbulence model for closure. As with the Euler equations, the flow domain is discretized and the RANS equations are reduced to a large linear system of equations which are solved iteratively.

The advantage of modelling viscous effects is the ability to predict and simulate separated flows, and potentially, obtain more accurate drag estimates. The principal disadvantage with the RANS method is the requirement for a very fine spacing of mesh cells in the direction normal to the body surface in order to resolve the extreme gradients encountered in the boundary layer flow. The resulting mesh generally contains a very large number of cells, often of the order of millions for a large three-dimensional study. The use of turbulence models significantly reduces the required number of cells, but the assumptions made inevitably limit the behaviour and accuracy of the turbulent flow. However, for the purposes of design, simulating fluid flows with extreme accuracy is less important than reproducing the trends in performance for a given change in the design parameters. Therefore, the loss of a certain degree of accuracy is accepted, with the knowledge that the associated speed-up considerably reduces the cost of performing a design exercise.

Two methods representing the state-of-the-art in CFD and turbulence research are LES and DNS. Large eddy simulations (LES) solve the large turbulent eddies directly, and model the small eddies. Direct numerical simulation (DNS) aims to directly solve all scales of turbulence. At present, neither of these approaches is employed for design applications due to the prohibitive computing power required. Therefore, in this thesis the RANS simulation is the most advanced solution approach used as part of the design search applications.

2.3 Optimization Methods

Presently, a vast array of design search and optimization (DSO) techniques are available to the designer, and these are suited to many different design problems. In the current study, the range of methods is limited to those dealing with continuous and numeric variables, with non-linear and deterministic objective functions (a given set of inputs always gives the same output). Four categories of optimization are considered below, all of which can be portrayed at their most basic level by the flowchart in Figure 2-3. The component geometry is discretized into a number of continuous design variables, and for each set of variables the shape is analyzed using CFD giving a measure of performance. The job of the optimization algorithm is to manipulate the design variables in an efficient search for improved designs and performance.

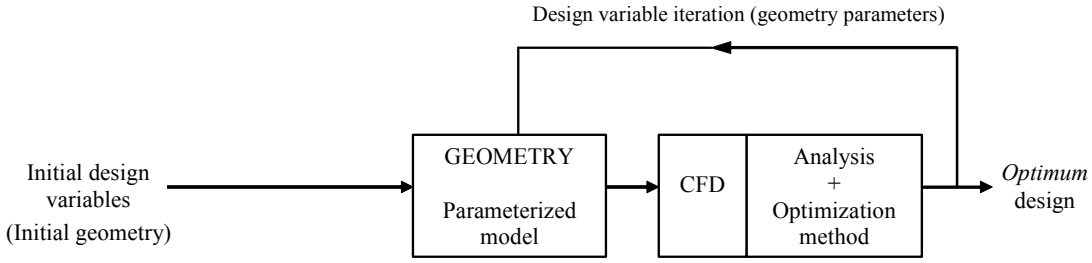


Figure 2-3 Flowchart illustrating the design search and optimization process.

2.3.1 Gradient Based Methods

The gradient-descent methods, also known as *hill-climbers*, were the first mathematical algorithms to be used for engineering design optimization problems. Fundamentally, these involve the calculations of local gradients of the objective function in order to determine the direction of highest potential improvement; a line search tool is then used to find the optimum in the chosen search direction. Often, the gradients are calculated using finite differencing, but they can also be obtained directly from the analysis code. So-called *adjoint* CFD methods can provide a faster, more accurate calculation of the gradients. Development of such methods, led to a large extent by Jameson (see Jameson [1988] or Jameson [1999]), has advanced to the stage where the sensitivities of the solution to orders of 1000's of variables can be calculated at a cost of only a few objective function calculations. However, optimization methods are in less demand compared to direct CFD analysis software. Further, if the optimization is to be used with the types of design (CAD) parameterization discussed above, the adjoint must be integrated with the geometry and meshing tools. Thus, adjoint methods are generally not available commercially. In general, gradient-based

methods are simple to implement but the determination of the gradient for multi-dimensional design problems can be very costly, particularly if the objective function involves the use of CFD. In addition, the presence of noise in the objective function can result in inaccurate calculations for the search direction, delaying convergence of the algorithm.

2.3.2 Gradient Free and Global Optimization

There are optimization algorithms which work without the need to calculate the gradients. These can be grouped into two classes: pattern searches and evolutionary algorithms (Keane and Nair [2005]). Pattern searches operate by sampling points in the region of the current best design. A trial step is made in some direction from the current point. If the new point yields better performance then a further step is made in this direction, but if it is worse then an alternative direction is trialled. If steps in all directions yield poorer performance, the step size is modified in some way. The search is often deemed to be converged when the step size falls below a threshold. Since the step can take place in any direction, pattern searches are very often convergent, but only to the nearest optimum; they are not suitable for searching multi-modal landscapes in a global sense, unless the process is restarted in multiple locations.

Evolutionary algorithms are a class of the more general field of stochastic optimization, in which the selection of designs to be tested has a random element. This is in contrast to deterministic methods, in which a given set of initial conditions will always cause the optimizer to follow the same path. The random design perturbations are termed pseudo-random, since they are generated by a numerical algorithm and not a true normal distribution. However, this has the advantage that if a search is run with the state of the random number algorithm held constant, the optimization will proceed along the same path, i.e., the search is repeatable. Introducing a pseudo-random element means that evolutionary methods are able to avoid getting stuck on local optima, and instead can give a more global search of the design space. The genetic algorithm (GA) is a popular member of the evolutionary methods, and attempts to imitate the process of Darwinian evolution observed in nature. Each vector of design variables represents genetic material, and each design is a member of the *population*. The process starts by evaluating the objective function for an initial population (the first *generation*), and these are subjected to rules similar to those of natural selection; in particular, *fitness* and *crossover*. The higher achieving the individual, the more likely it is to contribute genetically to the next generation, i.e., survival of the fittest. Individuals that contribute to the next generation are paired up, and their offspring have attributes common to both parents, i.e., crossover. These rules ensure that the optimizer carries forward only the promising designs whilst also exploring the search space. Many GA's also feature *mutation*, whereby small random changes

are introduced into new individuals, in theory allowing any point in the design space to be searched. In summary, evolutionary optimizers give a thorough and global search of the design space, but require a relatively large number of objective calculations in an attempt to converge onto the optimum.

2.3.3 The Trust-Region Approach

The optimization algorithms listed above all require a large number of objective function evaluations, which makes a design search prohibitively expensive when the objective function involves high-fidelity CFD analysis. To improve the computational efficiency, it is possible to replace the objective function with a computationally cheap surrogate model when searching for a point with potentially improved performance. The model is trained to fit the collected response data; in some cases, particularly for simple function models, such a surrogate is only a valid approximation in a small region of the design space. The *trust-region* is an area of the design space surrounding the current best design point, over which the surrogate model is considered to be a close approximation to the true objective function. The surrogate is used to predict the location of the optimum within the reduced area, and the true objective function is calculated for this point. The trust-region algorithm adapts the size of this searchable space after each evaluation, theoretically guaranteeing convergence to a local optimum. Typically, the surrogate function is a second-order polynomial. While this approach is computationally efficient compared to a hill-climber or GA, for example, it is not suitable for global optimization.

2.3.4 Response Surface Model Methods

The trust-region method considers only a small region of the design space, and uses a surrogate model which approximates only the measured responses within this area. In contrast, the term *response surface model* (RSM) is used here to refer to a general surrogate which can be used to represent the entire design space and which can in general be multi-modal. The application of this RSM approach to design optimization problems is comparatively new. The optimization is performed in two stages. In the first stage, the objective function is evaluated at a series of design points distributed in an attempt to efficiently populate the design space; this is best accomplished using a formal design of experiments (DoE) array, examples of which are given by Grove and Davis [1992]. The second stage involves constructing a curve fit, called the *response surface*, through the collected response data. The choice of response surface model often depends on the nature of the objective function; a taxonomy of RSM's is given by Jones [2001]. The RSM method is therefore not an optimizer in its own right, but by mimicking the real objective

function surface, calls to the full function can be replaced by calls to the RSM, which is particularly advantageous when the objective function calculation requires the use of an expensive analysis code. The RSM is searched for areas of promising designs using an optimization method. Since many types of RSM are able to accurately model multiple local optima, a global optimization method, such as a GA, can perform a global search over the design space. Most commonly, the full objective function is then calculated for the optimum point predicted by the RSM, and the surface is updated. Successive searching and updating of the surface continues until convergence is reached. The RSM method is a popular choice for global optimization using expensive functions since it can be used to predict promising areas of the design space with relatively few objective function calls, compared to other design search methods (Jones et al. [1998]). Further savings in overall time can be made by constructing the RSM with simultaneous calls to the objective function, something that is not always possible in other approaches to design optimization. Not only does this speed up the optimization procedure by allowing the update points to be evaluated simultaneously, it also reduces the chances of stalling on a local minimum in the surface (Sóbester et al. [2004]).

2.4 Approaches for Aerodynamic Design Optimization

2.4.1 Direct Design Search

As has been previously discussed, the computational approach to aerodynamic design is a process of decision making, performed either by the designer or by an optimization algorithm, that changes the product definition based on the results of CFD analysis. Traditionally, these decisions involve the manipulation of a component's geometry in order to maximize a certain performance metric. The fundamental design search and optimization process is encapsulated in Figure 2-3, above. The component geometry is parameterized, and optimization of a performance scalar, calculated using CFD, is automated using an optimization algorithm. Such design search methods are here termed *direct*, because the desired direction of calculation is always from geometry to flow field variables; this is also referred to as the *analysis* approach in classical inverse design theory.

2.4.2 Inverse Design

The alternative to a direct design search is to specify a set of flow features, and search for a geometry which produces flow features matching those specified. This is the so-called *inverse design* method, referred to as simply the *design* approach in classical inverse design theory since the result of calculations is geometry. The objectives of the inverse and direct design methods are quite different. Inverse methods require that the flow features of the intended design are specified *a priori*, and traditionally this specification is the task of an experienced aerodynamic designer. Knowledge of the required flow features allows inverse methods to produce the corresponding design with very little computational expense, but the resulting design is only optimal if the specified flow features are also optimal for a given set of flow conditions. In contrast, the objective of a direct method is to systematically search for this optimal design. For the inverse method to be effective, the flow features specified by the designer should also be realistic, i.e., it must be possible for a shape to be designed which realizes the target flow features. The background and current use of inverse design within the aerospace community is summarized in what follows.

Inverse methods have been used extensively in the context of airfoil and wing design, where the target flow feature is the pressure or velocity distribution in the chordwise direction. The well established analytical solutions for airfoil theory have allowed inverse methods to be exploited to great effect. The literature on this subject is vast, and the reader is directed to the survey-type papers by Sobieczky [1990], Labrujere and Slooff [1993], Drela [1989], Volpe [1989] and Dulikravich [1990], and also to a book by Elizarov et al. [1997]. The formulation of a well-posed inverse problem for airfoil design is not at all trivial, as has been demonstrated in the pioneering work by Lighthill [1945]. Since that time, the following three variations of inverse method have been used for airfoil and wing design.

- Coupled solution methods
- De-coupled solution methods
- The optimization approach

Coupled solution methods are classed as *non-iterative*, but this is something of a paradox. The geometry is calculated directly via the solution of a boundary value problem, without the need for an iterative update scheme on this shape. However, since the inverse boundary value problem is non-linear in nature, its solution requires the use of an iterative process, such as Newton iteration (Giles and Drela [1987]). In this approach, the flow variables and the unknown geometric variables are solved as one set of unknowns, hence the name *coupled*.

For de-coupled (iterative) solution methods, the flow variables and geometric parameters are solved in a de-coupled fashion. The great advantage of this approach is that it allows the use of existing *direct* flow analysis codes, something which is not possible with coupled methods. The iterative methods start with an initial guess of the geometry, and at each subsequent iteration the geometry is derived via the solution to a boundary value problem. Nearly all of these methods aim to solve either the Dirichlet boundary value problem, or a Neumann type problem. In the Dirichlet problem, the boundary condition is the tangential velocity, derived from the prescribed surface pressure distribution; the solution then proceeds by updating the geometry aiming to achieve zero transpiration (normal) velocity. Neumann or *residual-correction* methods proceed by providing a pressure distribution for each iteration of the geometry, which is compared with the target pressure distribution. The difference between these profiles is the residual, and this must be minimized by the inverse process. The main challenge is to relate the residual at each point on the surface to the required changes in geometry; this is typically achieved by the use of linearized potential theory (Labrujere [1994]). For Neumann type methods, existing flow analysis codes can be utilized as a *black-box*, allowing alternative solvers to be substituted with minimal modification to the inverse code. In developing a residual-correction based method, a compromise must be found between the computational effort required in the correction calculations, and the number of iterations needed to reach a converged solution. This iterative process is illustrated as a flowchart in Figure 2-4.

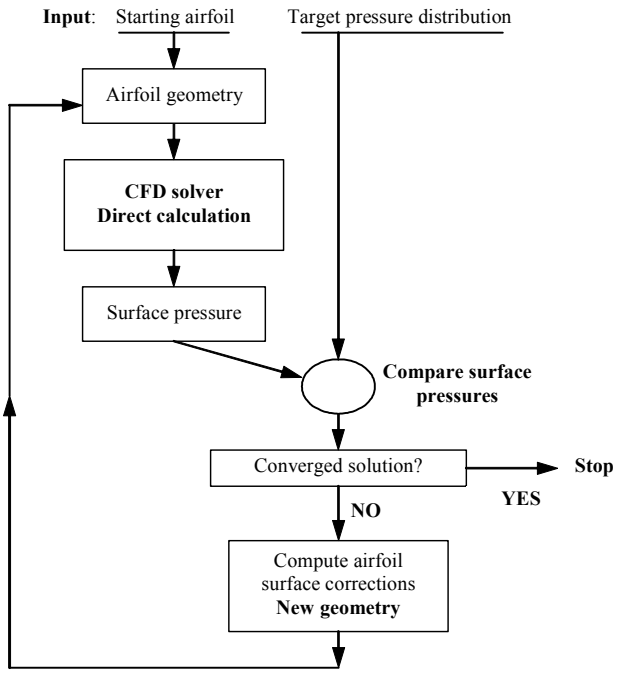


Figure 2-4 Flowchart illustrating a residual-correction type inverse design process.

The optimization approach to inverse design involves the perturbation of the shape using a parametric description of geometry coupled with a numerical optimization method. For example, LeGresley and Alonso [2000] use proper orthogonal decomposition to calculate gradients for the minimization of the sum-of-squares error between the target and computed pressure profiles. This method does not make use of analytical inverse theory, and since the surface pressure residual is converted into a scalar objective function for optimization, a large computational effort is required for sufficient convergence. However, the method is relatively simple to implement in practice, and can provide a good starting point for an inverse design study.

From a practical point of view, inverse design methods have proven to be popular because once the target flow feature is specified, the use of linearized potential or other analytical theory allows the required geometry to be obtained with very few CFD evaluations. For example, the residual-correction method of Takanashi [1985] uses an integral formulation of the full potential equations solving the Neumann problem, and is able to provide a converged solution in around 10 iterations. Of course, the inverse process always requires a construction of the geometry in order to perform CFD analysis. However, in contrast to optimization, the number of *variables* perturbed by the inverse process is not limited. The knowledge of the pressure (or velocity) at each surface point allows the geometry to be modified at each point. Thus, the geometry is typically described using a large set of co-ordinate data.

While its main application has been for airfoil and wing geometries, inverse design has been used for the design of various aerodynamic components, such as turbine blades (for example, by Goto and Zangeneh [2002]), and for various target flow features. In the current work, the term *inverse design* is not used to describe the airfoil design problem alone; rather it is used to refer to a general computational process which determines a geometry corresponding to the prescribed target flow feature, which may or may not be the surface pressure.

As stated above, inverse design has typically been used to determine the *optimized* design once the final target flow feature has been established. Traditionally, the optimum target pressure distribution is specified by an experienced aerodynamicist, a task which is fraught with difficulty. However, such a specification can also be the result of an optimization on these flow features, a concept first described by van Egmond [1990]. A similar concept is used in the design approach proposed in this thesis. Using an optimization algorithm to manipulate the target flow feature, development using high-fidelity CFD analysis is accelerated without requiring the specification of the target flow feature *ab initio*, which is the hallmark of classical inverse design and one of its principal drawbacks. In this work, the proposed method is used in both 2-D and 3-D applications, starting with the design of 2-D airfoils.

Chapter 3. Optimization Using Flow Feature Parameterization: Concept and Implementation

3.1 Introduction

The preceding chapter has given a summary of some of the current practices associated with aerospace design using CFD. The capability of computational analysis has increased dramatically in recent years, but while computational efficiency has improved there is also a desire to perform ever more accurate and detailed flow simulations. This has led to sustained research into more efficient optimization methods for use in aerodynamic design. It has been noted that, typically, designers choose to manipulate the geometry of a component and monitor the effect this has on performance, while it is also prudent to check the characteristics of relevant flow features. For example, in designing an airfoil one might manipulate the shape in an attempt to minimize drag, while also checking the position and strength of any shock waves in order to avoid flow separation close to the design point. Unfortunately, when performing automatic optimization in such a manner, an inherent limitation is the requirement for a large number of design variables in order to define, in sufficient detail, the geometry of the component being studied. The resulting process can be very expensive computationally, particularly when using high-fidelity CFD. As well as designing a component via the relationship between its geometry and its overall performance, it is also possible to design by matching flow features with a set specified by the designer, i.e., inverse design. For example, when designing an airfoil using inverse design, one starts with a target pressure distribution and obtains the corresponding airfoil shape. The inverse process is far more rapid than a direct search, but does not aim to optimize the performance; an *optimum* design is only obtained if the specified target flow features are also optimal for a given set of boundary conditions.

Thus far, a need has been identified for more efficient design optimization strategies involving high-fidelity CFD simulations, and that a reduction in the number of design variables as a result of improved parameterization methods can contribute to this. As introduced in Chapter 1, this thesis proposes that a concise and efficient parameterization can be achieved if the design variables describe key flow features

rather than the geometry directly. The proposed parameterization technique consists of two elements: a parametric model of flow features and an inverse design method to recover a corresponding geometry. Inverse design is therefore used as a tool and is called upon repeatedly within a more complex *direct* design optimization search. In the present chapter, the concept of the proposed parameterization method is detailed. The application of this method to the design of two-dimensional airfoil sections for reduced drag, and the associated computational setup, is described.

3.2 Embedded Multi-Fidelity Inverse Design (EMFID): The Concept

The design search method introduced here uses a parameterization of flow features coupled with inverse design. The motivation for investigating this strategy is a desire to reduce the number of design variables used in optimization while maintaining the ability to produce detailed and varied geometries. The flow features associated with aerodynamic flows can be rather complex, but in certain situations they may be simpler to represent, parametrically, than the corresponding geometry. The success of the proposed strategy is based on the prediction that a simple and low-dimensional parameterization of a key flow feature may result in a wider range of geometries, after inverse design, than a geometrical parameterization with the same number of dimensions. This supposition arises from two observations in aerodynamic design. First, it is known that a change to a flow characteristic in one geometrical region can have a global effect on the entire corresponding design. As an example of this, Figure 3-1 shows two airfoil surface pressure coefficient (C_p) distributions, for which the lower surface pressure differs aft of the 60% chord point, and which are otherwise identical. When each of these profiles is used as a target for inverse design, the airfoil shapes in the lower half of Figure 3-1 are obtained. It can be seen that the two airfoils are quite different, and the two upper surfaces differ significantly despite the pressure on these surfaces being identical. An exception to this regime is seen in transonic aerodynamics, since downstream flow perturbations are not felt upstream of a shock. However, in the majority of situations it is possible to influence a geometry globally using rather more subtle changes to the flow features.

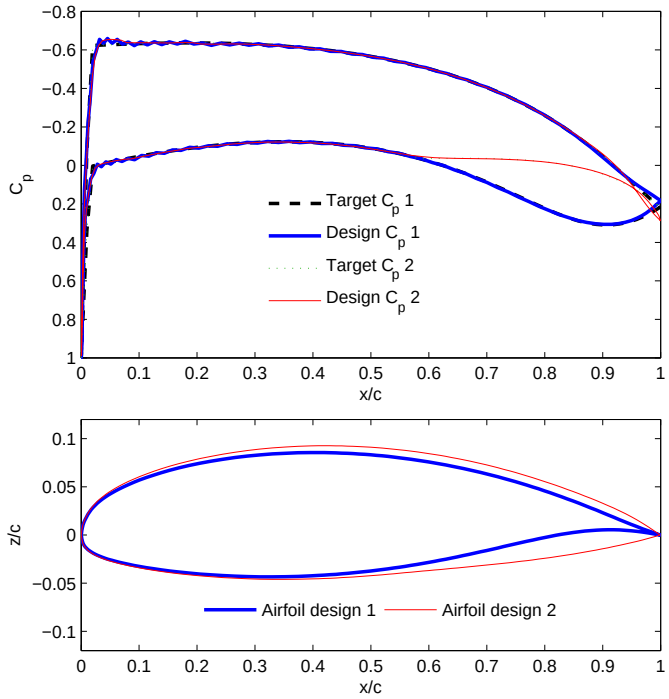


Figure 3-1 A localized change in surface pressure has a global effect on the corresponding airfoil shape.

The second aerodynamic observation supporting the proposed parameterization is that the flow features, such as shocks or vortex flows, can have a direct and strong coupling with body forces and, therefore, performance measures such as drag. For example, it is known that wave drag for an airfoil is a function of the strength and position of the shock on the upper surface, with a stronger shock resulting in increased drag. The shock wave details have an intuitive effect on drag, and hence one would expect them to be effective when used as design variables. This is in contrast to geometry variables, which typically have a more complex relationship with drag.

A parameterization of flow features can potentially allow detailed geometrical changes whilst giving a reduction in the number of design variables. This lower dimensionality means that the number of calls to the expensive CFD solver required to populate the design space can be reduced considerably compared to an optimization method acting directly on the geometry. However, for each call to the objective function an inverse design step must be performed, requiring additional computational expense. The effectiveness of this method relies on the saving made in reducing the number of high-fidelity CFD evaluations being greater than the relative cost of the inverse design steps. Increasingly, multi-fidelity approaches to design optimization are being used both to improve the reliability of the analysis and reduce the computational expense of a design search (Keane and Nair [2005]). Multiple levels of CFD model complexity have been

used simultaneously in an automated way in previous research, for example, by Keane [2003] and Alexandrov et al. [2000]. While high-fidelity aerodynamic optimization is desired and hence an expensive solver is used to evaluate the design metric, the key to the effectiveness of the proposed strategy is the use of a lower-fidelity CFD solver for the inverse design steps; thus we have a multi-fidelity search procedure.

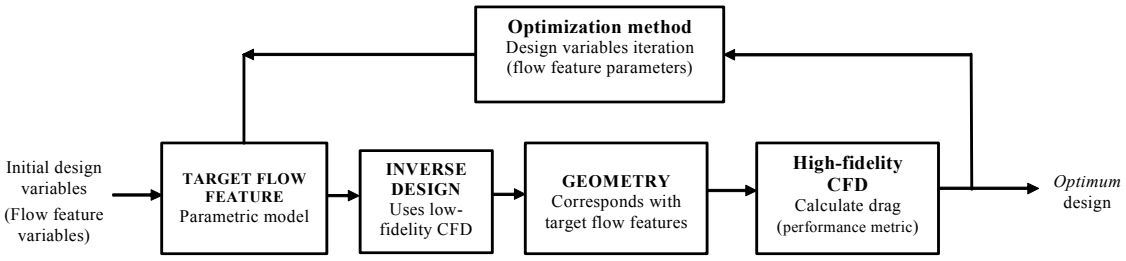


Figure 3-2 Flowchart illustrating the proposed EMFID design search process.

The proposed design search and optimization strategy is illustrated as a flowchart in Figure 3-2. A key flow feature is discretized into a number of design variables. Each iteration of these variables produces a target flow feature, and a geometry which achieves this target is determined using an inverse design method, making use of a low-fidelity CFD code. Following this, a performance metric is calculated for this geometry via high-fidelity CFD analysis. Throughout this thesis, the performance metric used as the objective function in all case studies is total drag (or coefficient of drag) at a fixed level of lift (or coefficient of lift). An optimization algorithm is used to minimize the drag, in an identical manner to the procedure shown in Figure 2-3, but in this case manipulating the design variables defining the target flow feature rather than the geometry itself. This design search procedure is referred to as embedded multi-fidelity inverse design (EMFID) from this point forward.

In order to effectively evaluate the design performance of an optimization process, comparisons must be made with an existing and established method. In the work described in this thesis, the operation of the EMFID method is compared with a conventional approach using a geometry-based parameterization, and which is considered to be a current benchmark in design optimization. This benchmark optimization process can be described by the flowchart shown in Figure 2-3. The method focuses purely on the geometry of a component and its resulting aerodynamic forces, namely drag. In the case studies given in later chapters, the EMFID and benchmark methods are run with equal computational expense, simulating the fixed computational budget that an aircraft design team may be afforded in an industrial situation.

3.3 Related Work

Fundamentally, the proposed method is a form of *reduced order modelling*, a concept which itself has been applied extensively. The purpose of reduced order modelling is to construct a lower-dimensional model which comprises fewer unknowns than the original high-dimensional model. Reduced basis methods decompose the field variable set, which could make up a surface pressure distribution, into a linear combination of known basis vectors and unknown coefficients (Keane and Nair [2005]). The basis vectors are typically from the relatively simple Lagrange subspace, and the coefficients are computed using a numerical scheme such that they approximately satisfy the discrete form of the governing equations characterising the physical system. The resulting system allows the field variables to be approximated using the known basis vectors and for a given selection of the coefficients. However, the basis vectors can be linearly dependant, reducing the efficiency of the approximation. Hence, principal component analysis, or proper orthogonal decomposition (POD), can be used instead to compute an optimal set of bases (principal modes) and coefficients that most efficiently reconstructs the original field variable data. Such analyses have been referred to as *reduced-reduced* basis methods (Burkardt et al. [2003]). POD has been used to great effect in aerospace design to provide approximate solutions to the Euler and Navier-Stokes equations and to perform inverse design, for example. The reader is referred to the articles by LeGresley and Alonso [2000], LeGresley and Alonso [2003], Bui-Thanh et al. [2004] and Burkardt et al. [2003]. The design approach investigated in this thesis relates to the subject of reduced order modelling in that it attempts to transform the geometry *field* variables into a more compact set of variables, by mapping them into the flow feature domain. Similar to POD, the proposed method uses variables which are dominant, although these are not necessarily orthogonal or independent.

The proposed optimization approach makes use of both low- and high-fidelity CFD analyses, and in this sense it is a multi-fidelity design search. Notable past research into multi-fidelity optimization is that of Robinson and Keane [1999], Alexandrov et al. [2000] and Keane [2003]. These authors exploit a cheap analysis by using it to approximate the same objective as the expensive solver and thus reduce the overall computational cost. However, the method proposed here does not use the low-fidelity CFD to calculate the design objective (drag); instead it is used purely for the purposes of inverse design.

In this work, the flow features are parameterized, and each call to the objective function requires the use of inverse design. As part of inverse design, the flow features are extracted from the results of low-fidelity CFD analyses and compared with the target flow features. Flow feature extraction ranges in complexity, from the simple task of computing surface pressure in potential flow, to determining the location of topological elements such as vortices and separation surfaces. The use of automated algorithms for these

complex tasks is well documented; examples include research by Jeong and Hussain [1995], Kenwright et al. [1999] and Haimes and Kenwright [1999]. Such flow feature extraction is implemented in subsequent chapters, although the techniques are not developed.

In previous research using inverse design, the airfoil pressure distribution is commonly used as the target flow feature, in order to optimize the airfoil shape or the sections of a three-dimensional wing; some relevant works are now listed. Parameterization of the airfoil pressure distribution was first attempted by van Egmond [1990], who formulated a set of aerodynamic shape functions which actually used a larger number of design variables than the geometrical models of the time; this was justified because the inverse design process employed required evaluation of the flow in the boundary layer only, meaning that a large number of evaluations could be afforded in manipulating the target pressure profile. Obayashi and Takanashi [1996] used aerodynamic design relationships and constraints to relate the surface pressure distribution to certain airfoil performance parameters. A genetic algorithm (GA) was used to optimize the pressure profile for minimum drag, after which an inverse design code was employed to recover the corresponding airfoil shape. Jameson [1999] optimized an airfoil shape using Euler based CFD analyses, and used the pressure distribution of the resulting airfoil as a target for inverse design using a RANS solver, since this gives more accurate viscous drag predictions. Ahn et al. [2001] used lift and drag relations together with CFD analyses on a series of airfoil geometries to build a response surface model (RSM), able to relate the surface pressure distribution to predictions for airfoil lift and drag. The target pressure distribution could then be optimized by using a GA search over the RSM, and an inverse design method (detailed by Kim and Rho [1998]) was used to determine the resulting airfoil shape. For each spanwise station of their three-dimensional wing, the number of design variables for the optimization of the pressure distribution was 15. Ahn et al. recognize that replacing airfoil section shape parameters with section pressure distributions gives a saving in computational cost. They report the computational cost of their work to be one-sixteenth of the cost of direct design methods, although no detailed comparisons are made.

Note that in the references given above, a single inverse design operation is performed once the target pressure distribution has been optimized by some means. The EMFID method, however, makes repeated use of the inverse step, and for every resulting geometry the drag is calculated using high-fidelity CFD. Hence, the method does not rely on empirical relationships or approximations in calculating the optimum target pressure distribution and airfoil shape for minimum drag. In the sections that follow, the application of EMFID to the design of two-dimensional airfoils is described.

3.4 Application of EMFID for 2-D Airfoil Design

The subject of airfoil section design is well known and understood, having been applied routinely in the aircraft industry for many years. As a preliminary assessment, the EMFID method is applied to the design of two-dimensional airfoil shapes, as this simplifies the comparisons between the method under investigation and existing technology. The distribution of pressure over the surface is chosen as the flow feature to be parameterized. As mentioned above, small perturbations in pressure distribution can often require large variations in the entire geometry, for a given set of flow conditions. Also, changes in surface pressure can be related directly to changes in geometry using simple aerodynamic approximations, which can be used to accelerate the inverse design process as described in Section 2.4.2. Initially, a wholly subsonic flow regime is used for analysis, allowing a more straightforward parameterization of the pressure distribution in EMFID. Subsequently, EMFID is applied to transonic airfoil design, and thus a parameterization is developed which incorporates a shock.

In using a simple design problem, the intention is to demonstrate that the EMFID approach can be an effective method in aerodynamic design. Since aircraft designers do not yet use high-fidelity RANS simulations universally for preliminary design, the method demonstrated here is unlikely to replace the existing airfoil design methods in the short term. Indeed, design packages such as XFOIL[1] (Drela [1989]) are now commonplace and facilitate very rapid airfoil design and analysis. However, once the proposed method has been verified for a simple problem, it will then be applied to a more complex three-dimensional problem in which the advantages of the approach are potentially more attractive.

As stated above, the EMFID method must be compared against a more conventional benchmark method. In both cases, the design objective is to minimize the drag coefficient (c_d) of the airfoil for a fixed level of lift (c_l), while allowing the angle of attack to float. Such a criteria is typically used in aircraft design situations, in which a fixed payload is usually specified. The setup and configuration of the two design methods is now described in full. Both methods employ the same high-fidelity flow solver, while EMFID additionally calls upon a low-fidelity code; the setup of the CFD analyses for this 2-D application is described next.

[1] Information can be found on the world wide web at http://raphael.mit.edu/xfoil/ (cited April 2006).

3.5 Airfoil Analysis: CFD Solver Setup

In the current work, the high-fidelity solver used is a RANS analysis using FLUENT®[1] and the low-fidelity code, used by the EMFID method, is the full potential flow solver VGK (ESDU [1996]). Calibration of the CFD solvers involves *verification* with respect to the dependence of the results on mesh size, and *validation* against a set of experimental results for a standard airfoil. For the newly proposed design method in particular, it is important that the airfoil pressure distribution must be reproduced accurately by both flow solvers. As already stated, in the current work design studies are carried out using both subsonic and transonic flow regimes; each condition requires a different computational mesh setup. In the following sub-sections, the general setup of the FLUENT and VGK flow solvers is explained, and following this the validation of the CFD solvers for both flow conditions is described.

3.5.1 FLUENT

A FLUENT RANS analysis is used to calculate the airfoil performance metric, i.e., total drag, in both the benchmark and EMFID design methods. The model is set up with the aim of minimizing the computational effort required for the analysis, giving robust convergence, while providing accurate results in close agreement with the experimental data. A commercial meshing tool, GRIDGEN®, is used to mesh the flow domain to be solved by FLUENT. The airfoil geometry is imported into GRIDGEN in the form of a data file containing a matrix of 206 (x,z) co-ordinate pairs, and the airfoil is normalized to a chord of one metre. The airfoil has a *sharp*, i.e., zero thickness, trailing edge. An *O*-mesh topology is employed (Steinbrenner and Anderson [1989]), where the mesh cell size is increased with distance from the airfoil surface. The details of the mesh size are different for the subsonic and transonic cases, and are given below.

In the FLUENT CFD model, the equations of momentum and continuity are solved in a coupled manner and the Spalart-Allmaras turbulence model (Spalart and Allmaras [1992]) is employed, as this is known to be a relatively accurate method for external flow over an airfoil, and provides robust convergence in the sense that a converged solution can be obtained for a large range of input geometries. This latter feature is desirable in conceptual design search processes when a large number of airfoils are analysed, some of which may be unusual or even nonsensical designs. The cell size adjacent to the wall is arranged such that the non-dimensional distance, y^+, over the airfoil surface lies in the range suitable for a log-law wall function representation of the boundary layer, i.e., between 30 and 60.

[1] Information can be found on the world wide web at www.fluent.com (cited April 2006).

In the design studies to follow, for each airfoil design analyzed using FLUENT, the desired performance metric is drag at a fixed value of lift. This is determined by running the analysis at three values of the angle of attack until the desired lift is achieved. The analysis is run to convergence at two initial angles; following this, the correct angle is calculated by assuming a linear lift-angle curve and the analysis is run at this angle, providing a converged solution at the desired value of lift to within ±1%.

Note that, although throughout this thesis the FLUENT analysis is termed *high-fidelity*, this solver is not necessarily very accurate compared to the latest state-of-the-art techniques such as LES and DNS solvers. The purpose of this work is to develop a process which works efficiently with expensive CFD solvers; however, the expense of this high-fidelity analysis must be limited since several hundred analyses may be required for a complete design search. The FLUENT RANS analysis is used since it is typical of the type of solver currently used for aerospace design, and it is available to run on the Microsoft compute cluster.

3.5.2 VGK

The method presented in this thesis requires a computationally inexpensive CFD solver to compute the airfoil pressure distributions during the inverse design step. The low-fidelity software used here is VGK, written by DRA Farnborough and distributed by the Engineering Sciences Data Unit (ESDU). VGK is a two-dimensional viscous coupled finite difference code which solves the full potential equations, written specifically for the analysis of airfoils. The airfoil geometry is input as a matrix of co-ordinates, following this a computational mesh is built in the flow domain using a series of radial and circumferential grid lines. The full potential equations are solved iteratively over the grid using a finite difference approach.

The full potential equations are exact for shock-free inviscid flows, but are only approximate for flows with significant shock strength, since such flows are rotational. However, the finite difference scheme utilized by the VGK code has undergone extensive development in an attempt to approximate the correct Rankine-Hugoniot relations, and the selection of suitable input parameters results in reasonable surface pressure predictions across weak shocks. The use of a potential formulation also means that VGK cannot produce results for flows where the boundary layer has separated from the surface. However, the viscous flow element of the code has been calibrated against experimental data which gives it the ability to estimate the location of the separation boundary and which also results in more accurate surface pressure and drag estimations.

The VGK CFD model is set up as a viscous solve and, in the first instance, using the same Reynolds number and flow speed as used in running FLUENT. Typically, each airfoil analysis takes around 2 seconds when running on a Xeon 2.8GHz compute node with 2Gb memory.

3.5.3 Subsonic CFD Validation

The task of verifying and validating the results of the FLUENT analysis requires an assessment of the numerical accuracy of the computational mesh and a comparison of the results with experimental data. In order to perform these assessments, an existing airfoil design, for which experimental results have been published, must be selected to form a baseline. The NASA low-speed (LS) airfoil family (McGhee et al. [1979]) provides geometries which are ideal for such a baseline. These airfoils have undergone extensive development and have been used as wing sections on low speed civil aircraft such as the Stoddard-Hamilton Glassair and the Adam Aircraft A500 and A700, as well as in experimental research (see, for example, Zerihan and Zhang [2000]). The low-speed family are designed to operate at a lift coefficient of 0.4, and typically a flow speed Mach number of 0.15 and Reynolds number of 4×10^6. These flow conditions are used in the present verification and validation studies. The NASA Technical Memorandum 4003 (Ferris et al. [1987]) contains detailed experimental data for the symmetrical NASA LS(1)-0013 airfoil. This data has been collated such that comparisons can be made with results from FLUENT and VGK for this same airfoil. Because the flow solvers are validated with the flow conditions specified in the NASA report, these conditions are used in calculating the design objective for the two design methods; this also allows any resulting airfoil designs to be compared directly with the NASA low-speed airfoils.

Note that throughout this thesis, validation of CFD simulations is performed at the design conditions and using standard geometries, which are not only *sensible* but also perform well. One could perform validation for a nonsensical shape, or for off-design conditions such as when separation occurs, provided that reliable and published experimental or computational data exists. Knowledge of the accuracy of the solution for a *good* design at *sensible* conditions is thought to be sufficient in this thesis; it is assumed that the optimization algorithm avoids poor designs and therefore the accuracy of such results is less important. What is important is that the CFD predicts the design trends to a good level of accuracy.

The baseline NASA 0013 airfoil has a blunt (finite-thickness) trailing edge; since the above CFD solvers are set up to operate on a sharp tailing edge, this NASA shape must be modified. The modified upper and lower airfoil contours, z_u and z_l, are generated from the original NASA 0013 contours, z_u^{0013} and z_l^{0013}, using the following expression.

$$z_u(x) = z_u^{0013}(x) - x \cdot z_u^{0013}\big|_{x=1} \quad , \quad z_l(x) = z_l^{0013}(x) - x \cdot z_l^{0013}\big|_{x=1} \quad , \text{ for } \quad 0 \le x \le 1. \tag{3.1}$$

The resulting shape is shown in Figure 3-3 with the original NASA airfoil. The modified airfoil is analysed using FLUENT and VGK for the purposes of validation.

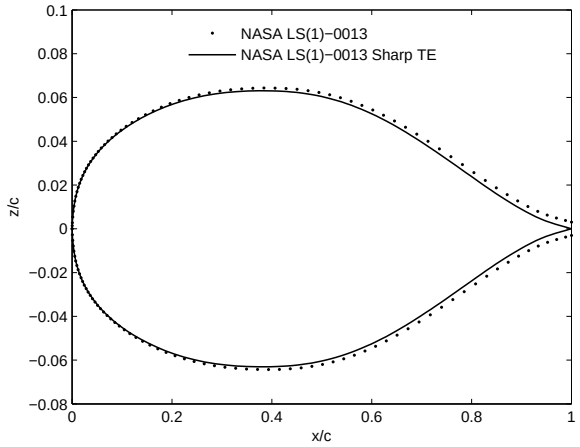


Figure 3-3 The NASA LS(1)-0013 airfoil, and a variation of this shape featuring a sharp trailing edge.

The subsonic FLUENT analysis has been verified with respect to the dependence of the results on the mesh setup. This setup is now summarized, and full details can be found in Appendix A1. The boundary of the *O*-mesh is a circle with a radius of approximately 29 metres. There are 398 cells defining the airfoil surface and 53 rows, giving a total of 21094 mesh elements. The cells are grown in size from the surface, with a first cell height of 0.045% airfoil chord. In this configuration an increase in the mesh size gives a negligible variation in the resulting force coefficients, indicating good numerical accuracy. On studying the convergence history of the solver using this setup, it is observed that 3000 iterations of the RANS calculations are sufficient to provide a converged solution. At this point the variation in the drag coefficient for the airfoil is within ±0.1 counts (±0.00001 c_d) of the fully converged value. Calculating the drag using the current CFD setup takes on average 19.6 minutes when running on four 2.4GHz processors; this includes the time required to iterate the angle of attack when a target lift is specified. Further details relating to the parallel processing management are given below in Section 3.9.

Figure 3-4 shows the FLUENT surface C_p distribution for the NASA LS(1)-0013 airfoil predicted using the above setup and for the flow conditions specified by Ferris et al. [1987], i.e., a flow speed Mach number of 0.15, a Reynolds number of 4×10^6 and an angle of attack of 4.1°. Also shown is the result

predicted by the VGK code and the experimental results (but with a blunt trailing edge), for the same airfoil and operating conditions. Figure 3-4 confirms the strong similarity between the pressure profiles from FLUENT, VGK and the experimental data. The increase in surface pressure predicted by the two CFD solvers is partly due to a small reduction in airfoil thickness, compared to the original NASA shape, as a result of closing the trailing edge (Figure 3-3).

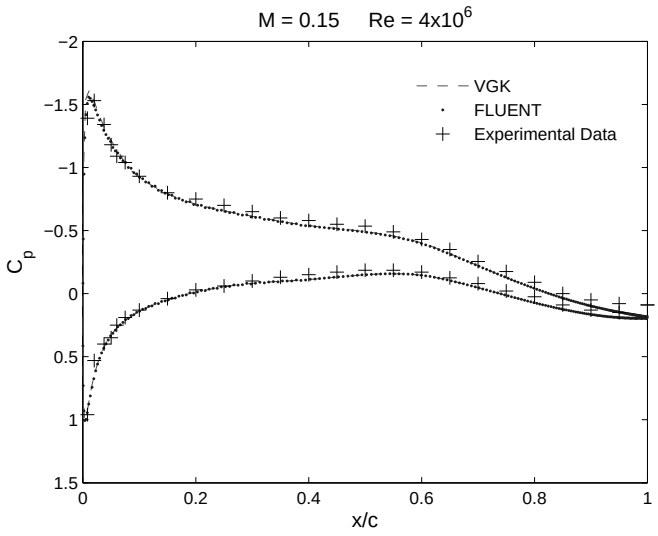


Figure 3-4 Comparison of pressure distributions generated using the FLUENT and VGK CFD solvers, for the NASA LS(1)-0013 airfoil. These are also compared against experimental data.

3.5.4 Transonic CFD Validation

For the purposes of evaluating the FLUENT analysis to be used for transonic flows, it is logical to use an airfoil designed to operate at high Mach numbers. The RAE 2822 airfoil is commonly used as a test case geometry for the validation of transonic CFD simulations. This is a cambered shape of 12% thickness and features a sharp trailing edge. Cook et al. [1979] have published experimental surface pressure measurements for this airfoil, for various flow Mach numbers and incidences. For the present CFD setup exercise and the design studies, a Mach number of 0.73 and Reynolds number 6.5×10^6 are selected.

As with the subsonic analysis, the mesh for the transonic case has been setup in an attempt to achieve sound numerical accuracy. Again, an *O*-mesh topology is used; the flow boundary has a radius of approximately 12 metres, with the airfoil normalized to a chord of one metre. Note that the external boundary is closer to the airfoil compared to the subsonic analysis setup, however this radius was found to be sufficient to provide the required accuracy (see Appendix A2). As in the subsonic mesh, there are 398 surface cells with a surface cell height of 0.045% chord, in this case giving 19104 mesh cells in total.

Full details of the mesh-dependency study are given in Appendix A2. Using this setup, a converged result, in terms of drag, takes on average 22.6 minutes when running on four 2.4GHz processors, including the time required to iterate the angle of attack when a target lift is specified. Note that this run time is higher than for the subsonic analysis, for an identical solver setup and number of iterations, despite the number of mesh cells being marginally higher in the subsonic case. This increased cost is associated with solving the flow in the region of the compression wave, as there is a greater level of numerical instability.

Figure 3-5 shows the surface pressure predictions using FLUENT and VGK, compared against the experimental data. The angle of attack used is 3.19° (case number 9 in Cook et al. [1979]). The surface pressure results from FLUENT are in good agreement with the experimental data. There is slightly higher compression aft of the shock on the upper surface, and lower pressure on the rooftop, but the shock position is predicted accurately. In the case of VGK, the solver setup used above for the subsonic analysis is not suitable for flows containing moderate strength shocks, since this causes the calculations to diverge. For this reason, the solver relaxation factors have been modified in order to assist convergence when transonic flow is encountered. It can be seen from Figure 3-5 that the surface pressure predictions from VGK closely match the FLUENT results, except in the region of the shock. The shock position is slightly too far aft and excessive compression is predicted, followed by an expansion. Nevertheless, these results are deemed to be accurate enough for the purposes of inverse design, and using this setup the solver convergence is improved.

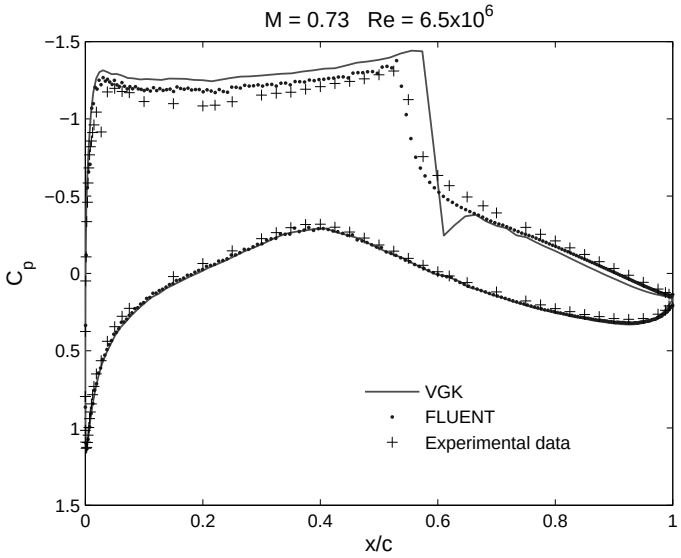


Figure 3-5 Comparison of pressure distributions predicted by the FLUENT and VGK solvers for the RAE2822 airfoil, shown with experimental data.

3.5.5 Determination of Drag

As already noted, the design objective for the airfoil optimization search is total drag for a fixed level of lift. The drag on the airfoil comprises two components: tangential or skin-friction forces, due to viscosity, and pressure drag. The pressure drag component arises from the boundary layer displacement effect and, in the case of flows with a Mach number higher than around 0.7, wave drag due to the presence of shocks. Predicting this drag accurately using CFD is significantly more challenging than the prediction of accurate surface pressures, since its value is numerically small and therefore prone to error during the calculations. It has been estimated that a 1% increase in lift-to-drag ratio during takeoff is equivalent to a 2800*lb* increase in total payload, for a civil aircraft (van Dam [1999]). The prediction of drag using CFD is therefore vitally important to aircraft manufacturers, but simultaneously to gain a reduction in drag is technically difficult and improvements are always incremental. Accurate methods of drag prediction are therefore of great interest to industry and academia.

Progress with drag prediction using CFD is reported frequently in the literature; examples cited here are the articles by Giles and Cummings [1999], Lock [1986], van Dam [1999] and Levy et al. [2003]. The two most popular methods used for calculating drag are surface force integration and far-field integration methods. Surface force integration simply integrates the normal and shear stresses over the body surface. This results in good predictions for lift, but errors in the prediction of the leading edge stagnation point can result in inaccurate drag estimations. This approach does not allow the calculation of individual viscous and wave drag components. The alternative is to use far-field methods, which calculate the momentum deficit over a control volume a large distance from the body surface. Since this does not rely on greatly accurate or detailed surface pressure measurements, it generally produces more accurate predictions. The wave drag contribution can be calculated separately by integrating flow properties before and after the shock, however this requires the shock location to be found.

In the present design studies drag is calculated by the high-fidelity analysis, FLUENT, while the low-fidelity code, VGK, is used only to calculate surface C_p distributions in EMFID. The drag in FLUENT is calculated via surface force integration. While this approach is not the most accurate, for the purposes of design the requirement is not necessarily for supremely accurate drag predictions. As previously stated, the requirement is to predict accurate trends in the performance, and the simple algorithm used in FLUENT is therefore likely to be adequate. Further, since the design objective is total drag, calculation of the individual drag components is not required.

3.6 The Benchmark Optimization Method

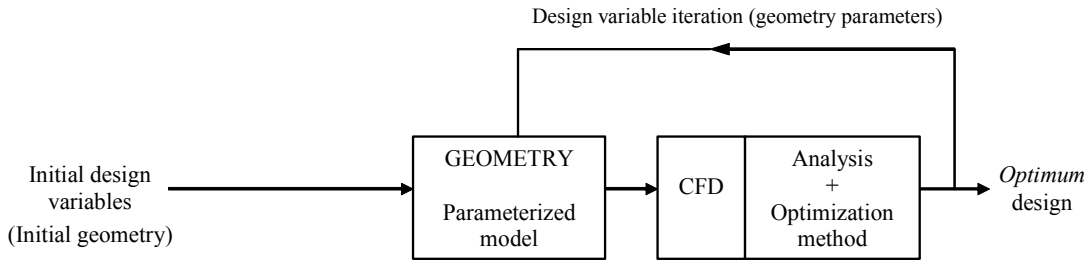


Figure 3-6 Flowchart illustrating the benchmark (direct) design search strategy.

A *traditional* design optimization method which makes use of a geometry-based parameterization is used here to act as the established and accepted practice in aerodynamic shape optimization. This is the so-called *benchmark* method, against which the EMFID parameterization method is compared. The benchmark method takes the same form as the strategy described in Figure 2-3, and this is shown again in Figure 3-6. The airfoil geometry is parameterized and design iterations, requiring high-fidelity CFD analysis, are automated using an optimization algorithm.

3.6.1 Parameterization Techniques

As already discussed, the choice of parameterization method is a critical factor in the performance of direct searches such as the benchmark design method used here. Song and Keane [2004] compared an interpolating B-spline method with an orthogonal shape function based method, and reported that while the spline approach is computationally expensive it is able to capture a larger range of geometries accurately; Samareh [1999] also reports that the use of polynomial splines is well suited to a two-dimensional study. Spline approaches also have the advantage that they are available in any CAD package. In selecting an appropriate parameterization, the logic used here is twofold. First, a parametric model is required which is typical of the current approaches used by academia and industry for airfoil conceptual design. This must be able to generate a large range of detailed shapes, using a representative number of design variables. Second, in addition to the aforementioned model, it is useful to investigate a parameterization which uses the same number of variables as the EMFID model (described later). Therefore, two geometry-based models are described here, the first uses 13 design variables, and the second uses six variables.

For the higher-dimensional benchmark model, an interpolating segmented cubic polynomial spline is chosen to parameterize the airfoil. This technique is described in Section 2.1.2. The curve is commonly used in design situations; however, when the representation is applied to an airfoil shape, problems arise at the leading edge where the gradient often approaches infinity. The use of a cubic polynomial in this region is unnatural; the designers of the NACA airfoil sections (Jacobs et al. [1933]) overcame this by using a polynomial expression which included an additional $\sqrt{x}$ term. Thus, for the leading edge segment of the airfoil the terms with x of order two and three have been replaced with a $\sqrt{x}$ term. Therefore, the gradient at the start of the spline does not have to be specified; the singularity at the leading edge gives an *infinite* gradient.

The airfoil shape is defined by ten cubic spline segments, or eleven data points in (x,z) space which are interpolated by the curve (Figure 3-7). The design variables are selected from the possible x and z movements of the data points, with the aim of minimizing the number of design variables while retaining the ability to produce smooth and varied airfoil shapes. The airfoil is separated into upper and lower surfaces, while the leading edge point (0,0) is shared by both surfaces and remains fixed. Of the remaining five points on the upper surface, the near leading edge point (point A in Figure 3-7) is free to move in both x and z directions and the trailing edge point (B) is fixed, while the other three points are constrained to movement in the z direction only. The same applies to the lower surface, with the exception of the point adjacent to the trailing edge point (C), which is free to move in both directions. The trailing edge point is fixed at the position corresponding to the initial design in the search process. Additionally, the gradient of each surface at the trailing edge is added to the list of design variables, since the exit angle is important in the governing aerodynamics. Thus, there are 13 design variables in total defining the geometry of the airfoil, which must be manipulated by the optimizer and analyzed using CFD software. It is thought that 13 is a reasonable number of variables for an airfoil design problem of this nature; it is not uncommon for such a problem to make use of 22 or more variables; see, for example, Song and Keane [2004], Lépine et al. [2001], Li et al. [2004] and Painchaud-Ouellet et al. [2006].

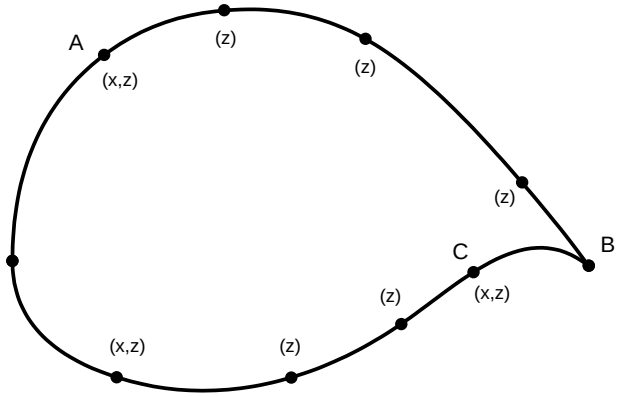


Figure 3-7 13-variable airfoil geometry parameterization for the benchmark method using polynomial splines, showing control point degrees of freedom.

The EMFID parameterization (described later) uses six design variables in total. It is desirable to make comparisons with a benchmark method which also considers six variables. For this purpose, a second geometry parameterization is employed, in which the airfoil is represented by a B-spline curve for each airfoil surface. These curves are cubic in form, and are calculated by requiring that the curve interpolates a number of data points. In this case there are five data points on each surface, positioned at x/c= [0 0.03 0.32 0.74 1]. In order to facilitate the singularity at the leading edge, i.e., an *infinite* gradient, the first and second B-spline control points are both fixed to be on the z axis. As a result of this action, the defining data points are not exactly interpolated, but the degree of control afforded by manipulating these points is only marginally affected. The leading edge point is fixed at (0,0) and the trailing edge point is fixed at the value corresponding to the initial design, while the vertical translation of the three remaining points are the parametric variables, giving the total of six design variables for the complete airfoil profile (Figure 3-8). Since there are fewer defining data points, this model is less capable of performing localized changes to the surface curvature, compared to the 13-variable model, and it cannot be used to specify the trailing edge angle. However, the use of fewer variables should make the process of optimization more straightforward.

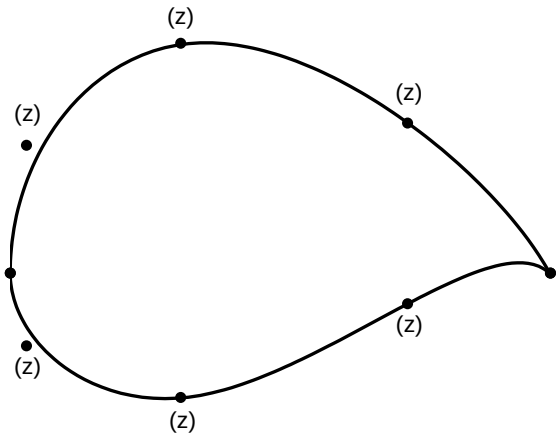


Figure 3-8 Six-variable airfoil parameterization for the benchmark method using B-splines, showing control point degrees of freedom.

3.6.2 Benchmark Optimization Setup

Using the high-fidelity CFD solver (FLUENT), for each airfoil design iteration the lift and drag coefficients (c_l and c_d) are calculated, which are then used as metrics of performance. As already noted, the design objective of the optimization procedure is to minimize c_d calculated at a constant value of c_l, allowing the angle of attack to float. For optimization using high-fidelity CFD, it is imperative to minimize the number of objective function calls. Therefore, the optimization strategy in the benchmark design method uses a response surface model (RSM) approach. The RSM optimization routine is implemented using the OPTIONS[1] design exploration system, operating in the MATLAB®[2] environment using the GEODISE[3] toolkit. The setup of this procedure is described next.

The DoE used to seed the initial data-base is a Latin hypercube, which has good coverage of the design space and has the advantage of representing each variable's range equally. Additionally, OPTIONS allows the random number sequence to be changed giving different, but repeatable, initial DoE sets. The response surface model used is an interpolating cubic spline radial basis function (RBF). Once the RSM has been built, it is searched using a genetic algorithm (GA), implemented in OPTIONS. The GA gives a relatively thorough search of the whole design space, which can be tolerated since calls to the response surface are very fast; here 5000 search evaluations are used. Instead of searching for a single optimum point on the surface, parallel update points are extracted from the search. For this method, five parallel

[1] Information can be found on the world wide web at http://www.soton.ac.uk/~ajk/options (cited April 2006).
[2] Information can be found on the world wide web at http://www.mathworks.com/ (cited April 2006).
[3] Information can be found on the world wide web at http://www.geodise.org (cited April 2006).

update points are requested, taken from a cluster analysis of the final population in the GA search. After evaluating the new design points, the RSM is updated.

For a given iteration of the design variables and call to the CFD analysis, it is possible that the calculations may fail for some reason (if the CFD calculations diverge, for example). An important decision concerns the handling of these failed design points. For the current method, design points which fail are not included when constructing the RSM. When updating the response surface, however, if all of the requested update points fail then one of these (the first) is included in the RSM and given an objective function value equal to the average of the objective values recorded so far. In this way, the coverage of the data set is statistically unaltered, while the RSM is altered such that the optimization algorithm does not stall, i.e., it does not request the same five update points repeatedly.

3.7 The EMFID Method

The EMFID design search strategy is illustrated in Figure 3-2. In summary, a call to the objective function in EMFID takes the design variables, generates the corresponding target pressure distribution and uses an inverse design process (low-fidelity CFD) to calculate an airfoil geometry which realizes the specified pressure variation. This shape is then passed to the high-fidelity CFD analysis (FLUENT) for calculation of c_d at the required value of c_l. In order to make fair comparisons between the alternative method and the benchmark direct search method described above, the high-fidelity CFD analysis and implementation of the optimization algorithm are set the same for both methods. Therefore, the RSM approach described for the benchmark strategy is also used as the optimization method for EMFID, shown in Figure 3-2. The EMFID method requires a parametric model of the C_p distribution, and additionally makes use of an inverse design method; these items are described next.

3.7.1 A Parameterization Technique for Subsonic Airfoils

The representation of the pressure (or velocity) distribution for an airfoil has been attempted by various authors using different approaches, almost invariably applied to the design of transonic airfoils. van Egmond [1990] formulated a set of aerodynamic shape functions for a transonic pressure distribution, capable of representing a wide range of airfoil flows (this was also used by Ahn et al. [2001]). Obayashi and Takanashi [1996] used B-spline curves to represent the pressure profile. Gopalarathnam and Selig

[2002] chose to parameterize the velocity distribution, by splitting the surfaces into segments over which the velocity difference was specified. All of these methods use in excess of 12 design variables, partly owing to their ability to represent transonic as well as subsonic flow regimes. Regarding this rather large dimensionality, it must be noted that the aforementioned authors do not use such a parameterization for the purpose of reducing the number of design variables for the optimization process, the EMFID *raison d'être*. In what follows, a parametric model is described for a subsonic, i.e., shock free, C_p profile; an evolution of this model for flows with a shock is discussed in the next section.

Similar to the benchmark parameterization method, the objective here is to allow the generation of a wide range of realistic subsonic C_p distributions, while also limiting the dimensionality of this model. Here, the C_p distribution is parameterized using a B-spline curve for each airfoil surface, extending from 1.5% chord to the trailing edge. Each B-spline curve contains four knots and four control points. These are constructed on a knot vector of four zeros and four ones, giving a cubic Bézier curve. The control point locations are determined by specifying that the curve must interpolate four data points. The chordwise positions of these data points are x/c=[0.015, 0.6, 0.85, 1]. The height of the trailing edge point is fixed while the heights of the three remaining points are the profile design variables, giving a total of six variables for the complete C_p distribution. Recall that the benchmark parameterizations, described above, use 13 and six design variables. The first 1.5% of this target C_p profile is used to represent the decay from stagnation pressure. On each surface, a straight line is constructed from C_p=1 to the start of the B-spline curve at 1.5% chord. Figure 3-9 shows an example of a C_p profile represented using this parameterization, illustrating the positions of the defining data points and the B-spline polygon.

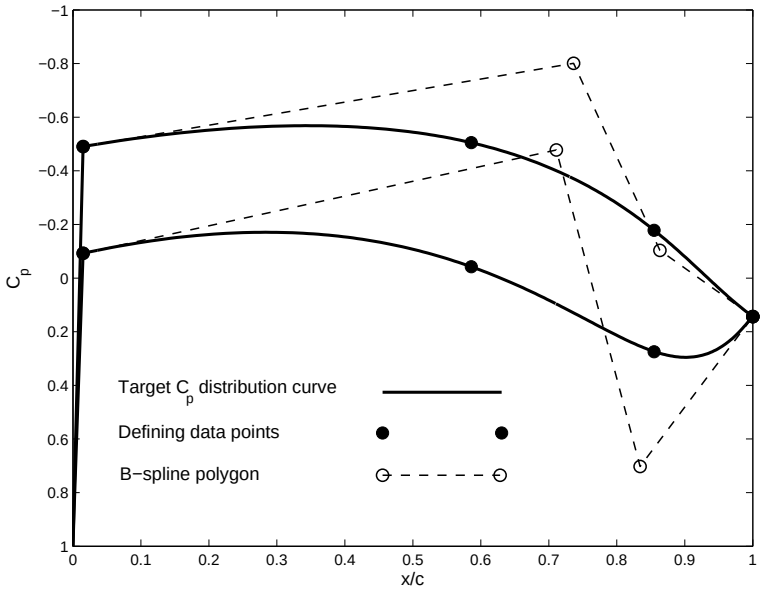


Figure 3-9 Parameterization of a subsonic C_p distribution using B-spline curves.

Using this parameterization technique, the EMFID design search involves manipulation of the shape of the C_p distribution, and since the integral of surface pressure in x equates to the normal lift this results in a different target lift depending on the variable values. To begin with in the current work, a direct coupling is maintained between the low-fidelity and high-fidelity analysis of each airfoil design. In other words, the inverse design (using VGK) and the performance calculation (using FLUENT) are performed for identical flow conditions and seeking to achieve the same target lift. Therefore, the target pressure coefficient values are scaled such that the enclosed area is equal to the required target c_l. Of course, this enclosed area is equal to the normal force and not the airfoil lift *per se*; however for small angles of attack this method provides a close approximation to the required target c_l.

For each iteration of the target pressure distribution generated by the optimization procedure (Figure 3-2), a geometry which realizes this target at the chosen flow conditions must be found by inverse design; this is the subject of Section 3.7.3. Following the inverse design step, the airfoil geometry is analyzed using the same high-fidelity CFD as used for the benchmark method. There are therefore three forms taken by the airfoil pressure distribution during an objective calculation in EMFID: the parameterized target, the profile which most closely matched this target during inverse design (VGK, see Section 3.7.3), and the profile generated in the final performance calculation (FLUENT). Figure 3-10 shows these three forms for the analysis of the NASA LS(1)-0413 airfoil (McGhee et al. [1979]). Since the flow conditions and target lift are the same in all three cases, the pressure distributions are the same (or at least very closely matched), confirming the direct coupling between the results of the analysis codes. It is not immediately

apparent how the EMFID method would perform if the design conditions in each analysis were different; this subject is to be investigated in Chapter 1.

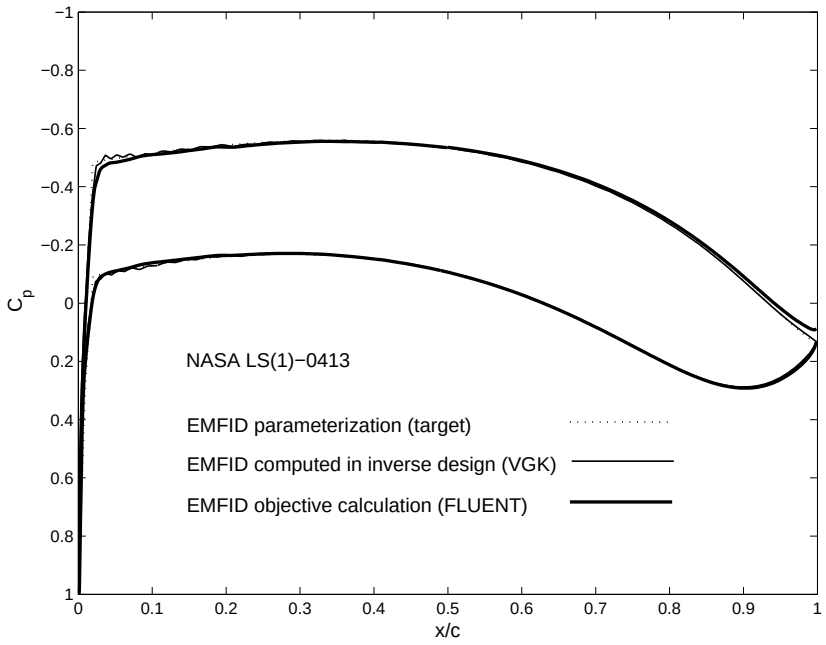


Figure 3-10 Pressure distributions generated in an objective calculation in EMFID, for the NASA LS(1)-0413 airfoil.

3.7.2 A Parameterization Containing a Shock

At high subsonic Mach numbers above around 0.7, the flow over an airfoil can become locally supersonic. On encountering an adverse pressure gradient the flow decelerates to a subsonic Mach number and in doing so a compression or shock wave is produced. Aircraft routinely fly in this transonic regime to improve engine efficiency, and mitigation of the associated wave drag has led to research into a new type of *supercritical* airfoil; see for example, the work by Harris [1990] and Sobieczky and Seebass [1984]. The shock manifests itself in the surface C_p profile as a sudden step in pressure, the strength and position of which can vary dramatically. In order to design transonic airfoils using EMFID, it seems logical at this stage to propose a parametric model for a C_p profile with a shock. To represent such a distribution parametrically must inevitably require more design variables than the model described above, while the geometry-based benchmark parameterization can remain unchanged. This could potentially be a shortcoming with the EMFID method. However, the ability to specify the position and strength of the shock in EMFID could allow the optimizer direct control over the wave drag. This direct coupling is not

present in the benchmark method, since there is a more complex relationship between the geometry variables and drag.

The transonic C_p distribution is parameterized using the aforementioned B-spline model as a starting point. Two variables are added to this description, these are the shock chordwise position, x_{shock}, and the pressure rise across the shock, ΔC_p. The total number of variables is therefore increased from six to eight. The original upper surface target pressure distribution given by a cubic B-spline curve, $C_p^{(s)}$, is modified as follows.

$$C_p^{(t)}(x) = C_p^{(s)}(x) + \Delta C_p, \qquad \text{for all} \quad 0 < x < x_{\text{shock}}. \tag{3.2}$$

$C_p^{(t)}$ is the upper surface pressure distribution used as a target for inverse design under transonic conditions. An example of a pressure distribution generated using this method is shown in Figure 3-11. This is a rather simple technique, and always generates a discontinuous step in pressure when ΔC_p is finite. This is somewhat restrictive, since it is known that weak shocks in particular often feature a more gradual pressure recovery. However, it is not obvious what advantage such a flow feature would bring, and so this simple model may be sufficient.

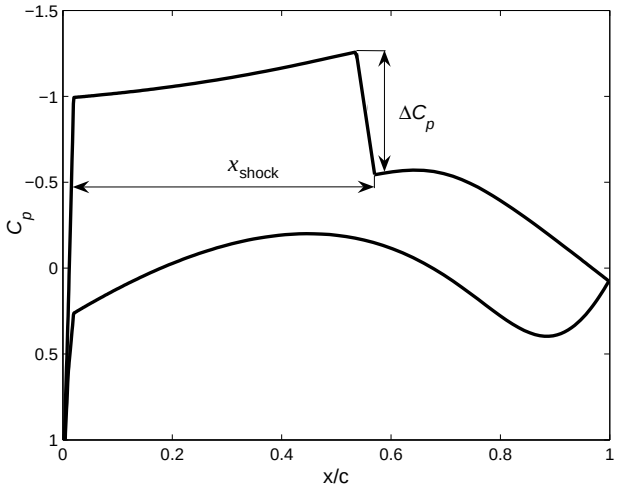


Figure 3-11 An example of a parameterized target pressure distribution, showing the two variables defining a shock.

3.7.3 Inverse Design

The EMFID method requires an inverse design process to derive an airfoil geometry which produces the specified target pressure distribution, or at least a reasonable approximation to it, at each iteration of the design search. Here, the full potential CFD code VGK is used to calculate the pressure distributions for this inverse process. To begin with in the airfoil design case studies to follow, VGK uses the same flow conditions as are used for the high-fidelity CFD calculation. It must be able to accurately converge onto the target pressure distribution while minimizing the number of calls to VGK used to do this. For the present airfoil design studies, the aim is to evaluate the EMFID method using both subsonic and transonic flow conditions, hence the inverse method employed must also be capable of running under these conditions.

The inverse design approach adopted here is principally the same as the method proposed by Davis [1980]. The method is based on the iterative residual-correction concept, illustrated in Figure 2-4. For a given chordwise station, the residual is the difference between the target and computed pressure distributions. The magnitude of this residual determines the amount of surface alteration performed for each station. Under certain assumptions, for a local Mach number, M_L, below unity the local surface pressure is proportional to the local curvature, and so for subsonic flows the corrections are performed on the surface second derivatives. At the k-th iteration the second derivatives, $z''(x)$, of the airfoil shape are corrected according to the following expression:

$$M_L < 1 \qquad z''_k = z''_{k-1} + \left(\frac{\mathrm{d}z''}{\mathrm{d}C_p} \right)_{approx} (C_p^T - C_p^{k-1}) \,. \tag{3.3}$$

For inverse design under transonic flow conditions, in regions where the local Mach number is greater than unity, the surface first derivatives, $z'(x)$, are corrected:

$$M_L > 1 \qquad z'_k = z'_{k-1} + \left(\frac{\mathrm{d}z'}{\mathrm{d}C_p} \right)_{approx} (C_p^T - C_p^{k-1}) \,. \tag{3.4}$$

For each chordwise ordinate, C_p^T is the target pressure coefficient and C_p^{k-1} is the design pressure coefficient computed by the flow solver (VGK) at iteration k-1. The gradient terms $\left(\mathrm{d}z''/\mathrm{d}C_p\right)$ and $\left(\mathrm{d}z'/\mathrm{d}C_p\right)$ are calculated using approximate flow formulae, which can be relatively crude given the iterative nature of the design process. The flow formulation adopted here is the same linearized potential theory used by Davis [1980]. Thus, the inverse design process proceeds as follows: 1) the pressure

distribution is calculated for an initial airfoil using the potential flow solver VGK, and the difference to the target pressure distribution is calculated; 2) an approximation to the pressure distribution is calculated using linearized potential flow theory, allowing the gradients $\left(\mathrm{d}z''/\mathrm{d}C_p\right)$ and $\left(\mathrm{d}z'/\mathrm{d}C_p\right)$ to be determined; 3) the surface derivatives are corrected using Eq. (3.3) and Eq. (3.4); 4) integration yields the corresponding airfoil surface. The process is repeated until convergence is reached, i.e., when the computed C_p profile is satisfactorily close to the target profile.

The integral of the square of the pressure error, I, between the target and computed profile is used as a measure of the convergence of the inverse design process, i.e.,

$$I = \frac{1}{2}\int_S (C_p^T - C_p^k)^2 \,\mathrm{d}S \,, \tag{3.5}$$

where S represents the airfoil surface. This error measure is also described by Jameson [1996]. The magnitude of I diminishes rapidly as the iterative procedure converges onto the desired pressure profile. Eventually, the computed pressure profile is unable to match the target any closer, and the error increases fractionally; at this point the process is deemed to be converged and is halted. Note that the initial airfoil, used to start the inverse process, is always the symmetrical NASA LS(1)-0013 shape (Ferris et al. [1987]).

The inverse design procedure requires calculation of surface first and second derivatives, and these must also be integrated twice once they are corrected. The procedure used here is a central differencing scheme for differentiation, and following correction of the surface derivatives a simple first order interpolation quadrature is employed. Regarding this integration calculation, for purely subsonic flows the specified boundary conditions are the positions of the leading edge and trailing edge; for flows with a shock the boundary conditions impose surface and first derivative continuity at the surface point immediately upstream of the shock. A number of authors have described problems in the leading edge region when performing the inverse design (as discussed by Milholen [2001]), largely caused by the high surface curvature in this region. This problem has also been encountered with the inverse process described here, i.e., the surface correction scheme is found to converge significantly more slowly in the leading edge region compared to the rest of the airfoil. However, the convergence can be greatly improved if the density of the defining co-ordinates is sufficiently high near the leading edge.

The objective of the design strategies described in this Chapter is to minimize drag for a fixed level of lift while allowing the angle of attack, α, to be varied: α therefore becomes a variable in the inverse design process. Indeed, since the airfoil leading and trailing edge positions are fixed during all surface

alterations, α must be made a variable if the specified target C_p distribution is to be matched closely. Increasing α gives a monotonic increase in lift, and since this increases the area between the C_p curves for the upper and lower surfaces, α can be adjusted at each inverse design iteration using an expression similar to Eq. (3.3):

$$\alpha_k = \alpha_{k-1} + \theta \sum_{S} (C_p^T - C_p^{k-1}) , \tag{3.6}$$

where α_k denotes the angle of attack at iteration k and S represents the airfoil surface.

In Eq. (3.6), θ is a relaxation factor applied at each iteration to prevent excessive corrections to the angle. A value for θ of 0.12° is used in the current study. The use of relaxation provides a facility to control the convergence of an iterative procedure, a bigger factor gives faster convergence but increases the risk of instability. The use of such factors is not uncommon in design optimization, although the appropriate magnitude is likely to depend on the problem setup. A relaxation factor is also applied to the pressure residual term in Eq. (3.3) and Eq (3.4), since experience with this inverse design method revealed that the surface corrections at each iteration can be overly large. The process likely requires the inclusion of relaxation factors because of its iterative nature and because of the simplicity of the surface pressure approximation. Thus, the surface alterations are expressed as:

$$M_L < 1 \qquad z_k'' = z_{k-1}'' + \phi_S \left(\frac{\mathrm{d}z''}{\mathrm{d}C_p} \right)_{approx} (C_p^T - C_p^{k-1}) , \tag{3.7}$$

$$M_L > 1 \qquad z_k' = z_{k-1}' + \phi_T \left(\frac{\mathrm{d}z'}{\mathrm{d}C_p} \right)_{approx} (C_p^T - C_p^{k-1}) . \tag{3.8}$$

ϕ_S and ϕ_T are the relaxation factors applied to the subsonic and transonic surface alterations, respectively. As with θ, these relaxation factors can be used to control the progress of convergence. Relaxation is also used by Davis [1980], although its magnitude is not specified. Experience with this inverse design method has shown that for transonic flows the magnitudes of ϕ_S and ϕ_T must be relatively small in order to provide robust convergence, while for a purely subsonic case ϕ_S can be significantly larger, accelerating convergence. Also, as previously mentioned, the VGK solver is setup differently for the transonic case in order to improve the CFD convergence for flows with strong shocks. For these reasons, it seems logical to use two distinct inverse design codes in this work. The *subsonic* code uses only Eq. (3.7), with a value of ϕ_S of 0.4. The *transonic* method employs both Eq. (3.7) and Eq. (3.8), using a more robust VGK setup

and relaxation factor magnitudes of ϕ_I=0.01, ϕ_S=0.05 on the upper surface and ϕ_S=0.3 on the lower surface.

The relaxation factor magnitudes specified here have been found by monitoring the number of iterations required for convergence of the inverse design and the minimum pressure error (Eq. (3.5)) achieved, as the relaxation factor is varied. Figure 3-12 and Figure 3-13 illustrate the process of selecting the magnitude of ϕ_S (Eq. (3.7)) for the subsonic inverse process. While Figure 3-12 shows that a factor of around 0.35 gives a slightly smaller error than 0.4, Figure 3-13 shows that a factor of 0.4 requires fewer iterations (35 compared to 54). Experience has shown that the form of these graphs is largely independent of the target pressure distribution.

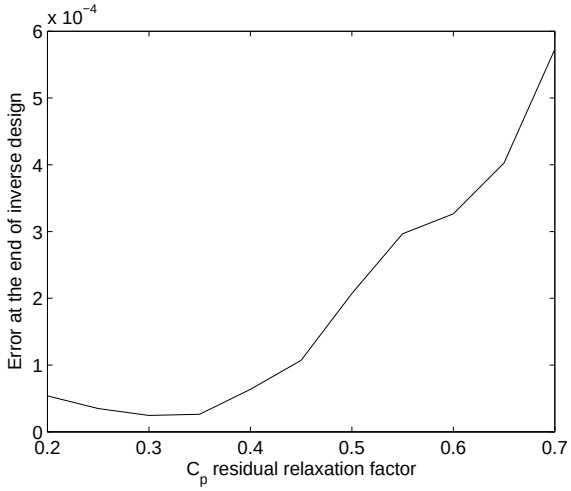


Figure 3-12 The error between the target and computed pressure distributions at the end of the inverse process, plotted against relaxation factor magnitude.

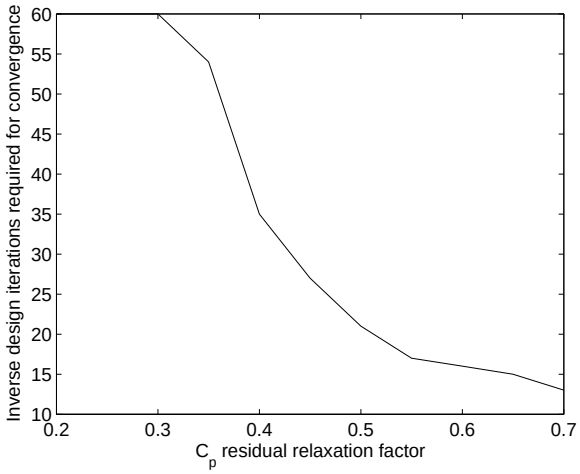


Figure 3-13 Number of iterations required for inverse design, plotted against relaxation magnitude.

Using the method described above, the subsonic inverse design step typically requires between 20 and 40 VGK calls to capture the target pressure profile, while the transonic method needs around 70. An example of a converged inverse design result using the subsonic setup is shown in Figure 3-14 and Figure 3-15. The target C_p profile is for a NACA 0012 airfoil at c_l =0.4, and for the flow conditions M=0.15 and Re=4×10^6. The converged C_p profile matches the target very closely, and this is reflected in the agreement between the corresponding geometries. A copy of the MATLAB code for the subsonic airfoil inverse design method is given in Appendix E.

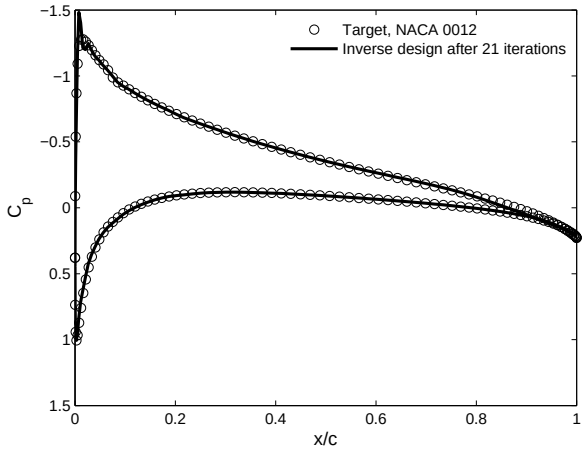


Figure 3-14 A converged inverse design result. The target is the C_p distribution for NACA 0012.

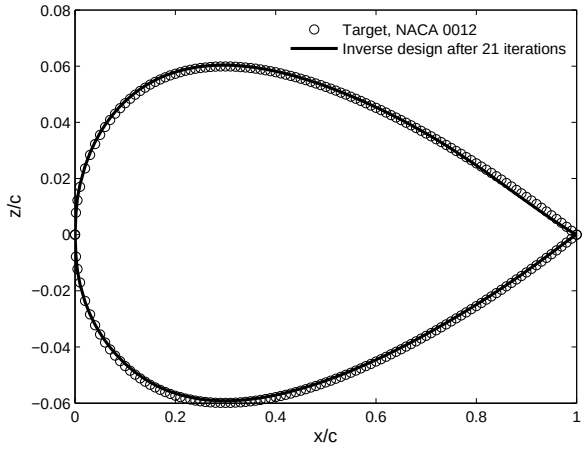


Figure 3-15 Comparison of geometries for a converged inverse design process, showing the design result and the shape corresponding to the target C_p profile.

3.8 Comparing the Two Methods

The two design methods described above are evaluated side by side to assess their relative efficiency. The more efficient design method is the one which reaches the highest level of design performance (minimum c_d) given a fixed computational budget. Also of interest is the rate of convergence of the search processes; do the methods converge, and, if so, to what level of design performance? In this work, as in any scientific investigation, it is fundamentally important to provide a *fair test*. That is, the control procedure and the experimental procedure should only differ in one variable, and one investigates the effect of this variable on the phenomenon of interest. In the present comparison this variable is the airfoil parameterization technique; the benchmark and EMFID procedures should ideally be identical in every other sense. The measures taken in attempting to provide a fair comparison are now outlined.

As described above, the design space is initially populated using a random Latin hypercube DoE and the RSM is calculated based on the objective values at these points. The shape of this initial RSM is heavily dependant on the DoE set used, and as such the progress of the optimizer can be very different in each case. In order to account for this, each method is run five times using different Latin hypercube DoE seeds. It is recognized that, statistically, five is not a sufficient number of evaluations in order to gauge the DoE dependence of a design search procedure. However, a complete design search run is computationally expensive, and so one is averse to performing many otherwise identical runs if this is unnecessary. It is thought that five is a sufficient number to provide a representative comparison.

The starting airfoil is the first point in the design search and is contained in the initial DoE set. Different initial designs are used for the subsonic and transonic design studies; in each case it is desirable to start the process with a design which performs poorly in terms of the objective function, thus testing the design improvement capability of the two methods. For the comparison between the benchmark and EMFID methods under subsonic flow conditions, the NACA 2414 four-digit airfoil (Jacobs et al. [1933]) is used as the initial design. This airfoil is designed for subsonic flow conditions, but is likely to perform poorly due to its primitive shape. For each parameterization technique used here, a set of design variables must be found which gives, to a reasonable approximation, the NACA 2414 geometry. This is achieved using a simple search technique which minimizes the difference between the parametrically computed airfoil and the NACA geometry. Figure 3-16 compares the original NACA geometry with the representations of this shape using the 13- and six-variable benchmark parameterizations. Both models align well with the target shape, the 13-variable definition matches particularly closely. The NACA 2414 pressure distribution for the subsonic design conditions is shown in Figure 3-17 (a), along with the EMFID parameterized representation of this C_p profile which is shown as a dotted line. On running the inverse design process

for this parameterized target, the C_p distribution shown by the dashed line in Figure 3-17 (a) is obtained. The corresponding airfoil geometry is shown in Figure 3-17 (b), along with the original NACA shape it is intended to emulate. It is observed that the shape resulting from the EMFID parameterization technique matches the *target* NACA 2414 airfoil with less precision than the two benchmark parameterizations. Ultimately, this is because the B-spline model is unable to exactly match the true NACA 2414 C_p distribution.

Starting airfoil for design studies at subsonic flow conditions

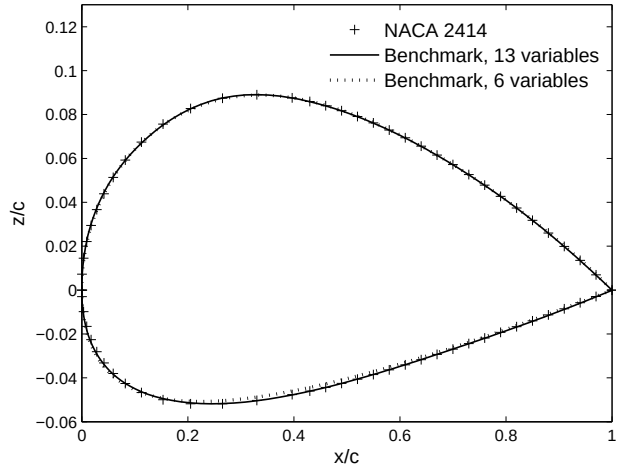
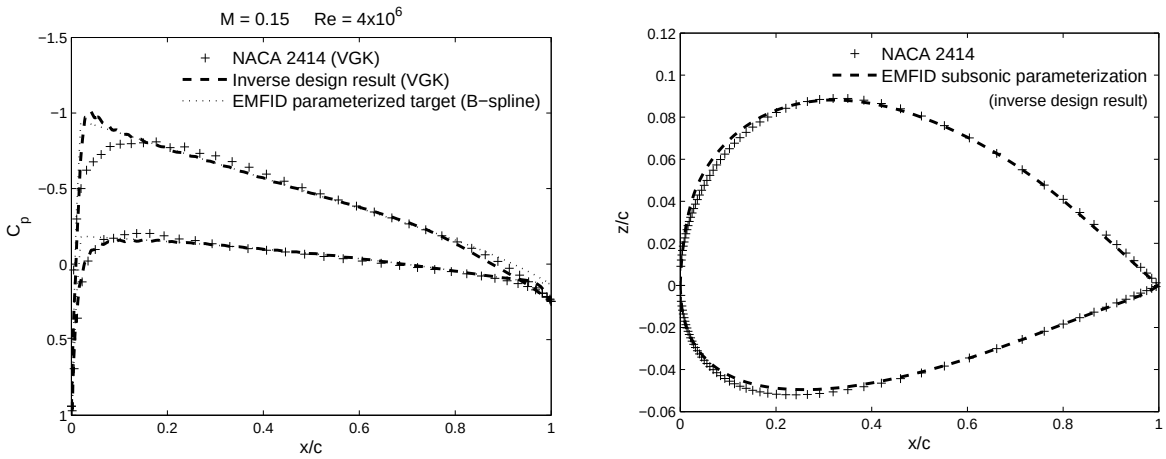


Figure 3-16 The NACA 2414 airfoil, and representations of this shape using the two benchmark parameterizations.



(a) **(b)**

Figure 3-17 (a) Pressure distribution for the NACA 2414 airfoil, calculated using VGK, the representation of this profile using the subsonic EMFID parameterization and the inverse design result. (b) The NACA 2414 airfoil and the shape resulting from inverse design on the parameterized target.

Starting airfoil for design studies at transonic flow conditions

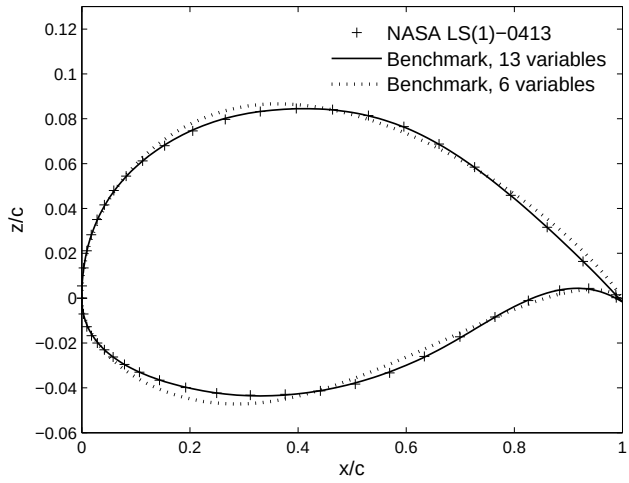
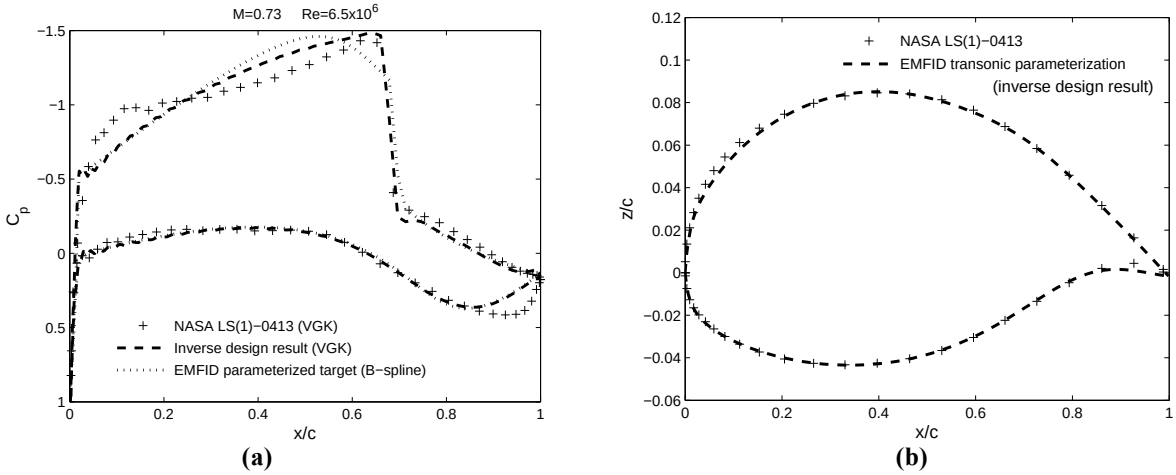


Figure 3-18 The NASA LS(1)-0413 airfoil, and representations of this shape using the two benchmark parameterizations.



(a) **(b)**

Figure 3-19 (a) Pressure distribution for the NASA LS(1)-0413 airfoil, calculated using VGK, the representation of this profile using the transonic EMFID parameterization and the inverse design result. (b) The NASA LS(1)-0413 airfoil and the shape resulting from inverse design on the parameterized target.

In the case of the transonic airfoil design studies, the initial airfoil shape is chosen to be the NASA LS(1)-0413 shape from the NASA low-speed family (McGhee et al. [1979]). This airfoil was designed using the same principles as the NASA supercritical airfoils and features a similar highly cambered aft lower surface. However, it is designed for subsonic conditions and is likely to perform poorly in terms of wave drag; this makes it a suitable starting design for evaluating the transonic design search processes. As for the subsonic design studies, a simple optimization search is run to determine the design variables which give a geometry closely matching the target initial shape. Figure 3-18 compares the target NASA geometry with the representations of this shape using the 13- and six-variable benchmark parameterizations. Figure 3-19 (a) compares the true 0413 C_p distribution with the EMFID parameterization of this profile and shows the inverse design result; Figure 3-19 (b) depicts the corresponding geometries. Similarly to the subsonic case, the 13-variable benchmark parameterization is able to represent the starting airfoil more closely than the EMFID model. However, in this case the six-variable benchmark model actually gives a rather poor representation; this is largely because the x positions of the defining data points have been arranged to best fit the NACA 2414 airfoil (Figure 3-16), leading to a reduced capacity to emulate other shapes. This effect demonstrates the inadequacy of manipulating only six surface points, each with a single degree of freedom.

The inferior ability to recreate standard airfoil shapes highlights a limitation of the EMFID method using the current B-spline model of the C_p distribution, because the 13-variable spline-based approach has been shown to do this more effectively. A more complex C_p distribution model could be configured such that the initial design is replicated more accurately, however such a model would involve more variables, defeating the object of the EMFID process. Moreover, it transpires that the current parameterizations are sufficiently flexible for the design of general subsonic and transonic airfoils. The minor discrepancy between the standard and recreated geometries does not disadvantage either method. Relative to the large range of possible airfoil shapes in the design space, the parametric benchmark and EMFID geometries align with the standard airfoils satisfactorily. Starting the two design search processes from these very similar shapes is thought to allow a fair comparison between the methods.

Another important factor when comparing the two methods is the bounds placed on the design variables. These must be equivalent for each method such that one method is not forced to search a much larger or smaller design space than the other. It is also desirable to maintain a conceptual design approach and permit novel and radical designs. To allow a direct comparison, the EMFID method must have bounds in pressure profile terms which are equivalent to the geometrical bounds of the benchmark method. However, this is difficult to achieve since a modification to the shape of one surface affects the pressure over both surfaces. In order to constrain the two methods fairly, bounds must be placed on design

parameters which exist in both methods. In the current study, constraints are therefore placed on the airfoil shape directly, namely on the maximum thickness, maximum camber and leading edge radius.

The variables for the benchmark method control the position of spline points on the airfoil surface, where those for the EMFID method control spline points which establish the shape of the target pressure distribution. These variables are therefore each given a relatively larger range, resulting in a large potential design space. For each set of variables selected by the optimization process of either method, if these variables generate an airfoil geometry which violates the constraints on thickness, camber or leading edge radius, the variable set and resulting geometry are rejected. The objective function is not calculated for rejected geometries, and they are not included when constructing the RSM. If, however, all five update points requested from the search of the RSM are rejected, one of these points is added to the surface and treated as a failed design point. This action prevents an identical RSM being generated which would stall the optimization process. With the bounds set up in this manner a direct comparison can be made between the optimization performance of the two methods, and their resulting geometries.

Figure 3-20 shows a more detailed diagrammatic description of the benchmark and EMFID design search processes. The optimization strategy, shown in Figure 3-20 a), is the same for both the benchmark method and EMFID. In summary, the process calculates the objective function as dictated by a random DoE, starting from the design variables representing the NACA 2414/NASA LS(1)-0413 airfoil, before building a response surface in order predict promising update points. Update points are evaluated until the computational budget is exhausted. A call to the objective function in EMFID (Figure 3-20b) takes the design variables, calculates the corresponding pressure distribution, and then scales it to give the required total area and target lift. The inverse design code (low-fidelity CFD) is then used to calculate an airfoil geometry which realizes the specified pressure variation. Provided that it satisfies the geometrical constraints, this shape is passed to the high-fidelity CFD analysis for calculation of c_d and α at the required value of c_l.

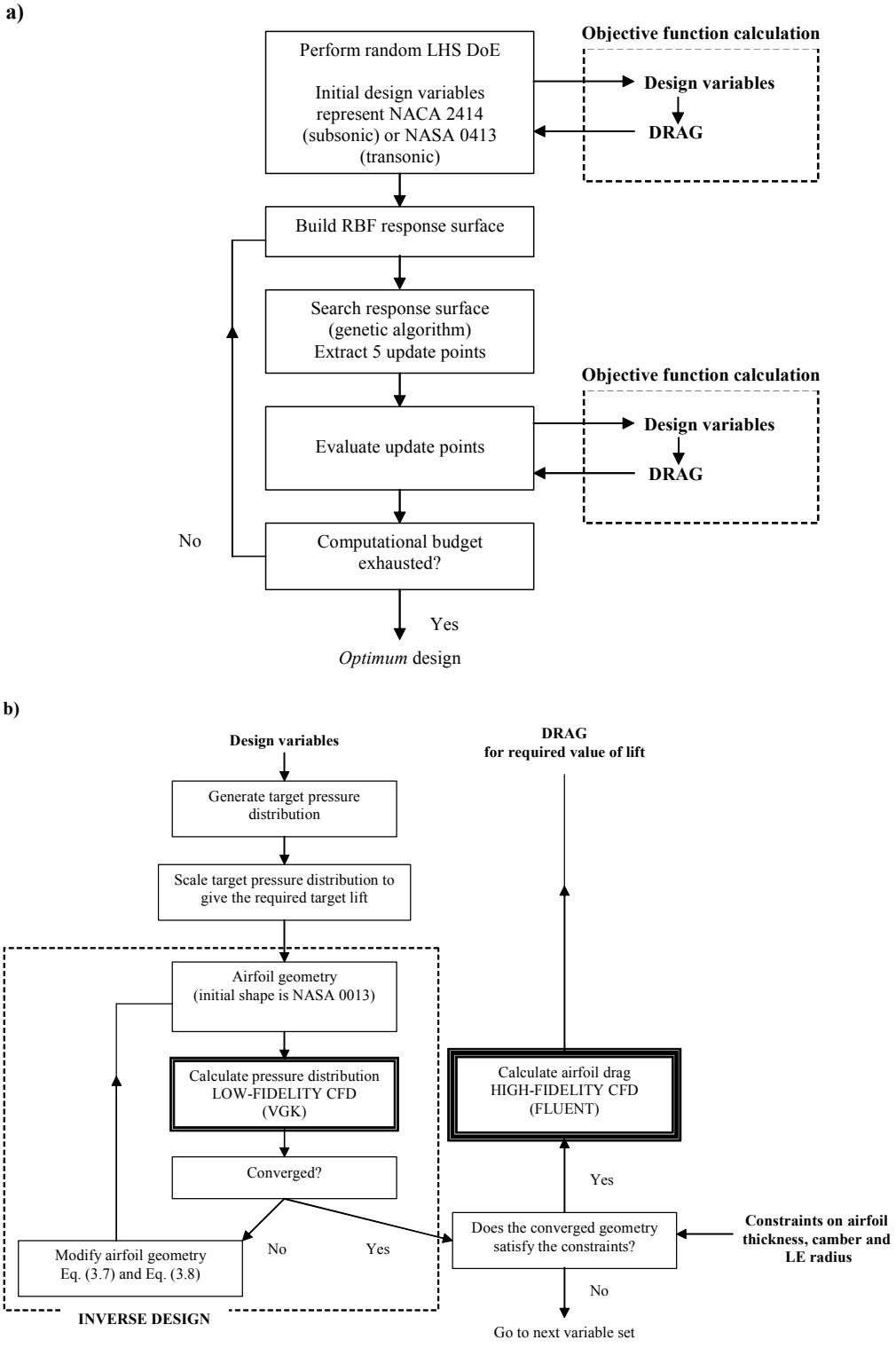


Figure 3-20 Detailed flowcharts: a) The optimization strategy used by the benchmark and EMFID methods, b) an objective function evaluation in EMFID.

3.9 Computational Expense

For the 2-D airfoil design work detailed in the next chapter, all CFD simulations and optimization are performed in the Windows environment on a cluster running Microsoft Windows Server 2003 and Microsoft Compute Cluster Server. This cluster consists of 2- and 4-processor nodes. All CPUs are *dual core* AMD Opterons, meaning that each node effectively has 4 or 8 processors, respectively. These nodes have 8Gb and 32Gb memory available, respectively, and all CPUs run at 2.41GHz clock speed. All the nodes operate with a 64 bit hardware/software interface and feature a 1 gigabit interconnect for the purposes of parallel computations. The FLUENT code includes in-built message passing interface (MPI) software, allowing efficient communication between CPUs during these parallel jobs. The computational mesh is split into a number of zones, with each CPU being assigned a different zone.

The two design methods described above are run with equal amounts of computational effort; the one which is able to reach a higher level of design performance is deemed to be the more efficient method. A single FLUENT drag calculation takes on average 19.6 minutes and 22.6 minutes for the subsonic and the transonic analyses, respectively, when running on a four-processor compute node. Since the time taken to build and search the response surface approximation is in the order of seconds, and this expense is encountered by both design methods, it is not considered in the allocated computational budget. Therefore, the time for an objective function evaluation in the benchmark process is simply equivalent to the expense of the FLUENT drag calculation, i.e., 19.6 or 22.6 minutes. An objective function evaluation for the EMFID method uses this effort plus the effort required in the inverse design step. Each VGK evaluation requires approximately 2 seconds. An average of 47 iterations are used in the subsonic inverse design process, and therefore the total objective function call demands 21.1 minutes computational time. Thus, the ratio of computational expense for the two methods at subsonic flow conditions is 1:1.078 benchmark to EMFID evaluations. Similarly, the transonic inverse design uses on average 55 iterations, resulting in an objective function time of 24.9 minutes and a ratio of computational expense of 1:1.099 for the transonic benchmark and EMFID evaluations. While, in both cases, an EMFID evaluation is only fractionally more expensive than a single FLUENT call, this can make a significant difference over the course of a full design search run. Nonetheless, due to the lower dimensionality in the EMFID search, fewer objective function calls should be required to minimize the drag. The assessment of the EMFID strategy relative to the benchmark method is the subject of the next chapter.

Chapter 4. Application of EMFID: Case Studies for 2-D Airfoil Design

4.1 Introduction

The setup of the EMFID and benchmark design search methods for the 2-D airfoil design problem has been described in full in the preceding chapter. These methods differ in their approach with respect to the parameterization of the airfoil. In EMFID a parametric B-spline model defines a target C_p distribution and inverse design is used to determine the airfoil shape, while for the benchmark process the airfoil geometry is discretized directly using a segmented spline approach. Each method employs the same high-fidelity objective function calculation and optimization algorithm. The purpose of the current chapter is to investigate the performance of these methods in minimizing the drag of an airfoil, for an equal amount of computational expense. A number of case studies are presented which are intended to demonstrate that the EMFID process can be effective for aerodynamic design problems.

The case studies begin by considering airfoil design under wholly subsonic flow conditions, specifically a Mach number of 0.15 and Reynolds number 4×10^6. In Case 1, the benchmark model is one which is thought to be typical of a parameterization used in a true aircraft design scenario; this uses 13 design variables. The subsonic EMFID parameterization is applied, which uses six variables. Case 2 adds to the results of Case 1 by considering a benchmark model which adopts the same number of variables as used in EMFID, i.e. six.

In case study 3, the objective function is calculated in transonic flow conditions, specifically a Mach number of 0.73 and Reynolds number of 6.5×10^6. Use of the six-variable subsonic pressure profile model is maintained, and hence inverse design is performed under subsonic conditions. Comparisons are drawn with the 13-variable benchmark parameterization. Finally, in Case 4, a parameterization of the pressure profile which permits the manipulation of the shock details is used in EMFID, and hence both the objective function and the inverse design are calculated under the same transonic conditions.

In each case study, key results are listed in the form of figures and tables. These results are discussed in detail and the conclusions pertinent to the EMFID concept are examined.

4.2 Case 1

In this case study, the five runs of the two methods are run side-by-side for the design of airfoils in subsonic flow conditions, using a starting design equivalent to the NACA 2414 airfoil as described in the previous chapter. Initially, the parameterizations employed are the six-variable subsonic B-spline model for EMFID, and the 13-variable cubic spline method for the benchmark process. The benchmark method is given a *budget* of 300 calls to the high-fidelity CFD code, which equates to 278 objective function evaluations in EMFID after taking into account the effort required for the inverse design. Based on the recommendation of Jones et al. [1998], the number of points in the initial DoE should be ten times the number of design variables. Hence, for the benchmark method this gives 130 design points, and for the EMFID method this requires 60 points. The methods use their remaining budget to evaluate designs during the update process, giving 170 update points for the benchmark method and 218 update points in EMFID. The design constraints on airfoil shape maximum thickness, maximum camber, leading edge radius and total lift, respectively, are:

$$\begin{aligned} 12.5\% \le z_{t\,max}/c \le 15\%, \\ 0\% \le z_{c\,max}/c \le 2.5\%, \\ r_{LE\ initial} \le r_{LE}, \\ c_l = 0.4. \end{aligned} \tag{4.1}$$

In these constraints $r_{LE\ initial}$ is the leading edge radius of the initial shape in the design process, i.e., the NACA 2414 airfoil. The constraints are arranged with relatively narrow ranges; experience has shown that if these ranges are large the convergence of the optimization is slow. The selected constraints allow both design methods to converge more quickly, and result in low-drag designs with conventional camber and thickness attributes. In addition to the above, a constraint is applied which states that the airfoil thickness at 95% chord shall be no smaller than 0.6% chord. As already noted, the target lift and flow conditions are the same as those used by McGhee et al. [1979], allowing comparisons to be made with the NASA low-speed airfoils.

The optimization-iteration histories for the two methods are shown in Figure 4-1. The progress of the optimizer in each case is plotted against the number of benchmark iterations; the number of iterations of EMFID has been scaled in this plot such that the x-axis can be interpreted as equal computational expense. The design objective is drag, which has been normalized by multiplying by the ratio of lift output from the CFD code to the target lift; this removes the numerical error generated if the airfoil c_l is not exactly 0.4. From Figure 4-1 it is seen that for a given computational cost, the EMFID method finds a better design performance than the benchmark search method. The rate of convergence towards the optimum is faster for the EMFID method. This is because EMFID is able to find significantly better designs in its initial DoE, and this in turn is due to the reduced problem dimensionality. Conversely, the benchmark method does not find improved designs in the DoE evaluation stage. While neither method has provided convergence onto a single optimum design, the five runs of the EMFID method show a greater level of convergence, indicated by the reduced range of objective values at the end of the search process.

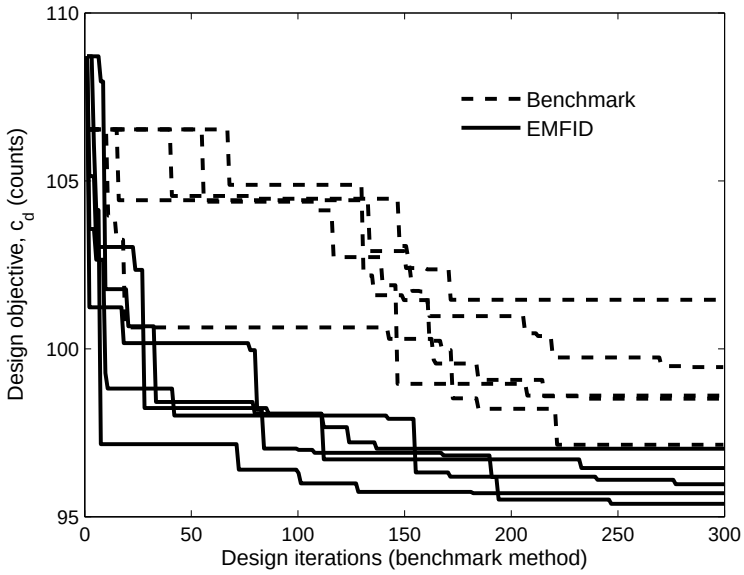


Figure 4-1 The five optimization histories for the benchmark and EMFID methods.

The geometry which was found to give the best performance for each of the five runs of the benchmark procedure is shown in Figure 4-2. It is clear that each initial DoE set has given a very different final result for the computational budget used. The designs in this case are visibly not converged, with most of these shapes featuring an undesirable *rippling*. This unconverged state is possibly a result of the optimization procedure locating many local optima, and this is a symptom of the high dimensional design space. These erratic surfaces are also an artefact of the very low Mach number (M=0.15): the ripples have a small drag

penalty at this flow speed. Table 4-1 provides data for these geometries. Despite the apparent differences, all the designs are made to be around 12-13% thickness.

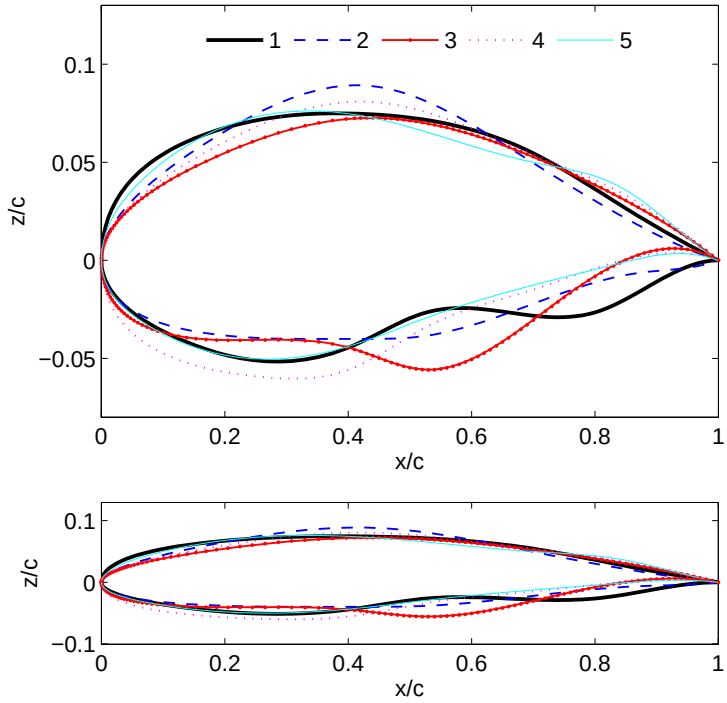


Figure 4-2 Final five geometries generated by the benchmark method. The lower figure shows the airfoils on equally scaled axes.

Table 4-1 Airfoil design data for the best geometries resulting from the five benchmark design searches: maximum thickness, maximum camber, angle of attack, lift coefficient and drag coefficient.

Airfoil design		$z_{t\,max}/c$ (%)	$z_{c\,max}/c$ (%)	α (°)	c_l	c_d (*counts*)
Best from run	1	12.55	2.27	2.01	0.4000	101.5
	2	12.96	2.46	1.95	0.4005	98.5
	3	12.62	1.55	0.88	0.4001	98.6
	4	13.85	2.16	0.59	0.4000	99.5
	5	12.59	1.93	0.46	0.4000	97.1
NACA 2414 (initial design)		14.00	2.00	1.63	0.3999	104.6
NASA LS(1)-0013		12.63	0.00	3.49	0.4000	104.7
NASA LS(1)-0413		12.72	2.32	-0.37	0.3981	98.4

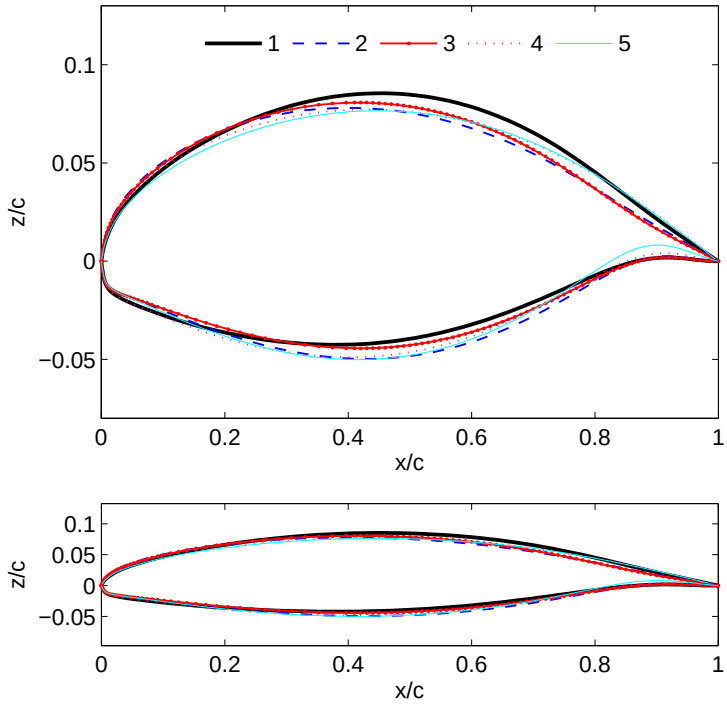


Figure 4-3 Final five geometries generated by the EMFID method.

Table 4-2 Airfoil design data for the best geometries resulting from the five EMFID design searches: maximum thickness, maximum camber, angle of attack, lift coefficient and drag coefficient.

Airfoil design		$z_{t\,max}/c$ (%)	$z_{c\,max}/c$ (%)	α (°)	c_l	c_d (*counts*)
Best from run	1	12.77	2.32	0.57	0.4001	96.0
	2	12.79	2.45	1.21	0.4000	97.0
	3	12.53	1.82	1.11	0.4003	96.5
	4	12.63	1.82	0.64	0.4000	95.7
	5	12.68	1.97	0.41	0.4001	95.4
NACA 2414 (initial design)		14.00	2.00	1.63	0.3999	104.6
NASA LS(1)-0013		12.63	0.00	3.49	0.4000	104.7
NASA LS(1)-0413		12.72	2.32	-0.37	0.3981	98.4

The best geometries from the five computations using the EMFID method are shown in Figure 4-3. The more advanced state of convergence is made clear by the greater similarity in the five geometries than the benchmark method achieved, and this is possible because of the reduced dimensionality of the design

space in EMFID. It is immediately apparent that these shapes are smoother than the benchmark designs, and this is inherent in the parameterization used. However, although the description of the C_p distribution is simple, the resulting geometry after inverse design can exhibit significant detail and complexity. Table 4-2 lists the airfoil design data for the five geometries resulting from the EMFID method. Crucially, and rather predictably from looking at the airfoil shapes, the drag for the EMFID designed shapes is lower than those from the benchmark search method. In all five cases the maximum thickness is made to be almost as small as the constraints allow (12.5%), and this is to be expected since the objective is to minimize drag. The maximum camber of the EMFID designs is also very similar. The lower problem dimensionality in EMFID has allowed the optimization procedure to explore promising areas of the design space more thoroughly than is possible in the benchmark method for the same computational effort.

Figure 4-4 compares the best two geometries from the five benchmark and EMFID searches. Two NASA low-speed airfoils are also shown: the initial design in the search process, NASA LS(1)-0413, which is 12.95% thick, and NASA LS(1)-0013, with a thickness of 12.84%. The 0413 airfoil has 2.2% camber and is designed to give a c_l of 0.4 at 0° angle of attack; the 0013 airfoil has zero camber and gives a c_l of 0.4 at approximately 3.5° angle of attack. The two best airfoils resulting from the benchmark and EMFID methods are 12.59% and 12.68% thick with maximum cambers of 1.93% and 1.97%, respectively. The NASA 0413 airfoil features the highly cambered aft lower surface which is a characteristic of many supercritical airfoils (the LS series were developed by NASA in parallel with the supercritical designs). The best EMFID design clearly adopts this feature, exaggerating it further compared to the 0413 shape. The benchmark method also identifies the feature as beneficial, but incorporates it to a lesser extent due to the early state of convergence. The upper surface of the benchmark shape exhibits the undesirable rippling, associated with the incomplete convergence of the high dimensional search space. Both designed airfoils closely resemble the NASA 0413 shape at the aft upper surface, giving very similar trailing edge angles. Note also that the NASA 0013 upper surface features an inflection point which, according to Ferris et al. [1987], gives reduced drag. Although the best EMFID airfoil does not have this characteristic, it is seen on two of the other final geometries in Figure 4-3.

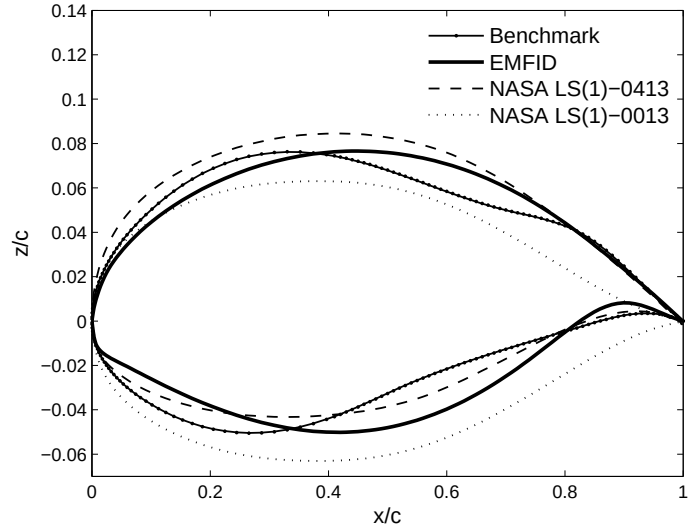
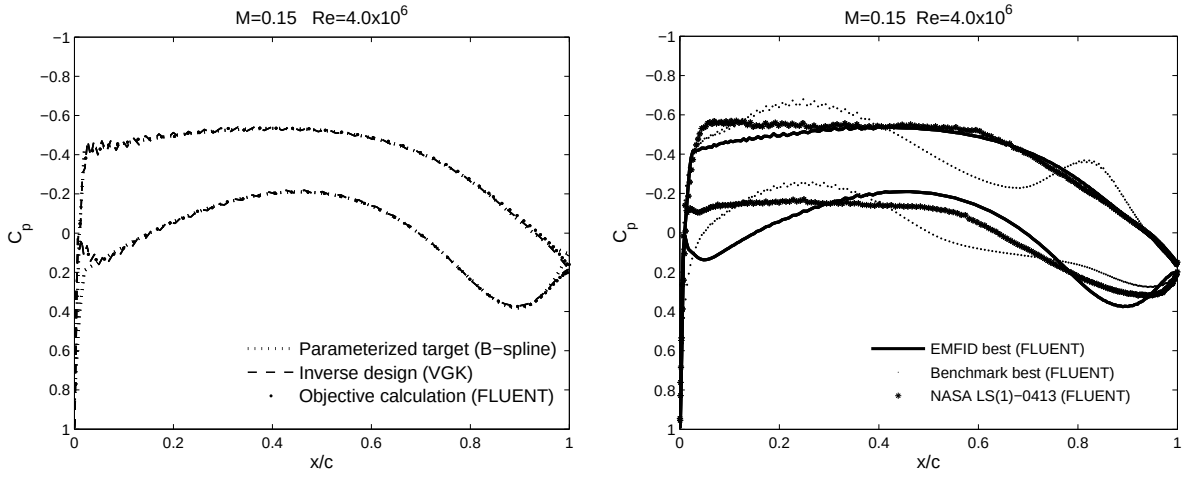


Figure 4-4 Comparison of the best performing geometry from each of the two methods, shown with two NASA low-speed airfoils of 13% thickness.



(a) **(b)**

Figure 4-5 (a) The *optimized* target pressure distribution, profile achieved during inverse design and profile output from FLUENT. (b) Pressure distributions due to the best performing EMFID and benchmark geometry, and the NASA LS(1)-0413 shape.

Figure 4-5 (a) shows the best EMFID target pressure distribution, together with the converged inverse design result and the pressure distribution generated when the corresponding (best) airfoil is analysed using FLUENT. Figure 4-5 (b) compares the C_p distributions for the best benchmark and EMFID airfoils (i.e., those in Figure 4-4) with the NASA 0413 C_p distribution, all computed using FLUENT. Note the erratic profile of the benchmark airfoil, an effect of the aforementioned rippling. The harsh pressure

gradients may result in separation at higher Mach numbers. Note also that the pressure at the aft upper surface of the EMFID design closely follows that of the NASA shape.

It is also useful to compare the performance of the designed airfoils in a more general sense. Figure 4-6 illustrates this as a lift-drag polar plot, showing the best designs from the benchmark and EMFID processes along with plots for the NASA airfoils 0013 and 0413. Of the NASA examples, the 0013 gives the lowest drag; this is at zero angle of attack since it is symmetrical. However, the 0413 airfoil gives lower drag at values of c_l greater than around 0.2, due to its improved lifting performance. The airfoils designed by the benchmark and EMFID methods give lower drag at their design lift coefficient, 0.4, but their drag increases more rapidly with lift compared to the NASA airfoils. Unsurprisingly, the designed airfoils have become specialized for c_l =0.4. Figure 4-7 shows lift plotted against angle of attack, and further illustrates this design point dependence. While the EMFID airfoil has been shown to give the lowest drag at c_l =0.4, it has a rather low maximum lift capability and exhibits very early stall compared to the other airfoils. The NASA 0413 shape has the highest camber line and therefore a higher lift for a given angle of attack, but it also performs well at high angles of attack and has the highest maximum lift. These observations are an artifact of the design objective used: the NASA airfoils are designed to perform well over a range of angles, whereas the geometries generated here are designed to minimize drag at a single angle of attack (or a single c_l). It is not the purpose of this work to improve on the NASA airfoils; rather it is to demonstrate the use of the EMFID method for aerodynamic design.

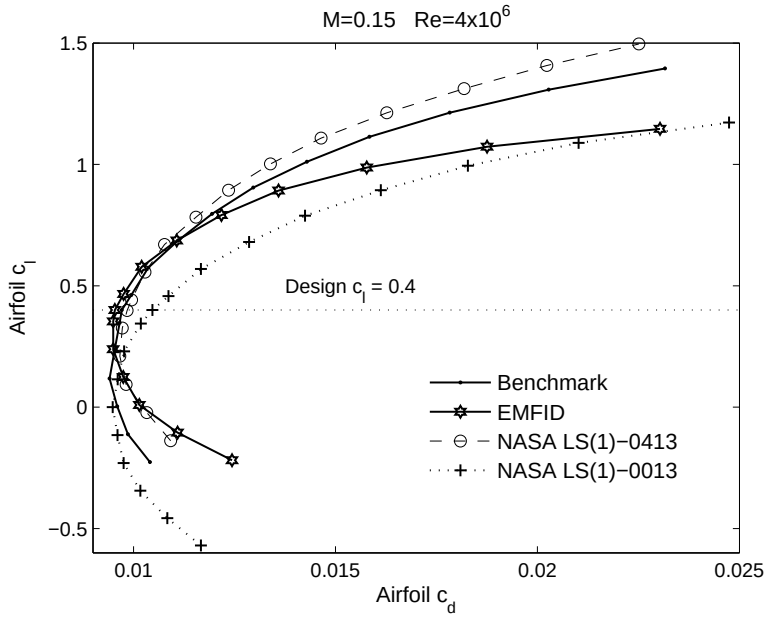


Figure 4-6 Lift-drag polar plot for the best designs from the benchmark and EMFID methods, shown with FLUENT results for two NASA airfoils.

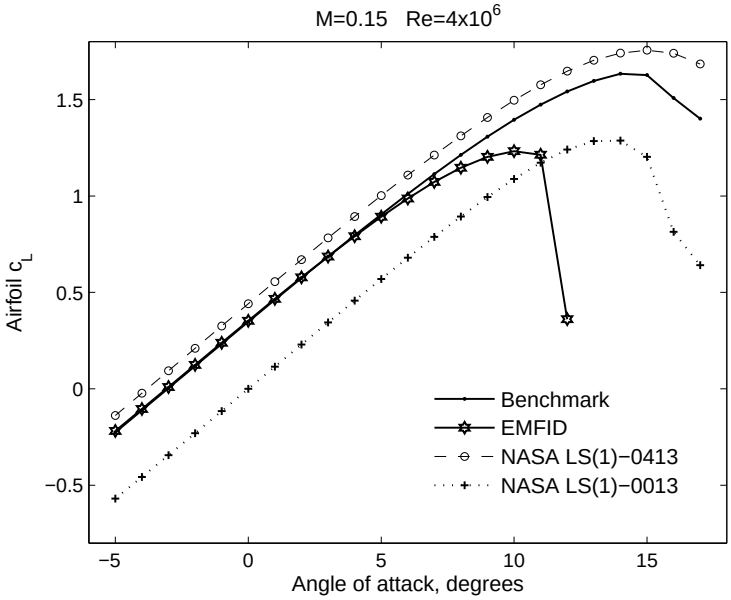


Figure 4-7 Lift vs. angle of attack for the best designs from the benchmark and EMFID methods, shown with FLUENT results for two NASA airfoils.

4.3 Case 2

The previous case study has shown that the EMFID method is able to find better designs in its initial DoE, and converge more quickly, compared to the benchmark method. These performance gains are due to the reduced dimensionality in the EMFID method, i.e., it is only required to search a six-dimensional design space compared to the 13-dimensional space for the benchmark method. In addition to this, because the EMFID DoE consists of a smaller number of objective evaluations (60 compared to 130), it is able to perform more evaluations during the RSM update stage than the benchmark process, for a given computational budget. The update phase of the design process is then clearly more likely to discover good designs, provided the design space is sufficiently populated to give accurate RSM predictions. These matters lead to the question: how much better would the benchmark process perform if its parameterization has the same dimensionality as the EMFID parameterization? This is the subject of the case study considered here. The benchmark method is run again, but in this case the six-variable B-spline model of the airfoil is used (Section 3.6.1), rather than the 13-variable model used in case study 1. It is desirable to compare the performance of the benchmark method, using this new parameterization, with the results from the first case study. For this reason, the design search is set up in the same way as previously discussed. In particular, the initial design (equivalent to NACA 2414), the constraints on

thickness, camber and leading edge radius, and the objective function calculation are all made to be the same. Again, five design searches are performed with different DoE seeds, but in this case there are 60 points in each DoE.

The optimization-iteration histories for the benchmark method in this configuration are shown in Figure 4-8; for comparative purposes, the traces for the EMFID and benchmark methods obtained in case study 1 are also shown. It can be seen that, compared to the 13-variable benchmark optimization, the method in this case has initially resulted in faster convergence, due to the reduced dimensionality of the design space. However, the traces for the 13-variable benchmark runs *overtake* the six-variable optimizations after 150-200 objective evaluations. For the simple airfoil parameterization employed here, it is difficult to achieve a drag force under 100 counts. The reduction in the number of degrees of freedom of the airfoil shape means that the detailed changes allowed by the 13-variable representation are no longer possible. This restricts the optimization process to the extent that it cannot provide the design improvement that is possible with the other parameterizations. Positively, the level of convergence achieved by the five searches is good, in fact this is better than the EMFID method attained; the region containing good designs in this case is clearly efficiently searched by the optimizer.

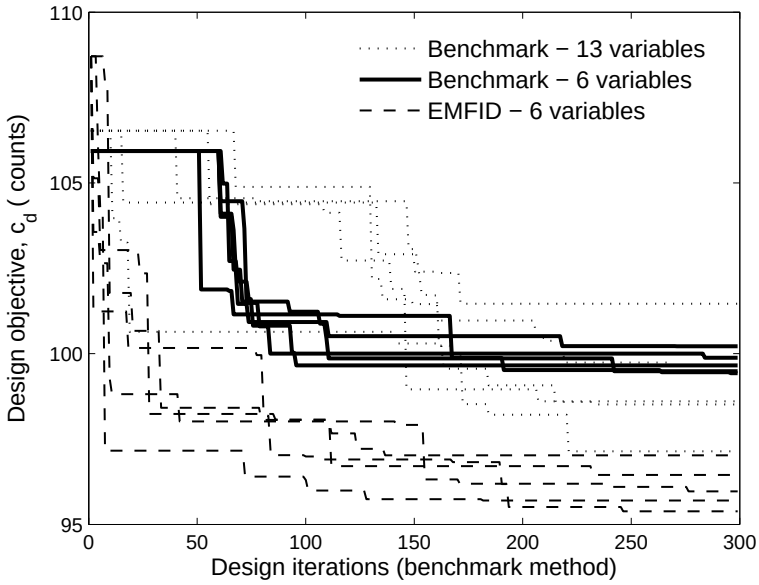


Figure 4-8 The five optimization histories for the benchmark method using the six-variable parameterization.

The geometries resulting from the five runs of the six-variable benchmark method are shown in Figure 4-9. These shapes are closely aligned, further emphasizing the relatively successful convergence of this

process. The airfoil shapes are smooth contours and do not exhibit the rippling seen in the previous benchmark geometries. However, this airfoil description cannot perform such localized changes. Thus, an efficient scheme might be to use the six-variable model for the DoE seed evaluations, map these points into the 13-variable space, and then perform the update process using the more detailed 13-variable parameterization. Such hierarchical systems are used routinely for industrial DSO problems (Keane and Nair [2005]). Table 4-3 lists the design data for the airfoils resulting from this case study. Curiously, the maximum thickness is not minimized to the 12.5% lower problem bound, as occurred with the previous parameterizations. This is due to the limited flexibility of this geometry model, and it is likely that the benefit of other curvature changes is greater than the penalty due to not minimizing the thickness.

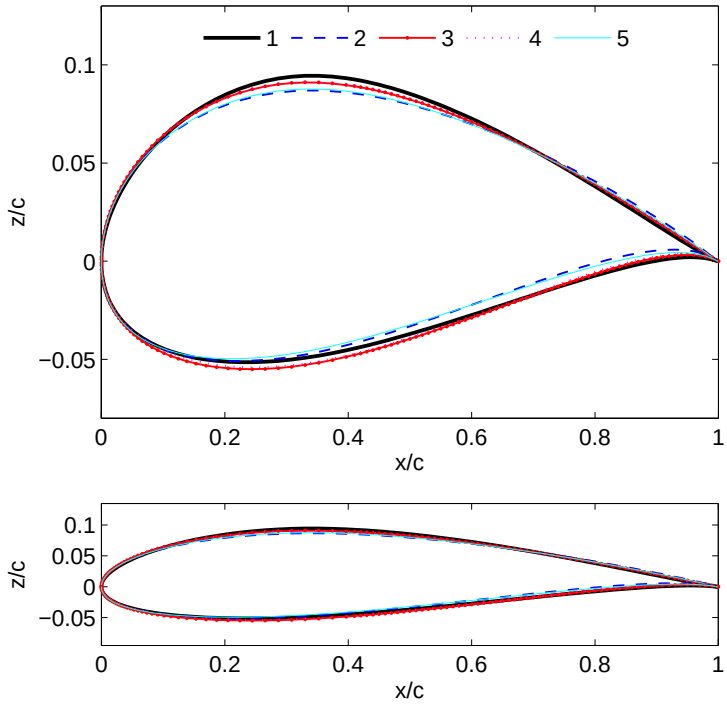


Figure 4-9 Final five geometries generated by the six-variable benchmark method.

Table 4-3 Airfoil design data for the five geometries resulting from the six-variable benchmark design search: maximum thickness, maximum camber, angle of attack, lift coefficient and drag coefficient.

Airfoil design		$z_{t\,max}/c$ (%)	$z_{c\,max}/c$ (%)	α (°)	c_l	c_d (*counts*)
Best from run	1	14.43	2.43	0.62	0.4002	99.7
	2	13.57	2.38	-0.13	0.4000	99.4
	3	14.47	2.15	0.46	0.4002	100.2
	4	14.40	2.18	0.36	0.4001	99.9
	5	13.50	2.40	0.09	0.4002	99.5

4.4 Case 3

It has been shown that a parameterization of the airfoil subsonic C_p distribution can result in an efficient design search process, and while the parametric description is simple it is able to describe detailed changes in shape that produce high quality airfoil designs. Although this is a useful result in its own right, the natural extension is to apply the EMFID method to the design of airfoils in the more demanding transonic flow regime. Modern civil aircraft wings are required to operate in transonic conditions, and drag reduction is a major consideration in the design process; this is therefore a more useful application for any airfoil design method. In such cases, a shock is typically manifested in the airfoil C_p distribution as a step increase in pressure on the upper surface; representation of the C_p profile using simple functions is therefore more difficult for transonic flows. Since the position and strength of this shock can vary, this could require two or more additional design variables for the representation, whereas the geometry description could remain unaltered. A reduction in design variables is the *raison d'être* of the EMFID process, and to use a complex and high-dimensional description of the C_p profile may nullify any benefit of designing in the flow feature domain.

It has previously been shown that there is good agreement between the parameterized C_p profile and the profile obtained during the high-fidelity objective calculation (see, for example, Figure 3-10 and Figure 4-5 (a)). However, the question asked in this case study is: how does the EMFID process perform if the parametric description of the airfoil is for subsonic conditions, while the objective function (drag) is calculated for the transonic regime? In this case, EMFID no longer involves a parameterization of the flow features of the intended design *per se*. Rather, the C_p profile model becomes simply a means of arriving at an airfoil shape using a small number of variables, but by using the model in this way the direct mapping between the parametric model and the *real* flow features is lost. The current case study investigates the relative performance of the EMFID method when used in this way, compared to the benchmark design method.

In this case study, the transonic FLUENT setup is used for the high-fidelity CFD (Section 3.5.4), i.e., specifying a flow speed Mach number of 0.73 and Reynolds number 6.5×10^6. This relatively modest Mach number is used as increasing this value reduces the robustness of the low-fidelity code (VGK), in terms of providing a converged solution. Therefore, in order to ensure that the flow is transonic and features a shock, a relatively high target lift of c_l=0.8 is specified for this FLUENT drag calculation. The benchmark and EMFID methods are run with the parameterizations that were used for case study 1. The benchmark process uses a 13-variable cubic spline model of the airfoil shape. EMFID uses the parameterization of the subsonic pressure distribution, and hence the subsonic inverse design process

from Case 1 is utilized, i.e., the VGK code is run at M=0.15. Each method is run five times using a different DoE seed, in each case starting from a representation of the same design, NASA LS(1)-0413. As in the first case study, the benchmark method uses 130 DoE points and 170 update points. However, the EMFID method uses 60 DoE points with 221 update points; three more objective calculations can be afforded in the design search since a slightly more expensive FLUENT drag analysis is now employed. The design constraints on airfoil shape maximum thickness, maximum camber, leading edge radius and total lift, respectively, are:

$$\begin{aligned} 12.5\% \le z_{t\,max}/c \le 15\%, \\ 0\% \le z_{c\,max}/c \le 2.5\%, \\ r_{LE\ initial} \le r_{LE}, \\ c_l = 0.8. \end{aligned} \tag{4.2}$$

The optimization-iteration histories for this transonic case study are shown in Figure 4-10. Again, this is a plot of the design objective against the number of benchmark iterations, where the number of EMFID iterations has been scaled to allow a comparison for equal computational expense. The result is similar to the plot obtained in the first case study. The EMFID design searches are able to find significantly improved designs during the DoE evaluation phase, allowing them to converge more quickly. The reduced dimensionality again results in a greater level of convergence by the end of the search process, indicated by the smaller range of objective values. However, in this case the best achieving benchmark search yields an airfoil with lower drag than four of the five EMFID computations; i.e., the traces show a degree of *overlap*. It appears that in this transonic case the EMFID process does not exhibit the efficiency gains over the benchmark method that it did in the subsonic case study.

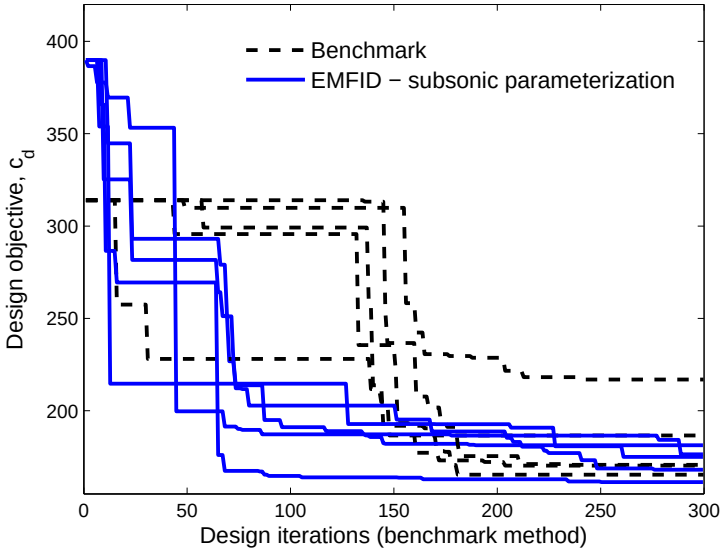


Figure 4-10 **The five optimization histories for the benchmark and EMFID methods, for transonic airfoil design.**

Figure 4-11 shows the five airfoil designs generated by the benchmark process for the transonic flow conditions. As observed in case study 1, the shapes here are clearly not converged onto a single best design. However, the undesirable rippling, seen on the airfoils designed for subsonic flow (Figure 4-2), is not as prominent on these designs, i.e., the airfoils are smoother. At the present high Mach number of 0.73, the flow is more prone to separation aft of any ripples than at Mach 0.15. Hence, there is a much higher drag penalty associated with the ripples in this case, and this explains why the optimization process has avoided such geometries. Figure 4-12 presents the best geometries generated by the five EMFID computations. While these shapes are all smooth, the level of similarity, i.e., convergence, between them is not as good as reported for case study 1 (see Figure 4-3). This is particularly true in the region of the aft lower surface, and this is known to be an important region in the design of supercritical airfoils. Again, it appears that the advantage of parameterizing the surface pressure profile is less evident in this transonic case. Table 4-4 gives the airfoil design data for the five best benchmark designs, and Table 4-5 lists this data for the best EMFID designs. Both sets of design searches have clearly associated thinner airfoils with reduced drag, and have minimized this thickness in each case.

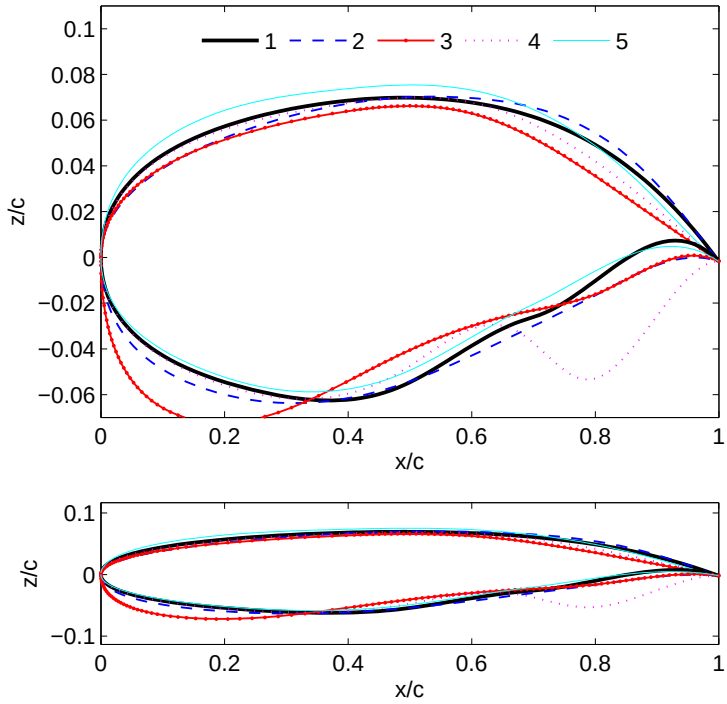


Figure 4-11 Final five geometries generated by the benchmark method, for transonic airfoil design.

Table 4-4 Airfoil design data for the five geometries resulting from the transonic benchmark design search: maximum thickness, maximum camber, angle of attack, lift coefficient and drag coefficient.

Airfoil design		$z_{t\,max}/c$ (%)	$z_{c\,max}/c$ (%)	α (°)	c_l	c_d (*counts*)
Best from run	1	13.10	2.00	1.59	0.7994	165.4
	2	12.90	1.90	3.08	0.8216	186.6
	3	12.64	1.66	2.64	0.8020	170.3
	4	12.75	1.84	2.30	0.8004	216.9
	5	13.17	2.22	1.40	0.8051	170.7
NASA LS(1)-0413 (initial design)		12.72	2.32	1.55	0.7981	304.3
NASA SC(2)-0712		11.76	2.21	1.31	0.8004	161.8
RAE 2822		12.10	1.27	2.95	0.8012	201.4

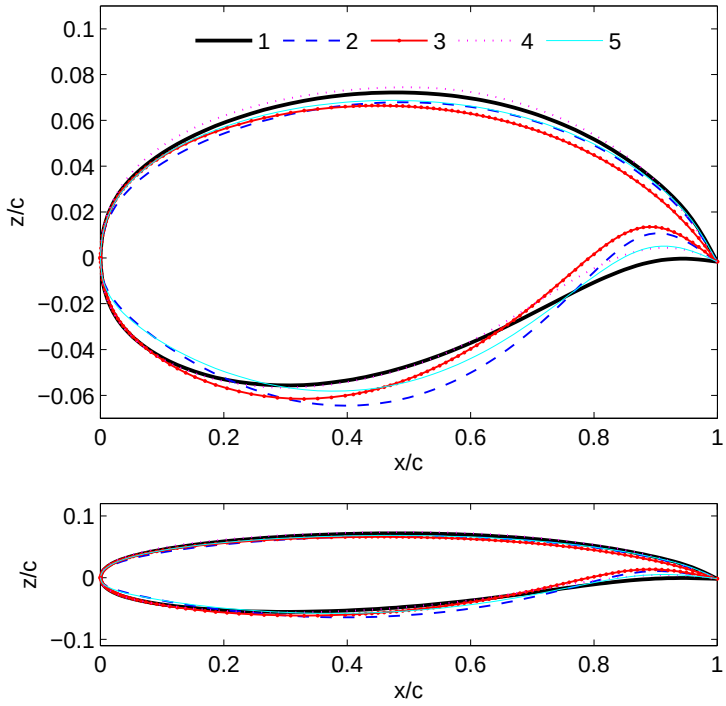


Figure 4-12 Final five geometries resulting from the transonic EMFID method.

Table 4-5 Airfoil design data for the five geometries resulting from the transonic EMFID design search: maximum thickness, maximum camber, angle of attack, lift coefficient and drag coefficient.

Airfoil design		$z_{t\,max}/c$ (%)	$z_{c\,max}/c$ (%)	α (°)	c_l	c_d (*counts*)
Best from run	1	12.50	2.06	2.33	0.8001	181.3
	2	13.17	2.34	1.97	0.8021	168.1
	3	12.66	2.40	1.32	0.7994	161.3
	4	12.79	2.46	1.57	0.7951	175.0
	5	12.63	2.14	2.02	0.8043	176.4
NASA LS(1)-0413 (initial design)		12.72	2.32	1.55	0.7981	304.3
NASA SC(2)-0712		11.76	2.21	1.31	0.8004	161.8
RAE 2822		12.10	1.27	2.95	0.8012	201.4

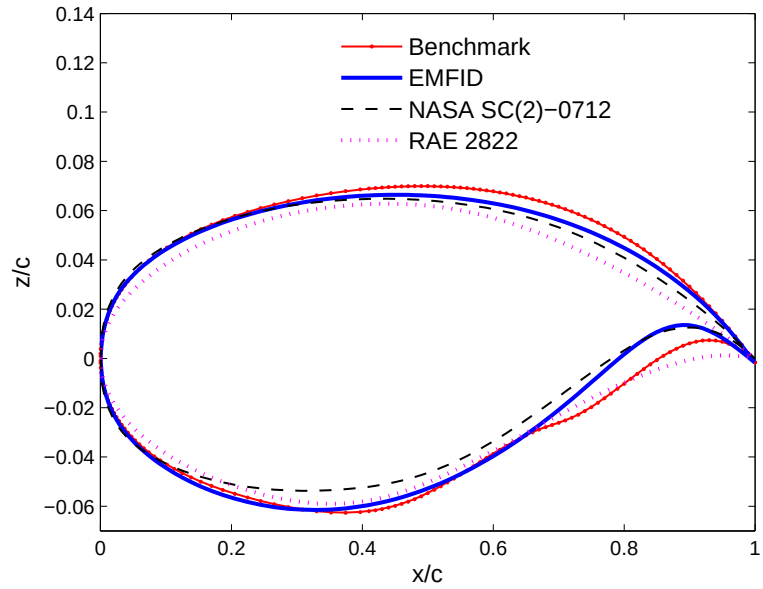
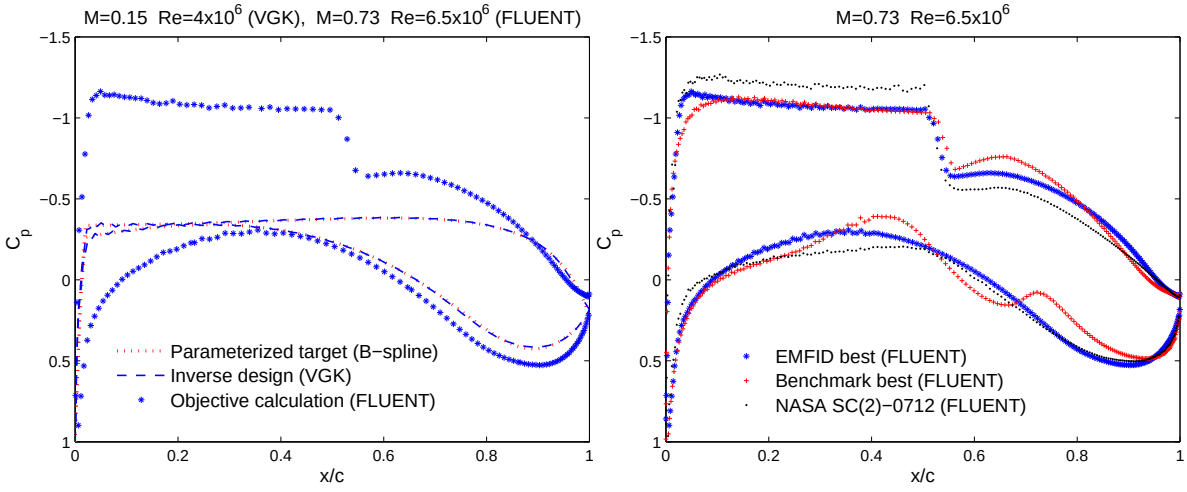


Figure 4-13 **Comparison of the best performing geometry from each of the two methods, shown with supercritical airfoils NASA SC(2)-0712 and RAE 2822.**



(a) **(b)**

Figure 4-14 **(a) The *optimized* target pressure distribution , profile achieved during inverse design and profile output from FLUENT. (b) Pressure distributions due to the best performing EMFID and benchmark geometry, and the NASA SC(2)-0712 shape.**

It is useful to compare the results of this transonic design study against existing established transonic shapes. Figure 4-13 compares the best design from the five runs of the benchmark and EMFID searches, with the supercritical airfoil NASA SC(2)-0712 published by Harris [1990] and the RAE 2822 airfoil used by Cook et al. [1979]. Data for these standard airfoils are given in Table 4-5. Note that, although their thicknesses are slightly smaller than is permitted by the constraint imposed in the EMFID and benchmark design methods (12.5%), these standard airfoils have similar attributes to the designed shapes, and are intended to operate under similar flow conditions. The benchmark designed airfoil features a raised upper surface, with a lower surface which closely follows that of the RAE 2822 shape. The EMFID optimization process has arranged the design variables defining the subsonic C_p profile, such that, following inverse design at subsonic conditions, the resulting airfoil is very similar to the NASA 0712 shape, and performs very well when subjected to a high speed flow. The main difference between these designs is the thickness (12.66% and 11.76%, respectively), see Table 4-5. Figure 4-14 (a) illustrates the EMFID parameterization, showing the optimized parametric C_p distribution, converged inverse design result, and C_p distribution obtained on running the objective function. This demonstrates the distinct difference between the parameterized and *real* flow features. Figure 4-14 (b) shows the same C_p data for the best EMFID airfoil compared with the data for the best benchmark design and NASA 0712. Unsurprisingly, the EMFID and NASA 0712 profiles are closely matched. The designed airfoils have reduced the shock strength for the present flow conditions, compared to the NASA 0712 shape, although it is interesting to note that the shock position is the same in all three cases. The slight rippling on the lower surface of the benchmark shape is reflected in the C_p profile. As previously discussed, this lack of smoothness is likely to be associated with the unconverged state of the optimization process, which in turn is an artefact of the parameterization used. In addition to this, the benchmark method is disadvantaged since the designed airfoils have surfaces which are, mathematically, cubic polynomials in form. The position and strength of shocks is extremely sensitive to the local curvature, and since each segment of the airfoil has a linearly varying curvature, the ability of the benchmark method to control shocks is compromised. This limitation is not as prominent in the EMFID method since the geometry description is in the form of co-ordinates which are generated using a more complex formulation for the surface curvature during inverse design.

This case study has established that the EMFID approach can be effective for the design of airfoils for transonic conditions. The five EMFID computations have all produced high performing airfoils, and the small spread of objective values in Figure 4-10 indicates more advanced convergence than the benchmark process achieved. However, the advantage is less convincing than was reported in Case 1, and although it is apparently more likely for a given design search using EMFID to produce a better airfoil, it has been

shown that the benchmark process may produce a superior design. The EMFID approach uses a transonic drag calculation, while the parameterization of the C_p distribution is subsonic. It is inferred that, because of the loss of the direct mapping between flow features, the shift in complexity brought about by the transonic drag calculation is felt more by EMFID than for the benchmark process, hindering the convergence of the EMFID optimization. The next step is to implement the parameterization of the transonic C_p profile, i.e., the profile containing a shock. In this way, EMFID again considers a parametric model of the true flow features. This investigation is the subject of the final 2-D case study.

4.5 Case 4

Previously, the EMFID method has made use of a parameterization of the subsonic C_p distribution. The subsonic model has the advantage of requiring the specification of only six design variables; it is therefore compact and has been shown to be a very effective representation for subsonic airfoil design. However, such a model is less effective when designing airfoils for transonic flow conditions. This result is not unexpected, since the high Mach number induces a shock which is not accounted for in the parameterization. The present case study reports the performance of EMFID when the parameterization includes the shock details. Although the representation requires two more design variables and involves a more complex inverse design process, the reinstatement of a parameterization of the true flow features may yield enhanced performance.

The transonic parameterization of the C_p profile uses the six-variable B-spline model with an additional two variables controlling the height and chordwise extent of the rooftop region. This eight-variable model was described in Section 3.7.2. In order to allow a comparison with the results obtained in Case 3, the EMFID search here is set up in the same way. In particular, the initial design (equivalent to NASA LS(1)-0413), the constraints on thickness, camber and leading edge radius, and the objective function calculation are all the same. In this case, the computational budget of 300 FLUENT calculations equates to 272 EMFID objective function calls. Five design searches are performed with different DoE seeds, with each search consisting of 80 points in the DoE and 192 update points.

Figure 4-15 shows the optimization-iteration histories for EMFID in this setup, overlaid onto the results from case study 3. Although the new optimization traces overlap the previous results, one of these computations finds an airfoil with lower drag than any of the previous designs. Additionally, the level of convergence reached collectively by the five runs using this new setup is significantly improved

compared to the EMFID method using a subsonic parameterization. The convergence is also faster; significantly better designs are found during the DoE evaluation phase. It is evident that using a parameterization of the surface C_p distribution is more efficient when the parametric model, inverse design, and objective calculation are all set up to capture the same flow conditions. In this case, this has required the specification of the shock details, but despite an increase in the number of design variables this configuration yields improved performance.

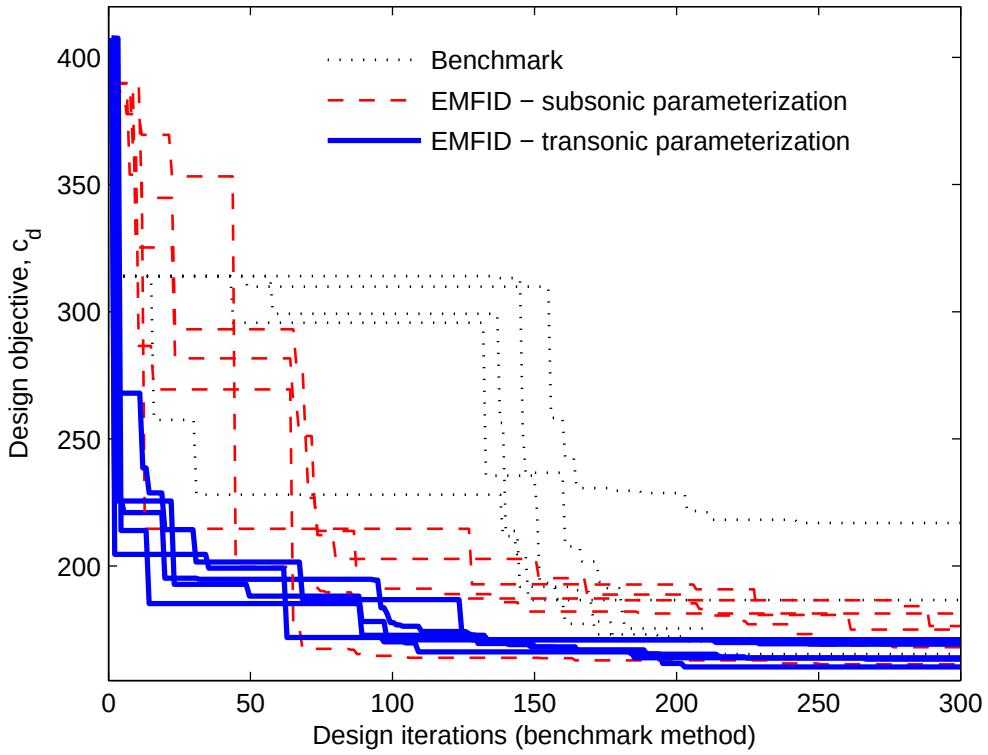


Figure 4-15 The five optimization histories for the EMFID method using a transonic C_p parameterization, shown with the results from case study 3.

Figure 4-16 shows the final five airfoils generated by the EMFID method using this setup, and Table 4-6 gives the associated data for these geometries. The advanced level of convergence achieved is clear to see; the design searches have resulted in very similar geometries. This suggests that there is a global optimum in the response surface which is relatively straightforward to find using the present optimization algorithm.

Figure 4-17 shows the best design from the EMFID searches using the present transonic parameterization, alongside the best result using the subsonic model and the NASA 0712 supercritical airfoil. The designed airfoil does not seem to follow the traditional supercritical shape of the 0712. The leading edge is similar

initially, in order to conform with the imposed constraints, but following this the lower edge has a high curvature giving it a rather pointed shape. This effect was also seen in the subsonic EMFID airfoil designed in case study 1, and suggests that the shape is specialized to operate well at the design conditions. This is in contrast to the NASA 0712 design, which has been designed in a multipoint fashion and therefore performs well over a range of operating conditions. Figure 4-18 (a) shows the target pressure distribution corresponding to the best EMFID result using the transonic parameterization, together with the converged inverse design result and the data from FLUENT for the same shape at c_l=0.8. Except for the region of the shock, these profiles closely match. The target C_p profile features a far aft shock position, suggesting that the design process is attempting to pull the shock in this direction; this is known to increase the efficiency of transonic airfoils. The inverse design process (using VGK) was unable to match this target shock position, however, but arranged the geometry such that the shock is as far aft as possible. The disparity between the VGK and FLUENT C_p profiles is due to the incapability of VGK to accurately predict shocks; this was demonstrated in Section 3.5.4. Figure 4-18 (b) shows the FLUENT data for the same best EMFID design, compared with the C_p data for the NASA 0712 airfoil and the best EMFID design using the subsonic parameterization. It is seen that the new EMFID process has successfully moved the shock further aft and reduced the shock strength slightly compared to the other airfoils, which results in reduced drag for a given lift. Thus, the EMFID method, and indeed the process of airfoil design, seems to benefit significantly from using a parameterization which includes the specification of shock details. The shock position and pressure jump are likely to be strongly linked to wave drag, and thus the EMFID approach allows a very intuitive means of reducing drag.

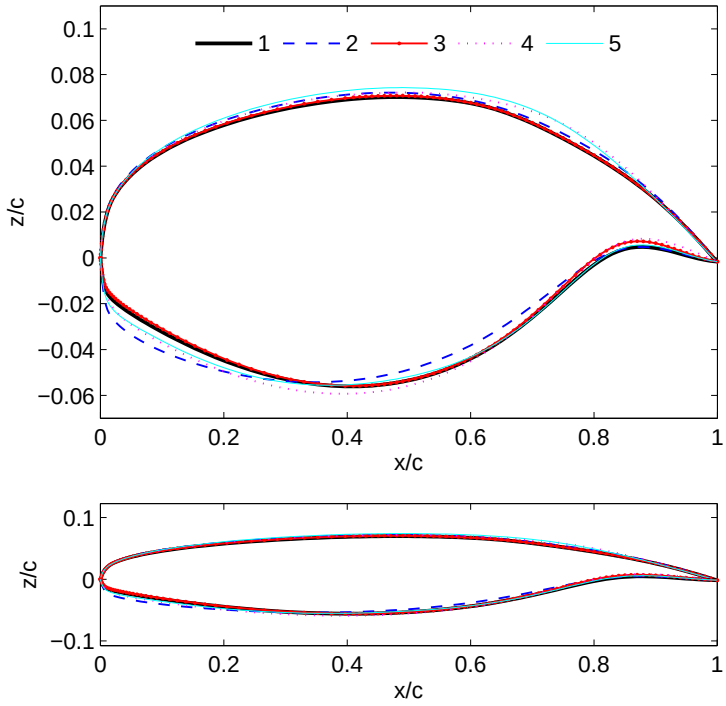


Figure 4-16 Final five geometries resulting from the eight-variable transonic EMFID method.

Table 4-6 Airfoil design data for the five geometries resulting from the eight-variable transonic EMFID design search: maximum thickness, maximum camber, angle of attack, lift coefficient and drag coefficient.

Airfoil design		$z_{t\,max}/c$ (%)	$z_{c\,max}/c$ (%)	α (°)	c_l	c_d (*counts*)
Best from run	1	12.54	2.12	2.02	0.8003	163.3
	2	12.68	2.08	1.80	0.7998	163.9
	3	12.62	2.23	1.83	0.7997	160.3
	4	13.08	2.49	1.63	0.7821	171.1
	5	12.84	2.32	1.75	0.7720	169.2
NASA LS(1)-0413 (initial design)		12.72	2.32	1.55	0.7981	304.3
NASA SC(2)-0712		11.76	2.21	1.31	0.8004	161.8
RAE 2822		12.10	1.27	2.95	0.8012	201.4

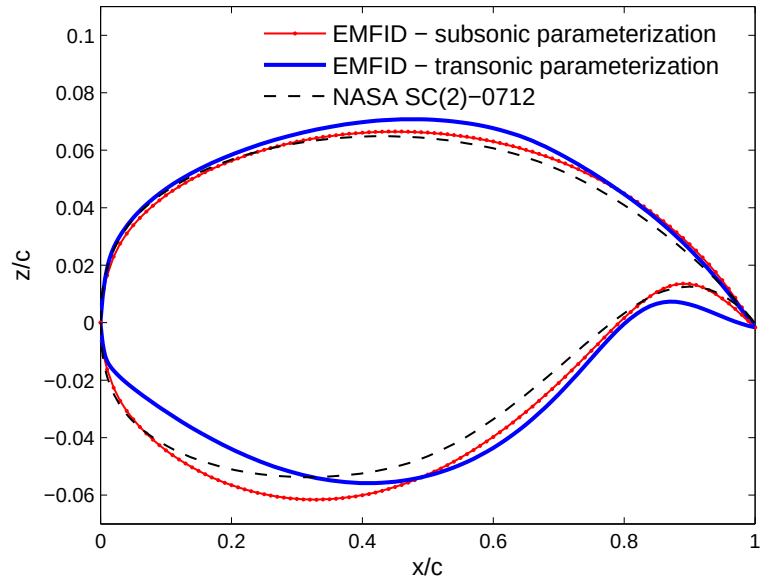
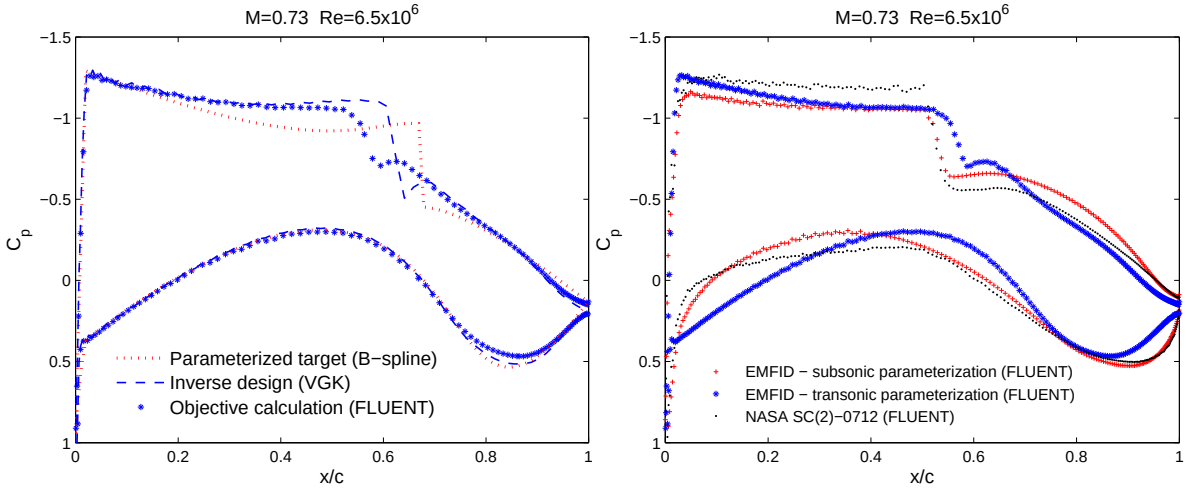


Figure 4-17 Comparison of the best performing geometry from the six-variable (subsonic) and eight-variable (transonic) EMFID methods, shown with the NASA SC(2)-0712 supercritical airfoil.



(a) **(b)**

Figure 4-18 (a) The *optimized* target pressure distribution, profile achieved during inverse design and profile output from FLUENT. (b) Pressure distributions due to the best performing EMFID and benchmark geometry, and the NASA SC(2)-0712 shape.

Finally, it is interesting to observe the performance of the designed transonic airfoils over a range of operating conditions, i.e., varying the angle of attack. Figure 4-19 shows the FLUENT predicted lift-drag polars for the best design from the EMFID subsonic and transonic parameterization methods, shown with FLUENT results for the NASA SC(2)-0712 and RAE 2822 airfoils. It can be seen that the airfoil design using the transonic parameterization gives the lowest drag at the design lift of c_l=0.8. However, this is the only condition at which the airfoil gives lower drag than the other airfoils; like the designs in Case 1, it is highly specialized for the single operating point it was designed for. The airfoil design using the subsonic parameterization is marginally better than NASA 0712 at the design lift coefficient. The NASA airfoil and, particularly, the RAE 2822 give better performance over a range of operating conditions including at typical cruise conditions, i.e., in the region of c_l =0.5. Figure 4-20 shows plots of lift vs. angle of attack, for the same airfoils as in Figure 4-19. Interestingly, the design from the subsonic parameterization gives the same maximum lift as the NASA 0712 supercritical airfoil. However, it should be noted that the design lift of c_l =0.8 is not that far from the maximum lift, which is approximately c_l =1.05. The transonic parameterization design has a poorer maximum lift capability, and this is consistent with the above statement that this airfoil is highly specialized for c_l =0.8.

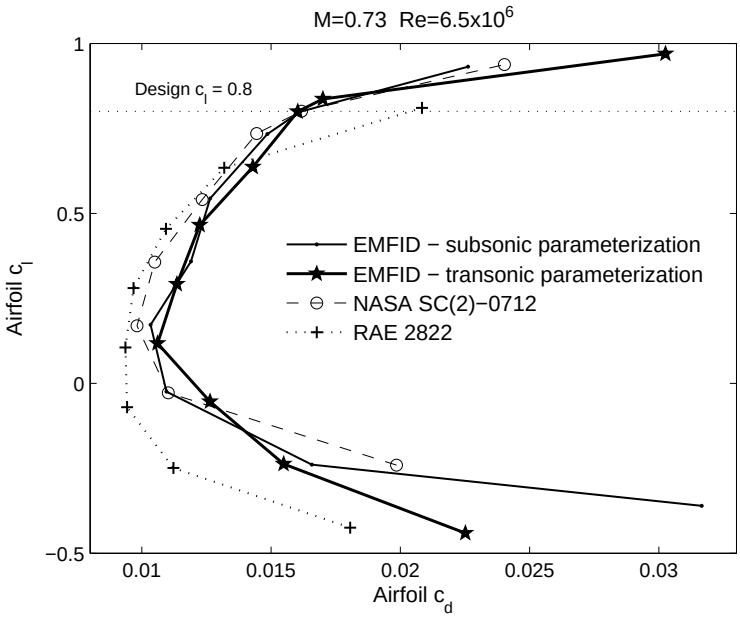


Figure 4-19 Lift-drag polar plot for the best designs from the six- and eight-variable EMFID methods, shown with FLUENT results for the NASA SC(2)-0712 and RAE 2822 airfoils.

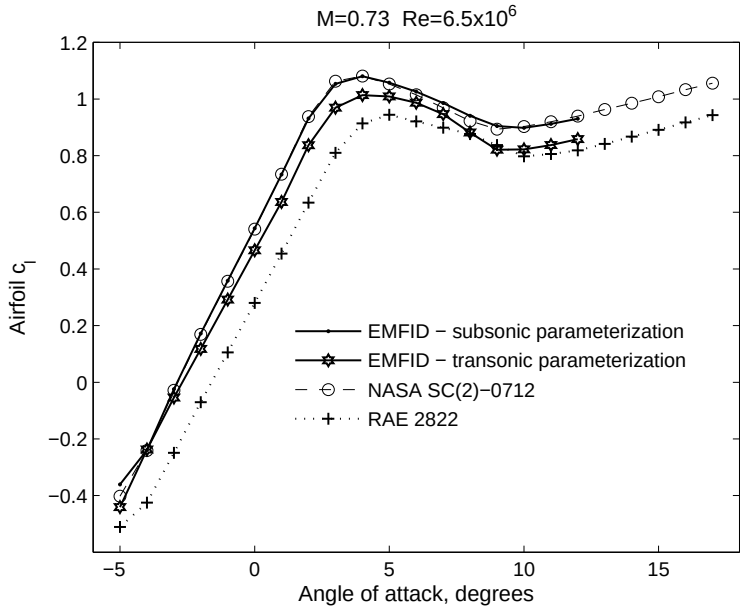


Figure 4-20 Lift vs. angle of attack for the best designs from the six- and eight-variable EMFID methods, shown with FLUENT results for the NASA SC(2)-0712 and RAE 2822 airfoils.

4.6 Corollaries from 2-D Airfoil Design

A design approach has been proposed in this thesis which uses a parameterization of the airfoil pressure distribution coupled with inverse design, with the aim of achieving a powerful but concise method of describing a design and thus a more efficient design optimization process. This EMFID method has been applied to 2-D airfoil design in a series of case studies; the purpose of the remainder of this chapter is to consider the wider implications of these results with respect to the concept of flow feature parameterization. The four case studies detailed in this chapter are summarized in Table 4-7. References are made to these case studies throughout the following discussion, and pertinent conclusions are drawn. The lessons learnt from this analysis are applied to a 3-D design problem in the next chapter.

Table 4-7 Summary of the 2-D airfoil case studies.

Case studies		Flow equations	Flow speed Mach number	Case description
2-D Airfoil design	1	RANS	0.15	Comparison of EMFID and benchmark methods
	2	RANS	0.15	Benchmark run using a low dimensional model
	3	RANS	0.15 VGK, 0.73 FLUENT	Comparison using a transonic drag calculation with a subsonic C_p model in EMFID
	4	RANS	0.73	EMFID run using a transonic C_p model

4.6.1 An Increase in Efficiency

In case study 1, the EMFID and benchmark design strategies are applied to the design of airfoils in a subsonic flow regime (M=0.15). The EMFID method uses a six-variable B-spline parameterization of the airfoil subsonic C_p distribution coupled with inverse design, while the benchmark process employs a more traditional 13-variable spline parameterization of the airfoil geometry. Each method is run five times using different DoE seed arrays, and each time starting from a design equivalent to the NACA 2414 airfoil, which is known to perform relatively poorly at the chosen flow conditions. It is found that, for a given computational cost, the EMFID method is able to return airfoil geometries which perform noticeably better than those of the benchmark method, in terms of drag. The EMFID design search converges significantly more quickly onto promising designs, and the five computations produce a greater level of convergence than the five benchmark runs, i.e., the best five EMFID designs show greater similarity. The consequences of this study are threefold. First, it can be concluded that a B-spline parameterization of the C_p distribution can yield, after inverse design, airfoil shapes of high quality and detail. The airfoils designed in this way out-perform the benchmark shapes, and have been shown to be competitive with at least two of the NASA low-speed airfoils, at the design conditions. Second, because the specified C_p profile model is relatively simple, it is noted that the resulting airfoils are inherently smooth in their shape. In contrast, the behaviour of the cubic-spline, used to define the airfoil in the benchmark method, means that a large proportion of the design space consists of shapes which exhibit *ripples* or are generally nonsensical in their shape. The absence of non-smooth designs accelerates the convergence of the EMFID design process. Note that there are many other parameterization methods which enforce airfoil-like shapes, but these invariably assume some fore-knowledge of successful airfoil

shapes and thus lead to an evolutionary design method. Third, the use of just six variables compared to 13 in the benchmark method allows the EMFID procedure to converge significantly more quickly. This low dimensionality results in a more thorough exploration of promising areas of the design space for a given computational expense. Thus, the objective of creating a more efficient design optimization process, via a reduction in the number of design variables, is realized.

Case study 2 augments the results from Case 1. An alternative benchmark method is run, using a B-spline parameterization of the airfoil which uses only six variables, i.e., the same number of dimensions as used in the EMFID process demonstrated in case study 1. It is observed that, although the reduction in dimensionality allows faster convergence, the ultimate level of performance achieved using this method is poorer than the previous computations using the 13-variable benchmark model. The manipulation of just six surface control points is apparently insufficient to facilitate the subtle and detailed changes that are possible with the previous benchmark parameterization, or indeed, the EMFID model. The interest in the EMFID concept is motivated by the claim that a low-dimensional parameterization of the airfoil pressure distribution can generate airfoil shapes of higher quality and complexity than a geometry-based parameterization with the same number of dimensions. While the results presented in case study 2 are not unequivocal proof, they emphatically support this claim.

While a parameterization of the pressure distribution has been shown to be very effective, the EMFID approach in this guise benefits significantly from the observation that the pressure distribution corresponding to the optimum airfoil, or at least a very good airfoil, is simple in its form. This is not to say that the pressure distribution itself is simple; airfoil pressure profiles can be vastly complex on a microscopic scale, and the prediction of such fine details can be difficult despite constant advances in simulation techniques and computing power. However, for the purposes of providing a target for inverse design, it is sufficient to capture the overall shape and curvature of the pressure profile. Hence, the pressure distributions of interest, i.e., those corresponding to very good airfoils, can be represented simply using B-spline functions, and this allows these airfoils to be generated by the current six-variable parametric model through the use of inverse design. In contrast, if it so happened that the optimum pressure distribution were more complex, it might not be possible to approximate it using such a low-dimensional model. It follows that the choice of flow feature to parameterize in EMFID should take into account the likely simplicity or complexity of the optimum flow features.

The parameterization of surface pressure has been shown to produce higher quality airfoil geometries than a geometrical model of the same dimensionality. However, this parametric model is not capable of achieving the level of detail permitted by the 13-variable benchmark model. Evidence of this can be seen

in Section 3.8, in which the 13-variable cubic spline was shown to reproduce the standard NACA and NASA airfoils with greater accuracy. Therefore, it may be assumed that if the two methods were run on for more than 300 FLUENT calls, the benchmark search would eventually result in a better performing airfoil than the EMFID method. However, in a design situation the objective is rarely to achieve ultimate performance. Rather, the aim is to maximize the design improvement for a fixed computational budget, i.e., to perform efficient optimization. It is these efficiency gains which make the EMFID approach attractive.

4.6.2 The Importance of Flow Feature Coupling

Case study 3 applies the EMFID and benchmark methods to the design of airfoils in a transonic flow regime, enforcing the existence of a shock which is manifested in the surface pressure distribution. The design objective function is total drag, calculated using FLUENT and specifying M=0.73 and c_l=0.8. To begin with, the EMFID and benchmark methods are run in the same way as in case study 1, with the exception of the higher Mach number and target lift. This means that the EMFID parameterization of the pressure distribution (and subsequent inverse design) corresponds to a much lower Mach number (M=0.15) than the objective function calculation (M=0.73), i.e., these flow features are *decoupled*. In case study 3 the EMFID method generated the best performing airfoil, but the performance of the method was not as good, relative to the benchmark method, as observed in the first (purely subsonic) case study. In particular, the level of convergence between the five EMFID searches is poor, and the five final geometries are rather different, especially in the region of the aft lower surface. Case study 4 considers a more complex eight-variable parameterization, which includes the specification of the shock position and pressure jump, and uses a transonic inverse design code at the *correct* Mach number M=0.73. The direct flow feature coupling is therefore restored in this case. Using this configuration, the convergence of the five computations is noticeably improved, and the five final geometries exhibit significantly greater similarity. It is therefore inferred that the EMFID method benefits from a direct coupling between flow features calculated using the low- and high-fidelity CFD. A number of possible explanations of this effect are offered in what follows.

First, as already mentioned, the relationship between the six *subsonic* variables and the drag calculated at M=0.73 is likely to be somewhat more complex than the relationship existing between the eight *transonic* variables and drag. This statement is surmised because the specified six parameters defining the subsonic pressure profile are weakly associated with the true (FLUENT) pressure profile. This can be observed in Figure 4-14, since the parametric (B-spline) pressure distribution is quite dissimilar to the profile

predicted using FLUENT. Because the design variables are weakly coupled with the *real* surface pressure, it follows that there is a rather complex relationship with drag.

The preceding discussion infers that the relationship between the design variables and drag is made to be more complex when the low- and high-fidelity flow regimes are very different. This might explain why the EMFID method performs poorly when the objective function contains a wave drag component but the parameterization of the pressure profile is subsonic. A second possible cause for this is that the six-variable subsonic model is simply less able to specify the airfoil shape in sufficient detail to achieve a good shape for the transonic conditions. In case study 2, it was observed that a simple six-variable B-spline representation of the airfoil was unable to produce such detailed changes as the more complex 13-variable model. Similarly, it may be that the parameterization of the subsonic pressure distribution is not able to perform sufficiently detailed changes in shape, for the purposes of designing an airfoil for transonic flows. Such detailed changes might be made possible with the eight-variable transonic pressure profile model, because the inverse design process is performed for transonic conditions. The inverse process in this case applies a boundary condition for surface modifications at the predicted shock location, and thus the airfoil is tailored very specifically for the transonic flow conditions which are subsequently used to calculate the objective function (drag) using FLUENT. Conversely, the subsonic inverse design process applies a single surface alteration scheme to the entire upper surface, and is run at much lower Mach number than the objective drag calculation, and this leads to difficulty in arriving at an efficient shape.

The eight-variable transonic pressure profile parameterization is able to generate airfoil shapes using the *correct* transonic flow conditions, but a further advantage of this approach is the ability to specify, and therefore strongly influence, the position and strength of the shock. These shock details are two of the eight design variables, and these variables therefore offer a direct means of controlling the wave drag. It is likely that these two variables dominantly control the wave drag, with the remaining six variables having little effect. Indeed, according to Inger [1993], under certain assumptions the wave drag can be expressed simply as a function of the preshock Mach number. Such a direct coupling between the design variables and drag can perceivably yield an objective function landscape which is more readily searched by the optimizer. The subsonic parameterization does not permit such direct control over wave drag; rather each of the six variables affects the total drag in a complex function involving all variables.

In summary, it has been found that, when using a parameterization of the airfoil pressure distribution, it is beneficial if the flow conditions of the parameterization correspond to those of the final objective function. In particular, it is important to include shock details in the parametric model if the design flow

regime is transonic. While such a result is intuitive, the specific reasons for this effect are complex, and there may be a number of contributing factors involved.

4.7 Towards a 3-D Application

The EMFID strategy has proved to be successful for 2-D airfoil design, since the anticipated gains in efficiency have been realized. It is desirable, therefore, to attempt to apply the concept to a 3-D design application, since the gain in computational effort is potentially much greater. The objective of the following chapters is to apply the EMFID approach to a 3-D problem, departing from the subject of airfoil section design and the use of the pressure distribution as the target flow feature. Thus, the efficacy of the EMFID strategy for a wider range of applications is assessed. In taking EMFID forward to a more complex design task, it is important to heed the results of the 2-D airfoil study discussed in this chapter. In particular, it is desirable to parameterize a flow feature which is simple in form when corresponding to high performing designs. Also, there should ideally be a strong coupling between the flow features predicted by the low- and high-fidelity CFD codes.

Chapter 5. Setup of EMFID for Wing Design in 3-D

5.1 Introduction

In the preceding chapters of this thesis, the EMFID method has been proposed and applied to the design of 2-D airfoil shapes. A number of useful conclusions have been drawn from this study. However, while a 2-D application is viewed to be valuable research, a more useful application is one which makes use of three-dimensional CFD simulations. The increasing capability of digital computers is allowing 3-D flow analyses to become more accessible in the design office, and such simulations are frequently coupled with parametric 3-D CAD software (Trapp and Sobieczky [1999]). Fundamentally, the aerodynamic component to be designed exists physically in three dimensions, and so designers are interested in, and increasingly aware of, complex 3-D flows. The significant computational cost of 3-D analyses arises from the requirement for a large number of mesh cells for which the fluid flow equations are calculated. Jameson [1996] suggests that around 3×10^5 cells are sufficient to give an accurate Euler solution, but this may take several hours to compute in practice. Thus, a design strategy such as EMFID, which can potentially make more efficient use of the CFD calls, is more attractive for an application which considers 3-D flows since the cumulative reduction in computational cost for a design search process is likely to be greater than for a 2-D study.

The present chapter describes the application of EMFID to the design of aircraft wings. As before, the design objective is to minimize drag for a fixed level of lift, and design studies are performed using Euler or RANS CFD analyses. In the preceding chapters, the parameterized flow feature was the surface C_p distribution, and this was shown to be an effective design approach for airfoils. One would expect that the strategy would be equally effective when the airfoils are used to make up sections of a wing. However, to avoid repetition and arriving at similar conclusions, alternative flow features are considered for the present wing design studies. It is hoped that this exploratory approach enhances the findings of the current work.

Recently, a reduction of aircraft drag has been achieved by the development of wing-tip devices. These passive geometries are designed to reduce the *induced drag* of a lifting wing by modifying the trailing vortex system. Since the wing-tip vortex is such a prominent flow feature, designing in the wing-tip region provides an interesting application for the EMFID method. Thus, the benchmark and EMFID methods are here applied to the design of a wing-tip device. In the sections that follow, induced drag and current practice with respect to the design of wing-tip devices is described. Following this, the parameterization of flow features for the design of such a device is investigated, and the setup of the benchmark and EMFID methods for this design study is detailed.

5.2 The Drag on a Finite Wing

As previously stated, the two drag components acting on an aerodynamic body are the skin friction drag and pressure drag. For an airfoil, the pressure drag consists of contributions from form drag and wave drag. For a wing of finite span, there is a third contribution to the pressure drag, namely the *induced drag*. For a lifting wing there exists a pressure imbalance between the upper and lower surfaces which results in a component of velocity in the spanwise direction, and generates flow around the wing tips towards the lower pressure region on the upper surface. As a result, the flow tends to establish a rotational motion which extends downstream from the wing, forming the wing-tip vortex. Wing-tip vortices can roll up into a large body of rotating air; Figure 5-1 shows such a wake structure generated as part of a study performed by NASA. The presence of these trailing vortices can cause a small plane to lose control in the wake of a large airliner, which is why aircraft must maintain a separation of several nautical miles during take-off and landing (Spalart [1998]).

Figure 5-1 An aircraft wake vortex study performed by NASA.

In addition to producing a rather disruptive wake, wing-tip vortices adversely affect the drag of the aircraft from which they emanate. The vortex induces a downward component of velocity at each point along the wing adjacent to the trailing edge, which reduces the local angle of attack seen at each wing section. This inclines the lift vector, creating an additional force component in the direction of the freestream velocity, i.e., adding an extra component to the drag. This drag component is the induced drag. The trailing vortex has the effect of modifying the wing surface pressure, and hence the C_p distributions at each wing section are not the same as for the corresponding 2-D airfoil. By using Prandtl's classical lifting line theory, the induced drag can be shown to be proportional to the square of lift, L (Anderson [1991]). The total drag, D, can therefore be approximated as

$$D = k_1 q + \frac{k_2 L^2}{q}, \tag{5.1}$$

where q is the dynamic pressure and k_1 and k_2 are constants relating to geometry (Kroo [2001]). The first term arises from viscous drag, while the second term is due to induced drag. From Eq. (5.1) it can be shown that the minimum drag for a given lift occurs when these two terms are equal. Therefore, the induced drag amounts to approximately half of the total drag at the conditions of minimum drag. In practice the induced drag can make up as much as 40% of cruise drag and 80-90% of the total drag in take-off configuration (Kroo [2005]), and so the induced component has a substantial effect on the performance and ultimately the design of wing geometries. It is because of induced drag that some large migratory birds fly in 'V' formation; by transferring the downwash of shed vortices to the wing-tips of adjacent birds, the flock can travel for longer distances than the individual.

One of the key results from Prandtl's classical lifting line theory (Anderson [1991]) states that the induced drag coefficient, $C_{D,i}$, is related to the lift coefficient according to

$$C_{D,i} = \frac{C_L^2}{\pi\, eAR}, \tag{5.2}$$

where AR is the wing aspect ratio and e is the span efficiency factor. Eq. (5.2) implies that the induced drag can be reduced in two ways. First, the aspect ratio can be increased by increasing the span relative to the average chord. However, increases in span are limited by requirements for structural weight and airport gate restrictions. Second, classical lifting line theory predicts that the minimum induced drag for a given span occurs when e takes its maximum value of 1, which happens to be when the spanwise distribution of sectional c_l is elliptic in form. Further, for a wing with no geometric or aerodynamic twist, an elliptic lift distribution is achieved using an elliptic wing planform shape. An attempt to minimize induced drag was one of the reasons why the elliptic planform was adopted in early aircraft design; the most famous example of which is the Supermarine Spitfire. However, due to structural constraints, manufacturing difficulties and the desire to operate at high Mach numbers, contemporary aircraft do not feature elliptic wings and neither do they feature an exactly elliptic lift profile. In modern wing design, once the wing span is specified the induced drag can be minimized by optimizing the spanwise loading subject to the aforementioned constraints. This can be achieved by varying the wing twist, airfoil section camber and chord distribution. Additionally, wing-tip devices have recently been shown to provide further improvements in drag reduction.

5.3 Wing-Tip Devices

Wing-tip devices have been studied and applied to aircraft designs since the mid-1970's. The primary motivation for incorporating a wing-tip device is to modify the trailing vortex flow beneficially to reduce lift-induced drag. Aircraft manufacturers are under increasing pressure to improve efficiency due to rising operating costs and environmental issues, and this has led to some innovative developments. Several different types of wing-tip device have been born out of this drive for efficiency.

It was Frederick Lanchester who first attempted to address the problem of wing-tip vortices (Jupp [2001]). In 1897 he proposed the idea of *capping planes*, now more commonly known as *end plates*, which were designed to minimize the spanwise component of flow across the wing. Unfortunately, the

concept of end plates never showed improvements in practice because the increase in drag due to viscosity and separation outweighs any induced drag benefit. End plates are therefore not used on aircraft, but are used routinely on racing car wings (Figure 5-2), since these wings have a very low aspect ratio and are often designed to maximize lift (downforce), with less interest in drag reduction.

Figure 5-2 A racing car front wing, featuring end plates.

A breakthrough in wing-tip devices came with the development of the winglet by Whitcomb at NASA in the 1970's. Like the end plate, the *Whitcomb winglet* was vertically positioned, but unlike the end plate this was a lifting surface which was designed to generate a flow field which mitigates the flow around the wing tip. This resulted in a net improvement in overall drag. At present, this improvement is relatively small, but the canted winglet is being increasingly adopted on commercial airliners, such as the McDonnell Douglas MD-11 and the Boeing 747-400 (Figure 5-3). Further advantages of using winglets are the potential for improved handling characteristics and opportunities for marketing, exploiting the high visibility of the upper surface of winglets to passengers. A modern variant of the winglet is the blended winglet (Figure 5-4), which features smoother upward sweep and is used on many Boeing aircraft (Faye et al. [2002]).

(a)

(b)

Figure 5-3 Examples of winglets on modern commercial aircraft (a) McDonnell Douglas MD-11 (b) Boeing 747-400, with a B747 freighter in the background.

Figure 5-4 Example of a blended winglet on the Boeing 737-800.

The wing-tip fence (Figure 5-5) can be seen as an advancement of the end-plate, with surfaces extending vertically above and below the wing main plane. This device is small relative to a winglet, but has been shown to be very effective in terms of drag reduction. It has been used principally on Airbus aircraft, such as the A300, A319/20 and the A380 (Jupp [2001]).

Figure 5-5 Example of a wing-tip fence on the Airbus A319.

A feature on some Boeing aircraft is raked wing-tips (Figure 5-6). The tip of the wing is in the same plane as the main wing but exhibits significantly higher sweep, both at the leading and trailing edges. This seemingly simple modification can yield significant drag reductions; a 5.5% reduction has been reported for raked wing-tips compared to between 3.5-4.5% for conventional winglets (Faye et al. [2002]). In fact, there are many examples of successful application of aft swept wing tips in nature, for example, the albatross and the common swift (Burkett [1989]). However, due to span restrictions, winglets or wing-tip fences are often used in preference to raked wing-tips when retro-fitting to existing aircraft.

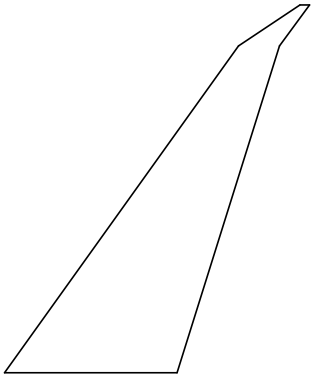


Figure 5-6 Illustration showing the planform geometry of the raked wing-tip on the Boeing 767-400.

Further research into wing-tip devices has resulted in some radical proposals. *Wing tip sails* (Spillman [1987]) are small wing devices which extend from the wing tip at various dihedral angles, which have the effect of producing a small thrust component which partially offsets the lift-induced drag. A similar device is the multiple winglets concept (Smith et al. [2001]), while the *Wing-Grid* (La Roche and Palffy [1998]) can be described as a cascade of small wing elements with end plates. The *spiroid tip* (Reneaux [2004]) is a spiral loop obtained when joining the tips of a vertical winglet extending from one half of the tip chord, with a horizontal extension from the other half of the chord. Finally, the *C-wing* and *boxplane* are two of a range of non-planar aircraft concepts gaining acceptance (Kroo [2005]). All of these devices are in an early research stage, and are unlikely to be adopted by aircraft manufacturers in the short-term. However, such innovative concepts must be considered seriously if efficient and therefore affordable air travel is to continue into the 21st century.

5.4 A Wing-Tip device for the ONERA-M6 Wing

The design of a wing-tip device is chosen as the application used to evaluate the EMFID method for 3-D flows. The problem of minimizing induced drag and the nature of EMFID as a flow-feature based parameterization means that wing-tip design represents a fitting and interesting assessment. For this study, a main wing geometry is required, onto which the wing-tip feature is designed. It is desirable that this initial wing should be a standard CFD test case so that the CFD results can be validated. The DLR-F4 (Redeker [1994]) and the DLR-F6 (Vassberg et al. [2005]) wing-body configurations are popular geometries which have been used as test cases for the AIAA drag prediction workshops (Levy et al.

[2003]); however, these are rather complex and published results are for flow simulations over the entire wing-body. A more simple wing-alone test case is the ONERA-M6; this is a swept wing with a constant symmetrical airfoil section, linear taper, zero dihedral and zero twist. Experimental surface pressure measurements for this wing have been published by Schmitt and Charpin [1979] for various flow Mach numbers from 0.7 to 0.92. Examples in the literature of the use of this wing as a CFD test case include the works by Nielson and Anderson [2002] and Frink [1996]. The M6 is not a particularly realistic aircraft wing geometry and has a relatively small span; an extension of the span by the addition of a wing-tip device is therefore likely to yield greater design improvement than for a more realistic starting geometry. The ONERA-M6 is therefore chosen as the baseline wing for modifications.

As a result of the 2-D airfoil design case studies described in the preceding chapters, it was inferred that the EMFID method benefits from the existence of a strong coupling between the parameterized flow feature and the objective quantity, drag. In the 2-D airfoil case, this requires the specification of shock details when designing for a transonic regime, since wave drag is present. Consequently, if wave drag is to be considered in the present 3-D design problem then this should be accounted for in the EMFID parameterization of the wing-tip device. However, the low-fidelity CFD code, described next, is unable to calculate flows for shocks, and for this reason a subsonic flow regime is used for the 3-D design studies. A Mach number of 0.7 is employed, as this is the lowest speed used by Schmitt and Charpin [1979], whose experimental data are used for the validation of the CFD analyses. A Mach number of 0.7 is low enough to avoid the occurrence of shocks, provided that the target lift coefficient is typical of aircraft cruise conditions. Since the M6 is swept back, one would not normally use it as a baseline for a subsonic wing design study, although previous design studies have considered the analysis of a swept transonic wing at Mach 0.7 (Chandrasekharan et al. [1985]). In the present investigation the Mach number is limited by the incapability of the low-fidelity CFD, while it is desirable to use a starting point for modifications for which the CFD is known to be valid. The analysis of the swept M6 wing at Mach 0.7 is therefore not illogical. Further, although in the aerospace industry the primary motive for incorporating sweep is to mitigate wave drag, there are many examples of wing sweep in nature (Liu et al. [2006]), and such geometries have been shown to be beneficial in terms of drag reduction (van Dam et al. [1991]).

In the 3-D case studies to follow, the task of designing a wing-tip device for the ONERA-M6 is performed using Euler simulations, and subsequently using RANS simulations. As in the case of the 2-D design studies, the EMFID and benchmark methods make use of the same high-fidelity CFD, while EMFID additionally makes use of a low-fidelity code.

5.5 Wing Analysis: CFD Solver Setup

For the present 3-D wing design studies, the high-fidelity analysis employed uses the FLUENT software and the low-fidelity solver is VSAERO, also a commercially available CFD code. The setup of these solvers in preparation for the design exercises is described below. The 3-D analysis setup is subjected to similar accuracy verification and validation assessments as performed for the 2-D analysis. Full details of the CFD validation and verification can be found in Appendix B.

5.5.1 FLUENT

Both the benchmark and the EMFID method call on FLUENT to calculate the drag as the design objective. Separate drag minimization studies are performed using Euler calculations, i.e., considering only the induced drag, or using the RANS equations, i.e., calculating the total drag including viscous effects. The FLUENT setup is different in each case, but in both cases the aim is to minimize the computational expense of an objective function call, while maintaining an acceptable level of accuracy. The meshing software, GRIDGEN, is used to generate the wing geometry and to mesh the flow domain to be solved by FLUENT. A multi-block structured mesh is used, where the hexahedral cells follow a *C-H* topology (Steinbrenner and Anderson [1989]). The ONERA-M6 wing planform is reproduced using the data supplied by Schmitt and Charpin [1979], i.e., the geometry is constructed at wind-tunnel model scale. The finite trailing edge thickness of this wing is retained, and the wing tip is capped using a half body of revolution which is truncated at the leading edge to match the leading edge sweep of 30°. The (X,Z) plane containing the wing root is designated a symmetry-plane boundary condition.

A verification exercise has been performed using the baseline M6 wing, to assess the dependence on the mesh setup of the drag result; this is detailed in Appendix B. The mesh used for RANS analyses has a total of 1.3M cells. This comprises 123 cells chordwise over the wing section and 70 cells in the spanwise direction. The distance of the far-field boundary from the wing is approximately 8 spans. For analysis using the RANS equations, the height of the first cell adjacent to the surface is 0.15mm, as this gives a wall y^+ between 30 and 60, i.e., in the range suitable for a log-law representation of the boundary layer. The k-ε turbulence model (Shih et al. [1995], Fluent [2003]) is employed, since this is known to be a robust and accurate method for 3-D flows containing vortical flow structures. A coupled solver is used to solve the continuity, momentum and energy equations. For simulations using the Euler equations, the configuration is the same as above, but since viscosity is not considered a turbulence model is not

required. Also, a slightly different mesh setup is used, as the absence of the boundary layer means that a lower cell density can be used normal to the wall. A first cell height of 1.5mm is used for this Euler mesh, reducing the number of cells to 800K.

For the purposes of validation, the results from FLUENT for the ONERA-M6 have been compared against the experimental data provided by Schmitt and Charpin [1979]. Figure 5-7 shows this comparison for the 99% span station of the wing; similar plots for six other spanwise stations along the wing are given in Appendix B. The FLUENT results appear to agree well with the wind-tunnel data, although there is a slight disparity towards the trailing edge. This point is very close to the aft corner of the wing where the pressure is strongly affected by the predicted vorticity.

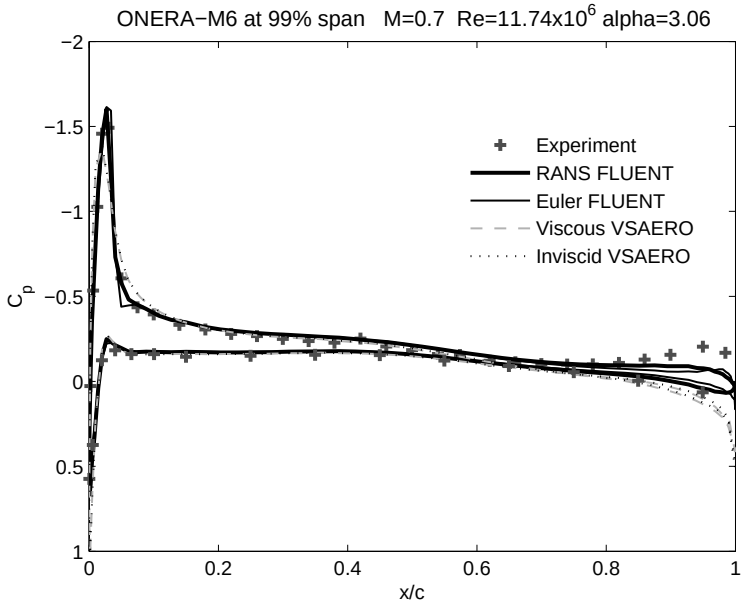


Figure 5-7 Surface pressure distribution for the ONERA-M6 wing at 99% span, showing the experimental data with viscous and inviscid results from FLUENT and VSAERO.

Using the above FLUENT setup, C_L and C_D are calculated using surface force integration, with a reference area equal to the projected area of the ONERA-M6 wing. For the design studies, the drag must be calculated for a fixed value of lift. In the 2-D analyses, the FLUENT solution was run to convergence at three angles of attack including the angle which produces the required target c_l. To reduce the computational cost of the 3-D analyses, the drag and lift are calculated at two angles of attack, the first gives a converged solution after 4000 iterations and the second after a further 3000 iterations. Following

this, the drag for the required lift is determined via linear interpolation using the two predicted C_L and C_D results. Since the wing geometry is only varying over the wing-tip device, variations in lift and angle of attack between designs is small, and hence the present interpolation method is sufficient for the drag calculation. In the design studies to follow a target C_L of 0.4 is used throughout, as this is a typical value at cruise conditions for a transport aircraft (Heffley and Jewell [1972]).

5.5.2 VSAERO

VSAERO is a CFD code based on the subsonic panel method, and is able to calculate the non-linear aerodynamic characteristics of an arbitrary geometry (Nathman [2003]). The program solves the linearized potential equation for inviscid, incompressible, irrotational flow using the Morino formulation (Letcher [1989]), i.e., a distribution of known source and unknown doublet singularities over the surface panels. The approach adopted is a low-order panel method, i.e., a piecewise constant distribution of singularities, which leads to only a single unknown potential on each panel (Maskew [1982]). The non-linear effects of the wake roll-up are considered using an iterative wake relaxation procedure, which attempts to align the wake panels with the local streamlines. Although the underlying calculation is for inviscid and incompressible flow, the effects of viscosity can be modeled using integral boundary layer calculations, while compressibility corrections can also be included, such as the Prandtl-Glauert linearization. The major computational expense in the VSAERO solution procedure arises from the determination of influence coefficients for the matrix equation, and the subsequent solution of this large linear system of equations. Nonetheless, a single analysis call is very quick, requiring approximately 30 seconds on a Xeon 2GHz processor.

In the current work, the VSAERO code is used in the inverse design process for EMFID. An inviscid flow configuration is used in the case studies where Euler FLUENT analyses are used, and similarly a viscous setup is used in RANS design studies since the presence of the boundary layer has an effect on the wing loading. In the viscous case, two iterations of the integral boundary layer equations within VSAERO are found to give sufficient convergence of accuracy. For the purposes of determining the spanwise loading on a wing, no wake iterations are used, but in cases where the details of vorticity are required a single iteration of the wake relaxation procedure is found to give improved accuracy. The panel configuration used is the same as that used by Robinson and Keane [1999] and Petruzzelli and Keane [2001]. Further details of the setup of the VSAERO analysis can be found in Appendix B.

Figure 5-7 shows a pressure distribution predicted by inviscid and viscous runs of VSAERO, which include the wake calculation, compared against the experimental data and the FLUENT results. In general, the surface pressure is predicted accurately except in the region near to the trailing edge. As previously mentioned, the pressure in this region is affected by the tip vortex. The flow in VSAERO is irrotational, and so off-body vorticity can only exist on the wake panels. This departure from the physical flow phenomenon is the likely reason for the discrepancy in surface pressure.

5.6 Investigating an Appropriate Flow Feature for EMFID

For the 2-D airfoil design studies, the distribution of surface pressure coefficient was chosen as the flow feature to be parameterized in EMFID. This decision was relatively straightforward because there exists a strong relationship between pressure and geometry, and this has been exploited by a large range of extensively researched inverse design methods. For the present 3-D case the choice is less obvious. The parameterization of a pressure *surface* over the wing is discarded for two reasons. First, the pressure profile has already been shown to be effective for 2-D section design, and to pursue this path would likely lead to similar conclusions. Second, such a parameterization would require a complex and high-dimensional parameterization scheme, resulting in a lengthy design search. Thus, an alternative flow feature is sought for use in EMFID for the design of a wing-tip device. In the current work, two such alternatives are considered: the details regarding the wing tip vortex, and the shape of the spanwise lift distribution. In what follows the geometry parameterization approaches used to investigate these two flow feature types are detailed and the feasibility of an EMFID parameterization of such flow features is considered.

5.6.1 Geometric Description of the Wing-Tip Device

In order to perform inverse design, a geometric description of geometry is required. In the 2-D airfoil design work, the geometric description used in EMFID was different to that used by the benchmark method because the inverse design process made corrections to the airfoil second derivatives directly. Here, the same geometry model can potentially be used by both design methods, i.e., the inverse design process in EMFID searches the benchmark geometry definition in an attempt to match the target flow

feature. Two parameterization approaches are described; the first considers five *gross* wing planform parameters which are distinct, the second uses a single variable which is varied along the span.

The first parameterization is one which might be used by an aircraft design team during a conceptual design exercise. There are five design variables, denoted here by the vector $\boldsymbol{X}$, describing the wing-tip device and these are listed in Table 5-1 along with their upper and lower bounds. Figure 5-8 shows a diagram of these variables. The wing-tip device is an extension of the ONERA-M6 wing, where the total span of the device is fixed (if the span is made a variable then the optimization process always maximizes it, since the present analysis does not consider structural constraints). Using these five *gross* variables, a design search process can result in planar devices, such as a raked wing-tip or twisted wing-tip, or it can produce a highly non-planar device such as a winglet.

Table 5-1 Gross wing-tip device design variables.

Design variable	Description	Lower bound	Upper bound
X_1	Dihedral angle, °	0	70
X_2	Leading edge sweep, °	0	70
X_3	Tip section incidence (washout), °	-20	20
X_4	Tip section chord length, m	0.1	0.41
X_5	Tip airfoil section maximum thickness, m	0.01	0.05

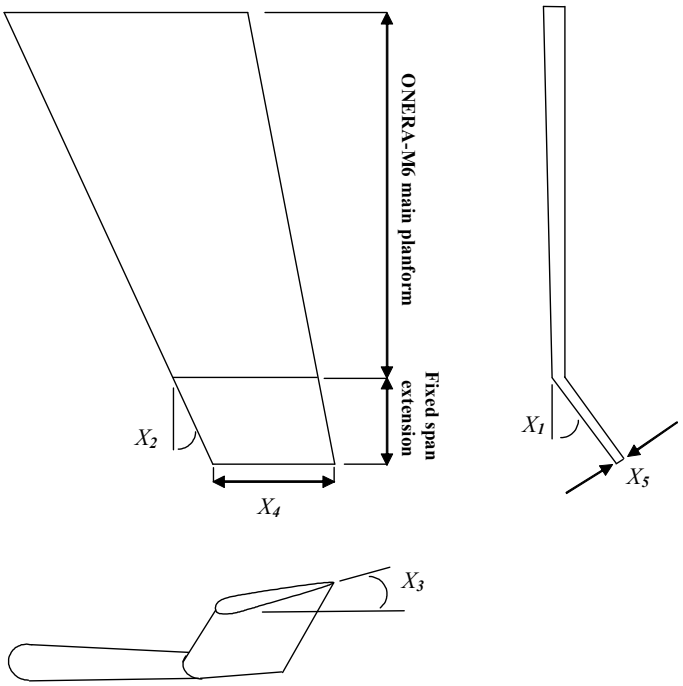


Figure 5-8 Diagram illustrating the five gross wing-tip design variables, as listed in Table 5-1.

The second geometry parameterization method describes the distribution of a single sectional variable along the span. Such a distribution could be the spanwise twist, chord or sectional camber distribution. The logic behind this approach is to benefit the EMFID method, as it is thought that this spanwise distribution is strongly coupled with the wing c_l profile. This is more akin to the EMFID setup for 2-D

airfoil design, in which the chordwise distribution of the position of the airfoil surface was varied and this is strongly linked to the C_p profile. In this work, the distribution of chord over the wing-tip device is varied, as this limits the number of failures when building the computational mesh and such a design search is likely to produce an interesting result. The chord is varied at six equispaced spanwise stations along the wing extension, including at the wing tip, giving six design variables. The chord at these stations is bounded in the range 0.35-0.5m. The dihedral is set to zero, i.e., constraining the problem to a planar wing-tip device, and the leading-edge sweep is maintained from the main wing, i.e., 30°. A typical example of a wing-tip device generated using this parameterization is shown in Figure 5-9.

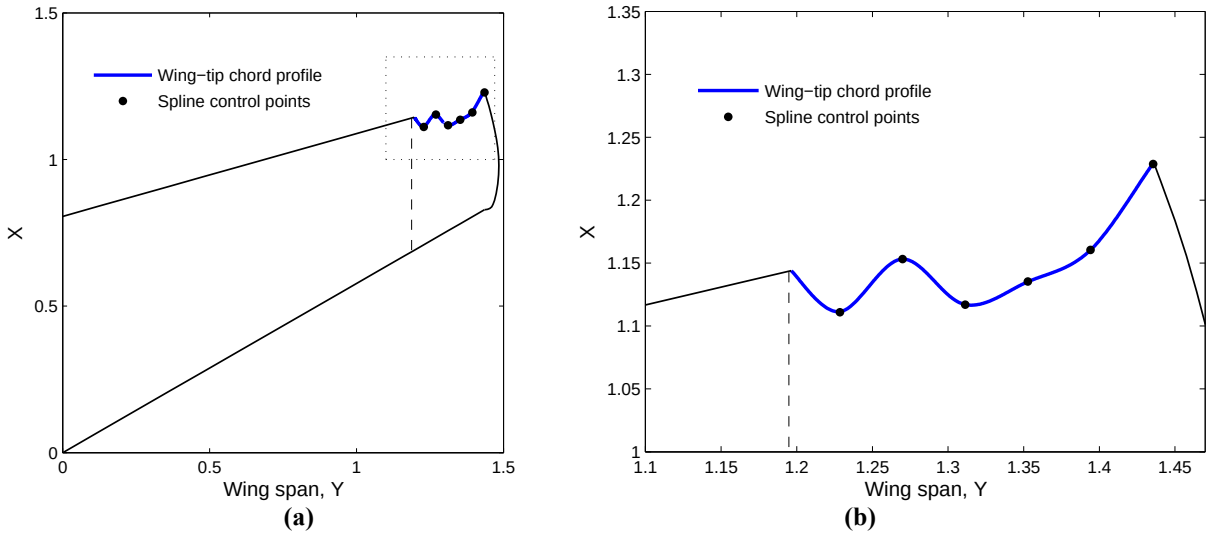


Figure 5-9 An example of a wing-tip device generated using a parameterization of the trailing edge chord distribution. (a) Planform view on equally scaled axes. (b) A close-up view of the wing tip region.

5.6.2 Target Wing Tip Vortex Properties

Having described the geometry parameterization methods for the design of a wing-tip device, in what follows these representations are used to investigate an appropriate flow feature to parameterize for the EMFID design strategy. In EMFID, the parameterized flow feature must act as a target for an inverse design process utilizing the low-fidelity VSAERO code. As stated in the above discussion, it is known that the occurrence of lift-induced drag is due to the energy lost in the wing-tip vortex. It seems reasonable, therefore, that to minimize the induced drag requires a change to the trailing vortex system. Hence, a parameterization of the vortex details in the EMFID search could perceivably provide efficiency gains for a design search procedure. The suitability of such an approach has been assessed by comparing the vortex predictions of the VSAERO and FLUENT CFD codes. This study is documented in Appendix

C since it employs quite different analysis techniques and thus it interrupts the main direction of this section. However, a summary of the investigation and the key conclusions are described here.

The vortex details which are considered as design parameters are the (Y,Z) position of the vortex centre on a plane downstream from the wing, the maximum vorticity, and the total crossflow circulation on the same plane. First in Appendix C, the techniques used to extract these vortex parameters from the FLUENT and VSAERO velocity data are described. A crossflow plane at X=4m is used. The position of the vortex centre is determined using the VORTFIND method of Pemberton [2003], the vorticity field is calculated as the curl of the velocities and circulation is computed by integrating the vorticity over the crossflow plane. Next, in order to compare the predictions of the FLUENT and VSAERO solvers, simulations are run for different geometries. The five-variable gross wing-tip parameterization (Figure 5-8) is used to generate 50 designs as dictated by a Latin hypercube DoE plan. The aforementioned vortex parameters, and induced drag, are computed for each of these geometries using VSAERO and Euler FLUENT simulations.

It is found that the predictions of the vortex position from FLUENT and VSAERO agree to an acceptable level of accuracy; however, the position parameters correlate poorly with drag. The peak vorticity of the tip vortex also correlates poorly with drag. Since these parameters do not have a strong relationship with the design objective, they are unsuitable as design variables. The crossflow circulation exhibits a much closer correlation with drag, for both CFD codes. However, it is noted that this parameter is physically equivalent to induced drag, and therefore a stronger correlation is expected. A parameterization of the circulation in EMFID would therefore be an exercise of minimizing the induced drag predicted by VSAERO, an approach used in many multi-fidelity or *zoom* optimization strategies. A parameterization of circulation is therefore not appropriate.

Ultimately, the wing tip vortex properties investigated here are not suitable as design variables because of the poor correlation with drag. Further, the ability of the VSAERO code to simulate the tip vortex is fundamentally limited because it considers potential flow, and therefore vorticity is everywhere zero except on the wake lines. More detailed tip-vortex properties can be obtained using higher-fidelity solvers, but the EMFID concept requires a low-fidelity code for the inverse design step, and even an Euler simulation with a coarse mesh is likely to make the EMFID method prohibitively expensive. The parameterization of the tip vortex is therefore discarded. Instead, the EMFID method may benefit from a parameterization of the spanwise lift distribution; this is described next.

5.6.3 Target Spanwise Lift Distribution

The distribution of aerodynamic loading in the spanwise direction from root to tip is of fundamental importance in the design of aircraft wings, and is considered from the outset in conceptual design. The related discussion in Section 5.2 introduced the outcome that the spanwise lift distribution is intrinsically linked to induced drag. A parameterization of the wing lift profile therefore holds potential for providing efficient reduction of drag using EMFID. As with the 2-D parameterization of the C_p profile (i.e., chordwise loading), it is expected that a parameterization of the wing spanwise loading will allow a reduction in the number of design variables, relative to a geometry parameterization, for a given range of geometric variation. The current section appraises the suitability of this flow feature for the design of the wing-tip device described by gross wing parameters (Figure 5-8). In this work, the measure of lift at a given wing section is determined by the product $c_l \times c$. References to the wing spanwise lift distribution, loading distribution and $c_l \times c$ profile are used interchangeably.

In the previous section related to wing-tip vortex parameters, it was concluded that the vortex predictions from VSAERO do not sufficiently approximate those of FLUENT in order to provide a meaningful design search process. However, the predictions for the spanwise lift distribution are expected to agree much more closely, since it is known that the section C_p distributions are accurately predicted by both CFD solvers (see Section 5.5). Figure 5-10 compares the lift profiles generated using VSAERO and FLUENT for the ONERA-M6 geometry, each giving the same total wing C_L. Some disparity can be seen between the two predicted results. In particular, at this level of lift VSAERO predicts higher loading towards the tip, and this removes some of the load from the root. The deficiency towards the tip is due to an inaccurate representation of the tip vortex, which in turn affects the wing surface pressure. However, the average percentage difference between the sectional $c_l \times c$ is less than 1%, and such similarity is certainly sufficient for the present design studies.

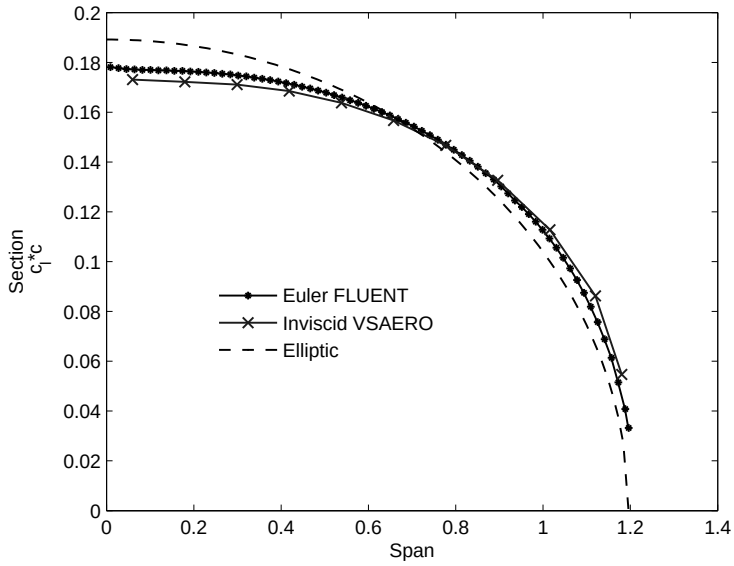


Figure 5-10 Comparison of lift profiles generated using FLUENT and VSAERO for the ONERA-M6 wing. These are compared with the elliptic distribution.

Figure 5-10 also shows an elliptic lift distribution, which, according to classical lifting line theory, gives the minimum induced drag for a given span. This concept was introduced in Section 5.2. Consequently, if it is known that an elliptic loading is optimum, manipulating this loading profile in a search for improved drag is a futile exercise. Instead, one could use an inverse design method with the elliptic profile as a target, and always reach an optimum design. However, as previously mentioned, the elliptic profile is not adopted in aircraft wing design, owing to structural and aerodynamic reasoning.

First, while the elliptic profile gives the minimum drag for a given span, designers are more often interested in finding the optimum for a specified maximum wing weight (Jupp [2001]). This is equivalent to keeping constant the integral of the bending moments across the span due to the aerodynamic loading. Thus, for a given maximum bending moment, the wing design with minimum induced drag has a larger span and a more *triangular* lift profile, compared to the elliptic case. Such an optimum features higher loading towards the wing root compared to the ellipse (Figure 5-10 shows that the ONERA-M6 in fact has a slightly lower inboard loading than the elliptic case, for the same wing C_L, further confirming the fact that this wing is a CFD test case and not a true aircraft geometry). The departure from a pure ellipse through the consideration of complex structural constraints means that the form of the spanwise loading curve is a pertinent issue.

Second, aside from structural issues, there are authors who claim to have found genuine improvements in induced drag compared to the elliptic case. For example, van Dam [1987] used numerical simulations to

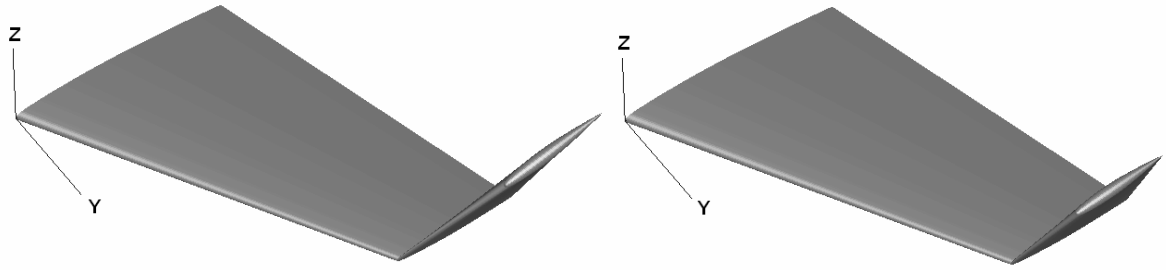
show that tapered and aft-swept wings can produce less induced drag than the theoretical elliptic case, quoting an 8% improvement. In later work, these results were validated using wind-tunnel experiments. (van Dam et al. [1991]). van Dam highlights that classical lifting line theory assumes a planar, horizontal wake, and neglects the influence of trailing wake deformation. The panel code used by van Dam performs calculations for the wake roll-up, and this is thought to have a favourable effect on drag for some configurations. Other work, such as that of Eppler [1997] and Phillips et al. [2006], neglects the wake rollup and consequently the minimum achievable drag corresponds to an elliptic load profile. Both CFD solvers used in the current work perform calculations for non-planar shed vorticity; therefore, if the conclusions of van Dam are followed, the elliptic loading is likely to be non-optimal.

Third, classical lifting line theory assumes inviscid flow, but in viscous flow the spanwise loading giving minimum total drag is actually modified relative to an ellipse, as described in the work by Rokhsaz [1993]. The spanwise lift distribution is based on circulation which is partly generated in the boundary layer, and thus the viscous drag is affected. Rokhsaz [1993] concludes that the optimum compromise between induced and viscous drag requires a small modification to the elliptic lift profile. In the present work analyses are initially performed for inviscid flow, but following this the effects of viscosity are considered. While the elliptic profile can be used as a reference case, it should not be considered as a known optimum. The lift profile, like an airfoil pressure distribution, must therefore be specified or optimized in some way; in this work it is parameterized for optimization using EMFID.

Given that the design problem demonstrated here is focused in the region of the wing-tip, it seems logical that the parameterization should consider only the lift distribution over the wing-tip device. Since the main wing geometry is fixed (as the ONERA-M6), the lift distribution over the wing is affected only by changes to the geometry of the wing-tip device. The aim is to parameterize the $c_l \times c$ profile over the wing-tip device, and using the parametric profile as a target for inverse design a wing-tip geometry can be found whose loading curve closely matches this target. However, using the five gross winglet design variables (Figure 5-8), for a given target wing-tip $c_l \times c$ profile one can imagine that there might be more than one geometry which realizes this target. Such non-uniqueness was not encountered in the 2-D airfoil design work because the entire pressure distribution was varied by the optimizer and specified as a target. Here, since only a portion of the geometry is varied, it is tempting to consider only a portion of target flow feature as this may reduce the number of design variables. As a primary study, the occurrence of this non-uniqueness and its potential effect on the performance of EMFID is investigated.

In the previous section related to wing-tip vortex properties, a 50 point DoE plan was used to generate a set of test geometries based on the variable bounds listed in Table 5-1. The best of these designs, in terms

of the drag at C_L=0.4, is shown in Figure 5-11 (a). This best design is a rather large winglet device; such a design can perform well since the drag objective in this case contains no viscous component. The lift profile over the winglet for this *target* design can be found using VSAERO. This portion of lift curve is then used as a target for inverse design using VSAERO, producing a new geometry which may or may not be the same as the target geometry.



(a) **(b)**

Figure 5-11 (a) The best geometry from the 50 point DoE set. (b) Geometry resulting from inverse design, in which the target flow feature is the winglet lift profile of the geometry in (a).

The inverse design method used here is an optimization search using the method of response surfaces. The variables of this search are the five gross wing geometry parameters, and the objective function is the square of the difference between the target and computed spanwise lift profiles over the wing-tip device. The objective is calculated using the inviscid VSAERO setup at a fixed level of lift (C_L=0.4), which requires three calls to the solver. The optimization procedure is similar to the 2-D benchmark search described in Chapter 3. The objective function is calculated for a DoE array of 50 designs (i.e., 10 times the number of variables), and a response surface is fitted to the data which is searched using a genetic algorithm in order to find promising update points. The process is repeated until 100 update points are reached. In this case, the RSM function employed is the Kriging model (Jones [2001]), which can be tuned by optimizing the hyper-parameters. Since this is a computationally costly process, the hyper-parameter tuning is performed only on building the first Kriging prediction (after the initial DoE is evaluated), after which the parameters are held fixed for the remaining design search, accelerating the process of building the RSM. Even so, the inverse design method in this form is lengthy, requiring 150 objective function calls plus the effort required to build and search the RSM; this amounts to a computational cost per inverse design call of the same order as a full FLUENT drag calculation.

The aforementioned inverse design method is used to assess the uniqueness of the present design problem. The spanwise lift profile over the winglet of the geometry in Figure 5-11 (a) is used as a target

for inverse design. The resulting new geometry is shown in Figure 5-11 (b). It can be seen clearly that the geometry is rather different from the target geometry, and this confirms the suspicion that the mapping between the winglet $c_l \times c$ profile and the five gross winglet geometry variables is non-unique. What must be established is how the non-unique inverse mapping might affect the performance of EMFID. The differing geometry of two designs sharing the same flow feature is unimportant, provided that their performance in terms of drag is the same. Figure 5-12 shows the spanwise lift profiles, over the entire wing, for the two geometries in Figure 5-11. It is seen that the profiles match well over the winglet portion of the span, since this is the objective of the inverse design. However, there is a distinct difference between the profiles over the main wing, despite the geometry in this region being constant. Consequently, the drag calculated using FLUENT is different for the two designs; the target design (the best design in the 50 point DoE) has a $C_{D,i}$ of 135.7 counts at C_L=0.4, while the new design has a $C_{D,i}$ of 141.1 counts. Unfortunately, this difference in performance is detrimental if EMFID were to operate on a portion of the lift profile. For a given target $c_l \times c$ profile, there can in general be a number of solutions, each with different levels of drag. The resulting design search process would not be systematic, and would additionally lead to a rather tortuous response surface which is difficult to search.

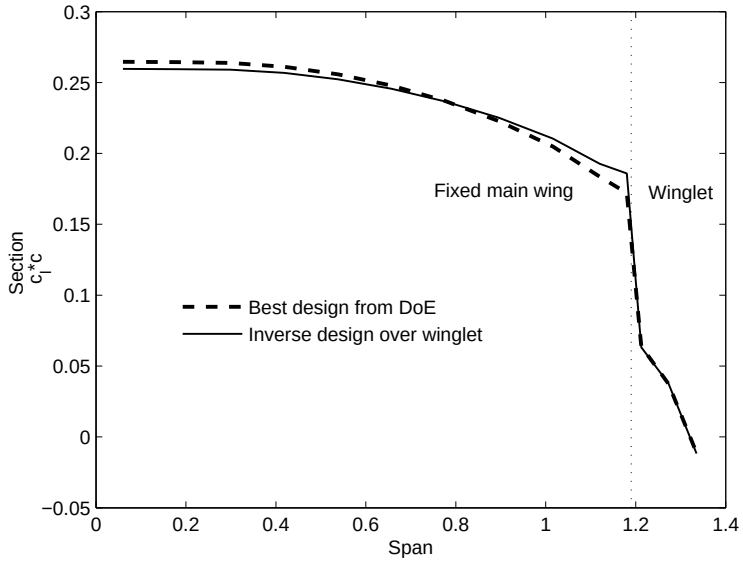


Figure 5-12 The spanwise lift profile for the best winglet design in the DoE. Also shown is the inverse design result when the winglet portion is used as a target.

Of course, the problems caused by the non-uniqueness could be eliminated by specifying the entire lift distribution as a target for inverse design. In this case, even if more than one solution exists for the inverse problem, because their lift profiles are the same, their downwash and therefore induced drag

would also be identical. However, it is thought likely that to parameterize the entire profile would require too many design variables, and for this test case any EMFID parameterization must have less than five variables in order to be competitive. Additionally, the inverse design method is prohibitively expensive. The optimization search takes around three hours, which is 50% of the time for an Euler FLUENT evaluation. The use of a scalar objective function means that the *information* stored in the shape of the lift profile is not used. The optimization approach does have the advantage that the data from all VSAERO calculations, in all preceding inverse design operations, in a design search can be appended to the data set, therefore accelerating the optimization. However, despite this, for the present application the inverse step is too costly, and the resulting EMFID design search would not be competitive with a simpler geometry-based method.

In summary, this section has evaluated the feasibility of a parameterization of the spanwise lift profile over the wing-tip device, when the geometry is described using the gross winglet parameters. It has been shown that, although this flow feature is reproduced with acceptable accuracy using both flow solvers, the non-unique inverse mapping from the flow feature to the five geometry variables prevents a meaningful design search process. This problem stems from the nature of the design variables; the five variables are distinct, and their effects can be made to cancel each other. For example, the effect of increasing the winglet dihedral may cancel the effect of reducing the twist angle or sweep. In the next section, the use of the spanwise lift profile is maintained but the geometry description is simplified, considering only a single design variable which is distributed in the spanwise direction.

5.6.4 Design of the Chord Distribution Using a Target Spanwise Lift Distribution

Thus far, the investigations into an appropriate flow feature for EMFID have made use of the gross wing design parameters shown in Figure 5-8. The representation of tip-vortex properties has been shown to be ineffective due to the incompatibility between the CFD solvers; predictions for the spanwise lift profile are in closer agreement, but the inverse mapping is non-unique. The present section investigates an alternative geometry description, for which the mapping is unique from the wing-tip device $c_l \times c$ profile to the shape.

For a given spanwise wing station, there are a number of parameters that can be used to change the locally generated lift. These include the local section incidence (twist), thickness, and chord length. In the geometry description used previously, all these parameters were modified at every station using a linear

variation along the span of the wing-tip device, and this lead to the non-uniqueness of the inverse problem. The present geometry description (introduced in Section 5.6.1) considers only a single variable, namely the chord, and the spanwise distribution of this chord is parameterized. This is somewhat akin to the 2-D airfoil shape representation, in which a local change in shape is directly related to the change in local surface pressure. In the same way, it is expected that a change in chord at a section is directly related to a change in the local lift, and thus a given wing-tip device $c_l \times c$ profile is uniquely mapped to a single chord distribution. In the present section, the uniqueness of this setup is demonstrated using an inverse design example.

The inverse design method adopted here for the determination of the chord distribution is similar to the method used by Qin et al. [2005], who focus on the twist distribution. The method is based on the assumption that the change in local section lift coefficient is linearly proportional to the change in local chord (or indeed, local incidence or camber). Thus, like the 2-D airfoil inverse method, there exists an approximation formula allowing local surface alterations to be made in an iterative *residual correction* fashion. The configuration of this inverse method is described in full in the following sections related to the setup of EMFID; in this section it is used to demonstrate the uniqueness of the present design problem. For this demonstration, a simple extension of the ONERA-M6 wing, with 20% additional span, is used as a *target* geometry. The spanwise lift distribution over this 20% wing extension is the target for the inverse design operation. The starting design for the process is the point where all six variables take their lower bound, i.e., a chord of 0.35m all the way along the wing-tip device, giving a trailing edge sweep equal to the leading edge sweep of 30°. The progress of this inverse design task is shown in Figure 5-13. While there are initially large disparities, after 27 iterations it can be seen that the process has converged onto both the target flow feature and the target geometry. The entire spanwise lift profile is matched closely, even though the target for inverse design is only the last 20% span of this curve. This indicates that the inverse mapping is unique, and further, it can be shown that different starting points and different targets all give a similarly unique result.

In summary, by using the distribution of chord as the geometry description a one-to-one mapping can be achieved relative to the wing spanwise lift profile; thus, a parameterization of the lift profile can be used to control the drag of the wing. This statement concludes the current investigation into an appropriate flow feature to parameterize in EMFID. The spanwise $c_l \times c$ profile over the wing-tip device is selected as the design flow feature for EMFID, while the chord distribution is chosen to describe the geometry of the wing-tip device. The setup of the benchmark and EMFID methods for the design of this 3-D component is the subject of the following sections.

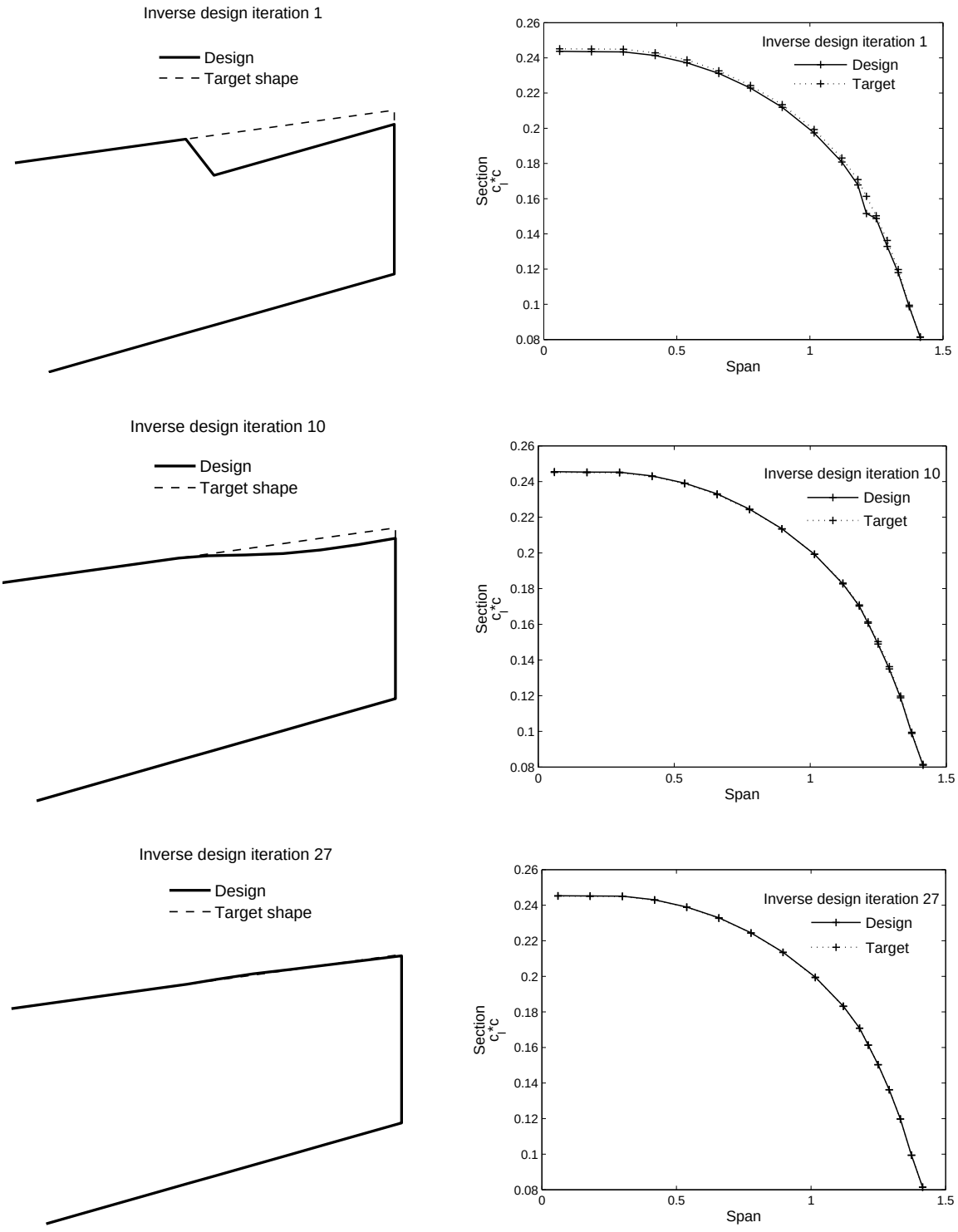


Figure 5-13 Design and target geometries, and corresponging lift profiles, after 1, 10 and 27 inverse design iterations.

5.7 Benchmark Configuration

The following sections of this chapter focus on the setup of the two design methods. As in the 2-D studies, a benchmark design search method is required which uses a more *traditional* geometry-based parameterization. The aim is to compare the performance of this approach with that of the EMFID method, which uses a parameterization of flow features coupled with inverse design. The benchmark design search method follows the same form as shown in Figure 3-6; for convenience this is shown again in Figure 5-14. The geometry of the wing-tip is parameterized, and for every iteration the high-fidelity CFD analysis is used to calculate drag; an optimization algorithm is used to automate the design iterations in the search for improved drag. In the wing design case studies given in the next chapter, design studies are performed using either Euler or RANS calculations as the high-fidelity CFD.

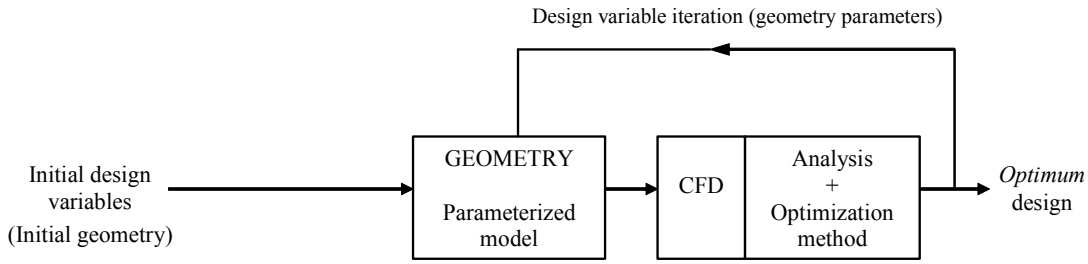


Figure 5-14 Flowchart illustrating the benchmark design search method.

In the present 3-D design problem, the same parameterization of the chord distribution is used both in the benchmark method and for inverse design in EMFID. As previously stated, the wing-tip device is planar, i.e., having zero dihedral, has a leading edge sweep equal to that of the main wing, and has a span of 20% of the span of the main wing. The chord distribution is discretized by specifying the chord at six equispaced stations along the wing-tip device, where one of these stations is at the wing tip. There are therefore six design variables for the benchmark parameterization. As the chord at each station is varied, the local airfoil section details are scaled proportionally. For the 2-D airfoil benchmark discretization, a cubic polynomial spline was used to represent the airfoil surface at the points not specified. In the present case, the trailing edges of the six spanwise stations are linked, via the GRIDGEN software, using a Catmull-Rom spline curve. Such a smooth definition is not a strict requirement for the wing trailing edge, as it is for an airfoil shape, but using this curve improves the quality of the resulting mesh and avoids the possibility of small vortices emanating from sharp corners. Figure 5-15 shows an example of the benchmark shape definition for an arbitrary set of variables; Figure 5-15 (c) shows the detail of the

interpolating curve. Note that continuity of the first derivative of chord is not imposed at the point where the wing-tip geometry joins the main ONERA wing, but this is certainly achievable with a suitable selection of the design variables.

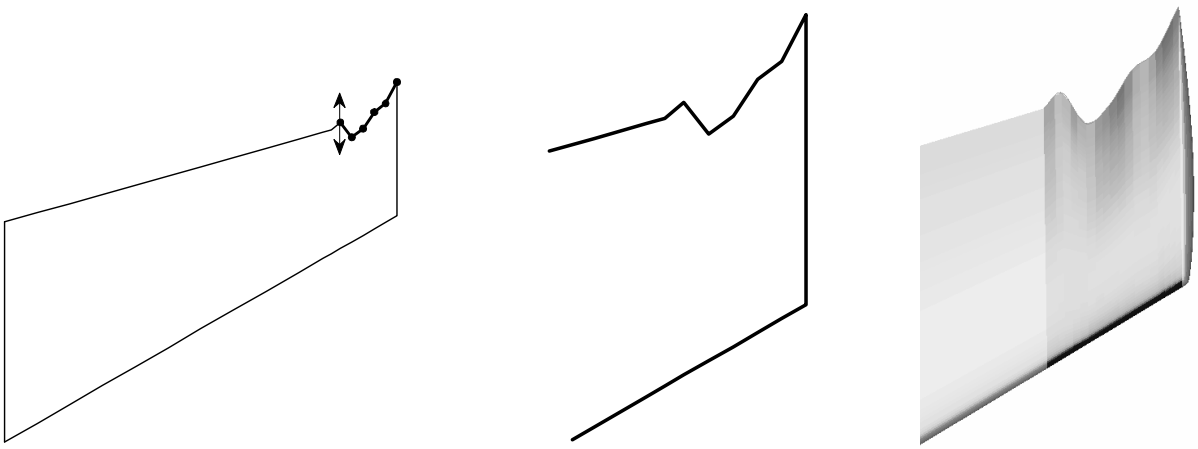


Figure 5-15 (a) Parameterization of the wing-tip chord distribution, (b) Discretization of the chord function and linear intepolation, (c) interpolation using a Catmull-Rom spline in GRIDGEN.

The optimization algorithm used for the following design studies is the method is response surfaces. The approach is similar to the benchmark optimization algorithm used in the 2-D airfoil design searches. The objective function is initially calculated for 60 designs, i.e., 10 times the number of design variables, as dictated by a Latin hypercube DoE plan. The design points in this initial database are selected such that the design space is populated efficiently, in theory allowing global optimization via a surrogate response surface model. The RSM used in this case is the Kriging technique (Jones [2001]), implemented using the OPTIONS design system. Kriging is used in preference to a RBF surface, since the number of design variables and objective function calls is small for the present 3-D design problem and this significantly reduces the expense of calculating the Kriging prediction. The Kriging hyper-parameters are tuned whenever new points are added to the dataset and a new RSM is required; since this contributes to a large proportion of the computational cost, a relatively quick multi-start simplex search is used for the optimization. Once the Kriging predictor is built, it is searched using a 5000 iteration GA, returning five cluster centroids from the final GA population which are used as update points. The true objective function (FLUENT drag) is calculated for the update points and the RSM is re-built, in theory providing a more accurate representation of the true objective function surface. The process of building, searching and

updating the RSM is continued until a fixed budget is reached. Objective function calculations which fail are not included in the RSM training data, unless a situation arises where all five update points fail, and in this case the first of these points is added to the training data and given an objective function value equal to the average of the objectives calculated so far.

5.8 EMFID Configuration

The EMFID design search strategy is shown in Figure 3-2, which is repeated in Figure 5-16 for convenience. The target flow feature for the present 3-D problem is the spanwise lift distribution over the wing-tip device, and the corresponding chord distribution is found using inverse design which utilizes the low-fidelity CFD, VSAERO. This inverse design process uses the benchmark chord profile parameterization (described in Section 5.7, above). As in the previous studies, in order to make fair comparisons between the benchmark and EMFID methods, the high-fidelity CFD analysis and implementation of the optimization algorithm are the same in both cases. Therefore, the RSM approach described for the benchmark strategy is also used as the optimization method for EMFID. The differences in the EMFID approach are the parameterization technique and use of inverse design; these items are detailed in the following sections. Appendix F gives further detailed information about the computational setup of this EMFID search, including a detailed flowchart diagram depicting the optimization strategy.

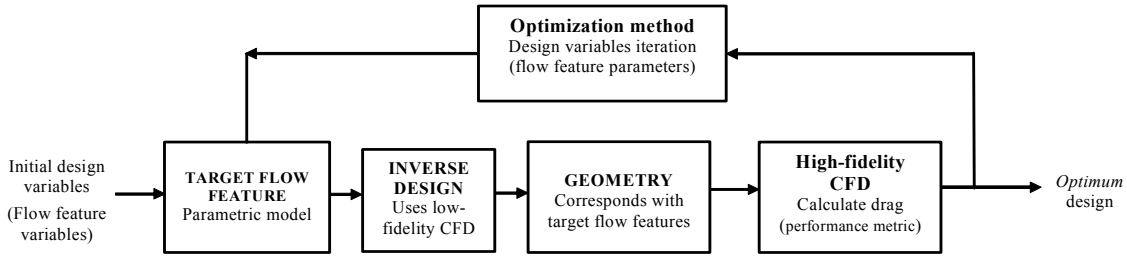


Figure 5-16 Flowchart illustrating the EMFID parameterization and design search process.

5.8.1 Parameterization Techniques

In previous work involving the representation of the spanwise lift distribution, an elliptic profile has commonly been employed, but such a shape is non-optimal in many cases. Alternative functions have been shown to be effective, such as the triangular or hybrid elliptic-triangular curves proposed by Qin et al. [2005]. Here, the requirement is for a function which can be used to represent a small portion of the end of the lift profile, and such a function should be defined using very few variables. The number of variables should be five or less, since the purpose of EMFID is to reduce the dimensionality of the design space compared to the benchmark method, which uses six variables. Strictly, the section c_l at the tip of the wing should be zero, and, as in the case of an ellipse, the lift profile function should feature a singularity at this point. However, the VSAERO code deals with discrete surface panels which are relatively large in the spanwise direction, and since the flow variables are calculated at each panel center the lift on the wing at the tip station is not calculated. Therefore, the point of zero lift and singularity do not need to be modeled, and the parameterized curve can end at the position of the last panel center on the wing surface. From the outset, it is not obvious what form the optimum lift profile assumes. However, it is likely that it can be described using a function which is smooth and decreases the lift monotonically with span. For simplicity, polynomial curves are employed.

In this work, two parameterization techniques are used and compared. One is a quadratic curve defined using three variables, and one is a cubic polynomial utilizing four variables. The lift profile parameterization starts at Y_1=1.19m, i.e., the ONERA-M6 wing tip station, and ends at Y_2=1.43m, i.e., the position of the last VSAERO panel centroid for a total span of 120% of the M6 span. The quadratic and cubic curves are determined by specifying the $c_l \times c$ values of three and four points, respectively, which are equispaced between y_1 and y_2 and are interpolated by the curve in each case (Figure 5-17). The $c_l \times c$ values are the profile design variables, and hence for EMFID there are three and four variable parameterizations. Since only the wing-tip portion of the lift profile is specified, the loading on the main wing can float depending on the wing-tip loading. The parametric quadratic or cubic lift curve for the wing-tip is used as a target for the inverse design procedure, which yields the wing-tip chord distribution while keeping the total wing lift constant at C_L=0.4. This inverse step is described next.

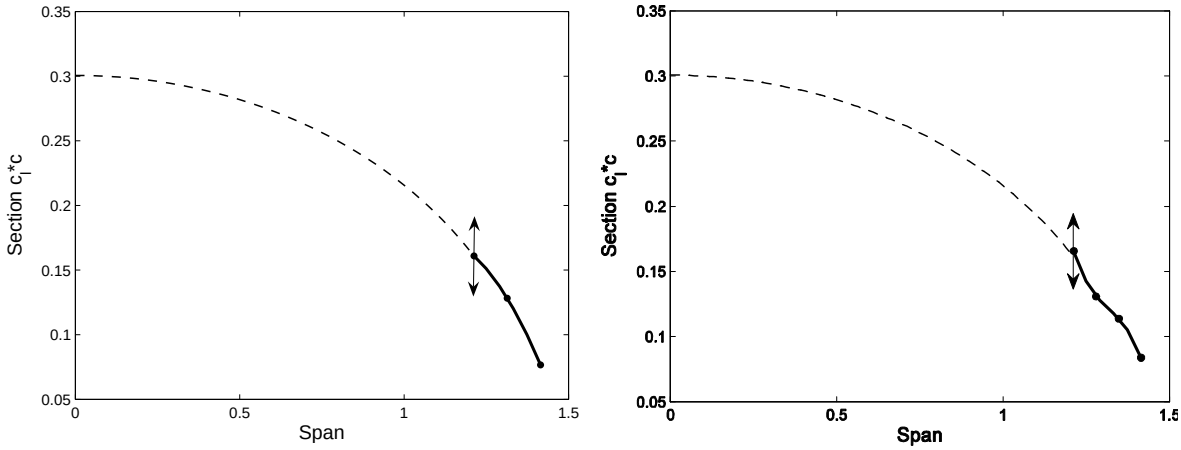


Figure 5-17 Parameterizations of the wing-tip device lift profile using (a) a quadratic and (b) a cubic polynomial.

5.8.2 Inverse Design

The inverse design method employed in the 3-D EMFID design searches makes use of the same wing-tip device geometry description as used for the benchmark method (Section 5.7). However, this need not necessarily be so; for EMFID the detail afforded by the geometry description is not limited, since the geometry is not parameterized. This is why, in the 2-D EMFID searches, the geometry description was a series of 206 co-ordinate pairs which were perturbed directly by the inverse process. Conversely, in the benchmark airfoil design search, a small number of discrete surface points were perturbed in order to give a reasonable number of design variables, which necessitated an interpolating function to define the airfoil at all other points. However, for the present 3-D inverse design method, a finely detailed geometry description for the wing-tip device is not necessarily desirable since the number of defining points is translated into the number of CFD panel nodes for the VSAERO analysis. Increasing the number of surface panels raises the computational cost of each VSAERO analysis, and this can be significant if many CFD calls are required for convergence of the inverse design process. In fact, a chord distribution with six points across the wing-tip gives a sufficiently narrow panel width in the spanwise direction, and for this reason the benchmark chord parameterization is used during this inverse design. Also, using the benchmark geometry description facilitates a fair comparison between the benchmark and EMFID methods, since the variables bounds can be made the same in both cases.

The 3-D inverse method is similar to a method used by Qin et al. [2005], and uses a residual correction approach comparable to the 2-D inverse method. Since it uses a linearized surface alteration scheme, it is significantly more rapid, and more capable of matching the target lift profile, than the optimization approach to inverse design used in Section 5.6.3 above. The method makes the assumption that an increase in section chord is linearly proportional to an increase in sectional lift. Thus, at iteration k of the inverse process, the chord, c, at a given section is adjusted according to the simple expression

$$c_k = c_{k-1} + \phi\, c_{k-1}(c_l^t - c_l^{k-1}), \tag{5.3}$$

where c_l^t is the target lift coefficient at the given spanwise station and c_l^{k-1} is the lift coefficient at this station computed in iteration k-1. The scalar ϕ is a relaxation factor which takes a value of 3; this magnitude is established following a process identical to that described in Section 3.7.3. Eq. (5.3) is applied to each of the six spanwise stations at every iteration of the inverse process. The starting design is a chord distribution which is a simple 20% span extension of the sweep and taper of the ONERA-M6.

As previously stated, the present inverse method provides a solution for a fixed total wing lift of C_L=0.4, since this is the design C_L at which the drag is calculated using FLUENT. Since only the wing-tip geometry is altered, changes in lift and angle of attack between designs are small, but in order to guarantee that the target lift is achieved, the angle of attack, α, of the VSAERO analysis is iterated at each iteration using the expression

$$\alpha_k = \alpha_{k-1} + \theta(c_l^t - c_l^{k-1}), \tag{5.4}$$

where θ is a relaxation factor which is assigned a value of 2°.

For the inverse design approach described above, a number of (rather heavily relaxed) iterations are required because of the crude nature of the linear local approximation used for alterations. The measure of convergence used is the sum of the square of the differences between the six target and six computed section c_l's. The criteria for convergence is the point when this measure falls below 10^{-7}. Convergence in this sense is usually achieved in less than 100 iterations; if 100 iterations are performed then the inverse process is terminated and the design giving the closest matching lift profile, of all 100 iterations, is used as the final result. However, if an inverse design convergence metric of less than 10^{-6} is not achieved, the process is deemed to be insufficiently converged, and the FLUENT drag calculation is not performed for the resulting design. This measure is taken to avoid calculating the objective function for a geometry

whose $c_l \times c$ profile does not correspond to the specified design variables, as this would result in inaccurate RSM predictions.

5.9 Comparing the Two Methods

The preceding sections of this chapter have detailed the setup of the benchmark and EMFID methods for the design of the geometry of a wing-tip device. The next chapter describes a number of case studies which compare the relative performance of these methods. As with the 2-D case studies, several measures are implemented here in order to achieve as fair a comparison as possible.

The initial design, i.e., the first of the initial DoE evaluations, is ideally made to be the same for both methods. This initial point is chosen to be a simple extension of the ONERA-M6 wing, with a linear chord profile. In order to reproduce this initial shape using the EMFID parameterizations, a set of lift profile variables must be found such that the corresponding geometry, resulting from inverse design, closely matches this extended ONERA wing. The lift profile for the true initial shape is calculated using VSAERO and the quadratic and cubic curves are fitted to this in a least squares sense; this determines the required variables. The inverse design process is run using each of these fitted lift profiles as a target; the resulting geometries are shown in Figure 5-18. As expected, there are disparities between the desired ONERA-M6 extension and the designs achieved by the EMFID parameterizations, but these differences are relatively small.

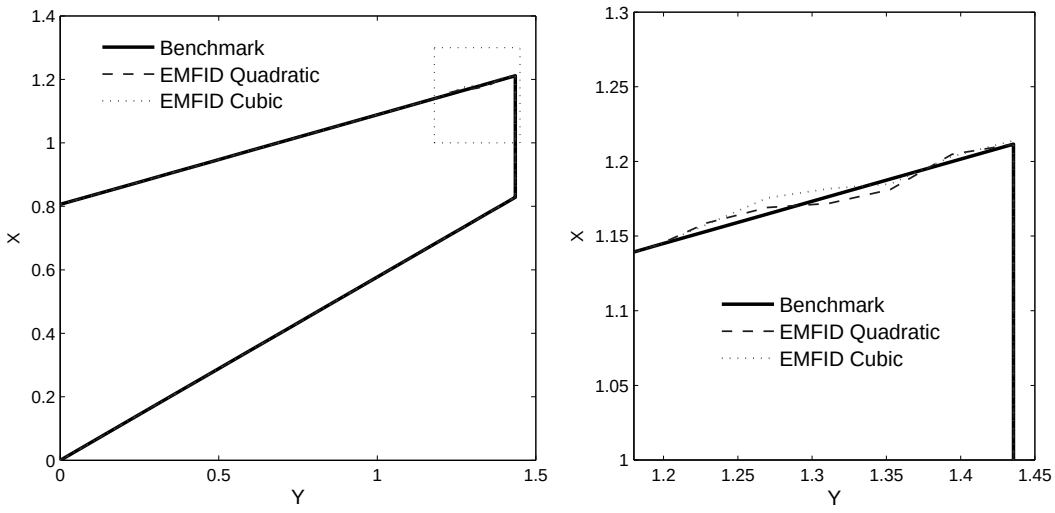


Figure 5-18 Comparison of initial geometries used in the design searches, showing the benchmark parameterization and representations of this using the two EMFID parameterizations.

The two design methods are run for an equal total computational expense. The CFD simulations for the 3-D case studies described in the next chapter are run using the Microsoft compute cluster, for FLUENT calls, and a Linux cluster for VSAERO. In the case of the benchmark method the computational effort is considered to be expended entirely in the FLUENT drag calculations, which require, on average, 6.64 hours for an Euler solution and 16.59 hours for a RANS solution when running on two four-processor compute nodes. An objective function evaluation in EMFID additionally requires effort for the inverse design. A call to VSAERO takes approximately 35 and 41.5 seconds when using the inviscid and viscous configurations, respectively, and running on a Xeon 2.8GHz Linux node with 2Gb memory. An average of 84 VSAERO calls are required for the inverse design process, giving a total time of 49.8 and 58.1 minutes for the inviscid and viscous runs, respectively. Therefore, the ratio of computational expense for the EMFID and benchmark methods is 1.12:1 when Euler simulations are used, and 1.06:1 for RANS simulations. In addition to this, for either design method, if a design point causes errors in the computational mesh, the design is treated as a failed point and the cost of running FLUENT for this point is not taken out of the total computational budget. Equally, if the sum-of-squares error in an EMFID inverse design solution is greater than 10^{-6}, the inverse process is deemed to be insufficiently converged and the design is treated as a failed point; in such cases the objective function cost is that of the inverse design only, and does not include the expense for running FLUENT.

It is fundamentally important that the two methods are run with the same size of design space. Since the EMFID inverse design step uses the benchmark parameterization as its geometry description, the methods can be constrained via the same common variables. The chord at each spanwise station must lie between

0.35-0.5m. The EMFID inverse design process may result in a geometry in which the chord at one or more of the six stations is outside of this required range. In such a case, the offending chord values are *repaired* such that they equal the upper or lower bound, depending on which is nearer to the original value.

As observed in the 2-D airfoil studies, the location in the design space of points in the initial DoE seed has a significant effect on the convergence of the design search. As before, each method is run five times, using a different initial DoE set at each attempt. In doing so, the dependence on the DoE seed is reduced and a greater understanding of the level of convergence is established.

In summary, this chapter has introduced the area of wing-tip device design and has described the setup of the EMFID and benchmark methods for this 3-D application. In the next chapter, the relative performance of these parameterization strategies is demonstrated using the results from two case studies.

Chapter 6. Application of EMFID: Case Studies for 3-D Wing Design

6.1 Introduction

In this chapter, the concept of flow feature parameterization is applied to the design of a wing-tip device for the ONERA-M6 wing. The setup of the EMFID and benchmark methods for this 3-D problem is described in the preceding chapter. The objective is to minimize the total drag force by varying only the chord distribution of an extension of the standard ONERA wing. The benchmark and EMFID methods are compared for an equal computational expense. The benchmark method uses a geometry-based parameterization of the chord distribution, while the EMFID strategy uses a parameterization of the wing lift (or specifically, $c_l \times c$) distribution and inverse design is used to determine the chord profile. Two case studies are considered here, which differ in the high-fidelity CFD formulation used. The first case uses the solution to the Euler equations, and the second uses a RANS approach for solving the Navier-Stokes equations. In both cases, the flow Mach number is M=0.7 and the design lift coefficient is C_L=0.4, eliminating the possibility of shocks and the associated wave drag component. Therefore, the Euler case computes only induced drag while in the RANS case the drag includes the induced, skin friction and profile components. As in Chapter 4, for each case study key results are listed in the form of figures and tables. Following this, the results of the work are analysed and conclusions are drawn.

6.2 Case 5

In this first 3-D case study the high-fidelity objective function is calculated using Euler simulations. The two design methods are run side-by-side, using a starting design which closely approximates a simple extension of the ONERA-M6 wing of 20% additional span. The benchmark process employs the six-variable Catmull-Rom spline model, while EMFID is run for both the three-variable quadratic and four-variable cubic discretizations of the $c_l \times c$ profile; these parameterizations are detailed in Sections 5.7 and 5.8.1. Therefore, this case study effectively considers three parameterizations: the benchmark and two EMFID variants. Each of these methods is run five times using a different initial Latin hypercube DoE seed at each attempt, and using an equal computational budget of 120 calls to the FLUENT analysis. To begin with, the methods calculate the objective function at $10d$ points as dictated by a Latin hypercube DoE, where d is the number of problem dimensions. The benchmark method therefore has a DoE size of 60, and evaluates an additional 60 design points in updating the RSM. In the EMFID process, the DoE consists of 30 and 40 points for the quadratic and cubic parameterizations, respectively, and the remaining budget is used to evaluate update points. Because the FLUENT analysis is not run for design points for which the inverse design is insufficiently converged, the budget used in evaluating the DoE points, and also the budget expended in the update phase, is not fixed. However, because of the additional expense of the inverse design, it is known that the maximum number of EMFID objective function calls permitted by the computational budget is 107 (compared to 120 in the benchmark process).

The optimization-iteration histories for the two methods are shown in Figure 6-1, showing the design objective (drag) plotted against the number of iterations for the benchmark method. The EMFID traces have been scaled for each design point according to its computational cost; the cost of those design points for which the FLUENT analysis is not run is only the average cost of running the inverse design. Thus, the abscissa can be viewed as computational expense. At first glance, it appears that the two EMFID parameterizations are able to converge more quickly than the benchmark method, and all design searches settle at a very similar level of performance. A number of observations are reported in what follows.

First, as has been previously shown in the 2-D case studies, the EMFID method is able to find better performing wing-tip designs in the initial DoE phase than the benchmark process. This is likely to be because the EMFID parameterizations of the $c_l \times c$ profile are inherently smooth, while the benchmark geometry description can result in *rippled* designs. This is an advantage of the EMFID method, and was also found to be the case in the 2-D airfoil design studies, as described in Section 4.6.1.

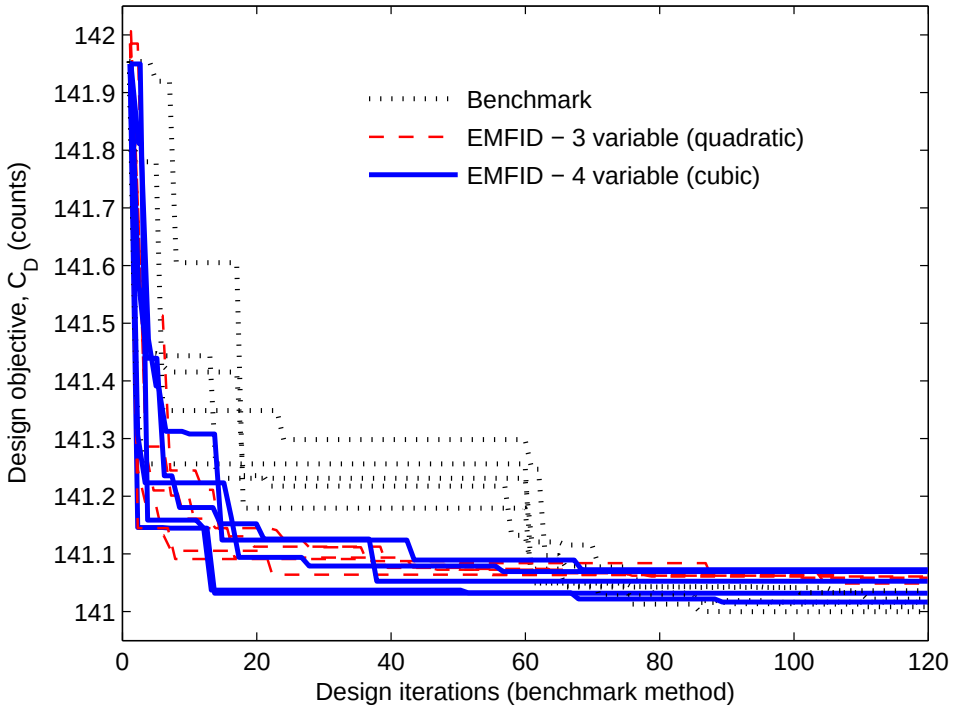


Figure 6-1 Optimization-iteration histories for the benchmark and EMFID methods, showing traces for the three- and four-variable EMFID computations. Drag is calculated using the Euler FLUENT analysis.

Second, in the case of the benchmark method once the DoE phase is complete the selected update points very quickly yield high performing designs, and this explains the sudden drop in drag at just after 60 iterations. Consequently, it is clear that the RSM optimization could proceed successfully with fewer than $10d$=60 DoE points, and that using fewer DoE points could allow the benchmark method to converge faster, perhaps as effectively as EMFID. However, it is true to say that prior to beginning a conceptual design study, the complexity of the objective function landscape and therefore the density of points required to efficiently populate the design space are not known from the outset. Hence, $10d$ points have been used in the DoE, following the recommendation of Jones et al. [1998]. Conversely, having found efficient designs in the DoE phase, the EMFID searches find little improvement during the RSM update phase.

Third, comparing ultimate performance, the benchmark method generates the wing-tip design with the lowest drag. Further, two of the benchmark searches produce a better design than any of the EMFID computations produced. Therefore, it appears that the EMFID parameterization is unable to generate the low drag shapes that are possible with the higher-dimensional benchmark model. However, the range of drag values for all final designs is less than 0.1 drag counts. Indeed, the total design improvement

achieved by the best design, starting from a simple extension of the ONERA-M6 wing, is approximately 1 drag count. Because the chord function modifications are only performed in a small region close to the wing-tip, the drag reductions are relatively small; this is despite the dependence of induced drag on the wing-tip geometry. Such small gains would be equally encountered if one were to vary twist or airfoil section details over the same span extension, and this perhaps highlights the marginal advantage that is afforded through the use of wing-tip devices. However, as previously mentioned, the designer generally does not know the outcome of the design process until it is undertaken. Understanding the potential improvement to be gained by certain modifications is one of the purposes of such systematic design procedures. Since the drag convergence of the design methods is within a fraction of a drag count, it is also likely that the results are affected by the accuracy of the FLUENT solver. However, the results in Figure 6-1 suggest that the benchmark method is more likely to achieve a better design by the end of the search process.

The fourth and final observation is related to the level of convergence achieved by the five runs using each of the three parameterizations. Table 6-1 gives the drag coefficient values for the five final designs generated using the benchmark and EMFID methods, i.e., for the designs at the end of the search process shown in Figure 6-1. Although Figure 6-1 shows that the convergence rate of the two EMFID variants is very similar, a closer level of convergence, i.e., a smaller range of final objective values, is attained using the three-variable parameterization than using the four-variable model. This is to be expected since the extra variable in the cubic representation increases the complexity of the search space. However, this four-variable method results in a greater spread of final design objectives than the six-variable benchmark method; this is unexpected behaviour. Recall that the EMFID geometries are *repaired* if they lie outside the bounds of the benchmark search space, while the benchmark designs undergo no such constraints. It is likely that this action disadvantages the EMFID optimization in its search for improved designs. Despite this, as previously discussed, the relative level of objective values achieved by the end of the design search processes is similar for all the traces in Figure 6-1. For comparison, the FLUENT drag coefficient for the original M6 wing with no wing-tip device is 168.6 counts. As expected, this is higher than the drag of the 20% extended M6 geometry which is used as the initial design in the search process.

Table 6-1 Design objective (drag coefficient calculated using FLUENT Euler simulations) for the five best designs resulting from the benchmark and EMFID methods.

	FLUENT (Euler) C_D, counts		
Best design from run	Benchmark	EMFID - 3 variables	EMFID - 4 variables
1	141.008	141.061 (max)	141.032
2	141.022	141.053	141.017 (min)
3	140.999 (min)	141.053	141.053
4	141.036 (max)	141.046 (min)	141.070
5	141.016	141.059	141.073 (max)
Initial design	141.954		
ONERA-M6 (no wing-tip device)	168.557		

Comparing the computed wing-tip designs, Figure 6-2 shows the best five geometries resulting from the five runs of the benchmark method. All of these designs feature a highly swept back trailing edge compared to the main wing; indeed, the chord at the tip station has in all five cases been maximized within the search bounds. Further analyses have been performed which verify that this aft-swept tip reduces drag; this study is documented in Appendix D. However, moving inboard, rather than converging onto a single design there appears to be two modes of *optimum* designs. Two of the five searches have resulted in an almost linear trailing edge up to the 1.4m span station, with a sweep very similar to that of the main wing. These runs have therefore brought about very little change to the starting design in this inboard region. Two other designs, including the best design of these five, are also modified little from the initial shape up to Y=1.4m, although these shapes are less linear. Finally, one of the searches has resulted in a rather radical geometry, exhibiting a *hook*-like shape with small chord values in the inboard region and the maximum allowable chord at the tip station.

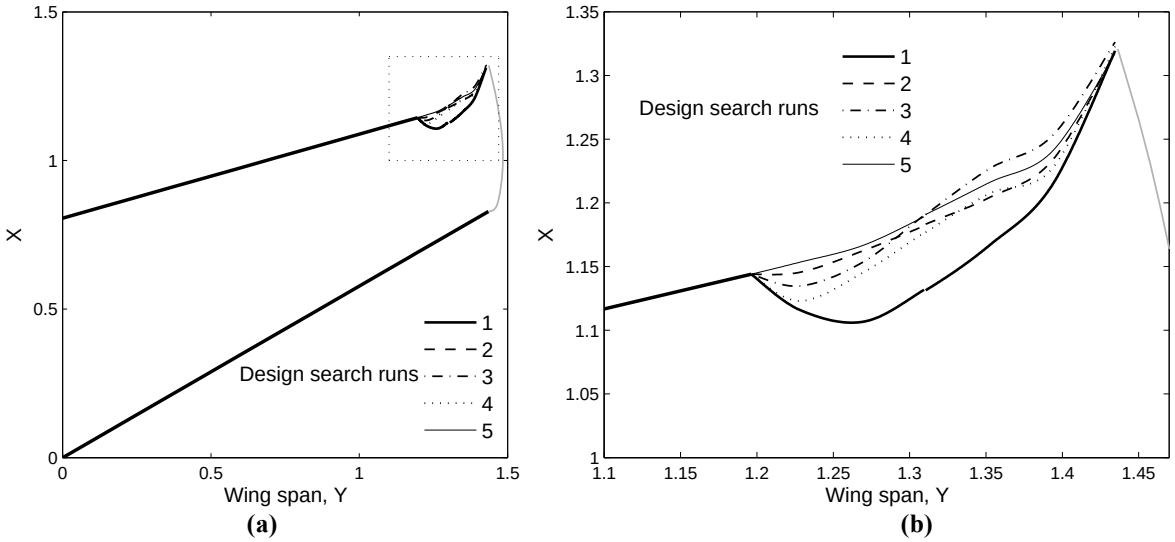


Figure 6-2 (a) Planform view of the best geometry resulting from each of the five benchmark computations (shown on equally scaled axes). (b) A close-up view of the wing-tip region.

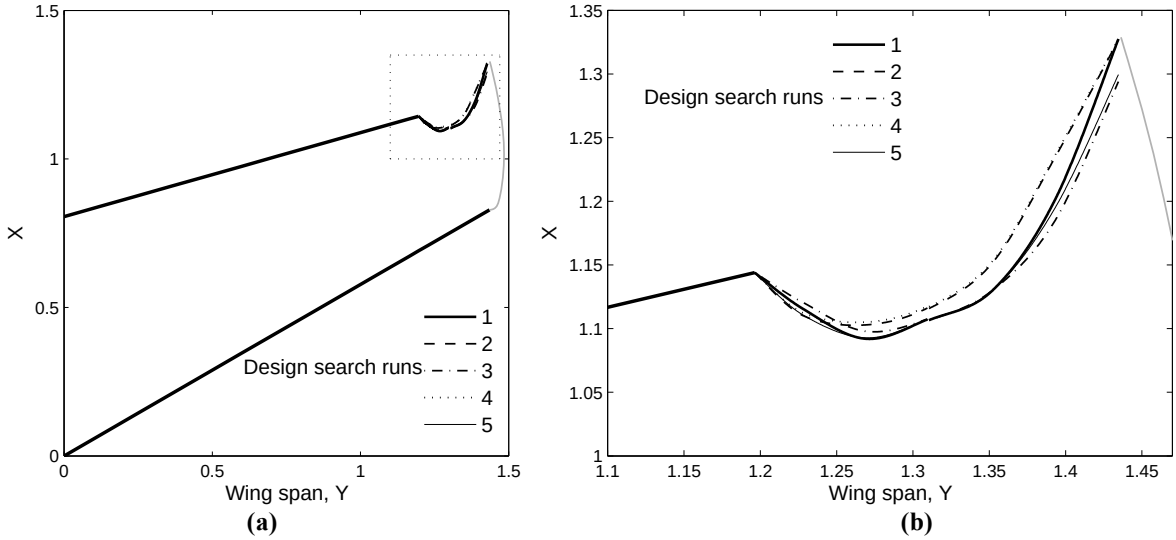


Figure 6-3 (a) Planform view of the best geometry resulting from each of the five EMFID computations using the three-variable (quadratic) parameterization (shown on equally scaled axes). (b) A close-up view of the wing-tip region.

Figure 6-3 presents the five best geometries resulting from the EMFID computations using the three-variable parameterization of the $c_l \times c$ distribution. Table 6-1 gives the corresponding drag coefficient values. All of these designs exhibit the distinctive *hook*-like shape, and there is strong similarity between them. Due to the smaller dimensionality of this parameterization, the region of the design space

containing the optimum is more readily searched, resulting in more advanced convergence. However, it is likely that this parameterization is unable to perform the complex changes that are possible with the direct (benchmark) representation of the chord profile. Consequently, better designs are generated by both the benchmark process and the four-variable EMFID parameterization.

The final geometries resulting from the EMFID design searches using the four-variable (cubic curve) parameterization are shown in Figure 6-4, and their drag coefficient values are listed in Table 6-1. These designs are rather different to those shown in Figure 6-2 and Figure 6-3. Again, all designs feature a highly aft swept tip section, but in this case four of the five shapes exhibit an inboard *bump* in the chord distribution. It is interesting to note that this bump was not produced by the two previously described parameterizations. It is known that the cubic polynomial model of the $c_l \times c$ profile is able to generate geometries similar to those shown in Figure 6-2 and Figure 6-3, and yet the design searches have converged, in four out of five cases, onto a radically different shape. Although two of the benchmark designs produce marginally lower drag, these designs all perform comparably. Of course, if these designs were optimal then it is likely that the benchmark process would also have converged to give these shapes. However, this result shows that EMFID is able to generate intricate geometries using a four-variable parameterization of the $c_l \times c$ profile.

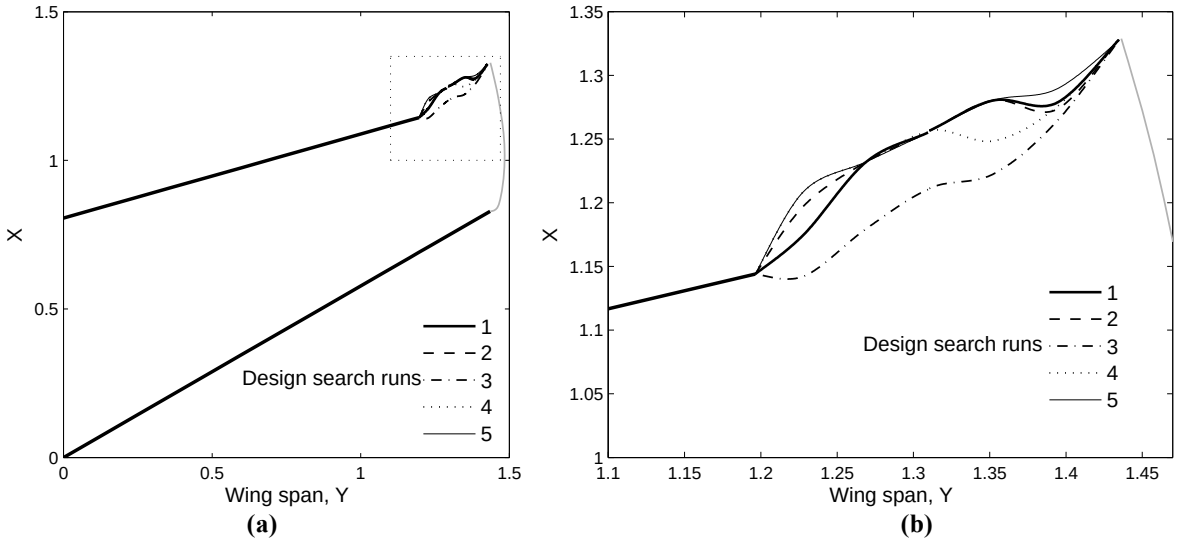


Figure 6-4 (a) Planform view of the best geometry resulting from each of the five EMFID computations using the four-variable (cubic) parameterization (shown on equally scaled axes). (b) A close-up view of the wing-tip region.

The wing-tip chord distributions designed in this case study have resulted in an improvement in the wing drag coefficient of approximately 1 count. The overriding feature of all of these designs is the highly aft-swept tip, and this supports the conclusions of van Dam [1987] and Burkett [1989] who claim that aft-sweep can give a reduction in induced drag. However, while the aft-swept tip seems to be strongly related to drag, the more inboard chord function appears to have a more complex relationship with drag. Figure 6-5 compares the best geometries generated using the benchmark and two EMFID parameterizations, illustrating the clear distinction between them. There are two possible explanations for the range of geometries seen in this case study. First, there could be a number of basins of attraction, i.e., local optima, in the design space containing similarly performing designs in terms of drag. The three parameterizations have varying abilities to exploit these local optima; in particular the EMFID three- and four-variable methods seem to favour *hook*- and *bump*-like distributions, respectively. Despite this, it is interesting to note that the two EMFID parameterizations produce geometries which are similar to two of the benchmark designs (Figure 6-5). Second, the gradient of the objective function, with respect to the inboard chord variables, is likely to be rather shallow. This leads to a large area of the design space containing designs with similar performance. In this case, the induced drag has a low dependence on the inboard chord function.

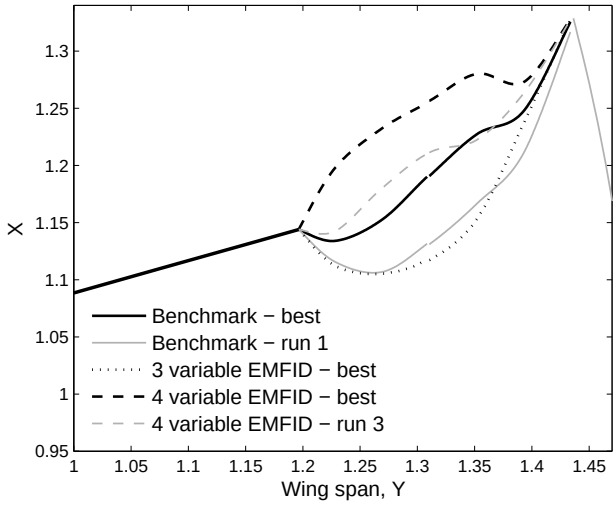


Figure 6-5 Comparison of the best designs generated using the benchmark and EMFID parameterization methods.

This case study has compared the benchmark and EMFID methods for the design objective of minimizing the drag calculated using Euler simulations, i.e., induced drag. As previously demonstrated, the reduced dimensionality of the EMFID parameterizations allows promising designs to be found during the DoE phase, and this helps to accelerate the convergence of the design search. Although the benchmark method

generated the lowest drag design, all design searches converged to a similar level of drag. An aft-swept tip station is clearly associated with lower induced drag, but the inboard chord distribution is not converged, and it is likely that the geometry of this region has a much smaller effect on induced drag. However, this region is more likely to influence the drag due to viscosity, and therefore the design search methods are next run using RANS simulations.

6.3 Case 6

In this case study, the high-fidelity objective function, drag, is calculated using FLUENT RANS simulations. As in Case 5, the benchmark and EMFID methods are run side-by-side starting from a geometry equivalent to a 20% extension of the ONERA-M6 wing. However, since the computational cost of the RANS simulations is significantly higher than Euler computations, only the four-variable cubic curve is used as a $c_l \times c$ profile parameterization and the quadratic model is not employed. The benchmark process uses the six-variable Catmull-Rom spline representation of the chord distribution as before. In case study 5, it was observed that the benchmark method finds significantly improved designs immediately upon starting the update phase, and therefore may benefit from using a smaller database of DoE points. Hence, in this case study a reduced DoE size of $5d$=30 points is used for the benchmark method, where d is the problem dimensions. The total budget of objective function calls is 60, allowing an additional 30 update points to be evaluated. Equivalently, the EMFID method uses a DoE size of $5d$=20 points, with the remaining budget used to evaluate update points. Recall that the FLUENT drag calculation in not performed for design points which result in insufficient convergence of the inverse design process. As in Case 5, since the budget expended in the DoE phase is not fixed, the number of objective function calls in the update phase can vary, but it is known that the maximum number of objective function calls which include the FLUENT analysis is 57 (compared to 60 in the benchmark process).

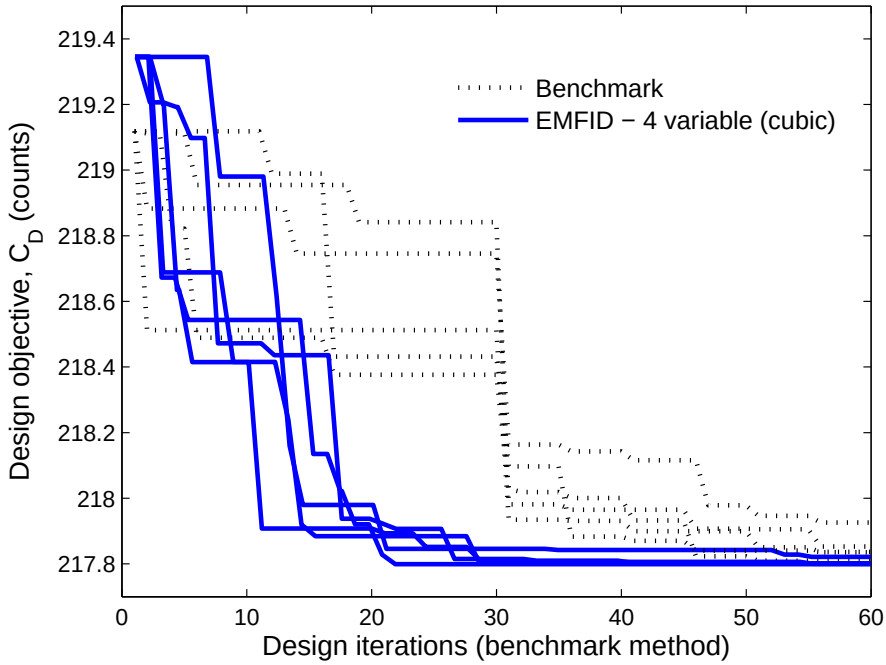


Figure 6-6 **Optimization-iteration histories for the benchmark and EMFID methods. Drag is calculated using the RANS FLUENT analysis.**

The optimization-iteration histories for the present comparison are shown in Figure 6-6. As in Figure 6-1, the EMFID search traces have been scaled according to the computational expense of individual design points. In particular, the expense of the EMFID design points for which the FLUENT analysis was not run is only the average cost of running the inverse design. This allows the abscissa to be viewed as computational cost. As has been previously observed, the EMFID search method is able to converge more quickly than the benchmark method, and this is due to the lower dimensionality of the $c_l \times c$ profile parameterization. The EMFID search process is able to find better performing geometries in the initial DoE phase because the smooth nature of the $c_l \times c$ profile prevents the *rippled* geometries which can occur using the benchmark parameterization. As a result of the relatively poor performance of its DoE points, the benchmark method again shows a sharp reduction in drag corresponding to the start of the update phase. The method may benefit from a still smaller initial database, but this may result in insufficient exploration of the design space. The improvement in drag coefficient as a result of the new designs is approximately 1.5 counts, and both design methods converge to a similar level of drag. However, the EMFID searches show a noticeably closer level of convergence, i.e., a smaller range of final objective values, than those of the benchmark method. This is the expected behaviour, given that the EMFID method uses fewer variables. Recall that this result was not observed in case study 5; this reaffirms the notion that calculating the drag using Euler simulations, as in Case 5, results in an objective function

landscape with many local and shallow optima and which is difficult to search, and where various designs have similar drag. For the DoE seeds used here, the EMFID method generates the wing-tip geometry with the lowest drag overall; Table 6-2 gives the drag coefficient values for the five best designs from the benchmark and EMFID methods. For comparison, the original ONERA-M6 wing corresponds to a drag coefficient of 235.9 counts.

Table 6-2 Design objective (drag coefficient calculated using RANS FLUENT) for the five best designs resulting from the benchmark and EMFID methods.

	FLUENT (RANS) C_D, counts	
Best design from run	Benchmark	EMFID - 4 variables
1	217.926 (max)	217.822 (max)
2	217.808 (min)	217.799 (min)
3	217.835	217.801
4	217.853	217.801
5	217.836	217.800
Initial design	219.118	
ONERA-M6 (no wing-tip device)	235.979	

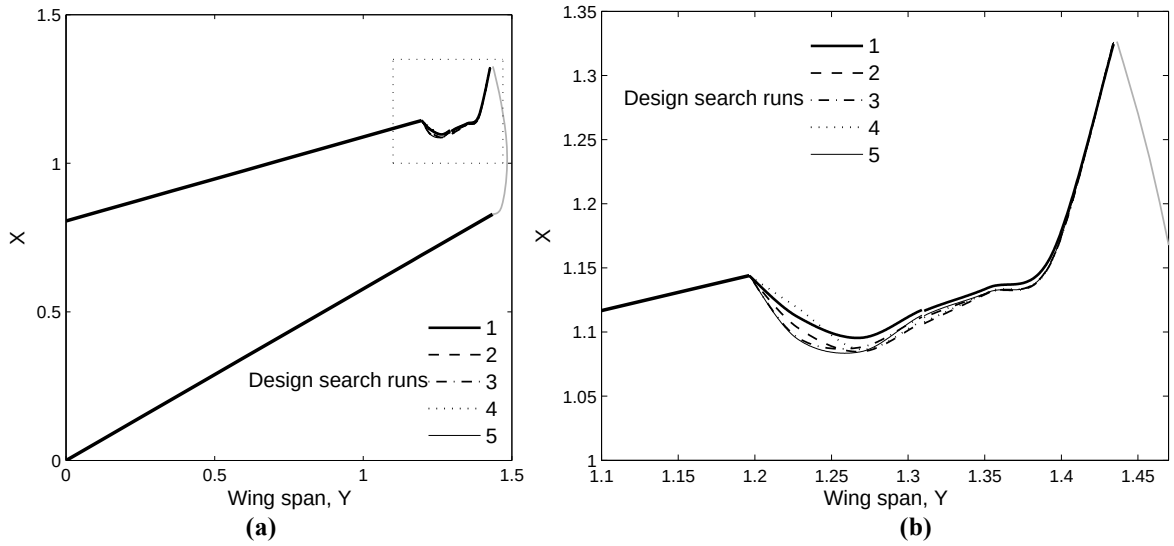


Figure 6-7 (a) Planform view of the best geometry resulting from each of the five benchmark computations (shown on equally scaled axes). (b) A close-up view of the wing-tip region.

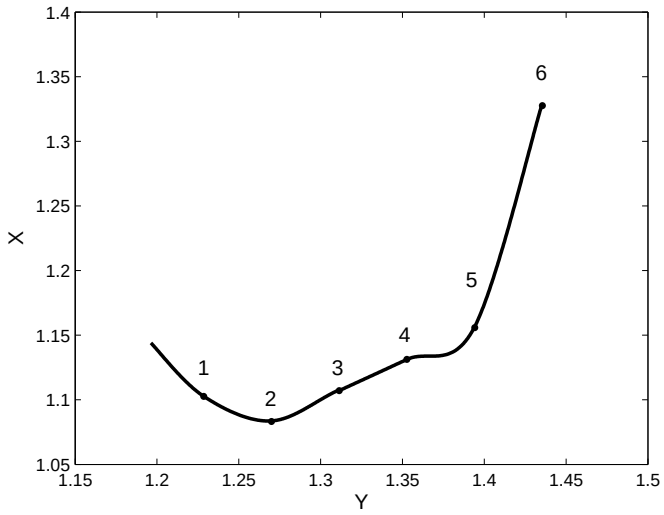


Figure 6-8 **Illustration of the best chord distribution resulting from the benchmark method, showing the control points and the interpolating Catmull-Rom spline.**

Figure 6-7 displays the best wing-tip chord distribution generated by each of the five runs of the benchmark method. It can be seen that these geometries are closely converged, indicating a clear design trend for reduced drag. This is in contrast to the results from case study 5, in which a range of inboard geometries were encountered. In all cases the tip station has been highly aft-swept in an attempt to minimize the induced drag (see also Appendix D). In addition, all of the design searches have identified a low wetted surface area with reduced viscous drag, and hence the inboard section has a much smaller chord giving the distinctive *hooked* shape. Indeed, the chord at the tip section has been maximized within the problem bounds, while the chord in the inboard region has been minimized. Figure 6-8 shows the Catmull-Rom spline interpolating the control points for the best benchmark design; the chord has been minimized at points 2, 3, 4 and 5, and maximized at point 6. Figure 6-8 also explains why there is a *bump* in the chord distribution between points 4 and 5: the nature of the Catmull-Rom spline means that the gradient of the curve at point 5 must be equal to the gradient of the line between points 4 and 6. Despite the unusual shape, this bump further reduces the wing wetted area, and therefore (rather fortuitously) gives lower drag. Interestingly, the chord at the furthest inboard control point (point 1 in Figure 6-8) has not been minimized, but its position gives a gentle curve up to the end of the main wing; this is investigated further in Appendix D.

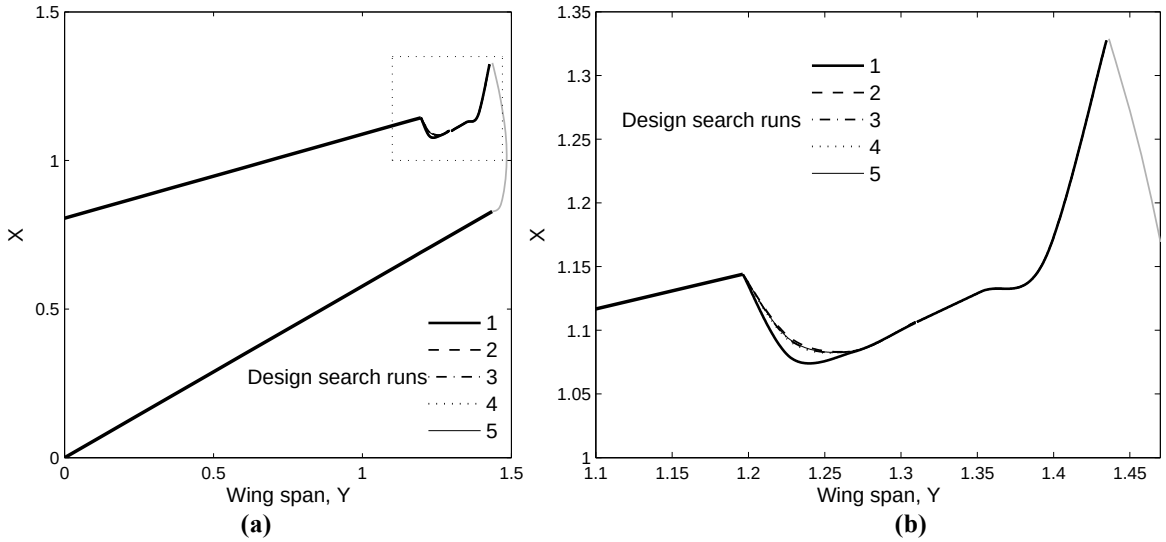


Figure 6-9 **(a) Planform view of the best geometry resulting from each of the five EMFID computations (shown on equally scaled axes). (b) A close-up view of the wing-tip region.**

Figure 6-9 shows the best geometries generated by the five runs of the EMFID method. These designs show strong similarity, further illustrating the advanced state of convergence between the five computations. The design trends are the same as observed in the results from the benchmark method, i.e., the maximum possible sweep at the tip station and minimized chord values inboard. The best designs found by the EMFID process are on the boundary, defined by the benchmark design space, for five of the six geometric variables. Thus, both design methods have revealed that the optimum chord distribution is on the problem bounds, and as a result the best designs from the two methods are very similar (Figure 6-10). Only the first chord variable (point 1 in Figure 6-8) shows some variation. However, it is likely that the EMFID geometries meet the bounds of the benchmark problem only because of the *repair* action, which is used if an EMFID geometry, resulting from the inverse design step, violates the bounds of the benchmark problem. Therefore, although Figure 6-6 shows that the EMFID process reaches the optimum design with significantly less computational expense, the parameterization used may not be able to represent this design exactly. The consequences of this result are discussed in the section to follow.

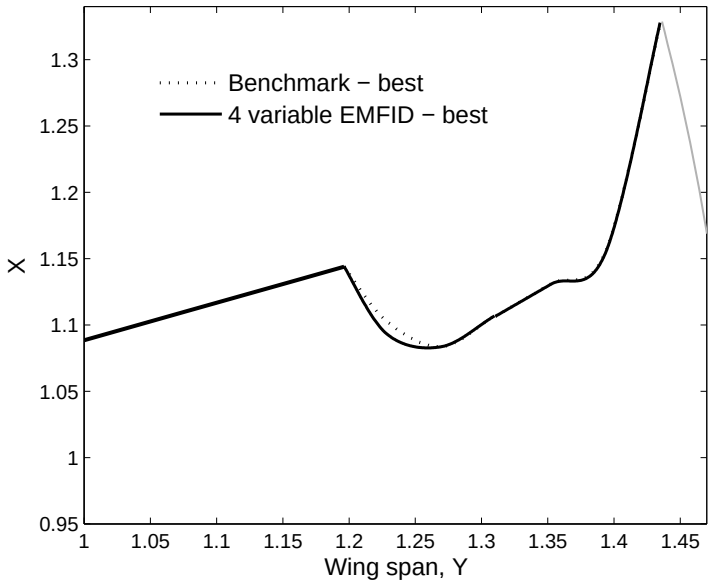


Figure 6-10 Comparison of the best geometry generated using the EMFID and benchmark methods.

6.4 Representing the Optimal Flow Features Set

One of the conclusions of the 2-D airfoil design study (Section 4.6.1) was that the EMFID method benefits from the relatively simple shape of the pressure distribution corresponding to the *optimum* (or best) airfoil design. This simplicity allows the pressure distribution to be represented using a low-dimensional B-spline parameterization. It follows that the parameterization of the $c_l \times c$ profile for a 3-D wing-tip device should be able to represent, in sufficient detail, the $c_l \times c$ profile corresponding to the optimum chord distribution. The effectiveness of the EMFID parameterization, used in the case studies in this chapter, is evaluated in what follows.

In case study 5, in which the drag was calculated using FLUENT Euler simulations, both the quadratic and cubic EMFID parameterizations facilitated rapid convergence of the design search. The four-variable cubic representation of the lift distribution resulted in the best EMFID design; however, this did not achieve the level of drag of the best benchmark designs (Figure 6-1). Figure 6-11 shows the $c_l \times c$ profile corresponding to the best design found by the benchmark method in case study 5. Also shown is a cubic polynomial, fitted to the best benchmark curve in a least-squares sense. Although the benchmark result appears to be relatively simple, the cubic representation is not sufficiently flexible to match it; in particular, the inflection point in the curve is too far inboard. This observation suggests that the EMFID

parameterization is not able to reproduce the best design from the benchmark process. If one assumes that the benchmark design is optimal, an EMFID search would not find this optimum.

It is interesting to determine the geometry corresponding to the fitted cubic curve in Figure 6-11, and obtain the FLUENT drag prediction for this shape. Figure 6-11 additionally shows the $c_l \times c$ profile giving a converged inverse design result when the cubic profile is used as a target. Figure 6-12 compares the benchmark geometry with the chord distribution resulting from this inverse process. Since their lift distributions are different, some disparity between the geometries is expected. However, the geometries show good similarity, and exhibit the same rearward sweep of the tip section. The FLUENT Euler analysis for the inverse designed wing gives a drag coefficient of 141.04 counts. For comparison, the best benchmark design has a drag of 141.00 counts, and the best EMFID design from Case 5 using the cubic parameterization has a drag of 141.02 counts (Table 6-1). Thus, the newly generated design performs comparably with the best benchmark design, despite the apparent mismatch between the spanwise lift profiles. However, as previously discussed, the dominant geometry variable is the tip chord, while the chord distribution inboard of the tip has a lesser effect on the induced drag. Given that the geometries in Figure 6-12 have the same tip chord, their similar induced drag performance is understandable. The fitted cubic curve has resulted in a relatively good design, even though the benchmark $c_l \times c$ profile has not been exactly matched. Thus, the results from case study 6 may provide a better evaluation of the EMFID parameterization, since the drag includes the viscous component and therefore has a greater dependence on the inboard geometry.

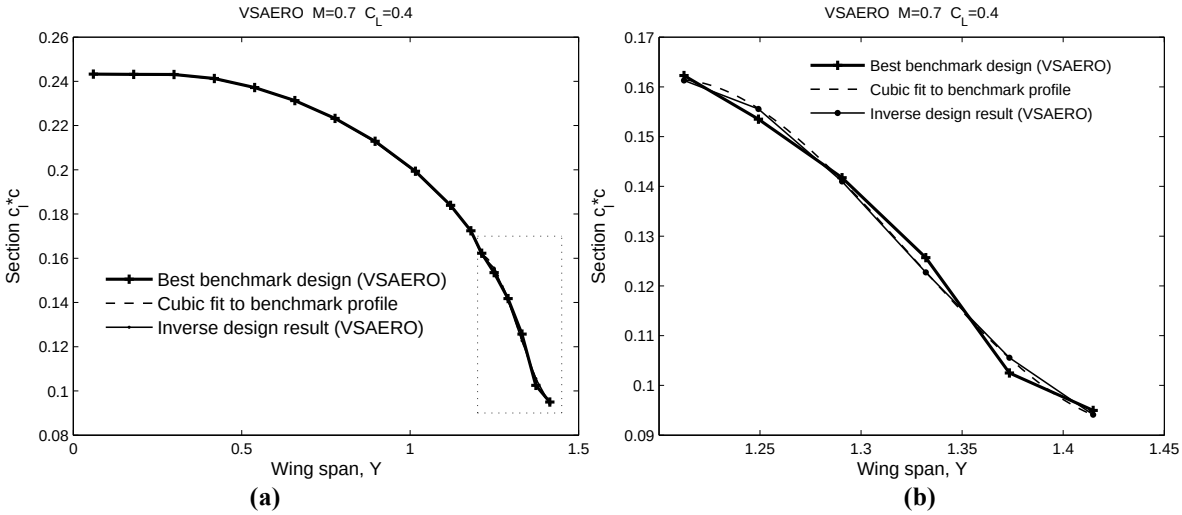


Figure 6-11 Lift distributions predicted by VSAERO, showing the lift profile for the best benchmark design from Case 5, a least-square fit of the cubic curve, and the inverse design result. (a) The entire wing lift distribution. (b) A close-up view of the profile over the wing-tip.

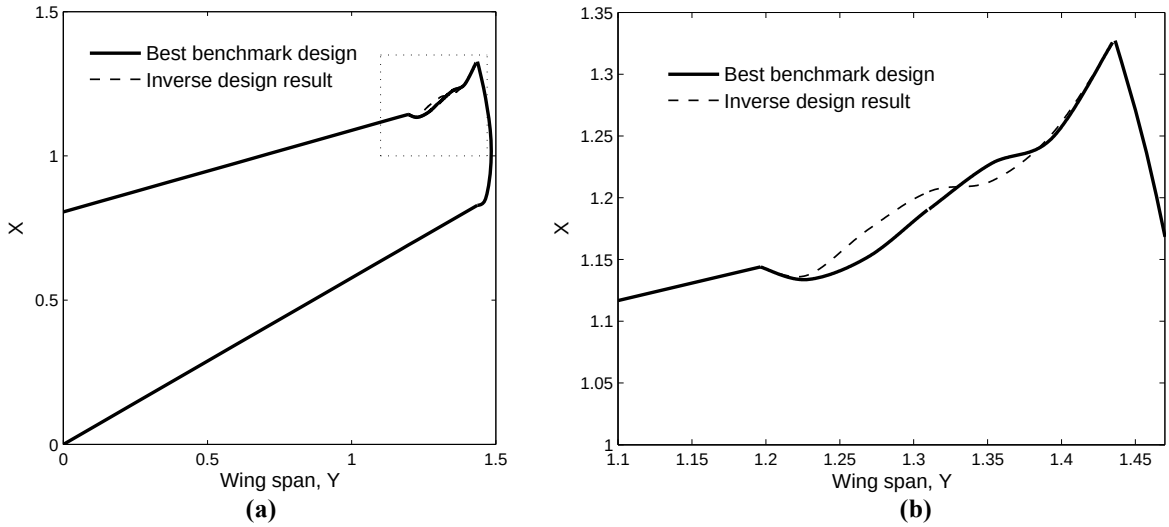


Figure 6-12 **The best benchmark geometry from Case 5, and the geometry resulting from inverse design. (a) Shown on equally scaled axes. (b) A close-up view of the wing-tip region.**

In case study 6 the drag was calculated using RANS simulations, and as a result the best designs feature a lower wetted area than the designs from Case 5. Again, it is desirable to assess the capability of the EMFID parameterization in emulating the flow features of the optimum (best) designs. Figure 6-13 shows the $c_l \times c$ profile for the best performing benchmark design found in case study 6. Also shown is the lift distribution obtained when the cubic parameterization is fitted to the benchmark profile in a least-squares sense. As previously shown, the cubic curve is not sufficiently flexible to represent the benchmark curve in great detail. Indeed, the fit is made more difficult in this case because the benchmark curve has two inflection points, and this is of course not possible for a cubic. The inverse design process has been run using the fitted cubic as a target; the resulting geometry is shown by the dashed line in Figure 6-14, and is plotted with the best benchmark design. Figure 6-14 shows a large dissimilarity between these two designs; the error between the lift profiles at approximately Y=1.25m has resulted in a significantly higher chord at this station on the newly generated design. The associated increase in the wetted area adversely affects the drag; the drag coefficient for the inverse designed wing, calculated using the RANS FLUENT solver, is 218.2 counts, which can be compared with 217.8 for the best benchmark design. Thus, an attempt to emulate the optimum (best) flow features using the EMFID parameterization has yielded a design with rather higher drag. This is explained by the relatively poor fit of the cubic curve combined with a higher drag penalty for a non-optimal inboard chord distribution.

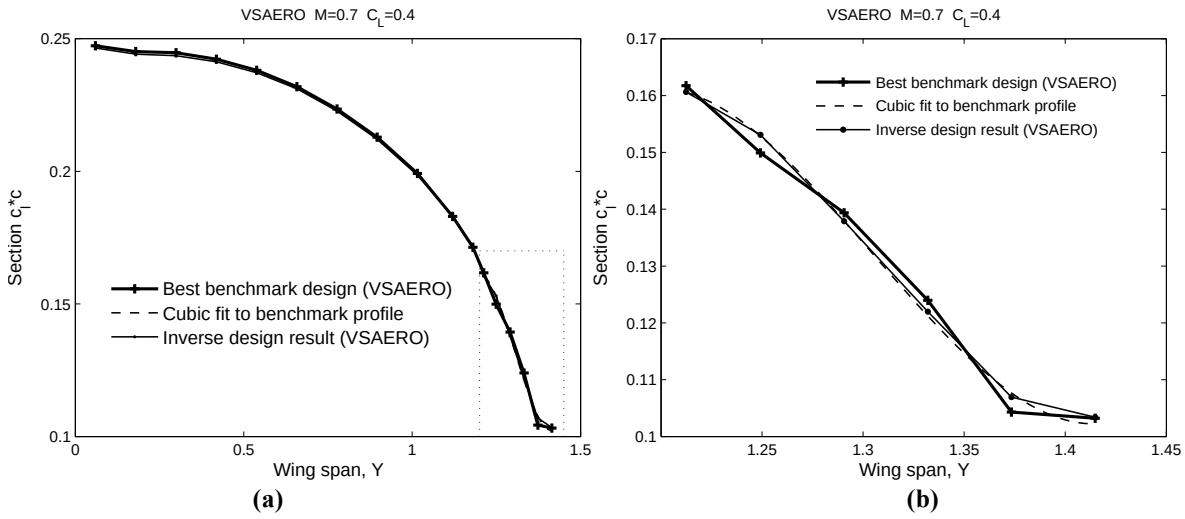


Figure 6-13 Lift distributions predicted by VSAERO, showing the lift profile for the best benchmark design from Case 6, a least-square fit of the cubic curve, and the inverse design result. (a) The entire wing lift distribution. (b) A close-up view of the profile over the wing-tip.

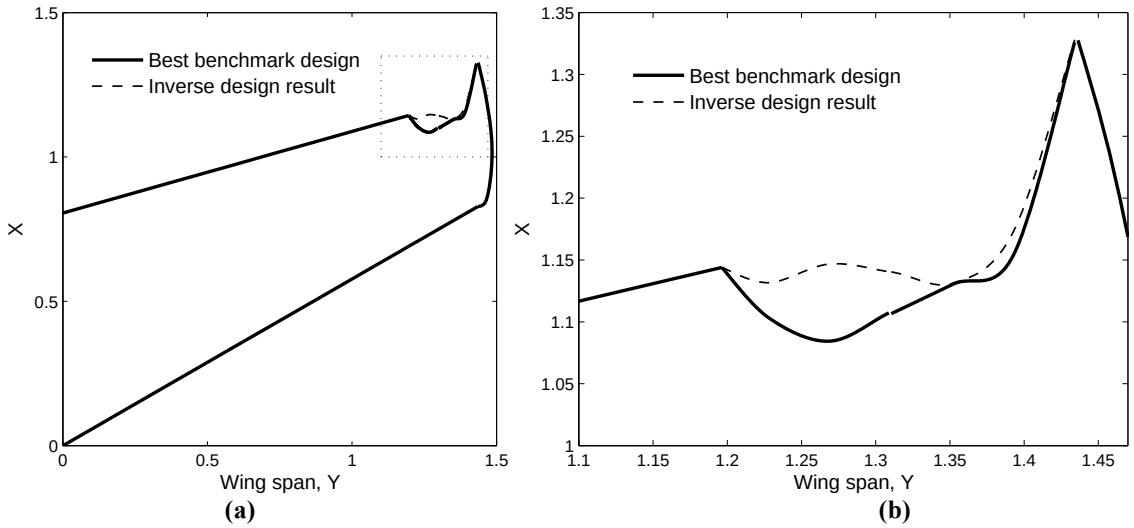


Figure 6-14 The best benchmark geometry from Case 6, and the geometry resulting from inverse design. (a) Shown on equally scaled axes. (b) A close-up view of the wing-tip region.

It has been shown that the EMFID parameterization of the $c_l \times c$ distribution over the wing-tip device is not able to represent the optimum chord distribution, and yet the EMFID and benchmark methods have resulted in almost identical geometries, as shown in Figure 6-10. The EMFID method is only able to generate this geometry because of the *repair* operation which is used to ensure that both methods have access to the same range of geometries. Recall that for a given call to the EMFID objective function, if

the geometrical variables, output from the inverse design process, lie outside the bounds of the benchmark problem then the variables are repaired such that they take the value of the nearest bound. For the optimum design in case study 6, five out of the six geometry variables are on the boundary of the design space. Thus, the repair operation in EMFID allows access to this optimum, even though the parameterization is not able to represent it.

Figure 6-15 shows the parametric lift profile optimized by the EMFID method, together with the converged inverse design result. It can be seen that this best target profile is very different from the profile corresponding to the best benchmark design. However, after the EMFID geometry is repaired, i.e., after the geometric variables have been corrected such that they lie within the bounds of the benchmark problem, the resulting lift profile is very similar to the benchmark curve. Thus, the designs which have been generated by the EMFID method in this guise prove little about the capability of a parameterization of the wing spanwise lift distribution. What has been shown is that the optimum benchmark lift profile cannot be represented accurately using the EMFID parameterization, but the use of fewer design variables in EMFID facilitates rapid convergence towards promising designs, and ultimately, has allowed the EMFID method to generate the best design in case study 6.

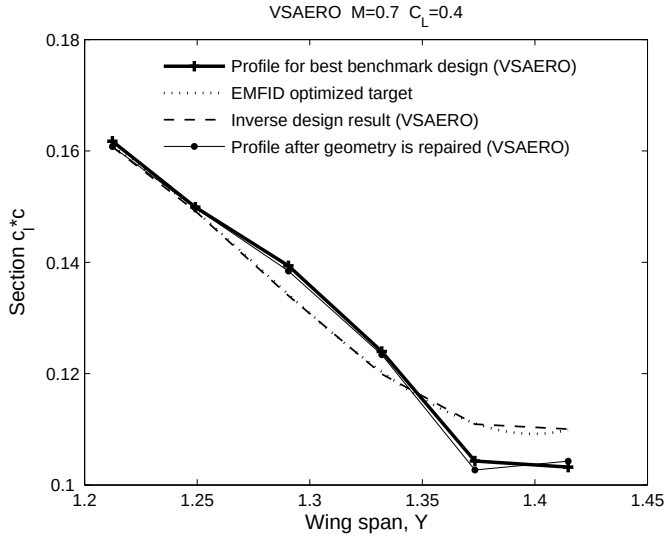


Figure 6-15 The lift profile *optimized* by the EMFID process, shown with the profile obtained after the geometry is repaired. Also shown is the best benchmark profile.

It is also useful to establish how close geometries, resulting from the EMFID parameterization and inverse design, can get to the optimum geometry. As previously shown, the best geometry from case study 6 was generated using the EMFID method; the corresponding VSAERO lift profile is shown in Figure 6-16. A design search has been run on the four-variable EMFID parameterization, in which the

objective function calculation runs the inverse design process using the parametric lift profile as a target. The objective is to minimize the difference between the geometry output from inverse design and the known optimum geometry (best EMFID design). The repair operation is not applied to the geometries. The cubic lift profile, optimized in this way, is shown in Figure 6-16 along with the converged inverse design result. The inboard region of the lift profiles in Figure 6-16 is very similar, but the profiles differ at the two stations nearest the tip. The geometry which was found to be the closest match to the previously located optimum is shown in Figure 6-17. The newly generated design and the best EMFID design from Case 6 show good similarity. The drag coefficient for the new design is 217.79 counts, which is very similar to, and slightly better than, the drag of the best EMFID design from Case 6, which is 217.80 counts. Despite this, the six geometry variables defining this new design do not match the target variables exactly, and in fact two of these variables violate the constraints used in the benchmark design search. Therefore, from these observations it can be concluded that the EMFID parameterization of the lift profile can represent the best designs from Case 6 to a reasonable degree of accuracy and the resulting shape has a similarly low drag, but this design is not within the original design constraints. However, this problem is an artefact of the desire to provide a fair comparison between the EMFID and benchmark methods, and it should not detract from the apparent capability of a parameterization of the lift profile.

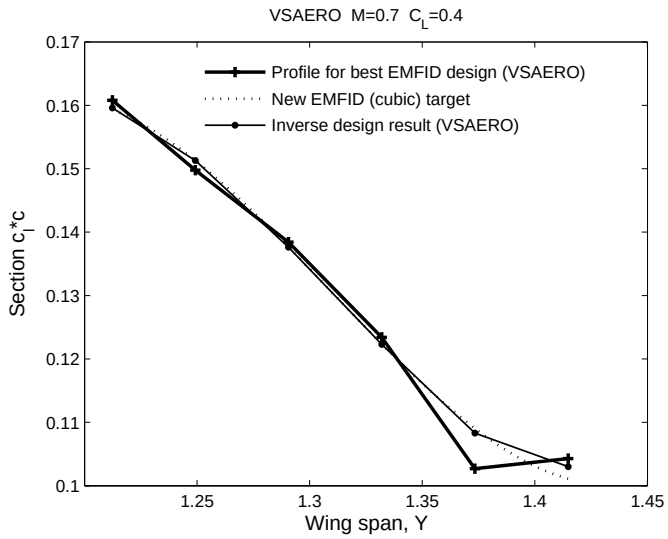


Figure 6-16 VSAERO lift profile for the best EMFID design from case study 6, shown with the cubic target profile optimized to minimize the difference to the optimum design.

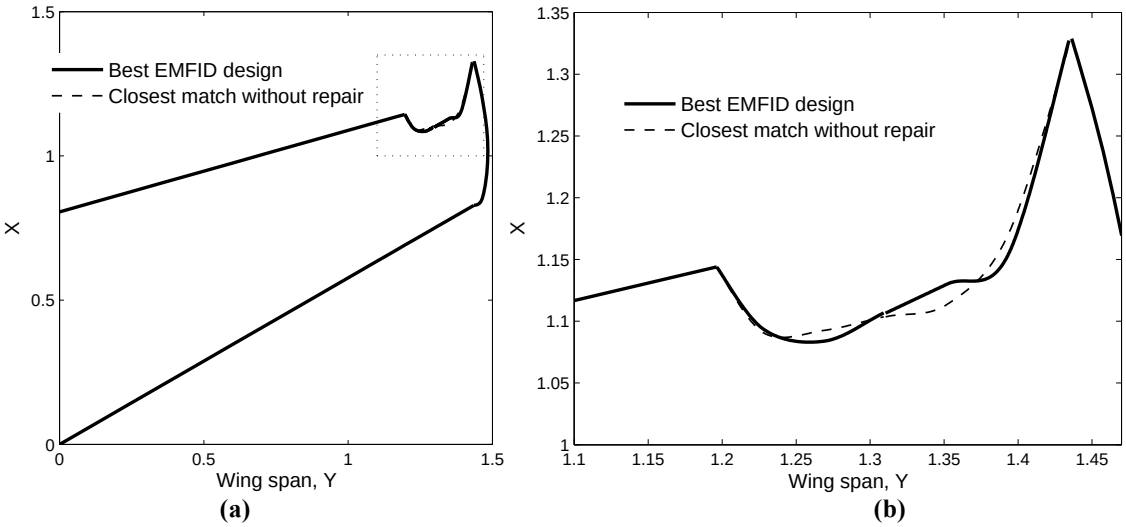


Figure 6-17 The EMFID geometry which was found to most closely match the best EMFID design. This was generated using the EMFID parameterization but without implementing the repair operation. (a) Shown on equally scaled axes. (b) A close-up view of the wing-tip region.

This section has described the capability of the EMFID parameterization in relation to the *optimal* flow features set, with and without the repair operation. Finally, the design search ability of the EMFID method is demonstrated with the geometry constraints and repair discarded. The EMFID design search has been run on, from the point when the best design was found, with the geometry constraints removed. Figure 6-18 shows the optimization-iteration history for this unconstrained design search. The search quite quickly finds improved designs, and achieves an additional drag coefficient reduction of 1 count, compared to the geometrically constrained optimization. Figure 6-19 displays the final geometry from this design search plotted with the best EMFID design from Case 6; the design trends are clearly similar, since the new design exhibits a further increase in the tip sweep and a further reduction in wetted area. Despite the removal of the geometric constraints, the tip sweep is not significantly greater than shown in previous designs, perhaps indicating that this level of sweep is optimal. However, two of the four $c_l \times c$ variables are close to their lower bound in the EMFID search space, i.e., the design space containing the parametric $c_l \times c$ profiles, and one of the four is close to its upper bound. Thus, this design search has not been bounded using the six geometrical variables, but the design search has instead arrived at a design which is on the EMFID problem bounds, due to the significant drag benefit brought about by minimizing the wetted surface area. Interestingly, the chord distribution between Y=1.24m and Y=1.34m is swept forward, rather than being swept back as featured in the previous designs.

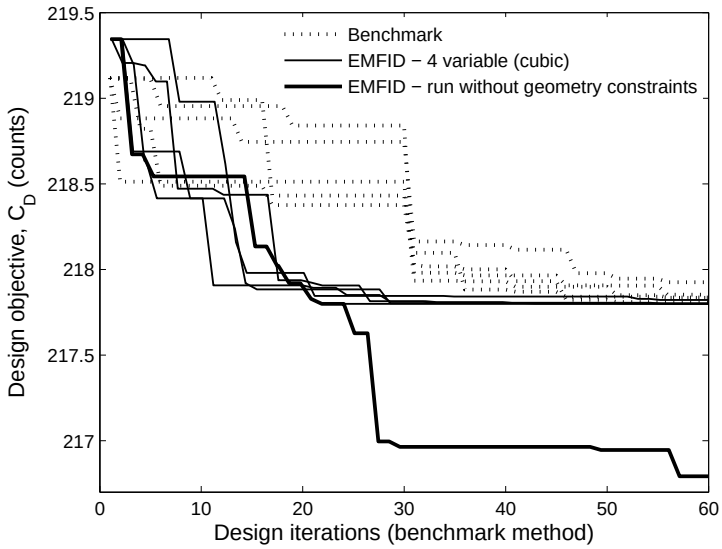


Figure 6-18 Optimization-iteration histories for the EMFID and benchmark methods, showing the result when the EMFID search is run without the constraints on geometry.

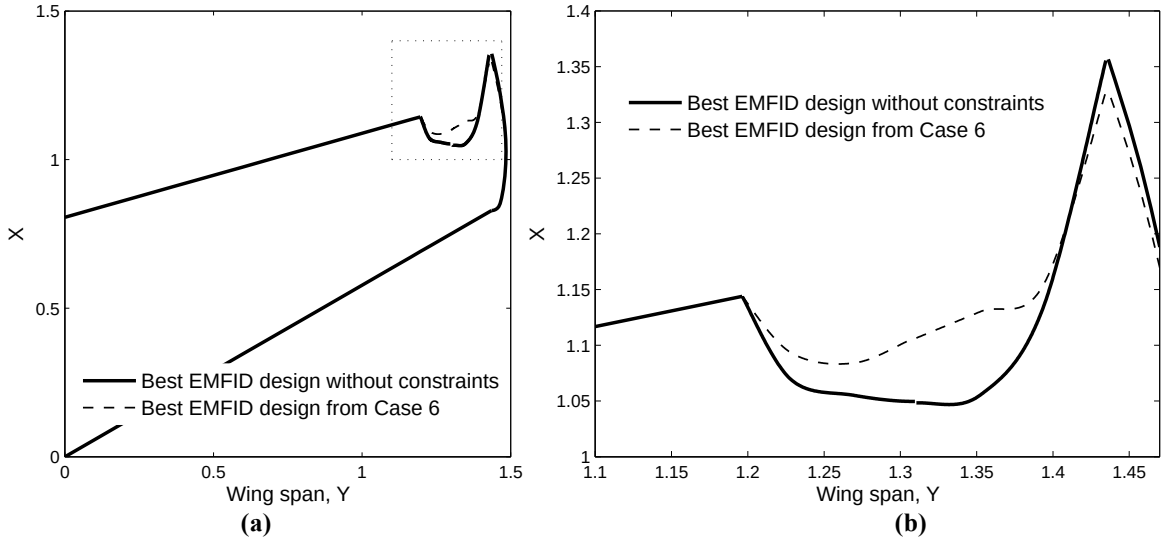


Figure 6-19 The best design found when the EMFID method is run without the constraints on geometry, shown with the best result from case study 6. (a) Shown on equally scaled axes. (b) A close-up view of the wing-tip region.

6.5 Improving Efficiency by Reducing Dimensionality

The preceding analysis has described how the EMFID parameterization of the lift profile is insufficiently detailed to represent the lift distributions of the best benchmark designs, but despite this the method can yield high performing geometries. In addition, the use of just four design variables (compared to six in the benchmark process), combined with the inherent ability to generate only *sensible* shapes, allows the EMFID method to converge rapidly towards promising designs. Recall that the EMFID concept is motivated by the notion that a parameterization of flow features can result in designs of higher detail and quality than a geometrical parameterization which uses the same number of variables. Therefore, it is prudent to consider how the four-variable EMFID parameterization of the lift profile performs in comparison to a four-variable benchmark model. Having applied the EMFID and benchmark methods to the design of the chord distribution, the design features which constitute a *good* design, in terms of the drag calculated using Euler and RANS simulations, are known. Thus, the expected performance of a four-variable benchmark model can be acknowledged.

In case study 5, the use of Euler simulations as the high-fidelity drag calculation meant that the design searches generated a relatively large range of final geometries, suggesting that the induced drag is insensitive to the inboard chord function. However, common to all the final designs is a high (maximized) chord at the tip section, giving rearward sweep, which appears to be strongly linked with a reduction in induced drag (see Appendix D). Therefore, a four-variable Catmull-Rom spline representation of the chord function would serve as a perfectly adequate parameterization for the application in case study 5, producing equally low drag designs. The convergence of a design search using such an approach would likely be faster than the six-variable computations demonstrated, and perhaps as effective as the EMFID runs.

Since the high-fidelity drag calculation in case study 6 uses a RANS solver, there is apparently an *optimum* design which achieves the best compromise between viscous and induced drag. Figure 6-20 shows the best design generated using the EMFID parameterization without the constraints on geometry, as described in the previous section. In attempting to represent this *optimum* using a four-variable geometry-based parameterization, the defining control points should be arranged assuming no knowledge of what constitutes a good design; the four Catmull-Rom spline control points are therefore equally spaced along the spanwise axis. Using this formulation, the four-variable spline which was found to most closely match the best EMFID geometry is shown as a dashed line in Figure 6-20. The two designs are similar, but the four-variable model lacks the detail required to optimize the wetted area. The FLUENT drag coefficient for the four-variable design is 217.01 counts, which is slightly higher than the drag of the

best EMFID design, which is 216.79 counts. From this comparison it can be seen that, when the full RANS analysis is employed, a four-variable benchmark method cannot achieve such low drag designs as the EMFID approach.

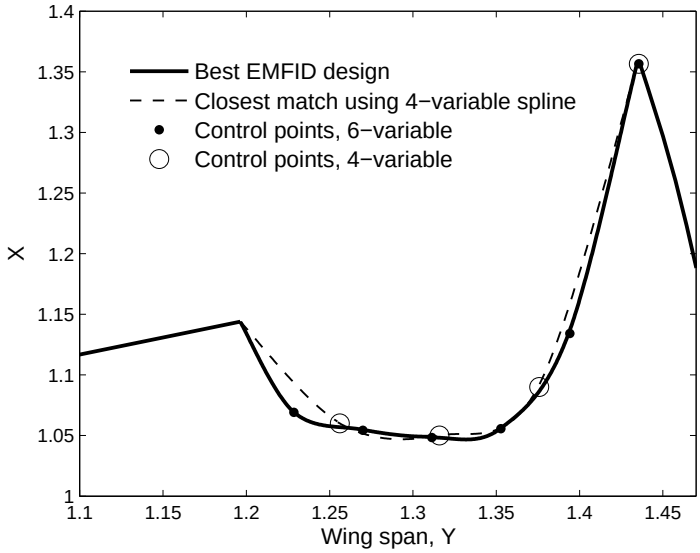


Figure 6-20 The best wing-tip design generated using the EMFID method, shown with a representation of this design using a four-variable Catmull-Rom spline.

It is important to look beyond the preceding case studies and consider the effectiveness of the EMFID parameterization in a more general sense. Although there are just four variables defining the target spanwise lift profile, the underlying geometry description uses a six-variable spline curve. The use of six control points in the spline facilitates finer control, and potentially higher quality designs, than a spline curve with four control points. Thus, in this sense the EMFID parameterization acts as a *reduced order* model, and, in general, can provide more detail than a geometrical model of the same dimensionality. However, the geometrical variation is limited by the flexibility in the shape of the defining lift profile, which, in this work, is a low-order polynomial. Thus, as in the 2-D case studies, the effectiveness of the EMFID concept relies on the ability to represent the *optimal* flow features using the chosen parameterization. Further, a parameterization of flow features is truly advantageous when the optimal flow features are relatively simple, while at the same time the optimal geometry is complex and difficult to describe using few variables.

6.6 Corollaries from the 3-D Case Studies

In summary, the case studies in this chapter have described the application of the EMFID and benchmark methods to the design of the chord distribution for a 3-D wing-tip device. The design objective is to minimize the drag predicted by high-fidelity Euler and RANS simulations in case studies 5 and 6, respectively. In Case 5, the EMFID method converged rapidly but did not produce the best design. A large range of geometries were generated, but all designs featured a highly aft-swept tip. In Case 6, when the CFD considered viscous effects, the generated designs again featured a highly aft-swept tip, but there was also a clear trend to minimize the wetted area inboard. The EMFID design search runs converged more rapidly and with closer convergence between the resulting designs than the benchmark method achieved. However, the EMFID parameterization is only able to represent the optimum Case 6 design because of the repair process which constrains all geometries to lie within the bounds of the benchmark problem. Further analysis of these results has revealed that without this repair process the EMFID parameterization can generate a design very similar to the optimum Case 6 design, and this approach may prove to be more effective than a geometrical description of the same dimensionality. The key results of this chapter, pertinent to the evaluation of the EMFID concept, are listed in what follows.

First, it has been shown that the lift distribution corresponding to the optimum wing-tip design is rather too complex to be represented by a low-order polynomial. Despite this, in both case studies the EMFID design search generated high performing geometries. In Case 5, this can be explained since there are a range of acceptable inboard chord distributions which can result in low induced drag, provided that the tip chord is maximized. In Case 6, the drag is more sensitive to the inboard chord profile, but the EMFID process achieves the optimum because the repair operation forces the geometry variables onto the boundary of the design space. The intention of the repair process was to ensure a fair comparison between the EMFID and benchmark methods, but in Case 6 it has (rather ironically) favoured the EMFID search. In subsequent analysis, the EMFID cubic parameterization has been shown to represent the optimum design to a reasonable degree of accuracy, and with a similar drag result. Given the result of Case 6, the comparison might benefit from an alternative means of bounding the two methods. In this sense, the parameterization of the spanwise lift profile is a promising but unproven concept.

Second, although the range of geometrical change afforded by the EMFID approach is limited compared to the aforementioned benchmark method, the EMFID parameterization can result in finer detail and complexity than a geometry based description of the same dimensionality. This is because, although the target lift profile is defined using four variables, the underlying geometry description, used for inverse design, is defined using six variables.

Third, the EMFID strategy has been shown to converge more rapidly towards promising designs compared to the benchmark method. The first reason for this is the reduction in the number of design variables. The second reason is that the specified lift profile is inherently smooth, and hence there is an inability to generate the nonsensical *rippled* shapes that the benchmark method evaluates, particularly in its DoE phase.

The application of the EMFID and benchmark methods to the design of the chord distribution for a wing-tip device has revealed that the EMFID strategy can potentially provide gains in computational efficiency. Although the EMFID strategy has not always produced the lowest drag designs in the results presented here, it achieves design improvement more quickly. Also, it has been shown that the parameterization of the wing spanwise lift distribution can be an effective technique for aerospace design.

Chapter 7. Conclusions and Recommendations

The process of design can be described as the manipulation of the product definition, based on the results of analysis, with the aim of achieving improved performance. In the field of aerodynamic design, the product definition traditionally employs a parameterization scheme based on geometry, and such an approach facilitates a systematic search of the design space using an optimization algorithm. However, when calculation of the design objective requires analysis using high-fidelity CFD, the computational cost of such a design search can be prohibitive, and this has led to research into more efficient methods for design. This thesis proposes that a geometry-based parameterization, such as a spline-curve technique, is not always the most efficient method of representing a design for the purposes of optimization. A novel approach has been proposed which uses a parameterization of flow features coupled with inverse design, with the aim of reducing the number of design variables used in the optimization process. The work documented in this thesis has evaluated the embedded multi-fidelity inverse design (EMFID) concept in comparison with a method which represents a benchmark in geometry-based parameterization. The EMFID and benchmark methods have been applied to the 2-D airfoil design problem and to the design of a 3-D wing-tip device.

This chapter reports the conclusions of the work described in the thesis. The progress and achievements of the EMFID concept are detailed, and specific contributions are listed. Finally, future avenues of research are highlighted, offering alternative applications of the work in this thesis.

7.1 Experience with a Parameterization of Flow Features

In Chapter 2, an array of computational techniques currently employed in the field of aerodynamic design was outlined, and a distinction was made between two design approaches: *direct* design optimization and inverse design. The EMFID process combines these two methods, with the aim of improving the efficiency of performing design optimization using high-fidelity CFD analyses. The underlying prediction is that a low-dimensional parameterization of flow features can, following inverse design, result in a larger range of geometrical variation and higher quality designs than a geometry-based parameterization of the same dimensionality. This reduces the number of design parameters required for optimization, leading to a more efficient search process. The EMFID method has been demonstrated in this thesis using four case studies in airfoil design and two further case studies considering wing design. Table 7-1 briefly describes these case studies, augmenting Table 4-7 with the 3-D cases from Chapter 6. The conclusions drawn from the six case studies are described next.

Table 7-1 Summary of the case studies reported in this thesis.

Case studies		Flow equations	Flow speed Mach number	Case description
2-D airfoil design	1	RANS	0.15	Comparison of EMFID and benchmark methods
	2	RANS	0.15	Benchmark run using a low dimensional model
	3	RANS	0.15 VGK, 0.73 FLUENT	Comparison using a transonic drag calculation with a subsonic C_p model in EMFID
	4	RANS	0.73	EMFID run using a transonic C_p model
3-D wing-tip design	5	Euler	0.7	Comparison of EMFID and benchmark methods – minimizing induced drag
	6	RANS	0.7	Comparison of EMFID and benchmark methods – minimizing total (induced and viscous) drag

In Chapter 3, the concept of flow feature parameterization was introduced and the practicalities in implementing the method for 2-D airfoil design were detailed. Following this, in Chapter 4, the EMFID process was applied to 2-D airfoil design in case studies 1 to 4. The conclusions from this work are fourfold.

- First, it has been shown that a B-spline representation of the airfoil surface pressure distribution can be an effective parameterization technique and is able to generate high-performing designs.

The low-dimensionality of this method results in rapid convergence of the design search. In addition, the smooth nature of the B-spline representation means that the resulting airfoils are inherently *sensible* shapes, and this further accelerates the design search towards promising designs.

- Second, it has been demonstrated that a geometrical parameterization, using the same number of variables as the EMFID process, is unable to generate such high-quality and detailed shapes, and the corresponding design search finds poorer designs than the EMFID method.

Thus, it is apparent that a parameterization of the pressure profile can result in a more efficient design search than a geometrical parameterization of the same dimensionality. However, it is noted that the EMFID method benefits from the relatively simple shape of the pressure distribution corresponding to the optimum (or at least, very low drag) airfoil design. This simplicity means that a low-dimensional parameterization can perhaps more effectively represent the pressure distribution than the geometry.

- Hence, the third conclusion states that, to be successful, the choice of flow feature to parameterize in EMFID should take into account the simplicity or complexity of the flow features corresponding to the optimum design.

- Fourth, and finally, the EMFID concept has been applied to the design of transonic airfoils, and it was found that the method benefits from using the same flow conditions for the parameterization, inverse design, and the final drag calculation.

In particular, it is important that the parameterization of the pressure profile includes the specification of the shock. Incorporating the shock strength and position into the design variable set provides the optimizer with a direct means of controlling the wave drag, and gives a simpler objective function landscape. Conversely, if the target pressure distribution is specified for subsonic flow conditions, the relationship between the design variables and (transonic) drag is likely to be complex. In addition, the use of a transonic inverse design procedure means that the resulting airfoils are inherently *tailored* for the transonic regime.

A parameterization of the surface pressure distribution has been shown to be an effective approach for subsonic and transonic airfoil design. Although, ultimately, higher-dimensional schemes may be able to represent finer detail, a design search using the six-variable EMFID parameterization is a highly efficient practice, producing higher performing designs than a 13-variable geometry-based method for a given

computational expense. Following the success of the EMFID concept in the latter guise, a logical progression was to attempt to apply the method to 3-D wing design. Chapter 5 introduces the application of wing-tip device design, and reports a study examining an appropriate flow feature to parameterize in EMFID and a suitable geometric representation for the wing-tip device. The details associated with the wing-tip vortex are a novel choice of flow feature, but the vorticity predictions of the panel code and high-fidelity CFD code do not agree sufficiently to provide a meaningful design search. Consequently, the spanwise loading distribution was chosen as the design flow feature. Concerning the geometry description, a set of five distinct geometrical wing-tip device parameters are inappropriate because these are mapped non-uniquely to the spanwise loading over the wing-tip. Thus, the spanwise variation of chord over the wing-tip is used, since each design corresponds to a unique lift distribution.

Chapter 6 presents two case studies in which the EMFID and benchmark methods are applied to the design of the chord profile over a wing-tip device. The use of a quadratic or cubic polynomial to represent the lift distribution over the wing-tip device allows the EMFID process to converge more quickly than the six-variable benchmark method. As in the 2-D airfoil case studies, this is due to both a lower dimensionality in EMFID and an inability to generate nonsensical shapes. When the objective function, drag, was calculated using Euler simulations, a large range of final geometries were generated using the design searches. The induced drag is relatively insensitive to the chord function inboard of the tip, but is more dependant on the chord at the tip station, and hence all final designs feature highly aft-swept tips. However, the benchmark method produced marginally better performing designs, and it was found that the lift distribution corresponding to the best benchmark designs is too complex to be represented using the EMFID quadratic or cubic curve. When the drag objective is calculated using RANS simulations, there is a clear trend to minimize the wetted area while maintaining the tip sweep. Although the EMFID design search only reached the optimum design because of the geometry repair process, it converged significantly more quickly to this optimum than the benchmark method. Again, the EMFID parameterization is not sufficiently detailed to be able to represent the optimum lift distribution. However, when the method is not forced to conform to the benchmark constraints, the EMFID scheme can produce geometries closely matching those that are repaired.

There are four key conclusions resulting from the 3-D application of EMFID.

- First, it is important that the flow feature to be parameterized in EMFID must map uniquely to geometry, i.e., each wing-tip lift profile should correspond with a single geometry.

- Second, the parameterization of the lift profile has been shown to be effective in the sense that it facilitates rapid convergence of the design search through a reduction in the number of design variables.

- Third, the loading distribution corresponding to the optimum design must be sufficiently simple that it can be represented using a low-dimensional parameterization. In the 3-D design scenario presented in this work, the optimum flow feature is rather too complex. If the optimum lift profile were simpler then perhaps a quadratic or cubic would be sufficient to reproduce it. However, because the optimum geometry is relatively complex, it remains the case that the EMFID approach has generated better performing designs than would be possible with a benchmark search using the same number of variables.

- Thus, fourth, it can be concluded that a parameterization of the lift profile is able to produce finer detailed designs than a geometrical-based scheme of the same dimensionality.

The significance of the EMFID concept as a design tool for aerospace design should be considered. Fundamentally, the method can be used as a low-dimensional means of representing any surface subjected to fluid flow, provided that there exists a suitable flow feature to parameterize. Crucially, due to the expense of performing inverse design at every iteration, the method is only likely to be computationally efficient when high-fidelity CFD analyses are used to calculate the design objective. From a practical point of view, implementation of the EMFID method requires more computational setup time than the benchmark method; in particular the method requires an inverse design procedure which calls upon a low-fidelity CFD code.

Referring to the initial discussions in Chapter 1 surrounding parameterization and Figure 1-2, a need has been identified for a design approach which uses a small number of design variables but which can generate high-performing designs. The concept of flow feature parameterization has been shown to address this need. However, to be successful there must exist a prominent flow feature which exhibits a simple variation. Further, the EMFID process is only advantageous when the flow feature variation is *simple* but the geometrical shape is *complex*. Thus, a parameterization of the surface pressure distribution for 2-D airfoil design is very effective, while a parameterization of the lift profile for 3-D wings has been shown to be promising in terms of efficiency but unproven in ultimate performance. Nonetheless, in all of the case studies demonstrated here, the EMFID strategy has provided gains in computational efficiency; for this reason, it can be a useful tool in the arsenal of an engineer.

Finally, the major contributions of this thesis are listed below.

- A low-dimensional parameterization of flow features can produce high quality geometries following inverse design. Further, such a parameterization can produce finer detail and local control of the shape than a geometry-based parameterization of the same dimensionality.
- For a given level of design improvement, a parameterization of flow features uses fewer design variables, and, combined with an inherent ability to generate *smooth* designs, this leads to a highly efficient optimization process.
- The flow feature parameterization concept is most advantageous when a flow feature can be represented simply while the corresponding geometry is relatively complex.
- The parameterized flow features, the inverse design operation, and the design objective should all be calculated for the same flow conditions, such as Mach number.

7.2 Recommendations for Further Research

There are a number of possible avenues for further research as a result of this thesis. For example, it would be desirable to examine alternative flow feature parameterizations, and to further investigate the mapping between the geometry and flow feature domains, for the purposes of reducing dimensionality for design. Also, there are many interesting applications to which the EMFID process could be applied, such as the parameterization of flow features for duct or nacelle flows, or for the design of wings in ground effect. In this section, two areas for future work of particular interest are outlined. These are: the design of airfoils for multiple operating conditions and the design of a full aircraft wing.

7.2.1 Application of EMFID to Multipoint Design of Airfoils

In the 2-D airfoil design case studies reported in this thesis, the design objective has been to minimize drag at a single value of lift. Consequently, the design searches have resulted in highly specialized airfoils, i.e., airfoils which yield very low drag at the specified level of lift, but give poor drag when operating at a different condition. This result can be seen clearly in the polar plots in Figure 4-6 and Figure 4-19. An interesting application of the EMFID strategy would concern the design of airfoils at multiple design conditions. In practice, this would involve calculating the drag at a number of different

incidence angles, and the design objective might be the sum, or weighted sum, of these drag values. This would require very little modification to the existing optimization framework.

The comparison between the EMFID and benchmark methods is interesting for such multipoint design. It is predicted that the benchmark process would be largely unaffected by the change to the design objective. However, the EMFID strategy is likely to be affected, since the parametric C_p distribution is specified for one design condition (lift) while there are a number of different pressure distributions, calculated using the high-fidelity solver, contributing to the design objective. This situation is akin to the configuration considered in case study 3 (Section 4.4), where the parameterization of the C_p profile was subsonic but the objective calculation was at transonic conditions, and hence the flow features were *de-coupled*. It would be interesting to investigate how significantly this de-coupling affects the EMFID design search for multipoint design. Clearly, parameterizing the pressure profile at each design condition is not practical, both because of the prohibitive number of design variables this would require, and because each design point would then yield a different airfoil shape.

For reference purposes, a copy of the subsonic airfoil inverse design code is given in Appendix E.

7.2.2 Application of EMFID to Wing Design

The 3-D case studies documented in this thesis have focused on the design of a wing-tip device, as this was thought to be an interesting application for a design method based on flow features and their extraction. However, the approach used in the design of the wing-tip chord distribution could equally be applied to the design of the full wing. Also, a rather small number of variables were used in case studies 5 and 6, to reduce the total computational expense when the two design searches are run using five different DoE seeds. A more interesting, and realistic, study would make use of a larger number of design variables. For the benchmark method, the trailing edge chord distribution along the entire wing might be described using a spline curve using ten design variables. In the EMFID method, the entire wing spanwise lift profile would be parameterized using perhaps five variables. This comparison would likely favour the EMFID process.

Also, recall that in the 3-D wing design studies the use of surface pressure as a design flow feature was avoided, as its effectiveness had already been demonstrated in 2-D. Indeed, if the EMFID method was run for wing design using a parameterization of the pressure distribution at the wing sections, this search would likely perform very well. Further, a surface pressure scheme could be used simultaneously with a parameterization of the lift distribution; the lift at the specified wing sections would then determine the

required area for the C_p distributions. The number of design variables required for such a procedure would be rather large, but, following the above conclusions, this would require fewer variables than an equivalent geometry-based scheme.

Finally, further benefit may be gained by investigating the parameterization of different flow features. While alternative flow features have been considered in this work, the aerodynamic loading has always been the flow feature of choice for the case studies demonstrated. The determination of forces is the objective of aerodynamics; their prediction is relatively straightforward and all CFD codes provide facility for their calculation. Also, detailed calculations regarding off-surface flow structures, such as vortices, are currently only possible using high-fidelity CFD codes. Thus, the loading distribution is an obvious flow feature to use as a target for an inverse design method utilizing low-fidelity CFD. In this thesis, the properties associated with the wing-tip vortex were discarded as a candidate flow feature because of the discrepancy between the predictions of the low- and high-fidelity CFD codes. However, if the panel code was replaced with Euler simulations using a coarse mesh, for example, then the vorticity predictions may agree to sufficient accuracy to provide a meaningful design search. As things stand, the method of flow feature parameterization is a promising tool on which to base future applications and research.

Appendix A1: CFD Verification and Validation for Subsonic Airfoil Analysis

When employing computational simulations, it is essential to perform an assessment of the accuracy that can be expected from the results. In general, such a study consists of two contributions. First, *validation* is a comparison of the computational predictions with established experimental data. Second, *verification* is used to quantify the dependence of the solution on *internal* error, i.e., software, numerical or discretization errors. In the case of CFD simulations, it is important to minimize the dependence of the solution with respect to the mesh size and topology, or at least to gain knowledge on this dependence. A verification procedure has been applied for all FLUENT mesh configurations used in this thesis. This appendix documents the mesh verification study for the subsonic 2-D airfoil analysis; similar procedures for the transonic airfoil CFD setup and the 3-D wing setup are given in Appendices A2 and B, respectively.

As stated in Section 3.5.3, the mesh dependence and validation studies for the subsonic FLUENT analysis are performed using the NASA LS(1)-0013 airfoil, for which experimental results at subsonic flow conditions have been published by Ferris et al. [1987]. The CFD is configured for the same flow conditions so that the predicted surface pressure results can be compared with the experimental measurements. The flow conditions are M=0.15, Re=4×10^6, and α=4.1°; these are used for the subsequent design studies thus retaining a valid mesh and CFD setup. For the case of mesh dependence, it is desirable to use a mesh whereby any increase in cell density gives a small change in the c_d prediction. For the verification studies reported in this thesis, the target accuracy is ±1 c_d count relative to the largest mesh evaluated; the largest mesh is taken to be the most accurate. The mesh verification for the subsonic airfoil FLUENT analysis is now described.

The FLUENT computational mesh is constructed using the commercial geometry generation and meshing tool, GRIDGEN. The topology used for the 2-D airfoil analysis is a fully structured *O*-mesh, i.e, a series of radial grid lines, traced from the airfoil surface to the circular outer boundary, and circumferential grid lines. The CFD is setup as a RANS solution, using the coupled-implicit solver, Spalart-Allmaras turbulence model and enabling second order accuracy in terms of flow and the turbulence model. For this RANS configuration, the first step in the mesh setup process is to ensure the correct density of cells normal to the wall. The non-dimensional distance, or y^+, over the airfoil surface should either be less than

1, i.e., resolving the viscous sub-layer, or between 30 and 60, in which case FLUENT employs a log-law wall function. For the analysis used here, the target is a wall y^+ of 30, in order to reduce the computational expense. An initial mesh was constructed, and the first cell height was perturbed until the surface y^+ was approximately 30; a first cell of 0.45mm was found to be suitable. Figure A1-1 shows the corresponding variation of wall y^+ over the airfoil surface.

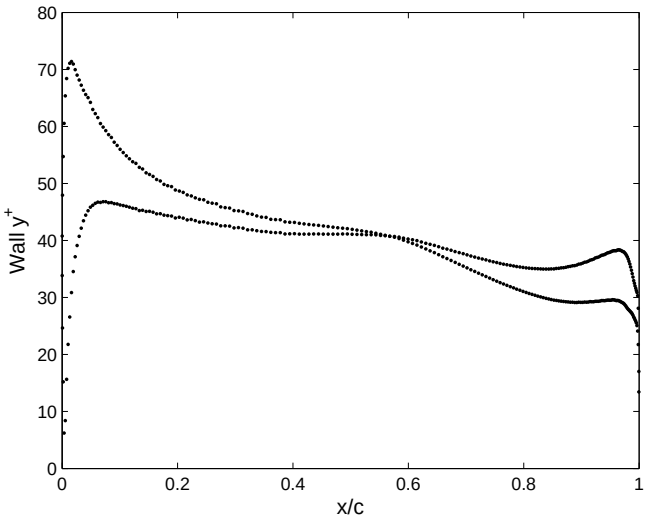


Figure A1-1 Wall y^+ for the subsonic airfoil FLUENT analysis.

The structured mesh cells are grown in height from the airfoil surface to the external boundary, and the rate of growth of these cells can be controlled in GRIDGEN. A growth factor of 1.2 is selected, as this gives a sufficient cell density to capture the boundary layer growth (Figure A1-2) and reduces the total number of mesh cells.

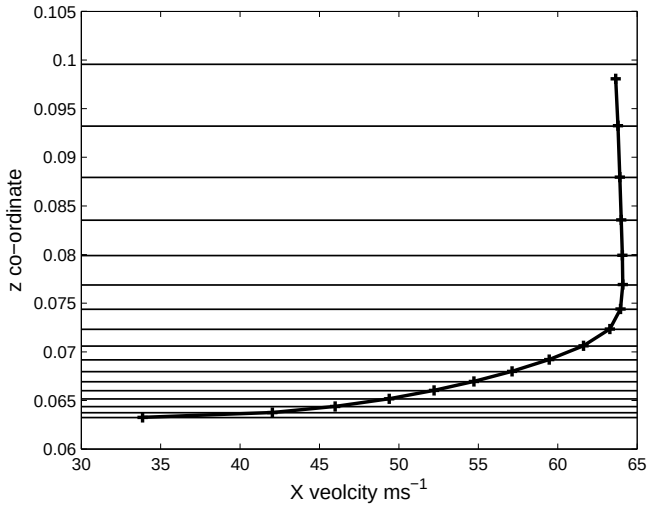


Figure A1-2 Velocity in the *x* direction versus *z* co-ordinate at *x*=0.4, showing the growth of cells normal to the airfoil surface.

In this 2-D analysis, the *pressure-far-field* boundary condition is used for the outer flow domain, which models the freestream conditions at infinity. In order to assume infinite-extent conditions, the flow boundary must be placed far enough away from the airfoil surface such that the solution is not affected. A number of CFD evaluations were performed using different values for the domain size, i.e., different mesh sizes. Figure A1-3 shows the resulting drag predictions plotted against the total number of cells in the mesh. As the boundary is moved away from the surface, the change in c_d prediction generally decreases. As previously mentioned, it is desirable to select a mesh for which the c_d is within ±1 count of the most accurate result recorded. In this case, the largest domain size evaluated was approximately a 100m radius, and this is taken to be the most accurate result. Figure A1-3 shows dotted lines corresponding to ±1 c_d counts from this 100m result. Just within this range is the mesh with 21094 cells, corresponding to a domain size of 28.7m radius (28.7 airfoil chord lengths). Therefore, this domain size is chosen for the final CFD configuration.

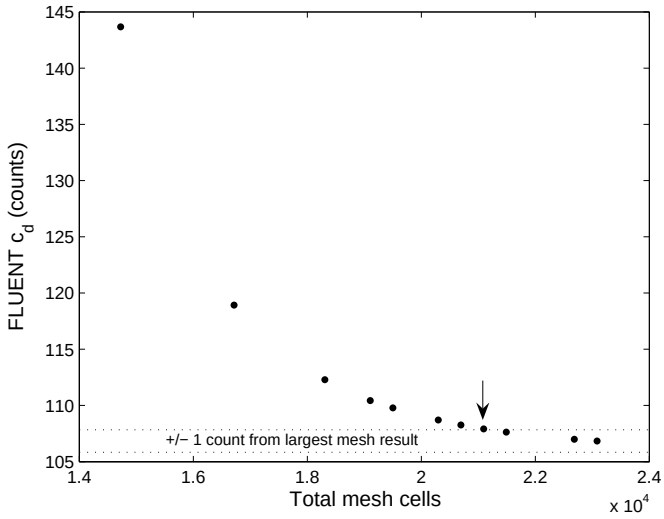


Figure A1-3 Variation of drag as the domain size is increased, showing the tolerance of acceptable accuracy.

The number of mesh cells along the airfoil surface has a significant effect on the solution. Figure A1-4 shows the variation of the predicted drag as the number of cells is increased. Again, a coarse mesh gives an inaccurate solution, and as the mesh is refined the rate of change of drag decreases. From this graph, it is seen that a surface cell density corresponding to a total mesh size of between 35000 and 40000 cells should be used, as this would give an error of less than 1 drag count relative to the largest mesh evaluated. However, such a mesh would result in a prohibitively expensive analysis for the purposes of a 2-D airfoil design study. Therefore, a smaller mesh size of 21094 cells is used, which corresponds to 200 cells over the chord of the airfoil. While this mesh is not ideal, it is seen to be adequate for a design study. Further to this, in performing the mesh dependence study the error in the solution has been quantified.

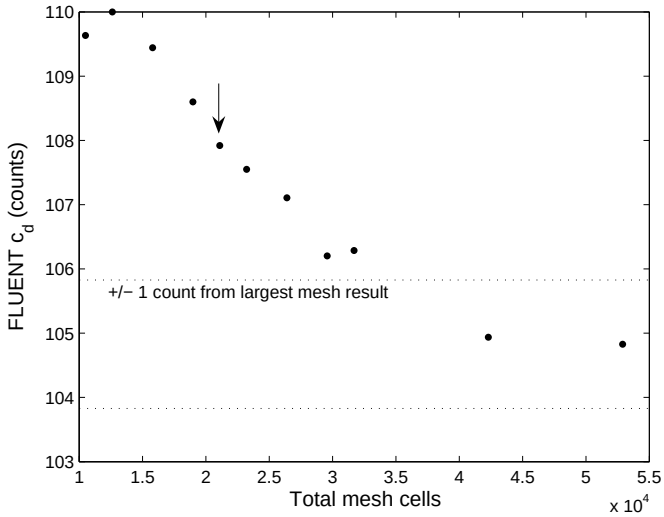


Figure A1-4 Variation of drag as the number of surface cells is increased, showing the tolerance of acceptable accuracy.

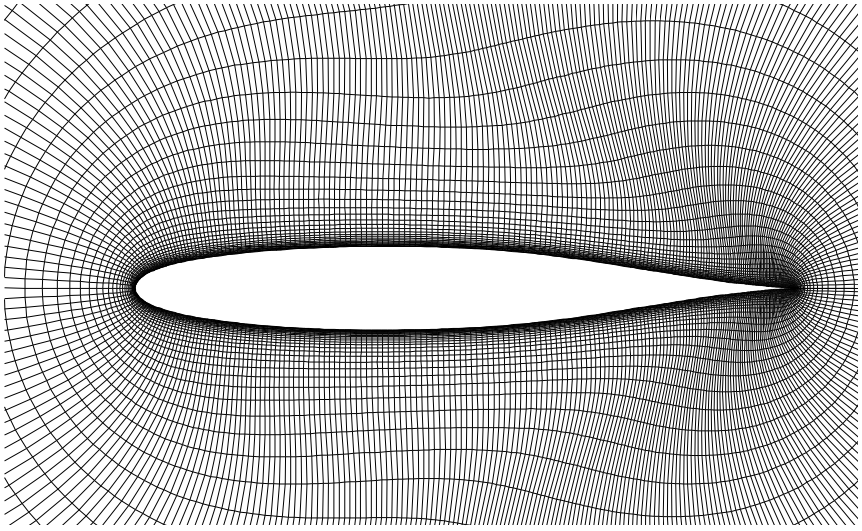


Figure A1-5 The final 2-D subsonic airfoil mesh.

The final subsonic 2-D mesh configuration has 398 cells defining the airfoil surface and 53 rows, giving a total of 21094 mesh elements (Figure A1-5). On studying the convergence history of the solver using this setup, it is observed that 3000 iterations of the RANS calculations are sufficient to provide a converged solution to within ±0.1 counts (±0.00001 c_d) of the fully converged value (Figure A1-6). Note that although this mesh setup is used for widely varying geometries during a design search, no grid quality checks are performed. This is because the mesh topology is sufficiently simple to avoid adverse affects,

such as skewed cells, in the vast majority of cases. On studying the robustness of the mesh, it is found that highly cambered airfoils can cause the mesh at the trailing edge to overlap itself. However, this is not deemed to be problematic since this causes a failed FLUENT result rather than an anomalous drag result, and in any case such cambered geometries are known to perform poorly and are therefore avoided by the optimization algorithm.

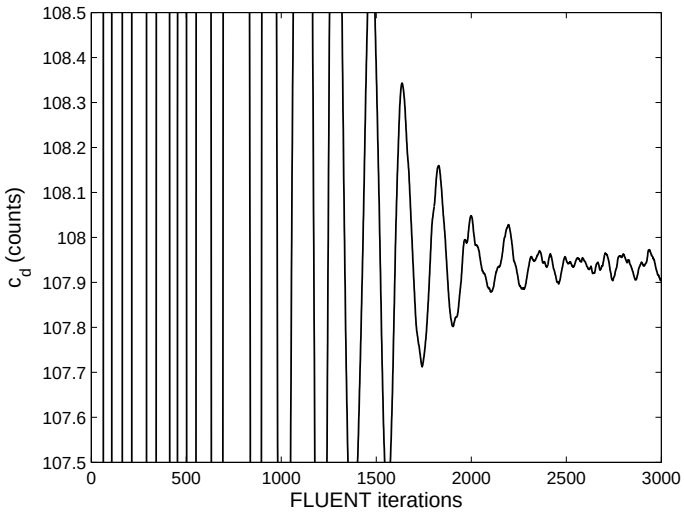


Figure A1-6 Convergence of the drag coefficient during the FLUENT solution procedure.

Table A1-1 details the setup of the mesh and CFD solver used for the subsonic RANS analysis. The surface pressure predictions for the NASA LS(1)-0013 airfoil using this setup have been compared against the experimental data given by Ferris et al. [1987]; this is shown in Figure 3-4.

Table A1-1 Information regarding the setup of the subsonic 2-D airfoil CFD solver.

GRIDGEN v15.08	
Mesh	2-D structured, quadrilateral
First cell thickness	0.045% chord
Total mesh elements	398 columns × 53 rows = 21094 cells
Wall y^+ range (approx.)	30-60
FLUENT v6.3.19	
Solver	Coupled implicit formulation of RANS equations, also solving the energy equation

Solver boundary condition		Pressure =101325Pa, M=0.15
Solver Courant number		5
Turbulence model		Spalart-Allmaras
Turbulence boundary condition		Modified turbulent viscosity ν_t=0.001m^{2}s^{-1}
Turbulence model constants:	C_{b1}	0.1355
	C_{b2}	0.622
	C_{v1}	7.1
	C_{w2}	0.3
	C_{w3}	2
Prandtl number		0.667
Energy Prandtl number		0.85
Wall Prandtl number		0.85
Discretization scheme		Second order upwind
Flow medium		Air as an ideal gas
Flow Mach number		0.15
Viscosity		1.53×10^{-5}kgm^{-1}s^{-1}
Reference temperature		300K
Reference density		1.177kgm^{-3}

Table A1-1 (continued)

Finally, below is a copy of the input journal file used for running the FLUENT simulations in batch mode. This includes the specification of all solver settings.

```
file
set-batch-options
yes
yes
no
quit
file
read-case
"Airfoil2D.cas"
quit
grid check
def mod
solver density-based-implicit y
energy yes
visc spal-all yes
quit

def mat
change-create air air yes ideal-gas yes constant 1006.43 no yes constant 0.000015315…
…no no no no no
quit

def o-c
op-pres 0.0
```

```
quit

define
boundary-conditions
pressure-far-field
pressure-far-field-5
no
101325
no
0.15
no
300
no
0.99744
no
0.071497
yes
no

quit
quit
solve
set
courant-number
5
discretization-scheme
amg-c
1
nut
1
quit
quit
monitors
residual
n-save 10000
n-display 10000
check-convergence?
yes
yes
no
yes
no
convergence-criteria
0.0000001
0.1
100
plot?
no
print?
yes
scale-by-coefficient?
yes
quit
force
drag-coefficient
yes
wall-4

no
yes
"cd_history.txt"
no
no
```

```
0.99744
0.071497
lift-coefficient
yes
wall-4

no
yes
"cl_history.txt"
no
no
-0.071497
0.99744
moment-coefficient
yes
wall-4

no
yes
"cm_history.txt"
no
no
12.0
0.0
quit
quit
quit
report
reference-values
compute
pressure-far-field
pressure-far-field-5
quit
area
1
quit
quit
solve
initialize
compute-defaults
pressure-far-field
pressure-far-field-5
quit
initialize-flow
quit
quit
solve/iterate
3000
quit
file
write-case
"Airfoil2D_run.cas"
yes
write-data
"Airfoil2D_run.dat"
yes
quit
exit
```

Appendix A2: CFD Verification and Validation for Transonic Airfoil Analysis

Case studies 3 and 4, reported in Chapter 4, consider the design of airfoils for transonic flow conditions. Because this flow regime is very different to the subsonic regime, a slightly different mesh configuration is required for the analysis. This appendix describes a similar mesh dependence study to the one reported in Appendix A1, in this case using the RAE 2822 airfoil and the flow conditions specified by Cook et al. [1979].

The transonic airfoil mesh is an *O*-mesh topology consisting of quadrilateral cells. The CFD solver setup is identical to the subsonic configuration described in Appendix A1, except for the flow Mach and Reynolds numbers. Again, the target wall y^+ is between 30 and 60. A first cell height of 0.45mm was found to satisfy this criterion. Figure A2-1 shows the variation of the wall y^+ over the airfoil surface; the large variation is because of the large difference in near-surface velocity, but most of the points lie between 30 and 60. The height of the cells is increased as they are grown from the surface to the outer boundary; a growth factor of 1.2 is employed as this gives a significant reduction in the total number of mesh cells while maintaining a sufficient number of grid points in the region of the boundary layer flow.

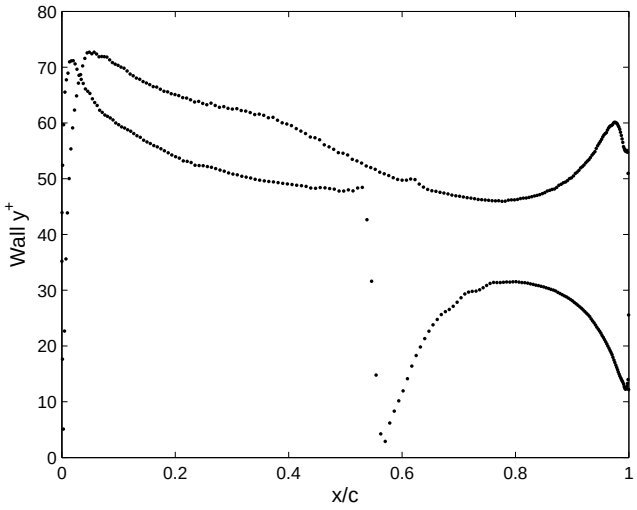


Figure A2-1 Wall y^+ for the transonic airfoil FLUENT analysis.

As in the case of the subsonic airfoil mesh, it is desirable to achieve a mesh configuration whereby any increase in cell density gives a small change in the c_d prediction. The target is that the mesh should give a c_d result no more than ±1 count from the prediction using the largest mesh evaluated. First, the domain size is selected. Figure A2-2 shows the change in c_d as the domain size is increased. The largest domain size evaluated is a radius of 100m; this is taken as the most accurate result. The point just within the tolerance of 1 count corresponds to a radius of approximately 12m; this size is selected for the final mesh.

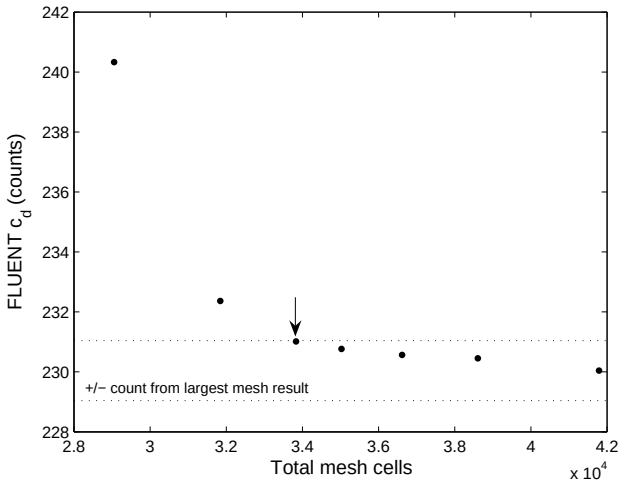


Figure A2-2 Variation of drag as the domain size is increased, showing the tolerance of acceptable accuracy.

Next, the density of cells along the chord of the airfoil is configured. Figure A2-3 shows the variation of drag as the number of mesh cells defining the airfoil surface is varied. The smallest and largest meshes correspond to 100 and 400 surface cells, respectively; the mesh with 400 surface cells is taken to give the most accurate drag result. The smallest mesh within 1 drag count of this point is the mesh with 200 surface cells. It is important that the density of cells is sufficiently high on the upper surface of the airfoil, in order to accurately predict the position of shocks. Figure A2-4 shows the surface pressure distributions predicted by FLUENT for three meshes with 100, 200 and 400 surface cells. The profiles generated using 200 and 400 surface cells align very closely, including in the region of the shock, while the profile generated using 100 surface cells shows some disparity. Thus, the use of 200 surface cells results in sufficient accuracy in terms of the prediction of drag and shock details, and hence this cell density is used for the final mesh.

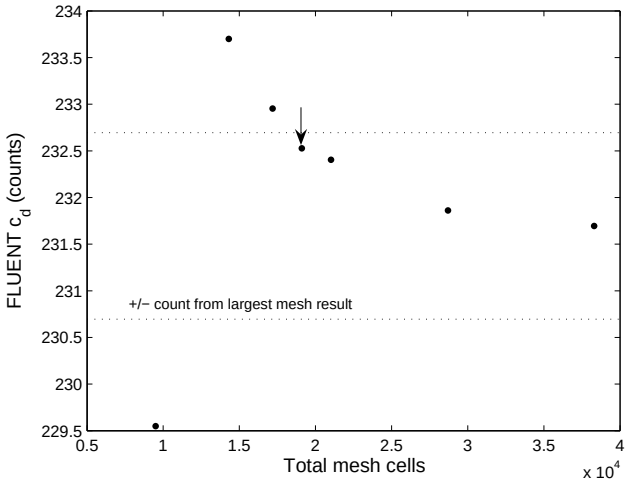


Figure A2-3 Variation of drag as the number of surface cells is increased, showing the tolerance of acceptable accuracy.

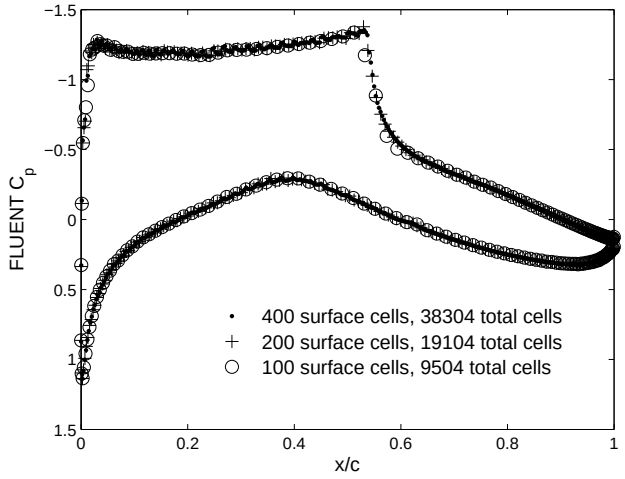


Figure A2-4 Surface pressure distributions for different FLUENT mesh configurations, varying the number of cells defining the airfoil surface.

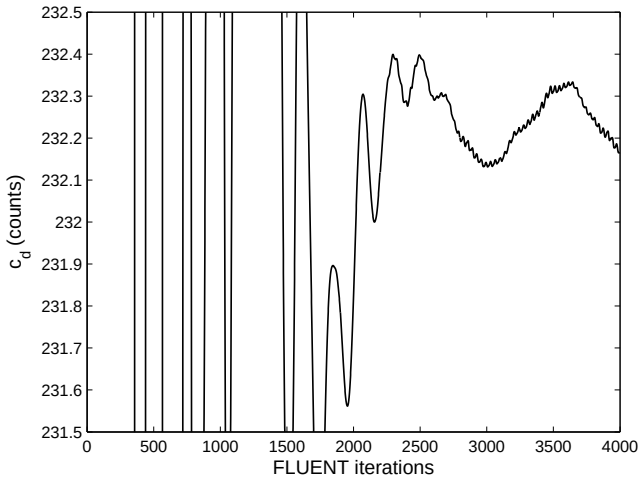


Figure A2-5 Convergence of the drag coefficient during the FLUENT solution procedure, for the transonic airfoil analysis.

The final transonic 2-D mesh configuration has 398 cells defining the airfoil surface and 48 rows, giving a total of 19104 mesh elements. The drag convergence history of the solver using this setup is shown in Figure A2-5. The convergence is rather oscillatory at the end of this trace, but it can be seen that 3000 iterations are sufficient to provide a converged solution to within ±0.1 counts (±0.00001 c_d) of the fully converged value. Table A2-1 details the mesh and solver setup for this transonic airfoil analysis.

Table A2-1 Information regarding the setup of the transonic 2-D airfoil CFD solver.

GRIDGEN v15.08		
Mesh		2-D structured, quadrilateral
First cell thickness		0.045% chord
Total mesh elements		398 columns × 48 rows = 19104 cells
Wall y^+ range (approx.)		30-60
FLUENT v6.3.19		
Solver		Coupled implicit formulation of RANS equations, also solving the energy equation
Solver boundary condition		Pressure =101325Pa, M=0.73
Solver Courant number		5
Turbulence model		Spalart-Allmaras
Turbulence boundary condition		Modified turbulent viscosity ν_t=0.001m^{2}s^{-1}
Turbulence model constants:	C_{b1}	0.1355
	C_{b2}	0.622

C_{v1}	7.1
C_{w2}	0.3
C_{w3}	2
Prandtl number	0.667
Energy Prandtl number	0.85
Wall Prandtl number	0.85
Discretization scheme	Second order upwind
Flow medium	Air as an ideal gas
Flow Mach number	0.73
Viscosity	4.59×10^{-5}kgm^{-1}s^{-1}
Reference temperature	300K
Reference density	1.177kgm^{-3}

Table A2-1 (continued)

Additionally, as stated in the main thesis text, the full potential solver (VGK) is run using non-default values for the relaxation factors when solving transonic flows. The values of the VGK relaxation factors which are changed from the default values are: *GVISCC*=0.03, *NVISCC*=20, *GVISCF*=0.015, *NVISCF*=20.

Finally, a copy of the FLUENT input journal file, used in the verification and design studies, is listed below.

```
file
read-case
"Airfoil2D.cas"
quit
grid check
def mod
solver c-i y
energy yes
visc spal-all yes
quit

def mat
change-create air air yes ideal-gas yes constant 1006.43 no yes constant 0.0000458674…
…no no no no no no
quit

def o-c
op-pres 0.0
quit

define
boundary-conditions
pressure-far-field
pressure-far-field-5
no
```

```
101325
no
0.730
no
300
no
0.998450491
no
0.055647243
yes
no

quit
quit
solve
set
courant-number
5
discretization-scheme
amg-c
1
nut
1
quit
quit
monitors
residual
n-save 10000
n-display 10000
check-convergence?
yes
yes
no
yes
no
convergence-criteria
0.0000001
0.1
100
plot?
no
print?
yes
scale-by-coefficient?
yes
quit
force
drag-coefficient
yes
wall-4

no
yes
"cd_history"
no
no
0.998450491
0.055647243
lift-coefficient
yes
wall-4

no
```

```
yes
"cl_history"
no
no
-0.055647243
0.998450491
moment-coefficient
yes
wall-4

no
yes
"cm_history"
no
no
12.0
0.0
quit
quit
quit
report
reference-values
compute
pressure-far-field
pressure-far-field-5
quit
area
1
quit
quit
solve
initialize
compute-defaults
pressure-far-field
pressure-far-field-5
quit
initialize-flow
quit
quit
solve/iterate
3000
quit
file
write-case
"Airfoil2D_run.cas"
yes
write-data
"Airfoil2D_run.dat"
yes
quit
exit
```

Appendix B: CFD Verification and Validation for 3-D Wing Analysis

This appendix details the setup of the 3-D wing analysis codes, FLUENT and VSAERO. This includes the configuration of the FLUENT mesh and the associated mesh size verification study, the setup of the VSAERO code for viscous and wake-relaxation solutions, and a comparison of the results from both codes with experimental data.

Verification of RANS Analysis

The 3-D case studies reported in Chapter 6 use the FLUENT solver to calculate drag as the design objective. This required the construction of a computational mesh for the solution of the RANS equations. A verification study has been performed to assess the accuracy of this 3-D wing analysis with respect to the mesh size and configuration. As in the 2-D airfoil mesh verification studies, the aim is to generate a mesh whereby any increase in the number of cells gives a small increase in drag. The target is to achieve an accuracy of ±1 C_D count relative to the largest mesh evaluated. The design studies documented in Chapter 6 consider the design of a wing-tip device as an extension to the ONERA-M6 wing. For the purposes of verification and validation of the CFD solvers, the ONERA-M6 wing alone is employed. Geometry and surface pressure data for the M6 wing can be found in the work by Schmitt and Charpin [1979]. The M6 wing is analysed at wind-tunnel model scale; it has a total span of 1.19m and a mean aerodynamic chord of 0.64m.

The 3-D wing mesh is generated using the commercial software GRIDGEN. The mesh employs a multi-block structured topology using hexahedral cells. A *C*-mesh block is used to surround the main wing planform, and an *H*-mesh block extends from the wing trailing edge; this topology is shown in Figure B-1. The plane parallel to the wing root is designated a *symmetry* boundary, while the remaining external boundary uses the *pressure-far-field* condition. The CFD setup for this RANS solver uses the coupled-implicit formulation of the governing equations, realizable κ-ε turbulence model and enabling second order accuracy in terms of flow and the turbulence model.

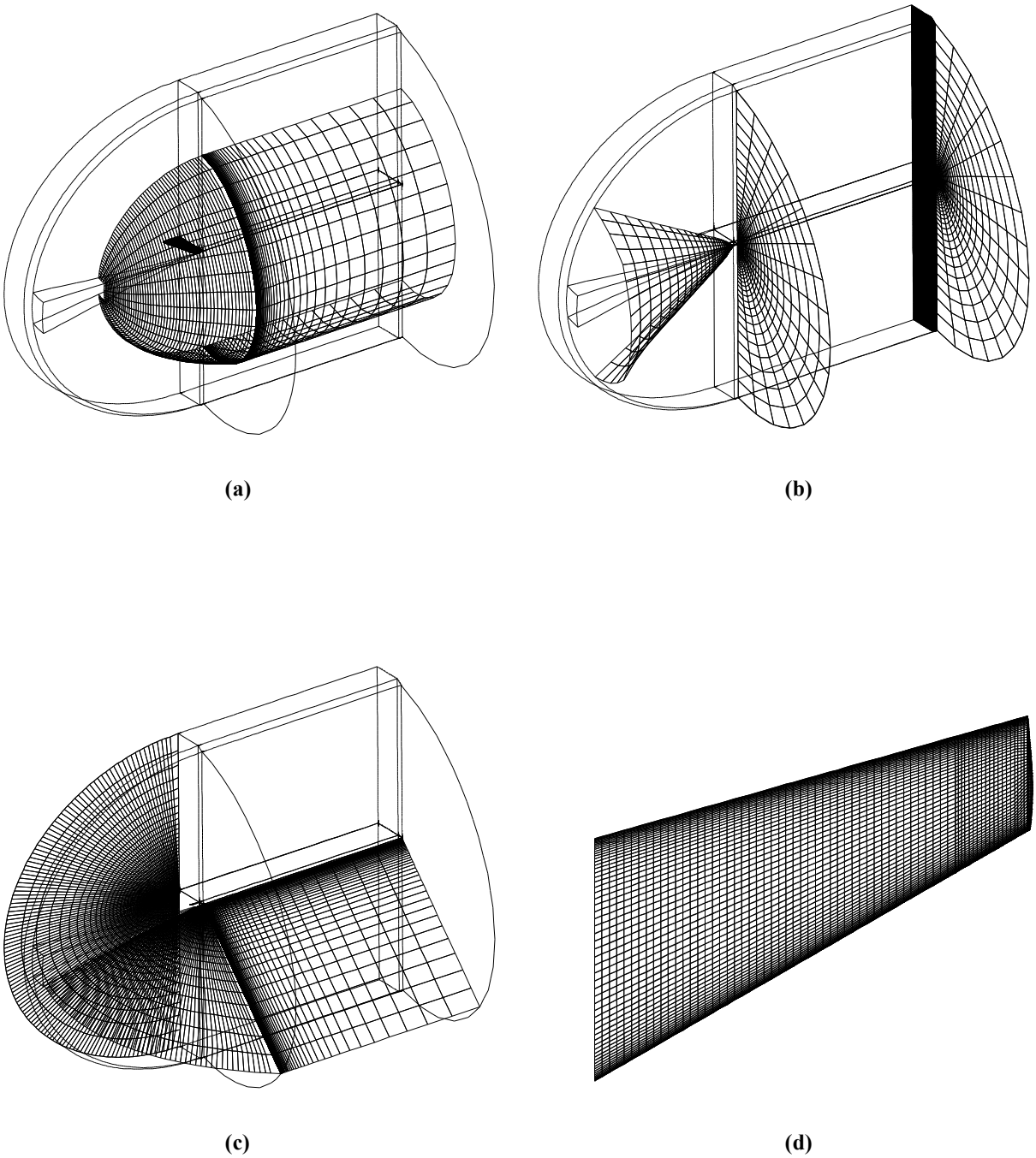


Figure B-1 Final 3-D wing RANS analysis mesh. (a) View of constant ε planes through the flow domain. (b) View of constant η planes through the flow domain. (c) View of constant ζ planes through the flow domain. (d) Planform view of the wing surface mesh.

The first step in the mesh setup process is to ensure the correct density of cells normal to the wing wall. The non-dimensional distance, or y^+, over the wing surface should be greater than 30, but ideally close to 30 in order to improve the accuracy of the log-law relationship. Thus, the first cell height was perturbed until the surface y^+ was approximately 30; a first cell of 0.15mm was found to be suitable. Figure B-2 shows the corresponding variation of wall y^+ over the wing surface.

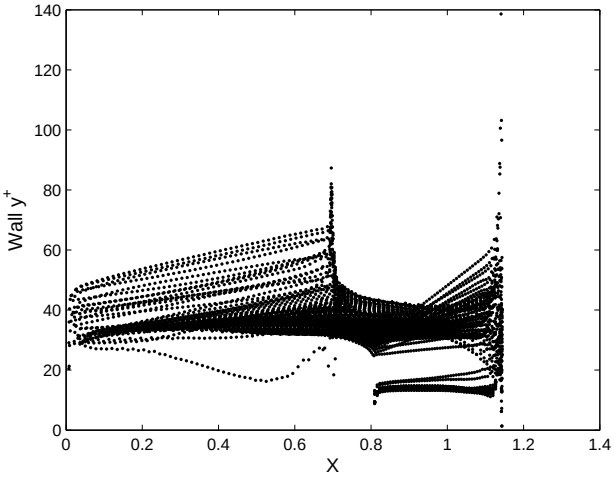


Figure B-2 Wall y^+ for a FLUENT analysis of the ONERA-M6 wing.

The hexahedral cells are grown in size with distance from the wing surface, as this significantly reduces the computational expense of the analysis. A growth factor of 1.13 is used in this case, as a higher growth rate causes the meshing algorithm to fail. Three other key parameters have been used to control the mesh configuration, and these have been varied in order to asses their effect of the predicted drag. Figure B-3 shows the variation of drag as the distance of the outer domain from the wing surface is increased. It is seen that, apart from the smallest domain distance equal to 3m, increases in the domain distance have very little effect on the predicted drag. However, the second smallest mesh in Figure B-3, corresponding to a domain distance of 5m, results in undesirable oscillations in the drag convergence. Thus, a boundary-to-wing surface distance of 8m is chosen, which equates to approximately 12 chord lengths. The number of cells on the wing surface in the chordwise direction is chosen according to the results shown in Figure B-4(a). The smallest mesh which was found to predict C_D to within 1 count of the largest mesh has 123 chordwise cells. Finally, the number of cells in the spanwise direction is primarily chosen to satisfy the requirements of the wing-tip device. Figure B-4(b) shows the variation in the predicted drag as the number of spanwise cells is increased. Again, the change in the drag results is small, and all of the evaluated meshes predict the drag to within 1 C_D count. However, subsequent to these mesh evaluations

an additional wing-tip device block was appended to the tip of the ONERA-M6 wing mesh. Because the chord distribution of this wing-tip device is varied in the 3-D case studies in Chapter 6, the density of spanwise cells must be sufficient to resolve the curvature of the chord profile. The number of spanwise cells over the wing-tip device is 20. Consequently, the meshes with fewer than 50 spanwise cells show a large discontinuous *jump* in cell density between the wing and wing-tip device, and this can produce anomalous results. Thus, 50 spanwise cells are used to construct the main wing mesh.

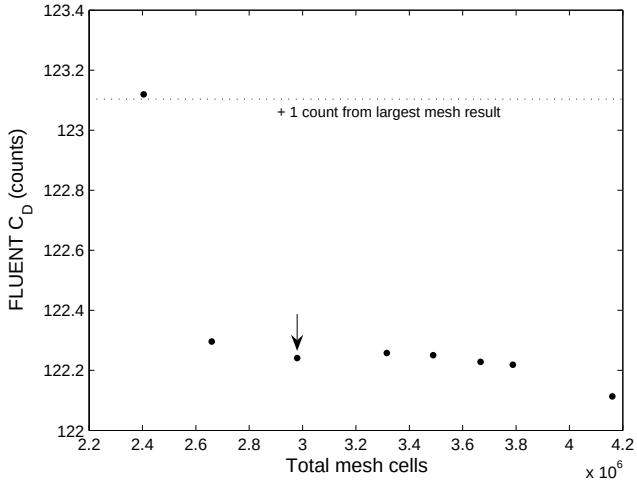


Figure B-3 **Variation of drag as the domain size is increased, showing the tolerance of acceptable accuracy.**

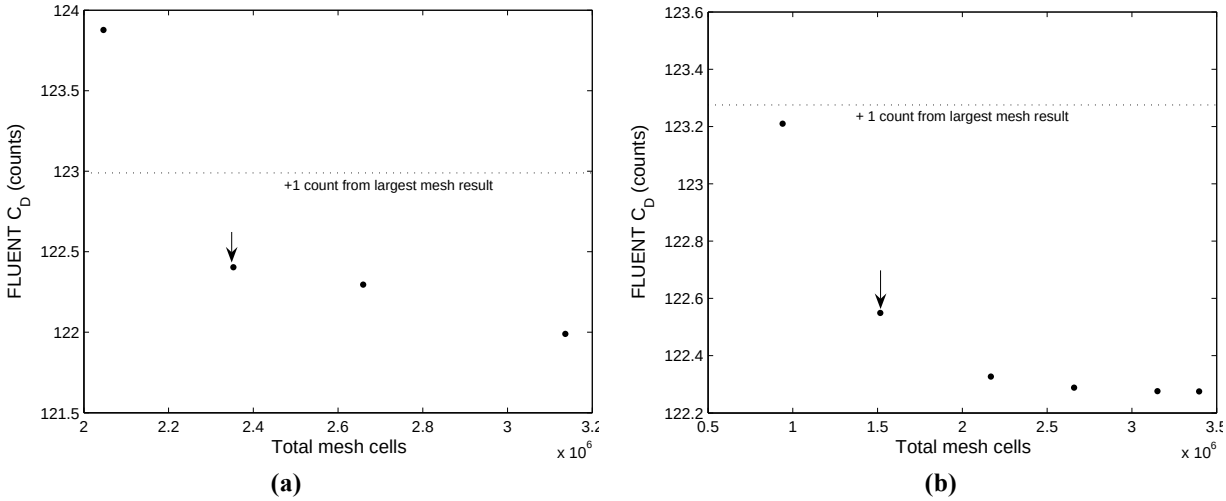


Figure B-4 **(a) Variation of drag as the number of chordwise cells is increased. (b) Variation of drag as the number of spanwise cells is increased.**

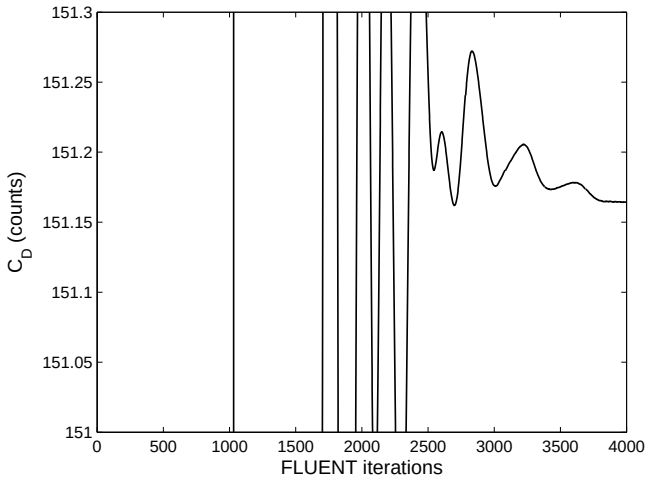


Figure B-5 Convergence of the drag coefficient during the FLUENT RANS solution procedure.

Figure B-1 shows the topology and mesh structure of the final mesh used for the RANS-based wing-tip design studies described in Chapter 6. The figure shows the blocking arrangement for the wing-tip device; in this case the wing-tip is simply an extension of the ONERA-M6 planform. The mesh setup described above is employed for all wing analyses performed during the design search, and for each new design point the mesh is generated by GRIDGEN in batch mode using a *.glf* file as input; see Appendix F for further details. Because the mesh is used for a wide range of geometries, a number of measures have been applied in an attempt to make it robust, in the sense that a good quality mesh is obtained for the vast majority of designs. For example, GRIDGEN's PDE solver is run during the batch run operation to improve the mesh quality. Also, for each new mesh generated during the design search, a check is performed to ensure that the total number of cells in the mesh is correct (1313520 cells for the RANS mesh). If the mesh does not contain the correct number of cells, the objective function evaluation is aborted. This check avoids some mesh generation errors. Other mesh errors may occur, and this is an inherent problem with automatic aerodynamic optimization processes. However, the mesh configuration described here is found to be of good quality for a large range of designs.

When the FLUENT RANS simulation is run for this mesh configuration, the drag convergence history shown in Figure B-5 is obtained. It can be seen that the drag is converged to within a small fraction of a drag count after 4000 iterations (note the small scale on the vertical axis). Because the change in the drag objective when manipulating the wing-tip chord distribution is small, 4000 CFD iterations are used in the analyses for the 3-D design studies in Chapter 6. In addition, in order to calculate drag at a fixed level of lift, the RANS analysis is run at two incidence angles consecutively; the first uses 4000 CFD iterations

and the second uses 3000 iterations. The time taken for these 7000 iterations is, on average, 16 hours 35 minutes when running on eight 2.4GHz processors. Table B-1 details the setup of the mesh and FLUENT solver for this 3-D wing RANS analysis.

Table B-1 Information regarding the setup of the 3-D wing RANS analysis.

GRIDGEN v15.08	
Mesh	3-D structured, hexahedral
First cell thickness	0.15mm
Total mesh elements	1313520 cells
Wall y^+ range (approx.)	>30
FLUENT v6.3.19	
Solver	Coupled implicit formulation of RANS equations, also solving the energy equation
Solver boundary condition	Pressure =26500Pa gauge, M=0.699
Solver Courant number	5
Turbulence model	Realizable κ-ε model (two equations)
Turbulence boundary conditions	κ=1, ε=1
Turbulence model constants: $C_{2\text{-}\varepsilon}$	1.9
TKE Prandtl number	1
TDR Prandtl number	1.2
Energy Prandtl number	0.85
Wall Prandtl number	0.85
Discretization scheme	Second order upwind
Flow medium	Air as an ideal gas
Flow Mach number	0.699
Viscosity	2.38×10^{-5}kgm^{-1}s^{-1}
Reference temperature	300K
Reference density	1.484kgm^{-3}

Finally, a copy of the FLUENT input journal file, used for the 3-D wing design studies using RANS analyses, is listed below.

```
file
set-batch-options
yes
```

```
yes
no
quit
file
read-case
"3D_grid.cas"
quit
def mod
solver density-based-implicit yes
energy yes
visc ke-realizable yes
visc near-wall-treatment non-equilibrium-wall-fn yes
quit

def mat
change-create air air yes ideal-gas yes constant 1006.43 no yes constant 2.38441e-5…
…no no no no no
quit

define
boundary-conditions
pressure-far-field
pressure-far-field-5
no
26500
no
0.699
no
300
no
%COS_ALPHA%
no
%SIN_ALPHA%
no
0
yes
no

no

quit
quit
solve
set
discretization-scheme
amg-c
0
epsilon
0
k
0
quit
quit
monitors
residual
n-save 10000
n-display 10000
check-convergence?
yes
yes
no
no
yes
```

```
no
no
convergence-criteria
0.0000001
0.0000001
0.0000001
scale-by-coefficient yes
plot?
no
print?
yes
scale-by-coefficient?
no
quit
force
drag-coefficient
yes
wall-4

yes
yes
"cd_history.txt"
no
no
%COS_ALPHA%
%SIN_ALPHA%
0
lift-coefficient
yes
wall-4

yes
yes
"cl_history.txt"
no
no
%NEG_SIN_ALPHA%
%COS_ALPHA%
0
moment-coefficient
yes
wall-4

yes
yes
"cm_history.txt"
no
no

0.0
0.0
0.0
quit
quit
quit
report
reference-values
compute
pressure-far-field
pressure-far-field-5
quit
area
0.75322
```

```
quit
quit
solve
initialize
compute-defaults
pressure-far-field
pressure-far-field-5
quit
initialize-flow
quit
quit
solve/iterate
1000
quit
solve
set
discretization-scheme
amg-c
1
epsilon
1
k
1
quit
quit
iterate
3000
quit
file
write-case
"wing2.cas.gz"
yes
write-data
"wing2.dat.gz"
yes
write-profile
CpData.prof
wall-4

pressure-coefficient
quit
write-profile
y-plusData.prof
wall-4

y-plus
quit

quit
parallel timer print
quit
exit
yes
```

Euler Analysis Configuration

The design case studies reported in Chapter 6 make use of an Euler solver for the calculation of the induced drag of the wing. Since the Euler equations do not consider the effects of viscosity, the computational mesh can employ less densely spaced cells adjacent to the wing surface in the normal direction, compared to the mesh used for the RANS analysis. The mesh described above for the RANS analysis is used as a baseline for the Euler mesh; this mesh configuration is sound for solving the RANS equations, and therefore should be sound as an Euler mesh.

Figure B-6 shows the variation of the drag predicted by the FLUENT Euler solver, as the size of the first cell adjacent to the wing is increased. Because the growth factor of the cells and the external domain size is constant, increasing the first cell size reduces the total number of cells in the mesh. Surprisingly, increasing the first cell height has a significant effect on the induced drag, despite the absence of the boundary layer flow. However, this loss of accuracy is accepted in favour of a lower computational cost of this Euler analysis. The first cell height is increased to 1.5mm (compared to 0.15mm in the RANS mesh), giving a total mesh size of 796720 cells.

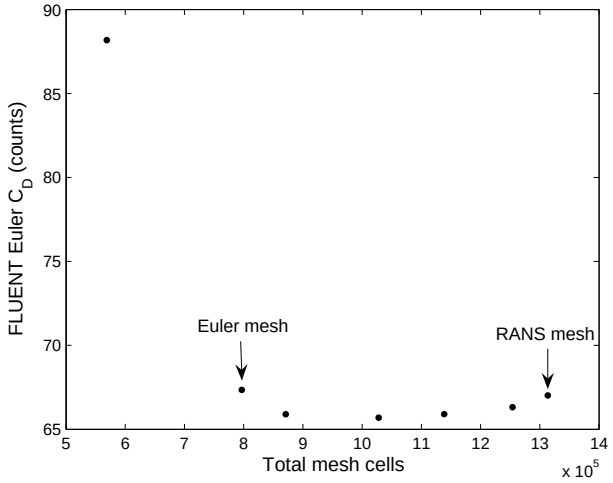


Figure B-6 Variation in the FLUENT Euler drag as the first cell height is increased (this reduces the total number of mesh cells).

In order to calculate drag at a fixed level of lift, the Euler analysis is run at two incidence angles consecutively; the first uses 4000 CFD iterations and the second uses 3000 iterations. The time taken for each complete Euler analysis is, on average, 6 hours 38 minutes when running on eight 2.4GHz processors. A summary of the Euler analysis setup is given in Table B-2.

Table B-2 Information regarding the setup of the 3-D wing Euler analysis.

GRIDGEN v15.08	
Mesh	3-D structured, hexahedral
First cell thickness	1.5mm
Total mesh elements	796720 cells
FLUENT v6.3.19	
Solver	Coupled implicit formulation of Euler equations, also solving the energy equation
Solver boundary condition	Pressure =26500Pa gauge, M=0.699
Solver Courant number	5
Discretization scheme	Second order upwind
Flow medium	Air as an ideal gas
Flow Mach number	0.699
Reference temperature	300K
Reference density	1.484kgm^{-3}

For completeness, below are the input commands from the start of the log file used to execute the FLUENT Euler analysis. These commands differ from the RANS input log file. Apart from these commands, the remainder of the Euler input file is the same as in the RANS input file.

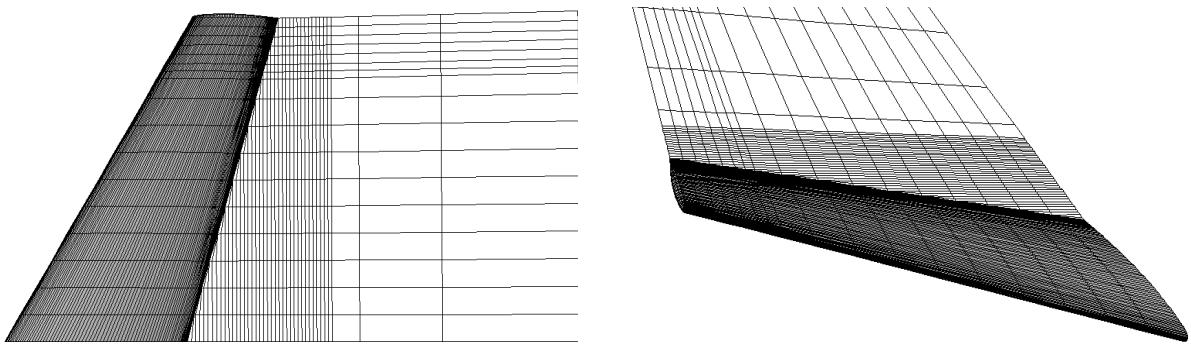
```
file
set-batch-options
yes
yes
no
quit
file
read-case
"3D_grid.cas"
quit
def mod
solver density-based-implicit yes
energy yes
visc inviscid yes
quit

def mat
change-create air air yes ideal-gas yes constant 1006.43 no no no no no no
quit
```

VSAERO Setup

VSAERO is a CFD code based on the vortex panel method, and is able to calculate the non-linear aerodynamic characteristics of an arbitrary geometry. In this thesis, VSAERO is used for the purposes of inverse design as part of the EMFID parameterization strategy. An advantage of this code is the very low computational cost of running each analysis. The code can predict the wing surface pressure to a reasonable degree of accuracy, but since the flow is irrotational the flow vorticity is limited to the surface panels.

A further advantage of the VSAERO code is that it requires discretization of only the body and wake surfaces, and not the entire flow domain. The surface panelling scheme used in this work is the same as that used by Robinson and Keane [1999] and Petruzzelli and Keane [2001]; this is shown in Figure B-7. The wing surface is discretized using 17 panels in the spanwise direction and 103 chordwise panels on both the upper and lower wing surfaces. The wake panels are constructed on the horizontal plane, with 38 panel rows densely spaced near the wing and a further 11 rows extending downstream.



(a) **(b)**

Figure B-7 Panelling scheme used in VSAERO. (a) Planform view. (b) Front isometric view.

VSAERO includes a wake relaxation procedure to provide corrections to the wake panel structure, and a viscous-coupled element for predicting the effect of the boundary layer. Each of these schemes is performed iteratively. Figure B-8(a) shows the number of wake relaxation iterations plotted against the VSAERO predicted induced drag. It is seen that after just 1 iteration the drag prediction is dramatically improved, and is within 1 drag count of the prediction using 10 wake iterations. Thus, a single wake relaxation iteration is used when predictions of the trailing vorticity are required. Similarly, Figure B-8(b)

shows the number of viscous iterations plotted against the predicted total drag. It is seen that 1 iteration has no effect on the solution. This is because VSAERO performs the integral boundary layer calculations after the drag computation. Therefore, 2 viscous iterations have an effect on the drag, resulting in a much improved drag figure; hence, 2 iterations of the viscous scheme are used when viscosity is to be considered in the inverse design. Each call to VSAERO takes, on average, 41.5 seconds when the 2 viscous iterations are required, and 35.1 seconds for an inviscid solution.

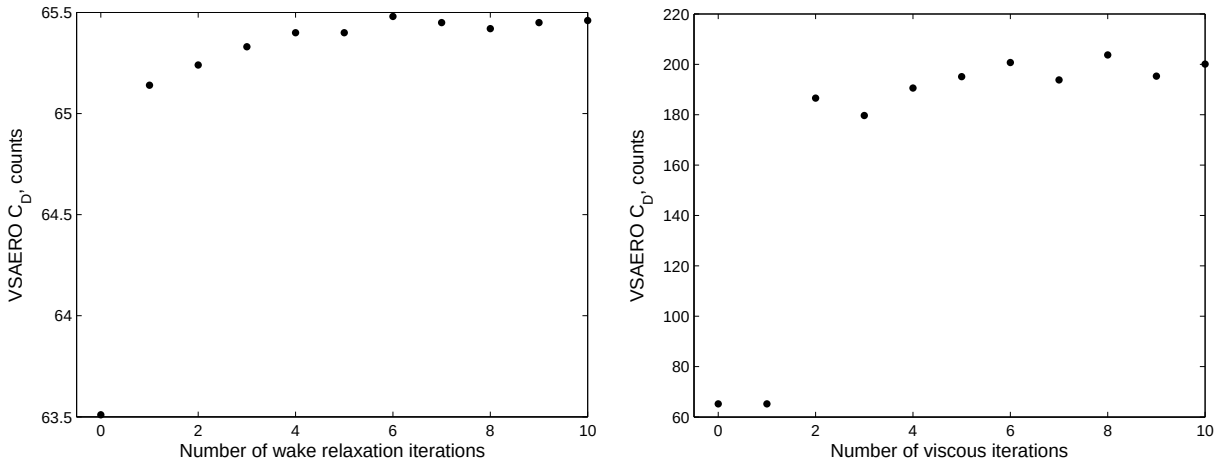


Figure B-8 (a) Number of wake relaxation iterations vs. VSAERO drag. (b) Number of viscous iterations vs. VSAERO drag.

3-D Wing Analysis Validation

The FLUENT and VSAERO solvers have been validated using the experimental data for the ONERA-M6 wing published by Schmitt and Charpin [1979]. In addition to the comparison shown in Figure 5-7, Figure B-9 shows the surface pressure distributions at six other spanwise stations along the M6 wing. Note that, for clarity, only the RANS FLUENT and viscous VSAERO results are compared in Figure B-9. The Euler and inviscid VSAERO surface pressure results, respectively, are almost identical to the results shown.

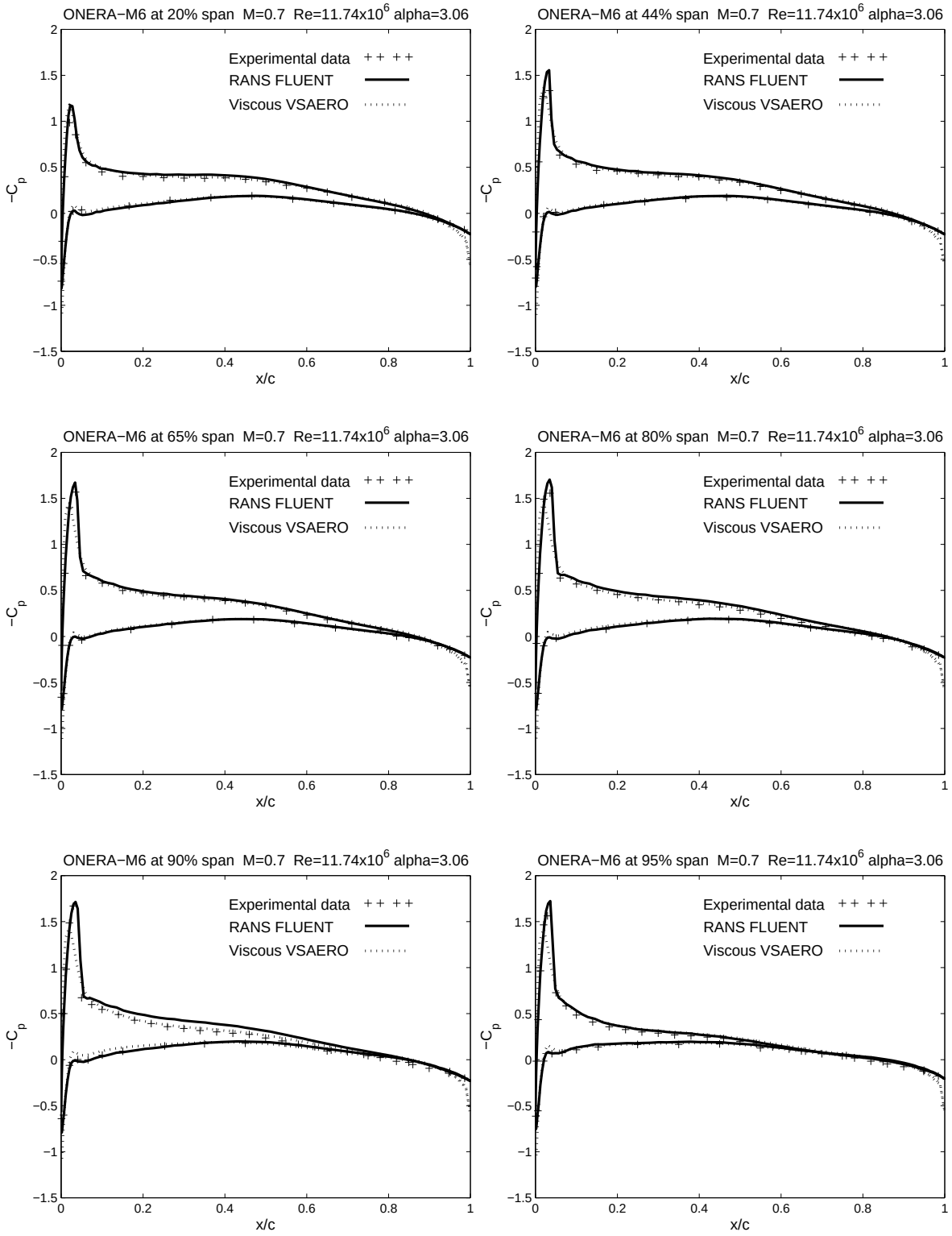


Figure B-9 Surface pressure distributions over the ONERA-M6 wing at six spanwise stations, predicted by the FLUENT RANS analysis and viscous VSAERO simulations.

Appendix C: Investigating the Wing Tip Vortex as the Parameterized Flow Feature in EMFID

In addition to the discussion in Section 5.6, this Appendix reports an investigation into the feasibility of using a parameterization of wing tip vortex properties to design a wing-tip device for minimum drag.

The subject of wing-tip vortex production and suppression has seen significant research, largely motivated by the desire to minimize induced drag and reduce the safe distance between passenger aircraft to increase airport capacity (Spalart [1998]). Recent research has made use of classical lifting line theory (an example is the work by Rossow [2006]), numerical computation (Murayama et al. [2001]) and wind-tunnel testing (Gerontakos and Lee [2006a]; Zhang et al. [2006]). Notable research specific to winglet configurations is that of Gerontakos and Lee [2006b], who investigated experimentally the effect of winglet dihedral on the vortex strength, and found that the dihedral can reduce induced drag; also Eppler [1997] used a development of classical lifting line theory which considers induced lift, and concluded that *winglets up is much better than winglets down*. Unfortunately, the relationship between the vorticity and geometry is non-linear in nature, and this complexity is increased when one considers the roll-up of the wake. It is therefore likely that a more accurate prediction of the vortex position and strength is to be achieved by means of numerical simulations, such as those performed using VSAERO, rather than analytically.

In order to manipulate the wing design using the tip vortex, the properties of the vortex must first be identified. There is a large number of flow parameters associated with a real vortex, but the VSAERO code assumes all vorticity lies on the wake lines and therefore assumes a point vortex. Although the vortex structure is simplified somewhat, one can extract useful information in the form of the vortex strength and position; further, the integration of these point vortices gives the cross-flow circulation. Therefore, the properties considered to be of interest here are the tip vortex (Y,Z) position at a constant X plane downstream of the wing, the maximum vorticity of the vortex and the total circulation on the (Y,Z) plane. As a primary study, these flow features should be extracted from the VSAERO and FLUENT

results and compared, as this provides an indication of the potential for a vortex-based parameterization in EMFID. The flow feature extraction technique for each of these variables is now considered in turn.

The position of a vortex centre can be found using a number of techniques, as reviewed by Jeong and Hussain [1995]. These include finding the point of minimum pressure, maximum vorticity or regions featuring closed or spiraling streamlines. The method used here is the VORTFIND method described by Pemberton [2003]. A 2-D slice of the flow on a (Y,Z) plane downstream from the wing-tip is considered, and using the velocities at each data point the resultant velocity vectors on this cross-flow plane are determined. Strictly, the plane should be normal to the axis of rotation of the vortex; however, it is assumed to be normal for small angles of attack. The angle, α, of each velocity vector relative to the Y axis is calculated. Each data point is then assigned a value of β according to the inequalities

$$\begin{aligned} 0° \le \alpha < 120° \qquad & \beta = 0, \\ 120° \le \alpha < 240° \qquad & \beta = 1, \\ 240° \le \alpha < 360° \qquad & \beta = 2. \end{aligned} \tag{C.1}$$

Next, for each data point the two nearest points with different values of β, relative to each other and itself, are found. The distances to the nearest values of β=0, 1, and 2 are p, q and r, respectively. Thus, one of p, q or r is always zero. Then, for each point the expression

$$l = p^2 + q^2 + r^2 \tag{C.2}$$

is computed. The vortex centre is defined as the point where l is minimum, i.e., the point which is closest to points with different values of β. This approach is used to locate the vortex centre as predicted by the VSAERO and FLUENT solvers. The cross-flow plane is taken at X=4m in all cases; this position is chosen as it is approximately equidistant from the wing and the FLUENT far-field boundary, and at this plane the FLUENT mesh cells are acceptably dense in the (Y,Z) region of the vortex.

The point of maximum vorticity is not, in general, at exactly the same location as the vortex centre as defined above. Therefore, the vorticity, ξ, is calculated across the entire cross-flow plane as the curl of the velocity, V,

$$\xi = \nabla \times V = \left(\frac{\partial w}{\partial Y} - \frac{\partial v}{\partial \mathrm{Z}} \right) \approx \left(\frac{\mathrm{d}w}{\mathrm{d}Y} - \frac{\mathrm{d}v}{\mathrm{dZ}} \right), \tag{C.3}$$

where v and w are the point velocities in the spanwise and vertical directions, respectively. Because the gradient terms on the right hand side of Eq. (C.3) are computed by finite differencing, the velocities given at the mesh points over the cross-flow plane are interpolated onto a regular grid. Cohn and Koochesfahani [2000] describe the accuracy of various approaches for remapping velocity fields; a simple polynomial interpolation is used for this study. From this, Eq. (C.3) is evaluated for every data point and the maximum vorticity is found. Using the vorticity field, the total circulation, Γ, over the same cross-flow plane is computed using

$$\Gamma = -\int_A (\nabla \times V) \cdot \mathrm{d}A = -\iint \xi \cdot \mathrm{d}Y\mathrm{d}Z \,, \tag{C.4}$$

where A is an area of 5m×5m in the plane of integration, enclosing the wake of the whole wing. The integration is performed numerically using first-order quadrature.

Having discussed the techniques used to identify the vortex position and strength, the predictions of VSAERO and FLUENT are now compared. In this study, the VSAERO and FLUENT solvers are run for a set of geometries generated using the *gross* winglet parameterization shown in Figure 5-8. The total half-wing span is 1.35m, or a span extension of 0.15m due to the wing-tip device. 50 different geometries are generated, whose defining variables are dictated by a random Latin hypercube DoE based on the variable bounds given in Table 5-1. Both CFD solvers are run in inviscid mode at a fixed angle of attack of 3°, and VSAERO additionally performs the wake roll-up calculation. The aim is to compare the relative performance trends with respect to vortex strength, vortex position and drag for the two codes.

The first of the 50 evaluated designs is a simple extension of the ONERA-M6, i.e., a wing-tip which continues the sweep and taper and has zero dihedral. From the outset, it is interesting to compare the predictions for this initial geometry with those corresponding to the best and worst of the 50 designs generated using the DoE plan. The best wing-tip design, in terms of the drag predicted by FLUENT, incorporates significant twist (washout), an increase in sweep and a very small dihedral angle. The worst design (with the highest FLUENT drag) has a large dihedral angle and negative twist (washin). Figure C-1, Figure C-2 and Figure C-3 show the geometry, velocity vectors and vorticity contours for the initial, best and worst geometries, respectively. In all three cases, the velocity and vorticity data are from the FLUENT (Euler) simulations, and a black cross on the vorticity plots indicates the vortex core centre as predicted by VSAERO. Table C-1 additionally gives the maximum vorticity, circulation and drag data from both solvers, for the three designs. Note that the VSAERO velocity data is normalized by dividing by a reference velocity, and hence the vorticity and circulation magnitudes for VSAERO are relative.

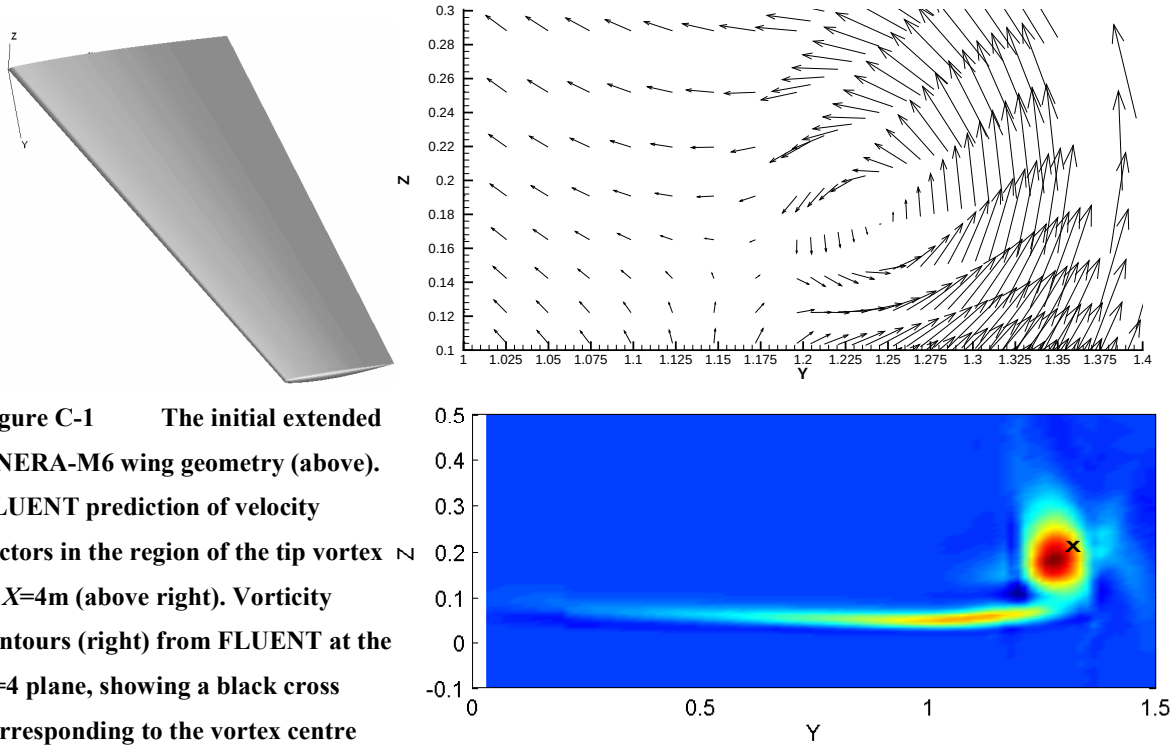


Figure C-1 The initial extended ONERA-M6 wing geometry (above). FLUENT prediction of velocity vectors in the region of the tip vortex at *X*=4m (above right). Vorticity contours (right) from FLUENT at the *X*=4 plane, showing a black cross corresponding to the vortex centre predicted by VSAERO.

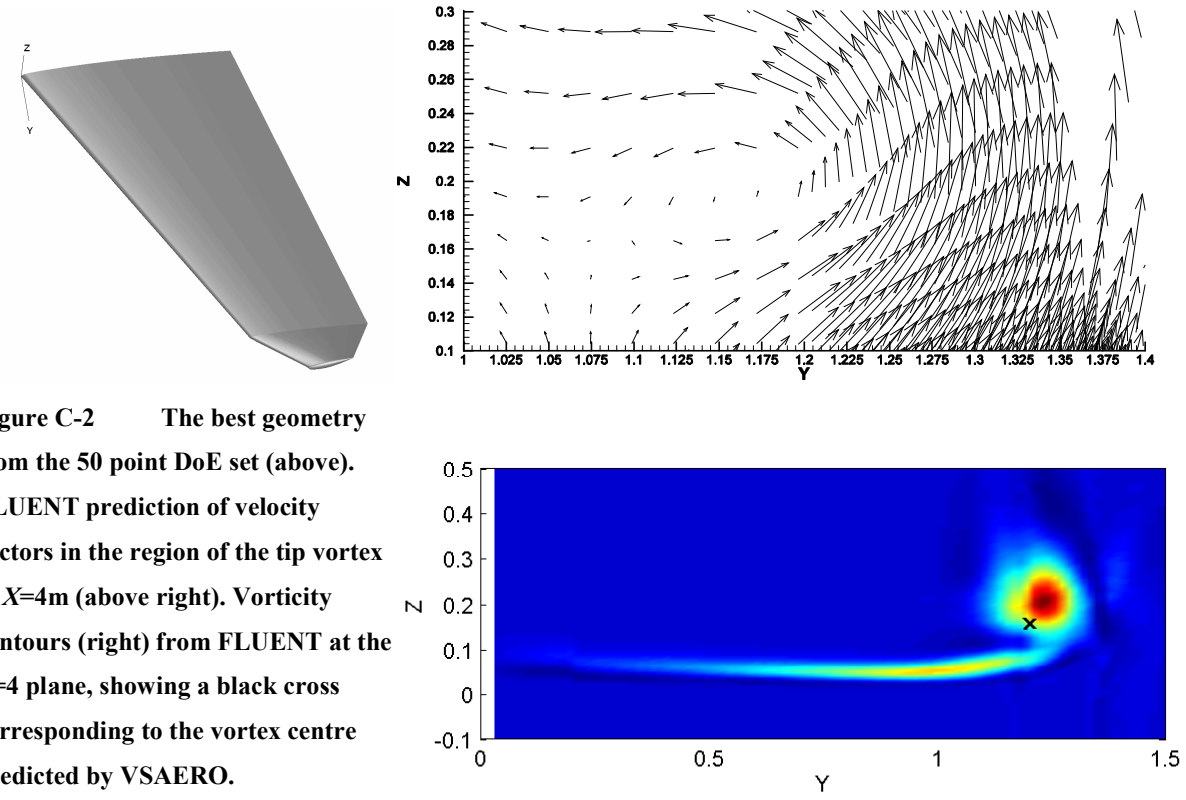
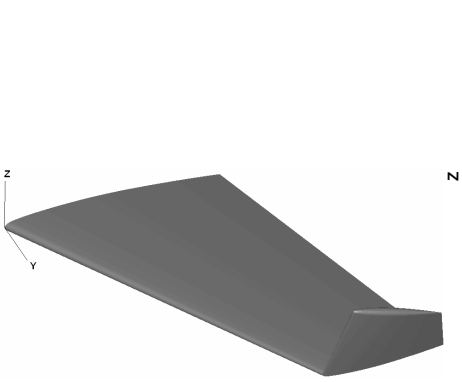
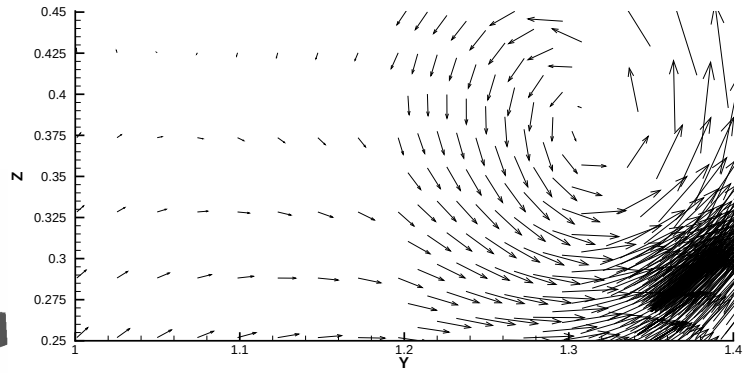


Figure C-2 The best geometry from the 50 point DoE set (above). FLUENT prediction of velocity vectors in the region of the tip vortex at *X*=4m (above right). Vorticity contours (right) from FLUENT at the *X*=4 plane, showing a black cross corresponding to the vortex centre predicted by VSAERO.





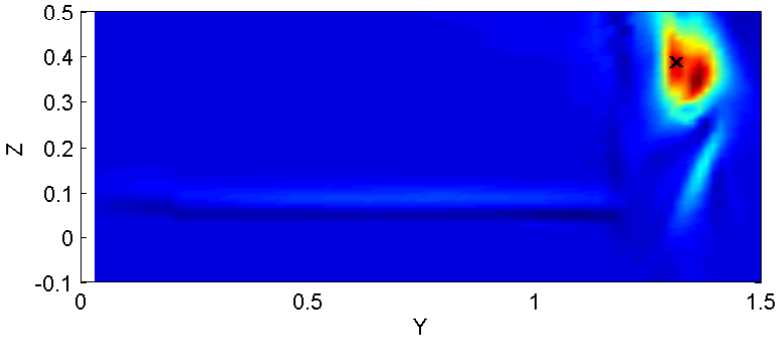


Figure C-3 The worst geometry from the 50 point DoE set (above). FLUENT prediction of velocity vectors in the region of the tip vortex at *X*=4m (above right). Vorticity contours (right) from FLUENT at the *X*=4 plane, showing a black cross corresponding to the vortex centre predicted by VSAERO.

Table C-1 Data relating to the wing-tip vortex for the initial, best and worst designs in the 50 point DoE set, as predicted by FLUENT and VSAERO.

DoE design point		(*Y*,*Z*) plane at *X*=4m				C_D (counts)
		Vortex centre *Y*	Vortex centre *Z*	Max. ξ	*Γ*	
1 (extended ONERA-M6)	FLUENT	1.2414	0.18276	808.934	22.6546	73.884
	VSAERO	1.3175	0.2102	187.875	0.088132	102
26 (best FLUENT C_D)	FLUENT	1.175	0.20	471.2886	22.3097	67.474
	VSAERO	1.2033	0.1555	45.75	0.086989	87.34
10 (worst FLUENT C_D)	FLUENT	1.3081	0.3798	921.150	23.9952	129.25
	VSAERO	1.3126	0.3858	364.5	0.091761	112.9

Figures C-1 to C-3 illustrate that VSAERO predicts the location of the vortex centre with reasonable accuracy compared to FLUENT. The Euclidean distance between the vortex positions predicted by the two CFD codes ranges between 7.5-80mm, for the three illustrated designs. However, based on the

vertical positions given in Table C-1, it can be concluded that the vortex position has a weak coupling with drag and is therefore unlikely to be suitable as a design variable in EMFID. The maximum vorticity and circulation in the cross-flow plane are likely to be more suitable, and one would expect an increase in these parameters might increase the drag. Indeed, Table C-1 shows that, for both FLUENT and VSAERO, an increase in the peak vorticity and circulation follows a monotonic increase in drag. Investigating these effects further, Figure C-4 shows the drag and peak vorticity calculated using FLUENT for all 50 designs evaluated. For clarity, the data has been sorted in ascending order with respect to the FLUENT drag data. With the benefit of more data, it becomes apparent that, in general, there is no correlation between the peak vorticity and drag. This can be explained, since there can exist a vortex flow which contains a *hotspot* of vorticity but for which the cumulative vorticity across the wake (i.e, the circulation) can be rather low. Conversely, it is possible for a large and energetic vortex to have a low peak vorticity. Therefore, this quantity would not be suitable when used as a design variable; rather it makes sense to use an integrated quantity, such as circulation.

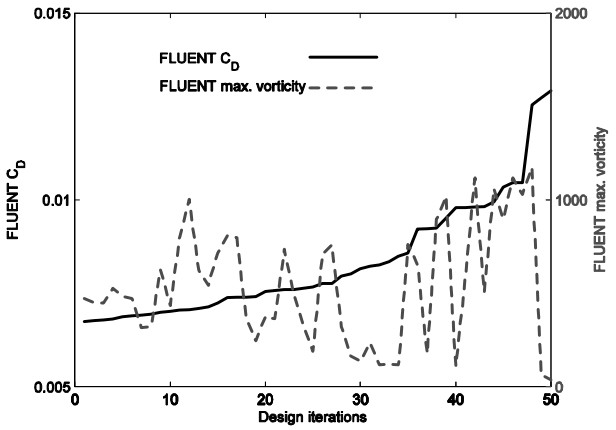


Figure C-4 FLUENT predictions for drag and maximum vorticity for all 50 designs, sorted by the drag values.

Figure C-5 (a) gives the same FLUENT drag curve as Figure C-4, but in this case it is shown with the circulation, also from FLUENT, for all designs. It can be seen that there is a correlation between total circulation and drag, and this is expected since the calculation of circulation (an integration of vorticity over the cross-flow plane) is directly related to the calculation for induced drag (integration of cross-flow velocities). Given this fact, the similarity between the curves in Figure C-5 (a) is not as close as one would expect. This is likely to be due to the numerical dissipation of the Euler solution as the flow moves away from the wing trailing edge, which does not affect the drag because drag is computed using the wing surface pressure. Also, the mesh cells in the region of the vortex are inevitably coarser than those on

the wing surface; hence, the process of interpolating the velocities onto the integration grid suffers a loss of accuracy. Nevertheless, if the circulation from FLUENT was made to be a design variable and one was to minimize this variable, the resulting wing design would give good performance in terms of drag. However, for efficiency EMFID requires the use of low-fidelity CFD for inverse design. Figure C-5 (b) shows the circulation results from VSAERO for all 50 designs, shown with the same FLUENT drag curve. While the similarity between these curves is not as strong as shown in Figure C-5 (a), both show a similar upward trend. Design iterations 18, 21, 24 and 28 are rather anomalous, but it can be seen that minimizing the circulation predicted by VSAERO would result in a design with relatively low FLUENT drag. However, since the circulation is broadly equivalent to induced drag, an EMFID design search which uses a parameterization of circulation would result in an exercise of minimising the drag predicted by VSAERO. This effectively amounts to a multi-fidelity *zoom* approach to optimization. Thus, a parameterization of the crossflow circulation is not suitable for the EMFID design search process.

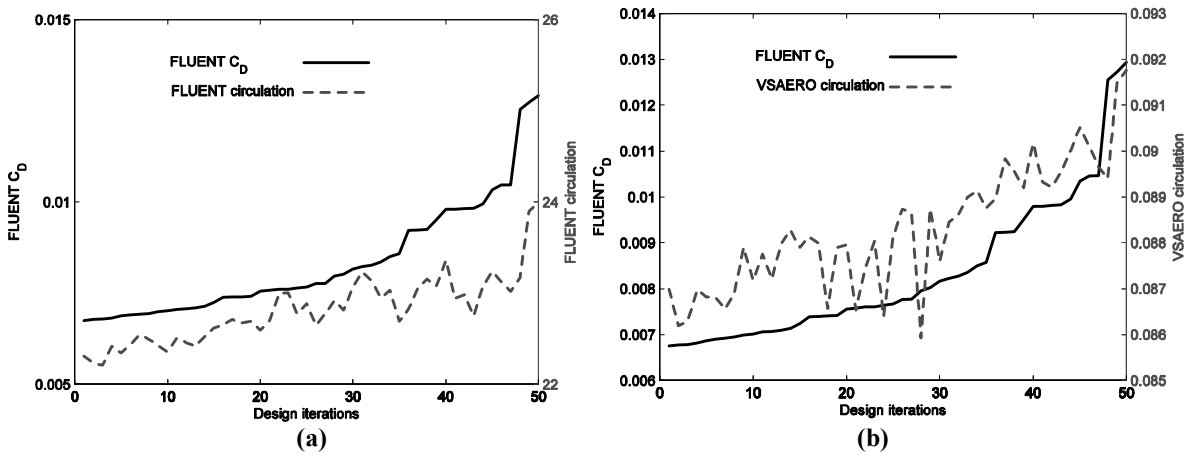


Figure C-5 (a) FLUENT predictions of drag and circulation for all 50 design points, where the points have been sorted in ascending drag order, (b) FLUENT drag predictions and VSAERO circulation results for the same designs.

In summary, this section has described the extraction of flow features relating to the wing tip vortex. These flow features have been simulated using FLUENT and VSAERO for a range of geometries. It is found that the predictions of the vortex position from the two codes agree to acceptable accuracy. However, the VSAERO predictions for these position parameters and the peak vorticity do not correlate well with the design objective, drag, and are therefore not suitable design variables for use in the EMFID design search process.

Appendix D: Design Trends from the 3-D Case Studies

In Chapter 6, case studies 5 and 6 reported the results of a comparison between the EMFID and benchmark methods for the design of a wing-tip device. This section examines the geometrical features of the designed wing-tip devices in more detail. One of the reasons for performing systematic design search studies using computational analyses is to identify and understand how the design features of interest are related to aerodynamic forces, and how the design could be manipulated in order to achieve improved performance. The case studies described in Chapter 6 focused on the design of the trailing edge chord distribution in a region close to the wing tip, with the objective of minimizing drag. The EMFID and benchmark methods, which differ in the approach used to manipulate the design, have resulted in similar conclusions in the sense that the best designs all share common geometrical features. In particular, a large chord is applied at the wing tip, and, when the analysis considers the drag due to viscosity, the wing wetted area is strictly controlled in the region further inboard. Figure D-1 shows the best design generated using the benchmark method in case study 6, as well as two modified versions of this geometry. In what follows, the effects of these two design changes are investigated in turn using additional CFD analyses.

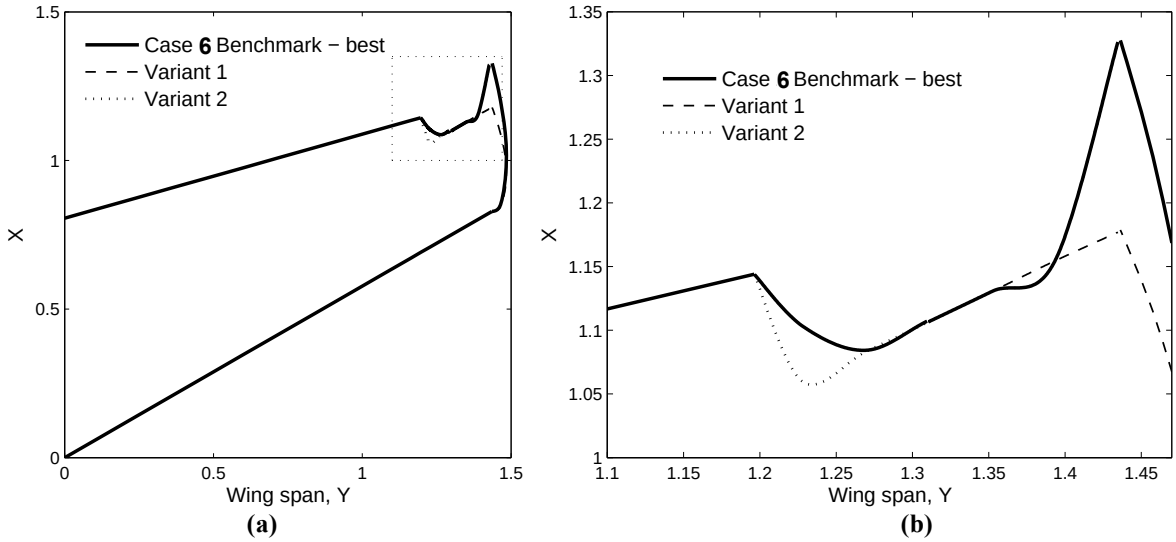


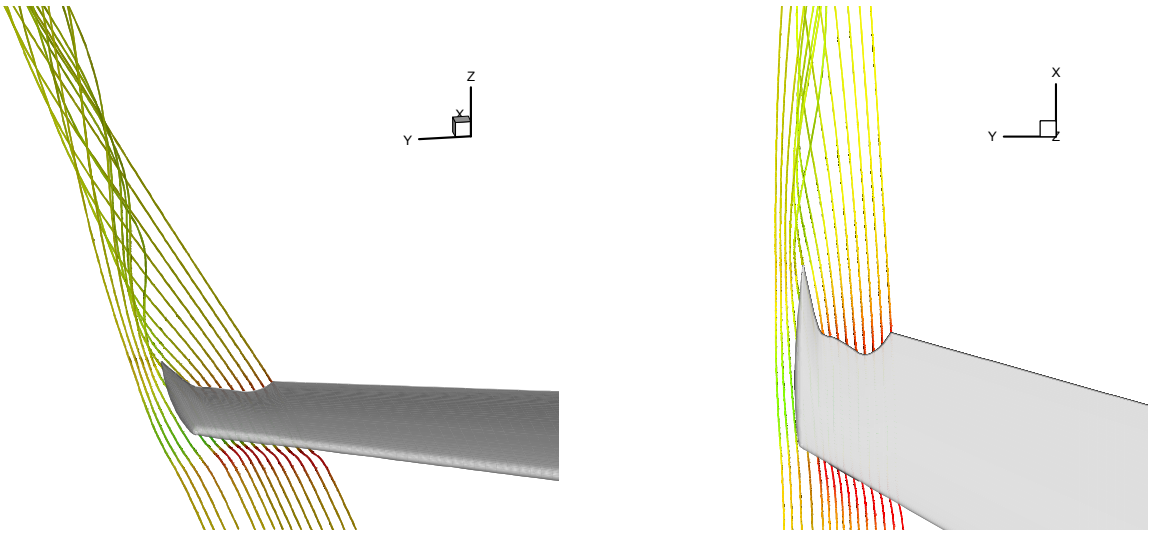
Figure D-1 The best benchmark geometry from case study 6 (RANS simulations), shown with two variants of this design. (a) On equally scaled axes. (b) A close-up view of the wing-tip region.

Table D-1 Drag coefficients calculated using FLUENT Euler and RANS analyses for the three designs in Figure D-1.

Design	Drag coefficient, C_D, computed using FLUENT analysis at C_L=0.4	
	Euler	RANS
Case study 6 – best benchmark	141.173	217.808
Variant 1 (Figure 6-11)	141.923	218.397
Variant 2 (Figure 6-11)	141.387	217.929

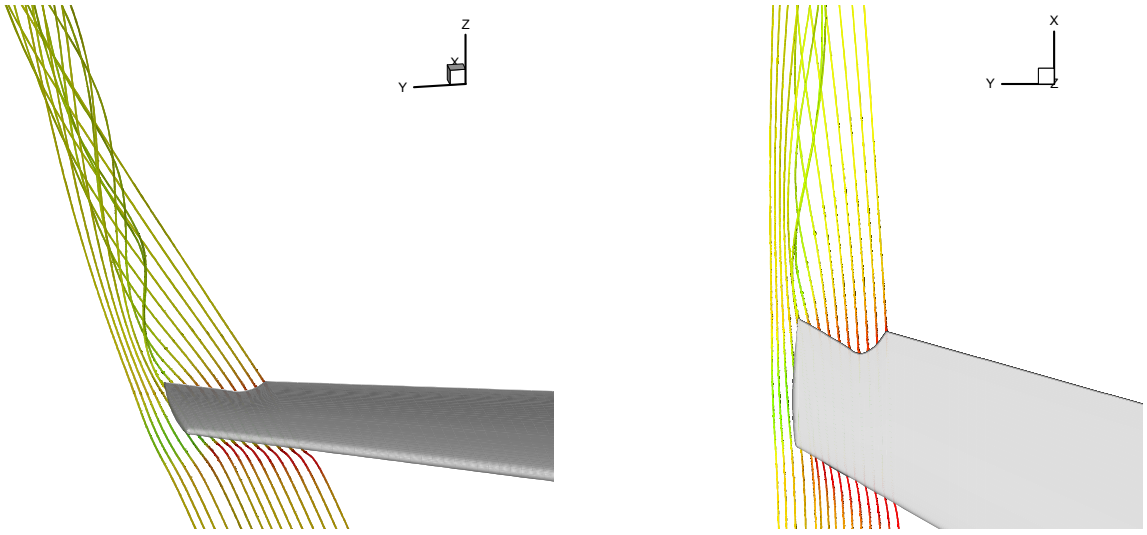
Sweeping back the tip section of the wing has the effect of reducing the lift-induced drag, a result also reported by van Dam [1987] and Burkett [1989], among others. This design feature is common to the final (best) geometries resulting from the two design search methods in both case study 5, which used Euler FLUENT drag calculations, and in Case 6, which employed RANS simulations. It is useful to be able to quantify the change in drag when the tip section is swept back. The *variant 1* design (Figure D-1) is defined as follows. Design variables 1 to 5, which define the five inboard chord values, are set to be the same as the optimized benchmark design, while variable 6, defining the tip chord, takes the value of the lower problem bound. Thus, the sweep at the tip station is reduced considerably. The FLUENT analysis has been run for both the benchmark design and variant 1, using both the Euler and RANS solvers; Table D-1 gives the resulting drag coefficient values. From the results generated using the Euler solution, it is

seen that removing the aft-swept tip increases the induced drag by 0.75 counts, or 0.5%. The RANS analyses additionally consider viscous and boundary layer effects. Since removing the tip sweep reduces the wetted area, the increase in induced drag is partially offset by a reduction in viscous drag. However, overall there remains a net increase in drag of 0.59 counts, or 0.27%. It is interesting to visualize the influence of an aft-swept tip on the trailing vortex flow; Figure D-2 and Figure D-3 show streamlines emitted from the trailing edge of the benchmark design and variant 1, respectively. Since the relative drag levels are similar, the changes to the vorticity are subtle. However, it can be seen that when the tip sweep is removed the streamlines adjacent to the tip-cap are more perturbed, and the trailing vortex grows in size more quickly with distance from the wing.



(a) **(b)**

Figure D-2 **Flow visualization showing streamlines emitted from the trailing edge of the wing-tip device, for the best design from case study 6. (a) planform view, (b) front isometric view.**



(a) **(b)**

Figure D-3 **Flow visualization showing streamlines emitted from the trailing edge of the wing-tip device, for a design with the tip chord minimized. (a) planform view, (b) front isometric view.**

The second design trend scrutinized here is the link between the chord distribution inboard of the tip and drag. In case study 5, it was found that there are a range of geometries which can yield good performance, suggesting that there is no dominant design trend. In Case 6, in order to minimize the skin friction drag the chord over the wing-tip device was minimized, except at the tip and at the furthest inboard control point. As previously described, the tip chord is maximized in order to reduce the induced drag. However, although the trend is that minimizing the inboard variables gives lower drag, the chord at the furthest inboard control point (design variable 1) takes a value in the middle of the bounding range. This suggests that further reducing the chord at this station results in higher drag. To confirm this, an additional design is analysed; *variant 2* in Figure D-1 is the same as the benchmark design, except that variable 1 takes the value of the lower problem bound. The FLUENT drag data for this modification are listed in Table D-1. As expected, the Euler (induced) drag for variant 2 is higher than for the unmodified benchmark design, resulting in an increase of 0.21 drag counts, or 0.15%. The RANS analysis also predicts higher drag for variant 2 compared to the benchmark design. The increase in induced drag is partially offset by a reduction in viscous drag, and hence the difference is less (0.12 counts), but this difference is clearly sufficient to drive the optimization away from this sub-optimal design.

Appendix E: Inverse Airfoil Design Code

Listed below is the MATLAB inverse design code used for inverse design of subsonic airfoils. The function additionally calls the external files *create_datX.m*, which generates the VGK *.DAT* (geometry) file, and *vgkgo_win.m* which executes the VGK solver for the calculation of surface pressure distributions.

```
function [x,zu,zl,alpha_in,I_vec]=Inverse_Airfoil(x,Xut,cput,Xlt,cplt)

% Inverse design for subsonic airfoils
% Xut, cput, Xlt, cplt, are the upper and lower surface data defining the
% target pressure distribution.
% x is the desired output distribution of points, zu and zl are the z
% ordinates of the output airfoil. alpha_in is the output angle of attack,
% i.e., the angle requried to achieve the target pressure profile. I_vec is
% the convergence history of the inverse process.
% Uses a residual correction method, where VGK is used to calculate the
% surface pressure distributions.

%%% fname is the filename for all VGK run files
fname='Foil2EMFIDv2';

x_original=x;

%%% VGK is sensitive to the input x vector, this one works well
xt=linspace(0,1,101);
xt(1)=(0.005*1);
xt=[0 (0.002*1) xt];
x=[0.2.*((exp(0.3.*linspace(0,10,100))-1)/(exp(3)-1)) xt(24:end)];

[Xut,cput]=interp(Xut,cput,x);
[Xlt,cplt]=interp(Xlt,cplt,x);

%%% As an initial design, start with the NASA LS(1)-0013 airfoil, with the same
resolution as new x
load NASA_0013_200.DAT
[a,b]=size(NASA_0013_200);
xn=NASA_0013_200(1:a/2,1)';
zu=NASA_0013_200(1:a/2,2)';
zl=NASA_0013_200((a/2)+1:end,2)';
[x,zu]=interp(xn,zu,x);
[x,zl]=interp(xn,zl,x);

%%% Constrain the TE point
zu2_end=0;
zl2_end=0;

%%% initial angle of attack
alpha_in=0;

Ib=10;
```

```
xb=x;
zub=zu;
zlb=zl;
alpha_inb=alpha_in;

I_vec=[];
for g = 1:60
       disp(['%%%%% ITERATION ' num2str(g) ' %%%%%'])

    %%% fac is used to accelerate the convergence, after 10 iterations the
    %%% size of the corrections is doubled
       fac=0.5;
       if g>10 fac=1; end
       [x,zu,zl,cperr,status,CL,CD,I,Udiff,Ldiff]=GetGeom(x,zu,zl,fname,Xut,cput,Xlt,c
plt,zu2_end,zl2_end,alpha_in,fac);
       I
       I_vec=[I_vec I];
       if I>Ib | status==0
              x=xb;
              zu=zub;
              zl=zlb;
              alpha_in=alpha_inb;
              break
       else
              xb=x;
              zub=zu;
              zlb=zl;
              Ib=I;
              alpha_inb=alpha_in;
       end

       if cperr==1 break; end

       %%% alpha correction, uses alpha relaxation factor of 0.12
       alpha_in=alpha_in-Ldiff*0.12;
       alpha_in=alpha_in+Udiff*0.12
end


%%% interpolate the result onto the desired x vector, and exit
xn=x;
[x,zu]=interp(xn,zu,x_original);
[x,zl]=interp(xn,zl,x_original);

return

function
[x,zu,zl,cperr,status,CL,CD,J,Udiff,Ldiff]=GetGeom(x,zu,zl,fname,Xut,cput,Xlt,cplt,zu2
_end,zl2_end,alpha_in,fac)
%%% RUNS VGK TO OBTAIN A CP PROFILE FOR THE CURRENT AIRFOIL

x_stag=0.015;

%%% create the VGK .DAT (geometry) file
create_datX(x,zu,zl,fname);

%%% Run VGK!
[CL,CD,status]=vgkgo_win(fname,alpha_in,0.15,4000000,0.075)
if status==0 disp('Cp profile gives invalid airfoil geometry'); x=x; zu=zu; zl=zl;
cperr=0; J=1; Ldiff=1; Udiff=1; return; end

[Xu,cpu,Xl,cpl]=getpress(fname);
```

```
if Xu(2)==0 Xu=[Xu(1) Xu(3:end)]; cpu=[cpu(1) cpu(3:end)]; end     % With a very fine
LE x vector, sometimes VGK gives X(2)=0,
if Xl(2)==0 Xl=[Xl(1) Xl(3:end)]; cpl=[cpl(1) cpl(3:end)]; end     % causing a
singularity.
[Xu,cpu]=interp(Xu,cpu,x);
[Xl,cpl]=interp(Xl,cpl,x);

%%% Check the convergence of the inverse design process
%% cperr=1 for a converged cp plot
cperr=0;
J=0;

%% for alpha correction
Udiff=sum(cpu(3:20)-cput(3:20));
Ldiff=sum(cpl(3:20)-cplt(3:20));

%% for Cp error measure (discard the stagnation region)
diff{1,1}=(cpu(x>x_stag)-cput(x>x_stag)).^2;
diff{2,1}=(cpl(x>x_stag)-cplt(x>x_stag)).^2;
diff{1,2}=x(x>x_stag);
diff{2,2}=x(x>x_stag);

for i = 1:2
    for j=2:length(diff{i,1})
        J = J+0.5*(diff{i,2}(j)-diff{i,2}(j-1))*(diff{i,1}(j)+diff{i,1}(j-1));
    end
end

%%% The convergence criterion here is 0.00005 - this is a very tight match
if J < 0.00005 cperr=1; return; end

%% NUMERICAL DIFFERENTIATION %%
[x,zud]=Diffate4(x,zu);
[x,zudd]=Diffate4(x,zud);
[x,zld]=Diffate4(x,zl);
[x,zldd]=Diffate4(x,zld);

%%% Upper Surface geometry correction %%%
Cpu=Cpcalc(zudd);
CpuO=Cpu;
[Cpu,I]=sort(real(Cpu));
zudd_=zudd(I);
[Cpu,dzdCp]=Diffate4(Cpu,zudd_);  % NUMERICAL DIFFERENTIATION USING FUNCTION
zudd2 = zudd + fac.*0.4.*dzdCp.*(cput-cpu);

%%% Lower Surface geometry correction %%%
Cpl=Cpcalc(zldd);
CplO=Cpl;
[Cpl,I]=sort(real(Cpl));
zldd_=zldd(I);
[Cpl,dzldCp]=Diffate4(Cpl,zldd_);% NUMERICAL DIFFERENTIATION USING FUNCTION
zldd2 = zldd - fac.*0.4.*dzldCp.*(cplt-cpl);



%% NUMERICAL INTEGRATION %%
[x,zud3]=integrate4(x,zudd2,0);
[x,zld3]=integrate4(x,zldd2,0);

[x,zu3]=integrate4(x,zud3,0);
[x,zl3]=integrate4(x,zld3,0);

%% calculate the constant of the first integration
```

```
const_u=zu2_end-zu3(end);
const_l=zl2_end-zl3(end);

zu = zu3 + const_u.*x;
zl = zl3 + const_l.*x;

%%% PLOTTING
% figure(11);
% plot(x,zu,'b',x,zl,'b')
% axis([0 1 -0.1 0.1]);
% figure(12);
% plot([0],[1.12471],'k+',[Xu Xl(end:-1:1)],[cpu cpl(end:-1:1)],'k--',[Xut Xlt(end:-
1:1)],[cput cplt(end:-1:1)],'k:')
% myax=axis;
% axis([myax(1:2) -1.5 1]);
% set(gca,'YDir','reverse')

return


function Cp=Cpcalc(zudd)
%%% Approximation of surface pressure distribution
gam=1.4;%ratio of specific heats
M=0.15;       %freestream mach
c=1;   %airfoil chord
Cs=(2/gam*(M^2))*(((   ((gam+1)/2)     /(1+((gam-1)/2)*(M^2)    ))^(gam/(gam-1)))-1);
Ms=1/(1-(((gam+1)/2)*Cs));

Cp=(-2/((Ms^2)*(gam+1))).*((1-(Ms^2))-((1-
(Ms^2))^(1.5)+0.75*(Ms^2)*(gam+1)*(c/pi).*zudd).^(2/3));
return

function [x,Yi]=integrate4(x,y,y0)
Yi(1)=y0;

for k=2:length(y)
       Yi(k)=Yi(k-1) + ((y(k-1)+y(k))/2).*(x(k)-x(k-1));     % trapezium rule
end
return

function [X,Yd]=Diffate4(x,y)
%% Forward/backward difference on first/last points
%% Use central difference with remaining points
%% Also uses functions dif and centdiff

Yd(1) = dif(x(1),y(1),x(2),y(2));

Yd(length(y)) = dif(x(length(y)-1),y(length(y)-1),x(length(y)),y(length(y)));

Yd=centdiff(y,x,Yd);

X=x(1:end);
Yd=Yd(1:end);
return

function yd=dif(x1,y1,x2,y2)
yd=(y2-y1)/(x2-x1);
return

function yd=centdiff(y,x,yd)
for k=2:length(y)-1
       yd(k)=(y(k+1)-y(k-1))/(x(k+1)-x(k-1));
end
```

```
return

function [X,Y]=interp(x,y,X)
%LINEAR interpolation
%for positive x. Assumes values in x increase along its length
xs=(1/10000).*round(x.*10000);
for i=1:length(X)
    if X(i)<xs(1)|X(i)-xs(length(x))>0.01 disp(i); disp(xs); disp(X);
error('interp');end
    for t = 1:length(x)-1
        if X(i)>=xs(t) & X(i)<=xs(t+1) break; end
    end
    Y(i)=((((y(t+1)-y(t))/(x(t+1)-x(t))))*(X(i)-x(t)))+y(t);
end
return

function [Xu,cpu,Xl,cpl]=getpress(fname)
% Get pressure distribution from VGK .BRF file

%check for the existence of the .BRF file
fid=fopen([fname '.BRF']);
if fid == -1 error(['Vgk output file ' fname '.BRF not found']), end
c=0;
%% Extract the data
while feof(fid) == 0
    myline=fgetl(fid);
    h = sscanf(myline,'%s');
    if strcmp(h,'XCP')==1

        while c>=0
            c=c+1;
            myline =fgetl(fid);
            if strcmp(myline,'')==1 break;end
            S=sscanf(myline,'%f');
            if isempty(sscanf(myline,'%f'))==1
                c=c-1;
                I=c;
                continue
            end
            X(c)=S(1);
            cp(c)=S(2);
        end
    end
end

Xl = X(I+1:length(X));
cpl = cp(I+1:length(cp));
Xu = X(1:I);
cpu = cp(1:I);
return
```

Appendix F: Problem Solving Environment Setup

This appendix provides further detailed information relating to the practical computational setup of the design search process described in this thesis. The information is presented in the form of flowchart diagrams and examples of MATLAB code. The setup described is for the 3-D EMFID design search process, but a very similar system is used for all optimization work.

The Optimization Strategy

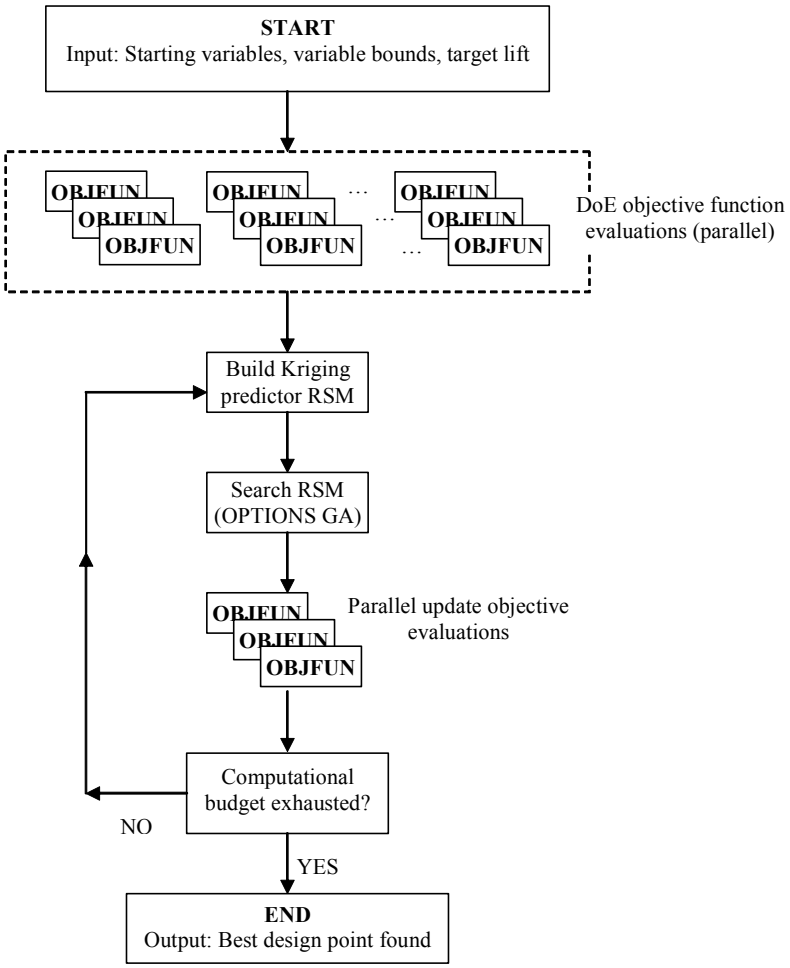


Figure F-1 The optimization strategy.

Figure F-1 shows the optimization strategy employed in the 3-D EMFID and benchmark design search methods. As stated in the thesis, this framework has been coded using the MATLAB language and operates under the Windows environment. All FLUENT simulations are run on the Microsoft compute cluster. The design search process is executed by running a MATLAB function, which specifies all of the starting parameters and which performs the actions shown in Figure F-1. The first step is to evaluate the objective function as dictated by a Latin hypercube DoE sample plan. These objective function calls are run in parallel, with typically five to 10 evaluations being run simultaneously. Each evaluation submits a new job to the compute cluster (see below). Note that the FLUENT jobs themselves are also run in parallel, in the sense that the calculation is performed using a number of processors (eight CPUs for the 3-D runs) utilizing MPI. Hence, with this doubly parallel capability, up to 80 CPUs are working simultaneously. After the DoE evaluations, the Kriging RSM is built and searched using OptionsMatlab[1] to predict update points, and the objective function is evaluated in parallel for these points. The process of building, searching and updating the RSM is repeated until the limit of the computational cost budget is reached.

The Objective Function

As stated above, the objective function is called for multiple design points simultaneously. This parallel objective function is implemented in MATLAB using three MATLAB functions. The first takes the input design variables and submits the job to the compute cluster, returning a unique job identifier (including the working directory for the job). The second uses this job identifier to *poll* the job; i.e., wait until the job has finished by checking for the existence of the output file. The third uses the job identifier to retrieve the job data from the output file and return the objective function value. The MATLAB syntax is therefore:

```
% Submit the parallel jobs
Job_ID_1=Objective_Fun(VARS1);
Job_ID_2=Objective_Fun(VARS2);
Job_ID_3=Objective_Fun(VARS3);
Job_ID_4=Objective_Fun(VARS4);
Job_ID_5=Objective_Fun(VARS5);
...

% Poll the jobs
Job_Poll(Job_ID_1)
Job_Poll(Job_ID_2)
Job_Poll(Job_ID_3)
Job_Poll(Job_ID_4)
```

[1] Part of the GEODISE toolkit, for which information can be obtained on the world wide web at http://www.geodise.org/ (cited April 2006).

```
Job_Poll(Job_ID_5)
...

% Parse the jobs to obtain the objective function value
obj1=Objective_Fun_Parse(Job_ID_1);
obj2=Objective_Fun_Parse(Job_ID_2);
obj3=Objective_Fun_Parse(Job_ID_3);
obj4=Objective_Fun_Parse(Job_ID_4);
obj5=Objective_Fun_Parse(Job_ID_5);
...
```

Listed below is the code for these three MATLAB functions, for the 3-D EMFID objective function.

```
function retrievalID = Fun_EMFIDWingTip(VARS)
% Objective function for 3-D EMFIDv9
% Input VARS, submits job to the cluster, output retID

global todaydate RUNNUMBER EMFIDNUMBER local_path

RUNNUMBER = RUNNUMBER + 1;        % RUNNUMBER is a unique number for each parallel run
EMFIDNUMBER = EMFIDNUMBER + 1;    % EMFIDNUMBER is used for linux running of VSAERO

VARS=VARS';

% Create the unique directory for running the job
RUNDIR = [local_path  todaydate '_' num2str(RUNNUMBER) '_' num2str(VARS(1))...
num2str(VARS(2)) num2str(VARS(3)) num2str(VARS(4)) '\'];
RUNDIR = strrep(RUNDIR,' ','');
RUNDIR = strrep(RUNDIR,'.','d');
dos(['mkdir ' RUNDIR]);
retrievalID.uniquedir = RUNDIR;

retrievalID.RUN = RUNNUMBER;
retrievalID.VARS = VARS';

% copy the required files to the new directory
make_spawn_EMFID('Spawn_EMFID_Base.m',RUNDIR);
dos(['move ' local_path 'EMFID3D\Spawn_EMFID.m ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\make_winglet_glyph_dir.m ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\Base_Wingletv4_Win.glf ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\make_winglet_log.m ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\Test_Log.log ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\Base_Log.log ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\Restart_Log.log ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\getlift.m ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\getdrag.m ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\GetJobID.m ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\GetJobStartTime.m ' RUNDIR]);
dos(['copy ' local_path 'EMFID3D\GetJobStartTime_10.m ' RUNDIR]);
save([RUNDIR 'VARS.mat'],'VARS','RUNNUMBER','EMFIDNUMBER','todaydate');

% Spawn a new MATLAB process for executing FLUENT
wd=cd;
cd(RUNDIR);
dos('matlab -nodesktop -nosplash -minimize -r Spawn_EMFID -logfile fluent_out.log &');
cd(wd);
return
```

```
function job_poll(retrievalID)
% wait for the existence of OUT_DATA.mat

f=exist([retrievalID.uniquedir 'OUT_DATA.mat']);
if f==1 return; end
pause(5);

return
```

```
function I = Fun_FluentWingTip_parse(retrievalID)
% Objective function _Parse for 3-D EMFIDv9
% retrieve objfun for job with retID retrievalID

global todaydate errorI local_path EMFIDNUMBER

EMFIDNUMBER=EMFIDNUMBER - 1;    % for emfid inverse design runs
RUN=retrievalID.RUN;
ERROR=0;

% Load output data file
local_path2 = retrievalID.uniquedir;
if exist([local_path2 'OUT_DATA.mat'])~=0
       results=load([local_path2 'OUT_DATA.mat']);
       CD=results.CD3
       CL=results.CL
else
       disp('Results file not found');
       ERROR=1;
end

% Calculate the objective function
if ERROR==1
       I=0;
else
       I=CL/CD;
end

return
```

As can be seen in the *Fun_EMFIDWingTip* function, the objective function calculations are performed in a new (spawned) MATLAB process for each parallel evaluation. The *.m* file script *Spawn_EMFID.m* is created containing the correct run directory, and this is run in the spawned MATLAB process; an example of this code is listed below.

```
% Spawn_EMFID.m
% Runs in spawned matlab process and submits inverse design and fluent jobs

RUNDIR='Z:\EMFID3D_11\110507EVI2_14_0d176880d149880d118150d077163\';
cd(RUNDIR);

% First, perform the inverse design (Linux)
load([RUNDIR 'VARS.mat']);          %loads VARS, RUNNUMBER, EMFIDNUMBER, todaydate

date=[todaydate '_' num2str(RUNNUMBER)];
EMFIDDIR=['V:\EMFID3D_' num2str(EMFIDNUMBER) '\vsaero_top\'];% V:=home1\utp-10\trb100


% Place the input data in the remote (Linux) directory
% A daemon running in Linux detects this and executes inverse design
if exist([RUNDIR 'InvDes_DATA.mat'])==0
    while 1
        if exist([EMFIDDIR 'OUT.mat'])==0
            save([EMFIDDIR 'IN_DATA.mat'],'VARS','date','-v4');
            while 1
                if exist([EMFIDDIR 'OUT.mat'])~=0 delete([EMFIDDIR 'OUT.mat']); break;
end
                pause(60);
            end
            break
        end
    end


% Once inverse design has finished
    n=dos(['move ' EMFIDDIR 'OUT_DATA.mat ' RUNDIR 'InvDes_DATA.mat']);
    if n==0 PASS=1; else PASS=0; disp('Problem in inverse design step'); end
else
    PASS=1;
end


if PASS==1
A=load([RUNDIR 'InvDes_DATA.mat']);
VARSg=A.VARSg   % VARSg = geometrical (i.e. benchmark) variables

OBJ=A.I;
disp(['Inverse design objective Function value ' num2str(OBJ)])

if OBJ>0.000001 save([RUNDIR 'OUT.mat']); exit; end  % if Inv Des not converged

% Now run fluent
targCL=0.4;
alpha1=3.06;  %%%% sensitive to these
alpha2=4;    %%%%
```

```
% Build the mesh using Gridgen, first generate the glyph script
RUNDIR2=strrep(RUNDIR,'\','/');
make_winglet_glyph_dir('Base_Wingletv4_Win.glf',VARSg,RUNDIR2);
while 1

    % Run gridgen
    [status,output]=dos('"C:\Program Files (x86)\Pointwise\GridgenV15\Win32\bin\ ...
       Gridgen" -b Work_Glyph.glf');

    if isempty(findstr('FLEXlm error',output))==1 break; end
end


% Test the mesh size by running a quick fluent
if exist([RUNDIR '3D_grid.cas'])>0
    make_winglet_log('Test_Log.log',alpha1,RUNDIR)

    [s,test]=dos(['C:\Fluent.Inc\ntbin\win64\fluent 3d -r6.3.19 -t4 -ccp ...
       msnode-001 -hidden -i ' RUNDIR 'Work_Log.log']);

    A=findstr(test,'hexahedral');
    B=findstr(test,'cells');
    if isempty(A)==0 & isempty(B)==0
        test=test(A-9:A-1);
            cells=str2num(test);
    else
            cells=1;
    end


% Run the two FLUENT evaluations
    if cells==1313520
        disp('mesh ok');
        make_winglet_log('Base_Log.log',alpha1,RUNDIR);

        dos(['C:\Fluent.Inc\ntbin\win64\fluent 3d -t8 -r6.3.19 -ccp msnode-001 ...
       -hidden -i ' RUNDIR 'Work_Log.log'])

        [CL1,ERROR1]=getlift(4000,RUNDIR)
        [CD1,ERROR2]=getdrag(4000,RUNDIR)
        make_winglet_log('Restart_Log.log',alpha2,RUNDIR);

        dos(['C:\Fluent.Inc\ntbin\win64\fluent 3d -t8 -r6.3.19 -ccp msnode-001 ...
       -hidden -i ' RUNDIR 'Work_Log.log'])

        [CL2,ERROR3]=getlift(7000,RUNDIR)
        [CD2,ERROR4]=getdrag(7000,RUNDIR)
        CD3=((CD2-CD1)/(CL2-CL1))*(targCL-CL1)+CD1

% Save the output file containing lift and drag, and exit
        if ERROR1 + ERROR2 + ERROR3 + ERROR4 ==0 save([RUNDIR ...
       'OUT_DATA.mat'],'CD3','targCL','time'); end
    end

    dos(['del ' RUNDIR '3D_grid.cas']);
end
end % end PASS if

exit
```

As seen above, FLUENT is executed using a call to DOS. The FLUENT version used for the work in this thesis is v6.3.19. While the FLUENT *host* process is run on an interactive machine, the *ccp* flag *-ccp msnode-001* is used to tell FLUENT to request the eight *node* processors from the compute cluster. MATLAB waits for this DOS command to finish before continuing the execution of the script.

The geometry and mesh generation software, GRIDGEN, is run on the local interactive machine and its operation not submitted as a job on the cluster. GRIDGEN version 15.08 is used for the work in the thesis. As seen above, the software is run using a call to DOS. The flab *-b* is used for batch execution, and all commands are stored in the specified *.glf* file, in this case *Work_Glyph.glf*. Using the commands in the *.glf* file, GRIDGEN takes the input design variables for the wing-tip device, generates the geometry and mesh for this design, and outputs the mesh (*.cas*) file. MATLAB waits for this DOS call to finish, i.e., it waits for GRIDGEN to finish exporting the mesh file, before continuing the execution of the *Spawn_EMFID.m* script.

The *.glf* file for the generation of the 3-D wing mesh is very long, and so it is not listed here. However, an example of the 2-D airfoil *.glf* file is given below. This is run in a very similar way, in the sense that GRIDGEN takes the input design parameters and outputs the *.cas* file.

```
# Gridgen Journal File V1 (Gridgen 15.08 REL 1)
# Created Tue May 30 15:11:52 2006 package require PWI_Glyph 1.6.8
gg::memClear
gg::aswDeleteBC -glob "*"
gg::aswDeleteVC -glob "*"
gg::aswSet "FLUENT" -dim 2
gg::defReset
gg::tolReset
# Delay screen updates and checking for user input until script is finished.
gg::updatePolicy DELAYED

# Import data files
gg::dbImport "Z:/EMFID2D_TT1/parallel1/Gridgen_Lower.dat" -type SEG
gg::dbImport "Z:/EMFID2D_TT1/parallel1/Gridgen_Upper.dat" -type SEG
set _DB(-1) [gg::dbGetByName -- {Gridgen_Upper-pcurve-1}]
set _DB(-2) [gg::dbGetByName -- {Gridgen_Lower-pcurve-1}]

# Fit a curve to LE points, using data file co-ords
gg::dbCurveBegin -type CUBIC
  gg::dbCurveAddPt [list 0.005137 0.012663 0]
  gg::dbCurveAddPt [list 0.0035 0.0095058 0]
  gg::dbCurveAddPt [list 0.002179 0.0063424 0]
  gg::dbCurveAddPt [list 0.001188 0.0035921 0]
  gg::dbCurveAddPt [list 0.000511 0.0015691 0]
  gg::dbCurveAddPt [list 0 0 0]
  gg::dbCurveAddPt [list 0.000511 -0.0016107 0]
  gg::dbCurveAddPt [list 0.001188 -0.0036874 0]
  gg::dbCurveAddPt [list 0.002179 -0.0065229 0]
  gg::dbCurveAddPt [list 0.0035 -0.0097895 0]
  gg::dbCurveAddPt [list 0.005137 -0.012955 0]
set _DB(3) [gg::dbCurveEnd]
```

```
set _ggTemp_(1) [ggu::utilMergeDefaults [list 0 ] ""]
unset _ggTemp_(1)
set _ggTemp_(2) [list 0.005137 0.012663 0]
set _ggTemp_(3) [gg::dbSplit $_DB(-1) -u $_ggTemp_(2)]
set _DB(4) [lindex $_ggTemp_(3) 0]
unset _ggTemp_(3)
unset _ggTemp_(2)
set _ggTemp_(4) [list 0.005137 -0.012955 0]
set _ggTemp_(5) [gg::dbSplit $_DB(-2) -u $_ggTemp_(4)]
set _DB(5) [lindex $_ggTemp_(5) 0]
unset _ggTemp_(5)
unset _ggTemp_(4)

gg::conOnDBEnt [list \
    $_DB(3) \
    $_DB(4) \
    $_DB(5) \
  ]
gg::dbEnable $_DB(-2) FALSE
gg::dbEnable $_DB(-1) FALSE
gg::dbEnable $_DB(3) FALSE
gg::dbEnable $_DB(4) FALSE
gg::dbEnable $_DB(5) FALSE
set _CN(3) [lindex [gg::conGetAll] 2]
set _CN(1) [lindex [gg::conGetAll] 0]
set _CN(3) [gg::conJoin $_CN(3) $_CN(1)]
set _CN(2) [lindex [gg::conGetAll] 0]
set _CN(3) [gg::conJoin $_CN(3) $_CN(2)]
set _CN(4) [gg::conSplit $_CN(3) [gg::conGetPt $_CN(3) -arc 0.5]]

# No. points on each surface
gg::conDim $_CN(3) 200

# Upper LE spacing
gg::conBeginSpacing $_CN(3) -sub 1 0.001
gg::conEndSpacing $_CN(3) -sub 1 0.003
gg::conDim $_CN(4) -dimension $_CN(3)
gg::conBeginSpacing $_CN(4) -sub 1 0.003
gg::conEndSpacing $_CN(4) -sub 1 0.001

set _CN(4) [gg::conJoin $_CN(4) $_CN(3)]

gg::domExtrusionBegin $_CN(4) -edge  -default HYPERBOLIC
  gg::domExtrusionAtt -local 1 -flip
  gg::domExtrusionAtt -local 1 -s_init 0.00045
  gg::domExtrusionAtt -local 1 -growth_geometric 1.1
  gg::domExtrusionAtt -stop_height 15
  gg::domExtrusionStep 1
  gg::domExtrusionAtt -local 1 -growth_geometric 1.2
  gg::domExtrusionStep 200
set _ggTemp_(1) [gg::domExtrusionEnd]
set _DM(1) [lindex $_ggTemp_(1) 0]
unset _ggTemp_(1)

# Enforce **FIRST_CELL on TE
set _CN(5) [lindex [gg::conGetAll] 1]
gg::conBeginSpacing $_CN(5) -sub 1 0.00045

# Run the pde solver on the domain to remove any negative volumes
set _ggTemp_(1) [list $_DM(1)]
gg::domEllSolverBegin $_ggTemp_(1)
  gg::domEllSolverAtt $_DM(1) -edge [list 1] -spacing_calc 0.00045
  gg::domEllSolverStep -iterations 1 -nodisplay
```

```
gg::domEllSolverEnd
unset _ggTemp_(1)

gg::blkBegin -type STRUCTURED
  gg::faceBegin
    gg::faceAddDom $_DM(1)
  gg::faceEnd
set _BL(1) [gg::blkEnd]
gg::aswSetBC [list \
    $_CN(4) \
    ] \
  "Wall"
set _CN(6) [lindex [gg::conGetAll] 2]
gg::aswSetBC [list \
    $_CN(6) \
    ] \
  "Pressure Far Field"

gg::aswExport "Z:/EMFID2D_TT1/parallel1/Airfoil2D.cas" -merge_blocks
```

References

Ahn, T., Kim, H. J., and Rho, O. H., "Inverse Design of Transonic Wings Using Wing Planform and Target Pressure Optimization," *Journal of Aircraft*, Vol. 38, No. 4, July-August, 2001, pp. 644-652.

Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., "Optimization with Variable-Fidelity Models Applied to Wing Design," *38th Aerospace Sciences Meeting and Exhibit*, AIAA 2000-0841, January 2000.

Anderson, J. D., *Fundamentals of Aerodynamics*, Second edition, McGraw-Hill, 1991.

Barrett, T. R., Bressloff, N. W., and Keane, A. J., "Airfoil Shape Design and Optimization Using Multifidelity Analysis and Embedded Inverse Design," *AIAA Journal*, Vol. 44, No. 9, September, 2006a.

Barrett, T. R., Bressloff, N. W., and Keane, A. J., "Airfoil Design and Optimization Using Multi-Fidelity Analysis and Embedded Inverse Design," *2nd AIAA Multidisciplinary Design Optimization Specialist Conference*, Newport, R.I., 1-4 May 2006b.

Barrett, T. R., Bressloff, N. W., and Keane, A. J., "Aerodynamic Design Optimization Using Flow Feature Parameterization," *3rd International Conference on Flow Dynamics*, Matsushima, Japan, 7-9 November 2006c.

Bloor, M. I. G. and Wilson, M. J., "Efficient Parameterization of Generic Aircraft Geometry," *Journal of Aircraft*, Vol. 32, No. 6, November-December, 1995, pp. 1269-1275.

Bui-Thanh, T., Damodaran, M., and Willcox, K., "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 42, No. 8, August, 2004, pp. 1505-1516.

Burkardt, J., Qiang, D., Gunzburger, M., and Lee, H. C., "Reduced Order Modelling of Complex Systems," *20th Biennial Conference on Numerical Analysis*, University of Dundee, 24-27 June 2003, pp. 29-38.

Burkett, C. W., "Reductions in Induced Drag By the Use of Aft Swept Wing Tips," *Aeronautical Journal*, Vol. 93, No. 930, December, 1989, pp. 400-405.

Chandrasekharan, R. M., Murphy, W. R., Taverna, F. P., and Boppe, C. W., "Computational Aerodynamic Design of the Gulfstream IV Wing," *23rd AIAA Aerospace Sciences Meeting*, Reno, AIAA-85-0427, 1985, pp. 1-6.

Cohn, R. K. and Koochesfahani, M. M., "The Accuracy of Remapping Irregularly Spaced Velocity Data Onto a Regular Grid and the Computation of Vorticity," *Experiments in Fluids*, Vol. 29, No. 7, December, 2000, pp. 61-69.

Cook, P. H., McDonald, M. A., and Firmin, M. C. P., "Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements," Experimental Data Base for Computer Program Assessment, AGARD Report, AR 138, 1979.

Davis, W. H., "Technique for Developing Design Tools from the Analysis Methods of Computational Aerodynamics," *AIAA Journal*, Vol. 18, No. 9, September, 1980, pp. 1080-1087.

de Boor, C., "On uniform Approximation by Splines," *Journal of Approximation Theory*, Vol. 1, 1968, pp. 219-235.

de Boor, C., "On Calculating with B-Splines," *Journal of Approximation Theory*, Vol. 6, 1972, pp. 50-62.

Drela, M., "Elements of Airfoil Design Methodology," in *Applied Computational Aerodynamics*, AIAA Progress in Astronautics and Aeronautics, vol. 125, 1989, pp. 167-189.

Dulikravich, G. S., "Aerodynamic Shape Design," *Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications*, AGARD 780, November 1990.

Elizarov, A. M., Il'inskiy, N. B., and Potashev, A. V., *Mathematical Methods of Airfoil Design: Inverse Boundary-Value Problems of Aerohydrodynamics*, John Wiley & Sons Inc., 1997.

Eppler, R., "Induced Drag and Winglets," *Aerospace Science and Technology*, Vol. 1, No. 1, 1997, pp. 3-15.

ESDU, "The VGK Method for Two-Dimensional Aerofoil Sections," ESDU 96028, 1996.

Faye, R., Laprete, R., and Winter, M., "Blended Winglets for Improved Airplane Performance," *Boeing Aero*, Vol. 17, January, 2002, pp. 16-31.

Ferris, J. C., McGhee, R. J., and Barnwell, R. W., "Low-Speed Wind-Tunnel Results for Symmetrical NASA LS(1)-0013 Airfoil," NASA Technical Memorandum 4003, 1987.

Fluent, "FLUENT 5 User's Guide," Lebanon, 2003.

Frink, N. T., "Assessment of an Unstructured-Grid Method for Predicting 3-D Turbulent Viscous Flows," *34th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, AIAA 96-0292, 1996.

Gerontakos, P. and Lee, T., "Near-field Tip Vortex Behind a Swept Wing Model," *Experiments in Fluids*, Vol. 40, 2006a, pp. 141-155.

Gerontakos, P. and Lee, T., "Effects of Winglet Dihedral on a Tip Vortex," *Journal of Aircraft*, Vol. 43, No. 1, January-February, 2006b, pp. 117-124.

Giles, M. B. and Drela, M., "Two-Dimensional Transonic Aerodynamic Design Method," *AIAA Journal*, Vol. 25, No. 9, September, 1987, pp. 1199-1206.

Giles, M. B. and Cummings, R. M., "Wake Integration for Three-Dimensional Flowfield Computations: Theoretical Development," *Journal of Aircraft*, Vol. 36, No. 2, March-April, 1999, pp. 357-365.

Gopalarathnam, A. and Selig, M. S., "Hybrid Inverse Airfoil Design Method for Complex Three-dimensional Lifting Surfaces," *Journal of Aircraft*, Vol. 39, No. 3, May-June, 2002, pp. 409-417.

Goto, A. and Zangeneh, M., "Hydrodynamic Design of Pump Diffuser Using Inverse Design Method and CFD," *Journal of Fluids Engineering*, Vol. 124, June, 2002, pp. 319-328.

Grove, D. M. and Davis, T. P., *Engineering, Quality and Experimental Design*, 1st edition, Longman, 1992.

Haimes, R. and Kenwright, D., "On the Velocity Gradient Tensor and Fluid Feature Extraction," *14th Computational Fluid Dynamcs Conference*, Norfolk, AIAA 1999-3288, 1999.

Harris, C. D., "NASA Supercritical Airfoils, A Matrix of Family-Related Airfoils," NASA Technical Paper 2969, March, 1990.

Heffley, R. K. and Jewell, W. F., "Aircraft Handling Qualities Data," NASA Technical Report, NASA-CR-2144, December, 1972.

Hicks, R. M. and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, July, 1978, pp. 407-414.

Hoyle, N., "Automated Multi-Stage Geometry Parameterization of Internal Fluid Flow Applications," PhD thesis, School of Engineering Sciences, University of Southampton, Southampton, U.K., 2006.

Inger, G. R., "Application of Oswatitsch's Theorem to Supercritical Airfoil Drag Calculation," *Journal of Aircraft*, Vol. 30, No. 3, May-June, 1993, pp. 415-416.

Jacobs, E. N., Ward, K. E., and Pinkerton, R. M., "The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel," N.A.C.A. T. R. No. 460, 1933.

Jameson, A., "Aerodynamic Design Via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233-260.

Jameson, A., "Full-Potential, Euler, and Navier-Stokes Schemes," in *Applied Computational Aerodynamics*, AIAA Progress in Astronautics and Aeronautics, vol. 125, 1989, pp. 39-91.

Jameson, A., "The Present Status, Challenges, and Future Developments in Computational Fluid Dynamics," *Progress and Challenges in CFD Methods and Algorithms*, AGARD CP-578, 1996, pp. 1-38.

Jameson, A., "Re-Engineering the Design Process Through Computation," *Journal of Aircraft*, Vol. 36, No. 1, January-February, 1999, pp. 36-50.

Jeong, J. and Hussain, F., "On the Identification of a Vortex," *Journal of Fluid Mechanics*, Vol. 285, 1995, pp. 69-94.

Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, December, 1998, pp. 455-492.

Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, December, 2001, pp. 345-383.

Jupp, J., "Wing Aerodynamics and the Science of Compromise," *Aeronautical Journal*, Vol. 105, No. 1053, November, 2001, pp. 633-641.

Keane, A. J., "Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods," *Journal of Aircraft*, Vol. 40, No. 4, July-August, 2003, pp. 742-750.

Keane, A. J. and Nair, P. B., *Computational Approaches to Aerospace Design: the Pursuit of Excellence*, first edition, John Wiley & Sons, U.K., 2005.

Kenwright, D. N., Henze, C., and Levit, C., "Feature Extraction of Separation and Attachment lines," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 2, April-June, 1999, pp. 135-144.

Kim, H. J. and Rho, O. H., "Aerodynamic Design of Transonic Wings Using the Target Pressure Optimization Approach," *Journal of Aircraft*, Vol. 35, No. 5, 1998, pp. 671-677.

Kroo, I., "Drag Due to Lift: Concepts for Predication and Reduction," *Annual Review of Fluid Mechanics*, Vol. 33, 2001, pp. 587-617.

Kroo, I., "Nonplanar Wing Concepts for Increased Aircraft Efficiency," *VKI Lecture Series on Innovative Configurations and Advanced Concepts for Future Civil Aircraft*, Stanford, ISBN 2-930389-62-1, June 6-10 2005.

La Roche, U. and Palffy, S., "WING-GRID, a Novel Device for Reduction of Induced Drag on Wings," *20th ICAS Congress*, Naples, Italy, 1998, pp. 2303-2309.

Labrujere, T. E. and Slooff, J. W., "Computational Methods for the Aerodynamic Design of Aircraft Components," *Annual Review of Fluid Mechanics*, Vol. 25, 1993, pp. 183-214.

Labrujere, T. E., "Residual-Correction Type and Related Computational Methods for Aerodynamic Design Part 1 Airfoil and Wing Design," *Optimum Design Methods for Aerodynamics*, AGARD R-803, April 1994.

LeGresley, P. A. and Alonso, J. J., "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition," *Proceedings of Fluids 2000 Conference and Exhibit*, AIAA 2000-2545, June 2000.

LeGresley, P. A. and Alonso, J. J., "Dynamic Domain Decomposition and Error Correction for Reduced Order Models," *41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, AIAA 2003-0250, 6-9 January 2003.

Lépine, J., Guilbault, F., and Trépanier, J. Y., "Optimized Nonuniform Rational B-Spline Geometrical Representation for Aerodynamic Design of Wings," *AIAA Journal*, Vol. 39, No. 11, November, 2001, pp. 2033-2041.

Letcher, J. S. J., "Convergence of Lift and Drag Predictions by a Morino Panel Method (VSAERO)," *AIAA Journal*, Vol. 27, No. 8, August, 1989, pp. 1019-1020.

Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S., and Hemsch, M. J., "Data Summary from the First AIAA Computational Fluid Dynamics Drag Prediction Workshop," *AIAA Journal*, Vol. 40, No. 5, September-October, 2003, pp. 875-882.

Li, W., Krist, S., and Campbell, R., "Transonic Airfoil Shape Optimization in Preliminary Design Environment," *Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004, pp. 3650-3671.

Lighthill, M. J., "A New Method of Two-Dimensional Aerodynamic Design," Aeronautical Research Council, London, R & M Rept. 2112, June, 1945.

Liu, T., Kuykendoll, K., Rhew, R., and Jones, S., "Avian Wing Geometry and Kinematics," *AIAA Journal*, Vol. 44, No. 5, May, 2006, pp. 954-963.

Lock, R. C., "The Prediction of the Drag of Aerofoils and Wings at High Subsonic Speeds," *Aeronautical Journal*, Vol. 90, No. 896, June-July, 1986, pp. 207-226.

Maskew, B., "Prediction of Subsonic Aerodynamic Characteristics: A Case for Low-Order Panel Methods," *Journal of Aircraft*, Vol. 19, No. 2, February, 1982, pp. 157-163.

McGhee, R. J., Beasley, W. D., and Whitcomb, R. T., "NASA Low- and Medium-Speed Airfoil Development," NASA Technical Memorandum 78709, 1979.

Milholen, W. E., "Efficient Inverse Aerodynamic Design Method for Subsonic Flows," *Journal of Aircraft*, Vol. 38, No. 5, September-October, 2001, pp. 918-923.

Murayama, M., Nakahashi, K., and Sawada, K., "Simulation of Vortex Breakdown Using Adaptive Grid Refinement with Vortex-Center Identification," *AIAA Journal*, Vol. 39, No. 7, July, 2001, pp. 1305-1312.

Nathman, J., "VSAERO User's Manual, Version 7.0 " Analytical Methods Inc., Redmond, Washington, May, 2003.

Nielson, E. J. and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, June, 2002, pp. 1155-1163.

Obayashi, S. and Takanashi, S., "Genetic Optimization of Target Pressure Distributions for Inverse Design Methods," *AIAA Journal*, Vol. 34, No. 5, May, 1996, pp. 881-886.

Painchaud-Ouellet, S., Tribes, C., Trépanier, J. Y., and Pelletier, D., "Airfoil Shape Optimization Using a Nonuniform Rational B-Splines Parameterization Under Thickness Constraint," *AIAA Journal*, Vol. 44, No. 10, October, 2006, pp. 2170-2178.

Pemberton, R. J., "A Vortex Identification Technique for Grid Adaption," PhD thesis, School of Engineering Sciences, University of Southampton, Southampton, U.K., 2003.

Petruzzelli, N. and Keane, A. J., "Wave Drag Estimation for use with Panel Codes," *Journal of Aircraft*, Vol. 38, No. 4, 2001, pp. 778-782.

Phillips, W. F., Fugal, S. R., and Spall, R. E., "Minimizing Induced Drag with Wing Twist, Computational-Fluid-Dynamics Validation," *Journal of Aircraft*, Vol. 43, No. 2, March-April, 2006, pp. 437-444.

Piegl, L. and Tiller, W., *The NURBS Book*, 2nd edition, Springer-Verlag, 1997.

Qin, N., Vavalle, A., and Le Moigne, A., "Spanwise Lift Distribution for a Blended Wing body Aircraft," *Journal of Aircraft*, Vol. 42, No. 2, March-April, 2005, pp. 356-365.

Redeker, G., "DLR-F4 Wing Body Configuration," A Selection of Experimental Test Cases for the Validation of CFD Codes, AGARD Report, AR-303, Vol. 2, 1994.

Reneaux, J., "Overview on Drag Reduction Technologies for Civil Transport Aircraft," *Euopean Congress on Computational Methods in Applied Science and Engineering*, Jyväskylä, 24-28 July 2004.

Robinson, G. M. and Keane, A. J., "A Case for Multi-Level Optimization in Aeronautical Design," *Aeronautical Journal*, Vol. 103, No. 1028, 1999, pp. 481-485.

Robinson, G. M. and Keane, A. J., "Concise Orthogonal Representation of Supercritical Airfoils," *Journal of Aircraft*, Vol. 38, No. 3, 2001, pp. 580-583.

Rokhsaz, K., "Effect of Viscous Drag on Optimum Spanwise Lift Distribution," *Journal of Aircraft*, Vol. 30, No. 1, January-February, 1993, pp. 152-154.

Rossow, V. J., "Classical Wing Theory and the Downward Velocity of Vortex Wakes," *Journal of Aircraft*, Vol. 43, No. 2, March-April, 2006, pp. 381-385.

Samareh, J. A., "A Survey of Shape Parameterization Techniques," *CEAS/AIAA/CASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, CP-209136, June 1999, pp. 333-343.

Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers," Experimental Data Base for Computer Program Assessment, AGARD AR-138, May, 1979.

Shih, T., Liou, W. W., Shabbir, A., Yang, Z., and Zhu, J., "A New k-ε Eddy Viscosity Model for High Reynolds number Turbulent Flows," *Computers Fluids*, Vol. 24, No. 3, 1995, pp. 227-238.

Smith, M. J., Komerath, N., Ames, R., Wong, O., and Pearson, J., "Performance Analysis of a Wing with Multiple Winglets," AIAA paper 2001-2407, 2001.

Smith, R. E., Bloor, M. I. G., Wilson, M. J., and Thomas, A. M., "Rapid Airplane Parametric Input Design (RAPID)," *12th AIAA Computational Fluid Dynamics Conference*, Washington DC, AIAA-1995-1687, 1995.

Sóbester, A., Leary, S. J., and Keane, A. J., "A Parallel Updating Scheme for Approximating and Optimizing High Fidelity Computer Simulations," *Structural and Multidisciplinary Optimization*, Vol. 27, No. 5, July, 2004, pp. 371-383.

Sobieczky, H. and Seebass, A. R., "Supercritical Airfoil and Wing Design," *Annual Review of Fluid Mechanics*, Vol. 16, 1984, pp. 337-363.

Sobieczky, H., "Progress in Inverse Design and Optimization in Aerodynamics," *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD CP-463, March 1990.

Song, W. and Keane, A. J., "A Study of Shape Parameterisation Methods for Airfoil Optimisation," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2004-4482, August 2004.

Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *30th Aerospace Sciences Meeting and Exhibit*, Reno, AIAA 1992-0439, 1992.

Spalart, P. R., "Airplane Trailing Vortices," *Annual Review of Fluid Mechanics*, Vol. 30, 1998, pp. 107-138.

Spillman, J. J., "Wing Tip Sails; Progress to Date and Future Developments," *Aeronautical Journal*, Vol. 91, No. 910, December, 1987, pp. 445-453.

Steinbrenner, J. P. and Anderson, D. A., "Grid-Generation Methodology in Applied Aerodynamics," in *Applied Computational Aerodynamics*, AIAA Progress in Astronautics and Aeronautics, vol. 125, 1989, pp. 91-130.

Takanashi, S., "Iterative Three-Dimensional Transonic Wing Design Using Integral Equations," *Journal of Aircraft*, Vol. 22, No. 8, August, 1985, pp. 655-660.

Trapp, J. C. and Sobieczky, H., "Interactive Parametric geometry Design," *37th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, AIAA 99-0829, January 1999.

van Dam, C. P., "Induced-Drag Characteristics of Crescent-Moon-Shaped Wings," *Journal of Aircraft*, Vol. 24, No. 2, February, 1987, pp. 115-119.

van Dam, C. P., Vijgen, P. M. H. W., and Holmes, B. J., "Experimental Investigation on the Effect of Crescent Planform on Lift and Drag," *Journal of Aircraft*, Vol. 28, No. 11, November, 1991, pp. 713-720.

van Dam, C. P., "Recent Experience with Different Methods of Drag Prediction," *Progress in Aerospace Sciences*, Vol. 35, 1999, pp. 751-798.

van Egmond, J. A., "Numerical Optimization of Target Pressure Distributions for Subsonic and Transonic Airfoils Design," *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD 463, March 1990.

Vassberg, J. C., Sclafani, A. J., and DeHaan, M. A., "A Wing-Body Fairing Design for the DLR-F6 Model: A DPW-III Case Study," *23rd AIAA Applied Aerodynamics Conference*, Toronto, AIAA 2005-4730, June 2005.

Volpe, G., "Inverse Airfoil Design : A Classical Approach Updated for Transonic Applications," in *Applied Computational Aerodynamics*, AIAA Progress in Astronautics and Aeronautics, vol. 125, 1989, pp. 191-220.

Xing, X. Q. and Damodaran, M., "Inverse Design of Transonic Airfoils Using Parallel Simultaneous Perturbation Stochastic Approximation," *Journal of Aircraft*, Vol. 42, No. 2, March-April, 2005, pp. 568-570.

Yamashita, H., Yonezawa, M., Obayashi, S., and Kusunose, K., "A Study of Busemann-type Biplane for Avoiding Choked Flow," *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, AIAA-2007-0688, 8-11 January 2007.

Zerihan, J. and Zhang, X., "Aerodynamics of a Single Element Wing in Ground Effect," *Journal of Aircraft*, Vol. 37, No. 6, November-December, 2000, pp. 1058-1064.

Zhang, H. J., Zhou, Y., and Whitelaw, J. H., "Near-Field Wing-Tip Vortices and Exponential Vortex Solution," *Journal of Aircraft*, Vol. 43, No. 2, March-April, 2006, pp. 445-449.