USING SURFACE SENSITIVITY FOR ADJOINT AERODYNAMIC OPTIMISATION OF SHOCK CONTROL BUMPS

BENJAMIN LEE HINCHLIFFE

A thesis submitted in partial fulfilment of the
requirement for the degree of Doctor of Philosophy

Department of Mechanical Engineering
The University of Sheffield
Supervised by Professor Ning Qin

September 2016

I would like to dedicate this thesis to my parents, Mike and Tracey, my wife, Sophie,
and my daughters, Evie and Holly. Thank you for all your support.

ABSTRACT

The purpose of this research is to use the surface sensitivity to aid the design and placement of flow control devices and to develop a new and efficient method of calculating the surface sensitivity using the mesh adjoint equations. The mesh adjoint equation provides a simplification of the adjoint optimisation framework which can speed up an optimisation by removing the bottleneck of needing to calculate the mesh sensitivity.

The surface sensitivity can be used as a design tool a designer to the most important regions on an aircraft surface. This thesis focusses on using shock control bumps and surface contour bumps in drag sensitive regions on transonic aerofoils and wings to reduce drag. Usually a designer has the surface pressure and streamlines to guide the device placement, however these can mislead as it is not clear which areas will have the most impact on drag reduction. The drag surface sensitivity gives a direct link between the drag coefficient and a potential change in the wing surface in the form of a derivative. This method was proved successful for reducing drag when optimisation was localised to the drag sensitive regions on the wing.

A new method for calculating the surface sensitivity using the Delaunay Graph Mapping (DGM) mesh movement has been developed. This provides an explicit and efficient mapping of the mesh sensitivity to the surface senstivity. Previously, this required the solution of a large and costly linear system using a mesh movement such as Linear Elasticity (LE) to move the mesh. The DGM method is compared against analytical solutions, finite difference and the LE mesh adjoint to show that the DGM mesh adjoint will provide an accurate calculation of the gradients on the wing surface. The DGM mesh adjoint has been shown to successfully find a minima when optimising shock bumps on a 3D geometry showing that it is a robust and capable method for optimisation.

PUBLICATIONS

Journal Articles

Benjamin L. Hinchliffe and Ning Qin. "Using surface sensitivity from mesh adjoint for transonic wing drag reduction", AIAA Journal, (awaiting publication)

Gabriele L. Mura, **Benjamin L. Hinchliffe**, Ning Qin, and Joel Brezillon. "Efficient Method to Eliminate Mesh Sensitivity in Adjoint Based Optimization", AIAA Journal (Undergoing review)

Conference Papers

Benjamin L. Hinchliffe and Ning Qin. "Using surface sensitivity from mesh adjoint solution for transonic wing drag reduction", 54th AIAA Aerospace Sciences Meeting, AIAA SciTech, AIAA paper 2016-0560
http://dx.doi.org/10.2514/6.2016- 0560

Gabriele L. Mura, **Benjamin L. Hinchliffe**, Ning Qin, and Joel Brezillon. "Using A Fast and Explicit Mesh Movement Method To Efficiently Compute Mesh Sensitivity", 54th AIAA Aerospace Sciences Meeting, AIAA SciTech, AIAA paper 2016-0291
http://dx.doi.org/10.2514/6.2016-0291

Feng Zhu, **Benjamin Hinchliffe** and Ning Qin. "Using Mesh Adjoint for Shock Bump Deployment and Optimisation on Transonic Wings", 53rd AIAA Aerospace Sciences Meeting, AIAA SciTech, AIAA paper 2015-1488,
http://dx.doi.org/10.2514/6.2015-1488

Benjamin Lee Hinchliffe and Ning Qin. “Robust 3D Shock Bump Design", Advanced Aero Concepts, Design and Operations, RAeS Applied Aerodynamics Conference 2014.

ACKNOWLEDGEMENTS

Firstly, I would like to thank Professor Ning Qin for his guidance and support throughout my PhD. The discussions I have had relating to CFD and related topics have given me a good knowledge of the field.

I would like to thank Feng Zhu and Gabriele Luigi Mura for the collaborations and the helpful discussions throughout my research. Thanks are also necessary for the aerodynamic CFD group under Ning Qin.

The work was supported by the ESPRC/Airbus ICASE award. I would like to thank Airbus for their help throughout this investigation.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Acronyms

Symbol	**Description**
CFD	Computational Fluid Dynamics
CFRP	Carbon Fibre Reinforced Plastic
CG	Conjugate Gradient
CST	Class Shape function Transforms
DBD	Dielectric Barrier Discharge
DGM	Delaunay Graph Mapping
DNS	Direct Numerical Simulation
FD	Finite Difference
GA	Genetic Algorithm
LE	Linear Elasticity
NLF	Natural Laminar Flow
NS-DBD	Nano Second-Dielectric Barrier Discharge
RANS	Reynolds Averaged Navier-Stokes
SCB	Shock Control Bump
T-S	Tollmien-Schlichting
VG	Vortex Generator

Greek Symbols

Symbol	**Description**
α	Angle of attack
α_i	DGM sub-areas
α_k	Optimisation step size
β_k	Conjugate gradient improvement factor
δ_{ij}	Dirac delta function

$\boldsymbol{\epsilon}$	Strain tensor
ϵ	Finite difference step size
η	Non-dimensional spanwise coordinate
γ	Ratio of specific heats
κ_t	Thermal conductivity
$\boldsymbol{\lambda}$	Lagrangian multiplier
$\boldsymbol{\lambda_{flow}}$	Vector of flow adjoint variables
$\boldsymbol{\lambda_{mesh}}$	Vector of mesh adjoint variables
λ, μ	Lamé constants
μ	Dynamic viscosity
ν	Poissons ratio
Ω	Vorticity
ψ	Non-dimensional chordwise coordinate
ρ	fluid density
$\boldsymbol{\sigma}$	Stress tensor
σC_D	Standard deviation of drag coefficient
τ_{ij}	Shear tensor for the ij plane
$\xi(\psi)$	Non-dimensional coordinate normal to chordwise direction

Latin Symbols

Symbol	**Description**
A_i, $B_{i,j}$	CST Bernstein polynomial coefficients
$\mathbf{B}$	Nodes used for Deluanay graph vertices
$\mathbf{B}_k$	Hessian like matrix for Quasi-Newton methods
$\%bl$	Percentage of bump length
$\%c$	Percentage chord
c	Chord
$\overline{C}_D$	Average of drag coefficient
C_D	Drag coefficient
C_j	j^{th} otimisation constraint

C_l or C_L	Coefficient of lift
C_{my}	Coefficient of moment
C^n	n^{th} order continuity
$C_{N_2}^{N_1}(\psi)$	CST class function
C_p	Specicifc heat at constant pressure
C_v	Specific heat at constant volume
$\mathbf{D}$	Vector of deign variables
D_i	i^{th} design variable
d_k	Optimisation step size scaling factor
$\mathbf{E}$	DGM mesh deformation matrix
E	Total energy
E	Youngs modulus
e_i	Volume ratios for DGM
$\mathbf{F}^c$, $\mathbf{F}^v$	Flux vector for invsicid and vicous terms respectively
H	Total enthalpy
$H(\eta)$	CST height function
$\mathbf{I}$	Identity matrix
I	Objective function
$\mathbf{K}$	Linear elasticity mesh deformation matrix
k	Turbulence kinetic intensity
$K_{r,n}$	Binomial coefficient
$\mathcal{L}$	Lagrangian function
M	Mach number
N_1, N_2	CST class function exponents
$N_\mathbf{B}$	Number of nodes used for the Delaunay vertices
N_{DV}	Number of desgin variables
$N_\mathbf{X}$	Number of computational mesh nodes
p	Static pressure
Pr	Prandtl number

q	Heat flux vector
$\mathbf{R}(\mathbf{W})$	Flow residual vector
R	Gas constant
Re	Reynolds number
S	Fluid volume surface
S	Magnitude of vorticity
$S(\psi)$	Shape functions for CST
$Sx_i(\psi)$	Shape functions for 3D CST in x-direction
$Sy_j(\psi)$	Shape functions for 3D CST in x-direction
$\mathbf{T}$	Mesh deformation residual
T	Temperature
u	Velocity component in the x-direction
V	Fluid volume
ν	Molecular kinematic viscosity
v	Velocity component in the y-direction
V_i	DGM sub-volumes
$\mathbf{W}$	Vector of conservative variables
w	Velocity component in the z-direction
$\mathbf{X}$	Computational domain

Part I

THESIS MOTIVATION

1

INTRODUCTION

1.1 BACKGROUND

The Global Market Forecast [1] shows that aircraft use has previously doubled approximately every 15 years. If this trend continues then by 2031 demand will have doubled once again. The figures in 1.1 show that rapid increase in demand and the sheer number of aircraft required to fulfil the requests.

In addition to the number of aircraft it is also forecast that the prices of oil will increase therefore making it more expensive to fly unless aircraft efficiency is improved. Air traffic currently contributes 2% of all CO2 emmisions caused by human activity where flights produce 628,000,000 tonnes of CO2 per year. The Advisory Council for Aeronautical Research in Europe (ACARE) have set targets to reduce fuel consumption by 2020, to achieve this goal a 20-25% CO2 reduction through the designing of efficient aircraft will be required.

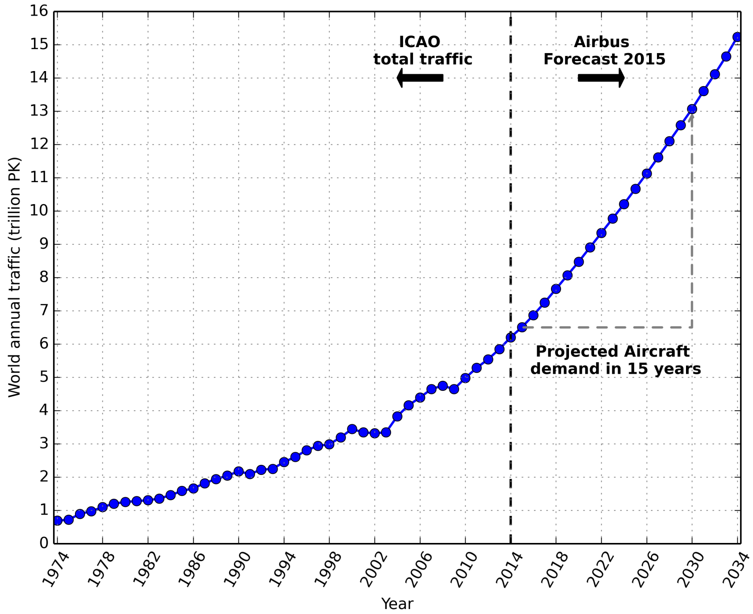


Figure 1.1: World annual traffic forecast, data from Airbus Global Market Forecast [1]

Modern civil transport aircraft normally fly in a transonic regime. At this flow condition, a local shock wave can form on the upper surface of the aircraft wing. Across the shock wave there is an increase in entropy which increase drag which is called wave drag. Wave drag can dramatically affect aircraft performance at high transonic speeds. Furthermore, the boundary layer suffers from increasingly strong adverse pressure gradients as the shock wave strength grows and can eventually cause separation, leading to a large drag rise. As a result, the lift-to-drag ratio is suddenly reduced, which limits aircraft performance.

The reduction of wave drag is a key technique to improve aircraft performance in terms of fuel economy. Flow control techniques for drag reduction can be broken down into two approaches: active and passive. The shock control bump belongs in the passive approach and has shown the capability to reduce shock strength and wave drag. Control of the shock using surface curvature was first explored in the late 1970's by Tai [8]. This paper proposed a set of aerofoils (dromedaryfoils) which were a combination of a supercritical aerofoil with a humped upper surface to treat the drag divergence Mach number. The 2D shock control bump on a natural laminar flow (NLF) aerofoil was first proposed by Fulker *et al.* [9] and Ashill *et al.* [10]. The basic idea is to employ the concave part of the bump upstream before the primitive shock to induce a series of isentropic compression waves, which significantly weakens the shock strength and reduces wave drag without a large viscous drag penalty.

Aerodynamic optimisation is key to achieving more efficient designs for aircraft. To evaluate the flow around an aerodynamic body Computational Fluid Dynamics (CFD) is used. To get good capturing of the flow physics, computationally expensive codes are required which eliminate the possibility of using global optimisation methods.

Aerodynamic optimisation using gradient-based methods has been used for the last 30 years [11, 12] and it is becoming more and more popular also in the industry where most of the time large meshes are used. In this framework there is a need to quickly map the aerodynamic surface into areas featuring different degrees of sensitivity. Not only areas of strong sensitivity are important but also those areas featuring null surface mesh sensitivity represent equally important information. This is a concrete industrial need and the solution of this problem will help the designer to focus only on specific area in a very large design space.

The flow-adjoint successfully decouples the sensitivity of the flow variables from the design variables [11, 12]. In contrast to the continuous approach, the discrete approach requires the evaluation of the sensitivity of the volume mesh w.r.t. (with respect to) those DVs (Design Variables) that describe a geometric change [13]. This comparatively less expensive part of the sensitivity chain, i.e. metric sensitivity, becomes more computationally expensive as the mesh grows in size. This has hindered previous simulation involving large meshes [14].

The surface sensitivity has previously been used to evaluate the success of an optimisation. Park *et al.* [15] used automatic differentiation to find the surface sensitivity for C_L, C_m and C_n and then identified areas where control surfaces could be placed. Rallabhandi *et al.* [16] used the discrete adjoint to find the sensitivity for ground metrics for boom mitigation. The surface sensitivity was then used to evaluate the success of the optimisation. Economon *et al.* [17] used the continuous adjoint where the surface sensitivity to C_D, C_L and C_{my} was plotted on the surface of several geometries.

This thesis will use a method called the mesh adjoint which uses the mesh deformation description to form a mapping between the mesh and the surface. This mapping is then differentiated to give a direct expression of the metric sensitivity in terms of the rate of change of the surface w.r.t. the design variables [14]. This is a much less computational expensive term to calculate in comparison to the metric sensitivity. When the mesh adjoint is calculated the surface sensitivity to the optimisation objective function is calculated and is easy to extract. This surface sensitivity will then be used to guide the placement of shock control bumps for drag reduction at transonic speeds.

1.2 THESIS OUTLINE

Chapter 2 is the literature review for flow control which discusses several different approaches to locally changing the flow for some aerodynamic benefit. Several devices are considered including active devices and passive devices.

Chapter 3 shows the parametrisation of the shock bump used for this work. Two different methods are considered for 2D and 3D geometries.

Chapter 4 presents the governing equations for fluid dynamics. The discretisation and numerical methods used to solve the flow are also discussed. The flow assumptions are also addressed in this chapter and the description of the turbulence model is also outlined.

Chapter 5 shows the two main methods for moving the mesh, the Delaunay Graph Method (DGM) and Linear Elasticity (LE). The main equations and implementation are demonstrated.

Chapter 6 gives a overview of optimisation methods. Several different gradient based methods are reviewed and demonstrated. Gradient calculation is the most important part of gradient-based optimisation, several techniques exist for accurate gradient calculation. Finally, the optimisation chain shows how the CFD simulations and gradient calculations fit into the optimisation chain.

Chapter 7 is the first results section. The 2D optimisations on the RAE5243 for a cubic-spline shock bump are presented and analysed. A multi-objective optimisation is also shown where a weighted sum objective is used with weights calculated using ‘unattainable aspiration’.

Chapter 8 shows the results for using the solution from the mesh adjoint, to derive the objective function surface sensitivity for flow control design placement. Several regions for optimisation are identified on the M6 and F6 wings. The shock bumps in this section are parametrised using CST.

Chapter 9 presents the development of a novel method for calculating the mesh adjoint sensitivities by using an explicit definition of the mesh points to the surface. This creates a fast and efficient method for calculating the mesh adjoint solution. The new method is used in an optimisation on the M6 wing to show that the method is robust and accurate.

Chapter 10 gives some concluding remarks based on the methods proposed in this thesis.

2

LITERATURE REVIEW

2.1 FLOW CONTROL

Flow control is the act of changing the flow physics to enact some favourable change. Flow control devices have been used for many years to give more desirable flow features, examples include Vortex Generators (VGs) and flaps and slats.

Flow control can be divided into two categories, active and passive. The exact definitions of active and passive flow control are somewhat fluid in the literature. In this work, the definitions supplied by Gad-El-Hak [18] are used to describe different groups of devices. A passive device is characterised as an un-powered device with no moving parts. Passive devices are often simpler compared to active devices and are easier to implement. A device designed for a specific on-design condition may introduce large negative effects at off-design conditions. It is therefore important to consider off-design performance of passive devices in the design process to minimise the negative impact.

An active flow control device is one which requires some input energy to function, an example of an active device is blowing/suction. A major benefit of active devices is that they can be operational for only certain parts of a flow regime, giving a benefit when necessary and reducing any potential negative effects at off-design conditions by being turned off. Active devices can also adapt to different flow regimes to extend the devices usefulness. This can be done through a feedback/forward loop or through a predetermined operational cycle.

The subset of active devices which are controlled by a loop are referred to as reactive devices. A feedback loop has a sensor placed upstream to monitor the incoming flow such that the flow control device can adapt to incoming flow. One example of this could be to quell the growth of Tollmien-Schlichting (T-S) waves by sensing the incoming waves and superimposing the inverse wave using a flow control device. In the case of a feed-forward device the sensor is placed downstream and monitors changes that have been made by the flow control device.

The following sections will look at different types of control and the applications in different flow regimes.

2.1.1 *General Flow Control*

2.1.1.1 *Laminar Flow control*

Laminar flow control focuses on increasing the amount of laminar flow that is present on the wing surface. This involves delaying the transition location for as long as possible to achieve a reduction in skin friction drag. In 2D flows T-S waves are a major factor which governs the transition location and the ultimate conversion to a turbulent boundary layer. For 3D flows cross-flow instabilities are also present creating a much more complex transition structure.

By far the simplest method to promote more laminar flow is through shape modification. Natural Laminar Aerofoils (NLF) manipulate the surface pressure distribution such that a favourable pressure gradient extends to the longitudinal location of the pressure minimum, this is achieved through surface curvature. The favourable pressure gradient decreases the pressure to a point on the control surface and thus increase the laminar extension, reducing the skin friction. At transonic speeds a major drawback of NLF aerofoils is the strong shock waves which are generated compared to the supercritical counterparts. This causes the wave drag to increase significantly. NLF aerofoils are also very sensitive to surface roughness making the technology high maintenance to keep surfaces clean. Figure 2.1 shows an example of a NLF aerofoil and a supercritical aerofoil for comparison.

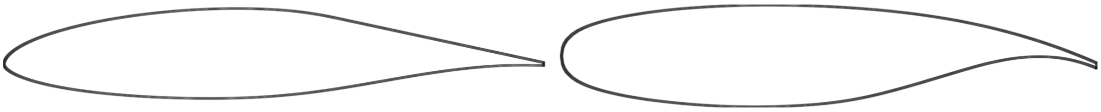


Figure 2.1: Examples of (left) NLF (RAE5243) aerofoil and (right) supercritical (NASA SC(2)-0714) aerofoil

Another method for controlling the boundary layer to promote laminar flow is to use suction. Suction removes the current boundary layer and replaces it with high energy flow from the free-stream which removes the instabilities and refreshes the velocity profile at the wall. Suction also reduces the boundary layer thickness. Although suction can be used to great effect the extra weight of mechanisms required may outweigh the overall benefits.

Wall heating/cooling can also be used to promote laminar flow near the surface by reducing viscous effects. In a liquid the wall should be heated to provide a reduction in viscous effects however, for flows that are supersonic or slower in air, heat should be removed from the aerodynamic surface, this increases the near-wall density and enhances the near-wall momentum. By reducing the viscosity close to the surface the velocity profile becomes fuller and is more resistant to instabilities in the boundary layer. In air there is an additional benefit when cooling is used, namely the density is increased close to the surface which enhances the near-wall momentum.

Although increasing laminar flow reduces the skin friction drag, there are drawbacks. Laminar flow is much more susceptible to separation compared to a turbulent boundary layer and it is also difficult to maintain a laminar boundary layer due to contamination on the wing. This contamination could take the form of insects, rivets or even icing.

2.1.1.2 *Separation Control*

Separation occurs when the flow on a surface experiences strong adverse pressure gradients in the boundary layer, detaching the flow from the surface. When the flow is separated a large increase in pressure drag occurs and the velocity in the near-wall region is reversed.

To stop the flow from separating, flow control devices can be used to energise the boundary layer. A common practice is to use vortex Generators (VGs) which create stream-wise vortices which travel downstream with the flow and bring high energy flow from the free-stream into the near-wall region.

Modern aircraft wings have multiple elements, called flaps and slats, to achieve higher lift at take-off and landing. When these devices are deployed the slat deploys from the wing leading edge tilting towards the incoming flow. This allows the incoming flow to move over the upper surface of the wing more easily at higher angles of attack, therefore generates greater suction above the wing. When the slat is deployed a gap can be present between the slat on the main aerofoil body, this is called a slot. This gap transports high pressure flow from the underside of the wing to the upper surface and re-energizes the boundary layer.

Flaps are mostly located at the trailing edge of the wing and come in several different combinations. A simple hinged flap rotates downwards to increase the camber on the aerofoil which increases lift. More complicated arrangements also increase the chord and therefore the surface area to increase lift and reduce wing loading, such as the Fowler flap. Slotted flaps also allow flow to move from the underside of the aerofoil to prevent separation and can consist of several elements such as the triple-slotted flaps which are present on the Boeing 747.

2.1.1.3 *Plasma Actuators*

A Dielectric Barrier Discharge (DBD) plasma actuator consists of an upper electrode and a lower electrode arranged asymmetrically and separated by a dielectric barrier (e.g. Kapton film, Teflon, e.t.c.), see figure 2.2. A high voltage power supply is applied to one electrode (the other is kept as ground), the result is a plasma which forms on the dielectric barrier above the lower electrode. For most low-speed applications DBD plasma actuators have the desired effect of a ‘body-force’ on the flow ‘pushing’ the plasma away from the upper electrode. The direction of this body force depends solely on the arrangements of the electrodes therefore the power supply can be applied to either electrode [19],this also gives greater freedom in other settings.

Another application of the DBD actuator involves short nano-second pulses with a large voltage supply. The dominant effect is a rapidly heated plasma forms on top of the dielectric barrier. Under certain conditions the plasma actuator can produce plasma which is heated rapidly enough to create a compression wave which is shed from the plasma region.

It is important to grasp the physics behind the creation of plasma to fully understand the operation of a plasma actuator. Despite uncertainty in the finer physics of

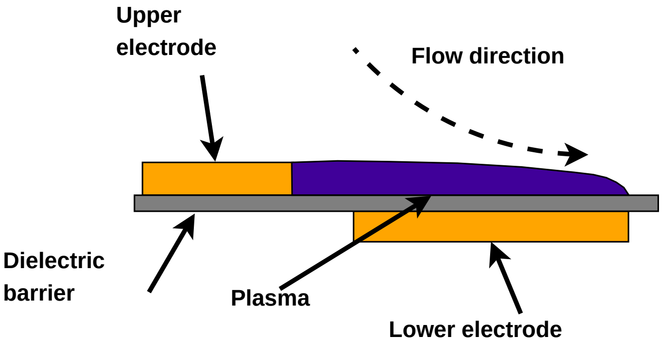


Figure 2.2: Example arrangement for a DBD plasma actuator

plasma production, a significant amount of progress has been made into explaining the formation of plasma such that an accurate model has been made for the DBD plasma actuator [20][21][19].

Post and Corke [21] showed the use of plasma actuators in controlling separation. They used wind tunnel tests to show that a plasma actuator placed at the leading edge of an aerofoil can reattach the flow to the surface of the aerofoil past the baseline stall angle of attack.

Rizzetta and Visbal [22] simulated plasma actuators under counter-flow operation for straight and serpentine configurations on the leading edge of an MAV to delay separation. LES was used to simulate the flow and separation from the aerofoil. Despite different physical effects these actuators were operated in continuous and pulsed modes and all cases found a benefit by lessening the vertical extent of the shear layer. The pulsed mode is deemed to be the best of the two operation modes as it uses less energy due to a reduced duty cycle. The straight actuator generated coherent spanwise vortices whereas the serpentine actuator delayed the appearance of 2D modes and accelerated transition.

Despite the promise of plasma actuators, as a flow control device, there remains the issue of whether there is a net benefit to using the device. Plasma actuators require a power input from a high voltage source. This would require the use of a large transformer to get the voltage required to create a plasma. In addition, the flow benefit needs to outstrip the power requirements.

2.1.2 *Shock Control*

2.1.2.1 *Wave Drag Reduction*

For a wing moving through a fluid when speed is increased such that $M \in [0.7, 1.2]$ there is a dramatic increase in drag. Despite the wing travelling at a Mach number less than 1, the flow can accelerate to supersonic speeds in local pockets around the wing due to the curvature of the body forcing the flow to speed up. These pockets of supersonic speed are terminated by a shock wave. A shock wave forms when a body is moving fast enough through the flow so the sound waves propagating from the body being to coalesce. Across a shock wave there is a sharp change in pressure, density and temperature. The extra drag component is called wave drag and is due to the increase in entropy across the shock.

2.1.2.2 *Plasma Modelling for High-Speed Applications*

For high-speed flow control the regular DBD actuator is unsuitable due to the large increase in energy consumption needed to make the body- force have an effect on the flow. Despite this [19] have shown that a DBD plasma actuator can still be effective in reducing separation at higher speeds (M=1.5-2.0).

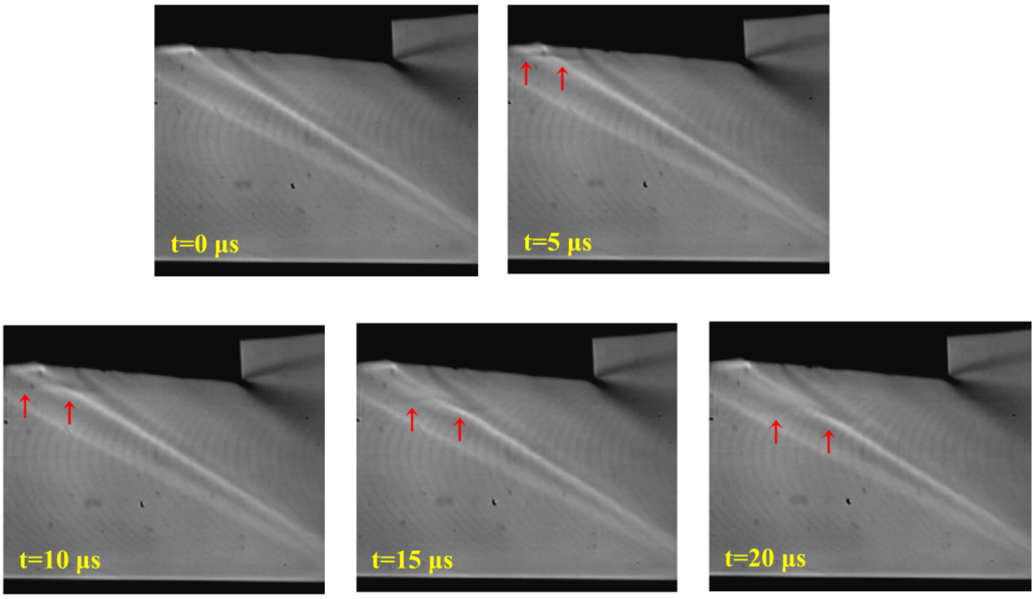


Figure 2.3: Schlieren images of a NS-DBD plasma actuator placed at the foot of a wedge with a shock present [2]. Each image has been taken at a different time after the initial pulse.

The Nano Second Dielectric Barrier Discharge (NS-DBD plasma) actuator is much more appropriate in flows where the region of control involves $M > 1$. When placed at the foot of a shock wave a NS-DBD plasma actuator has been shown to 'spread' the shock wave, shown in figure 2.3. Figure 2.3 shows the effect of placing a NS-DBD actuator at the base of a wedge with an oblique shock present where the flow is travelling from left to right. The arrows in the figure point to the radiating compression wave and the interaction with the shocks. Figure 2.4 shows the topology of the wedge with the plasma actuator arrangement and flow direction. The intention is to introduce compression of the flow in the shock region. On an aerofoil this should ease the adverse pressure gradients across the shock region and weaken the shock thus reduce wave drag.

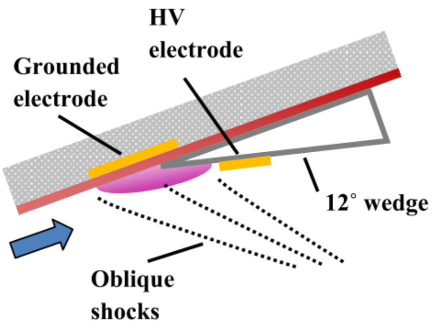


Figure 2.4: Topology of a NS-DBD actuator at the base of a wedge [2].

2.1.2.3 *Wing Sweep and Supercritical Wing*

To combat the dramatic increase in drag due to wave drag two methods have been used widely in aircraft design, namely wing sweep and the supercritical wing. A supercritical wing is characterised by a longer flatter section on the upper surface. This flatter potion minimises the angular change in flow and therefore postpones the formation of the shock until it is further downstream of the wing. This is usually located after the maximum thickness to chord point.

The swept wing was introduced since only the velocity components perpendicular to the leading edge govern the behaviour of the flow. Therefore, increasing the wing sweep reduces the perpendicular speed over the wing and weakens the shock. The drawback is that at lower speeds there is an impact on performance.

2.1.2.4 *Passive Cavity Flow Control*

The EUROSHOCK I project [23] was carried out between 1993-1995 and focussed on passive shock control. One method of shock control researched was the passive cavity. Figure 2.5 shows the passive cavity set-up. The passive cavity was found to spread the pressure change across the control region creating a lambda shock structure. The cavity allows for the flow from the high pressure side (downstream) to flow to the low pressure side (upstream). Due to the lower pressure gradient across the shock location, the wave drag has been reduced.

Despite the reduction in wave drag a significant overall drag reduction was not achieved because of the increase in viscous drag due to boundary layer thickening and the holes in the plate.

Although no net reduction of drag was achieved a porous cavity could still be useful to delay the onset of buffet.

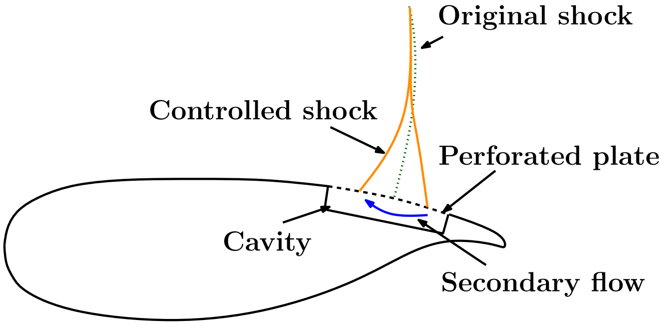


Figure 2.5: Figure of passive cavity control device.

2.1.2.5 *Active Cavity Flow Control*

The EUROSHOCK II project [24] project spanned 1996-1999 and focussed on cavity flow with suction, discrete slot suction, hybrid control and contour bump.

Cavity flow control with suction has a perforated plate above a cavity with suction being applied to the flow in the cavity, see Figure 2.6. From the previous section, the cavity with no suction creates a thickened boundary layer downstream of the device and created an increase in drag due to viscous effects. To reduce the negative viscous effects while maintaining the wave drag benefit, suction is introduced to the cavity flow.

One experiment looked at several different amounts of suction in the range $C_\mu \in [0.0,\ 0.143]$, where $C_\mu = 0.0$ is the passive cavity case where there is no suction and C_μ is defined as:

$$C_\mu = \frac{q}{\rho_{inf} U_{inf} \delta^*} \tag{2.1}$$

where q is the suction mass flux per span, ρ is the density, U is the flow velocity and δ^* is the displacement thickness of the incoming boundary layer. The inclusion of cavity suction removes some of the benefit of the wave drag reduction, in fact larger amounts of suction cause stronger shocks to occur and therefore increase wave drag. The suction allows for a fuller boundary layer velocity profile downstream of the device, as expected. When suction is at $C_\mu = 0.055$ there is no significant negative impact on the boundary layer flow and therefore viscous drag will not be changed by the presence of the control device.

2.1.2.6 *Shock Control Using Grooves/Slots*

Two interesting methods for shock control were introduced in papers by Smith *et al.* [25, 26]. The first, in 2001, employed the use of a stream-wise slot at the shock location.

The introduction of slots causes a bifurcation of the shock to form a λ-structure which gives a more gradual change in pressure across the control region. The boundary layer directly downstream from the slot is less full and indicates some potential

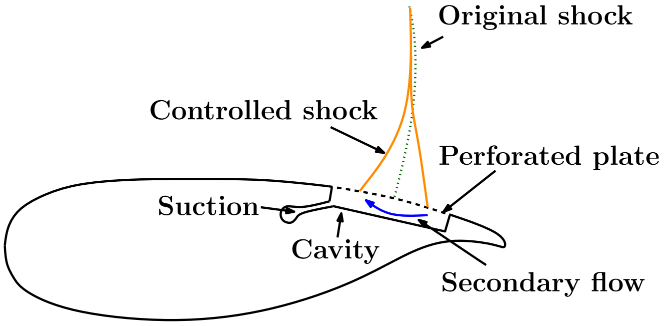


Figure 2.6: Figure of active cavity control device.

separation. The boundary layer further downstream shows some recovery due to vortices caused by the device leading edge.

Overall, slots could provide wave drag reduction with some viscous drag penalty. The device does introduce some stream-wise vortices which could be beneficial in cases where there is separation downstream of the device.

The second paper was published in 2003 and investigated further the vortical flow generated by the device. The findings showed that the vortical structure generated was much more complicated involving two sets of counter rotating vortices.

2.1.2.7 *Shock Control Bumps*

One of the first appearances of a bump-like device for wave drag reduction can be found in [8] from 1977. This report suggested adding a hump on a supercritical aerofoil (a dromedaryfoil) to treat wave drag. It is described that the fore part of the bump replaces the shock with a series of isentropic compression waves and the rear of the bump gives incipient separation due to a sharp peak. Ultimately, the research was inconclusive and requierd experimental testing.

Fuller *et al.* [9] and Ashill *et al.* [10] first proposed the 2D shock bump as a device to reduce wave drag on a laminar type aerofoil. Interest in the shock bump grew significantly after these papers were published and in 1996 the EUROSHOCK II project [24] began. The EUROSHOCK II project had that the shock bump had the most promise for shock control from a range of devices researched. Several aspects were explored including a parameter study and combining shock bumps with suction. The analysis was thorough and involved 2D and 3D analysis, however there was no rigorous optimisation of the devices.

Qin *et al.* [27] were the first to propose a shock bump with a finite bump width, this adds complexity to the design through extra design variables and spanwise geometric variations. Prior to this much of the work looked at extending the 2D bump continuously along the wing span. It was found that the 3D shock bump configuration provided better robustness and a similar reduction in drag to the 2D bump.

Due to the extensive work on shock bumps, several variations of bump geometries have been considered. In general all the bumps consist of a front part and a rear part. The front part is a ramp or smooth curve which compresses the flow. The rear part returns the flow to the aerofoil and can sometimes have an effect on the boundary layer health downstream of the device.

2.1.2.8 *3D Contour SCB*

The contour bump consists of a smooth curve joining to the aerofoil with zero gradient. There have been several methods for parametrising the contour SCB some examples include piece-wise cubic curves [27], Hicks-Henne [28] curves or Class-Shape function Transforms (CST) [29].

The optimal contour bump replaces the shock with a series of isentropic compression waves which dramatically reduces the increase in wave drag across the shock region. The effect of the bump is highly dependent on the values of the bump height and peak location, these were shown to be the most sensitive parameters [30]. A larger height will cause more compression of the local flow and will increase reductions in wave drag but a higher bump may cause separation after bump and increase overall drag. The peak location should be downstream of the shock; EUROSHOCK II suggested that the peak of an asymmetric bump should be approximately 5% downstream of the original shock location.

In 3D several different designs were explored by Wong in his thesis [3]. Figure 2.7 shows 3 different designs which have been considered. The first is the 2D style bump where the bump extends uninterrupted over a certain span. This bump has been shown to reduce drag but is not particularly versatile and not as robust as the 3D bumps. The eye-based bump was also researched, the bump consists of pointed leading and trailing edges with the middle of the bump being the widest section. This bump achieved reasonable drag reductions but generated some reverse flow in the rear part of the bump at the bump/aerofoil interface which hindered drag reduction. The final design is a rectangular based contour bump. This bump shows a very good drag reduction, in some cases better than the 2D bump. This bump consists of zero gradient at the bump/aerofoil interface and provides a smooth contour in all directions. The 3D bumps are usually of a larger height than the 2D bumps and it has also been shown that the 3D bump is more robust than the 2D bumps.

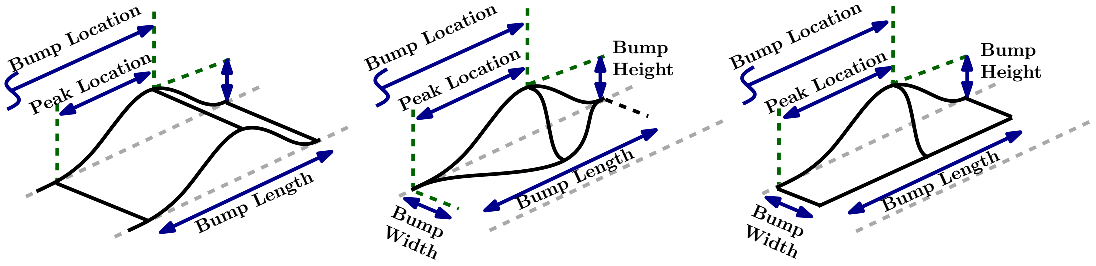


Figure 2.7: Contour type shock bumps from (left) 2D (middle) Eye-based 3D bump (right) Rectangular based 3D bump [3]

Figure 2.8 shows three different shock locations; the left figure shows the shock location upstream of the bump, the centre figure shows the optimal shock location and the right figure shows the shock location downstream of the bump.

When the shock is too far upstream this can cause an initial compression of the flow giving a wave drag reduction, however due to the bump curvature a secondary shock appears after the bump peak generating wave drag. When the shock location is at the optimal point the shock is converted into a series of isentropic compression waves on the leading ramp, reducing wave drag significantly, and the pressure contours become more spaced out indicating a lesser compression. Due to the deflection of the flow away from the aerofoil the pressure contours form this characteristic 'knee shape' which is typical of the contour SCB. When the shock moves downstream, past the peak location, the flow begins to re-expand and causes a much stronger shock than without the bump being present. This can causes a rise in wave drag and other negative effects such as separation. In addition to these 2D effects there is also a 3D spanwise component. The 3D bumps presented by Qin *et al.* [30] reduced to zero and had zero gradient at the spanwise edges. After optimisation it was also shown that the distance between the bumps in the spanwise direction tended to zero however, the effect of the bump on the flow in between the bumps still produced isentropic compression waves to treat the shock.

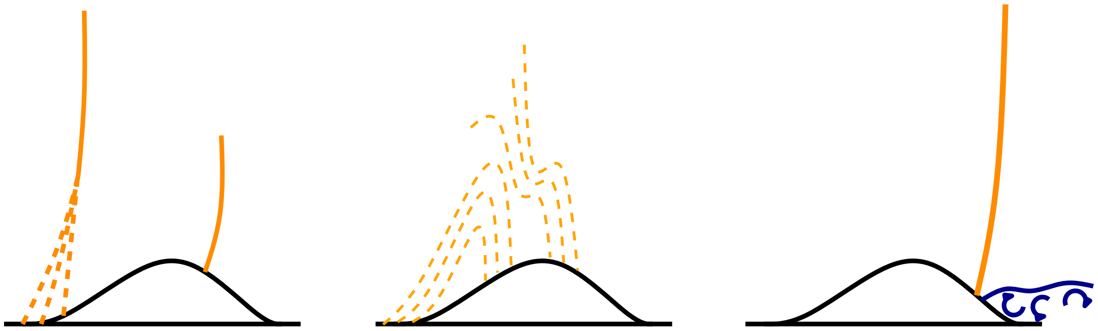


Figure 2.8: Contour bump with effects of shock location (left) Upstream of bump (middle) Optimal location (right) Downstream of bump

2.1.2.9 *3D Wedge SCB*

Much of the literature is focussed on variations of two particular geometries; the wedge bump and the contour bump. The wedge bump consists of a front ramp, with a leading edge angle, a crest and plateau region and the rear ramp, with trailing edge angle. The 3D wedge bump has several different configurations as shown in Figure 2.9. The leading edge ramp of a wedge bump has an angle at the fore which triggers a shock leg to form. Then ramp deflects the flow away from the aerofoil and the flow experiences a compression until the crest is reached. The angle at the crest causes another shock leg to form on the plateau section of the bump. The rear ramp returns the flow to the surface of the aerofoil.

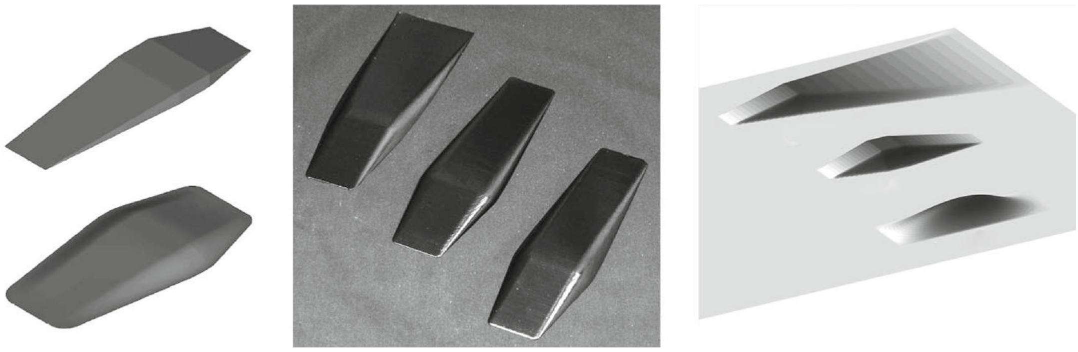


Figure 2.9: Wedge type shock bumps from (left) Ogawa *et al.* [4] (middle) Bruce and Babinsky [5](right) Colliss *et al.* [6] (As shown in [7])

Shock bumps are known for being sensitive to the shock location. Figure 2.10 shows the effects of change in shock location. The left shows the effect on the shock if the shock is upstream. There is an initial small bifurcation of the shock which gives a more gradual pressure change in the shock region. However, the expansion of the flow due to the rear ramp will cause a secondary shock to form and therefore act against the wave drag reduction that may be achieved by the initial shock bifurcation. The centre figure shows the optimal shock location and bifurcation of the shock, this case shows the normal operation of a wedge SCB. The right figure shows the shock location being downstream of the bump. The ramp triggers a front shock leg and there is still a bifurcation of the shock but due to the re-expansion of the flow at the rear of the bump the rear shock leg is stronger causing the device to operate inefficiently.

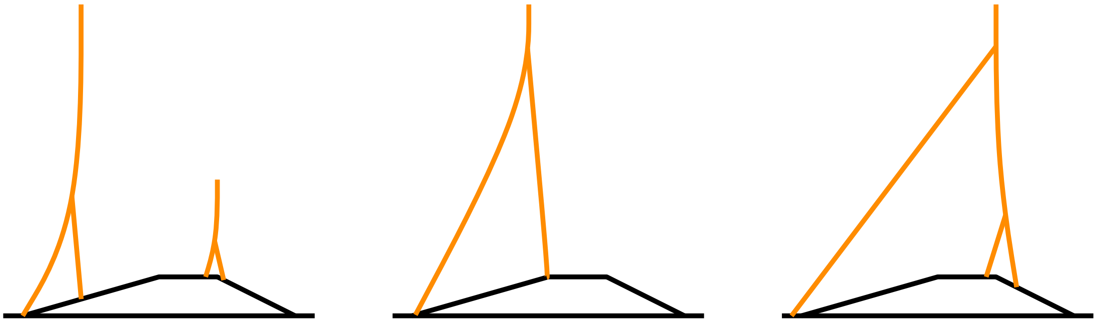


Figure 2.10: Wedge bump with effects of shock location (left) Upstream of bump (middle) Optimal location (right) Downstream of bump

König *et al.* [31] considered a bump which had an angled leading edge like a wedge bump but had a smooth profile similar to a contour bump. When this configuration was placed in a wind tunnel, as the angle of attack is changed from design there is very little movement in the shock location in the control region. It is suggested that the design may offer some robustness at off design due to the sharp leading angle but no comparison was made to other shock bump configurations, in particular the smooth contour bump. Despite this the bump does show good robustness when varying the angle of attack and generates streamwise vortices which could prove useful in alleviating shock induced separation.

Further studies considered geometries which could reduce the detrimental effects on the boundary layer health by altering the rear ramp configuration [6].

In 3D the distance between bumps has to be considered. For the wedge bump the extent of the spanwise influence of the shock control has to be considered to preserve the wave drag reduction and to retain any potential benefits from the streamwise vortices.

2.1.2.10 *2D vs. 3D SCB*

When moving from the 2D style bumps to 3D they're are many considerations to take into account such as bump width, distance between bumps, streamwise angular orientation and so on. The main difference of 3D bumps in comparison to 2D bumps is that they have geometric variations in the spanwise direction. This means that the boundary layer can now go around the bounds and therefore the shock-boundary layer interactions will be different.

Zhu *et al.* [29] showed that the 3D bumps can vary in height along the span so that the shock is treated locally. They also showed that for a complicated 3D shock (i.e. the shock footprint is not a straight line) the 3D shock bumps can be placed to follow the shock line. Since the shock bumps are very sensitive to changes in the shock location it is clear that placing a 2D type bump in a flow with 3D shock structure in the spanwise direction will prove a significant issue for 2D bump performance.

It has been widely documented that 3D shock bumps can produce streamwise vortices [4, 32, 31] due to the spanwise variation of the shock bump geometry. While the savings at the design conditions between 2D and 3D bumps can show similar performance benefits [4, 32, 31, 33, 34], or in some cases the 3D geometry can perform better than the 2D [30], it is widely believed that the generation of vortices will give better benefits at off-design conditions. At off-design conditions the shock may cause adverse pressure gradients to form causing shock induced separation, the formation of streamwise vortices could combat this by moving the low momentum flow from the near wall region and replacing it with high momentum flow from the freestream.

Qin *et al.* [30] and Holden and Babinsky [35] showed that the placement of the bumps was important to achieve similar shock effects to the 2D bump. In the case of Qin *et al.* , they found that under optimisation on an unswept NLF wing and a 3D contour SCB that the distances between bumps tended to zero. It was noted that the bump height was larger than the 2D counterpart. This was also noted by Eastwood and Jarrett [36] where the wedge bump was found to have more height when the bumps were placed further apart. One possible explanation for this is through the transonic area rule. When the distance between the bumps grows the bump height must increase to keep the cross sectional area the same.

Pätzold *et al.* [33] tested optimised single truncated shock bumps on unswept and swept wings. They found that when imposing a leading edge angle on a wing and placing the bumps normal to the leading edge it is difficult to maintain the performance benefits which are found for the unswept wing case. This hints that there is possible crossflow interference affecting the operation of shock bumps. Eastwood [34] researched placing small arrays of bumps on a swept wing with bumps normal to the leading edge angle and also found that there was a performance degradation compared to the unswept wing but some benefit at off-design could still be achieved. Eastwood and Jarrett [36] explored the idea of angling the wedge type bumps on swept wings to regain a similar performance benefit to the unswept wing. They showed that angling the bump roughly in the local flow direction reduced the negative impacts from the spanwise flow.

2.1.2.11 *Adaptive Bumps*

There have been some promising studies into adaptive bumps which would extend the useful envelope of the shock bump. The major issues are extra system weight (actuators etc), maintenance due to moving parts and life-expectancy of the device. However, the NASA report on morphing technology [37] shows that adaptive flow control devices are definitely possible.

Monner *et al.* [38] suggested that the use of a flexible spoiler skin could create a convex surface distortion by using Carbon Fibre Reinforced Plastic (CFRP) tubes with 2 symmetrical hollows. This would create a shape very similar to the contour SCB and could be used to increase the height and gradient of slope to suit the incoming flow. The concept of being able to move the surface was demonstrated but there were no flow results to show the potential benefits of the device.

Stanewsky [39] reviewed the benefits of adaptive technologies, such as an adaptive shock bump and flap combination, and highlighted some potential systems that could be employed. A study of Cash Operating Costs (COC) and Direct Operating Costs (DOC) was undertaken where drag reduction, weight increase and and increase in maintenance costs were all considered. The values of COC and DOC were found assuming 600 flights at a range of 3500 nm over the course of a year. The adaptive bumps were tested on a turbulent type wing at two flight conditions, M=0.82 and M=0.84, which gave a fuel reduction of 1.23% and 2.11% respectively when adaptive bumps were applied. This leads to a COC and DOC reduction of 1.3% and 0.8% respectively. This shows the significant potential for the use of shock bumps of transonic commercial aircraft.

Jinks *et al.* [40] recently showed a coupled aerodynamic-aeroelastic optimisation of a flexible surface in the shock location. The same author also suggested a way in which this device could be achieved mechanically [41]. It was shown that a small bump height was enough to trigger a bifurcation in the shock which is necessary to control the shock with this type of bump. Further rigorous study of the drag and weight requirements would be necessary to assess the potential benefits of applying this device for shock control.

2.1.2.12 *Multi-point/Multi-Objective Analysis*

Despite the promise shown, adaptive technologies still have significant drawbacks. It is therefore necessary to explore avenues to make the fixed devices more robust to minimise the negative effects at off-design conditions.

This can be done through uncertainty or multi-point simulation. Uncertainty quantifies the effect of potential changes in a flow variable on the objective function and optimises to reduce the negative effects. Multi-point analysis involves calculating the variable of interest at several different flight conditions (often located close to the design point) and uses some metric as an objective function. For example, it is common to use a simple weighted sum of drag values to evaluate the overall performance. The drawback of a multi-point method is that is requires weights to be given to each flight condition to mark the importance in the optimisation. These weights are often left to the designer to choose and can have a large effect on the final topology which is chosen.

Lee *et al.* [42] used the MSES solver and evolutionary algorithms to optimise a shock bump on the RAE5243 aerofoil. MSES solves the Euler equations and is coupled with a boundary layer solver for the near-wall flow. The optimisation was run with two objectives, the first is the mean of the drag values and the second is the standard deviation of the drag values:

$$\begin{aligned} \overline{C}_D &= \frac{1}{K}\sum_{i=1}^{K} C_{D_i} \\ \sigma C_D &= \sqrt{\frac{1}{K-1}\left[\sum_{i=1}^{K}\left(C_{D_i}-\overline{C}_D\right)^2\right]} \end{aligned} \tag{2.2}$$

After the optimisation a Pareto front was formed and several designs were selected and further evaluated. The main difference between the robust bumps and the single-point design bump is that the bump height was reduced and the bump had moved downstream, both of these changes would combat a strong re-expansion on the rear part of the bump

MSES is a fairly low cost solver and the case is only 2D, therefore a global method such as an evolutionary algorithm can be used. When a more expensive solver is used, such as a Navier-Stokes code with turbulence model, the cost of the objective function increases significantly. This is particularly clear when a 3D geometry is considered. For these cases it is better to use a gradient based solver, this comes at the cost of only being able to find local minima. Due to the expense there is no comprehensive 3D multi-point optimisation in the literature.

Eastwood and Jarrett [34] showed a numerical investigation into 2D and 3D ramp type bumps. The bump topology consists of a shorter compression ramp at the front with a longer ramp at the rear with no plateau at the peak. The shoulders of the ramp form an angle with the bump and wing surface. This paper hypothesised that the robustness of 3D bumps over 2D bumps comes from the streamwise vortices generated by the 3D configuration. The 3D shock bump showed a break up of the shock induced separation downstream and held the shock in place for a variation in the M_{inf}. This shows great promise for 3D shock bump robustness, however further exploration into into other geometries may yield better results.

Part II

MODELS AND METHODS

3

PARAMETRISATION OF A SHOCK CONTROL BUMP

To optimise a surface it is common practice to use a parametrisation, this is often necessary to reduce the number of design variables or to restrict the geometry to feasible shapes. For the rest of this thesis the contour SCB will be used (and referred to as 'shock bump'). Several parametrisations have been used previously for use with shock bumps; Hicks-Henne [43], cubic-spline [30], PARSEC [44] and CST [29].

In this work the cubic spline is used for 2D analysis. This approach is beneficial as the parameters are intuitive and provide a simple way to express the bump and analyse the effect of changing a physical variable. The drawback of the cubic spline approach is that only C^1 continuity can be guaranteed.

In 3D the CST is used to describe the bump which offers a higher degree of freedom and variation in shapes which gives a larger design space to be explored. In addition to this the CST method has C^2 continuity which guarantees a smoother profile and is often a requirement for industrial uses.

3.1 TWO DIMENSIONAL BUMP

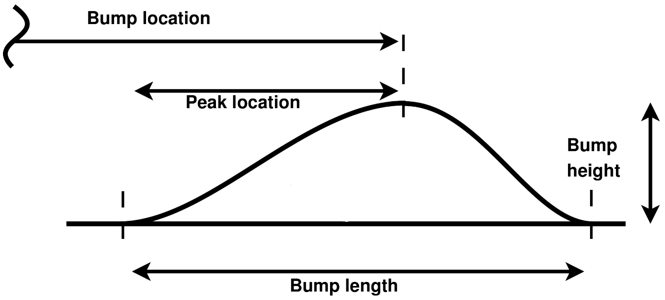


Figure 3.1: 2D shock bump geometry using cubic spline

Figure 3.1 shows a representation of a 2D shock bump parametrised using a cubic spline, as shown in Qin et al [30] and Wong [3]. The description of the geometry consists of 4 parameters: bump height, bump length, bump location and peak location. The bump height measure the distance from the bottom of the bump to the peak (lowest point to highest point). Bump length is the distance from the bump leading edge to the bump trailing edge. The bump location is measured from the leading edge of the aerofoil to the peak of the bump, this allows the bump to move along the aerofoil to find the ideal location to treat the shock. The peak location is the distance of the bump peak from the bump leading edge.

The bump surface is described by two cubic curves. The first links the bump leading edge to the bump peak and the second joins the bump peak to the bump trailing edge. At the bump leading edge, bump peak and bump trailing edge the gradient is zero to give a smooth contour. The bump is added to the surface of an aerofoil by adding the y components.

The intuitive design parameters are good for exploring the key design factors of a shock bump but it lacks versatility and is only continuous to the first order.

3.2 CST THREE DIMENSIONAL BUMP

The design parametrisation used previously for a 3D shock bump control is generally simple, an example using cubic splines is shown by Qin *et al.* [27]. With industrial manufacturing methods, the second order continuity, C^2, is required. In order to satisfy C^2 continuity, the order of piecewise polynomials has to be increased to 5. Higher order polynomials will lead to a higher degree-of-freedom and contain more than one peak in the curve; this will cause uncontrollable waviness in the bump. A different shock control bump parametrisation is proposed in this work based on the CST parametrisation method [45]. The CST method has two parts: the class function and the shape function. The class function determines the basic type of geometry and the shape function is then employed to define the details.

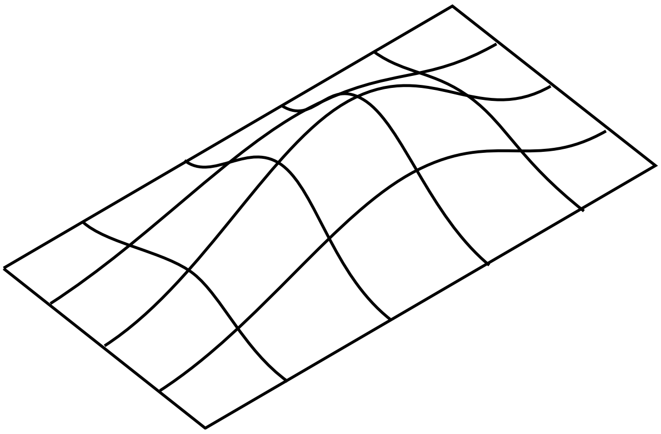
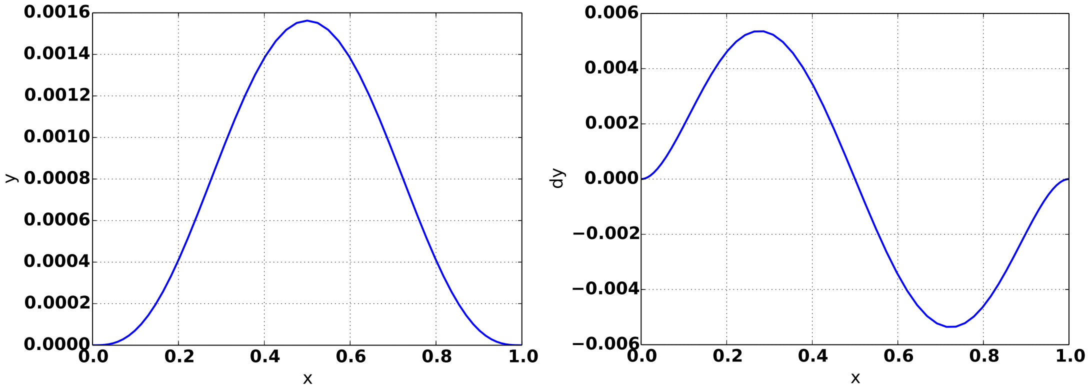


Figure 3.2: 3D shock bump parametrised using CST

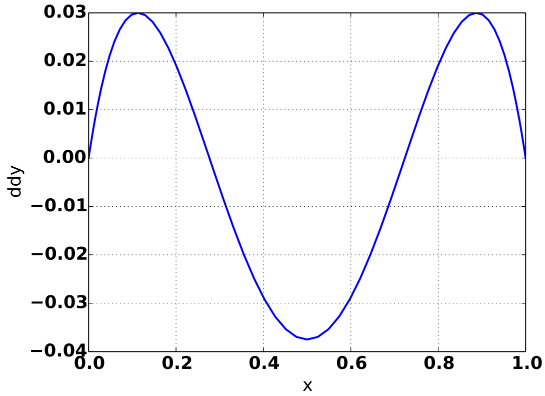
If the class parameters N1 and N2 are set to 3 this sets the general class of shapes to follow a similar shape as the cubic-spline. Setting the shape function $S(\psi) = 1$, as in Eq. (3.1), a bump like curve is obtained with 1^{st} and 2^{nd} derivatives are zero at start and end points. The curve and 1^{st} and 2^{nd} derivative distribution are shown in Fig. 3.3.

$$\xi(\psi) = \psi^{3.0} \cdot (1 - \psi)^{3.0} \tag{3.1}$$



(a) Bump curve

(b) 1^{st} derivative



(c) 2^{nd} derivative

Figure 3.3: Bump curve and derivatives using the CST parameterisation method

The figures above clearly show that the geometry, 1^{st} and 2^{nd} derivatives are all zero at the start and end position of the bump. In addition, because the class parameters are exponential parameters of the class function, if N1 and N2 are set to 3, the bump peak value reduces to 1/64 for the shape function $S(\psi) = 1$. Because the bump maximum height and bump peak crest position parameters do not directly appear in the CST function, the shape function parameter values should be a similar magnitude to bump height. Multiplying by 64 in the CST equation moves the peak value back to

1 when $S = 1$. This is convenient for the user when setting up their design parameter range. The full description of the 2D CST bump with shape function is:

$$\xi(\psi) = 64 \cdot C_{3.0}^{3.0}(\psi) \cdot \sum_{i}^{n} A_i \cdot S_i(\psi) \tag{3.2}$$

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1} \cdot (1-\psi)^{N_2} \tag{3.3}$$

$$S_i(\psi) = \sum_{r=0}^{n} K_{r,n} \psi^r (1-\psi)^{n-r} \tag{3.4}$$

$$K_{r,n} = \binom{n}{r} = \frac{n!}{i!(n-i)!} \tag{3.5}$$

$$x = \psi \cdot x_{length} \tag{3.6}$$

$$z = \xi \cdot x_{length} \tag{3.7}$$

The three-dimensional bump is an extension of the two-dimensional bump using a second Bernstein polynomial. Finally, the bump patch may not be strictly a rectangle, so the bump length distribution along spanwise is a function of span. The definition of a three-dimensional bump with sweep angle is shown in the following equations:

$$\xi(\psi, \eta) = 64 \cdot C_{3.0}^{3.0} \cdot \sum_{i}^{N_x} \sum_{j}^{N_y} \left[B_{i,j} \cdot Sx_i(\psi) \cdot Sy_j(\eta) \right] \cdot H(\eta) \tag{3.8}$$

where:

$$H(\eta) = 64 \cdot C_{3.0}^{3.0}(\eta) \cdot \sum_{i}^{n} A_i \cdot S_i(\eta) \tag{3.9}$$

$$x = x_{leading}(y) + \psi \cdot x_{length}(y) \tag{3.10}$$

$$y = \eta \cdot y_{width} \tag{3.11}$$

$$z = \xi \tag{3.12}$$

$$x_{length}(y) = x_{leading}(y) - x_{trailing}(y) \tag{3.13}$$

where $x_{leading}$ and $x_{trailing}$ are the leading edge and trailing edge values of x at a given y, which can be a higher polynomial function depending on the distribution shape. The CST bump equations can provide higher flexibility of a local bump, and generate symmetric or asymmetric bumps in three-dimensional space. The orders of Bernstein polynomials are recommended to be below 4, this allows the CST bump to provide high flexibility with a reasonable number of design variables and produce the most realistic bump shapes.

4

NUMERICAL FLOW SOLVER AND GOVERNING EQUATIONS

Computational fluid dynamics (CFD) is playing an increasing important role in engineering applications. Industrial methods have incorporated the use of computational techniques alongside experimental testing such that it is now an integral part of the design process. For the numerical optimisation undertaken in this thesis the CFD flow solver is used to evaluate the objective function for each new aerofoil/wing design.

CFD is an ongoing field of study where new advancements are frequently made which improve calculation efficiency or provide increased accuracy of the solution. Despite these efforts, there are still many assumptions and approximations used throughout CFD. Due to this, care must be taken when setting up a simulation to ensure a trustworthy final solution.

The governing equations are an application of the physical laws of nature defined by Newton to a suitable model of the flow. These equations can be written in differential or integral form, both forms are based on the same equations but the major difference is that the integral form will allow discontinuities, such as shocks, in the solution. This is due to the differential forms assuming continuity which is necessary for taking derivatives. The fluid will be considered as a continuum and will be subject to compressibility effects.

4.1 THE NAVIER-STOKES EQUATIONS

As previously stated, the governing equations can be derived by using the three conservation laws, mass, momentum (Newtons 2^{nd} law) and energy, then applying these to an appropriate flow model such as a fluid control volume or infinitesimal fluid element.

For an arbitrary fluid volume V with boundary ∂V, differential surface boundary dS and normal vector to the surface $((n))$, the integral form of the governing equations is:

$$\frac{\partial}{\partial t}\int\int\int_{V} \mathbf{W} d\mathbf{V} + \int\int_{\partial V} [\mathbf{F}^c + \mathbf{F}^\nu] \cdot \mathbf{n} \cdot d\mathbf{S} = 0 \tag{4.1}$$

or in differential form:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot [\mathbf{F}^c + \mathbf{F}^\nu] = \frac{\partial \mathbf{W}}{\partial t} + R(\mathbf{W}) \tag{4.2}$$

where **W** is the vector of conserved quantities:

$$\mathbf{W} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix} \tag{4.3}$$

ρ is the density of the fluid, u, v, w are the velocity components in the x, $,y$ $,z$ directions and E is the total energy for each control volume. $\mathbf{F}^c$ and $\mathbf{F}^v$ represent the convective (inviscid) and viscous flux tensors and are defined in three dimensions as:

$$\mathbf{F}^c = \begin{bmatrix} \mathbf{F}^c_x \\ \mathbf{F}^c_y \\ \mathbf{F}^c_z \end{bmatrix}, \qquad \mathbf{F}^v = \begin{bmatrix} \mathbf{F}^v_x \\ \mathbf{F}^v_y \\ \mathbf{F}^v_z \end{bmatrix} \tag{4.4}$$

These flux components themselves are also vectors which can be viewed in greater detail:

$$\mathbf{F}^c_x = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ \rho H u \end{bmatrix}, \quad \mathbf{F}^v_x = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix}, \tag{4.5}$$

$$\mathbf{F}^c_y = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ \rho H v \end{bmatrix}, \quad \mathbf{F}^v_y = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix}, \tag{4.6}$$

$$\mathbf{F}^c_z = \begin{bmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ \rho H w \end{bmatrix}, \quad \mathbf{F}^v_z = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix} \tag{4.7}$$

where p is the static pressure, H is the total enthalpy, τ_{ij} is the viscous stress tensor for a fluid element and q_i is the heat transfer flux vector (i, $j \in x$, y, z).

The viscous stress tensor due to the molecular viscosity are given by:

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \tag{4.8}$$

$$\tau_{yy} = 2\mu\frac{\partial v}{\partial y} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \tag{4.9}$$

$$\tau_{zz} = 2\mu\frac{\partial w}{\partial z} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \tag{4.10}$$

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{4.11}$$

$$\tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \tag{4.12}$$

$$\tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \tag{4.13}$$

where μ is the dynamic viscosity. The dynamic viscosity is a function of temperature and a relationship was proposed in the form of Sutherland's law:

$$\mu = \mu(T) = \mu_0\left(\frac{T}{T_0}\right)^{\frac{3}{2}}\frac{T_0 + S}{T + S} \tag{4.14}$$

where μ_0 is the reference viscosity, T_0 is the reference temperature and S is Sutherland's temperature. For a case in air:

$$\mu_0 = 1.7894 \times 10^{-5} \quad \text{kg m}^{-1}\text{s}^{-1} \tag{4.15}$$

$$T_0 = 288.15 \quad \text{K} \tag{4.16}$$

$$S = 110.4 \quad \text{K} \tag{4.17}$$

The heat flux term can be calculated using the thermal conductivity relation (Fourier's law):

$$q_x = -\kappa\frac{\partial T}{\partial x} \tag{4.18}$$

$$q_y = -\kappa\frac{\partial T}{\partial y} \tag{4.19}$$

$$q_z = -\kappa\frac{\partial T}{\partial z} \tag{4.20}$$

where κ is the coefficient of thermal conductivity and is defined as:

$$\kappa = \mu\frac{C_p}{Pr} \tag{4.21}$$

where Pr is the Prandtl number and C_p is the specific heat at constant pressure. The ratio $\frac{C_p}{Pr}$ is approximately constant for most gases therefore the Prandtl number can be taken as 0.72 everywhere. The total enthalpy H can be related to the total energy per control volume E using the following equation.

$$H = E + \frac{p}{\rho} \tag{4.22}$$

where

$$\begin{aligned} E = \frac{p}{\gamma - 1} + \frac{\rho}{2}\left(u^2 + v^2 + w^2\right) = \\ C_p T + \frac{1}{2}\left(u^2 + v^2 + w^2\right) - \frac{p}{\rho} \end{aligned} \tag{4.23}$$

putting the above definition of E into equation 4.22 gives

$$H = C_p T + \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{4.24}$$

γ refers to the ratio of specific heats and is expressed as $\gamma = \frac{C_p}{C_v}$. In air $\gamma = 1.4$. For closure of the governing equations the equation of state for an ideal gas is used. This equation shows that for an ideal gas the density is proportional to the pressure and temperature.

$$p = \rho R T \tag{4.25}$$

where R is referred to as the gas constant. In air the gas constant has a value $R = 287.05 \text{ J kg}^{-1} \text{ K}^{-1}$.

4.1.1 *Modelling Turbulence*

The previous section showed the equations for the conservation of mass, momentum and energy. These equations can be used to model fluid flows directly using Direct Numerical Simulation (DNS). DNS requires a very fine mesh such that all relevant turbulence features can be resolved in time and space. If DNS is employed then no turbulence modelling is necessary. However this method becomes computationally prohibitive due to the size of the problem. Other methods are implemented to model the turbulence so the mesh and time step do not need to be as fine and therefore the size of the problem is reduced. One way of doing this is to introduce Reynolds decomposition to time-average the Navier-Stokes equations. This gives rise to the Reynolds Averaged Navier Stokes equations.

After introducing Reynolds decomposition, where the flow properties are divided into the mean and fluctuating components, new terms arise increasing the number of unknowns in the system of equations. These new terms are referred to as Reynolds stresses due to turbulence. The Reynolds stresses introduce more unknowns than equations, in 3D the number of extra unknowns introduced, in addition to the original five (ρ, u, v, w, p), is 6 and only 5 equations. This situation is known as a 'closure problem' and requires either extra equations to close the set of equations or some assumptions of the unknowns need to be made. Several turbulence models have been created to approximate values for the Reynolds stresses by formulating more equations and making some assumptions to close the equations.

4.1.1.1 *Reynolds-Averaged Navier-Stokes (RANS) Formulation*

The Reynolds decomposition takes an instantaneous flow variable and decomposes it into a mean part and a fluctuating part. For the TAU code Favre-averaging is used such that compressibility effects are well represented.

The instantaneous flow variable (W) will be decomposed into mean and fluctuating parts.

$$\mathbf{W} = \tilde{\mathbf{W}} + \mathbf{W}'' \tag{4.26}$$

where ˜denotes the mean component and ″ denotes the fluctuating component of the flow (in this case a Favre average).

The mean part of equation 4.26 is formulated as:

$$\tilde{\mathbf{W}}(x,t) = \frac{1}{\bar{\rho}T}\int_{t}^{t+T} \rho(x,s)\mathbf{W}(x,s)ds \tag{4.27}$$

where $\bar{\rho}$ is the Reynolds averaged density. Decomposing the primitive variables for an instantaneous fluid flow gives:

$$\begin{gathered} u = \tilde{u} + u'' \qquad v = \tilde{v} + v'' \qquad w = \tilde{w} + w'' \\ \rho = \tilde{\rho} + \rho'' \qquad p = \tilde{p} + p'' \qquad T = \tilde{T} + T'' \end{gathered} \tag{4.28}$$

When equations 4.28 are substituted into equation 4.1 or equation 4.2 two extra terms arise which causes the 'closure problem'. The Reynolds stress tensor is the first term:

$$\tau_{ij} = -\overline{\rho u_i'' u_j''} \tag{4.29}$$

The turbulent heat flux is the second term:

$$q_{t_i} = \overline{\rho H'' u_i''} \tag{4.30}$$

where q_{t_i} is the turbulent heat flux.

The 'closure problem' still exists in the current set of equations, closure requires a model which can approximate the terms in equations 4.29, 4.30 and mean flow variables. The Boussinesq eddy viscosity hypothesis is used in almost all turbulence models and gives the Reynolds stress tensor in terms of the mean velocity gradients through μ_t.

$$\overline{\rho u_i'' u_j''} = \mu_t \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial \tilde{u}_k}{\partial x_k} \right) - \frac{2}{3}\delta_{ij}\bar{\rho}k \tag{4.31}$$

An approximation for the turbulent heat flux tensors q_{t_i} can be found:

$$q_{t_i} = -\kappa_t \frac{\partial \tilde{T}}{\partial x_i} \tag{4.32}$$

where κ_t is the thermal conductivity defined as:

$$\kappa_t = \frac{C_p \mu_t}{Pr_t} \tag{4.33}$$

where Pr_t is the turbulent Prandtl number.

4.1.1.2 *Spalart-Allmaras Turbulence Model*

The S-A one equation turbulence model was chosen for use in the simulations in this work. The S-A model was designed specifically for aerodynamic flows and has good performance for flows around an aerofoil or wing and shows an improvement in predicting separation over the algebraic Baldwin-Lomax model. Despite the S-A model being only a 1-equation model, it gives good predictions for an attached boundary layer and is simpler than the two equation sounterparts. Within TAU the partial derivatives of the S-A have been manually derived for the adjoint which makes it a good fit for the work in this thesis.

$$\begin{aligned}\frac{D\tilde{\nu}}{Dt} = \; & \overbrace{c_{b1}\tilde{S}\tilde{\nu}}^{\text{production}} \quad + \overbrace{\frac{1}{\sigma}\left[\nabla\cdot\left((\nu+\tilde{\nu})\nabla\tilde{\nu}\right)+c_{b2}\left(\nabla\tilde{\nu}\right)\right]}^{\text{diffusion}} \\ & \underbrace{-\,c_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2}_{\text{destruction}} \end{aligned} \tag{4.34}$$

where d is the distance from the point to the wall, ν is the molecular kinematic viscosity ($\nu = \frac{\mu}{\rho}$) and $\tilde{(\nu)}$ is the modified kinematic viscosity. The turbulent eddy viscosity is computed from:

$$\mu_t = \rho\tilde{\nu}f_{\nu 1} \tag{4.35}$$

given

$$f_{\nu 1} = \frac{\chi^3}{\chi^3 + c_{\nu 1}^3} \tag{4.36}$$

$$\chi = \frac{\tilde{\nu}}{\nu} \tag{4.37}$$

The following equation are addition terms required by the SA turbulence model:

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{\nu 2} \tag{4.38}$$

$$f_{\nu 2} = 1 - \frac{\chi}{1 + \chi f_{\nu 1}} \tag{4.39}$$

where S is the magnitude of the vorticity:

$$S = |\Omega| = \sqrt{2\Omega_{ij}\Omega_{ij}} \tag{4.40}$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{4.41}$$

The destruction term requires the wall blockage function to be defined:

$$f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right] \tag{4.42}$$

where g is a limiter function:

$$g = r + c_{w2}(r^6 - r) \tag{4.43}$$

$$r = \min\left[\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right] \tag{4.44}$$

The boundary conditions are:

$$\tilde{\nu}_{wall} = 0, \qquad \tilde{\nu}_{farfield} = [3\nu_\infty, 5\nu_\infty] \tag{4.45}$$

The other terms in the above equations are empirical constants given below:

$$\begin{aligned} \kappa &= 0.41, & c_{b1} &= 0.1355, & c_{b2} &= 0.622, \\ c_{w1} &= \frac{c_{b1}}{\kappa^2} + \frac{c_{b2}+1}{\sigma}, & c_{w2} &= 0.3, & c_{w3} &= 2.0, \\ c_{v1} &= 7.1, & \sigma &= \frac{2}{3} \end{aligned} \tag{4.46}$$

To avoid potential issues when using the SA turbulence model $\tilde{S}$ should not be allowed to become zero or negative. The RANS equations with SA turbulence model closes the set of equations.

4.1.2 *Discretisation and Numerical Schemes*

The CFD solver used for this work is 'TAU' which has been continuously developed by DLR. TAU is an unstructured, density based, finite-volume, Reynolds averaged Navier-Stokes solver. Validation of the TAU code can be found in the work by Gerhold *et al.* [46], Rudnik *et al.* [47], Kroll and Fasbender [48] and Schwamborn *et al.* [49].

The process for generating a solution from a CFD solver starts with the physical definition of the solution domain where a body of interest in the flow is described and any other boundaries are defined (such as the far-field and symmetry planes). The CFD solver uses discretised equations which require a set of points to be solved on and therefore a mesh is created within the computational domain. The governing equations can be discretised in space and time using the Finite Volume Method (FVM) across mesh elements.

4.1.2.1 *Finite Volume Method*

In CFD, the initial step is to descretise the computational domain in to a number of smaller volumes with no overlapping. This is done using the Finite Volume Method (FVM). The descritised Navier-Stokes equations can then be solved at the control volumes in the mesh. For an unstructured grid the control volumes can have a wide range of topologies dependant on the number of vertices belonging to each volume.

For a control volume i, in the computational domain Ω, the Navier-Stokes Equations the finite volume descritisation is:

$$\Omega_i \frac{\partial \mathbf{W}_i}{\partial t} + \mathbf{R}_i(\mathbf{W}_i) \tag{4.47}$$

In equation 4.47 $\mathbf{W}$ are the flow variables and $\mathbf{R}_i$ is the residual, this is the sum of all the fluxes measured across the boundary of each control volume i

$$\mathbf{R}_i = \sum_{j=1}^{nf} (\mathbf{F}_i^c - \mathbf{F}_i^v)\mathbf{n}_i S_i \tag{4.48}$$

where the sum is calculated over all faces of the control volumes in the computational domain, nf.

4.1.2.2 *Flux Discretisation*

TAU gives the options to use either a central or an upwind scheme for the calculation of the inviscid and viscous fluxes. For the simulations in this thesis the central scheme is used with scalar dissipation. The fluxes for the convective term can be written

$$F^c_{ij} = 0.5 \cdot (F^c_i + F^c_j) \cdot n_{ij} - 0.5 \cdot D_{ij} \tag{4.49}$$

where D_{ij} is the artificial dissipation term. The artificial dissipation term is included to help the numerical stability of the scheme.

The Jameson-Schmitt-Turkel (JST) scheme [50] blends 2^{nd} and 4^{th} order dissipation. This is useful as the majority of the flow field uses the 4^{th} order scheme, however this becomes unstable around areas where there are discontinuities in the flow. Therefore the 2_{nd} order scheme is used to deal with discontinuities in the flow, such as shocks.

4.1.2.3 *Evaluation of the Gradient*

To construct the gradient of the flow variables, which is required to calculate the fluxes, a Green-Gauss divergence theorem is used. For a scalar field u the Gauss theorem for the dual cell B_n around point P_n states that

$$\int_{B_n} \nabla u \, dB_n = \int_{\partial B_n} u\mathbf{n} \, d(\partial B_n) \tag{4.50}$$

For each dual grid cell, ∇u is assumed constant, therefore

$$\nabla u = \frac{1}{V_n} \int_{B_n} \nabla u \, dB_n = \frac{1}{V_n} \int_{\partial B_n} u\mathbf{n} \, d(\partial B_n) \tag{4.51}$$

where V_n is the volume of the dual cell B_n around point P_n. Hence, the gradient ∇u_n is found by the following descritised formula:

$$\nabla u_n = \frac{1}{V_n} \sum_{k=2}^{nf} 0.5 \cdot (u_n + u_k) \cdot \mathbf{n}^{dual}_{ij} \tag{4.52}$$

where $\mathbf{n}^{dual}_{ij}$ is the normal to the dual cell face. A representation of the dual cell for P_1 is shown in figure 4.1.

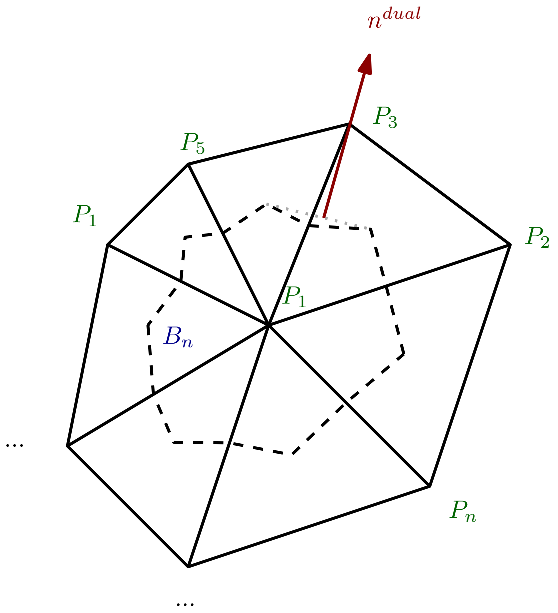


Figure 4.1: Dual cell contructed around point P_1

5

MESH DEFORMATION METHODS

Mesh deformation is important in CFD especially for cases where changes in the geometry require fast responses from changes in the mesh. In an optimisation chain when the topology of a surface is changed the mesh has to accommodate the change. If the domain is re-meshed then the computational time can be large and can create changes in the quantities of interest which are dominated by the discretisation error [51]. Mesh deformation is much faster and more convenient, however it is important that the mesh deformation method is robust enough for the geometry change and will preserve the main characteristics of the mesh.

Two mesh deformation methods are presented here: Linear Elasticity (LE) and Delaunay Graph Mapping (DGM). LE has been used extensively and has been shown to be a robust mesh deformation method [52] compared to other methods (torsion spring analogy and tension spring analogy). DGM was proposed by Liu *et al.* . [53] in which the computational domain is divided into Delaunay elements such that every mesh node resides in only one Delaunay element. The main difference between the two approaches is that the LE is an implicit method and DGM is an explicit method. To represent this mathematically, consider the linear system below

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{5.1}$$

In the case of LE, when solving for $\mathbf{x}$, the above equation requires the calculation of $\mathbf{A}$ and then inverting the matrix. In practice the inverse is never explicitly calculated due to the size of matrix $\mathbf{A}$, an iterative numerical scheme needs to be used to solve for $\mathbf{x}$. DGM gives the $\mathbf{A}^{-1}$ directly, therefore to solve for $\mathbf{x}$ requires only a matrix-vector product. Since DGM is not an iterative method it is an exact solution and since it only relies on a simple matrix-vector product it performs quicker and is easy to code.

5.1 LINEAR ELASTICITY

LE is a robust method for mesh deformation [52], it can handle large deformations while preserving the main mesh characteristics. This mesh deformation technique models the mesh as an elastic solid where a change in geometry will yield the largest deformation in the region close to the changed geometry and a much lesser mesh deformation in regions far away from the deformation. LE has been applied using refinement and de-refinement of the mesh for time-dependent problems when large changes in geometry are considered [54]. Stein *et al.* [55] applied elastic stiffness to the mesh deformation controlled by the inverse proportion to the cell volume. The effect was that refined regions in the mesh (boundary layer, wake, shock, etc.) were not deformed as much as regions where the mesh was less clustered. Yang and Mavriplis

[56] showed that these modifications made the LE method more robust than the tension spring analogy and torsion spring analogy.

The mesh movement scheme is governed by small displacements of an elastic solid body when forces are applied.

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{f} \quad \text{on } \boldsymbol{\Omega} \tag{5.2}$$

where $\mathbf{f}$ is some body force, $\boldsymbol{\sigma}$ is the stress tensor and $\boldsymbol{\Omega}$ is the discretised computational mesh. $\boldsymbol{\sigma}$ is related to the strain tensor $\boldsymbol{\epsilon}$ by

$$\boldsymbol{\sigma} = \lambda \mathrm{Tr}(\boldsymbol{\epsilon})\mathbf{I} + 2\mu\boldsymbol{\epsilon} \tag{5.3}$$

where Tr is the trace of a matrix, $\mathbf{I}$ is the identity matrix and λ and μ are Lamé constants defined as:

$$\lambda = \frac{\nu E}{(1+\nu)+(1-2\nu)} \tag{5.4}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{5.5}$$

where E is Young's modulus and ν is the Poisson ratio. E is strictly greater than zero and can be thought of as the stiffness of the material, increasing E represents a stiffening of the material. Poisson's ratio takes a value $\nu \in (-\frac{1}{2}, \frac{1}{2})$.

The strain tensor is defined as:

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \tag{5.6}$$

For the bounding surfaces of the domain $\partial\Omega$ Derichlet boundary conditions are applied. The governing equation of the LE mesh deformation method (equation 5.2) is discretised using a Galerkin finite element method with piecewise linear shape-functions for unstructured meshes.

5.2 DELAUNAY GRAPH MAPPING

DGM was first proposed by Liu *et al.* [53] as a fast, dynamic and non-iterative mesh deformation scheme. The method decomposes the computational domain into Delaunay elements such that every mesh point exists in only one element. The area/volume ratios are then calculated for each mesh point which are based on a subdivision of the element bounded by the Deluanay vertices. For a 2D deformation the Delaunay elements are triangles and for 3D the elements are tetrahedra. The decomposition of the domain into Delaunay elements is unique as this is a feature of a Delaunay graph. The mapping of the mesh points, which is a linear sum of the ratios and Delaunay element vertices, is a one-to-one mapping and is an explicit formulation.

The first step in this mesh deformation scheme is to identify which points are inside which Delaunay elements. This is done in python using the scipy.spatial module, the module uses an algorithm from the Qhull package called 'qt_findbestfacet'. After all the points have been found the relative area coefficients in 2D and relative volume coefficients in 3D need to be found. The equation for finding the ratios for a point P is as follows:

$$e_i = \frac{\alpha_i}{\alpha} \quad i = 1, 2, 3 \qquad [2D]$$

$$e_i = \frac{V_i}{V} \quad i = 1, 2, 3, 4 \qquad [3D] \tag{5.7}$$

In equation 5.7 the α_i/V_i represent the area/volume of the sub-elements of the Delaunay element in which the point P resides. α,V are the area/volume of the Delaunay element. Figure 5.1 represents the an exploded view of the sub-elements defined by P and the element vertices ABCD of the Delaunay element. In 2D the areas are

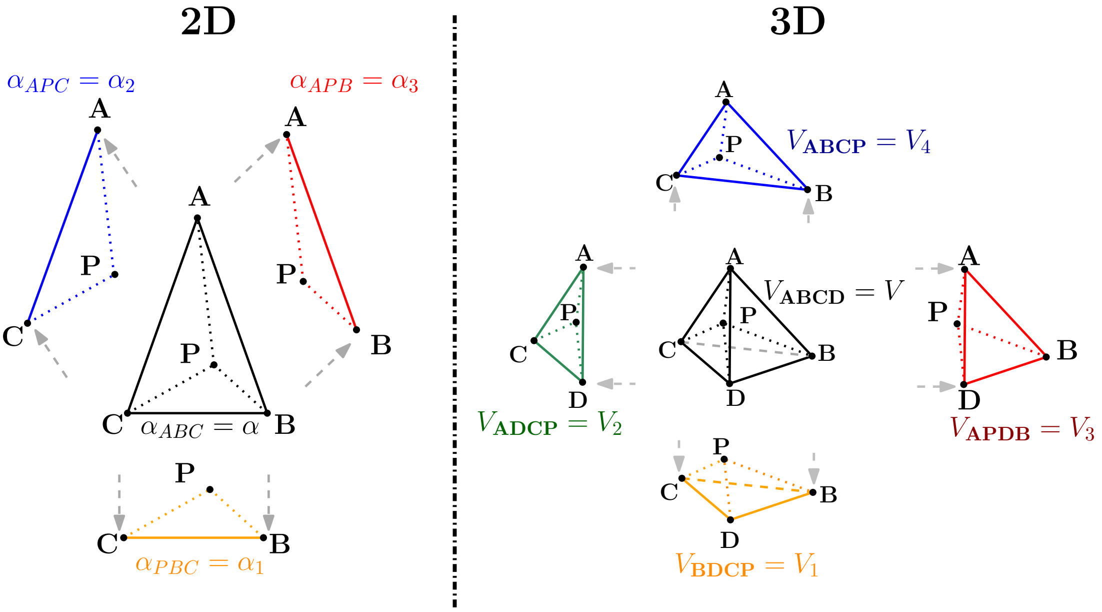


Figure 5.1: Calculation of area/volume ratios in 2D/3D

calculated using the determinant

$$\alpha_1 = \begin{vmatrix} x_P & y_P & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix}, \tag{5.8}$$

$$\alpha_2 = \begin{vmatrix} x_A & y_A & 1 \\ x_P & y_P & 1 \\ x_C & y_C & 1 \end{vmatrix}, \tag{5.9}$$

$$\alpha_3 = \begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_P & y_P & 1 \end{vmatrix}, \tag{5.10}$$

$$\alpha = \begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix}. \tag{5.11}$$

For the volume ratios in 3D a set of 4x4 determinants are used.

$$V_1 = \begin{vmatrix} x_P & y_P & z_P & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}, \tag{5.12}$$

$$V_2 = \begin{vmatrix} x_A & y_A & z_A & 1 \\ x_P & y_P & z_P & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}, \tag{5.13}$$

$$V_3 = \begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_P & y_P & z_P & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}, \tag{5.14}$$

$$V_4 = \begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_P & y_P & z_P & 1 \end{vmatrix}, \tag{5.15}$$

$$V = \begin{vmatrix} x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \\ x_D & y_D & z_D & 1 \end{vmatrix}. \tag{5.16}$$

The relationship between the point P and the Delaunay vertices is shown for only the 3D case, $(x_i, y_i, z_i),\ i = 1, 2, 3, 4$ represent the vertices of the Delaunay tetrahedra $ABCD$.

$$x_P = \sum_{i=1}^{4} e_i x_i \tag{5.17}$$

$$y_p = \sum_{i=1}^{4} e_i y_i \tag{5.18}$$

$$z_p = \sum_{i=1}^{4} e_i z_i \tag{5.19}$$

Moving the mesh now requires the movement of the Delaunay vertices. In the original paper [53] the Delaunay elements spanned from the aerofoil surface to the farfield, Therefore changes in surfaces or the farfield would cause the mesh to be deformed.

DGM has also been used as an overset grid method [57], the overset Delaunay graph is made much coarser than the original mesh. The original mesh points were moved in the usual Delaunay way, however the Delaunay vertices were moved using an implicit method (such as LE). This reduced the time taken to deform the mesh significantly compared to only using an implicit method.

Suppose that the Delaunay vertices have moved to new locations, let these points be denoted as $((x_i', y_i', z_i'),\ i = 1, 2, 3, 4)$. The deformed mesh points will be moved to the new relative position in the Delaunay element based on the following equations.

$$x_P' = \sum_{i=1}^{4} e_i x_i' \tag{5.20}$$

$$y_p' = \sum_{i=1}^{4} e_i y_i' \tag{5.21}$$

$$z_p' = \sum_{i=1}^{4} e_i z_i' \tag{5.22}$$

Notice that the relative volume ratios remain constant. Any further deformations from this position would not require any further calculation of the values of e_i as long as the Delaunay map remains the same.
For some large deformations the Delaunay elements may cross over each other or invert on themselves. This causes problems and the method degenerates as it could cause the mesh to 'fold' over onto itself. When this happens some mesh points may be in more than one Delaunay element which breaks the one-to-one mapping. In

these situations the deformation can be done in more manageable steps by breaking up the geometry movement into smaller movements.

6

OPTIMISATION METHODS

There are two main methods for searching for an optimum: gradient based, and gradient free. Gradient free methods search the entire design space and use only the objective information to guide the search path. Some examples of gradient free methods are Genetic Algorithms (GA), Kriging, and response surface analysis. Gradient free methods are global search methods therefore they will not get 'stuck' in local optima. Due to the number of objective function calculations required gradient free methods are more computationally expensive than gradient based methods.

Gradient based methods are local search methods which use information from the gradient of the objective function with respect to the design variables to define the search path. Since the method is local the final value found from the optimisation may not be a global optimum. In gradient based optimisation the starting point is very important since this will dictate the local region in which the algorithm will search. In a case where the design space has many local optima the final solution is very sensitive to this initial point.

6.1 GRADIENT METHODS

As the name implies, gradient methods require the gradient of a function to find the next point where the function will be evaluated. To find the next function evaluation point there are many methods available, this section will look broadly at three methods: Conjugate gradient, Newton methods and Quasi-Newton methods.

6.1.1 *Steepest Descent*

The steepest descent algorithm is the simplest form of a gradient method and requires information from the objective function and its gradient only. The basic concept of the algorithm is to calculate the gradient at a point x_0, then the algorithm calculates proportional steps of the negative gradient to approach the local minimum.

If $f(x)$ is the objective function, which in continuously differentiable in the region of interest, the steepest descent algorithm can be described mathematically in one dimension as

$$\begin{aligned} x_{k+1} &= x_k - \alpha_k d_k \\ d_k &= f'(x_k) \end{aligned} \tag{6.1}$$

where α_k is the step size. Equation 6.1 iterates through k to minimise $f(x)$ until $f(x_k - \alpha_k d_k) < f(x - \alpha d)$ for all other values of f at x or alternatively when $d_k = 0$. In practice these terminating conditions are assessed within a tolerance.

Figure 6.1(a) shows an example of the steepest descent algorithm to find a minimum of the Rosenbrock function shown in equation 6.2.

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \tag{6.2}$$

The 2D Rosenbrock function has a minimum at $f(1,1) = 0$. The starting location was defined as x_0,y_0= $(2.5, -2.5)$ and the method was run with a constant step size of $\alpha = 0.00057$. The step size was chosen from trial and error to find the largest step size where the method remained stable but gave reasonable convergence.

The algorithm was run for the maximum 100 iterations and the final 'optimal' point was $x, y = (0.08559243, 0.00590243)$ and function evaluation $f(0.08559243, 0.00590243) = 0.837971826703$.

This is some way off the true minimum however as method reached the maximum number of iterations before a minimum was reached. It is clear from figure 6.1(a) that as the method approaches the minimum the step towards the minimum becomes so small that it makes the method quite expensive.

6.1.2 *Newton Methods*

The Newton method is similar in composition to the steepest descent but the search direction is derived from a Taylor series expansion instead of directly using the gradient of the function. This leads to the need to calculate the objective function, the gradient and the Hessian. Let $f(x)$ be a one dimensional function which is continuously differentiable. If x_0 is the initial point of evaluation of f then using a Taylor expansion in the neighbourhood of x_0 gives

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 = q(x) \tag{6.3}$$

The derivative can also be represented using Taylor series:

$$f'(x) \approx q'(x) = f'(x_0) + f''(x_0)(x - x_0) \tag{6.4}$$

When the local minimum is found this gradient is equal to zero:

$$q'(x^*) = f'(x_0) + f''(x_0)(x^* - x_0) = 0 \tag{6.5}$$

Re-arranging equation 6.5 gives a formulation for finding the minimum.

$$x^* = x_0 - \frac{f'(x_0)}{f''(x_0)} \tag{6.6}$$

In pratice the local minimum is rarely reached in 'one-shot', therefore the method is iterated over k. In addition to this, a step size is introduced to increase the stability of the method.

$$x_{k+1} = x_k - \alpha_k d_k \tag{6.7}$$

$$d_k = \frac{f'(x_k)}{f''(x_k)} \tag{6.8}$$

The step-size is often a small positive number, selection of the step-size is an important factor in the stability but also the speed of the method. Choosing a step-size too large could cause instability but a step-size that is too small will give a slow convergence.

For a good choice of step-size the Newton method can show convergence which is super-linear or even quadratic, this is an improvement over the steepest descent method which only has linear convergence.

Figure 6.1(b) shows a comparison of the convergence of the steepest descent algorithm and Newtons method. The objective function is the 2D Rosenbrock function shown in equation 6.2. The Newton method converged after 18 iterations to the 'optimal' point $x^*, y^* = (1.00000006, 1.00000011)$. The objective function at this point is $f(x^*, y^*) = 1.837e^{-12}$ and the step size was kept constant at $\alpha = 0.95$. This is a very good approximation to the analytical minimum and actually converged, unlike the steepest descent method. The paths taken towards the minimum is different but they do approach the same optimal point.

6.1.2.1 *Quasi-Newton Methods*

The Newton method requires at least C^2 continuity, for some functions this is not the case or the calculation of the second derivative is too complicated and time consuming. In these cases the direct computation of the Hessian can be avoided by using an approximation. Quasi-Newton methods are defined as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{B}_k \nabla f(\mathbf{x}_k) \tag{6.9}$$

where B_k is a Hessian-like matrix. The Newton method and steepest descent are special cases of the above equation. When $\mathbf{B}_k = [\nabla^2 f(\mathbf{x}_k)]^{-1}$ the formula is the Newton method and if $B_k = \mathbf{I}$ the formula is the steepest descent algorithm.

There are a variety of methods available to calculate the entries of $\mathbf{B}_k$. Two widely accepted methods are the Davidson-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms. Both methods use the information from the gradient to build a quadratic fit to the function.

The DFP algorithm can be described as

$$\begin{aligned} \mathbf{B}_k &= \mathbf{B}_{k-1} + \mathbf{A}_k + \mathbf{H}_k \\ \mathbf{A}_k &= \frac{(\mathbf{x}_k - \mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{x}_{k-1})^T}{(\mathbf{x}_k - \mathbf{x}_{k-1})^T(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}))} && (6.10) \\ \mathbf{H}_k &= \frac{-\mathbf{B}_{k-1}(\nabla \mathbf{f}(\mathbf{x}_k) - \nabla \mathbf{f}(\mathbf{x}_{k-1}))(\nabla \mathbf{f}(\mathbf{x}_k) - \nabla \mathbf{f}(\mathbf{x}_{k-1}))^T \mathbf{B}_{k-1}}{(\nabla \mathbf{f}(\mathbf{x}_k) - \nabla \mathbf{f}(\mathbf{x}_{k-1}))\mathbf{B}_{k-1}(\nabla \mathbf{f}(\mathbf{x}_k) - \nabla \mathbf{f}(\mathbf{x}_{k-1}))} && (6.11) \end{aligned}$$

The first step of the algorithm ($k = 0$) the value of B_0 is the identity matrix. This is equivalent to a simple gradient search at the first iteration.

After the optimal is found at iteration N the sum of matrices is shown as:

$$\mathbf{B}_N = \mathbf{B}_{N-1} + \mathbf{A}_N + \mathbf{H}_N \tag{6.12}$$

Successively substituting in the previous iterations matrix, $\mathbf{B}_{N-1}$, gives:

$$\mathbf{B}_N = \mathbf{B}_{N-2} + [\mathbf{A}_{N-1} + \mathbf{A}_N] + [\mathbf{H}_{N-1} + \mathbf{H}_N] \tag{6.13}$$

$$\mathbf{B}_N = \mathbf{B}_{N-3} + [\mathbf{A}_{N-2} + \mathbf{A}_{N-1} + \mathbf{A}_N] + [\mathbf{H}_{N-2} + \mathbf{H}_{N-1} + \mathbf{H}_N] \tag{6.14}$$

$$\mathbf{B}_N = \mathbf{B}_0 + \sum_{k=0}^{N} \mathbf{A}_k + \sum_{k=0}^{N} \mathbf{H}_k \tag{6.15}$$

By construction of 6.10 and 6.11:

$$\sum_{k=0}^{N} \mathbf{A}_k = -[\nabla^2 f(\mathbf{x_N})]^{-1} \tag{6.16}$$

$$\sum_{k=0}^{N} \mathbf{H}_k = -\mathbf{I} = -\mathbf{B}_0 \tag{6.17}$$

Therefore, $\mathbf{B}_N = -[\nabla^2 f(\mathbf{x_N})]^{-1}$.

The BFGS algorithm is an adaptation of the DFP algorithm which instead of calculating the Hessian it calculates the inverse. The BFGS algorithm has shown to have better convergence than DFP. The matrix up-date formula is as follows:

$$\mathbf{B}_{k+1} = \left(\mathbf{I} - \frac{\Delta\mathbf{x}_k \nabla\mathbf{f}_k^T}{\nabla\mathbf{f}_k^T \Delta\mathbf{x}_k}\right) \mathbf{B}_k \left(\mathbf{I} - \frac{\nabla\mathbf{f}_k \Delta\mathbf{x}_k^T}{\nabla\mathbf{f}_k^T \Delta\mathbf{x}_k}\right) + \frac{\Delta\mathbf{x}_k \Delta\mathbf{x}_k^T}{\nabla\mathbf{f}_k^T \Delta\mathbf{x}_k} \tag{6.18}$$

$$\Delta\mathbf{x}_k = (\mathbf{x}_k - \mathbf{x}_{k-1}) \tag{6.19}$$

$$\nabla\mathbf{f}_k = \nabla\mathbf{f}(\mathbf{x}_k) - \nabla\mathbf{f}(\mathbf{x}_{k-1}) \tag{6.20}$$

Quasi-Newton methods, in general, have a slower convergence rate than Newton methods but are still faster than the linear convergence of the steepest descent algorithm. For the Newton method the calculation of the Hessian can add to the complexity of the code and is heavily dependent on the objective function. Quasi-Newton methods have a more multi-purpose framework and reduced computational complexity.

Figure 6.1(c) shows the optimisation on the Rosenbrock function of the Quasi-Newton method BFGS from the scipy module in python compared with Newtons' method and the steepest descent algorithm. This algorithm took 24 iterations to converge and the 'optimal' point found is $x^*, y^* = (1.000001, 1.000002)$ with objective evaluation $f(1.000001, 1.000002) \approx 0.0$. All three methods take different paths towards the same minimum, however Newtons' method shows the fastest convergence.

6.1.3 *Conjugate Gradient*

The conjugate gradient method is a popular method in unconstrained optimisation, especially when the number of design variables is large. The conjugate gradient method was original created as an algorithm for solving linear systems of equations. It was later suggested that it could be used as an iterative method, this method proved successful and gave rise to a subset of iterative methods called Krylov subspace methods.

Consider the problem

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \qquad \mathbf{A} \in \mathbb{R}^{n x n}, \qquad \mathbf{x}, \mathbf{b} \in \mathbb{R}^{n} \tag{6.21}$$

For an initial starting point x_0

$$g_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 \tag{6.22}$$

Let

$$\phi(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T \mathbf{A}\mathbf{y} - \mathbf{y}^T \mathbf{b} \tag{6.23}$$

then the minimum is achieved at point x^*

$$\nabla\phi(x^*) = \mathbf{A}\mathbf{x}^* - \mathbf{b} = 0 \tag{6.24}$$

The conjugate gradient method is essentially minimising ϕ over a subset contained in $\mathbb{R}^n$.

For the initial step select g_0 to be the steepest descent direction, $g_0 = d_0 = -\nabla f(\mathbf{x}_0)$. The steps of the method are as follows:

1) $\alpha_k = \dfrac{\mathbf{d}_k^T \mathbf{d}_k}{\mathbf{g}_k^T \mathbf{A}\mathbf{g}_k}$, *[Compute step length]*
2) $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{g}_k$, *[Evaluate approximate solution]*
3) $\mathbf{g}_{k+1} = \mathbf{g}_k - \alpha_k \mathbf{A}\mathbf{g}_k$, *[Compute the residual]*
4) $\beta_k = \dfrac{\mathbf{d}_{k+1}^T \mathbf{d}_{k+1}}{\mathbf{d}_k^T \mathbf{d}_k}$, *[Determine the improvement this step]*
5) $\mathbf{g}_{k+1} = \mathbf{d}_{k+1} + \beta_k \mathbf{g}_k$ *[Evaluate next search direction]*

In the method above the value of β_k is based on Fletcher-Reeves work. There are other well known formulas, two are shown below.

$$\text{Polak} - \text{Ribiere}: \qquad \beta_k^{(PR)} = \frac{(\mathbf{d}_{k+1})^T[\mathbf{d}_{k+1} - \mathbf{d}_k]}{\mathbf{d}_k^T \mathbf{d}_k} \tag{6.25}$$

$$\text{Hestenes} - \text{Stiefel}: \qquad \beta_k^{(HS)} = \frac{(\mathbf{d}_{k+1})^T[\mathbf{d}_{k+1} - \mathbf{d}_k]}{(\mathbf{g}_k)^T[\mathbf{d}_{k+1} - \mathbf{d}_k]} \tag{6.26}$$

The performance of the conjugate gradient method is dependant on the condition number $\kappa_2(\mathbf{A})$, it is likely that an exact value cannot be calculated. To ensure fast convergence it is necessary to use some preconditioning of the conjugate gradient method. The general idea of preconditioning is to replace the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with a symmetric positive definite system with the same solution. The preconditioned conjugate gradient method is the same as applying the regular conjugate gradient method to the following system with symmetric positive definite matrix $\mathbf{S}$:

$$\mathbf{S}\mathbf{A}\mathbf{S}\mathbf{y} = \mathbf{S}\mathbf{b} \tag{6.27}$$

$$\textit{where} \qquad \mathbf{x} = \mathbf{S}\mathbf{y} \tag{6.28}$$

Letting $\mathbf{M} = \mathbf{S}^2$, the initial step of the preconditioned conjugate gradient has $\mathbf{d}_0 = -\nabla f(\mathbf{x}_0)$, $\mathbf{z}_0 = \mathbf{M}\mathbf{d}_0$ and $\mathbf{g}_0 = \mathbf{z}_0$. The iterative steps are computed as follows:

1) $\alpha_k = \dfrac{\mathbf{z}_k^T \mathbf{d}_k}{\mathbf{g}_k^T \mathbf{A} \mathbf{g}_k}$, [Compute step length]

2) $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{g}_k$, [Evaluate approximate solution]

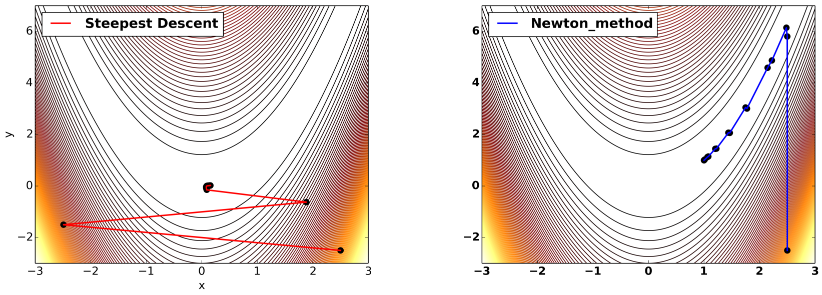
3) $\mathbf{g}_{k+1} = \mathbf{g}_k - \alpha_k \mathbf{A} \mathbf{g}_k$, [Compute the residual]

4) $\mathbf{z}_{k+1} = \mathbf{M} \mathbf{g}_{k+1}$, [Compute preconditioning step]

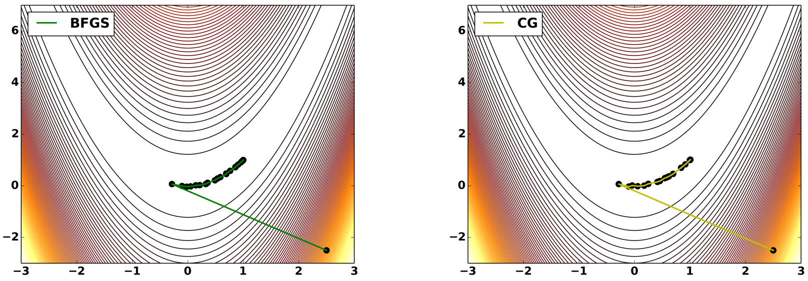
5) $\beta_k = \dfrac{\mathbf{z}_{k+1}^T \mathbf{d}_{k+1}}{\mathbf{z}_k^T \mathbf{d}_k}$, [Determine the improvement this step]

6) $\mathbf{g}_{k+1} = \mathbf{z}_{k+1} + \beta_k \mathbf{g}_k$ [Evaluate next search direction]

The matrix $\mathbf{M}$ is also symmetric positive definite by construction, $\mathbf{M} = \mathbf{S}^2 = (\mathbf{Q}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{Q}^T)(\mathbf{Q}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{Q}^T)$. $\mathbf{M}$ stays constant throughout the iterative process.



(a) Steepest Descent (b) Newtons Method



(c) BFGS (d) Conjugate Gradient

Figure 6.1: Several examples of different search paths for four gradient based optimisation methods on the Rosenbrock function.

Figure 6.1(d) shows the optimisation of the conjugate gradient algorithm (from the scipy optimisation toolbox) compared to the BFGS, Newtons' method and the steepest descent algorithm. The BFGS and CG algorithms both take very similar paths to the minimum, the search directions overlap frequently. The CG method took 22 iterations and found the optimal point $x^*, y^* = (1.0, 1.0)$. The optimal point is correct to 6 decimal places. the objective function evaluation is $f(1.0, 1.0) \approx 0.0$.

6.2 GRADIENT CALCULATION

This thesis focuses on using a fully turbulent viscous flow, therefore the calculation of each objective function dominates the expense of the optimisation chain. When this is the case, gradient based methods are the most suitable. There are many different ways of finding the gradient of the objective function, an excellent resource is Alan Le Moigne's thesis [58] or Alonso [59]. In this section several methods will be reviewed: Finite Difference (FD), complex step, analytical methods, automatic differentiation and the adjoint method.

An optimisation is often posed as a minimisation or maximisation of a function subject to constraints:

$$\begin{array}{lll} \text{min} & I(D_i) & \\ \text{w.r.t.} & D_i, & i = 1, 2, ..., n \\ \text{subject to} & C_j(D_i) \geqslant 0, & j = 1, 2, ..., m \end{array} \tag{6.29}$$

where I is the objective function and $\mathbf{D}$ is the vector of design variables. Gradient based methods require the calculation of the derivative of the objective function $\frac{dI}{dD_i}$, $(n \times 1)$, and the derivative of the constraints $\frac{dC_j}{dD_i}$, $(m \times n)$.

6.2.1 *Finite Difference*

Of the methods being reviewed the finite difference is by far the most simple to use and implement. A finite difference comes in several different formulations: forward, backward and central. The complexity can also be included by increasing the order of the formulation to achieve a more accurate solution at a greater cost.

The forward difference will be derived first. Taking a Taylor series of the function f around a point x in one dimension with a small step size h gives the following:

$$\begin{aligned} f(x+h) &= f(x) + \sum_{i=1}^{n} \frac{h^i}{i!} f^{(i)}(x) \\ &= f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(x) + ... \end{aligned} \tag{6.30}$$

Re-arranging equation 6.30 to make $f'(x)$ the subject of the equation gives:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h) \tag{6.31}$$

Equation 6.31 gives the first order forward difference for finding the first derivative of the function f. The truncation error is denoted as $\mathcal{O}(h)$

Similarly, for the backward difference consider the Taylor series:

$$\begin{aligned} f(x-h) &= f(x) + \sum_{i=1}^{n} \frac{(-h)^i}{i!} f^{(i)}(x) \\ &= f(x) - hf'(x) + \frac{h^2}{2!} f''(x) - \frac{h^3}{3!} f'''(x) + ... \end{aligned} \tag{6.32}$$

As for the forward difference, re-arranging equation 6.32 to make $f'(x)$ the subject of the equation gives:

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h) \tag{6.33}$$

Equation 6.33 is the first order backward finite difference with truncation error $\mathcal{O}(h)$.

For a higher order difference consider the subtraction of equation 6.32 from equation 6.30. This gives:

$$\begin{aligned} f(x+h) - f(x-h) &= \sum_{i=1}^{n} \frac{h^i}{i!} f^{(i)}(x) - \sum_{i=1}^{n} \frac{(-h)^i}{i!} f^{(i)}(x) \\ &= 2hf'(x) + 2\frac{h^3}{3!} f'''(x) + \ldots \end{aligned} \tag{6.34}$$

Dividing equation 6.34 by $2h$ gives:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2) \tag{6.35}$$

Equation 6.35 is the second order central difference scheme with truncation error of $\mathcal{O}(h^2)$. The forward and backward differences require two function evaluations, namely $f(x)$ and $f(x \pm h)$. The central difference requires three function evaluations: $f(x)$, $f(x+h)$, and $f(x-h)$. Despite $f(x)$ not featuring the equation 6.35 it still needs calculating to get the objective information at x. The extra computations required give an increased accuracy since the truncation term is smaller compared to the forward and backward differences (i.e. $\mathcal{O}(h) > \mathcal{O}(h^2)$).

Higher order finite differences can also be calculated by increasing the number of terms from the Taylor series used, however this always increases the number of function evaluations required. Finite differences can also be used to find higher order derivatives. For example:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^4) \tag{6.36}$$

Equation 6.36 is the sum of equation 6.30 and 6.32.

To relate this back to the optimisation problem shown in equation 6.29, using the forward difference as an example:

$$\frac{dI}{dD_i} = \frac{I(D_i + \epsilon) - I(D_i)}{\epsilon}, \qquad i = 1, 2, \ldots., n \tag{6.37}$$

where ϵ is a small perturbation in the design variable D_i. As the number of design variables increase the number of function evaluations are required and therefore this step will become a bottleneck of the optimisation process. The number of necessary function evaluations for forward and backward difference methods is $N_{DV} + 1$, the number of function evaluation for the central difference is $2N_{DV} + 1$.

Th choice of step size is critical and is particularly difficult for large sets of design variables. It has also been indicated that a link between the convergence of the flow solution may have an influence on the step size [60, 61, 62]. Examples of the use of finite differences for aerodynamic optimisation can be found in references [63, 64]. If the step size is too large much of the gradient information can be lost and the truncation error will be larger. When the step size is too small errors in the evaluation of the function can be amplified giving incorrect gradients. Often an analysis of gradient accuracy on step size will need to be undertaken increasing the set-up time of the optimisation chain.

6.2.2 *Complex Step*

The complex step method is very similar in set-up to the finite difference method, however in this case the step size involves a complex element. The original work for the complex step method was shown in the paper by Lyness and Moler [65] and Lyness [66] but it was not until Squire and Trapp [67] that the method became prominent in the literature.

Starting with a Taylor series of function f around a point x with step size ih gives:

$$\begin{aligned} f(x+ih) &= f(x) + \sum_{j=1}^{n} \frac{(ih)^j}{j!} f^{(j)}(x) \\ &= f(x) + ihf'(x) - \frac{h^2}{2!} f''(x) - \frac{ih^3}{3!} f'''(x) + \dots \end{aligned} \tag{6.38}$$

Taking the imaginary parts of 6.38 and re-arranging gives:

$$f'(x) = \frac{\mathrm{Im}[f(x+ih)]}{h} + h^2 \frac{f'''(x)}{3!} + \mathcal{O}(h^4) \tag{6.39}$$

Also, for the real part:

$$f(x) = \mathrm{Re}[f(x+ih)] + h^2 \frac{f''(x)}{2!} + \mathcal{O}(h^4) \tag{6.40}$$

Equations 6.39 and 6.40 show that the values of $f(x)$ and $f'(x)$ can be approximated to within $\mathcal{O}(h^2)$. In this instance h needs to be chosen such that the term $\mathcal{O}(h^2)$ such that it is within or close to machine accuracy. Introducing l as a measure of precision in the system then:

$$h^2 \left| \frac{f''(x)}{2!} \right| < l|f(x)| \tag{6.41}$$

$$h^2 \left| \frac{f'''(x)}{3!} \right| < l|f'(x)| \tag{6.42}$$

When the inequalities 6.41 and 6.42 are satisfied through the choice of h then the truncation error $\mathcal{O}(h^2)$ will be less than machine accuracy. Since the left hand sides in the inequalities are both positive they will become harder to satisfy as $f'(x) \to 0$ or $f(x) \to 0$. An example of using the complex step can be found in [68]. In contrast to the finite difference method, the complex step method is insensitive to the choice of the step size.

Despite the increase in stability of the complex step method, it still requires $N_{DV} + 1$ flow solutions to be computed. Since the method uses the imaginary part of the objective function then these will need to be stored when the flow solution is calculated increasing the memory necessary to run the flow solver.

6.2.3 *Automatic Differentiation*

The method of automatic differentiation involves applying the chain rule to each output from a code with respect to the inputs in forward mode or the derivative of the inputs with respect to the outputs in reverse mode. Automatic differentiation takes advantage of the structure of coding by acting directly on individual arithmetic operations $(\div, \times, +, -)$ and intrinsic functions $(\sin, \cos, \exp, ...)$.

Consider a function with four of the arithmetic terms outlined in the previous paragraph. In general the final derivative for the forward mode is:

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial w_1}\left(\frac{\partial w_1}{\partial w_2}\left(\frac{\partial w_2}{\partial w_3}\left(\frac{\partial w_3}{\partial w_4}\frac{\partial w_4}{\partial x}\right)\right)\right) \tag{6.43}$$

In reverse mode:

$$\begin{aligned}\frac{\partial y}{\partial x} &= \frac{\partial y}{\partial w_1}\frac{\partial w_1}{\partial x} = \left(\frac{\partial y}{\partial w_2}\frac{\partial w_2}{\partial w_1}\right)\frac{\partial w_1}{\partial x} \\ &= \left(\left(\frac{\partial y}{\partial w_3}\frac{\partial w_3}{\partial w_2}\right)\frac{\partial w_2}{\partial w_1}\right)\frac{\partial w_1}{\partial x} \\ &= \left(\left(\left(\frac{\partial y}{\partial w_4}\frac{\partial w_4}{\partial w_3}\right)\frac{\partial w_3}{\partial w_2}\right)\frac{\partial w_2}{\partial w_1}\right)\frac{\partial w_1}{\partial x}\end{aligned} \tag{6.44}$$

Forward mode is the most intuitive as it follows the order of calculation of variables. The best way to show how automatic differentiation works is through an example, consider:

$$\begin{aligned}f(x_1, x_2) &= x_1 x_2 + e^{\sin(x_1)} \\ &= w_1 w_2 + e^{\sin(w_1)} \\ &= w_3 + e^{w_4} \\ &= w_3 + w_5 \\ &= w_6\end{aligned} \tag{6.45}$$

Equation 6.45 shows the substitution of variables and arithmetic operations. Suppose that the derivative $\frac{\partial f}{\partial x_1}$ is to be calculated then the derivative at each step is shown in table 6.1.

The function f has two variables, table 6.1 gives the process for x_1 and a similar process can be done for x_2 to complete the gradient information for f.

Table 6.1: Table of forward automatic differentiation values

Substitution	Derivative w.r.t x_1
$w_1 = x_1$	$\frac{dw_1}{dx_1} = 1$ (seed)
$w_2 = x_2$	$\frac{dw_2}{dx_1} = 0$ (seed)
$w_3 = w_1 w_2$	$\frac{dw_3}{dx_1} = w_2 \frac{dw_1}{dx_1} + \frac{dw_2}{dx_1} w_1$
$w_4 = \sin(w_1)$	$\frac{dw_4}{dx_1} = \frac{dw_1}{dx_1} \cos(w_1)$
$w_5 = e^{w_4}$	$\frac{dw_5}{dx_1} = \frac{dw_4}{dx1} e^{w_4}$
$w_6 = w_3 + w_5$	$\frac{dw_6}{dx_1} = \frac{dw_3}{dx_1} + \frac{dw_5}{dx_1}$

In reverse mode the table is shown in table 6.2 where $\hat{w} = \frac{\partial f}{\partial w}$.

Table 6.2: Table of backward automatic differentiation values

Substitution	Derivative
$w_6 = w_3 + w_5$	$\hat{w}_6 = 1$ (seed)
$w_5 = e^{w_4}$	$\hat{w}_5 = \hat{w}_6 \cdot 1$
$w_4 = \sin(w_1)$	$\hat{w}_4 = \hat{w}_5 \cdot e^{w_4}$
$w_3 = w_1 w_2$	$\hat{w}_3 = \hat{w}_5 \cdot 1$
$w_2 = x_2$	$\hat{w}_2 = \hat{w}_3 \cdot w_1$
$w_1 = x_1$	$\hat{w}_1 = \hat{w}_3 w_2 + \hat{w}_4 \cos(w_1)$

Table 6.2 shows the derivatives used in the reverse mode of automatic differentiation. $\hat{w}_2$ and $\hat{w}_1$ represent the derivatives of x_2 and x_1 respectively. More information on automatic differentiation can be found in [69] and [70].

Although automatic differentiation is very powerful it requires direct interactions with the solver which may include access to the source code. Some codes for the implementation of automatic differentiation method are ADIFOR, TAMC, DAFOR, GRESS, Odysse and PADRE2.

6.2.4 *Analytical Differentiation*

This method, also known as direct differentiation, is the most direct method of calculating the gradient and extensive derivations can be found in the literature [58, 69, 51]. The objective function dependencies are defined as:

$$I = I(\mathbf{W}(\mathbf{D}), \mathbf{X}(\mathbf{D}), \mathbf{D}) \tag{6.46}$$

Taking the derivative of the objective function with respect to $\mathbf{D}$ gives:

$$\frac{dI}{d\mathbf{D}} = \left[\frac{\partial I}{\partial \mathbf{W}}\right]\frac{d\mathbf{W}}{d\mathbf{D}} + \left[\frac{\partial I}{\partial \mathbf{X}}\right]\frac{d\mathbf{X}}{d\mathbf{D}} + \frac{\partial I}{\partial \mathbf{D}} \tag{6.47}$$

The partial derivatives in the above equation are reasonably simple, they depend on the explicit dependencies of the objective function. The total derivative terms are a little more difficult. The $\frac{d\mathbf{W}}{d\mathbf{D}}$ term is expensive to solve as it requires an evaluation of the gradient for each flow variable for a small change in each design variable. The mesh sensitivity term $\frac{d\mathbf{X}}{d\mathbf{D}}$ is easier to solve by an analytical method, if the meshing algorithm is known, or through finite differences.

The term $\frac{d\mathbf{W}}{d\mathbf{D}}$ can be calculating by differentiating the residual of the flow equation. The residual of the flow equation and dependencies is shown below:

$$\mathbf{R}(\mathbf{W}(\mathbf{D}), \mathbf{X}(\mathbf{D}), \mathbf{D}) = 0 \tag{6.48}$$

Taking derivatives of the residual with respect to the design variables gives:

$$\frac{d\mathbf{R}}{d\mathbf{D}} = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right]\frac{d\mathbf{W}}{d\mathbf{D}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]\frac{d\mathbf{X}}{d\mathbf{D}} + \frac{\partial \mathbf{R}}{\partial \mathbf{D}} \tag{6.49}$$

re-arranging gives the following equation:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right]\frac{d\mathbf{W}}{d\mathbf{D}} = -\left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]\frac{d\mathbf{X}}{d\mathbf{D}} - \frac{\partial \mathbf{R}}{\partial \mathbf{D}} \tag{6.50}$$

When the design variables are purely geometric parameters then there is no explicit dependence of $\mathbf{R}$ on $\mathbf{D}$, therefore $\frac{\partial \mathbf{R}}{\partial \mathbf{D}} = 0$. For only aerodynamic design variables, i.e. Mach number or angle of attack, there is no explicit dependence of $\mathbf{X}$ on $\mathbf{D}$, hence $\frac{\partial \mathbf{X}}{\partial \mathbf{D}} = 0$. $\frac{\partial \mathbf{R}}{\partial \mathbf{W}}$ is the Jacobian of the flow residuals which is theoretically calculated in full in the flow solver. In practice this Jacobian is not calculated exactly and an approximation is found. The analytical differentiation method requires the exact Jacobian, however the approximate can also be used to find the sensitivity of the flow variables to the design variables. The Jacobian will only need to be calculated once as it is unchanged for each design variable, this means in the above equation only the right hand side changes.

The uses of analytical differentiation are limited due to the approximated Jacobian matrix. When an unstructured mesh is used this differentiation method can only be used with Euler equations [71, 72] whereas the method is easy to apply when a structured mesh is used. Some examples of this differentiation method in viscous flows are [73, 74] and when the flow is turbulent [75, 76].

6.3 THE DISCRETE ADJOINT FORMULATION

Adjoint methods are derived from control theory, this was first proposed by Lions [77] and continued by Pironneau [78] in Stokes' flow. Jameson [79] successfully developed this methodology for use with the Euler flow equation.

6.3.1 *Flow Adjoint*

Consider the objective as a function of flow variable $\mathbf{W}$, grid $\mathbf{X}$ and design variables $\mathbf{D}$:

$$I = I(\mathbf{W}(\mathbf{D}), \mathbf{X}(\mathbf{D}), \mathbf{D}) \tag{6.51}$$

For a steady state flow, the vector of flow residuals ($\mathbf{R}(\mathbf{W}, \mathbf{X}, \mathbf{D})$) is zero. One method to derive adjoint methods is to multiply the flow residual vector by a Lagrangian multiplier $\boldsymbol{\lambda}$, and add it to the objective function, Eq (6.51).

$$\mathcal{L}(\mathbf{W}, \mathbf{X}, \mathbf{D}, \boldsymbol{\lambda}) = I(\mathbf{W}, \mathbf{X}, \mathbf{D}) + \boldsymbol{\lambda} \cdot \mathbf{R}(\mathbf{W}, \mathbf{X}, \mathbf{D}) \qquad \forall \boldsymbol{\lambda} \tag{6.52}$$

Taking the derivative of equation (6.52) with respect to the design variables $\mathbf{D}$ gives:

$$\frac{d\mathcal{L}}{d\mathbf{D}} = \left(\frac{\partial I}{\partial \mathbf{W}}\right)\frac{d\mathbf{W}}{d\mathbf{D}} + \left(\frac{\partial I}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{d\mathbf{D}} + \frac{\partial I}{\partial \mathbf{D}} + \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\frac{d\mathbf{W}}{d\mathbf{D}} + \frac{\partial \mathbf{R}}{\partial \mathbf{X}}\frac{d\mathbf{X}}{d\mathbf{D}} + \frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right) \tag{6.53}$$

Equation (6.53) can be re-arranged as:

$$\frac{d\mathcal{L}}{d\mathbf{D}} = \left(\frac{\partial I}{\partial \mathbf{D}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right) + \left(\frac{\partial I}{\partial \mathbf{W}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)\frac{d\mathbf{W}}{d\mathbf{D}} + \left(\frac{\partial I}{\partial \mathbf{X}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{d\mathbf{D}} \tag{6.54}$$

In equation (6.54) the calculation of $\frac{d\mathbf{W}}{d\mathbf{D}}$ is computationally prohibitive. Since the choice of $\boldsymbol{\lambda}$ is arbitrary, equation (6.54) can be simplified by choosing $\boldsymbol{\lambda}$ to satisfy:

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)^T \boldsymbol{\lambda} = -\left(\frac{\partial I}{\partial \mathbf{W}}\right)^T \tag{6.55}$$

Equation (6.55) is known as the adjoint equation. $\frac{\partial \mathbf{R}}{\partial \mathbf{W}}$ represents the exact Jacobian of the flow field. With this choice of $\boldsymbol{\lambda}$ the expensive term $\frac{d\mathbf{W}}{d\mathbf{D}}$ no longer needs to be calculated.

$$\frac{d\mathcal{L}}{d\mathbf{D}} = \frac{dI}{d\mathbf{D}} = \left(\frac{\partial I}{\partial \mathbf{D}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right) + \left(\frac{\partial I}{\partial \mathbf{X}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{d\mathbf{D}}, \qquad \forall \boldsymbol{\lambda}, \mathbf{D} \tag{6.56}$$

Equation (6.56) can be further simplified when the design variables describe purely geometric changes such that there is no explicit dependence of I and $\mathbf{R}$ on $\mathbf{D}$. (i.e. $\frac{\partial I}{\partial \mathbf{D}} = 0$ and $\frac{\partial \mathbf{R}}{\partial \mathbf{D}} = 0$).

$$\frac{d\mathcal{L}}{d\mathbf{D}} = \frac{dI}{d\mathbf{D}} = \left(\frac{\partial I}{\partial \mathbf{X}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{d\mathbf{D}}, \qquad \forall \boldsymbol{\lambda}, \mathbf{D} \tag{6.57}$$

This only requires to solve the adjoint equation once to obtain the sensitivity derivatives for each objective function. The calculation of the sensitivity derivatives of the objective function is decoupled from the design variables. The grid sensitivities $\frac{d\mathbf{X}}{d\mathbf{D}}$ could be calculated analytically if it is possible or through another method such as finite difference. For a complex grid, especially unstructured grid, the analytical grid sensitivity is difficult to obtain. Therefore, it could be calculated using finite differences, which are:

$$\begin{aligned}\left(\frac{\partial I}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{dD_k} &\approx \frac{I(\mathbf{W},\mathbf{X}(D_k+\epsilon),D_k)-I(\mathbf{W},\mathbf{X}(D_k),D_k)}{\epsilon}\\ \left(\frac{\partial R}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{dD_k} &\approx \frac{R(\mathbf{W},\mathbf{X}(D_k+\epsilon),D_k)-R(\mathbf{W},\mathbf{X}(D_k),D_k)}{\epsilon}\end{aligned} \tag{6.58}$$

The above stated formulas are discrete adjoint formulae since the flow governing equation is discretised before it is differentiated. In this work, the discrete adjoint formula is solved using DLR TAU solver [46, 80].

6.3.2 *Mesh Adjoint*

As presented in above subsection, the grid sensitivities $\frac{d\mathbf{X}}{d\mathbf{D}}$ could be calculated by finite difference or analytically. Le Moigne [58] shows the grid sensitivities for an algebraic structured mesh deformation. The analytical solution of entire grid sensitivities would be difficult (or impossible) to obtain for an unstructured mesh deformation algorithm such as spring analogy, therefore finite differences could be used instead.

To perform the finite-difference, the number of calculations required is the number of design variables (N_{DV}) times the size of volume mesh. For a large N_{DV} and/or a large mesh the finite difference form a bottleneck in the optimisation procedure. The mesh deformation could also be time consuming. An additional issue, which is the same with any finite difference approach, is that it is hard to determine the perturbation step size to obtain accurate sensitivities. Nielsen and Park [14] proposed the introduction of another adjoint equation, the mesh adjoint equation, to eliminate the need to calculate the grid sensitivities. In this approach, the objective function shown in equation (6.51) will be minimised subject to:

$$\begin{aligned}\mathbf{R}(\mathbf{W},\mathbf{X},\mathbf{D}) &= 0\\ \mathbf{T}(\mathbf{X},\mathbf{D}) &= 0\end{aligned} \tag{6.59}$$

where $\mathbf{T}$ is the residual of the mesh deformation method. Then the two residual functions are added into the objective function with two adjoint operators $\boldsymbol{\lambda}_{flow}$ and $\boldsymbol{\lambda}_{mesh}$ for flow and mesh variable vectors respectively:

$$\mathcal{L} = I(\mathbf{W},\mathbf{X},\mathbf{D}) + \boldsymbol{\lambda}_{flow}\cdot\mathbf{R}(\mathbf{W},\mathbf{X},\mathbf{D}) + \boldsymbol{\lambda}_{mesh}\cdot\mathbf{T}(\mathbf{X},\mathbf{D}) \tag{6.60}$$

Taking the derivative of Equation (6.60) w.r.t. the design variables ($\mathbf{D}$) and rearranging gives:

$$\begin{aligned}\frac{d\mathcal{L}}{d\mathbf{D}} = &\left(\frac{\partial I}{\partial \mathbf{D}} + \boldsymbol{\lambda}_{flow}^T\frac{\partial \mathbf{R}}{\partial \mathbf{D}} + \boldsymbol{\lambda}_{mesh}^T\frac{\partial \mathbf{T}}{\partial \mathbf{D}}\right) + \left(\frac{\partial I}{\partial \mathbf{W}} + \boldsymbol{\lambda}_{flow}^T\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)\frac{d\mathbf{W}}{d\mathbf{D}}\\ &+\left(\frac{\partial I}{\partial \mathbf{X}} + \boldsymbol{\lambda}_{flow}^T\frac{\partial \mathbf{R}}{\partial \mathbf{X}} + \boldsymbol{\lambda}_{mesh}^T\frac{\partial \mathbf{T}}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{d\mathbf{D}}\end{aligned} \tag{6.61}$$

As in the previous subsection the choice of the Lagrangian variables is arbitrary. Therefore, it is appropriate to choose values for $\boldsymbol{\lambda}_{flow}$ and $\boldsymbol{\lambda}_{mesh}$ such that the most expensive terms in equation (6.61) do not need to be calculated. As before the most expensive terms are from the derivative $\frac{d\mathbf{W}}{d\mathbf{D}}$. For cases where there are a large number of design variables and/or a large computational grid (in terms of nodes/cells) the term $\frac{d\mathbf{X}}{d\mathbf{D}}$ will also become computationally prohibitive. To eliminate these terms the following linear systems of equations need to be solved:

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right)^T \boldsymbol{\lambda}_{flow} = -\left(\frac{\partial I}{\partial \mathbf{W}}\right)^T \tag{6.62}$$

$$\left(\frac{\partial \mathbf{T}}{\partial \mathbf{X}}\right)^T \boldsymbol{\lambda}_{mesh} = -\left(\frac{\partial I}{\partial \mathbf{X}}\right)^T - \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \boldsymbol{\lambda}_{flow} \tag{6.63}$$

where equation (6.62) is solved first. Equation (6.62) is the flow adjoint equation (as in equation (6.55)) and equation (6.63) is the mesh adjoint equation. When the flow and mesh adjoint equations are satisfied, the gradients of the objective function with respect to the design variables can be found using the following:

$$\frac{d\mathcal{L}}{d\mathbf{D}} = \frac{dI}{d\mathbf{D}} = \left(\frac{\partial I}{\partial \mathbf{D}} + \boldsymbol{\lambda}_{flow}^T \frac{\partial \mathbf{R}}{\partial \mathbf{D}} + \boldsymbol{\lambda}_{mesh}^T \frac{\partial \mathbf{T}}{\partial \mathbf{D}}\right), \qquad \forall \boldsymbol{\lambda}_{flow,mesh}, \mathbf{D} \tag{6.64}$$

Furthermore, for purely geometric changes, the terms $\frac{\partial I}{\partial \mathbf{D}}$ and $\frac{\partial \mathbf{R}}{\partial \mathbf{D}}$ are zero. This gives a much simpler derivative:

$$\frac{d\mathcal{L}}{d\mathbf{D}} = \frac{dI}{d\mathbf{D}} = \boldsymbol{\lambda}_{mesh}^T \frac{\partial \mathbf{T}}{\partial \mathbf{D}} \tag{6.65}$$

In this instance the derivatives are dependent entirely on the mesh adjoint operator derived in equations (6.62) and (6.63) and the derivative of the mesh deformation residual $\frac{\partial \mathbf{T}}{\partial \mathbf{D}}$.

6.3.2.1 *Linear Elasticity*

Linear elasticity is used extensively to deform the mesh in this work but the derivation of the mesh adjoint is the same for other implicit deformation methods [52, 81, 13]. Using the linear elasticity method the surface and volume mesh are related by:

$$\mathbf{K}\mathbf{X} = \mathbf{S} \tag{6.66}$$

where **S** represents the surface mesh points and **K** is the mesh deformation matrix. equation (6.66) can be rearranged such that it is in the form of a residual so equation (6.59) is satisfied:

$$\mathbf{K}\mathbf{X} - \mathbf{S} = 0 \tag{6.67}$$

Substituting $\mathbf{T} = \mathbf{K}\mathbf{X} - \mathbf{S}$ into equation (6.63) gives:

$$\left(\frac{\partial \mathbf{T}}{\partial \mathbf{X}}\right)^T \boldsymbol{\lambda}_{mesh} = \left(\mathbf{K}\frac{\partial \mathbf{X}}{\partial \mathbf{X}} - \frac{\partial \mathbf{S}}{\partial \mathbf{X}}\right)^T \boldsymbol{\lambda}_{mesh} = -\left(\frac{\partial I}{\partial \mathbf{X}}\right)^T - \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^T \boldsymbol{\lambda}_{flow} \tag{6.68}$$

since $\mathbf{S}$ has no explicit dependence on $\mathbf{X}$ equation (6.68) can be rearranged as:

$$\lambda_{mesh}^{T}[\mathbf{K}] = -\left(\frac{\partial I}{\partial \mathbf{X}} + \lambda_{flow}^{T}\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right) \tag{6.69}$$

Similarly, for equation (6.65):

$$\frac{dI}{d\mathbf{D}} = \lambda_{mesh}^{T}\frac{\partial \mathbf{T}}{\partial \mathbf{D}} = \lambda_{mesh}^{T}\left(\mathbf{K}\frac{\partial \mathbf{X}}{\partial \mathbf{D}} - \frac{\partial \mathbf{S}}{\partial \mathbf{D}}\right) \tag{6.70}$$

since $\mathbf{X}$ has no explicit dependence on $\mathbf{D}$:

$$\frac{dI}{d\mathbf{D}} = \frac{dI}{d\mathbf{S}}\frac{\partial \mathbf{S}}{\partial \mathbf{D}} = -\lambda_{mesh}^{T}\frac{\partial \mathbf{S}}{\partial \mathbf{D}} \tag{6.71}$$

An extra step was added in equation (6.71) (middle term) to highlight an interesting inference. This shows that the vector of mesh adjoint variables, λ_{mesh}, will provide the sensitivity of the objective function to changes in the surface ($\frac{\partial I}{\partial \mathbf{S}} = -\lambda_{mesh}^{T}$). This can be used to give the designer extra information when designing/optimising a wing or flow control device.

6.3.2.2 *Delaunay Graph Mapping*

The mesh adjoint equations will now be derived for the DGM as the mesh deformation method. The DGM for mesh deformation can be formulated in matrix form

$$\mathbf{X} = \mathbf{E}\mathbf{B} \tag{6.72}$$

where $\mathbf{E}$ is the mesh deformation matrix and $\mathbf{B}$ is the matrix of Delaunay element vertices. $\mathbf{B}$ includes all the points used for the Delaunay vertices including those on the farfield, surface and any intermediate points which may be used, such that $\mathbf{S} \subset \mathbf{B}$ where $\mathbf{S}$ is the surface region to be optimised.

It is important to note that the matrix $\mathbf{E}$ is constant and for only geometry changes in the surface region of interest the rate of change of the mesh w.r.t. the design variables can be represented in terms of $\mathbf{S}$ and $\mathbf{E}$.

$$\frac{\partial \mathbf{X}}{\partial \mathbf{D}} = \mathbf{E}\frac{\partial \mathbf{S}}{\partial \mathbf{D}} \tag{6.73}$$

The residual of the mesh deformation can be defined as $T = \mathbf{X} - \mathbf{E}\mathbf{B} = \mathbf{0}$. Substituting T into equation (6.63) gives:

$$\left(\frac{\partial \mathbf{T}}{\partial \mathbf{X}}\right)^{T}\lambda_{mesh} = \left(\frac{\partial \mathbf{X}}{\partial \mathbf{X}} - \mathbf{E}\frac{\partial \mathbf{S}}{\partial \mathbf{X}}\right)^{T}\lambda_{mesh} = -\left(\frac{\partial I}{\partial \mathbf{X}}\right)^{T} - \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)^{T}\lambda_{flow}$$

$$\lambda_{mesh}^{T} = -\left(\frac{\partial I}{\partial \mathbf{X}} + \lambda_{flow}^{T}\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right) \tag{6.74}$$

Similarly, substituting $\mathbf{T}$ into equation (6.65) gives:

$$\begin{aligned}\frac{dI}{d\mathbf{D}} &= \lambda_{mesh}^{T}\frac{\partial \mathbf{T}}{\partial \mathbf{D}} \\ &= \lambda_{mesh}^{T}\left(\frac{\partial \mathbf{X}}{\partial \mathbf{D}} - \mathbf{E}\frac{\partial \mathbf{S}}{\partial \mathbf{D}}\right) \qquad (6.75)\end{aligned}$$

$$= -\lambda_{mesh}^{T}\mathbf{E}\frac{\partial \mathbf{S}}{\partial \mathbf{D}} \qquad (6.76)$$

For the DGM it is clear that the gradient calculation is much simpler than the equivalent for LE. Comparing equations (6.74) and (6.76) for DGM and equations (6.69) and (6.71) for LE, shows that DGM is a simple vector-matrix product compared to the solution of a costly linear system for LE. The surface sensitivity can be extracted by noticing that $\frac{\partial I}{\partial \mathbf{S}} = -\lambda_{mesh}^{T}\mathbf{E}$.

$\mathbf{E}$ is a sparse matrix of size $[N_{\mathbf{X}} \times N_{\mathbf{B}}]$, each row contains only 4 non-zero entries for a 3D mesh and 3 non-zero entries for a 2D mesh so the mesh deformation is easy to store in memory.

6.4 OPTIMISATION CHAIN

Figure 6.2 shows a flow chart of the optimisation process. It starts with an initial geometry where the surfaces of interest are known. The surfaces which optimisation will be performed on are then parametrised using a method such as Class-Shape Transforms (CST). The surface is then deformed (for the first iteration there is zero deformation) and this information is passed to the mesh deformation method. The mesh is then deformed and stored to be used in the flow solution, flow adjoint and mesh adjoint.

Once the flow solution has been calculated the flow information is passed to the flow adjoint where the vector of flow adjoint variables (λ_F) is calculated. This variable is then passed to the mesh adjoint part so that the final derivatives of the objective function with respect to the design variables ($\frac{dI}{d\mathbf{D}}$) are calculated.

Finally, evaluation of the objective function and the derivatives are used to calculate a new search path for the next optimisation step. This new search path is passed back to the parametrisation.

This process is repeated until the termination condition has been reached, in this case it is a tolerance on the change of objective function.

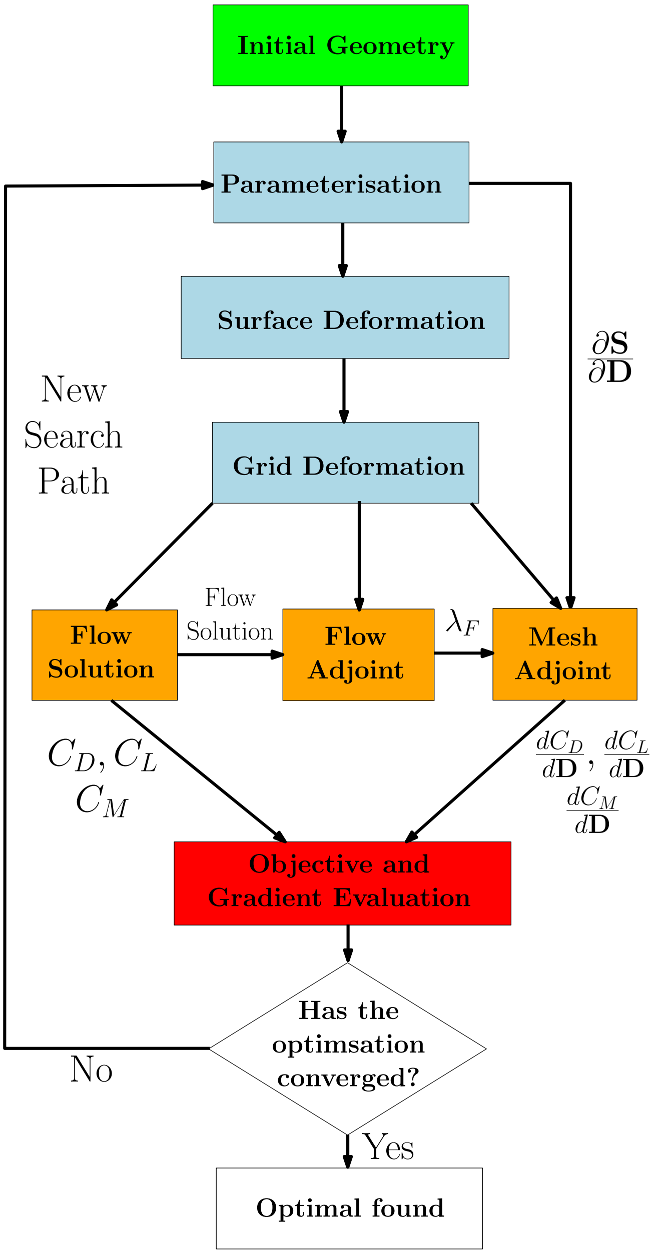


Figure 6.2: The flow of information in an optimisation

Part III

RESULTS AND DISCUSSION

7

OPTIMISATION RESULTS IN 2D

For the 2D optimisation the cubic-spline shock bump design, shown in chapter 3, is used. This has the benefit of intuitive parameters which will allow a better analysis of the effect of the major bump characteristics on the flow. A drawback of the cubic-spline geometry method is that only C^1 continuity can be guaranteed.

The primary function of the shock bump is to reduce drag by reducing the increase in entropy across the shock. When the bump is placed appropriately in the shock region, a bump with relatively low height can have a large effect on drag reduction. The bump geometry itself is simply added onto the aerofoil such that there is only an increase in the z-direction.

The 2D analysis uses the Natural Laminar Flow (NLF) RAE5243 aerofoil. NLF aerofoils tend to have stronger shocks, compared to supercritical counterparts, due to the steady pressure decrease required to promote laminar flow. This makes NLF aerofoils prime candidates for using shock bumps to reduce the effect of the shock. The flow in the simulations is assumed to be turbulent from the leading edge.

7.1 SINGLE DESIGN-POINT

The design point chosen for the RAE5234 comes from the wind tunnel experiment by Fulker and Simmons [82], the flow conditions are M= 0.68 and Re = 1.9×10^7. In the paper by Qin *et al.* [30] a fixed $C_l = 0.82$, this will also be adopted for this design point.

7.1.1 *Choice of mesh*

Figure 7.1 shows a close up view of the mesh around the RAE5243 aerofoil. In this case a strong shock is present on the upper surface of the aerofoil. Figure 7.4 shows that the shock is present at around 62% chord, this is represented by the clustering of pressure contours in this location. A shock bump placed in this location can reduce the drag due to the occurrence of a shock.

7.1.1.1 *Mesh Independence*

Figure 7.2 shows the mesh independence study for the mesh around the RAE5243 aerofoil. The mesh used has 47808 cells which are clustered around the surface and the wake region to accurately capture the physics in the boundary region. This mesh presents the least amount of cells, which will reduce the time of the overall optimisa-

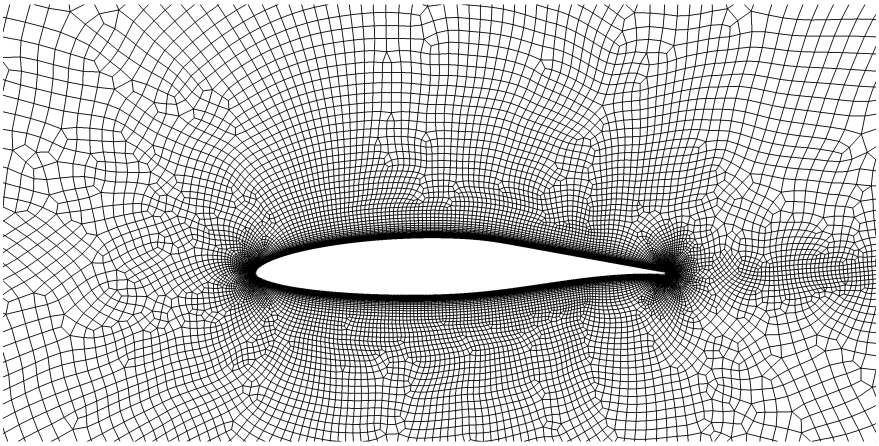


Figure 7.1: Mesh around the RAE5243 aerofoil

tion procedure, while still showing a drag coefficient close to the meshes with many more cells.

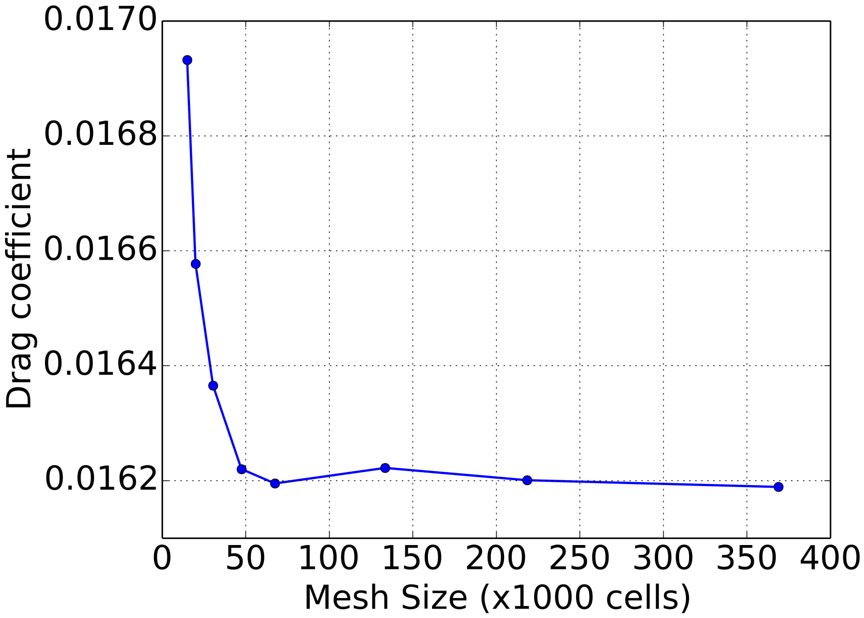


Figure 7.2: Mesh independence for the RAE5243 aerofoil

7.1.1.2 *Verification Against Experiment*

The wind tunnel experiment from Fulker and Simmons [82] ran at an $\alpha = 0.77°$ and had a fully turbulent boundary layer. Figure 7.3 compares the surface pressure from the wind tunnel to the computational solution from TAU run at the same conditions. The match between the two sets of data is very good. A favourable pressure gradient is present on the upper surface of the aerofoil before it is terminated by a shock at about 55% chord.

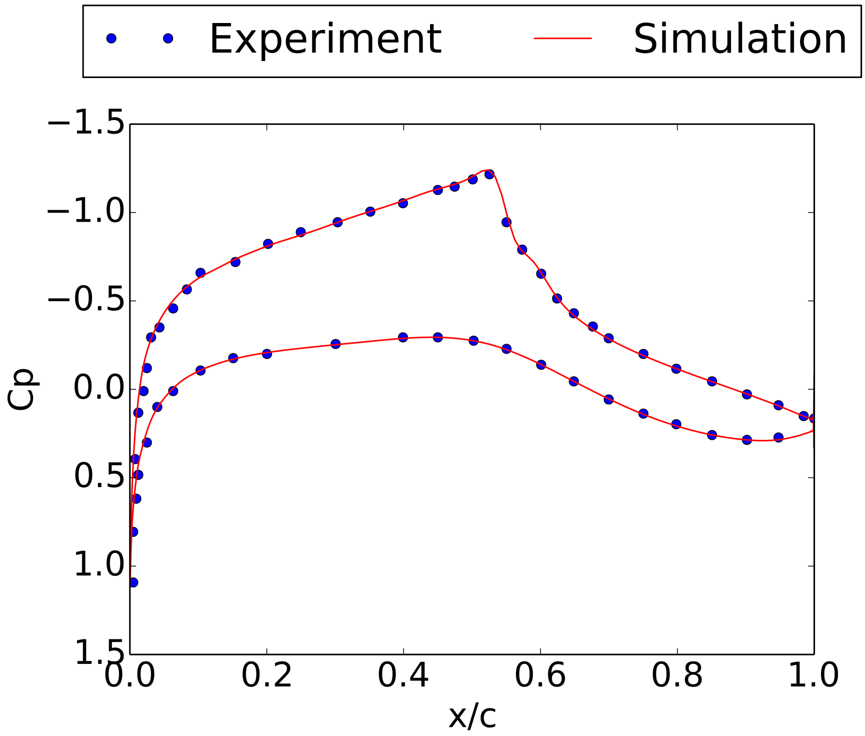


Figure 7.3: Comparison between computational and wind tunnel surface pressure

7.1.1.3 *2D Shock Bump Optimisation*

The shock bump will be optimised by using the mesh adjoint methods described in section 6.3 with the linear elasticity mesh movement strategy. The mesh for the RAE5243 is shown in figure 7.1. This gives good clustering around the aerofoil to capture the geometry change, due to the addition of a bump, and good resolution in the shock location.

Table 7.1 shows the optimisation bounds for the bump which will be placed on the RAE5243 aerofoil. The parameters have been chosen such that as large as possible design space can be searched to find an optimum bump.

Table 7.1: Optimisation bounds

	Bump height %c	**Bump loc.** %c	**Bump len.** %c	**Peak loc.** %bl
Lower Bound	0.3	55.0	1.0	30.0
Upper Bound	1.5	70.0	50.0	70.0
Starting Value	0.5	62.0	29.0	56.65

The bump height is bounded between $[0.3, 1.5]$ percent chord, this allows for a large bump to grow to treat the shock if required. The bump height is limited to exclude zero as is it was found a local minimum exists at zero bump height. This means the local minimum is at zero bump height despite the fact drag reductions can be found for higher bump heights.

The other 3 parameter bounds were chosen such that a long bump can form but is not likely to extend over the limits of the aerofoil. The minimum location for the start of the bump is 25%c and the end of the bump has a maximum of 105%c. Despite the end location being beyond the aerofoil it was deemed unlikely to occur as it would require extreme values of bump length, peak location and bump location.

Consider an optimisation problem of the form:

$$\text{minimise} \quad I(\mathbf{x}) \tag{7.1}$$

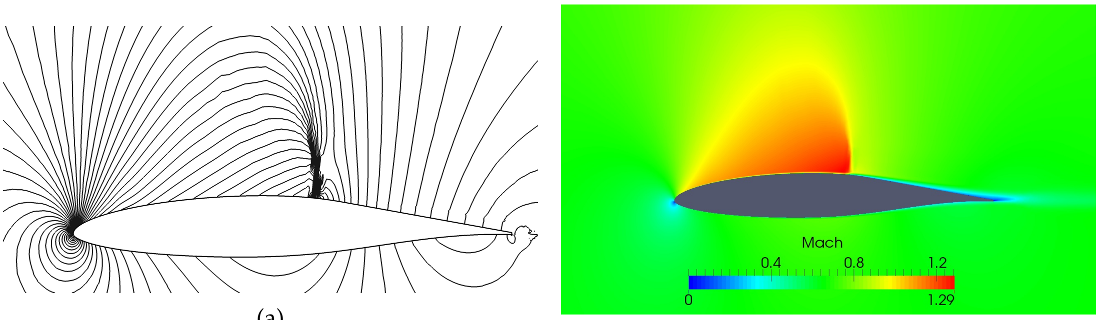
$$\text{subject to} \quad G_i(\mathbf{x}) = 0 \quad \text{for} \quad i = 0, 1, ..., n \tag{7.2}$$

$$\text{and} \quad C_j(\mathbf{x}) \geqslant 0 \quad \text{for} \quad j = 0, 1, ..., m \tag{7.3}$$

where $\mathbf{x}$ are the design variables, $I(\mathbf{x})$ is the objective function, $G_i(\mathbf{x})$ are the equality constraints and $C_j(\mathbf{x})$ are the inequality constraints.

It is known that the optimum found from a gradient based method is highly sensitive to the starting location. If the search space (in this case the objective is drag coefficient) is multi-modal the optimiser only finds a local minimum meaning there is no guarantee that the best optimum can be found. To try and speed up the optimisation and to find a good minimum guidelines from the literature is used to get a reasonable starting location.

The bump length has been chosen to be 30%c based on the findings from Qin *et al.* [30]. Figure 7.4 shows the pressure and Mach contours around the RAE5243 aerofoil for the baseline geometry. The clustering of the pressure contours indicate the presence of a strong shock on the upper surface of the aerofoil. The Mach contours show that the maximum Mach number is 1.29 and occurs at the shock location.



(a)

(b)

Figure 7.4: Pressure contours (left) and Mach contours (right) for the baseline RAE5243 aerofoil at the design condition

The Euroshock II project [24] suggests that for an asymmetric bump the peak location should be 5 - 10%c downstream of the shock for a bump length of 20%c. Extrapolating these findings for a bump length of 30%c and a shock location on the

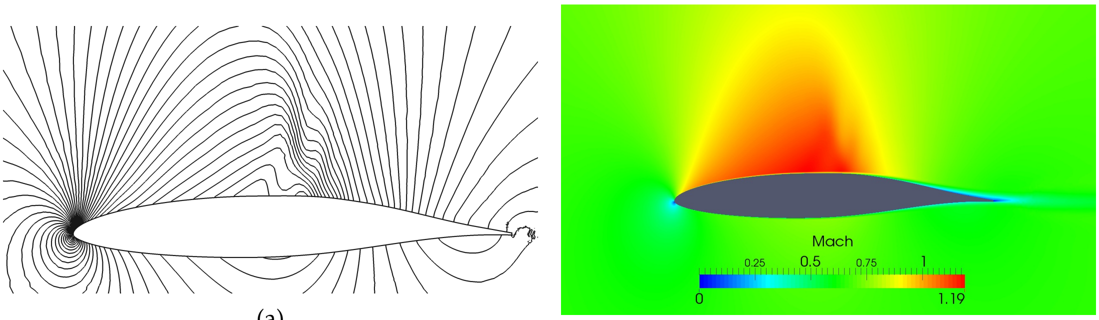
RAE5243 of 55.3%c gives the bump starting location as approximately 62%c. The peak location was chosen as 57%c to give a longer compression region on the front bump contour and a shorter rear part of the bump.

Table 7.2: Beginning and ending parameter values for single-point bump optimisation

	Bump height %c	**Bump loc.** %c	**Bump len.** %c	**Peak loc.** %bl
Starting Value	0.5	62.0	29.0	57
Ending Value	0.882	69.6	50	55.9

Table 7.2 shows the parameter values after optimisation. The bump height has increased to 0.882%c over the course of the optimisation to treat the strong shock. The bump length shows the most dramatic change, increasing from 30%c at the start of the optimisation to 50%c at the final iteration. This will have the effect of having a more gradual geometry change over the optimisation region, meaning the area will change more gradually. The bump location has moved downstream by over 7%c and the peak location has moved forward slightly by 1.1%bl.

Figure 7.5 shows the pressure contours around the RAE5243 aerofoil with the optimised bump included. Clearly the pressure change around the shock is much more gradual giving a 'knee' shape which is characteristic of this shock bump topology and the shock has been replaced completely with isentropic compression waves. The Mach contours show how the show has been spread across the optimisation region and the maximum Mach number has been reduced to 1.19.



(a)

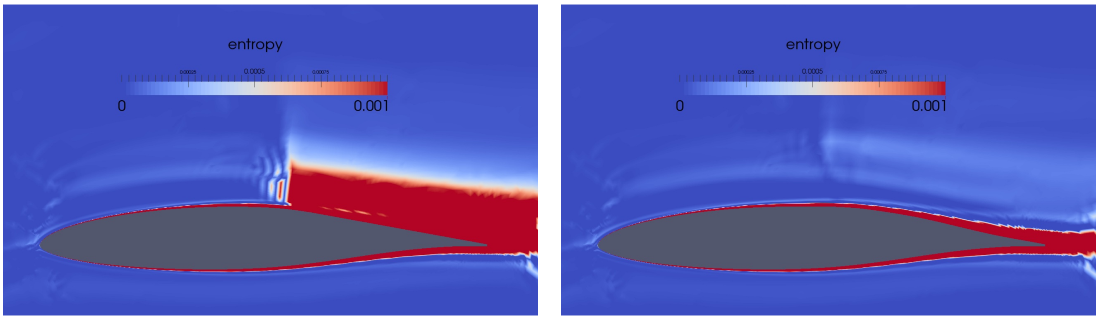
(b)

Figure 7.5: Pressure contours (left) and Mach contours (right) for the baseline RAE5243 aerofoil with optimised bump at the design condition

The near-field drag decomposition in table 7.3 shows a reduction in drag coefficient for the optimised case compared to the baseline geometry of over 36 drag counts. All the reduction comes from the pressure drag which shows a reduction of almost 35% and a small penalty in the skin friction is observed. The skin friction penalty is caused by the thickening of the boundary layer due to the placement of a bump in the flow.

Table 7.3: Near-field drag decomposition for the RAE5243 aerofoil

	c_d	$c_{d,pres}$	$c_{d,fric}$
Baseline	0.01622	0.01128	0.00495
Optimised bump	0.01259	0.00737	0.00522
Difference	-22.38 %	-34.66 %	5.45 %



(a) Baseline

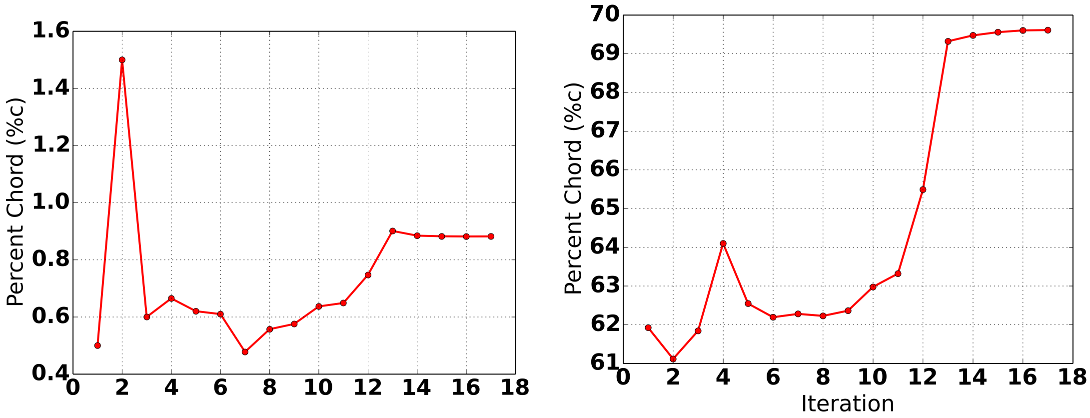
(b) Optimised bump

Figure 7.6: Normalised entropy for the baseline (LEFT) and optimised bump (RIGHT) aerofoil

A comparison of the normalised entropy between the baseline and optimised geometry is shown in figure 7.6. It is clear that the addition of the optimised bump has almost entirely stopped the entropy increase due to the shock wave on the upper surface of the aerofoil. This will result in a large wave drag reduction when the bump is added, this is reflected in the reduction of near-field pressure drag.

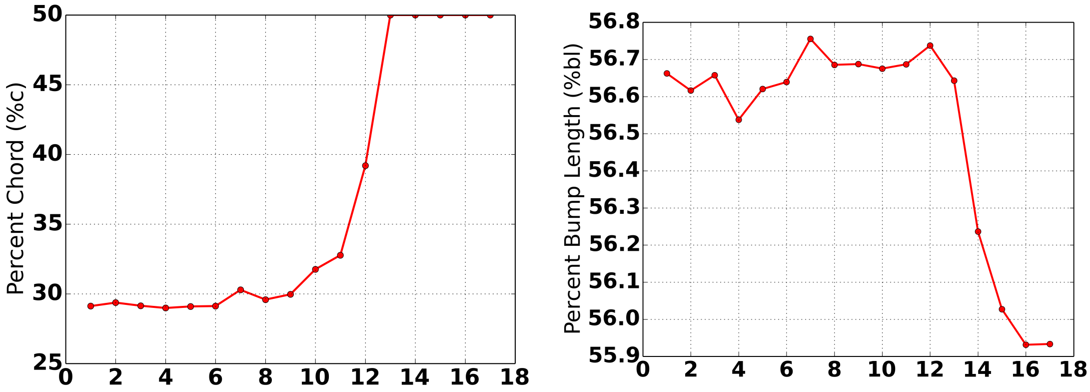
Figure 7.7 shows the change of the design variables at each optimisation iteration. The large spike in the optimisation history appears to relate to a spike in the bump height, it is possible that the large change in bump height caused separation on the aerofoil which would lead to an increase in drag. After iteration 8 the design variables move quickly towards the design variable values found at the final optimisation iteration.

Finally figure 7.8 shows the change in geometry when adding the shock bump and the surface pressure distribution comparison between the optimised bump and baseline geometries. The change in the geometry is very small, the geometry requires graphically expanding to see the difference clearly. For the surface pressure distributions the geometry with optimised bump shows a much more gradual change in pressure at the shock region, it is clear there is no shock when the optimised bump is in place.



(a) Bump height

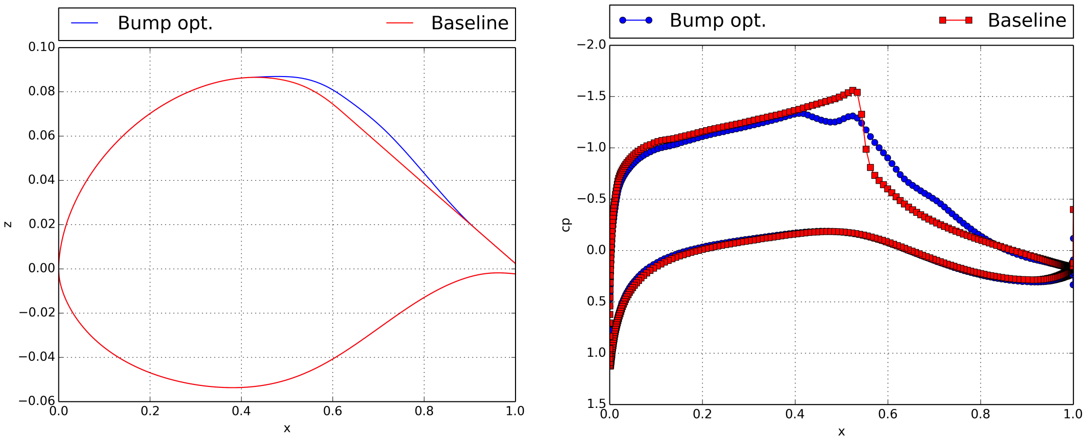
(b) Bump location



(c) Bump length

(d) Peak location

Figure 7.7: Progression of the design variables throughout the optimisation



(a) Geometry change

(b) Surface pressure

Figure 7.8: Surface pressure and geometry change of the RAE5243 aerofoil

7.2 MULTI-POINT

Multi-point analysis requires the assessment of a geometry at several different flow conditions to assess how robust the design is. If there are m flow conditions (or points) taken into account then there are m values of interest (e.g. drag, lift or moment coefficients). To use these values in an optimisation either a multi-objective approach must be taken or a combined objective is necessary. A true multi-objective optimisation is usually done using a global optimisation method. This produces several different solutions which lie on a Pareto front. For a gradient based approach multi-objective is difficult to implement therefore a combined objective is used. In this work a weighted sum will be used to combine the values of interest from the m points. The Mach number will be varied while the other flow conditions will be kept constant ($Re = 1.9 \times 10^7$ and $C_l = 0.82$).

7.2.1 *Choice of weights*

The choice of weights in a weighted sum are often defined based on experience or trial and error to achieve a good result from the optimisation. A good result is one where a compromise is found between the objectives. This can be very costly and gives no guarantee that the best optimal will be found and it also requires familiarisation with the problem at hand.

With multi-objective optimisation the result is often of the form of a Pareto front. Figure 7.9 shows an example of a Pareto front for objectives $\mathbf{I}_1$ and $\mathbf{I}_2$. The Pareto front is made up of solutions with no dominance over each other, therefore each point along the Pareto front represents a trade-off between objectives. For a solution J_1 to be dominant over another J_2 the following has to be true: $J_1 \geqslant J_2$ for weak dominance or $J_1 > J_2$ for strong dominance. In the case of using a weighted sum the optimal should exist somewhere on this Pareto front.

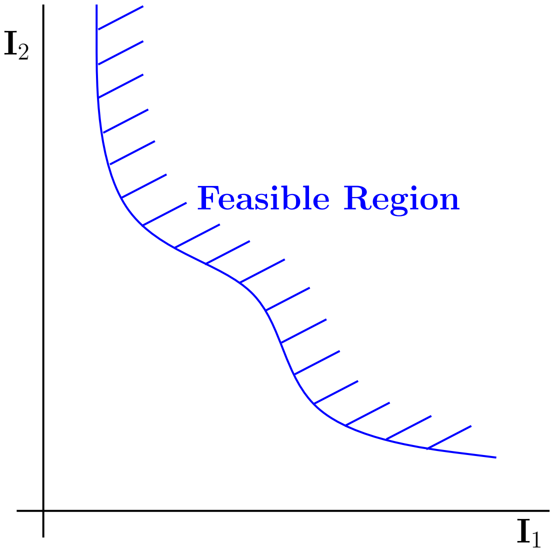


Figure 7.9: Example Pareto front

One interesting method for choosing these weights is by finding the "Utopia Point". The "Utopia Point" is the point where the solution for each value of interest is the same of the solution to each individual single-point optimisation. Usually this point is outside the feasible region and cannot be reached through optimisation.

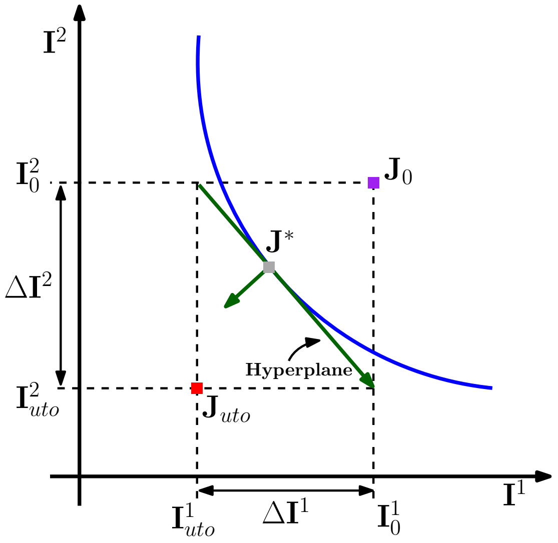


Figure 7.10: Illustration of the utopia point in reference to a Pareto front

Figure 7.10 shows a depiction of the "Utopia Point", $\mathbf{J}_{\mathrm{uto}}$, and Pareto front. Since the "Utopia Point" can not be reached, the next best solution needs to be found, $\mathbf{J}^*$. This is determined to be the point 'closest' to the "Utopia Point". The point $\mathbf{J}^*$ represents a compromise between the two values of interest of $\mathbf{I}^1$ and $\mathbf{I}^2$. At $\mathbf{J}^*$ the tangent hyperplane is defined by $\left(\frac{1}{\Delta\mathbf{I}^1}, \frac{1}{\Delta\mathbf{I}^2}\right)$ with normal $(\Delta\mathbf{I}^1, -\Delta\mathbf{I}^2)$.. Therefore, using weights $w_i = \frac{1}{\Delta\mathbf{I}^i}$ in an optimisation with a weighted sum objective will converge towards $\mathbf{J}^*$, more details on this in the next section.

The drawback is that to calculate the weights in this form requires m single-point optimisations to find the "Utopia Point" which is too costly. Marler and Arora [83] proposed a method called "unattainable aspiration" as a reasonable substitution which only requires m CFD calculations.

The weights take the form:

$$w^i = \begin{cases} \frac{1}{\alpha_1 C^i_{D_f} + \alpha_2 C^i_{D_p}}, & near - field \\ \frac{1}{\alpha_3 C^i_{D_w} + \alpha_4 C^i_{D_v} + \alpha_5 C^i_{D_{ind}}}, & far - field \end{cases} \tag{7.4}$$

The near-field or far-field formulation of drag can be taken however for this work the near-field formulation is taken for simplicity. At off-design the shock bump can increase the strength of the shock and trigger shock induced separation which will increase the pressure drag. Therefore using the pressure drag, not just the wave drag, in the weighting scheme is appropriate. From the single-point optimisation drag decomposition, see table 7.3, it is clear that all the drag reduction when placing the bump comes from the pressure drag and not the skin friction. Setting $\alpha_1 = 1$ and $\alpha_2 = 0$ will focus the optimisation on the reduction of pressure drag.

7.2.2 *Weighted Sum*

The objective for the multi-point optimisation will be a weighted sum of the drag value obtained from each CFD simulation per iteration for the multi-point optimisation. If there are m CFD simulations then the optimisation problem can be stated as follows

$$\text{minimise} \ \ J = \sum_{i=0}^{m} w_i C_D^i \tag{7.5}$$

$$\text{subject to} \ \ \sum_{i=0}^{m} w_i C_j = \alpha \quad j = 1, 2, ..., n \tag{7.6}$$

$$\text{and} \ \ \sum_{i=0}^{m} w_i C_k \geqslant \beta \quad j = 1, 2, ..., b \tag{7.7}$$

The weighted sum is a simple approach to finding a combined objective for multi-point simulation and has the benefit of the optimal value being a Pareto optimal [83] for positive weights. A major benefit of using a weighted sum is that the point to which the optimisation converges has a normal to the tangent hyperplane which is collinear to the a normal vector of the hyperplane formed by the weights used in the sum [84].

7.2.3 *Optimisation*

Table 7.4 shows the weights which will be used in the weighted sum objective, equation 7.5, for the multi-point optimisation. These weights were derived from the near-field pressure drag components from the single-point design. for some constraints α and β. The weighting puts more emphasis on the lower Mach numbers around the design point to try to achieve a fair compromise for all flow conditions. Since the pressure drag has been reduced at the design point $M = 0.68$ the weight for this point is larger than the weights for the Mach numbers around it.

Table 7.4: Weighting scheme for weighted sum objective

Mach	**0.66**	**0.67**	**0.68**	**0.69**	**0.70**	**0.71**
Weights	0.2083	0.1950	0.2072	0.1669	0.1210	0.1017

Table 7.5 shows the starting and finishing values for the design variables for the multi-point optimisation. The parameters which changed the most are the bump height and the peak location. The bump height has increased significantly in comparison to the single-point design and the peak location has moved forward.

Table 7.6 shows the drag coefficient comparison between the single-point, multi-point and baseline geometries. The multi-point design improves on the baseline geometry consistently for the range of Mach numbers from $M = 0.67$ and above. For $M = 0.66$ there is a slight drag penalty, however the drag penalty for the multi-point

Table 7.5: Optimisation starting and finishing design variables

	Bump height %c	**Bump loc.** %c	**Bump len.** %c	**Peak loc.** %bl
Starting Value	0.882	69.6	50	55.9
Ending Value	1.42	70	50	44.7

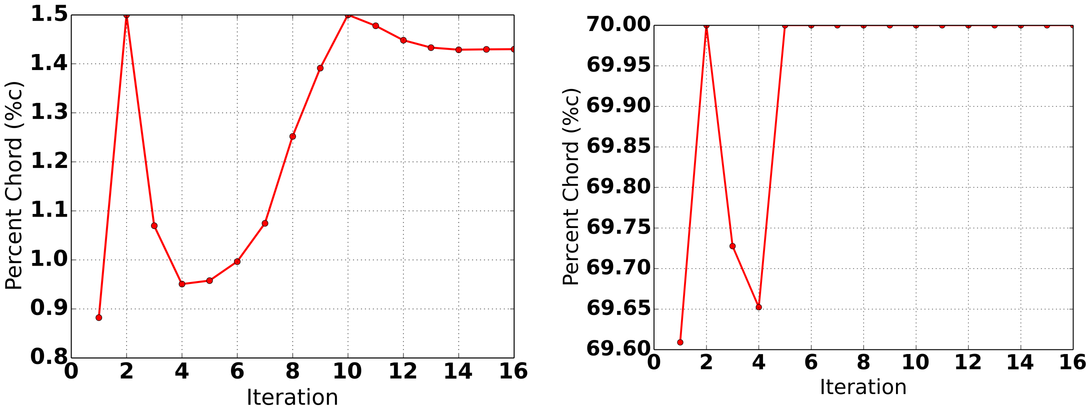
Table 7.6: Comparison of drag values from the baseline, single-point and multi-point cases for the RAE5243

	Baseline	**Single -point**	**Multi -point**	**Diff. base**	**Diff. single**
M=0.66	0.01210	0.01252	0.01243	2.73 %	-0.72%
M=0.67	0.01326	0.01338	0.01299	-2.04 %	-2.92%
M=0.68	0.01622	0.01259	0.01429	-11.90 %	13.50%
M=0.69	0.02151	0.01563	0.01464	-31.94 %	-6.33%
M=0.70	0.02997	0.02156	0.01705	-43.11 %	-20.92%
M=0.71	0.04224	0.02566	0.02352	-44.32 %	-8.34%

is less than that of the single-point design. The comparison to the single-point shows the multi-point design achieves a better drag reduction than the single-point design everywhere except at $M = 0.68$ which is the single-point design Mach number.

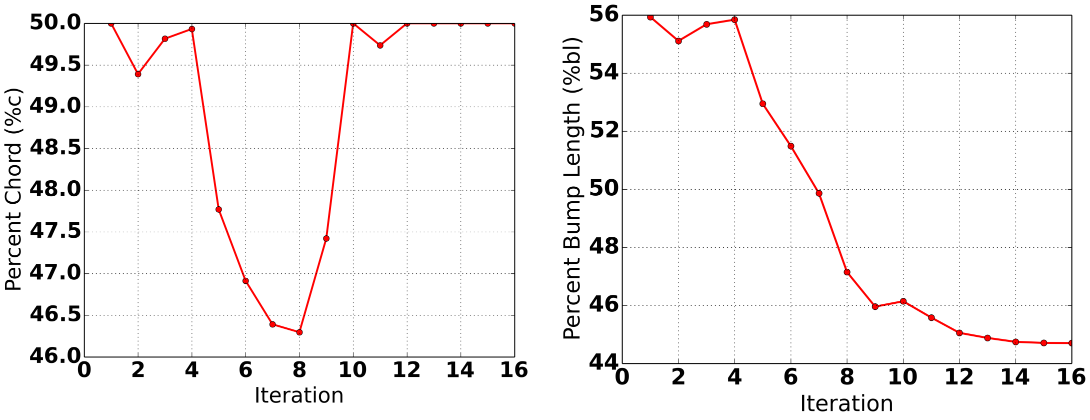
The design variable progression presented in figure 7.11 shows that the bump location does not vary that much throughout the simulation. The bump length shortens slightly but then grows towards the maximum again and the peak location steadily moves forwards in relation to the bump length. Figure 7.12 shows the difference between the baseline, single-point and multi-point geometry. The surface pressure distribution for each condition shows that there are still shocks present but the surface pressure has been perturbed by the existence of the bump.

The multi-point optimisation has given a geometry which delays the drag divergence Mach number significantly from the baseline and single-point cases and gives a drag reduction for much higher Mach numbers.



(a) Bump height

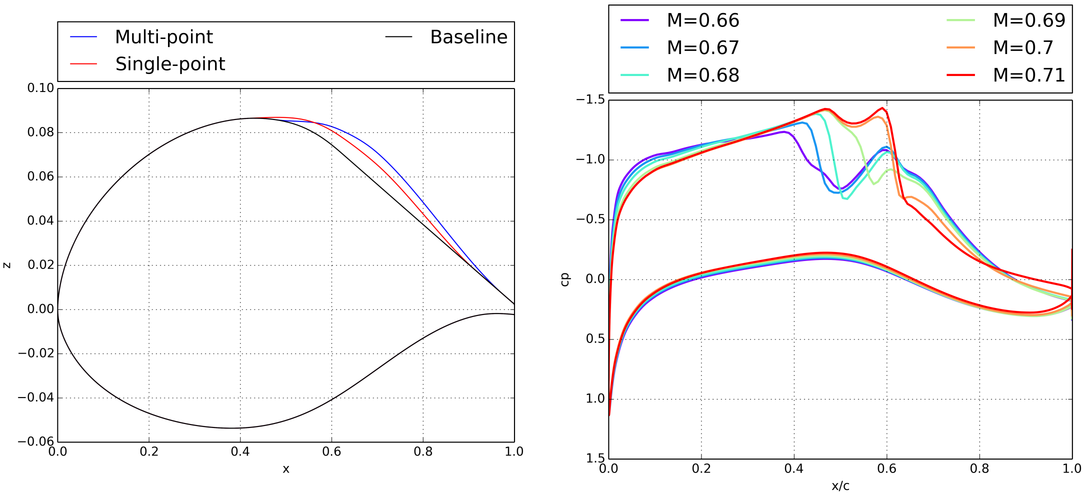
(b) Bump location



(c) Bump length

(d) Peak Location

Figure 7.11: Progression of the design variables throughout the optimisation



(a) Geometry change

(b) Surface pressure

Figure 7.12: Surface pressure and geometry change of the RAE5243 aerofoil after multi-point optimisation

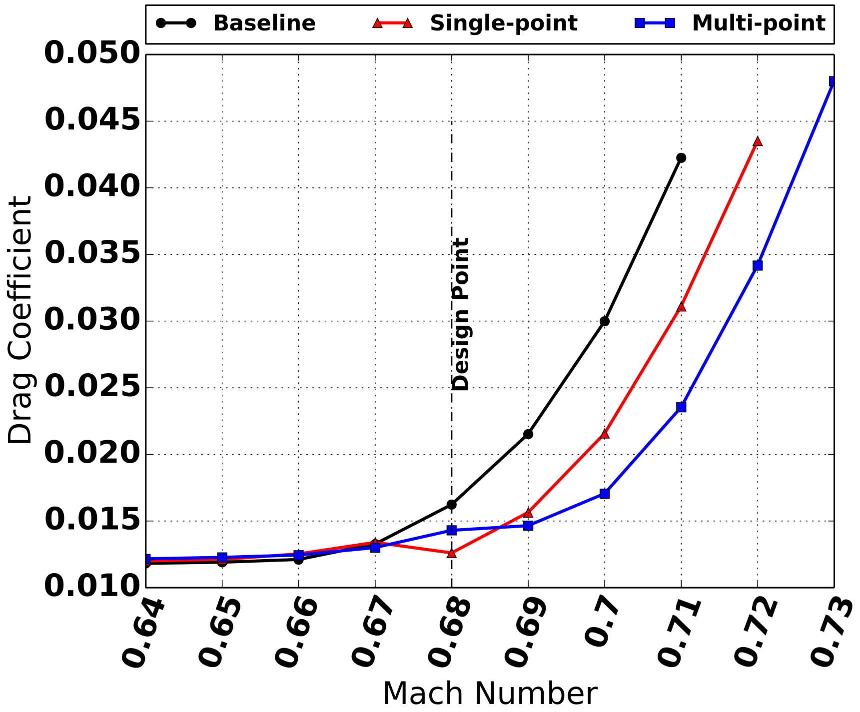


Figure 7.13: Robustness comparison between multi-point, single-point and baseline designs

8

USING SURFACE SENSITIVITY AS A DESIGN TOOL

In previous research, the deployment position of a shock control bump is based on what is viewed as the shock location. This can lead the designer to use a poor deployment position or an important area may be missed due to the designers' lack of experience. This could cause a detrimental effect on bump performance which will be less than the expectation or the optimization may get stuck in a local extremum without further reduction of wave drag. In the worst case, the bump can cause other issues, such as separation, which will degenerate wing performance. In addition to this, there may also be regions which have an effect on the drag with no discernible traces to indicate the area from the flow solution. It is desirable to clearly identify the sensitivity of drag to changes in the surface geometry. The shock control bump is then deployed in the most sensitive areas. This would be of particular use for retrofitting the device on an existing wing.

In section 6.3 it was shown that when calculating the mesh adjoint for geometric changes, $\frac{dI}{dS}$ can be extracted from the formulation with no extra calculation required. When this is plotted on the wing surface it will show areas where moving the surface will have the most effect on drag.

Using the aerofoil from the previous section (RAE5243) the sensitivity of the objective function (e.g. Coefficient of drag) to changes in the surface are calculated for the baseline geometry. These are compared to the aerofoil after the shock bump is added.

Figure 8.1 shows the drag coefficient surface sensitivity for the RAE5243 aerofoil for the baseline geometry and the geometry after the optimised bump has been added. The starting location of the bump is at 41.7%c and the bump peak is at 69.6%c, therefore the shock location is entirely on the compression part of the bump. It is clear that after the placement of the bump the strength of the sensitivity due to the shock has reduced significantly.

The sensitivity has also reduced, and in some cases changed sign, for the areas in front of the optimised bump. This is attributed to the fact that adding a bump is effectively adding camber to the aerofoil and since the C_l is fixed the angle of attack will be less for the optimised case compared to the baseline geometry.

This section explores using the surface sensitivity for the design and optimisation of flow control devices on a 3D wing. The M6 and F6 wing are used as they are covered widely in the literature and give a good benchmark for assessing the success of the optimisation.

The optimisation chain in section 6.4 shows the general flow for an optimisation using the mesh adjoint. For this section there are some extra considerations to be added. Figure 8.2 shows the ammended optimisation chain. Two extra steps are included directly before the parametrisation step. The 'Surface Mesh Sensitivity' is

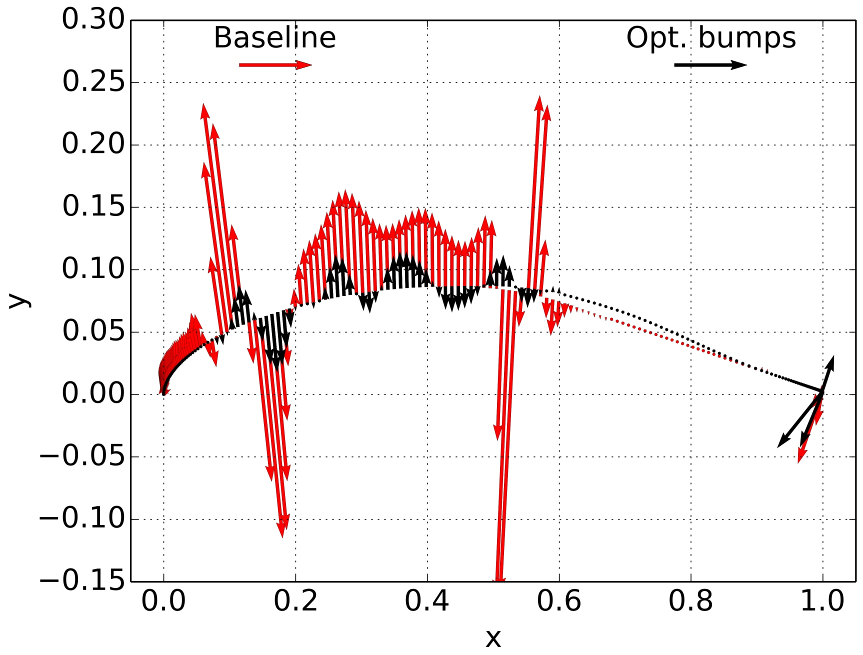


Figure 8.1: Comparison between baseline and optimised bump drag surface sensitivity vectors showing a reduction in magnitude of the vectors in the shock region due to the placement of a shock bump.

found using the mesh adjoint then the most sensitive areas are extracted. The control bumps are then prepared for deployment in the drag sensitive regions at the parametrisation step.

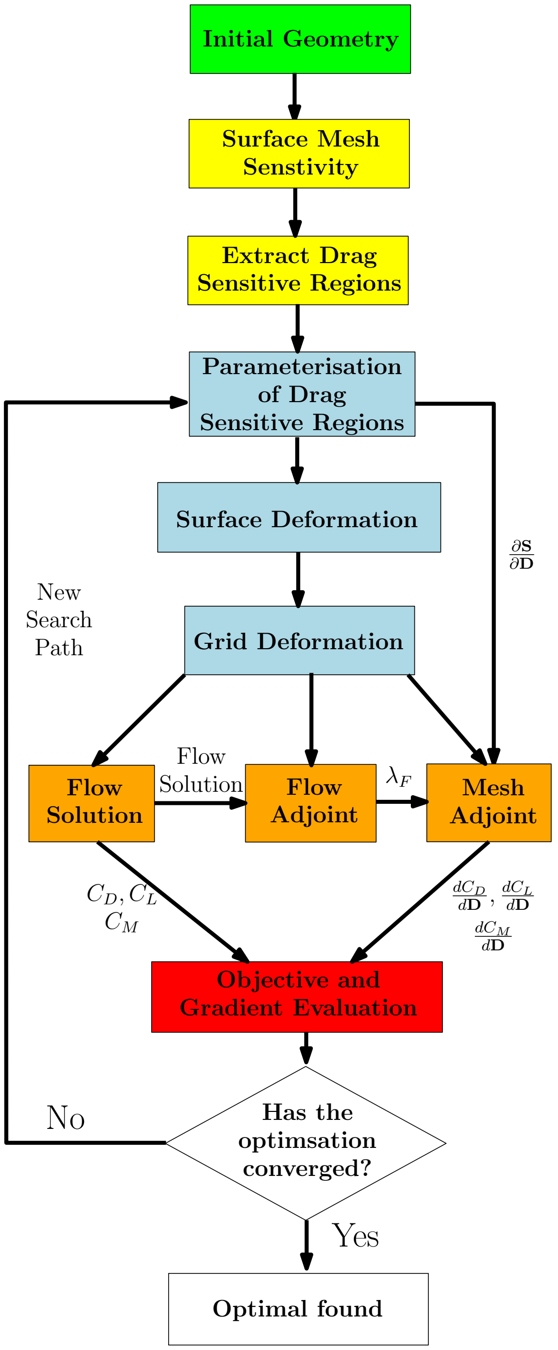


Figure 8.2: Optimisation chain including evaluation of surface sensitivity for flow control device placement

8.1 F6 SURFACE SENSITIVITY

Deriving the surface sensitivity can be an incredibly useful tool in wing design to identify the areas where drag reduction can be achieved. The F6 wing case defined by Zhu *et al.* [29] will be used in this section. The flow demonstrates a complicated flow structure with a lambda shock footprint on the wing surface and shock induced flow separation after the rear shock. The flow conditions are $M = 0.8$, $C_l = 0.6$ and $Re = 5 \times 10^6$ based on a chord length of 1.5m. The wing span is 5.36m, with a crank at 1.81m (33.77%span), a leading edge sweep of 27.1° and has an aspect ratio of 9.5.

Figure 8.3 shows the mesh independence study for the F6 wing. The mesh chosen for this study is the 4.2 million cell mesh. While it is noted that the meshes at 7.8 million and 10 million cells demonstrate better solution independence the meshes are too large in this case to perform the mesh adjoint method with Linear Elasticity mesh movement due to the RAM required. The mesh adjoint is first calculated for the original wing. During the optimisation, adding the shock bump to the wing will only move surface points along the z-direction. Therefore, only the sensitivity in the z-direction ($\lambda_{mesh,z}$ or lambda_gz) is used to identify the sensitivity regions with most drag reduction potential. $\lambda_{mesh,z}$ are the sensitivities of C_d w.r.t. changes of the surface in the z-direction. The drag surface sensitivity for the baseline wing is shown in figure 8.4 and the bump placement has been highlighted.

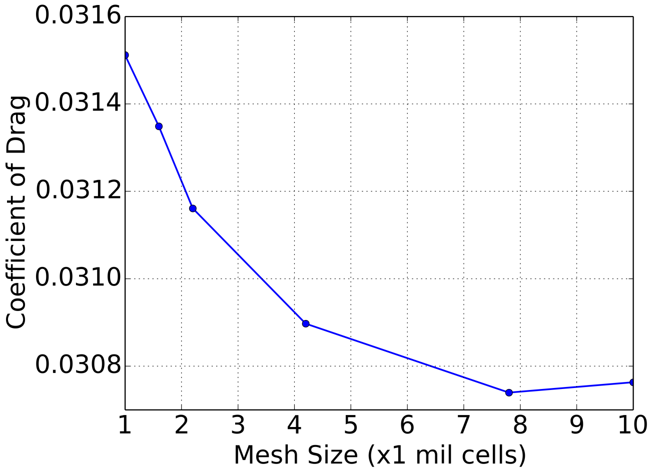


Figure 8.3: Mesh independence study for the F6 wing

Despite the F6 wing having a slightly more complex geometry with a crank, the sensitivity is in fact more simple with a clear shock line being identified. This is surprising as figure 8.5 shows the pressure contours and surface streamlines which indicate that there is in fact a lambda shock structure on the upper surface of the wing. The surface sensitivity has identified that the front shock line in this case will give little drag improvement if the surface is moved in the positive z-direction.

The strongest surface sensitivity points in the shock line are extracted and fitted using a 3^{rd} order polynomial. The area towards the wing tip was neglected as the

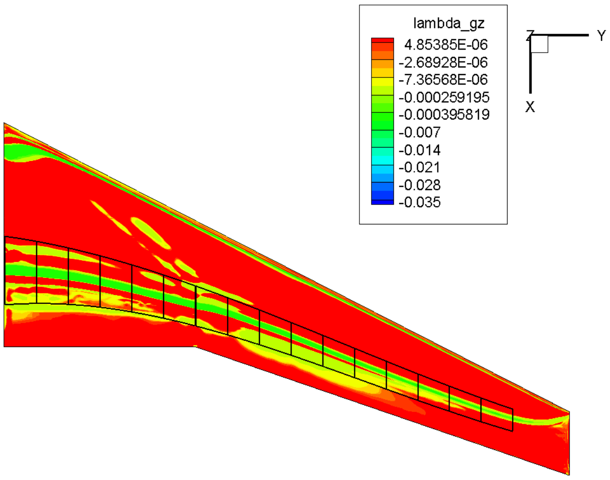


Figure 8.4: Extraction of sensitivity data with 16 bump patches placed in the shock region

sensitivity weakens significantly in this region and it would cause a problem for bump placement as the sensitivity line approaches the wings leading edge.

The bump array starts at 0.19%span and ends at 89.8%span. For 3D contour bumps the work by Qin *et al.* [30] showed that the gaps between bumps as a design variable tends towards zero during optimisation.

The optimisation region is defined by a front line and rear line, specifying the extension of the bumps. Previous research by Wong *et al.* [32] and Qin *et al.* [30] suggested that the shock control bump length should be between 20% to 40% chord length. 16 bumps are placed in the optimisation region with bump length of 30% local chord and width 5.6%span. Finally, the bump has to be deployed w.r.t. the shock feature line. The previous research from Qin and the EUROSHOCK II project [85] suggests that the peak of shock control bump should be between $5 - 10\%$ downstream of the shock for an asymmetric bump. The bump will be placed downstream in terms of shock feature line. The bump has 40% of the bump length in front of the sensitivity line and 60% afterwards. The CST parametrisation of the bumps is used once again as described in chapter 3.

Figure 8.6 shows the final bump array and the effect on the pressure contours and surface streamlines. Comparing figures 8.5 and 8.1 shows that allowing the bumps to grow has weakened the pressure change in the shock line region.

For the baseline wing the surface streamlines show that there is a large shock induced separation on the outboard wing. When the bumps have been deployed separation line is pushed down towards the trailing edge. This is most notable at the outboard part of the separation. This means that the shock bumps have reduced the adverse pressure gradients in the shock region and therefore the flow separation is slightly mitigated.

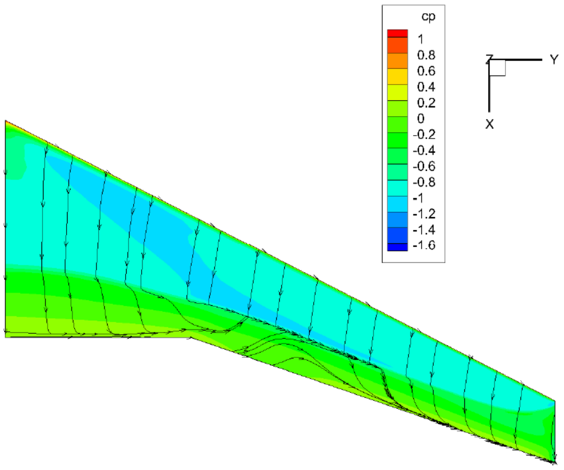
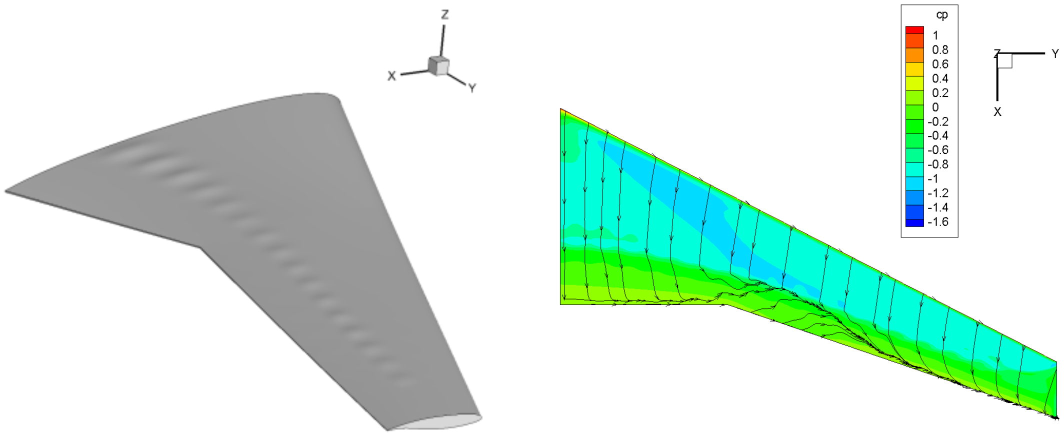


Figure 8.5: Pressure and surface streamlines for baseline F6 wing



(a) F6 wing with optimised bumps

(b) Pressure and surface streamlines of F6 wing with optimised shock bumps

Figure 8.6: Surface distortion and pressure distribution on the upper surface of the M6 wing

Figure 8.7 shows the spanwise wave drag distribution on the F6 wing. There is a wave drag reduction across the wing covered by the optimisation region compared to the baseline wing. The largest difference in wave drag is at the inboard of the wing. The growth of the bumps is limited in the crank area of the wing as the a high bump may cause the flow separation to be worse and increase drag.

Table 8.1 shows the drag decomposition comparison between the baseline wing and the wing with bumps deployed. This tool is provided by ONERA and further details can be found in the paper by Destarac [86]. The total drag is reduced by 4.03% which includes 33.76% wave drag reduction and 1.15% pressure drag reduction. The penalty of friction drag is only increased by 0.59% which is 0.35 drag counts.

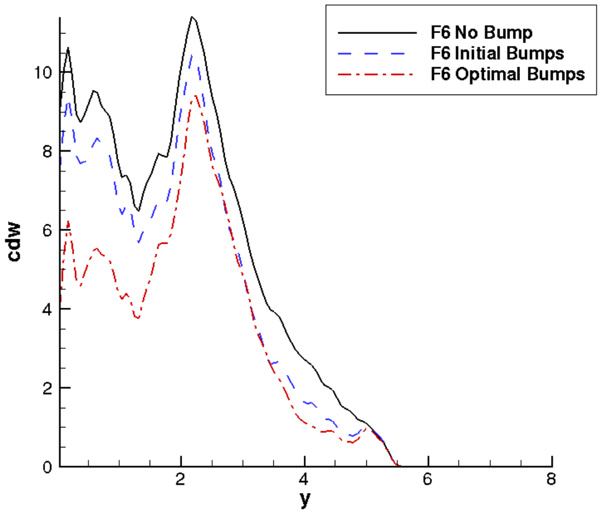
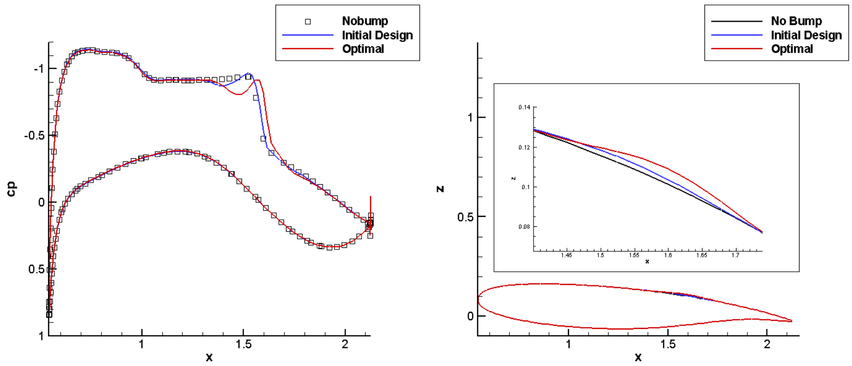


Figure 8.7: F6 spanwise wave drag distribution

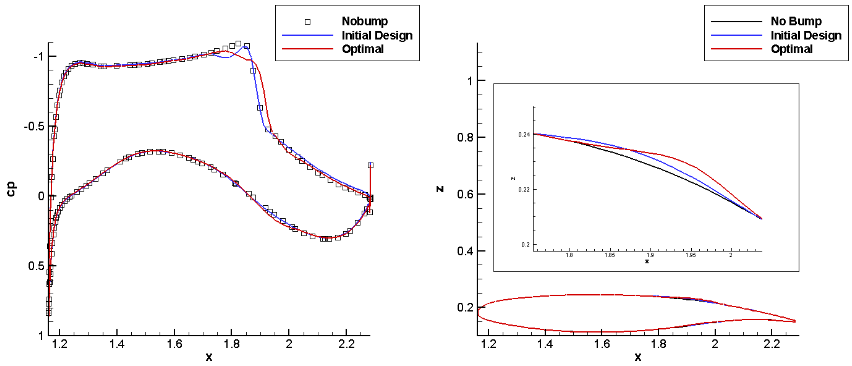
Table 8.1: Drag comparison between F6 optimised shock bumps and the baseline wing

(drag counts)	C_d	$C_{d,wav}$	$C_{d,ind}$	$C_{d,viscpre}$	$C_{d,fric}$	$\frac{C_L}{C_d}$
Baseline	308.97	32.23	133.68	77.20	59.76	19.42
Optimised	296.51	21.35	133.10	76.31	60.11	20.24
Reduction	-4.03%	-33.76%	-0.43%	-1.15%	0.59%	+4.20%

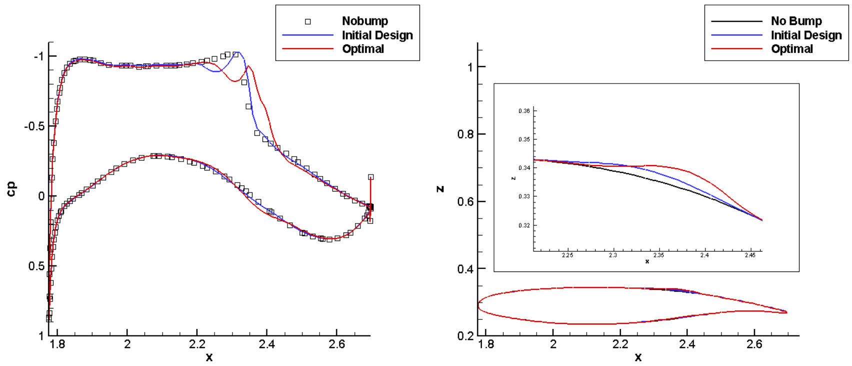
Figure 8.8 shows the surface pressure distributions and optimised bump geometry for a cut down the middle of the 4^{th}, 8^{th}, 12^{th} and 16^{th} bumps. The surface pressure around the shock in these figures shows a reduction in suction peak indicating a weakening of the shock. The shock location has been delayed in all case where the rapid pressure has moved downstream.



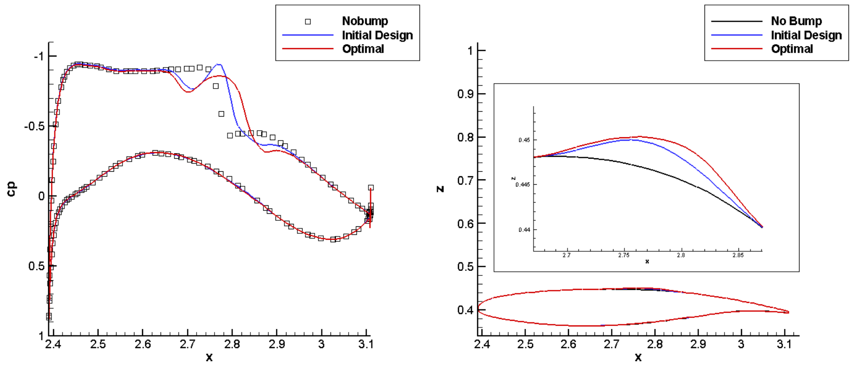
(a) 4th bump (19.78%span)



(b) 8th bump (42.16%span)



(c) 12th bump (64.55%span)



(d) 16th bump (86.94%span)

Figure 8.8: Comparison of surface pressure and bump geometry between baseline, initial bump and optimised bump F6 wing

8.2 M6 SURFACE SENSITIVITY

The M6 wing was chosen from the paper by Schmitt and Charpin [87]. The flow condition is chosen as Mach=0.84, $C_l = 0.274$ and Re=11.72 × 10^6 based on the aerodynamic mean chord of 0.646m. The wing has a span of 1.196m.

Figure 8.9 shows a mesh independence study was undertaken and it was determined that a mesh of approximately 2.6 million cells was sufficient to accurately capture the flow physics.

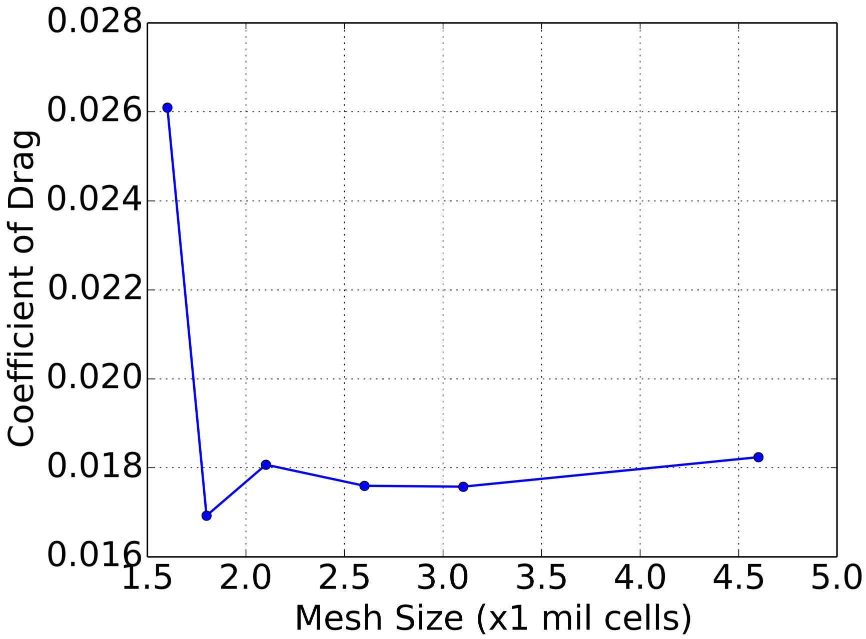
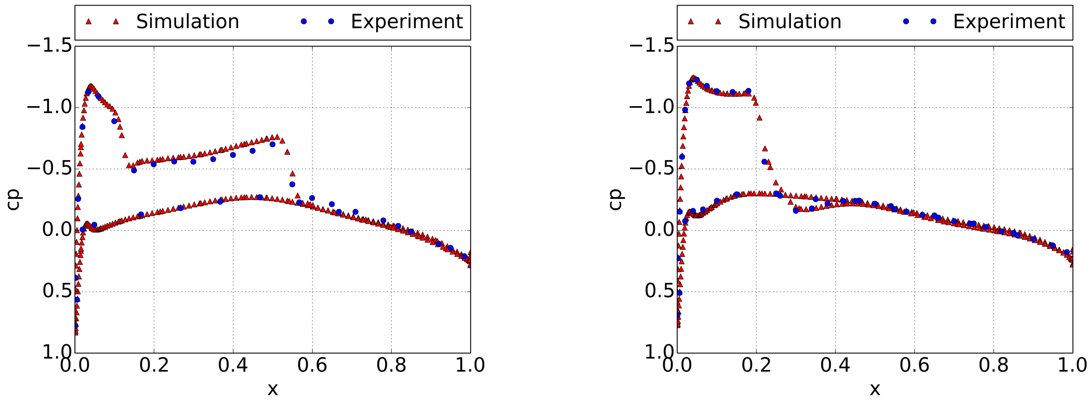


Figure 8.9: Mesh independence study for the M6 wing mesh

To verify the accuracy of the shock capturing using the TAU code, the M6 wing surface pressure profiles are compared to the experimental work done by Schmitt and Charpin [87]. Figure 8.10 shows a good agreement and shock capturing between the case computed using TAU and the experimental data.

Fig. 8.11 shows the surface sensitivity, $-\lambda_{mesh,z}$, on the upper surface of the M6 wing. The green, blue and yellow areas on the surface in Fig. 8.11 show regions where the gradient points in a negative z-direction. This implies that the drag reduces as the surface moves outwards (+z-direction). The red regions show areas with positive or very small derivative values. On the M6 wing, the sensitivity makes a clear lambda structure. The pressure plot confirms the lambda structure can be attributed to the shock locations. In addition to this there is a non-shock region near the wing root which suggests moving the surface in a positive z-direction in that region could also achieve a drag reduction.

The surface pressure and streamline plots are shown in Fig. 8.12. The surface pressure shows the lambda shock footprint on the wing surface but there is nothing at the wing root to suggest a potential region for drag reduction if the surface is



(a) 43.74%span

(b) 94.45%span

Figure 8.10: Comparison of surface pressure plots on the M6 wing surface between experiment and TAU simulation

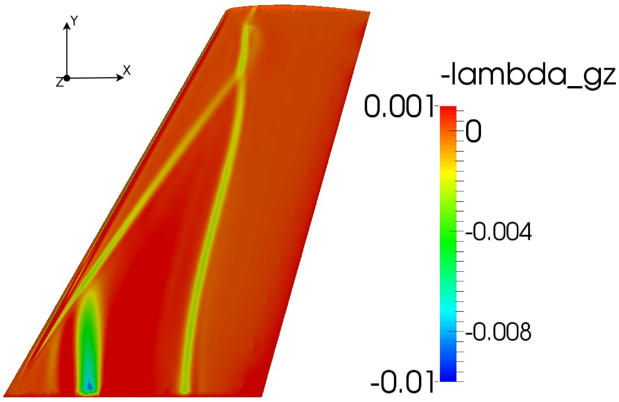
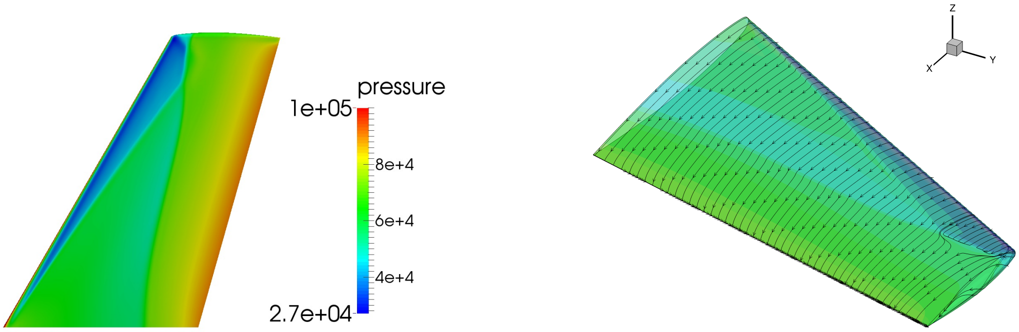


Figure 8.11: Surface sensitivity (λ_{mesh}) in the z-direction



(a) Surface pressure

(b) Surface streamlines

Figure 8.12: Surface pressure and streamlines on the M6 wing

distorted in this location. The surface streamlines show a small separation bubble at the wing tip where a strong sensitivity is evident in 8.11, however there is nothing at the wing root to suggest that a drag reduction can be achieved by distorting the wing surface.

Previous research [29] has focussed on optimising a shock bump array on the rear shock line only (neglecting near the tip). This work takes this further by applying bumps over the entire span of the wing. Furthermore, it explores surface changes in the front shock region and the sensitive non-shock region near the wing root.

8.2.1 *Rear Shock Leg*

The surface mesh points with large negative $\lambda_{mesh,z}$ values, shown in blue, green and yellow colour in Fig. 8.11. The rear shock line has the greatest potential for shock bump deployment as it shows the largest sensitivities of the two shock lines. First, the points with largest sensitivity in the rear shock line are extracted. Then the points are fitted and smoothed using 5^{th} order polynomials. The sensitivities feature line is then obtained for distributing the shock control bumps.

In order to achieve the best results from shock control, it is particularly important to involve the regions where the sensitivity is strongest. These regions are the wing tip and from the wing root to half span. The bumps are deployed starting at 0.83%span and ending at 99.17%span. 14 bumps are evenly distributed, which gives a bump width of 7%span. The bumps are distributed with no gaps in the spanwise direction. The optimisation region is defined by a front line and rear line, specifying the extension of the bumps. Previous research by Wong *et al.* [32] and Qin *et al.* [30] suggested that the shock control bump length should be between 20% to 40% chord length. For this work the bump length at any span position is 35% wing local chord.

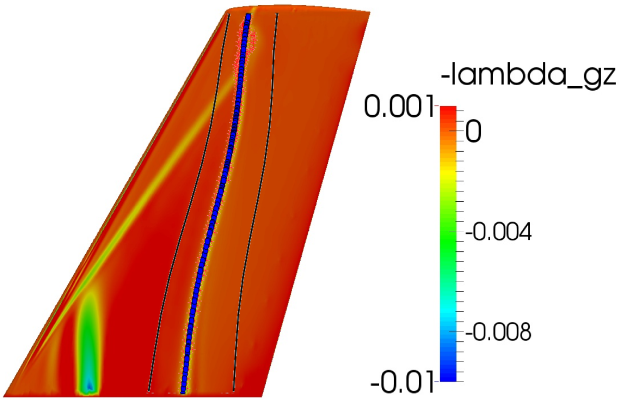


Figure 8.13: Extraction of rear shock sensitivity data with curve fitting and subsequent front and rear optimisation bounds

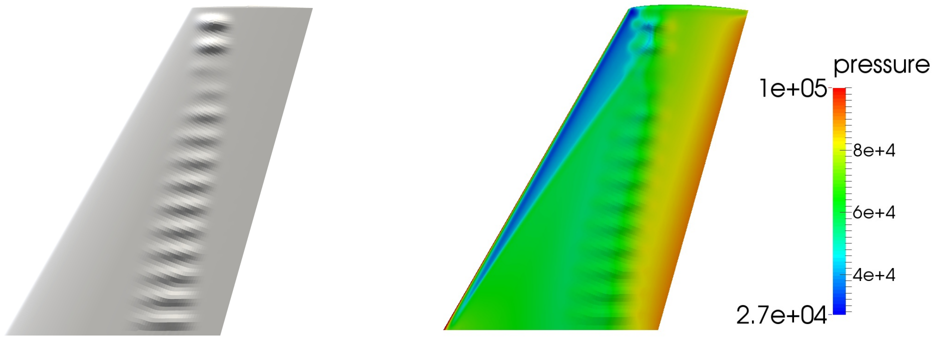
After the bump area is decided, the optimization can be carried out. In this case, the Bernstein polynomials for each bump have 3^{rd} order chordwise, 3^{rd} order spanwise, and 3^{rd} order of the height distribution. The length and width of the bumps are fixed, hence the total number of design variables is $[14 \times ((4 \times 4) + 4)] = 280$.

During the flow calculation the lift is constrained to guarantee the target C_l is matched. The objective function is then modified as:

$$I = C_d - \frac{\left(\frac{\partial C_d}{\partial \alpha}\right)}{\left(\frac{\partial C_l}{\partial \alpha}\right)}(C_l - C_{l,target}) \tag{8.1}$$

The lines in Fig. 8.15 show the spanwise wave drag for the baseline and optimised wings in drag counts (0.0001 of C_D). It is clear that the majority of the wave drag is localised around the tip and this is where the most drag reduction is found. Over the rest of the wing span the wave drag has been consistently reduced.

Further derivation of Eq. (8.1) as an objective function can be found in the paper by Reuther *et al.* [88]. Fig. 8.14(a) shows the topology of the bumps on the M6 wing for the final iteration in the optimisation. The red bars in 8.15 indicate the bump heights across the wing span as a percentage of local chord. It is obvious that at the tip the bumps are the tallest. This is what we would expect since the sensitivities in the positive z-direction at this location are very strong and this is where the most wave drag is found. It is also interesting to note that at the region where the 'legs' of the λ-shock concatenate, the shock bump height is very close to zero. In this region the possible reduction of wave drag for one of the shocks is offset by an increase of drag elsewhere and therefore the optimiser has driven the bump height to be smaller in this region. The highest bumps are found at the tip region on the wing where the wave drag is largest.



(a) Optimised bump topology

(b) Pressure distribution with optimised bumps

Figure 8.14: Surface distortion and pressure distribution on the upper surface of the M6 wing

Fig. 8.12 and 8.14(b) compare the surface pressure distributions on the upper surface of the M6 wing before and after optimisation. The wing with no bumps shows a very sharp change in pressure at the rear shock line which is smoothed in the case with the optimised bumps included. When the optimised bumps are placed on the wing the pressure change in the rear shock line region is much more gradual across the majority of the span. The pressure change has been reduced in most of the wing tip area but there is still a small region in between the bumps which contains a sharp change in pressure which is indicative of a small shock remaining.

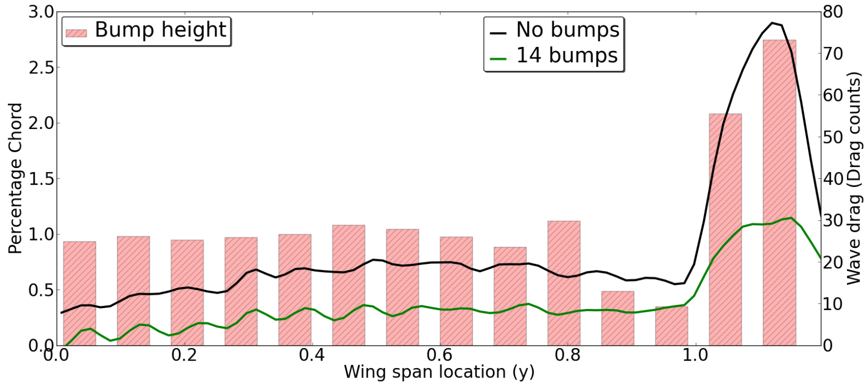


Figure 8.15: Percentage height distribution and spanwise wave drag

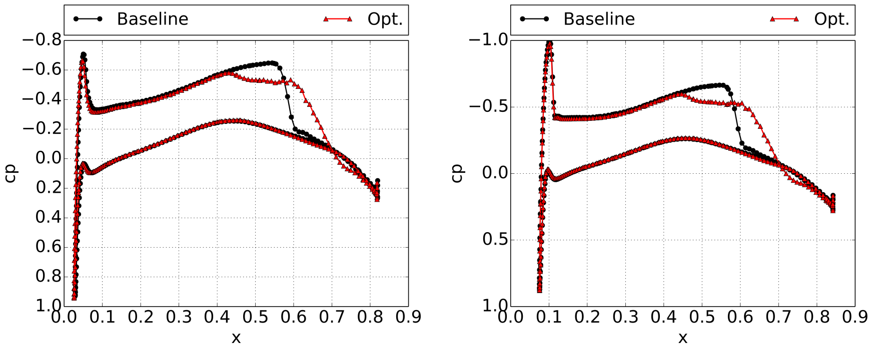
Fig. 8.16 shows spanwise slices along the M6 wing and the surface pressure distribution is plotted on each slice. For the 1st through to the 11th bump, the pressure distribution at the rear shock shows a much more gradual change compared to the baseline wing. This effect on the pressure distribution is typical of a well designed shock bump. The 12th bump is in the region where the front and rear shocks join. There is still a positive effect on the rear shock but it is much less significant in comparison to the bumps closer to the wing root. For bumps 13 and 14 at the wing tip, there is a strong re-expansion of the flow after the initial compression. However the pressure change after the re-expansion is not as severe. Between the 13th and 14th bumps, there is a strong re-expansion of the flow and the gradient of pressure after the re-expansion is still large, indicative of a shock wave.

Table 8.2 shows a comparison of the drag decomposition using a far field drag analysis tool. When the bumps are deployed, the drag is reduced by just under 19 drag counts. The table also shows that the pressure drag reduces while the skin friction remains nearly the same with a very slight increase, which is common when using shock bumps.

Table 8.2: Drag comparison between M6 wing with optimised shock bump along the rear shock line and baseline wing

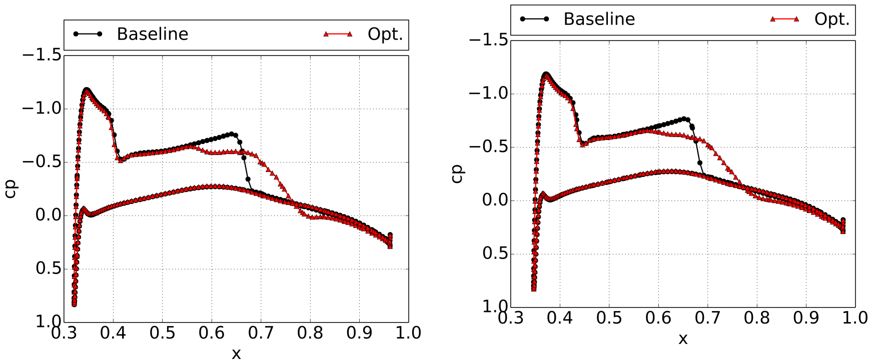
(drag counts)	C_d	$C_{d,wav}$	$C_{d,ind}$	$C_{d,viscpr}$	$C_{d,fric}$	$\frac{C_l}{C_d}$
Original	175.87	28.69	62.24	31.13	52.69	15.55
14 bumps	156.91	12.81	62.31	27.80	53.56	17.42
Reduction	-10.78%	-55.35%	0.11%	-10.70%	1.65%	12.03%

Figs. 8.11 and 8.17 compare the surface sensitivity for the upper surface of the original M6 wing and after the bump optimisation is applied. The sensitivity map after the optimisation can be used to evaluate the effectiveness of the drag optimisation. For most of the span the sensitivity of drag to changes in the surface has been reduced significantly in the rear shock area.



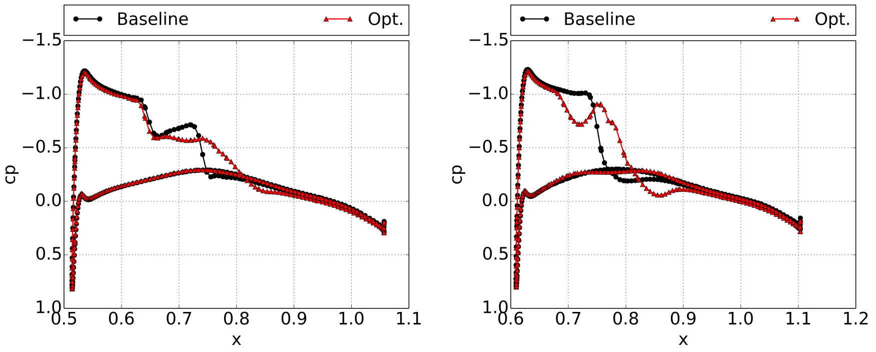
(a) 1st bump (4.33%span)

(b) 2nd bump (11.33%span)



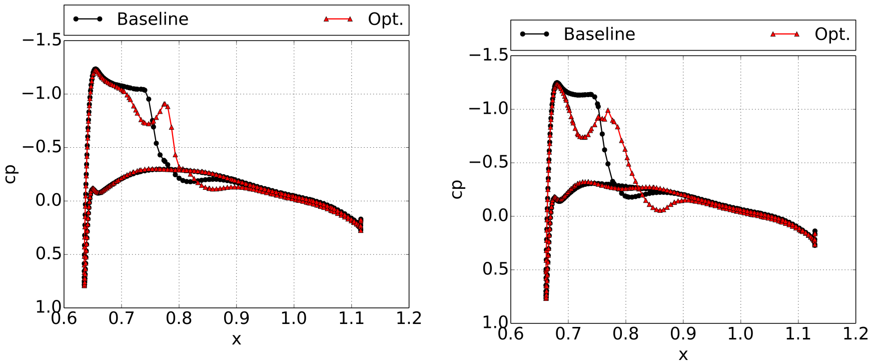
(c) 7th bump (46.33%span)

(d) Between 7th and 8th bump (49.83%span)



(e) 11th bump (74.33%span)

(f) 13th bump (88.33%span)



(g) Between 13th and 14th bump (91.83%span)

(h) 14th bump (95.33%span)

Figure 8.16: Streamwise surface slices showing the pressure distribution on the surface

The only area that still shows a relatively strong sensitivity in the optimisation region is between the two bumps at the tip of the wing. A possible explanation for this is that the shock is so strong in this location that the bumps cannot grow high enough to treat it without incurring a drag penalty elsewhere. It is interesting to note that in this comparison the non-shock sensitivity region is reduced after the bump optimisation despite no optimisation taking place in the local vicinity. This is not surprising since the flow over the surface in the rear shock and non-shock region are linked for subsonic flows.

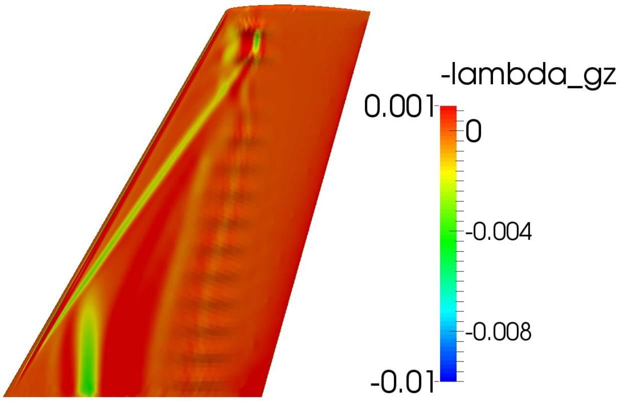


Figure 8.17: Surface sensitivity for final iteration of optimisation

8.2.2 *Choice of the Number of Spanwise Bumps*

In the previous section, 14 spanwise bumps have been deployed. To test the sensitivity of bump width as a parameter, several values of width are tested in this section. The span of the wing is divided into equal subdivisions such that the bumps cover the same optimisation region. The bump widths used are 12.08%span, 7.42%span, 6.92%span, 6.5%span and 4.83%span (relating to 8,13,14,15 and 20 bumps respectively). The bumps are equally distributed along the wing span. The bump array is optimised for each bump width as in the previous section, to reduce drag.

Figure 8.18 shows that bumps are present in all the case in the rear-shock line region. The surface sensitivity for the final iteration of the optimisation shows that the sensitivity is reduced in a similar manner for each bump width.

Figure 8.19 shows the height distribution across the span for the optimised bumps for all cases and the wave drag distributions. The height distributions all show the same trend of around 1% local chord height of bumps between 0%span and 75%span.

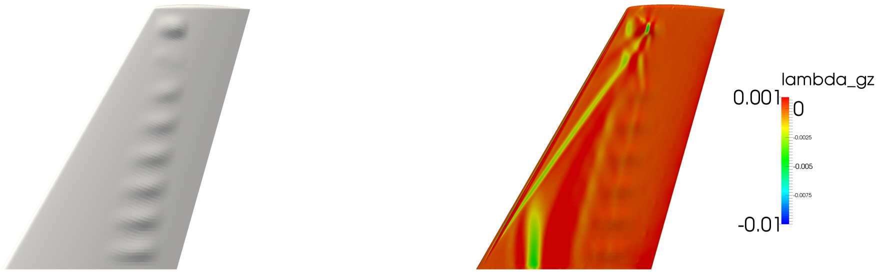
At the location where the two shock lines concatenate (between 75%span and 85.83%span) the bumps are consistently lower for all cases with a bump height of less than 0.5%c. The highest bumps are found at the tip region (between 85.83%span and 100%span) where the most wave drag is found. The spanwise wave drag distribution in figure 8.20, shows that the wave drag is being reduced by approximately the same amount across the entire wing span.

Table 8.3 shows the drag decomposition from each of the cases considered in this section with optimised bumps. σ represents the standard deviation of the drag values in that row. There is very little difference in the final drag values, this is reinforced by the very small standard deviation of the drag decomposition.

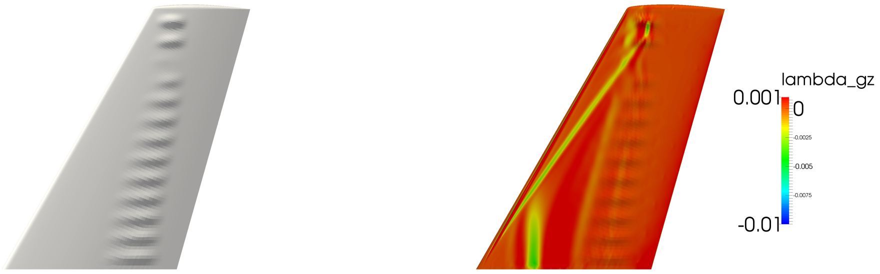
Table 8.3: Drag comparison of the M6 wing with different spanwise numbers of optimised bumps

No. Bumps	8	13	14	15	20	σ
($\times 10^{-4}$)						
C_d	156.11	156.24	156.48	156.16	156.49	0.1598
$C_{d,wav}$	12.49	12.58	12.81	12.50	12.84	0.1513
$C_{d,ind}$	62.32	62.31	62.31	62.30	62.24	0.0287
$C_{d,viscpre}$	27.83	27.79	27.80	27.76	27.79	0.0225
$C_{d,fric}$	53.47	53.56	53.56	53.59	53.62	0.0502

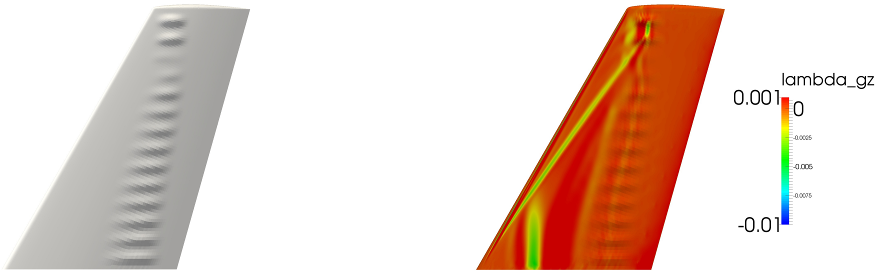
The results from this section show that the bump width is an insensitive parameter and the choice of bump width will not effect the final drag reduction that can be achieved by the shock bump array.



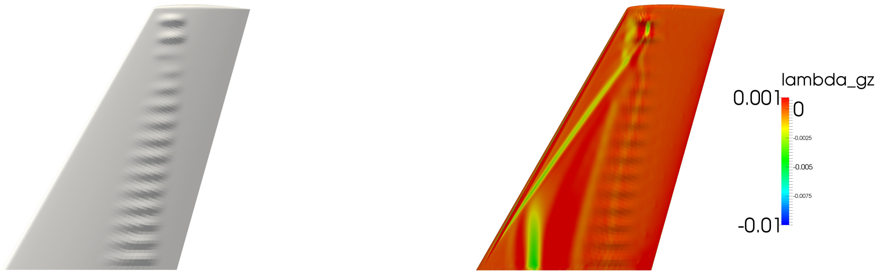
(a) 8 bumps



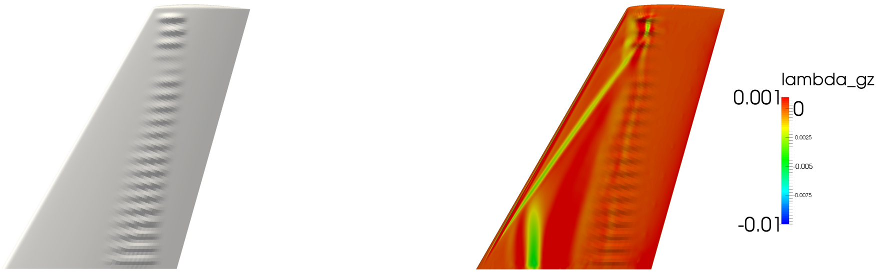
(b) 13 bumps



(c) 14 bumps

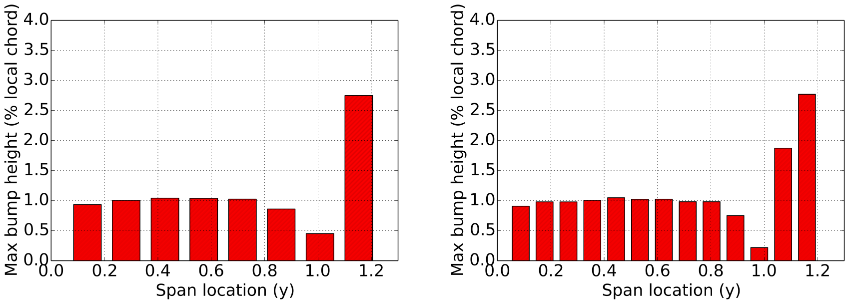


(d) 15 bumps

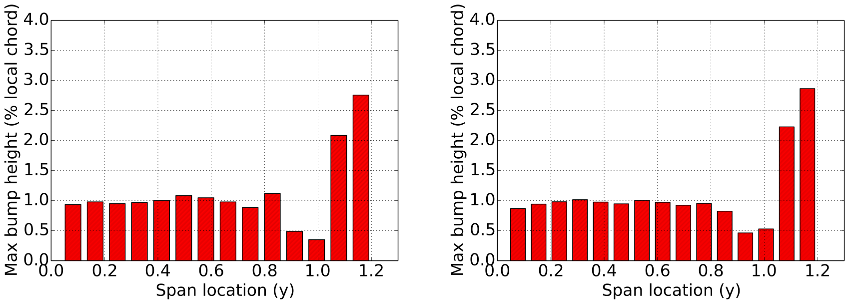


(e) 20 bumps

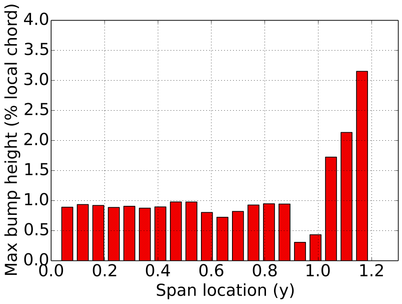
Figure 8.18: Comparison of different spanwise bump numbers and the effect on λ_{gz}



(a) Bump heights for 8 bumps (b) Bump heights for 13 bumps



(c) Bump heights for 14 bumps (d) Bump heights for 15 bumps



(e) Bump heights for 20 bumps

Figure 8.19: Spanwise distribution of bumps and wave drag for varying bump width

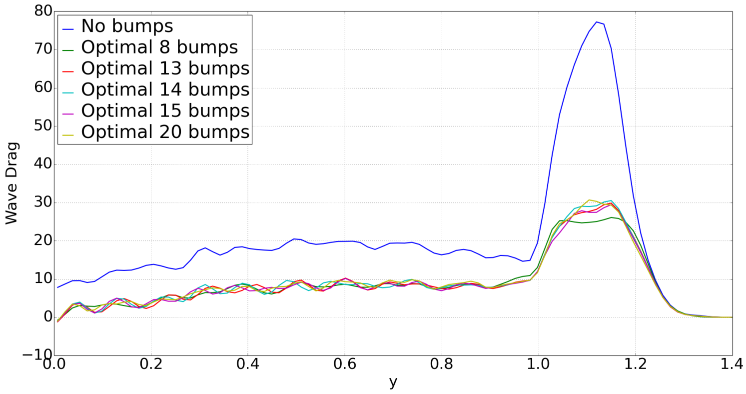


Figure 8.20: Spanwise wave drag distribution for varying bump width of M6 wing measured in drag counts

8.2.3 *Front Shock Leg*

The previous section focusses on the application of shock bumps in the rear shock leg region across the entire wing span. This section will focus on the front shock leg, near the leading edge.

Fig. 8.21 shows the optimisation region considered for this case. Due to the proximity of the leading edge, the bump region chosen is shorter in the streamwise direction compared to that of the previous section. The optimisation region is 25% local chord in streamwise length, within the boundaries suggested by Wong *et al.* [32] and Qin *et al.* [30]. There will be 8 bumps deployed with a bump width of 4.6%span each. The optimisation region begins at 41.67%span and ends at 79.17%span. As with the previous test case, each bump will have 40% before the feature line and 60% after.

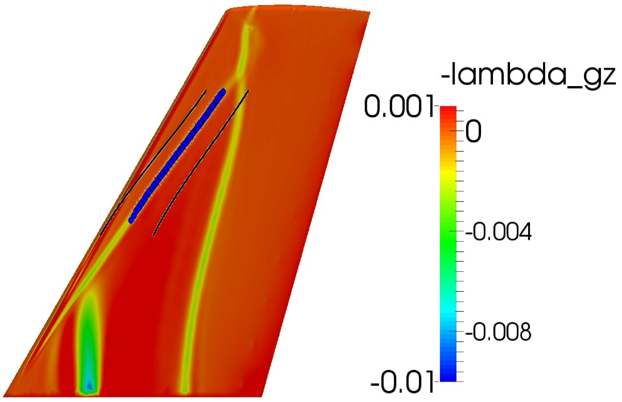
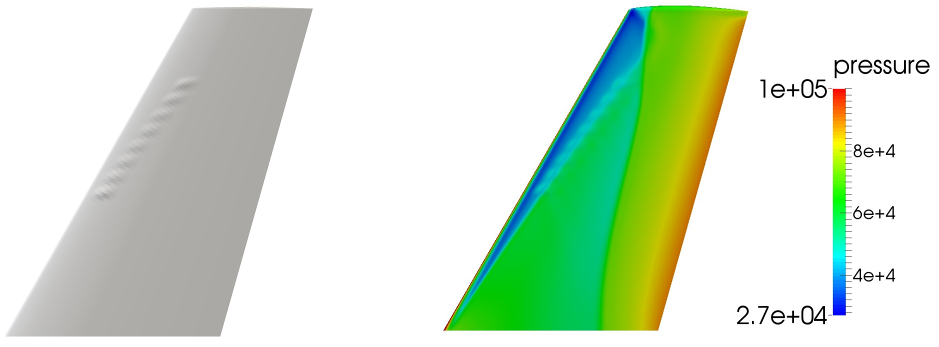


Figure 8.21: Extraction of the front shock leg sensitivity data with curve fitting and subsequent front and rear optimisation bounds

Fig. 8.22(a) shows the final optimised bump array on the front shock leg region. Although bumps have established themselves in this region for drag reduction, the height of the bump is small-in comparison to the rear shock control bumps. In addition to the small drag reduction in the immediate control region the wave drag is increased in other areas across the span, offsetting the drag reduction potential. The wave drag increase is due to a slight increase in the rear shock strength along the wing span. The overall wave drag reduction is limited by deploying bumps in this region.

Figs. 8.12 and 8.22(b) compare the pressure distributions with no bump and with optimised bumps. The bumps here have had a small effect on weakening the pressure peak in the shock location but there still remains a significant pressure change in the front shock line region.

Fig. 8.24 shows spanwise slices of the wing with the surface pressure distribution plotted at each slice. Between the 1st and 4th bumps there is little effect on the front shock. Bumps 5 through 8 shows a more visible effect on the front shock leg with a much more gradual pressure change in the shock region. For the entire



(a) Optimised bump topology

(b) Pressure distribution with optimised bumps

Figure 8.22: Surface distortion and pressure distribution on the upper surface of the M6 wing

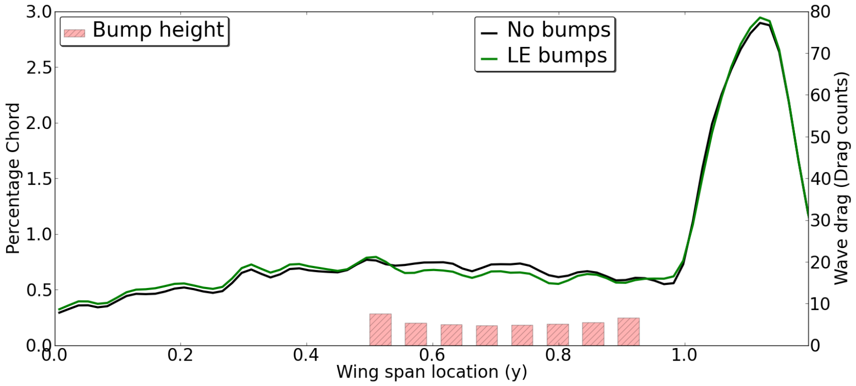


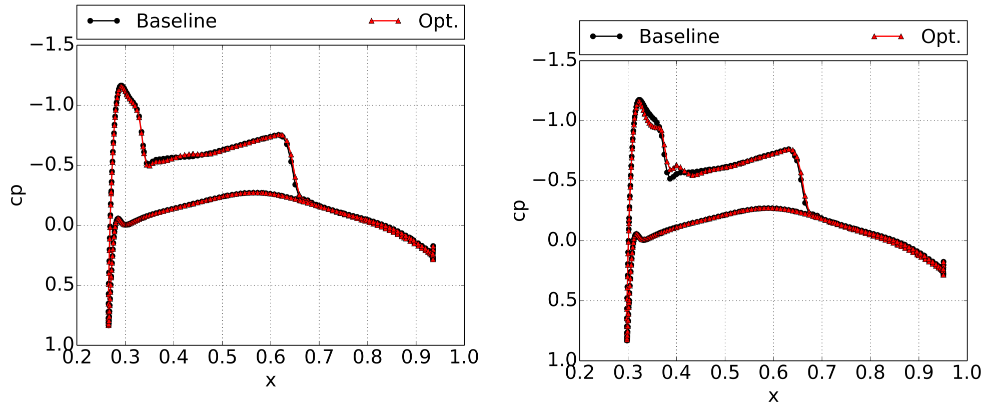
Figure 8.23: Percentage height distribution from root (left) to tip (right)

shock array there is no effect downstream at the rear shock. The bottom two pressure distributions show that there is very little effect on the flow from the bump control region.

Table 8.4 shows the comparison of the far field drag decomposition. This shows that the application of the shock bumps in this region has only had a small effect on drag reduction as the bumps were placed in a region where the sensitivity was not as strong.

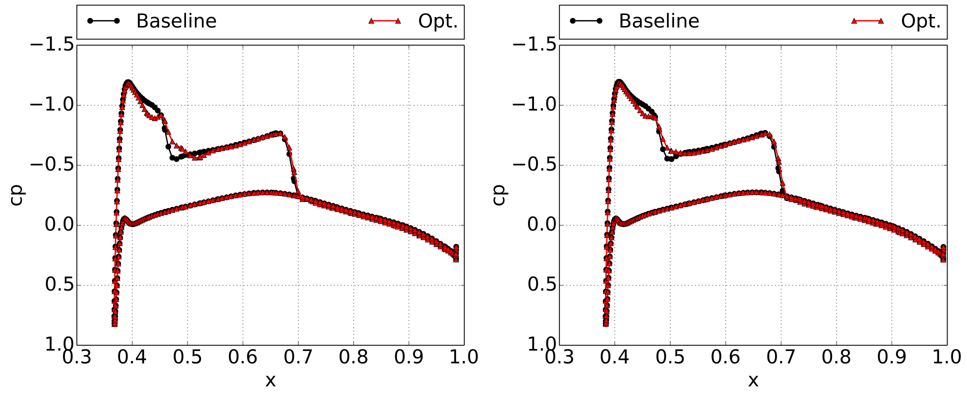
Table 8.4: Drag comparison between M6 wing with optimised shock bump along front shock line and baseline wing

(drag counts)	C_d	$C_{d,wav}$	$C_{d,ind}$	$C_{d,viscpre}$	$C_{d,fric}$	$\frac{C_l}{C_d}$
Original	175.87	28.69	62.24	31.13	52.69	15.55
LE bumps	173.71	28.61	62.26	30.08	52.76	15.74
Reduction (%)	-1.23%	-0.28%	0.03%	-3.37%	0.13%	1.22%



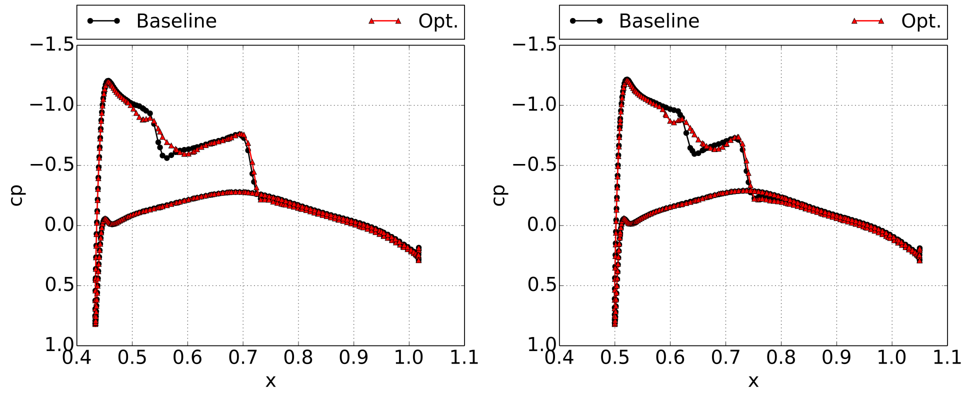
(a) 1 bump width toward the wing root (37.07%span)

(b) 1st bump (43.97%span)



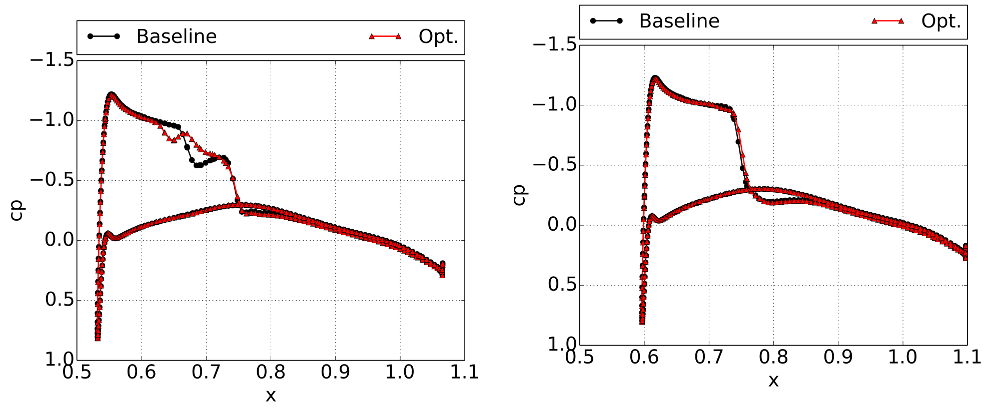
(c) 3rd bump (53.17%span)

(d) Between 3rd and 4th bump (55.47%span)



(e) 5th bump (62.37%span)

(f) 7th bump (71.57%span)



(g) 8th bump (76.17%span)

(h) 2 bump widths toward the wing tip (87.67%span)

Figure 8.24: Streamwise surface slices showing the pressure distribution on the wing surface

Figs. 8.11 and 8.25 compare the surface sensitivity in the z-direction with no bumps and with the optimised bumps. Again the optimised bumps have had an effect on the sensitivity in the optimisation region. However, a slightly stronger sensitivity is created closer to the wing root.

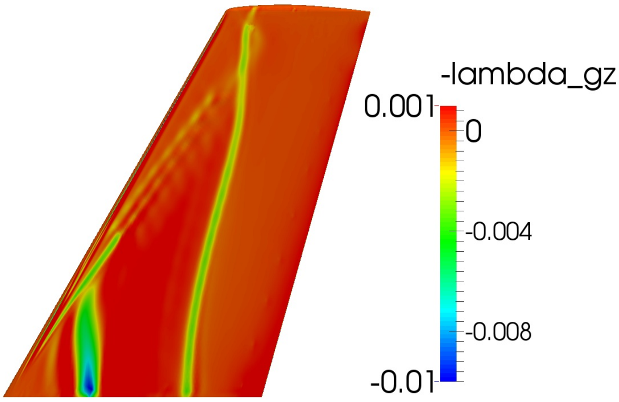


Figure 8.25: Surface sensitivity for final iteration of optimisation

8.2.4 *Non-Shock region*

This paper so far has focussed on optimisations in the shock regions and this section will look into an area where there is no shock but the sensitivity map suggests that a change in the surface in the positive z-direction will have a strong effect on drag reduction.

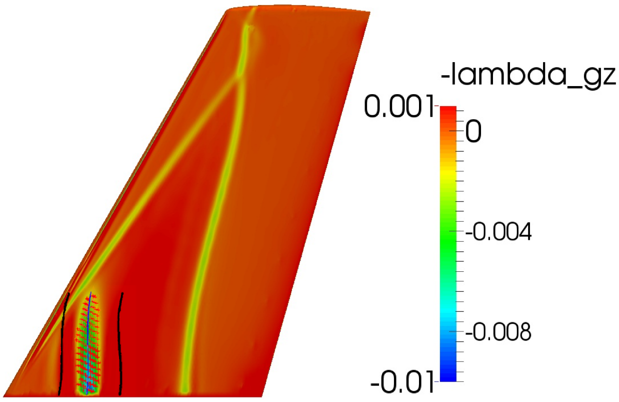


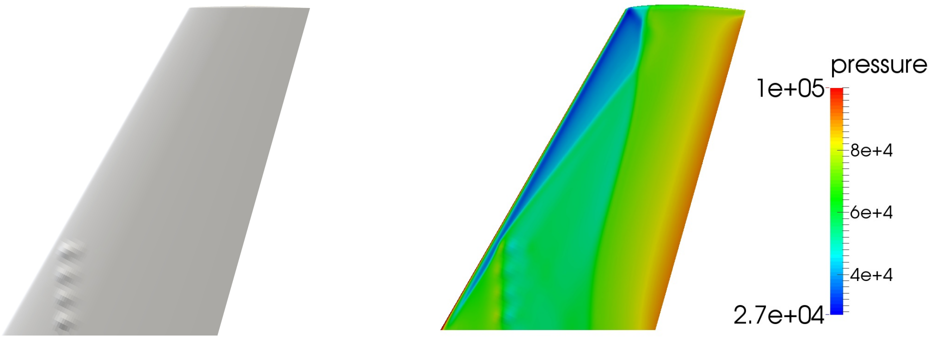
Figure 8.26: Extraction of non-shock sensitivity data (at the wing root) with curve fitting and subsequent front and rear optimisation bounds

Fig. 8.26 shows the optimisation region considered for this case. In agreement with the sensitivity map, the optimisation region is chosen to be 25% local chord in streamwise length. There are 4 bumps deployed with a bump width of 7.08%span.

The optimisation region begins at 0.83%span and ends at 29.17%span. As with the previous test case, each bump will have 40% before the feature line and 60% after.

Fig. 8.27(a) shows the topology of the optimised bumps. All 4 non-shock control bumps have grown to a similar height at approximately 0.5% local chord. It is interesting to note that although the bumps are located in the non-shock region, a strong effect on the shock wave strength, and therefore wave drag, is observed. The effect is also extended to a much wider region in the spanwise direction (> 33.33%span).

When the bumps are present the wave drag is reduced from the root to 75%span. At 75%span, the wave drag dramatically increases due to the concatenation of the front and rear shock legs. This concatenation point is further from the tip than in the no bump case and causes the wave drag at the tip to be slightly stronger.



(a) Optimised bump topology

(b) Pressure distribution with optimised bumps

Figure 8.27: Surface distortion and pressure distribution on the upper surface of the M6 wing

Figs. 8.12 and 8.27(b) show the surface pressure distribution on the upper surface of the M6 wing. The introduction of the optimised bumps has caused an increase in pressure in the local region and directly downstream of the bumps the pressure change is much less dramatic. In addition to this, the bump furthest out in the span is acting slightly on the front shock line.

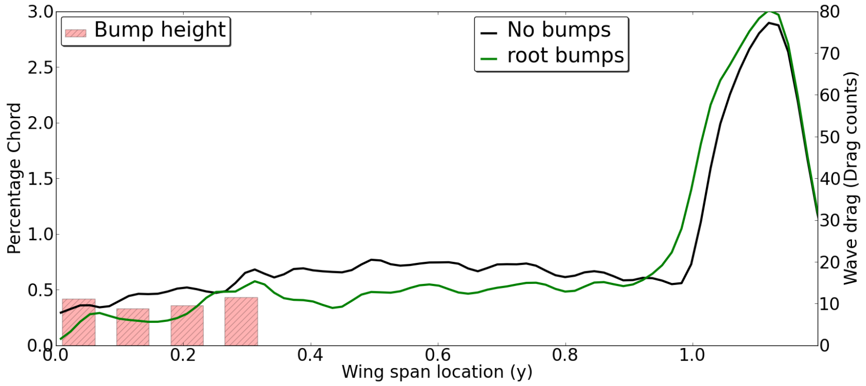
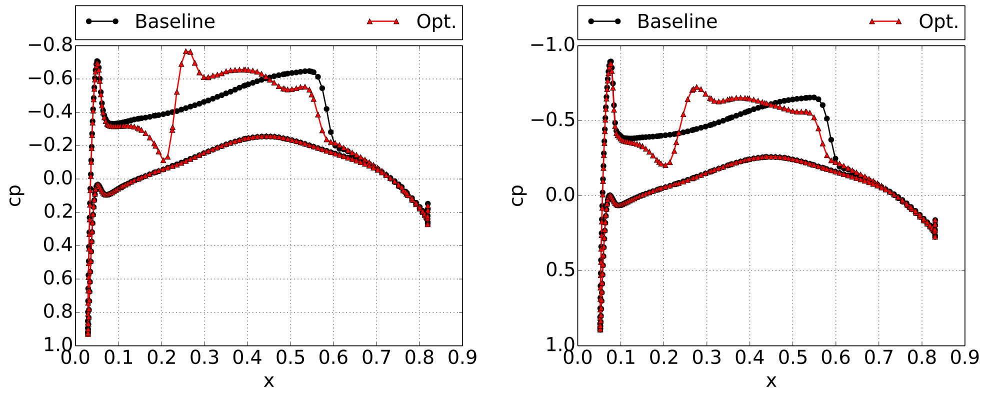
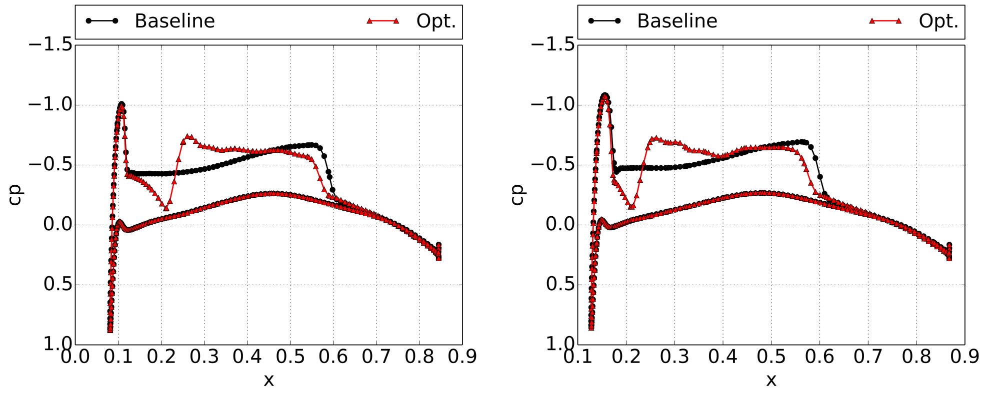


Figure 8.28: Percentage height distribution and spanwise wave drag



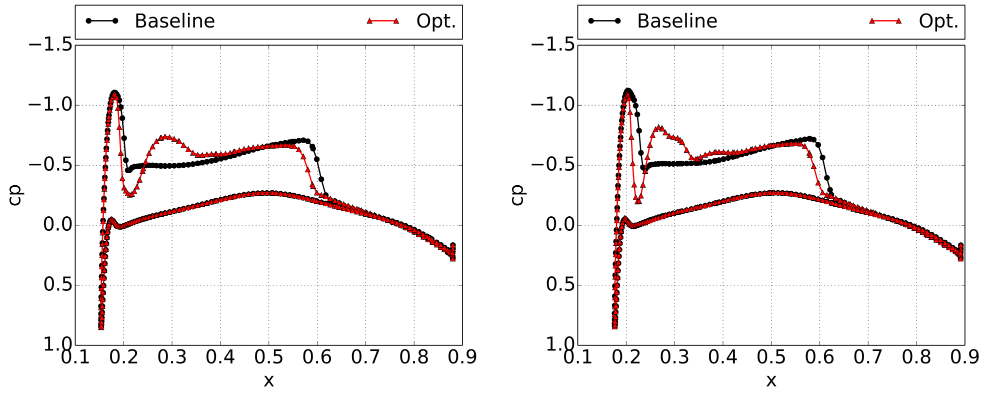
(a) 1st bump (4.37%span)

(b) Between 1st and 2nd bump (7.91%span)



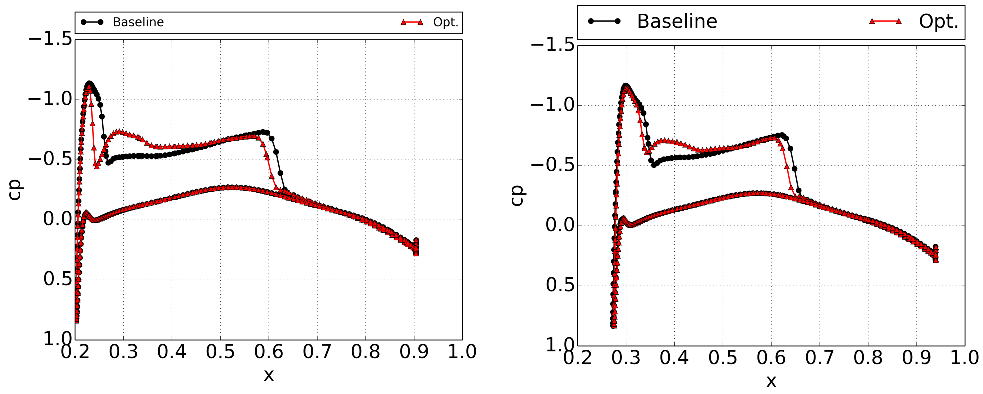
(c) 2nd bump (11.45%span)

(d) 3rd bump (18.53%span)



(e) Between 3rd and 4th bump (22.07%span)

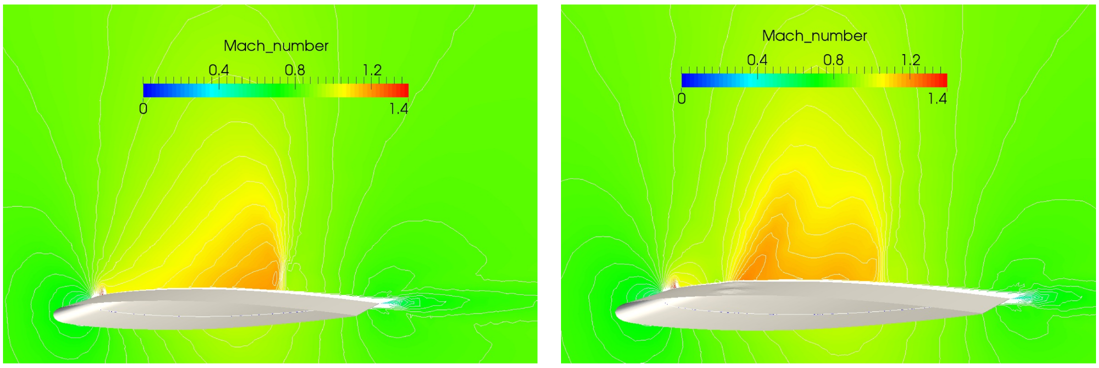
(f) 4th bump (25.61%span)



(g) 0.5 bump widths towards the wing tip (32.69%span)

(h) 2 bump widths towards the wing tip (43.31%span)

Figure 8.29: Streamwise surface slices showing the pressure distribution on the surface



(a) Original

(b) Optimised non-shock region

Figure 8.30: Pressure contours on a slice at 12.5%span

The surface pressure at various slices along the wing span, shown in Fig. 8.29, indicate that there is a local compression-expansion-compression caused by the placement of bumps 1, 2 and 3. Directly downstream of these bumps the pressure reduces further. Immediately before the shock the pressure has reduced enough such that the shock is weaker in comparison to the case with no control. The placement of bump 4 has forced extra compression of the flow around the front shock line. The placement of these bumps has moved the rear shock line slightly upstream of its original location. This causes the two shock lines to join earlier. Directly downstream of the optimisation region the shock has been weakened.

Fig. 8.30 shows a comparison of pressure contours from a slice at 12.5%span between the original wing and the non-shock region optimisation. The non-shock bumps cause a compression-expansion-compression in the flow which creates a region of low pressure between the compression and the rear shock. At the shock, the pressure has decreased below that of the baseline case and therefore has a weaker shock.

Table 8.5 shows the comparison of the far field drag decomposition. The drag reduction after the bumps are deployed is 4 drag counts and interestingly the skin friction change is trivial.

Table 8.5: Drag comparison between M6 wing with optimised shock bumps in non-shock region and baseline wing

(drag counts)	C_d	$C_{d,wav}$	$C_{d,ind}$	$C_{d,viscpre}$	$C_{d,fric}$	$\frac{C_l}{C_d}$
Original	175.87	28.69	62.24	31.13	52.69	15.55
Non-shock bumps	170.68	26.01	62.21	29.80	52.66	16.02
Reduction	-2.95%	-9.34%	-0.05%	-4.27%	-0.057%	3.02%

Figs. 8.11 and 8.31 show the surface sensitivity in the z-direction for the original M6 wing and the wing with optimised bumps. After the bumps are deployed the sensitivity downstream of the bumps at the rear shock line has been reduced. The sensitivity in the non-shock region has been reduced significantly but the sensi-

tivity map does suggest that increasing the bump length may give even more drag reduction.

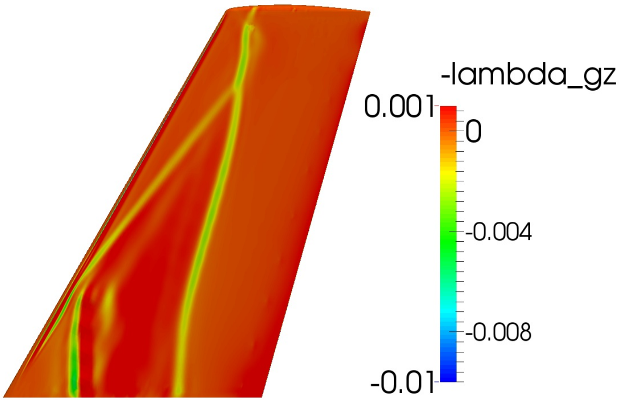


Figure 8.31: Surface sensitivity for final iteration of optimisation

8.2.5 *Entire sensitivity region optimisation*

The previous sections have looked at individual regions for optimisation of bumps. In order to investigate the combined effect of shock control and non-shock control bumps for all the drag sensitive regions bumps will be placed and optimised in all the sensitivity regions in this section.

Fig. 8.32 shows the optimisation region considered for this case. The optimisation region for the leading edge is 25% local chord in streamwise length and 35% for the rear region. There will be 25 bumps deployed, 9 in the front and rear regions and 7 in the concatenated region near the wing tip. The bumps in the front and rear shock leg regions have a width 6.88%span and the bumps in the concatenated region have a width 5.2%span. The optimisation region begins at 0.83%span and ends at 99.17%span. As with the previous test case, each bump will have 40% before the feature line and 60% after. The concatenated region begins at 62.75%span.

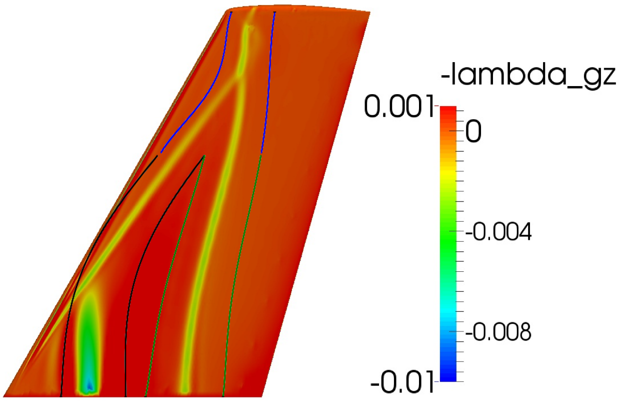
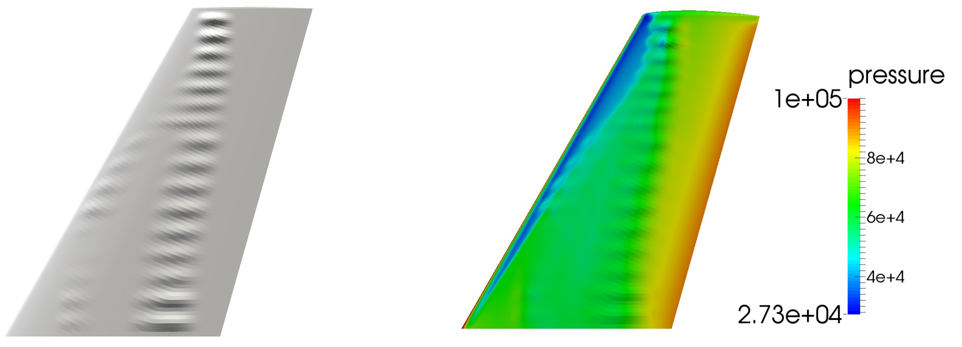


Figure 8.32: Extraction of sensitivity data with curve fitting and subsequent front and rear optimisation bounds

Fig. 8.33(a) show the topology of the optimised bumps, the bump heights for each optimisation region and the comparison of wave drag between the current case, the no bump case and the rear-shock line case. For the front shock leg region and rear shock leg optimisation region the bump patches are in line in the streamwise direction and, therefore, the bump heights are offset from each other in the bar chart for clarity. fig. 8.33 shows the spanwise wave drag is reduced a little more in the mid span of the wing compared to the rear-shock line case. However, this is offset by the increase in wave drag toward the tip region.

Figs. 8.12 and 8.33(b) compare the surface pressure distribution on the upper surface of the M6 wing. The introduction of the optimised bumps shows a much more gradual pressure change in the rear shock region, however the bumps in the non-shock region have not shown a large effect on the pressure. This may be due to the weakening of the sensitivity in this region, shown in the final iteration of the rear shock line optimisation.



(a) Optimised bump topology

(b) Pressure distribution with optimised bumps

Figure 8.33: Surface distortion and pressure distribution on the upper surface of the M6 wing

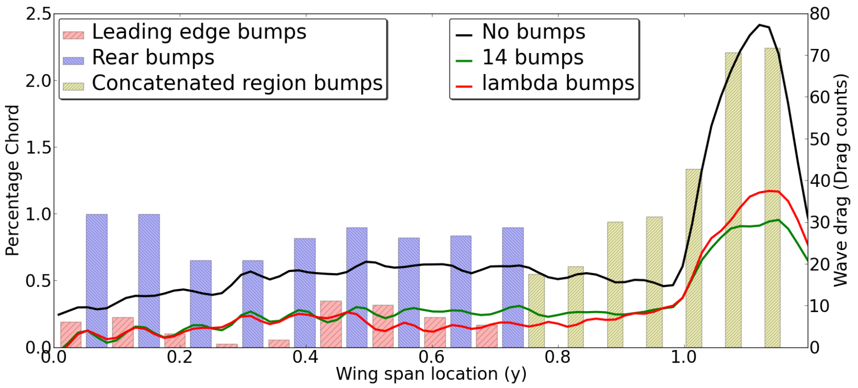


Figure 8.34: Percentage height distribution and spanwise wave drag

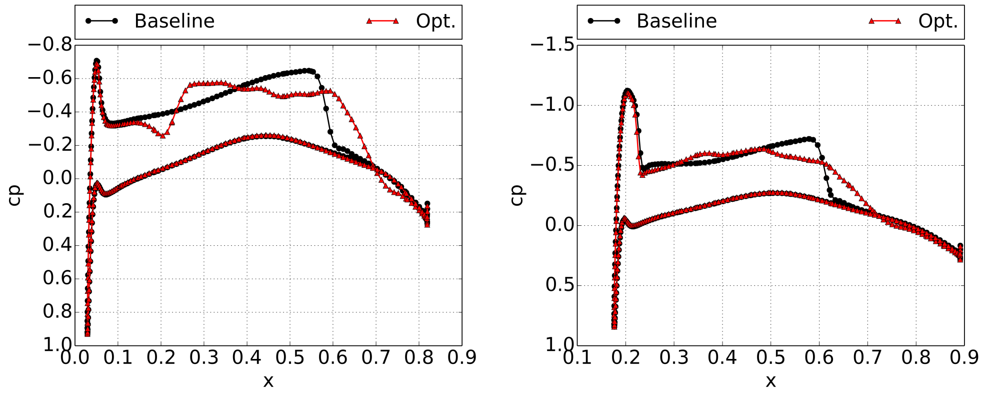
Fig. 8.35 shows spanwise slices along the wing where the surface pressure has been plotted. For the 1st bump the effect from the non-shock region bump causes a small compression and expansion in the flow.

The bumps further downstream act on the rear shock by making the pressure change much more gradual. By the 4th bump, the effect of the bump in the non-shock region is negligible, this is due to the small height of the bump. From the 6th to the 11th bump the pressure distribution at the front and rear shock is either much more gradual or the pressure peak before the shock is significantly reduced.

For bumps 14 through 16 the pressure distribution exhibits a small re-expansion of the flow, however the pressure peak does not return to the strength of the baseline case.

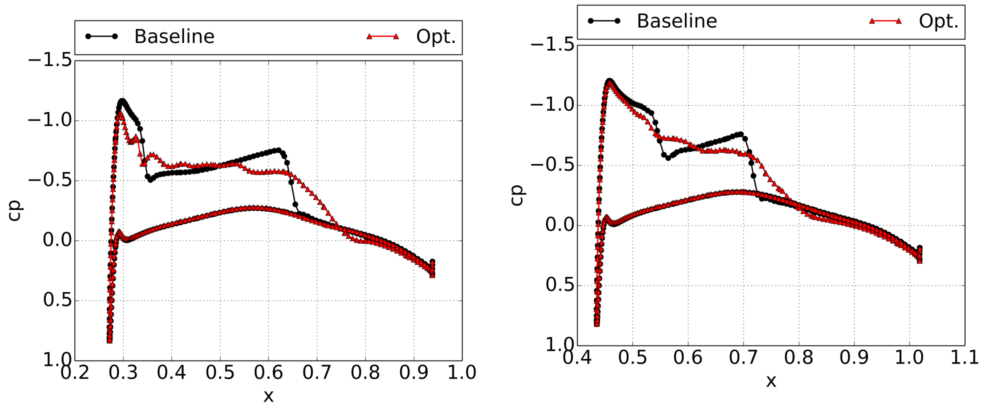
Table 8.6 shows the comparison of the far field drag decomposition. Comparing Tables 8.6 and 8.4, the overall drag reduction is 1 drag count better than using optimised bumps in the rear-shock region alone.

Comparing Figs. 8.12 and 8.36 the change in the surface sensitivity in the z-direction for the wing with optimised bumps can be seen. After the bumps are de-



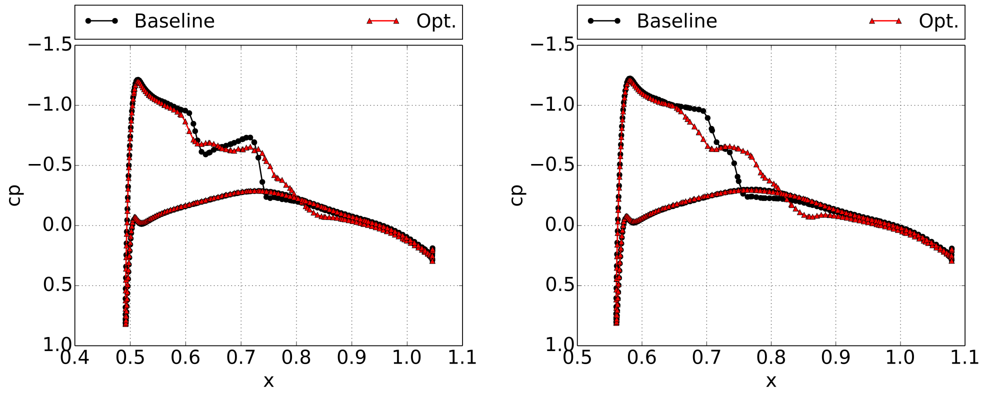
(a) 1st bump (4.27%span)

(b) 4th bump (24.91%span)



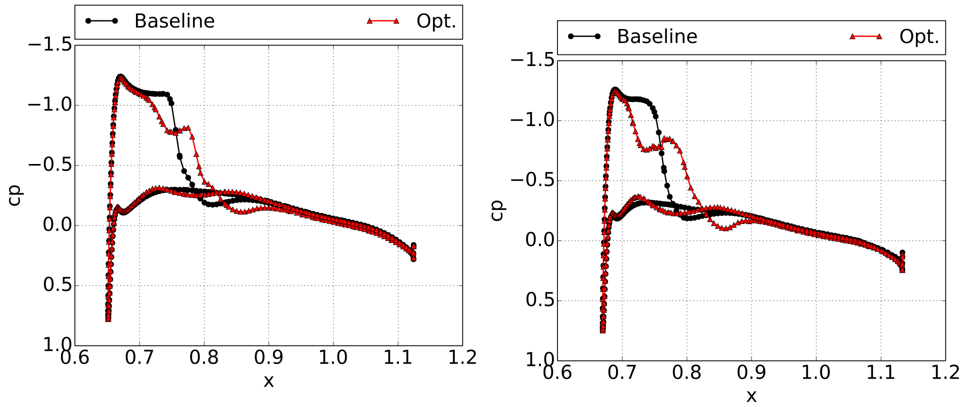
(c) 6th bump (38.67%span)

(d) Between 9th and 10th bump (62.75%span)



(e) 11th bump (70.55%span)

(f) 13th bump (80.95%span)



(g) Between 15th and 16th bump (93.95%span)

(h) 16th bump(96.55%span)

Figure 8.35: Streamwise surface slices showing the pressure distribution on the surface

Table 8.6: Drag comparison between M6 wing with optimised shock bumps in all sensitivity regions and baseline wing

(drag counts)	C_d	$C_{d,wav}$	$C_{d,ind}$	$C_{d,viscpre}$	$C_{d,fric}$	$\frac{C_l}{C_d}$
Baseline	175.87	28.69	62.24	31.13	52.69	15.55
14 bumps	156.91	12.81	62.31	27.80	53.56	17.42
Baseline Reduction	-10.78%	-55.35%	0.11%	-10.70%	1.65%	12.03%
All regions	155.33	12.39	62.33	26.93	53.68	17.60
Baseline Reduction	-11.68%	-56.81%	0.15%	-13.49%	1.88%	13.18%

ployed the sensitivity downstream of the bumps at the rear shock line has been reduced. The sensitivity in the non-shock region has been reduced significantly.

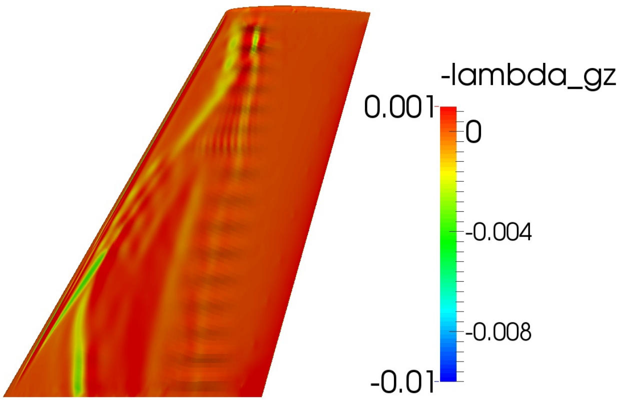
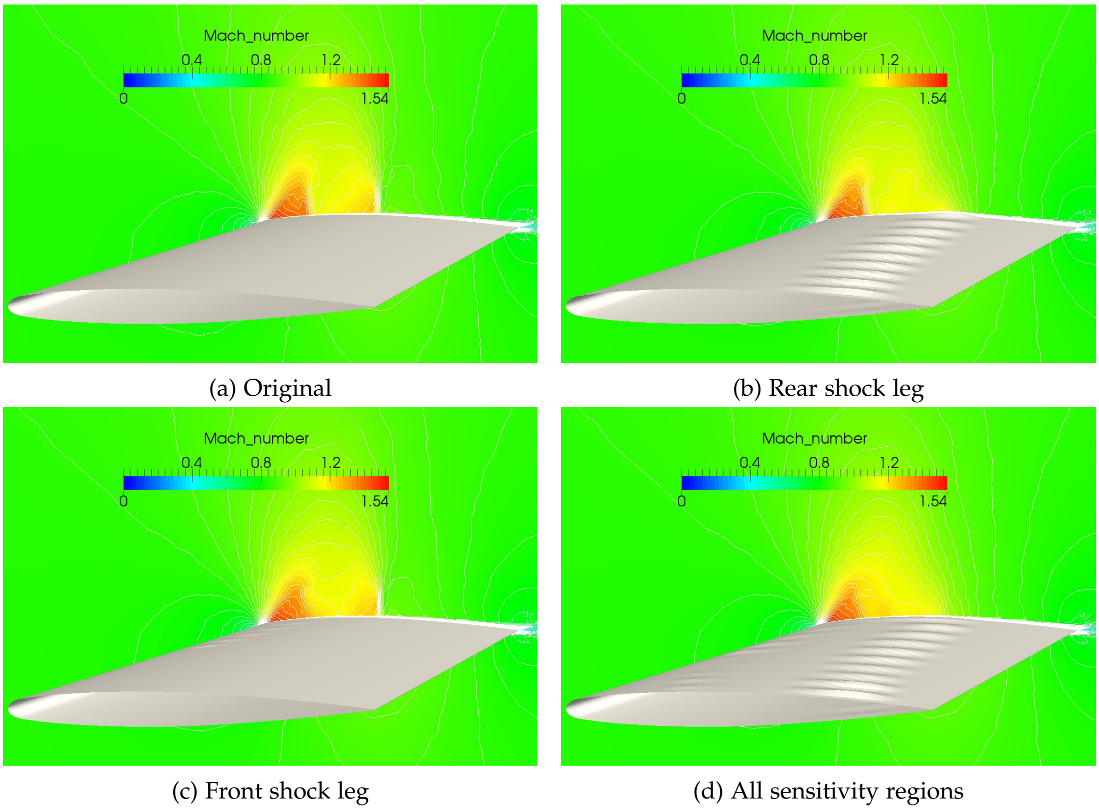


Figure 8.36: Surface sensitivity for final iteration of optimisation

This integrated study indicates that the most effective control for drag reduction on the M6 wing is to distribute shock control bumps along the rear shock leg of the λ-shock wave. In this case, adding bumps in the other two regions has very little benefit.

Fig. 8.37 shows a slice of the flow domain at 66.67%span. Pressure contours are plotted on the slices. For all optimised bump cases the pressure around the control region changes more gradually and it is clear that the effect on the pressure from the shock bumps is local. This can be observed from a comparison between the front and rear shock legs to the case where all sensitivity regions are considered.



(a) Original

(b) Rear shock leg

(c) Front shock leg

(d) All sensitivity regions

Figure 8.37: Pressure contours for a slice at 66.67%span

8.2.6 *M6 Shock Bump Robustness*

One of the major drawbacks of shock control bumps discussed in the literature is that they could be sensitive to changes in the shock location due to changes in the flight conditions, such as Mach number, the incidence or lift conditions [4, 34, 89]. Qin *et al.* [27] showed that the 3D bumps design used in this paper is more robust than the corresponding 2D bump when the lift is varied at design Mach number.

The robustness of the optimised bumps from the rear-shock line case are analysed in this section as this region has been identified as the most important area for drag reduction and is expected to have the greatest variation in shock location.

Fig. 8.38 shows the drag values plotted against varying Mach number with fixed $C_L = 0.274$ for the M6 wing with no bumps and with the optimised 3D bumps in the rear shock region obtained in section 8.2.1. The 3D bumps achieve a drag reduction in the Mach range $[0.824, 0.865]$, which is around the design Mach number. For lower Mach numbers there is very little drag penalty for the optimised bump array. For increasing Mach number, above $M = 0.865$, there is a drag penalty when the bump array is present. This shows a good overall robustness in the neighbourhood of the cruise Mach number, allowing speeds of up to 4% over and 2% lower than the cruise speed without incurring a penalty for design C_L.

When shock bump robustness is referred to in the literature it is often the case that a movement in the shock location relative to the bump can cause a stronger shock

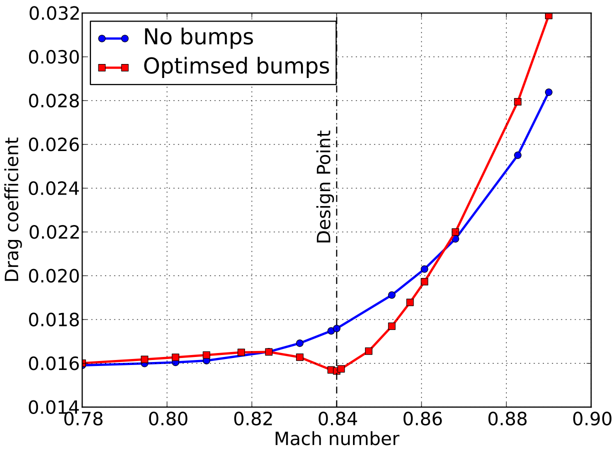


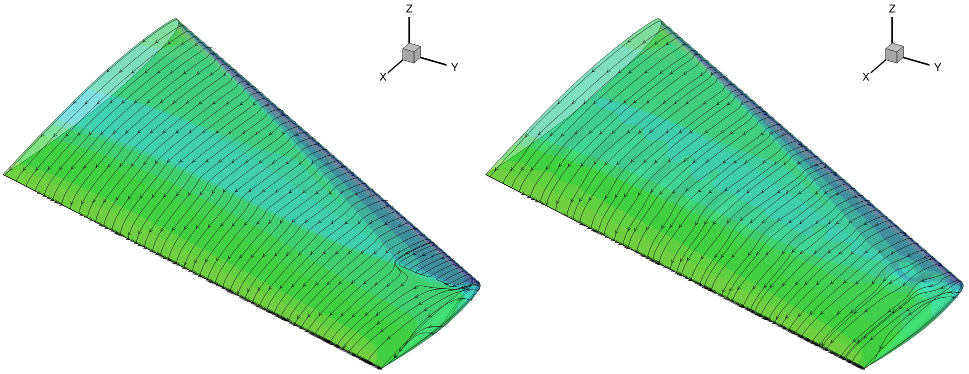
Figure 8.38: Drag coefficients for changes in Mach number around the design point.

or cause multiple shocks or separation to occur [5]. This infers that the dramatic rise in drag after the design Mach number can be attributed to an increase in wave drag. Figure 8.39 shows the surface streamlines for skin friction on the surface of the M6 wing at the Mach numbers around the dramatic increase in drag.

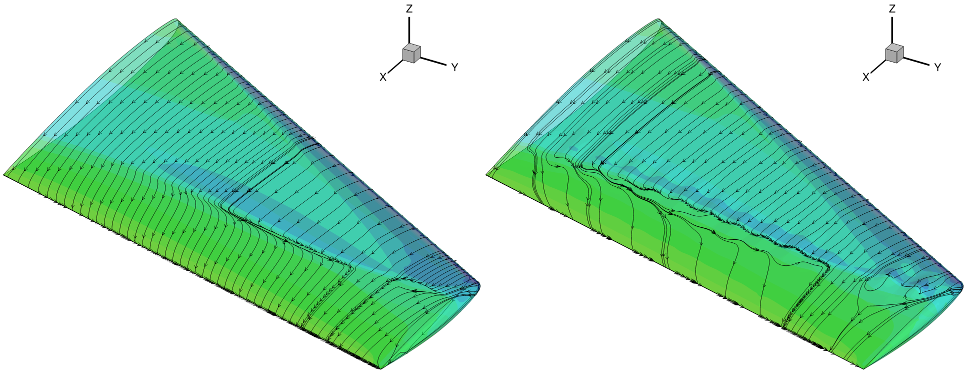
For $M = 0.84$ (the design Mach number) the baseline case shows a small separation bubble towards the tip, this is not present after the optimised bumps are added. As the Mach number is increased to $M = 0.86$ a small separation bubble begins to form on the baseline wing. For the wing with bumps the separation bubble is much larger. By $M = 0.883$ the flow over the wing with optimised bump has a large re-circulation region towards the tip which is not present on the baseline wing. The flow for the wing with bumps is mostly separated, the exception being close to the wing tip, where as the wing with no bumps only shows a separation bubble. At $M = 0.89$ both flows show significant flow separation however the baseline wing the flow re-attaches close to the wing tip and the wing with optimised bumps has a large re-circulation region near the tip.

To quantify the amount of drag which is attributed to the separation a far field analysis of the drag has been performed. The viscous pressure drag and wave drag are plotted in figure 8.40. The viscous pressure drag is the subtraction of the skin friction drag from the viscous drag. The viscous pressure drag shows a large increase in the drag just after ($M > 0.85$) the design Mach number and also causes a drag penalty before the design Mach number ($M < 0.82$) compared to the baseline wing. The wave drag shows a consistent drag reduction for $M > 0.82$.

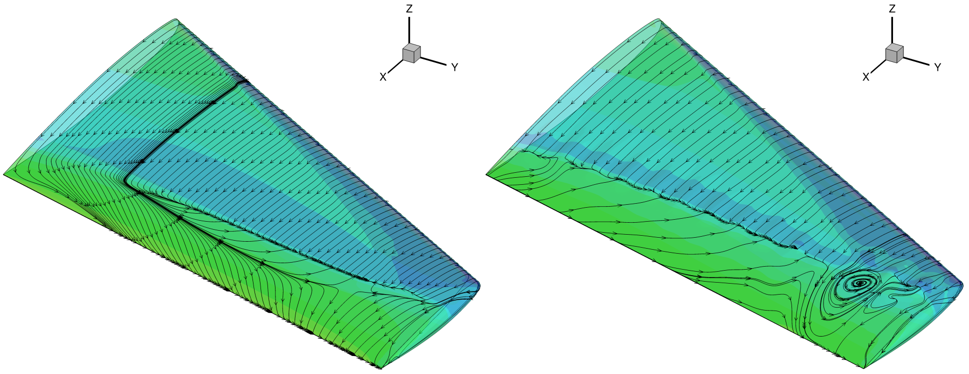
Fig. 8.41 shows the robustness for varying C_l at design Mach number. Both the 'no bumps' and 'optimised bumps' cases were run with increasing C_l until steady flow convergence could not be achieved after the drag rise point. For lower C_l values (< 0.22), the optimised bump array starts to show a penalty on the drag coefficient. From $C_l = 0.23$ the optimised bumps show a reduction in drag. Since adding shock bumps to the wing surface is similar in effect to adding camber, the wing with opti-



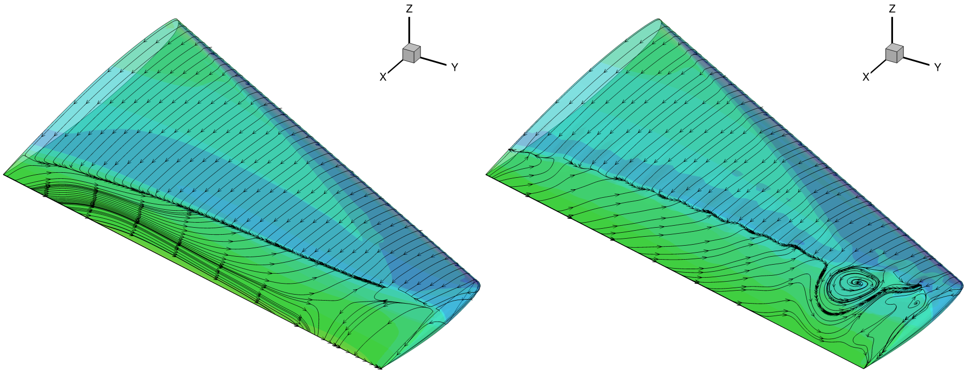
(a) M=0.84



(b) M=0.86

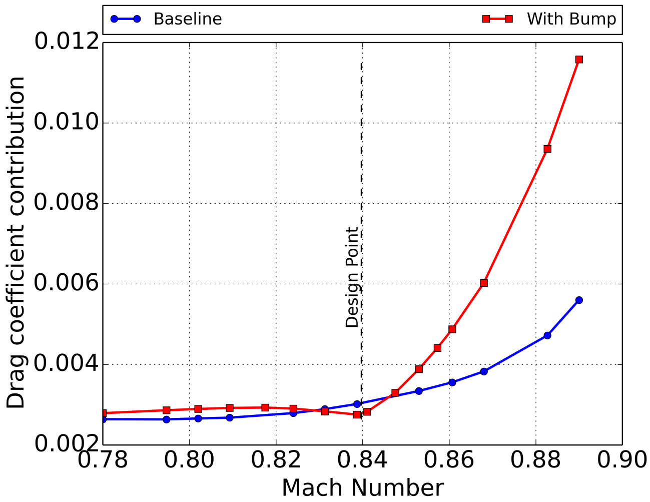


(c) M=0.883

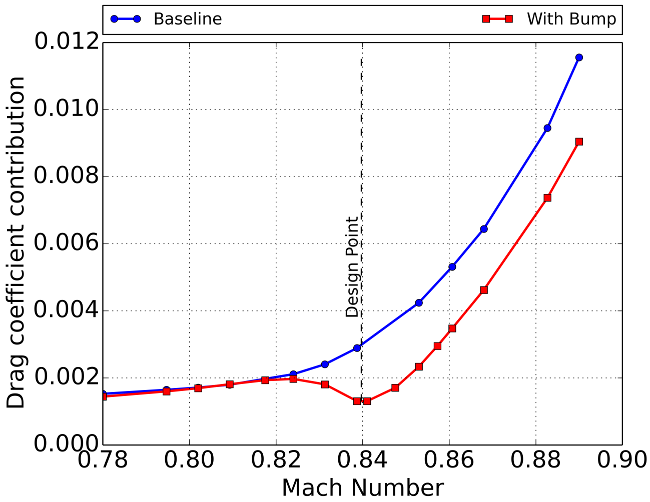


(d) M=0.89

Figure 8.39: Surface streamline comparison between the baseline wing (LEFT) and the wing with optimised bumps (RIGHT) at varying Mach numbers



(a) Viscous pressure drag



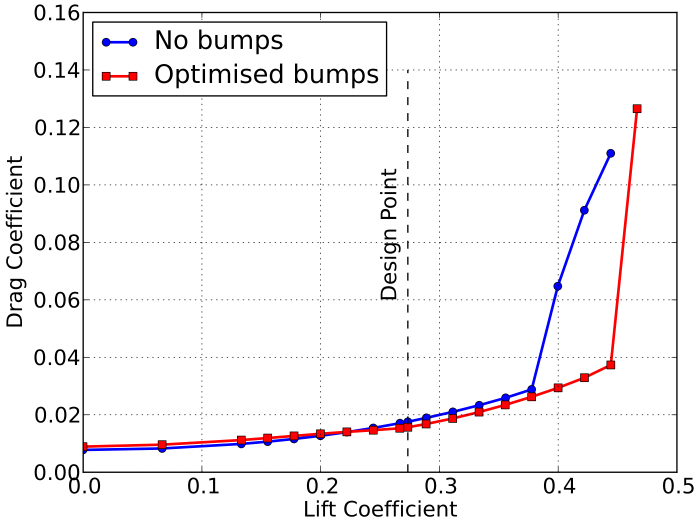
(b) Wave drag

Figure 8.40: Comparison of viscous pressure drag and wave drag between the baseline M6 wing and wing with optimised bumps.

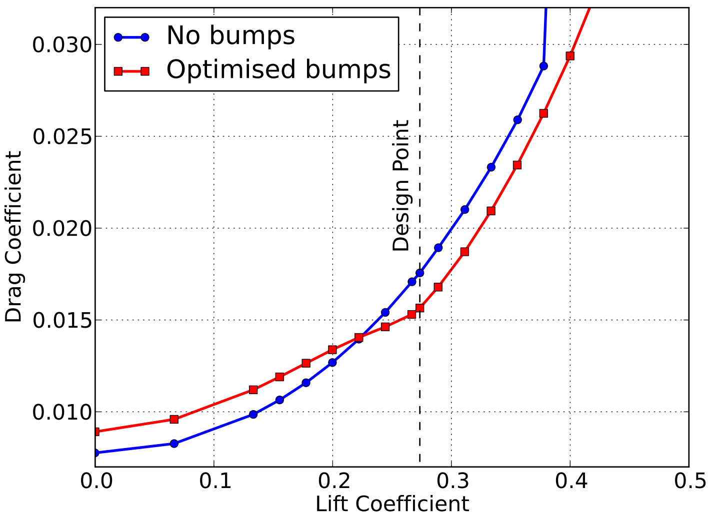
mised bumps will have a smaller angle of attack than the wing with no bumps for the same C_l.

This suggests the C_l number at which the wing stalls is likely to be higher for the wing with optimised bumps. Caution needs to be exercised in interpreting these results due to the difficulty for RANS based turbulence models for flow separation. The results from the 5^{th} AIAA Drag Prediction Workshop [90] has shown that the SA turbulence model typically predicts flow separation too early and under predicts the extent. However, it is believed that the general trend regarding the shock induced separation should be reasonable regarding the relation between the shock strength and flow separation.

In summary, when Cl is fixed at design, increasing or decreasing the Mach number from design will maintain a local benefit within a useful range (approx $0.82 < M < 0.865$). When Mach number is fixed at design, increasing or decreasing Cl from design maintains a drag reduction in a range until stall.



(a)



(b)

Figure 8.41: Drag coefficients for changes in Lift coefficient around the design point.

8.3 SUMMARY

This chapter has shown the use of the drag surface sensitivity, which was found using the mesh adjoint, to guide the placement of control bumps. The drag sensitivity clearly highlighted the regions where changing the surface would have the most effect on drag. These regions were then chosen for bump placement. In addition to this the drag surface sensitivity was also used to identify whether the optimisation had been successful. The drag surface sensitivity for the final optimisation iteration was compared to that of the baseline wing to compare the sensitivities in the chosen optimisation regions.

Optimising the bumps gave drag reductions in all the regions which were chosen from the surface sensitivity map but some regions were more successful than others.

The F6 wing provided a simple case to test the methodology. The surface pressure plot shows a lambda structure and it is not immediately obvious where shock control should be applied. The surface sensitivity map for drag highlighted the rear shock line clearly and therefore this region was chosen for optimisation. The optimisation was successful and a drag reduction of 12.46 drag counts was achieved.

The M6 wing proved a more complicated case. The lambda shock structure was very clear on the surface pressure plot and both these areas were highlighted by the surface sensitivity map. Another area was also found near the wing root which does not feature in the pressure or skin friction plots.

An interesting non-shock region of sensitivity was also discovered on the upper surface of the M6 wing. This region was shock free but the sensitivity map indicated that a surface change in the positive z-direction would give a change in the drag. This information is not available from either the surface pressure or skin friction distributions. Using the existing framework, bumps were added into the non-shock region. It was found that the optimised non-shock control bumps can also effectively reduce drag. The spanwise wave drag distribution showed that the bumps in the non-shock region reduced drag across 75%span. However due to the modification of the shock, the wave drag in the tip region increased, compared to the baseline case, and offset some of the drag reduction. Despite this there is still a drag reduction benefit.

Optimisations were performed for each region individually and finally with all regions combined. A drag reduction was found from all three regions with the best drag reduction being found in the rear shock line region. The non-shock region provided a reasonable drag reduction whereas the front shock line gave a small drag reduction. The optimised bumps placed in the entire sensitive region, when compared to the rear shock line region, showed a very slight improvement in drag reduction. For each of the optimisation regions the final optimisation iteration sensitivity showed that the sensitivities had reduced for the most part compared to the baseline wing.

Placing bumps in all the sensitivity regions on the M6 wing upper surface and optimising them for minimum drag, gives an extra reduction in drag by just one drag count in comparison to bumps in the rear shock region only. This shows that the most important region for drag reduction for the single-point optimisation is the rear shock line for this case.

A robustness study of the optimised bumps from the rear shock line case showed that the shock bumps are locally robust. It was further shown that the shock bumps consistently reduce the wave drag when a shock is present and the drag increase at off-design conditions is predominantly from the viscous pressure drag.

9

EFFICIENT METHOD TO ELIMINATE MESH SENSITIVITY USING THE ADJOINT

As shown in section 5.2 the Delaunay Graph Method (DGM) can be used to deform the mesh in the mesh adjoint method. Nielsen and Park [4] proposed a method to eliminate the necessity of calculating the mesh sensitivity ($\frac{d\mathbf{X}}{d\mathbf{S}}$) in the discrete adjoint optimisation procedure. They introduced a second adjoint equation, i.e. the mesh-adjoint equation, and, by solving it for the mesh adjoint vector, they eliminated the requirement for mesh sensitivity calculation. Their method is based on the linear elasticity method for mesh movement, which relates the volume mesh to the surface mesh by a linear system. This implicit relationship requires iterative methods to solve because the linear system is large. This will have an impact on the efficiency of the mesh adjoint.

The DGM eliminates the need to solve the large linear system in the mesh adjoint equation, therefore creating a simple and efficient way to calculate the gradient of the objective function to the design variables.

This section will focus on verifying the accuracy of the DGM method by comparing to finite differences, analytical solutions and the surface sensitivity derived using the mesh adjoint with linear elasticity.

9.1 VERIFICATION AGAINST FINITE DIFFERENCES

To verify the accuracy of the mesh adjoint using DGM, the surface sensitivities found using the DGM mesh adjoint will be compared to the finite differences for the RAE5243 aerofoil and at spanwise stations along the M6 wing. The finite differences involve the calculation of I for each surface mesh, **S**, perturbation, therefore a large number of computations is required to obtain $\frac{\partial I}{\partial \mathbf{S}}$ for each perturbed mesh. The choice of the step size is essential for the accuracy of the finite difference method. If the step size is too small the finite differences will be susceptible to computer round-off errors. On the other hand a step size which is too big will result in truncation errors.

For the DGM based mesh-adjoint approach, $\frac{\partial I}{\partial \mathbf{S}}_{DGM} = -\boldsymbol{\lambda}^{T}_{mesh}\mathbf{E}$ is used for the computation of surface mesh sensitivity, in contrast to the linear elasticity method $\frac{\partial I}{\partial \mathbf{S}}_{LE} = -\boldsymbol{\lambda}^{T}_{mesh}$. It should be noted that all the Delaunay tetrahedra vertices on the far field boundary are fixed and only the points on the aerofoil/wing surface are moving.

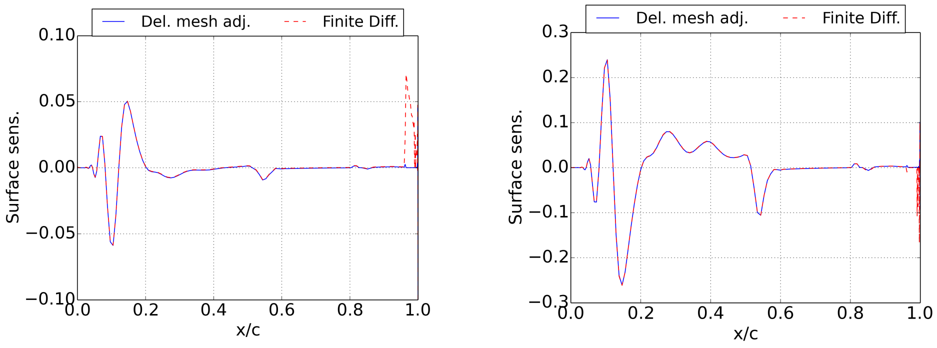
For the chosen verification case the tetrahedralisation has been constructed considering all the mesh points belonging to the solid boundaries and far field. However, different strategies can be implemented for a more robust implementation of the

DGM method, where the ultimate goal is to reduce the skewness of the Delaunay tetrahedra resulting from the Delaunay decomposition.

9.1.1 *Verification in 2D*

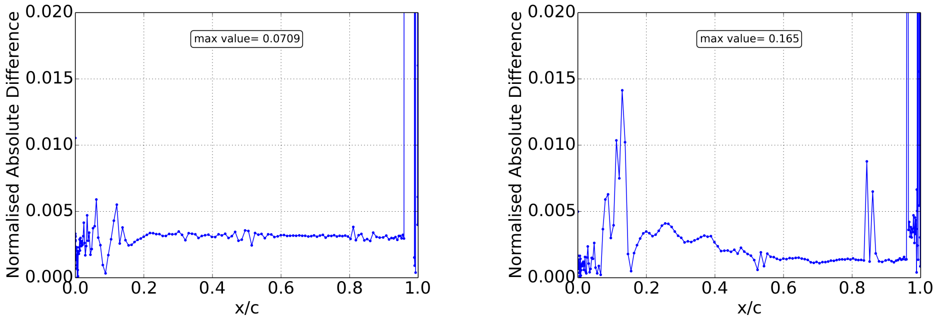
The surface sensitivity in 2D is plotted from the DGM mesh adjoint and finite differences for 3 objective functions, C_d, C_l and C_{my}. These results are shown in figures 9.1,9.2 and 9.3. In general, the sensitivities are very close between the finite differences and the Delaunay mesh adjoint. The exceptions are very close to the leading edge and trailing edge where the physics is complicated. For the drag surface sensitivity the strongest sensitivity, away from the leading and trailing edges, is found in the region 0.1-0.2%c with a smaller sensitivity closer to the shock region. The lift and moment surface sensitivity is the strongest, away from the leading and trailing edges, in the shock regions for the z component and in the region 0.05-0.2 %c for the x component.

Figures 9.1,9.2 and 9.3 also show the error of each of the sensitivity components. These figures confirm the largest differences are found in the leading and trailing edges. In addition to these areas, the lift and moment sensitivities also show differences in the shock region, however the differences are not of a significant magnitude. The objective function with least agreement between the DGM mesh adjoint and finite differences is the coefficient of lift but the differences are still relatively small.



(a) **x component**

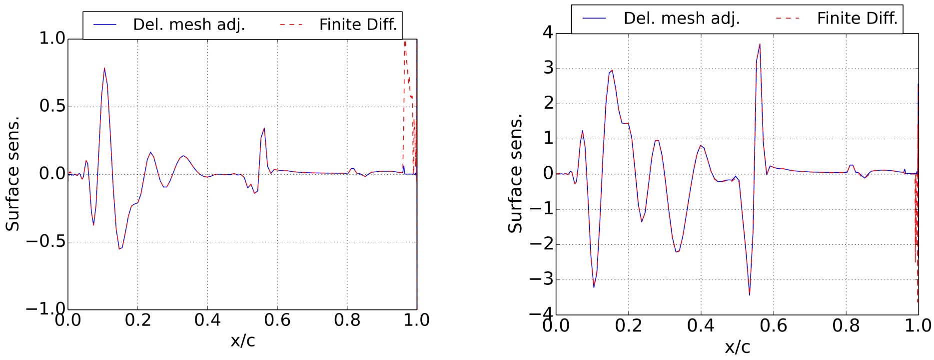
(b) **z component**



(c) **x component error**

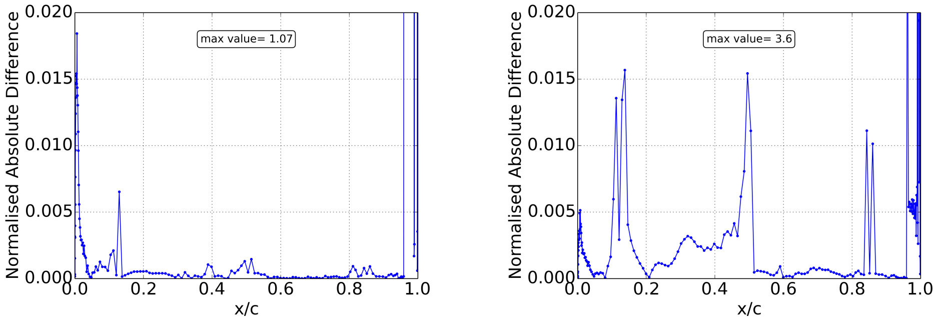
(d) **z component error**

Figure 9.1: A comparison between drag surface sensitivity and the error between the mesh adjoint with DGM and finite differences with $\epsilon = 10^{-04}$ for the upper surface of the RAE5243. (a) The x component of the drag surface sensitivity (b) The z component of the drag surface sensitivity (c) The error in the x component and (d) the error in the z component



(a) **x component**

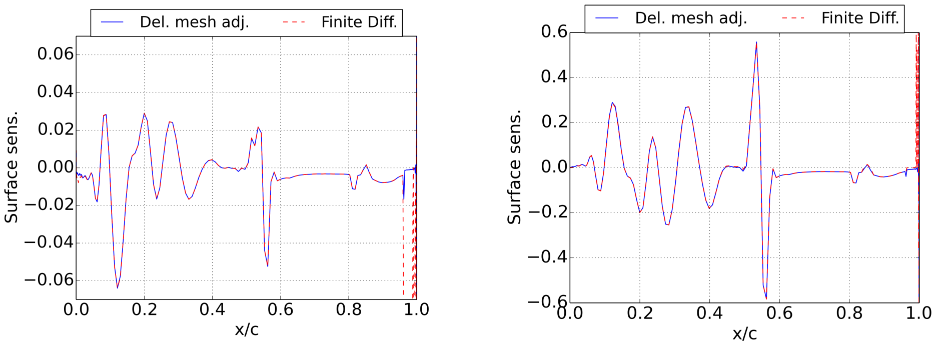
(b) **z component**



(c) **x component error**

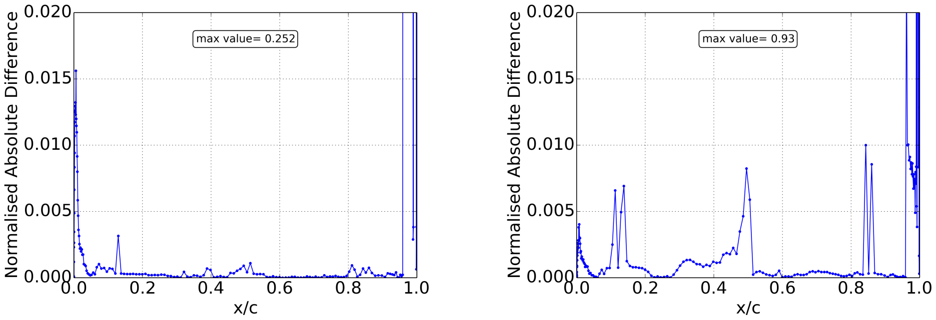
(d) **z component error**

Figure 9.2: A comparison between lift surface sensitivity and the error between the mesh adjoint with DGM and finite differences with $\epsilon = 10^{-04}$ for the upper surface of the RAE5243.(a) The x component of the lift surface sensitivity (b) The z component of the lift surface sensitivity (c) The error in the x component and (d) the error in the z component



(a) **x component**

(b) **z component**



(c) **x component error**

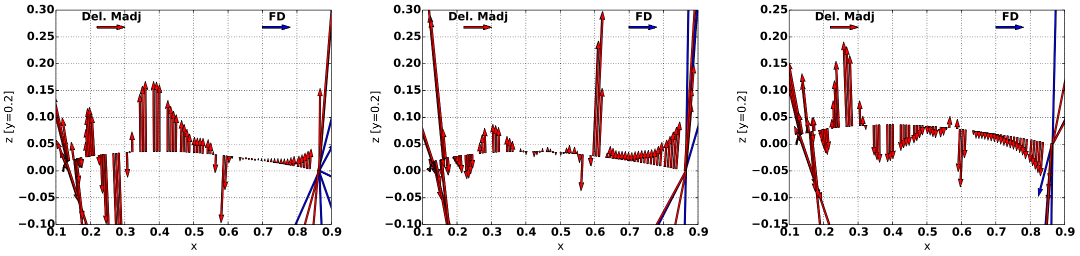
(d) **z component error**

Figure 9.3: A comparison between moment surface sensitivity and the error between the mesh adjoint with DGM and finite differences with $\epsilon = 10^{-04}$ for the upper surface of the RAE5243. (a) The x component of the moment surface sensitivity (b) The z component of the moment surface sensitivity (c) The error in the x component and (d) the error in the z component

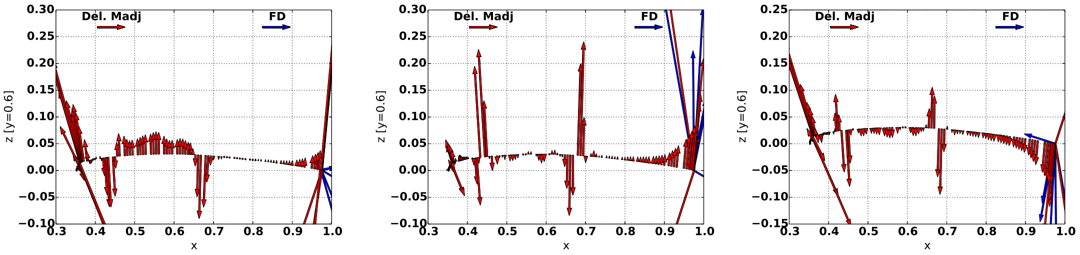
9.1.2 *Verification in 3D*

In figure 9.4 the surface sensitivity comparison between finite difference and DGM mesh adjoint are compared at 3 spanwise stations across the M6 wing. The stations considered are at 16.67%span, 50%span and 83.33%span. These stations include points within a tolerance of the slice.

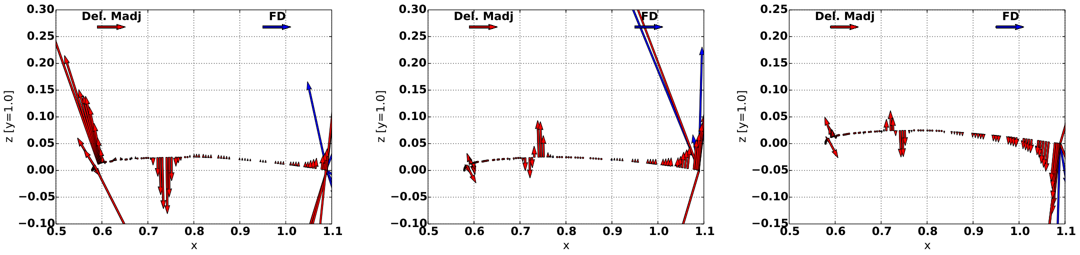
In all the figures the trailing edge shows the least agreement. Away from the trailing edge, the C_d surface sensitivity shows a strong sensitivity region near the leading edge and shock regions which show very good agreement. For C_{my} and C_l the sensitivity is strongest (disregarding sensitivities near the trailing edge) in the shock location with the next strongest region being closer to the leading edge.



(a) C_D at 16.67%span (b) C_L at 16.67%span (c) C_{my} at 16.67%span



(d) C_D at 50%span (e) C_L at 50%span (f) C_{my} at 50%span



(g) C_D at 83.33%span (h) C_L at 83.33%span (i) C_{my} at 83.33%span

Figure 9.4: Comparison of the upper surface sensitivity for points around 3 different spanwise slices of the mesh adjoint with DGM to Finite Differences with step size $\epsilon = 10^{-06}$

9.2 VERIFICATION AGAINST ANGLE OF ATTACK

To get the sensitivity to the angle of attack using the DGM mesh adjoint method the geometry of the aerofoil/wing will be rotated by an angle while the far-field is kept fixed. Doing this ensures that the adjoint method is still in terms of only geometric changes and does not require any flow variable gradient calculations. The analytical solution comes from the TAU code where the objective functions (C_D, C_L and C_{my}) are linearised with respect to angle of attack.

Table 9.1 shows the comparison between the DGM mesh adjoint with a geometry rotation of 10^{-04} ° and the analytical solution for C_D, C_L and C_{my}. The 2D RAE5243 aerofoil and the 3D M6 wing are used as test cases for this verification. The table shows that the agreement is very good with the maximum error being just over 0.5% for the rate of change of C_L w.r.t. α for both the 2D and 3D cases.

Table 9.1: Results for verification against the analytical solution of angle of attack

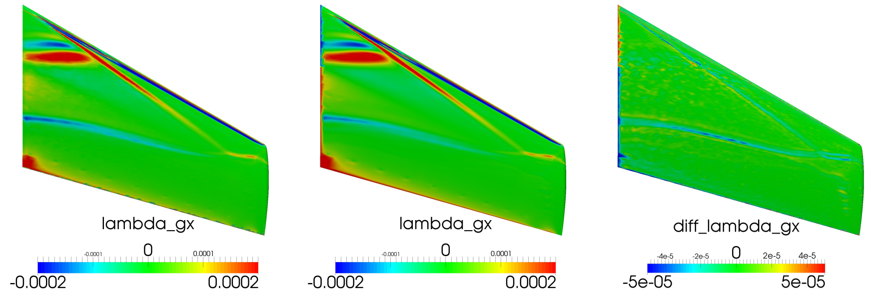
	2D	**3D**
$\frac{dC_D}{d\alpha}$		
Analytical	0.005563	0.0073782
Del. Madj.	0.005567	0.0073784
Difference	-0.0719 %	-0.00271 %
$\frac{dC_{my}}{d\alpha}$		
Analytical	0.133995	0.0914981
Del. Madj.	0.133864	0.0913225
Difference	0.00979 %	0.192 %
$\frac{dC_L}{d\alpha}$		
Analytical	0.001247	0.001416
Del. Madj.	0.001254	0.001424
Difference	-0.558 %	-0.562 %

9.3 VERIFICATION AGAINST THE MESH ADJOINT WITH LINEAR ELASTICITY

The mesh adjoint with linear elasticity has been used extensively in this work, therefore a comparison of the surface sensitivity to the DGM mesh adjoint will give a good benchmark for accuracy of the method. Two cases are considered, the M6 wing and the F6 wing.

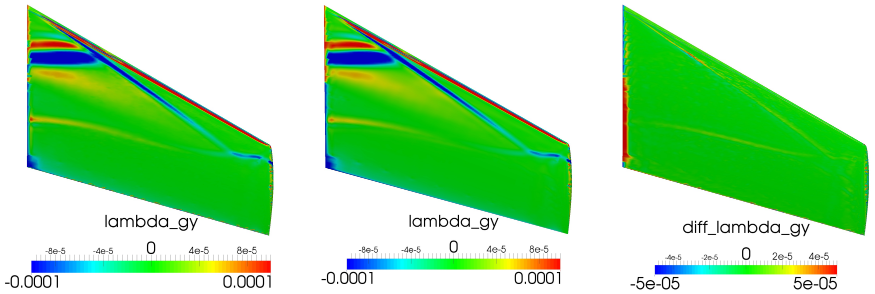
9.3.1 *M6 wing*

Figure 9.5 shows the comparison of the mesh adjoint with LE and DGM and the difference between the 2 drag surface sensitivities in the x, y and z directions. In all directions the sensitivity near the symmetry plane is different, this is due to post-processing techniques. The largest differences, in all directions, are found along the shock lines. The differences are of a very small relative scale. There is also some small scale differences that can also be seen on the tip. No significant difference is apparent between the two methods, in particular, the non-shock region sensitivity in both methods appears to match incredibly well.



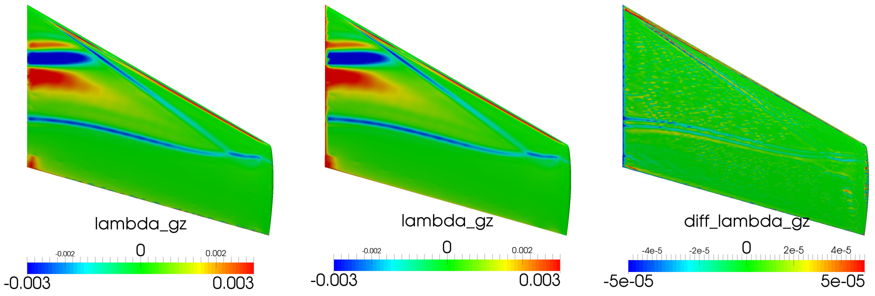
(a) DGM (b) LE (c) Difference

x-component



(d) DGM (e) LE (f) Difference

y-component



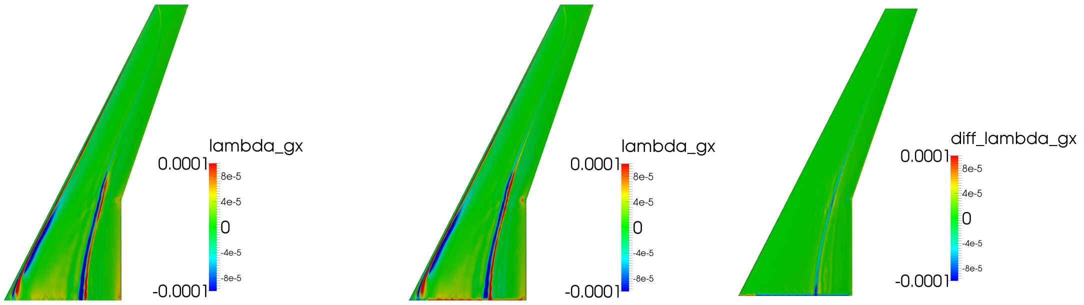
(g) DGM (h) LE (i) Difference

z-component

Figure 9.5: Comparison of the upper drag surface sensitivity on the M6 wing for the mesh adjoint using DGM and mesh adjoint using LE

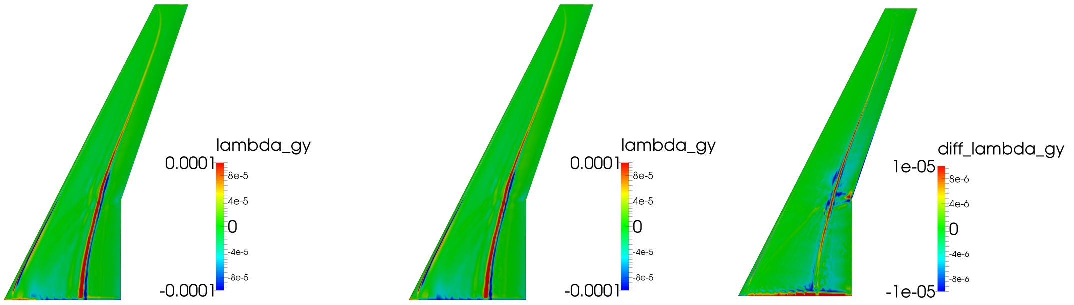
9.3.2 *F6 wing*

Figure 9.6 shows the surface sensitivity for drag on the F6 wing. The difference in sensitivity is most apparent along the shock line close to the wing root and near the crank. The differences shown in the y and z directions show very small differences and therefore the agreement between the two methods is good. In the x direction the differences are slightly larger, even reaching the same order as the original sensitivity. Despite this discrepancy in the shock line the rest of the sensitivity agrees very well and the trend between the two sensitivity methods matches. In the y direction some small differences can also be seen close to the leading edge.



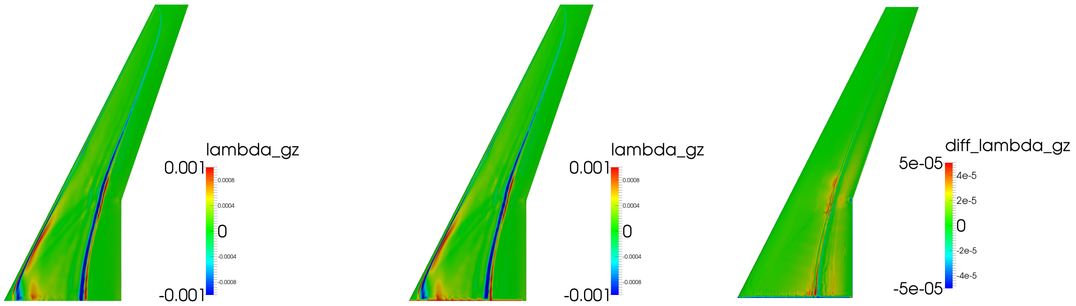
(a) DGM (b) LE (c) Difference

x-component



(d) DGM (e) LE (f) Difference

y-component



(g) DGM (h) LE (i) Difference

z-component

Figure 9.6: Comparison of the upper drag surface sensitivity on the F6 wing for the mesh adjoint using DGM and mesh adjoint using LE

9.4 3D SHOCK BUMP OPTIMISATION USING MESH ADJOINT WITH DGM

To show that the DGM with mesh adjoint is robust and can give consistently usable gradients an optimisation will be performed. The rear shock line case from section subsection 8.2.1 is considered. Figure 9.7 shows the surface sensitivity, showing only regions where adding material in the z-direction is going to reduce drag. The optimisation bounds are also highlighted, 14 bumps with equal width will be optimised in this region. Table 9.2 shows the drag values form the final iteration of the optimisation. The drag values differ by less than a drag count hinting that the same minima has been found from the differing optimisation processes. Interestingly, the difference in skin friction drag is negligible.

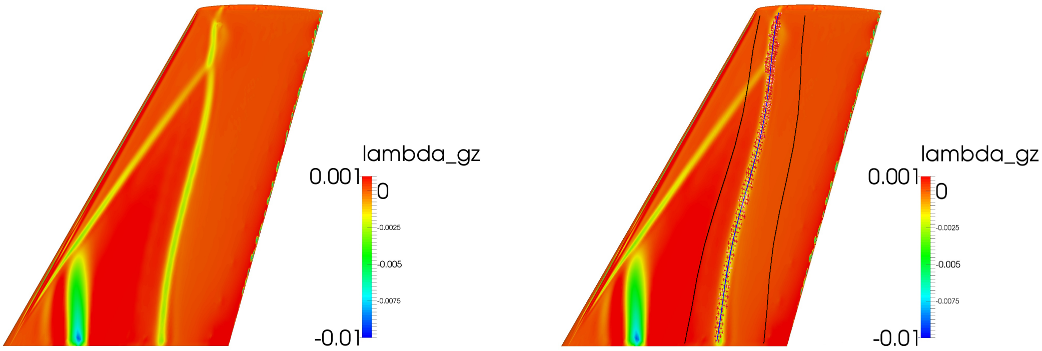


Figure 9.7: Sensitivity map of the baseline M6 wing using mesh adjoint with DGM

Figure 9.8 shows a comparison of the surface sensitivities from the final iterations of the optimisations using DGM (left) and LE (right). The sensitivity in the rear shock line has reduced significantly due to the presence of the optimised bumps and has been reduced to a similar degree in both optimisations. Figure 9.4 is a bar chart of the bump heights from the optimised DGM and LE cases. The bump heights are approximately the same for both cases, the largest, but still relatively small, difference is in the tip region where the wave drag is strongest. The optimisation history of both optimisations is shown in figure 9.4. It is clear that the two methods have taken different paths to find the drag minimum. Both cases experience a drag spike early in the optimisation but after this the drag reduction is fairly monotonic.

Table 9.2: Near-field drag decomposition for M6 wing with optimised results using mesh adjoint with DGM and LE

	C_D	$C_{D,pres}$	$C_{D,fric}$
DGM	0.01565	0.01029	0.00536
LE	0.01569	0.01033	0.00536
Difference	-0.255 %	-0.387 %	0 %

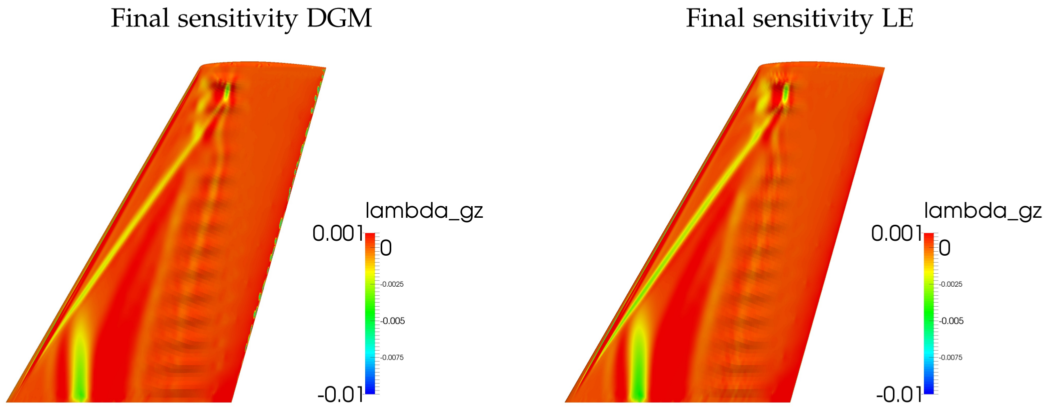
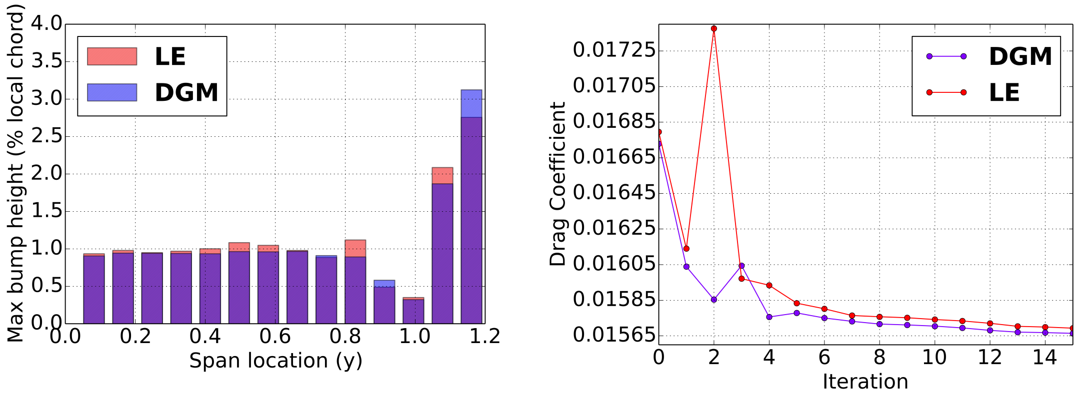


Figure 9.8: Comparison of the upper surface sensitivity on the M6 wing for the mesh adjoint using DGM and mesh adjoint using LE



(a) Comparison of the optimised shock bump height

(b) Comparison of the optimisation convergence

Figure 9.9: Comparison of final height of optimised bumps and optimisation convergence for DGM and LE mesh adjoint

9.5 BENCHMARKING (DGM VS LE)

During this work two different methods have been used for solving the mesh adjoint equation. The DGM method has an explicit formulation involving the product between a matrix and vector. In contrast the LE method involves an implicit formulation involving the need to solve a large linear system.

This section will show benchmarking data for the two methods using the M6 case shown in the previous chapter. The computing resources available are The University of Sheffield cluster, Greengrid, and the CFMS cluster in Bristol. Due to memory constraints the LE mesh adjoint method could not be run on Greengrid. This method was run on high memory nodes on CFMS instead. The DGM mesh adjoint was run on Greengrid. Since Greengrid comprises of two different kinds of cores the DGM mesh adjoint was run on both.

The DGM and LE mesh adjoint methods are both run in serial to give an accurate comparison of the time requirements. The DGM method uses a hybrid of C and python, whereas the LE method is coded in only C and uses the PETSc libraries to solve the linear system. Initially the DGM and LE methods require the calculation of the mesh deformation matrices (**K** for LE (equation 6.69) or **E** for DGM (equation 6.76)). This is done only once for each new mesh. In an optimisation, the surface sensitivity is calculated at each iteration, therefore the calculation of the surface sensitivity is benchmarked.

Table 9.3: Benchmarking for DGM and LE

	Greengrid (DGM)		**CFMS (LE)**
Core (Intel(R) Xeon(R))	E5410 @ 2.33GHz	E5620 @ 2.40GHz	X5550 @ 2.67GHz
Available RAM	7.8Gb	23.5Gb	70.8Gb
Mesh deformation matrix	929.59s	879.63s	170.62s
Surface sensitivity	323.14s	248.87s	1532.09s
Optimisation estimate (16 iterations)	6099.83 s 1.69 hrs	4861.55s 1.35 hrs	24684.06 s 6.85 hrs

Table 9.3 shows the amount of time taken to compute the mesh deformation matrices and surface senstivities in seconds. Clearly the time take to calculate the mesh deformation matrix is cheaper for the LE method but this is only a one off cost per new geometry. The surface sensitivity needs to be calculated for each iteration step for an optimisation.

In the previous section the number of iterations required to find a minimum when placing shock bumps on the M6 wing was 16. The row 'Optimisation estimate' in Table 9.3 shows how long each method would take for calculating only the mesh adjoint parts. This comprises of 1 calculation of the mesh deformation matrix and 16 calculations of the surface sensitivity. Despite the differences in computing architecture favouring the LE method, it is clear that the DGM mesh adjoint method will provide a significant time saving in comparison to the LE mesh adjoint method.

9.6 SUMMARY

This chapter has presented the verification of the mesh adjoint method using DGM mesh movement. The DGM mesh movement strategy is fast and efficient and when it is applied to the mesh adjoint gives an explicit one-to-one mapping between the rate of change of the mesh to the rate of change to the surface w.r.t. the design variables.

The DGM mesh adjoint method was verified using finite differences, analytical solutions and the LE mesh adjoint for several objective functions. In all the cases the agreement between each of the methods was good with the exceptions of the trailing edge and the leading where there is complicated physics.

The DGM mesh adjoint was used in an optimisation of shock bumps to show that it is a viable method that could provide robust movement and accurate gradients consistently. The optimisation results were compared against the LE mesh adjoint results from the previous chapter and showed that they approached the same amount of drag reduction. Despite the two methods finding the same drag reduction the optimisations took two different paths to get there.

Finally, the time taken to compute the DGM and LE mesh adjoint methods was compared. The data showed that the DGM method offers a large reduction in time for surface sensitivity calculation compared to the LE method despite the DGM method being run with less available memory and on less powerful cores.

10

CONCLUSION

10.1 CONCLUDING REMARKS

Optimisation is a major factor in the generation of new designs and devices for the reduction of drag on an aerodynamic body. This work has shown that using the surface sensitivity from the mesh adjoint solution to guide the placement of flow devices can be a useful tool in increasing aerodynamic efficiency.

The mesh adjoint method using linear elasticity was used to produce the drag surface sensitivity. This identified the regions on the wing where changes in the surface will have the most effect on drag. Using previous studies from the literature an optimisation region was defined. It was shown that the bump width is an insensitive parameter and therefore can be chosen to suit the problem and optimisation region at hand.

The surface sensitivity was also used to assess the success of the optimisation procedure. If the strength of the sensitivity after surface modification in the optimisation region is reduced then the optimisation was deemed to be successful. Interestingly, when certain areas were optimised there is an obvious change in sensitivity strength on other regions away from the optimisation region. This hints towards a potential link between sensitive regions.

The F6 wing surface shows a lambda structure for the pressure on the surface and shows a large separation outboard of the wing crank. It is not directly obvious from these flow features where to place a device for the most effect on drag. However the surface sensitivity presented a simpler case for bump deployment where there was only one shock line which would have any significant effect on drag. When the optimised bumps were deployed the wave drag was decreased consistently across the wing span. In addition to this, the separation of the flow around the crank was also less severe.

Several sensitivity regions were identified for the M6 wing. These were the rear-shock line, front-shock line and the non-shock region. The front and the rear shock line regions followed the lambda shock structure which can clearly be seen in the surface pressure, however the non-shock region does not feature in either the surface pressure or skin friction. This non-shock region shows a significant sensitivity strength but could have been disregarded as a region of optimisation if the surface sensitivity was not consulted.

In the three M6 wing optimisation regions the optimised bumps all resulted in a drag reduction. The greatest drag reduction was found after optimisation in the rear-shock line and the least drag reduction was found after optimisation in the front shock line. When all the sensitivity regions were considered for optimisation, there was no significant improvement of drag reduction over the rear-shock line case. This was deemed to be due to there being a link between sensitivity regions. The surface sensitivity from the final optimisation iteration showed a clear link between the non-shock sensitivity region and the rear-shock line sensitivity. Optimisation in the front shock line region also increased the strength of the sensitivity in the other 2 regions.

An efficient method for calculation of the mesh adjoint solution was developed and demonstrated to be robust in a 3D optimisation. The DGM mesh deformation technique gives a one-to-one mapping between the surface mesh and the computational mesh which was explicit. This can be exploited when calculating the mesh adjoint to give an accurate evaluation of the surface gradients.

The mesh adjoint with DGM was tested against a variety of methods to prove that it was capable of producing accurate gradients for objective functions of C_d, C_l and C_{my}. For the cases where the sensitivity was compared directly the majority of the sensitivities matched closely, however close to the leading edge and trailing edges of the aerofoil/wing the sensitivities showed some disagreement. Since the physics in these regions is very complicated, involving large flow gradients, it is not unreasonable to see some differences in the sensitivities in these regions. In particular the trailing edge has a geometric singularity which creates issues when trying to calculate certain values (e.g. normals to the surface). These inaccuracies do highlight the importance of having well defined meshes in the leading and trailing edges to minimise any potential discretisation errors which may influence the sensitivity.

Finally, in the 3D optimisation the DGM mesh adjoint was compared against the LE mesh adjoint. Although the two methods took different routes to the optimal solution the final results from both optimisations was very similar. Therefore, the DGM mesh adjoint could be successfully implemented to replace current methods to give an efficient and robust calculation of the sensitivities for adjoint optimisation.

10.2 SUGGESTIONS FOR FUTURE WORK

The shock control bumps here were optimised based on a single design condition for 3D geometries and a single and multi-point optimisation was undertaken in 2D. chapter 7 shows a 2D multi-point optimisation method which increases the drag divergence Mach number and delays stall. The 3D bump on the M6 wing has been shown to be reasonably robust but further optimisation may provide further drag savings at off-design conditions, the analysis shows that the drag saving would be best found from reducing the separation after the bump, one method to achieve this would be to introduce vortex generators to alleviate the adverse pressure gradient. A multi-point optimisation of the shock bumps on a 3D wing would be an important step into analysing the effect of bump shape on shock bump robustness.

The literature review at the beginning of this work shows there are a variety of geometries under the umbrella term of shock control bumps but there is no direct comparison of the effectiveness of each topology. A study of a variety of different bump shapes added to a consistent geometry (flat plate, 3D wing, etc.) would identify the strengths and weaknesses of each geometry variation and give a direct comparison of bump performance at consistent flow conditions and surface geometries.

The surface sensitivity has been used to identify areas to optimise for drag reduction. The objective function used for the optimisations in this thesis have included a lift correction (equation 8.1). This lift correction will also feature in the gradient and therefore could be plotted as part of the surface sensitivity to further guide the optimisation.

Any manipulation of the surface due to the optimisation will also have an effect on lift and moment. The surface sensitivities for these two quantities can also be plotted, as shown in figure A.7. A combined study of the drag, lift and moment surface sensitivities would be useful to identify regions in which one quantity can be changed with little effect on the other two. This could also be useful to identify regions which are the most or least sensitive to surface changes, for drag, lift and moment, and could give further information to a wing designer to assess the trade off when placing a device or moving the wing surface.

The DGM mesh adjoint method has been developed in this thesis and shows an efficient, explicit method for calculating the surface sensitivities. This is particularly useful when the mesh size increases and therefore makes it suitable for use with complex geometries. Issues have arisen in the past using implicit methods for the mesh adjoint due to the computational expense of solving a large linear system. For example, solving the adjoint for a whole aircraft is cumbersome due to the size of the mesh necessary to resolve the flow accurately. With the DGM, the speed and reduction in memory requirements make this a much more feasible analysis. A whole aircraft geometry could be used to identify sensitive areas where surface deformations would affect a change in the objective function.In particular this could be useful in complex regions such as around the nacelle and pylon where flow features, such as shocks, may occur and could be treated with a flow control device (e.g. shock bumps).

Part IV

APPENDIX

A

APPENDIX A

A.1 OPTIMISATION HISTORIES FOR CFD SIMULATIONS

A.1.1 *RAE5243*

This section contains the optimisation history from the simulations performed in chapter 7 on the RAE5243 aerofoil.

A.1.1.1 *Single-point*

The single-point optimisation for the RAE5243 aerofoil took 17 iterations to find a minimum. Figure A.1 shows the optimisation history for the optimisation. The starting point for the optimisation already produced a drag reduction compared to the baseline ($c_d = 0.01622$. The first few iterations give a dynamic change in drag values, this settles after iteration 8 giving a monotonic decrease to the optimal design.

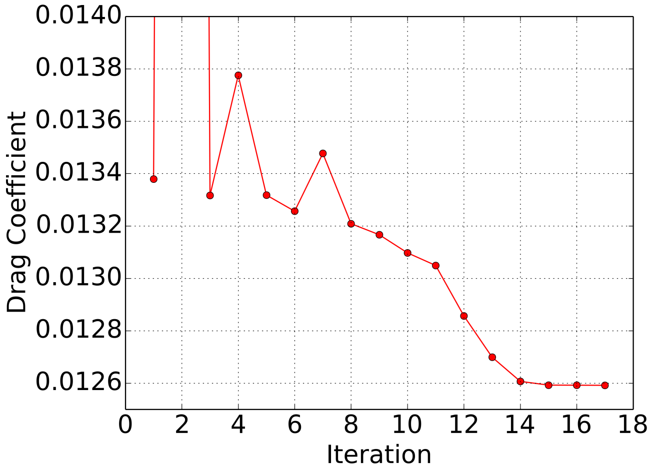
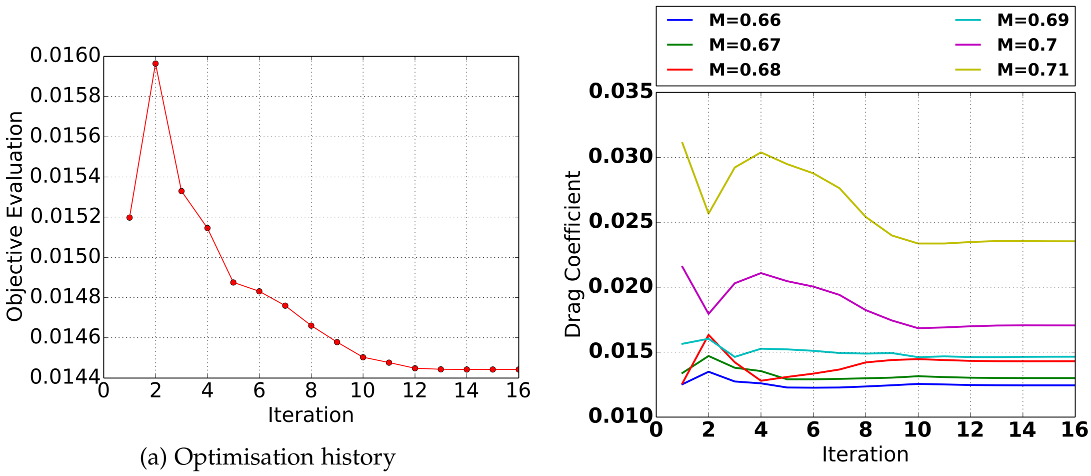


Figure A.1: Optimisation history for a single-point optimised shock bump on the RAE5243 aerofoil

A.1.1.2 *Multi-point*

Figure A.2 shows the objective function and the drag evaluations for each flow condition considered in the multi-point simulation of the RAE5243. For the drag evaluation, the starting point for the design variables are those found from the single-point optimisation. The initial spike in the objective function is caused by the rapid increase in the bump height shown in figure 7.11.



(a) Optimisation history

(b) Drag evaluation for each Mach point

Figure A.2: Progression of drag and objective function for RAE5243 multi-point optimisation

A.1.2 *M6 wing optimisation*

This section contains the optimisation histories for each of the optimisations performed in chapter 8 on the F6 and M6 wing.

A.1.2.1 *Rear-Shock leg*

Fig. A.3 shows the optimisation history of the drag coefficient for shock bumps placed along the rear shock leg. After a large spike in drag at iteration 2 the drag quickly reduces monotonically to the local minimum.

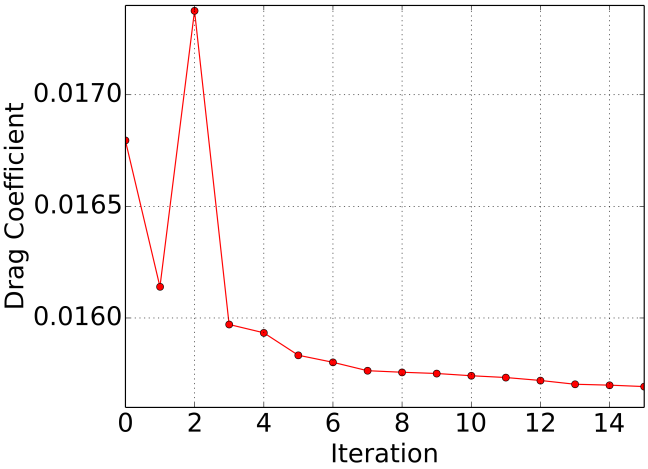


Figure A.3: Optimisation history

A.1.2.2 *Front Shock leg*

Fig. A.4 shows the optimisation history of the drag coefficient for shock bumps when placed along the front shock leg. The drag monotonically decreases throughout the whole optimisation after starting at a drag value above the baseline ($C_D = 0.0176$).

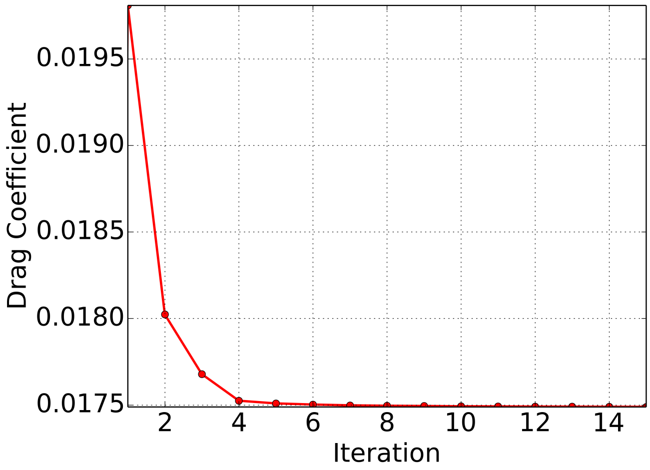


Figure A.4: Optimisation history

A.1.2.3 *Non-shock region*

Fig. A.5 shows the optimisation history of the drag coefficient for bumps placed in the sensitive non-shock region at the root of the M6 wing.

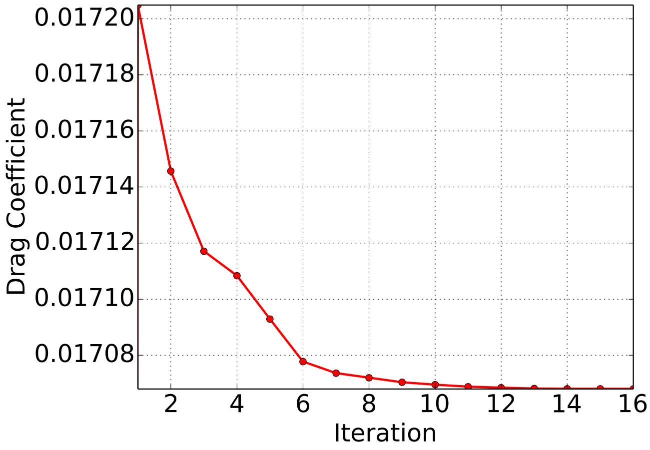


Figure A.5: Optimisation history for non-shock region optimisation

A.1.2.4 *Entire sensitivity region*

Fig. A.6 shows the optimisation history of the drag coefficient when bumps are placed in all the sensitive regions on the M6 wing surface. The initial iteration has a very large drag value due to the fact the M6 wing surface has been significantly changed. This rapidly reduces as the bump arrays are optimised.

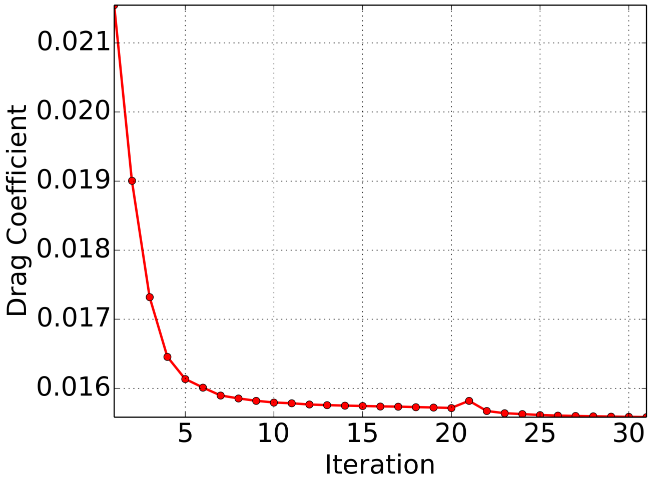


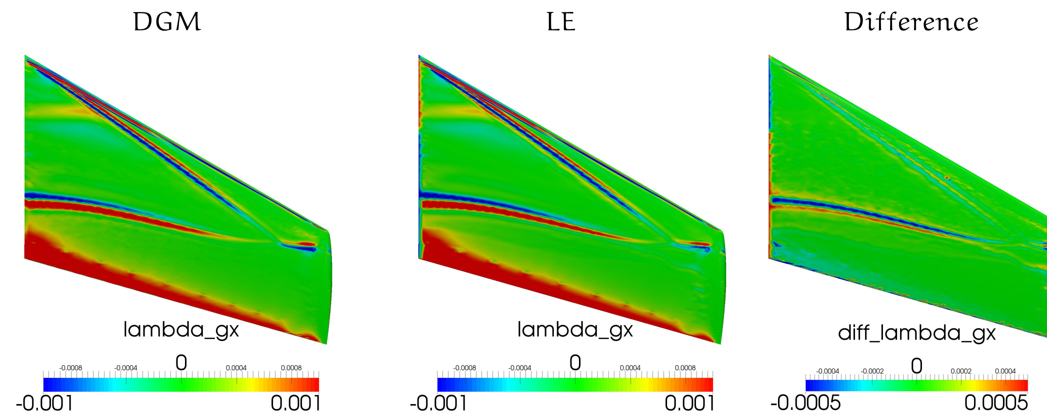
Figure A.6: Optimisation history

A.2 COMPARISON OF DGM AND LE MESH ADJOINT FOR COEFFICIENTS OF LIFT AND MOMENT

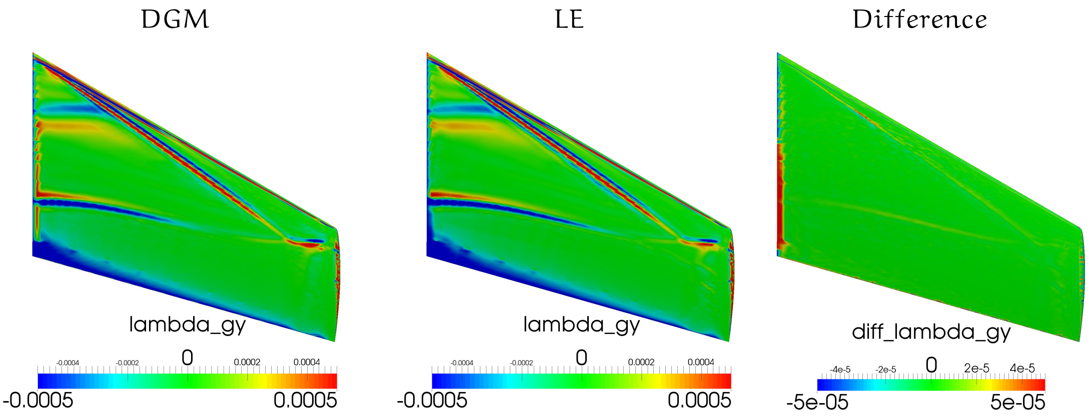
A.2.1 *M6 wing*

In section 9.3 several carpet plots were shown to compare the drag surface sensitivity from the DGM mesh adjoint method and the LE mesh adjoint method. Figures A.7 and A.8 shows the surface sensitivity for lift and moment on the upper surface of the M6 wing to further compare the results.

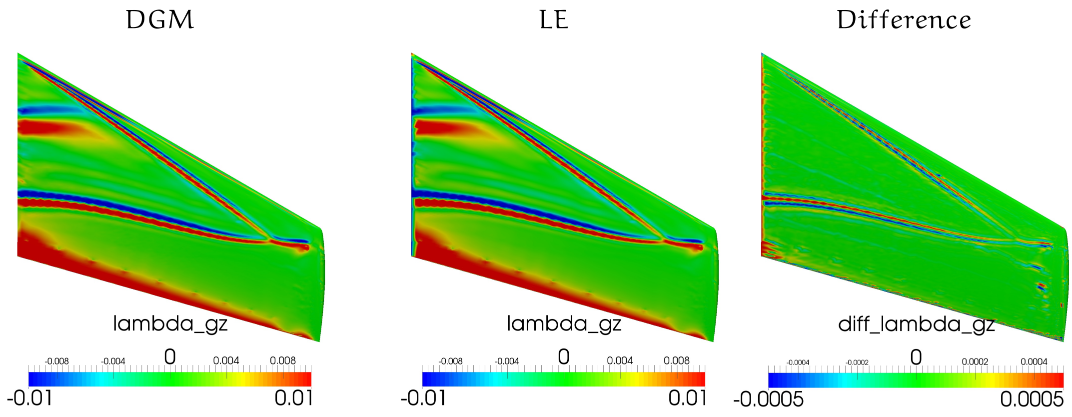
Btoh the lift and moment surface sensitivity comparison shows an excellent agreement between the two methods with the difference between the two methods in all directions being at a much lower order than the surface sensitivity.



x-component

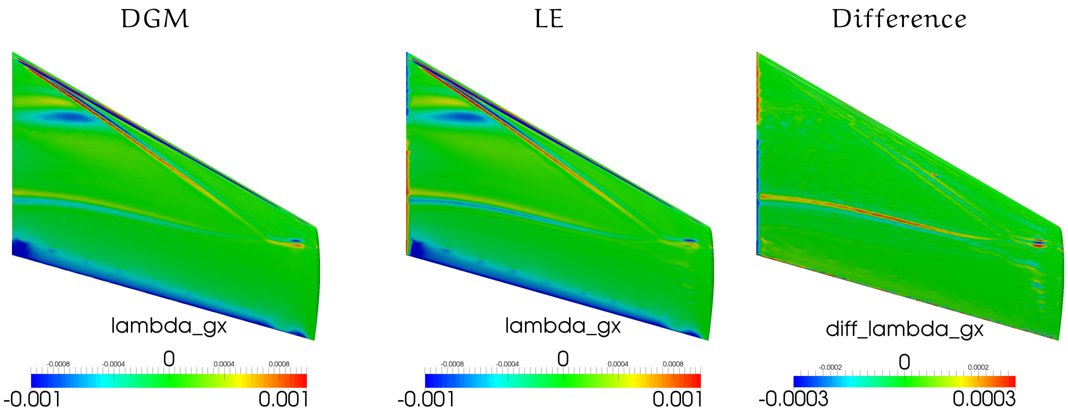


y-component

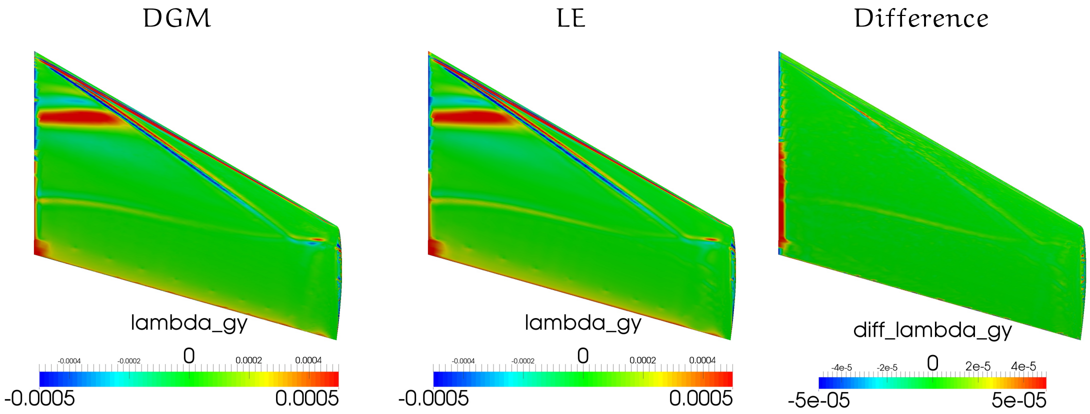


z-component

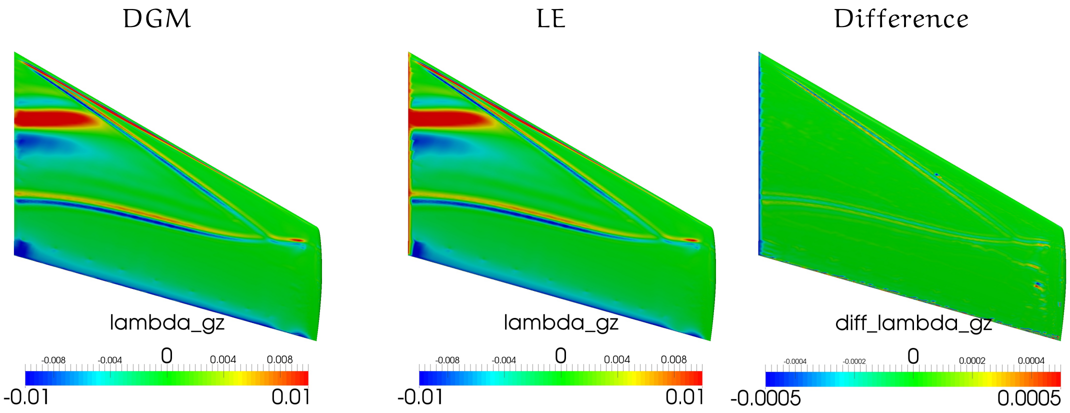
Figure A.7: Comparison of the upper lift surface sensitivity on the M6 wing for the mesh adjoint using DGM and mesh adjoint using LE



x-component



y-component

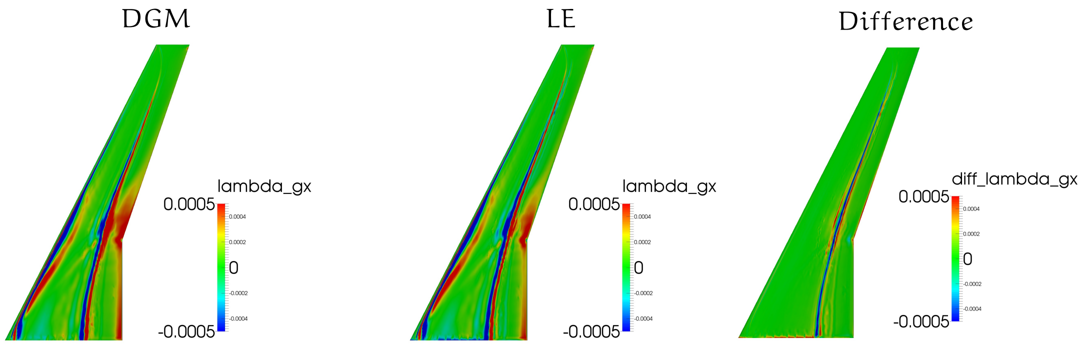


z-component

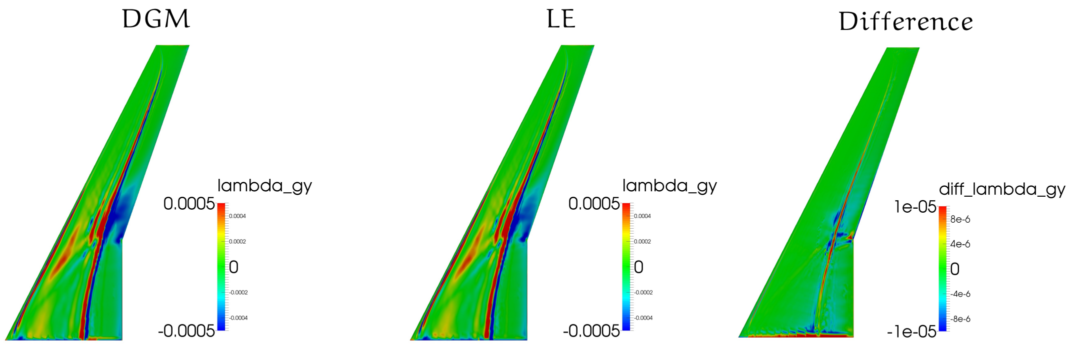
Figure A.8: Comparison of the upper moment surface sensitivity on the M6 wing for the mesh adjoint using DGM and mesh adjoint using LE

A.2.2 *F6 wing*

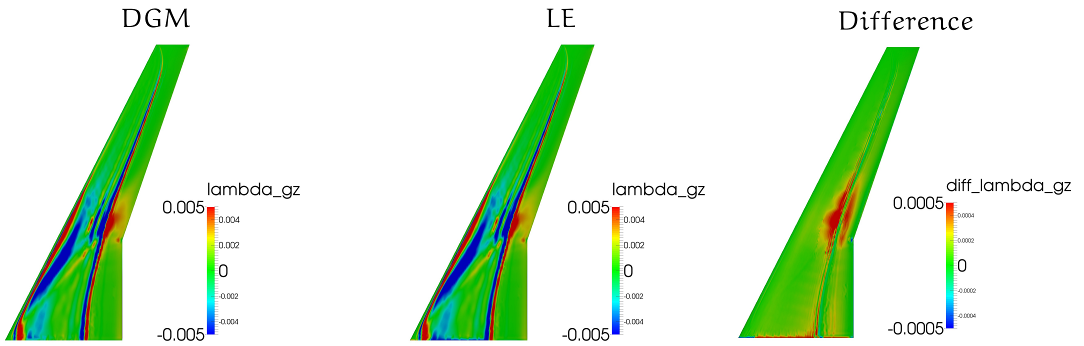
Figures A.9 and A.10 show the surface sensitivity for lift and moment on the upper surface of the F6 wing. The lift and moment surface sensitivity in the y and z directions show good agreement where the difference is much less than the sensitivity from both methods. However in the x-direction the difference between the two methods in the shock regions is a large fraction of the original sensitivities this is the most noticeable for the lift case.



x-component

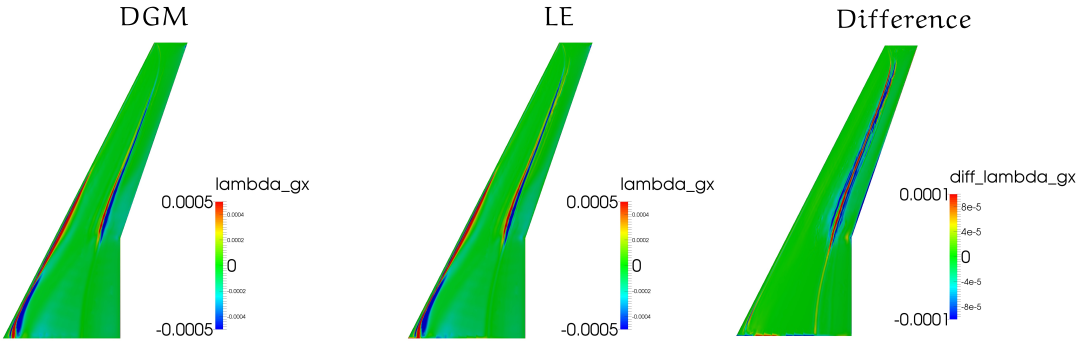


y-component

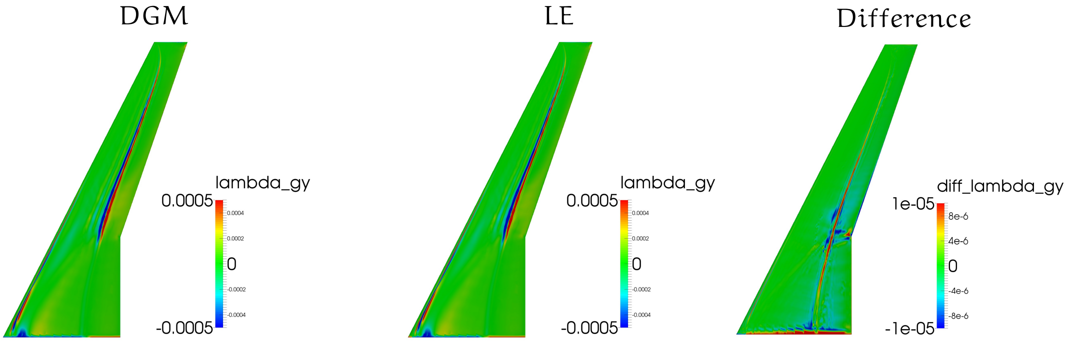


z-component

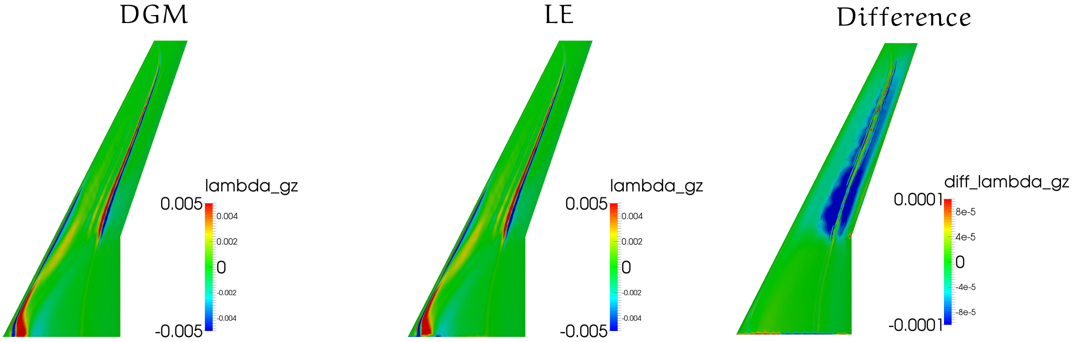
Figure A.9: Comparison of the upper lift surface sensitivity on the F6 wing for the mesh adjoint using DGM and mesh adjoint using LE



x-component



y-component



z-component

Figure A.10: Comparison of the upper moment surface sensitivity on the F6 wing for the mesh adjoint using DGM and mesh adjoint using LE

LIST OF REFERENCES

[1] Airbus, "Global market forecast, flying by numbers." `http://www.airbus.com/company/market/forecast/`, 2016. Online; accessed 5-June-2016.

[2] M. Nishihara, D. Gaitonde, I. V. Adamovich, and A. Klotchko, "Effect of Nanosecond Pulse Discharges on Oblique Shock and Shock Wave - Boundary Layer Interaction," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, AIAA paper*, (Grapevine, Texas), January 2013. AIAA 2013-0461.

[3] W. S. Wong, *Mechanisms and Optimisations of 3D Shock Control Bumps*. PhD thesis, The University of Sheffield, October 2006.

[4] H. Ogawa, H. Babinsky, M. Pätzold, and T. Lutz, "Shock-wave/boundary-layer interaction control using three-dimensional bumps for transonic wings," *AIAA journal*, vol. 46, no. 6, pp. 1442–1452, 2008.

[5] P. J. K. Bruce, S. P. Colliss, and H. Babinsky, "Three-dimensional shock control bumps: effects of geometry," in *52nd AIAA Aerospace Sciences Meeting and Exhibit*, (Washington, Maryland), January 2014. AIAA 2014-0943.

[6] S. P. Colliss, H. Babinsky, K. Nübler, and T. Lutz, "Vortical structures on three-dimensional shock control bumps.," in *51st AIAA Aerospace Sciences Meeting and Exhibit*, (Grapevine, Texas), January 2014. AIAA 2014-0943.

[7] P. J. K. Bruce and S. P. Colliss, "Review of research into shock control bumps," *Shock Waves*, 2014.

[8] T. C. Tai, "Theoretical aspects of dromedaryfoil.," Tech. Rep. 77-0104, David W. Taylor Naval Ship Research and Development Center, November 1977.

[9] J. L. Fulker, P. R. Ashill, and M. J. Simmons, "Study of simulated active control of shock waves on an aerofoil," Tech. Rep. 93025, DERA, May 1993.

[10] P. R. Ashill and J. L. Fulker, "A novel technique for controlling shock strength of laminar-flow aerofoil sections," *DGLR Bericht*, pp. 175–175, March 1992.

[11] A. Jameson, "Optimum aerodynamic design using control theory," *Computational Fluid Dynamics Review 1995*, 1995.

[12] A. Le Moigne and N. Qin, "Variable-fidelity aerodynamic optimization for turbulent flows using a discrete adjoint formulation," *AIAA Journal*, vol. 42, no. 7, pp. 1281–1292, 2004.

[13] M. Widhalm, J. Brezillon, C. Ilic, and T. Leicht, "Investigation on adjoint based gradient computations for realistic 3d aero-optimization," in *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Multidisciplinary Analysis Optimization Conferences*, (Fort Worth, Texas), September 2010. AIAA 2010-9129.

[14] E. J. Nielsen and M. A. Park, “Using an adjoint approach to eliminate mesh sensitivities in computational design,” *AIAA journal*, vol. 44, no. 5, pp. 948–953, 2006.

[15] M. Park, L. Green, R. Montgomery, and D. Raney, “Determination of stability and control derivatives using computational fluid dynamics and automatic differentiation,” in *17th Applied Aerodynamics Conference, AIAA paper*, 1999. AIAA 1999–3136.

[16] S. K. Rallabhandi, E. J. Nielsen, and D. B., “Sonic-boom mitigation through aircraft design and adjoint methodology,” *Journal of Aircraft*, vol. 51, no. 2, pp. 502–510, 2014.

[17] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, “Su2: An open-source suite for multiphysics simulation and design,” *AIAA journal*, vol. 54, no. 3, pp. 828–846, 2016.

[18] N. Kroll and J. K. Fassbender, *Flow Control: Passive, Active, and Reactive Flow Management*, vol. 89. Springer Science & Business Media, 2006.

[19] Y. B. Suzen and P. G. Huang, “Simulations of flow separation control using plasma actuators,” in *44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA paper*, (Reno, Nevada), 2006. AIAA 2006-877.

[20] D. M. Orlov and T. C. Corke, “Numerical simulation of aerodynamic plasma actuator effects,” in *43 rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA paper*, (Reno, Nevada), January 2005. AIAA 2005-1083.

[21] M. L. Post and T. C. Corke, “Separation Control on HIgh Angle of Attack Airfoil Using Plasma Actuators,” *AIAA Journal*, vol. 42, pp. 2177–2184, Nov. 2004.

[22] D. P. Rizzetta and M. R. Visbal, “Numerical investigation of plasma-based flow control for transitional highly-loaded low-pressure turbine,” *AIAA journal*, vol. 45, no. 10, pp. 2554–2564, 2007.

[23] E. Stanewsky, J. Délery, J. Fulker, and W. Geissler, *EUROSHOCK-Drag Reduction by Passive Shock Control: Results of the Project EUROSHOCK, AER2-CT92-0049 Supported by the European Union, 1993–1995*, vol. 56. Springer Science & Business Media, 1997.

[24] E. Stanewsky, J. Délery, J. Fulker, and P. de Matteis, *Drag Reduction by Shock and Boundary Layer Control: Results of the Project EUROSHOCK II. Supported by the European Union 1996-1999*, vol. 80. Springer Science & Business Media, 2002.

[25] A. N. Smith, H. Babinsky, J. L. Fulker, and P. A. Ashill, “Control of normal shock wave/turbulent boundary-layer interaction using streamwise slots,” in *41st AIAA Aerospace Sciences Meeting and Exhibit, AIAA paper*, (Reno, Nevada), January 2001. AIAA 2001-0739.

[26] A. N. Smith, H. Babinsky, P. C. Dhanasekaran, A. M. Savill, and D. W. N., "Computational investigation of groove controlled shock wave.boundary layer interaction," in *39th AIAA Aerospace Sciences Meeting and Exhibit, AIAA paper*, (Reno, Nevada), January 2003. AIAA 2001-0739.

[27] N. Qin, D. Monet, and S. T. Shaw, "3d bumps for transonic wing shock control and drag reduction," in *CEAS Aerospace Aerodynamics Research Conference*, (Cambridge, UK), Royal Aeronautical Society, June 11-12 2002.

[28] T. Yun, P. Liu, and Z. Li, "Multi-objective optimization of shock control bump on a supercritical wing," *Sci China Tech Sci*, 2008.

[29] F. Zhu, B. Hinchliffe, and N. Qin, "Using mesh adjoint for shock bump deployment and optimisation on transonic wings," in *53rd AIAA Aerospace Sciences Meeting, AIAA SciTech*, (Kissimmee, Florida), January 2015. AIAA 2015-1488.

[30] N. Qin, W. S. Wong, and A. Le Moigne, "Three-dimensional contour bumps for transonic wing drag reduction," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 222, pp. 619–629, March 2008.

[31] B. König, M. Pätzold, T. Lutz, E. Krämer, H. Rosemann, K. Richter, and H. Uhlemann, "Numerical and experimental validation of three-dimensional shock control bumps," *Journal of Aircraft*, vol. 46, no. 2, pp. 675–682, 2009.

[32] W. S. Wong, A. Le Moigne, and N. Qin, "Parallel adjoint-based optimisation of a blended wing body aircraft with shock control bumps," *Aeronautical Journal*, vol. 111, pp. 165–174, March 2007.

[33] M. Pätzold, T. Lutz, E. Krämer, and S. Wagner, "Numerical optimization of finite shock control bumps," in *44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV. AIAA Paper*, January 2006. AIAA 2006-1054.

[34] J. P. Eastwood and J. P. Jarrett, "Toward designing with three-dimensional bumps for lift/drag improvement and buffet alleviation," *AIAA journal*, vol. 50, no. 12, pp. 2882–2898, 2012.

[35] H. A. Holden and H. Babinsky, "Shock/boundary layer interaction control using 3d devices," in *41st AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV. AIAA Paper*, January 6-9 2003. (2003-0447).

[36] N. R. Jones, J. P. Eastwood, and J. P. Jarrett, "Adapting three-dimensional shock control bumps for swept flows.," *AIAA journal*, pp. 1–13, 2016.

[37] A. R. McGowan, "Avst morphing project research summaries in fiscal year 2001," tech. rep., NASA, Langley Research Centre, Hampton, Virginia, Tech. Rep. NASA/TM-2002-211769. 2002.

[38] H. P. Monner, E. Breitbach, T. Bein, and H. Hanselka, "Design aspects of the adaptive wing-the elastic trailing edge and the local spoiler bump," *Aeronautical Journal*, vol. 104, no. 1032, pp. 89–95, 2000.

[39] E. Stanewsky, "Aerodynamic benefits of adaptive wing technology," *Aerospace Science and Technology*, vol. 4, no. 7, pp. 439–452, 2000.

[40] E. Jinks, P. Bruce, and M. Santer, "The use of actuated flexible plates for adaptive shock control bumps," in *53rd AIAA Aerospace Sciences Meeting, AIAA paper*, (San Diego, California), January 2015.

[41] E. Jinks, P. J. Bruce, and M. J. Santer, "Aero-structural design optimization of adaptive shock control bumps," in *54th AIAA Aerospace Sciences Meeting, AIAA paper*, (San Diego, California), January 2016.

[42] D. S. Lee, J. Periaux, E. Onate, L. F. Gonzalez, and N. Qin, "Active Transonic Aerofoil Design Optimization Using Robust Multiobjective Evolutionary Algorithms," *Journal of Aircraft*, vol. 48, pp. 1084–1094, May 2011.

[43] K. Mazaheri, A. Nejati, and K. Chaharlang Kiani, "Application of the adjoint multi-point and the robust optimization of shock control bump for transonic aerofoils and wings," *Engineering Optimization*, pp. 1–23, 2016.

[44] F. Zhu, *Geometric Parameterisation and Aerodynamic Shape Optimisation*. PhD thesis, University of Sheffield, 2014.

[45] B. M. Kulfan, J. E. Bussoletti, *et al.*, "Fundamental parametric geometry representations for aircraft component shapes," in *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, vol. 6948, 2006.

[46] T. Gerhold, O. Friedrich, J. Evans, and M. Galle, "Calculation of complex three-dimensional configurations employing the dlr-tau-code," in *35th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings*, (Reno, Nevada), January 1997. AIAA 1997-0167.

[47] R. Rudnik, R. Heinrich, B. Eisfeld, and T. Schwarz, "Dlr contributions to code validation activities within the european high lift project eurolift," in *New Results in Numerical and Experimental Fluid Mechanics IV*, pp. 42–49, Springer, 2004.

[48] N. Kroll and J. K. Fassbender, *MEGAFLOW-Numerical Flow Simulation for Aircraft Design: Results of the second phase of the German CFD initiative MEGAFLOW, presented during its closing symposium at DLR, Braunschweig, Germany, December 10 and 11, 2002*, vol. 89. Springer Science & Business Media, 2006.

[49] D. Schwamborn, T. Gerhold, and R. Heinrich, "The dlr tau-code: Recent applications in research and industry," in *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*, Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006. 1997-0167.

[50] A. Jameson, W. Schmidt, and E. Turkel, "Numerical solution of the euler equations by finite volume methods using runge kutta time stepping schemes," in *14th fluid and plasma dynamics conference*, p. 1259, 1981.

[51] R. P. Dwight and J. Brezillon, "Effect of approximations of the discrete adjoint on gradient-based optimization," *AIAA journal*, vol. 44, no. 12, pp. 3022–3031, 2006.

[52] R. P. Dwight, "Robust mesh deformation using the linear elasticity equations," in *Computational Fluid Dynamics*, pp. 401–406, Springer, 2009.

[53] X. Liu, N. Qin, and H. Xia, "Fast dynamic grid deformation based on Delaunay graph mapping," *Journal of Computational Physics*, vol. 211, pp. 405–423, Jan. 2006.

[54] T. J. Baker and P. A. Cavallo, "Dynamic adaptation for deforming tetrahedral meshes," *AIAA Journal*, vol. 19, pp. 2699–3253, 1999.

[55] K. Stein, T. Tezduyar, and R. Benney, "Mesh moving techniques for fluid-structure interactions with large displacements," *Journal of Applied Mechanics*, vol. 70, no. 1, pp. 58–63, 2003.

[56] Z. Yang and D. J. Mavriplis, "Unstructured dynamic meshes with higher-order time integration schemes for the unsteady navier-stokes equations," in *43th AIAA Aerospace Sciences Meeting and Exhibit, AIAA paper*, (Reno, Nevada), January 2005. AIAA 2005-1222.

[57] T. Xiao, N. Qin, D. Luo, and S. Deng, "Deformable overset grid for unsteady aerodynamic simulation," in *54th AIAA Aerospace Sciences Meeting, AIAA SciTech*, (San Diego, California), 2016. AIAA 2016-2052.

[58] A. Le Moigne, *A discrete Navier-Stokes adjoint method for aerodynamic optimisation of Blended Wing-Body configurations*. PhD thesis, Cranfield University, December 2002.

[59] J. J. Alonso, "Introduction to sensitivity analysis and the adjoint method: Derivations, uses and properties," in *38^{th} Advanced CFD Lectures Series; Von Karman Institute for Fluid Dynamics*, von Karman Institute for Fluid Dynamics, September 14–16 2015.

[60] H. J. Kim, C. Kim, O. H. Rho, and S. Lee, "Aerodynamic sensitivity analysis for navier-stokes equations," in *37th AIAA Aerospace Sciences Meeting and Exhibit, AIAA paper*, (Reno, Nevada), 1999. AIAA 1999-0402.

[61] G. J. W. Hou, V. Maroju, A. C. Taylor, V. M. Korivi, and P. A. Newman, "Transonic turbulent airfoil design optimization with automatic differentiation in incremental iterative forms," in *12th AIAA Computational Fluid Dynamics Conference, AIAA paper*, (San Diego, California), 1995. AIAA 1995-1692.

[62] L. L. Green, P. A. Newman, and K. J. Haigler, "Sensitivity derivatives for advanced cfd algorithm and viscous modeling parameters via automatic differentiation," *Journal of Computational Physics*, vol. 125, no. 2, pp. 313–324, 1996.

[63] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats, "An assessment of airfoil design by numerical optimization," Tech. Rep. NASA-TM-X-3092, NASA, 1974.

[64] R. M. Hicks and G. N. Vanderplaats, "An assessment of airfoil design by numerical optimization," Tech. Rep. NASA-TM-X-3213, NASA, 1975.

[65] J. N. Lyness and C. B. Moler, "Numerical differentiation of analytic functions," *SIAM Journal on Numerical Analysis*, vol. 4, no. 2, pp. 202–210, 1967.

[66] J. N. Lyness, "Numerical algorithms based on the theory of complex variable," in *Proceedings of the 1967 22nd national conference*, pp. 125–133, ACM, 1967.

[67] W. Squire and G. Trapp, "Using complex variables to estimate derivatives of real functions," *Siam Review*, vol. 40, no. 1, pp. 110–112, 1998.

[68] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso, "The complex-step derivative approximation," *ACM Transactions on Mathematical Software (TOMS)*, vol. 29, no. 3, pp. 245–262, 2003.

[69] A. Keane and P. Nair, *Computational approaches for aerospace design: the pursuit of excellence*. John Wiley & Sons, 2005.

[70] N. R. Gauger, A. Walther, C. Moldenhauer, and M. Widhalm, "Automatic differentiation of an entire design chain for aerodynamic shape optimization," in *New Results in Numerical and Experimental Fluid Mechanics VI*, pp. 454–461, Springer, 2007.

[71] J. Elliott and J. Peraire, "Practical three-dimensional aerodynamic design and optimization using unstructured meshes," *AIAA journal*, vol. 35, no. 9, pp. 1479–1485, 1997.

[72] H. J. Kim, D. Sasaki, S. Obayashi, and K. Nakahashi, "Aerodynamic optimization of supersonic transport wing using unstructured adjoint method," *AIAA journal*, vol. 39, no. 6, pp. 1011–1020, 2001.

[73] O. Baysal and M. E. Eleshaky, "Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics," *AIAA journal*, vol. 30, no. 3, pp. 718–725, 1992.

[74] J. M. Lacasse and O. Baysal, "Design optimization of single-and two-element airfoils on multiblock grids.," in *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper*, (Panama City, Florida), 1994. AIAA 1994-4273.

[75] A. C. Taylor, III, G. W. Hou, and V. M. Korivi, "Methodology for calculating aerodynamic sensitivity derivatives," *AIAA journal*, vol. 30, no. 10, pp. 2411–2419, 1992.

[76] V. M. Korivi, A. C. Taylor, III, G. W. Hou, P. A. Newman, and H. E. Jones, "Sensitivity derivatives for three-dimensional supersonic euler code using incremental iterative strategy," *AIAA journal*, vol. 32, no. 6, pp. 1319–1321, 1994.

[77] J. L. Lions, *Optimal control of systems governed by partial differential equations*, vol. 170. New York: Springer Verlag, 1971. Translated by S. K. Mitter.

[78] O. Pironneau, "On optimum profiles in stokes flow," *Journal of Fluid Mechanics*, vol. 59, no. 1, pp. 117–128, 1973.

[79] A. Jameson, "Aerodynamic design via control theory," *Journal of scientific computing*, vol. 3, no. 3, pp. 233–260, 1988.

[80] M. Leatham, S. Stokes, J. Shaw, J. Cooper, J. Appa, and T. Blaylock, "Automatic mesh generation for rapid-response navier-stokes calculations," in *FLUIDS 2000 Conference and Exhibit*, (Denver, Colorado), June 2000. AIAA 2000-2247.

[81] C. Ilic, M. Widhalm, and J. Brezillon, "Efficient polar optimization of transport aircraft in transonic rans flow using adjoint gradient based approach," in *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, September 2012.

[82] J. Fulker and M. Simmons, "An experimental study of shock control methods," DRA/AS/HWA/TR94007/1, 1994.

[83] T. R. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 6, no. 26, pp. 369–395, 2004.

[84] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[85] E. Stanewsky, J. Délery, J. L. Fulker, and P. De Matteis, "Drag reduction by shock and boundary layer control, Results of the project EUROSHOCK II, Notes on Numerical Fluid Mechanics and Multidisciplinary Design," *Springer*, vol. 80, 1997.

[86] D. Destarac, "Far-field/near-field drag balance and applications of drag extraction in cfd," *VKI Lecture Series*, vol. 2, November 3-7 2003.

[87] V. Schmitt and F. Charpin, "Pressure distributions on the onera-m6-wing at transonic mach numbers," *Experimental data base for computer program assessment*, AGARD AR-138, May. 1979.

[88] J. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders, "Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1," *Journal of Aircraft*, vol. 36, no. 1, pp. 51–60, 1999.

[89] P. J. K. Bruce and H. Babinsky, "Experimental study into the flow physics of three-dimensional shock control bumps," *Journal of Aircraft*, vol. 49, no. 5, pp. 1222–1233, 2012.

[90] D. W. Levy, K. R. Laflin, E. N. Tinoco, J. C. Vassberg, M. Mani, B. Rider, C. Rumsey, R. A. Wahls, J. H. Morrison, O. P. Brodersen, S. Crippa, D. J. Mavriplis, and M. Murayama, "Summary of data from the fifth aiaa cfd drag prediction workshop," in *51st AIAA Aerospace Sciences Meeting*, (Grapevine, Texas), January 2013. AIAA 2009-3767.