The University Of Sheffield.

Efficiency Improvements to Adjoint-Based Aeroelastic Optimisations using a Trim-Corrected and Hybrid Mesh Deformation Strategy

Anthony Stannard

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

The University of Sheffield
Faculty of Engineering
Department of Mechanical Engineering

May 2022

Abstract

This purpose of this research is to increase the efficiency of the aeroelastic shape optimisation process for commercial aircraft. Aeroelastic simulations capture the interaction between aerodynamic loading and structural displacements. High-fidelity aeroelastic simulations are computationally expensive, hence an adjoint-based approach to aircraft shape optimisation is the most suitable approach when large numbers of design parameters are present. The coupled nature of the fluid-structure interaction (FSI) is reflected in the resulting adjoint equations that are used to find the gradient. Previous coupled-adjoint optimisations performed in literature have used high-fidelity solvers for both computational fluid dynamics (CFD) and computational structural mechanics (CSM) while also satisfying the trim constraints within the FSI simulation. This project builds on those studies by proposing a simple yet powerful control surface parameterisation method for satisfying the trim constraints within the FSI simulation. An additional contribution of this work is an investigation into the effects that different mesh deformation algorithms have on the rate of convergence of the coupled-adjoint.

An important aspect of capturing the FSI is an effective mesh deformation strategy. The algorithm used for deforming the mesh in an FSI simulation needs to be robust to large deformations but also efficient due to the large number of times it will be required. The radial basis function (RBF) mesh deformation strategy with a data-reduction algorithm is a popular method for achieving robust and efficient deformations within FSI simulations. A key contribution of this work is the finding that the application of a data-reduction algorithm to the input field of the mesh deformation strategy has a significantly negative effect on the convergence of the coupled-adjoint whilst having only a negligible effect on the convergence of the FSI simulation. The Delaunay Graph Mapping (DGM) mesh deformation algorithm is employed to obtain faster convergence of the coupled-adjoint than the RBF approach. To increase the efficiency of optimisation process, a hybrid mesh deformation strategy is proposed by using the RBF approach within the FSI simulation and the DGM approach within the coupled-adjoint.

The gradients that are obtained via the hybrid mesh deformation approach are successfully validated. The hybrid mesh deformation strategy is then applied to two optimisation scenarios in the transonic flow region. The first is a lift constrained wing optimisation. The second is a lift and trim constrained optimisation performed on a full transport aircraft configuration. The developed trim-corrected and hybrid mesh deformation optimisation strategy is shown to demonstrate a more efficient coupled-adjoint aeroelastic shape optimisation process.

Publications

Journal Articles

Anthony Stannard and Ning Qin, “Hybrid Mesh Deformation for Aerodynamic-Structural Coupled Adjoint Optimization”, *AIAA Journal*, (Approved for publication)

Acknowledgements

I would like to thank Professor Ning Qin for providing me with the opportunity to study the area of adjoint-based optimisation and particular thanks for persuading me not to quit in 2019.
Special thanks are required for John Pattinson who continuously helped me throughout my studies. His help was totally selfless and without which I could not have completed this thesis.
I would also like to thank Murray Cross for facilitating the project and for his advice throughout my time on this project.
I would also like to acknowledge Airbus UK and EPSRC for funding the research through an ICASE studentship. The HPC computing time on the CFMS system is also acknowledged as it provided the essential backup enabling me to complete my studies despite the disruption brought about by the pandemic in 2020.

Table of Contents

Nomenclature

a	=	Delaunay sub-element area
a	=	Proper orthogonal decomposition interpolation coefficients
a_{ij}	=	Matrix element
$\boldsymbol{a}$	=	Rotation axis
A	=	Delaunay element area
$\boldsymbol{A}$	=	Proper orthogonal decomposition projection matrix
$\boldsymbol{A}$	=	RBF matrix excluding polynomial terms
b	=	RBF volume deformation scaling factor
$\mathbf{B}$	=	Approximation to Hessian matrix
$\boldsymbol{B}$	=	Delaunay boundary node vector
$\boldsymbol{B}$	=	Stiffness matrix geometric component
c_{ijkl}	=	Stiffness tensor
C_D	=	Drag coefficient
C_L	=	Lift coefficient
C_{MY}	=	Pitching moment coefficient
C_p	=	Specific heat at constant pressure
d	=	Distance
$\boldsymbol{D}$	=	Design parameters
$\mathbf{D}$	=	Stiffness matrix material component
e	=	Delaunay area ratio
E	=	Energy
E	=	Young's modulus
$\boldsymbol{E}$	=	Delaunay interpolation matrix
$\boldsymbol{f}$	=	Body force vector
$\boldsymbol{F}_{a_{surf}}$	=	Vector of surface forces on the CFD mesh
$\boldsymbol{F}_s$	=	Vector of surface forces on the CSM mesh
$\mathbf{F}^c$	=	Convective flux tensor

$\mathbf{F}^v$ = Viscous flux tensor

$\boldsymbol{g}$ = Vector of constraint functions

G = Global interpolation matrix

h = Specific enthalpy

h = Step size

H = Enthalpy

$\boldsymbol{H}$ = RBF matrix containing both RBF and polynomial submatrices

i = Imaginary number (Square root of minus one)

I = Cost function

$\boldsymbol{I}$ = Identity matrix

k = Turbulent kinetic energy

$\mathbf{K}$ = Krylov matrix

$\boldsymbol{K}$ = Structural stiffness matrix

L_e = Boolean localization matrix

$\mathcal{L}$ = Lagrangian operator

n_d = Number of domains

n_e = Number of elements

n_m = Number of modes

n_p = Number of points

n_S = Number of surface nodes

n_{Surf} = Number of surface nodes in CFD mesh

n_v = Number of nodes in CFD volume mesh

$\boldsymbol{n}$ = Normal vector

p = Static pressure

$\boldsymbol{P}$ = Polynomial section of RBF matrix

Pr = Prandtl number

q = Heat transfer flux vector

$\boldsymbol{Q}_n$ = An orthogonal matrix

r = Radial distance

R = Gas constant

$\boldsymbol{R}$ = Rotation matrix

$\boldsymbol{R}_a$ = Flow residual vector

$\boldsymbol{R}_m$ = Mesh deformation residual

$\boldsymbol{R}_n$ = An upper triangle matrix

$\boldsymbol{R}_s$ = Structural residual vector

s = Inequality constraint slack variable

S = Sutherland's temperature constant

S = Surface boundary

S_{ij} = Strain tensor

$\boldsymbol{S}$ = CFD surface mesh

$\mathbf{S}$ = Vector of surface coordinates

$\boldsymbol{S}_E$ = Energy source term

$\boldsymbol{S}_M$ = Momentum source term

t = Time

$\mathbf{t}$ = Surface traction

u = Velocity in x-direction

$\boldsymbol{u}$ = Velocity vector

$\boldsymbol{u}_{a_{surf}}$ = Surface displacement field on the CFD mesh

$\boldsymbol{u}_{a_{vol}}$ = Volume displacement field on the CFD mesh

$\boldsymbol{u}_s$ = Surface displacement field on the CSM mesh

v = Kinematic viscosity

v = Velocity in y-direction

w = Test function

w = Inverse distance weighting interpolation function

w = Velocity in z-direction

$\boldsymbol{W}$ = Flow state variables

$\boldsymbol{X}_J$ = CFD volume mesh at the jig shape

$\boldsymbol{X}_{\Delta}$ = CFD volume mesh at aeroelastic equilibrium

$\boldsymbol{Y}$ = Snapshot matrix arising from Design of Experiments

α = Angle of attack

α = Line search parameter

β = Vector of coefficients for radial basis functions

γ = Ratio of specific heats

γ = Vector of coefficients for radial basis functions

δ = Control surface deflection angle, °

δ = Kronecker's delta

$\partial\Omega$ = Control surface

ε = Euclidean distance

$\varepsilon_{i,j}$ = Distance between point i and point j

$\boldsymbol{\varepsilon}$ = Strain tensor

$\boldsymbol{\zeta}$ = Coefficients of the RBF interpolation

$\boldsymbol{\eta}$ = Coefficients of the RBF interpolation

θ = Aitken acceleration factor

κ = Coefficient of thermal conductivity

λ = Eigenvalue of a matrix

λ = Lagrangian multiplier

$\boldsymbol{\lambda}$ = Lamé parameter

$\boldsymbol{\lambda}$ = Second coefficient of viscosity

$\boldsymbol{\lambda}_a$ = Flow-adjoint vector

$\boldsymbol{\lambda}_s$ = Structural-adjoint vector

μ = Dynamic viscosity

μ = Lamé parameter

μ_t = Eddy viscosity term

ν = Poisson's ratio

ρ = Density

σ = Normal stress

τ = Shear stress

φ = Radial basis function

φ = Shape basis function

ϕ = Proper orthogonal decomposition modes

Ω = Control volume

Ω = Magnitude of vorticity

Subscripts

a = Aerodynamic mesh variable

CS = Control surface

D = Delaunay vertex node

$\boldsymbol{D}$ = Derivative with respect to $\boldsymbol{D}$

eff = The effective value

E = A term defined in terms of energy

i, j, k = Indices

J = Jig shape

m = Mesh

M = A term defined in terms of energy

s = Structural mesh variable

$surf$ = Surface CFD mesh

T = Turbulent term

vol = Volume CFD mesh

x = Acting in the x-direction

y = Acting in the y-direction

z = Acting in the z-direction

0 = Baseline value

ρ = Cartesian axis

Δ = Aeroelastic equilibrium shape

∞ = The freestream value

Superscripts

c	=	Convective term
e	=	The specific element of a mesh
T	=	Transpose operator
v	=	Viscous term
$*$	=	Converged term

List of Figures

List of Tables

Declaration

I, Anthony Stannard, confirm that the Thesis is my own work. I am aware of the University's Guidance on the Use of Unfair Means (www.sheffield.ac.uk/ssid/unfair-means). This work has not been previously been presented for an award at this, or any other, university

1. Introduction

Civil aviation is an industry that has made the world a more accessible place for everyone. People can travel to locations around the world that would have been accessible to a very limited number of people only two generations ago. Climate change is caused by greenhouse gas emissions. The aviation industry is responsible for a significant proportion of global emissions. Across commercial aviation, significant resources have been committed to reduce greenhouse gas emissions. The EU commissioned report, Flight Path 2050 [1], has set a series of targets for the industry to achieve by the year 2050. Specifically, these targets aim to reduce CO_2 emissions per passenger kilometer by 75%, noise emissions by 65% and NO_x emissions by 90%.

The most recent global market forecast by Airbus [2] published in 2019 predicted a doubling of aircraft traffic over the next 15 years. While the ramifications of the COVID-19 pandemic on the commercial aviation industry are not yet known, it remains likely that air traffic will continue to grow in the next decades. This means major advancements in aviation technology must be achieved and implemented quickly. The aircraft industry is relentlessly pursuing more advanced designs to improve efficiency as much as possible. Aircraft optimisation is going to be key in reaching these objectives. Strategies aimed at improving the aerodynamic efficiency of an aircraft at transonic speeds have been pursued vigorously in both academia and industry. The studies conducted by Daoud *et al.* [3] and Schuhmacher *et al.* [4] show the first integration of aerodynamic optimisations into the design process for commercial aircraft. Aerodynamic optimisation techniques have reached maturity in the aviation industry over the last decade and they have been applied to a wide range of problems in aircraft design.

Optimising with respect to specific disciplines on their own has already significantly improved aircraft efficiency. Many authors have demonstrated the use of various aerodynamic optimisation techniques to improve aircraft efficiency [5]–[12]. Integrating multi-disciplinary optimisation (MDO) chains in an industrial context has been much more challenging. The use of MDO in industry is not always warmly received, in 1994 Kroo *et al.* [13] noted this with the proclamation: “When an aircraft designer hears that a new program will use multidisciplinary optimisation, the reaction is often less than enthusiastic”. However the field of MDO has significantly advanced since then and ignoring the strong coupling between all the disciplines that affect an aircraft during the design phase will increase the development risks. The use of MDO while developing a new aircraft enables realistic designs that fulfil the constraints of the involved disciplines to be realised faster. A challenge that emerges from the use of MDO is its higher computational cost. In 1995 Dudley *et al.* [14] noted that it is typically more computationally expensive to perform an MDO than perform a single disciplinary optimisation for each engaged discipline, that is still the case today.

The underlying objective behind most MDOs undertaken in commercial aviation is to reduce the fuel burn of an aircraft for any given flight. Two clear ways to achieve this aim involve improving the aerodynamic efficiency of the aircraft and to reduce the aircraft’s structural weight. These two objectives are strongly coupled. As a wing deforms in flight due to its aerodynamic loading, the flow around it will change and therefore modify the pressure distribution, potentially drastically [15]. Reducing the weight of the wing or modifying its shape changes what the shape of the wing will be in flight. This modified shape affects the aerodynamic performance of the wing. The interaction of flexible structures and fluid flow is a challenging problem. The field of computational fluid-structure interaction (FSI) was initially tackled in the 1970s and 1980s by Belytschko [16], Bathe and Hahn [17] and Donea *et al.* [18]. Since then, the remarkable increase in computational power and development of fast numerical methods have made it feasible to simulate FSI problems with high-fidelity models.

However, the computational cost of a single high-fidelity FSI simulation remains large and therefore, any FSI-based optimisation must use efficient algorithms for it to be useful.

The large cost of a single FSI simulation prohibits the use of gradient-free optimisation schemes that have large design spaces. It is possible to employ a surrogate model or metamodel to reduce the computational cost [19] but the accuracy of surrogate models is not guaranteed and they are often only relevant to specific problems [20], [21]. Laurent *et al.* [22] showed that gradient information can be exploited to improve the accuracy of metamodels therefore making the metamodel approach more viable with larger design spaces as less objective function evaluations would be required. However, the need for high-fidelity models with a large design space and the need to consider complex geometries leads to the choice of this thesis to employ a gradient-based optimisation scheme. Gradient-based algorithms are discussed in detail in the literature review section. It is determined that an adjoint-based optimisation scheme offers the best approach for optimising complex geometries in the shortest time.

In the case of an aeroelastic optimisation, one where no structural parameters change, why not just perform a purely aerodynamic optimisation and reverse engineer the jig shape that produces the optimum shape? There are two drawbacks to this approach. The first is that there is no guarantee that the required jig shape is feasible. The second is that a multi-point approach to find the optimum jig shape cannot be implemented. A multi-point optimisation here refers to an optimisation that considers several operating conditions, not just the design condition. While a multipoint optimisation to find the optimum flight shape can be performed, this does not translate to an optimised jig shape. This is because the jig shape derived from the optimised flight shape will deform to different shapes at different flight conditions. A shape which is optimised only for one point will be poor away from the design point [23]. An aeroelastic optimisation scheme can mitigate for this by implementing a multi-point jig shape optimisation. This means the jig shape can be optimised for directly, allowing a multi-point optimisation approach to mitigate against poor off-design performance.

1.1 Background on Aerodynamic Optimisation

The purpose behind creating optimisation tools in commercial aviation is to improve the performance of an aircraft with respect to a given metric. For shape optimisation, this metric is typically the aerodynamic drag of the aircraft. Optimisations can be separated into two major categories. These are optimisations based on either gradient-free algorithms or gradient-based algorithms. The aerodynamic drag function can be multimodal. For gradient-free algorithms, this does not pose a problem as they will not get stuck in a local minimum. A challenge for gradient-free algorithms is that they typically require many more evaluations of the objective function than a gradient-based algorithm would, although techniques such as meta-modelling can mitigate this. This is particularly a problem for high-fidelity aerodynamic shape optimisations as each flow evaluation is computationally expensive. However, with the ever-increasing capacity and speed of computers, many gradient-free optimisations have been performed. Jahangirian and Shahrokhi [24] used a genetic algorithm and an unstructured CFD solver to optimise an airfoil in transonic conditions. Hashimoto *et al.* [25] used a genetic algorithm to maximise the lift generation of high-wing aircraft configurations and they used an in-house RANS solver to solve the flow problem. They do not mention the length of time taken for the optimisations to complete, but the number of flow evaluations required in each were over 100 which suggests the optimisations took a large amount of CPU time. In the mentioned gradient-free optimisations, the number of design parameters were kept low to prevent the computational time

becoming too expensive. A means to reducing the computational time required for high-fidelity shape optimisations is to use a gradient-based algorithm.

By obtaining a gradient of the cost-function with respect to the design parameters, a search direction is acquired which allows the flow evaluations to be more targeted and thus less in number. The downside of such algorithms occurs when the cost-function being optimised is multimodal. A gradient-based algorithm searching for a minimum will always converge at the first minimum it approaches. Another potential downside is that it requires the optimisation's cost-function and constraints, to be continuous. Some aerodynamic optimisation cases of interest, such as those constraining flutter have behaviour that is discontinuous. Supercritical bifurcation of flutter is such a phenomenon [26] thus making it difficult to handle in a gradient-based optimisation. There are two main tasks a gradient-based algorithm undertakes each iteration. The first is obtaining the gradient which is used to obtain the search direction, the second is determining how far to move the design parameters in the search direction.

The gradient is directly related to the search direction as the gradient shows the direction of steepest ascent at the given design point. The search direction is also dependent on the optimisation algorithm that is employed. The steepest descent algorithm is the simplest method. At each design iteration, the steepest descent algorithm moves the design parameters in the opposite direction of the gradient. Other gradient-based algorithms with better convergence criteria are more commonly used to obtain the search direction at a particular design iteration. An introduction to gradient-based algorithms for optimisation are presented in Appendix A.

There are multiple ways to obtain the gradient at a given design point. The simplest and oldest method, provided that the gradient cannot be found analytically, is finite differences. One issue that arises when obtaining a gradient via finite differences is deciding upon which step size to use. The mathematical error that arises from a finite difference gradient that uses a step size of size h is of the order $O(h)$. Ideally, an infinitesimal step size could be used so that the order of the error is also infinitesimal. However, computers can only store numbers to a limited precision. A float type in the programming language C can handle a number to a precision of 7 decimal points, this takes 32 bits of memory. The double type can handle a number to a precision of 15 decimal points, but it takes 64 bits of memory. The limited precision of CFD solvers means that subtractive cancellation errors occur. As a consequence of this, a step size that is too small would result in a significant amount of cancellation error being present in the output gradient [27]. While it would be possible to write a CFD application that stores numerical values to an even greater precision, thus enabling a more accurate finite difference, this would be a poor use of memory and result in an inefficient solver.

The major drawback of the finite differences approach to obtaining the gradient is that each design parameter must be perturbed individually. If there are n design parameters, then there must be $n + 1$ flow evaluations to obtain the gradient. This quickly becomes prohibitive as the number of design parameters increases. The adjoint method provides a means to obtain the gradient with only 1 flow evaluation. The adjoint method has its beginnings in optimal control theory [28 - 29] in the 1960s. As the adjoint method enables a gradient to be found while only requiring a single flow evaluation, it makes the optimisation of shapes with large numbers of design parameters possible. As a result it forms the basis of this thesis. The state of the art in the adjoint method and its applications to aeroelastic optimisations are examined in the literature review chapter.

1.2 Objectives of the Thesis

This thesis aims to investigate new methods to increase the efficiency of aeroelastic optimisations whilst maintaining the feasibility of the optimised designs. The research uses state of the art mesh deformation and CFD to CSM interpolation techniques to produce an optimisation scheme that is applicable to complex geometries seen in commercial aviation. A destabilising effect of the employed mesh deformation scheme is discovered within the coupled-adjoint and a novel coupled-adjoint scheme is proposed to remedy this problem. Finally, a control surface parameterisation scheme is introduced to trim the aircraft during the FSI simulation. The sensitivity of this parameterisation scheme is derived and included in a new approach to performing aeroelastic optimisations. The individual objectives are listed below, and the outline of the thesis is provided in the following section.

- Employ mesh deformation algorithms in the FSI simulation and coupled-adjoint procedure that are more efficient than the conventional iterative mesh deformation methods. The algorithm employed in the FSI simulation must be robust enough to handle the large displacements that can occur due to aeroelastic effects.
- Conduct a study to investigate how the chosen mesh deformation algorithm effects the convergence of the FSI simulation and the coupled-adjoint procedure. The effects to investigate include both the overall time taken by the algorithm and also the number of iterations the procedures take to converge.
- Investigate the viability of using a different mesh deformation algorithm in the FSI simulation and the coupled-adjoint for the purposes of performing an aeroelastic optimisation. This approach is known as a hybrid mesh deformation strategy. The gradient produced by the hybrid mesh deformation strategy must only have negligible differences when compared to the gradient produced by a non-hybrid mesh deformation approach.
- Use a control surface parameterisation scheme so the pitching moment constraint can be satisfied within the optimisation. Furthermore, derive the sensitivity of the mesh to this parameterisation scheme so it can be implemented within an adjoint-based aeroelastic optimisation.
- Develop the software that will perform adjoint-based aeroelastic optimisations. The software to be developed must have the following capabilities:
    - Integrate an open-source structural solver into the codebase to enable FSI simulations and coupled-adjoint procedures.
    - Perform the coupled-adjoint chain for the specified cost-function so the gradient of the aircraft shape can be obtained.
    - Calculate the sensitivity of the CFD mesh to the control surface parameter.
    - Extend the gradient calculation to handle implicit parameters which are used inside the FSI simulation to satisfy the lift and pitching moment constraints.
    - Deform unstructured meshes using a number of mesh deformation algorithms and perform their transpose operations.
    - Extend the FSI simulation code with the trim-corrected and hybrid mesh deformation strategy.
    - Handle the overall optimisation chain.

1.3 Thesis Outline

The thesis begins by reviewing the state of the art with regards to the adjoint method for aerodynamic shape optimisation and aerostructural optimisation. The increasing maturity of the coupled-adjoint method for aerostructural optimisations is discussed and key areas which deserve further research are highlighted.

The methods which this study is built from are described in chapter 3. The methodology and tools used for the FSI simulations are introduced and the justification for their use in this work is presented. The adjoint method for aerodynamic optimisations is then introduced and from this the equations for coupled-adjoint equations are derived.

The novel contributions of this thesis are presented in chapter 4. The effect of using an explicit mesh deformation strategy on the coupled-adjoint formulation is examined. Following on from this, an observation that the popular RBF mesh deformation strategy with an accompanying data-reduction algorithm introduces a destabilising effect on the coupled-adjoint is shown. The reasons for this destabilising effect are investigated and an effective remedy is proposed with additional efficiency benefits. Following on from this, an approach for handling longitudinal trim within the aeroelastic optimisations is described. This approach is then combined with a flexible and easy to implement parameterisation scheme for control surfaces creating a powerful and novel approach to coupled-adjoint aeroelastic optimisations.

Chapter 5 utilises the proposed scheme for aeroelastic shape optimisation on two test cases. The first is a simple wing case designed to investigate divergence at transonic speeds. The second is an industrially relevant wide-bodied twin engine aircraft. The aerodynamic properties of the optimised configurations are then discussed. Finally, chapter 6 summarises the findings of the thesis and suggestions for future research are provided.

1.4 Test Cases used in the Thesis

The first test case employed for investigations and validating the methods of this thesis is the LANN wing [30]. The LANN wing is a moderate-aspect ratio transport wing configuration that was designed to measure unsteady pressures at transonic speeds [31].

Fig. 1 The converged shape of the LANN wing at a flow condition of Mach 0.73 and a 2° angle of attack.

The CFD mesh of the LANN wing is relatively low fidelity consisting of only 37,000 nodes in the whole volume mesh. The reason for using this low fidelity mesh was to aid in speeding up the development process.

The second test case used in this thesis is the Airbus XRF1. The XRF1 is an Airbus provided industrial standard multi-disciplinary research test case representing a typical configuration for a long-range wide body aircraft.

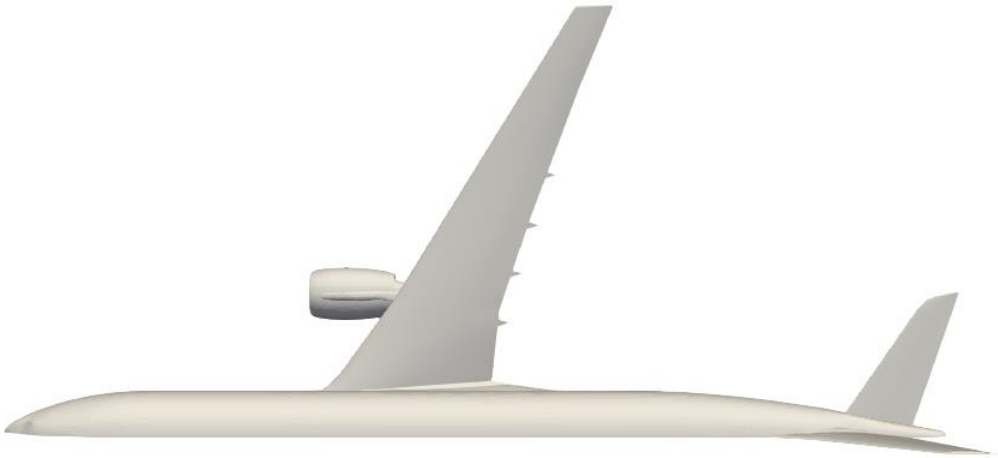

Fig. 2 The XRF1 aircraft model.

The CFD mesh for the XRF1 has a considerably higher fidelity than the LANN wing and it consists of 4.9 million nodes. This mesh represents the right side of the aircraft with a symmetry plane through the fuselage.

2. Literature Review

2.1 The Adjoint Method for Aerodynamic Optimisations

The adjoint method was first demonstrated to have a place in aerodynamic design by Jameson [32] in 1988. There are two main approaches to the formulation of the adjoint equation. They can either be formed by the continuous or discrete approach. The continuous approach is where the governing flow equations are first linearised before their adjoint is constructed and discretised. The discrete approach sees the discretisation occur at the start of the process. Nadarajah and Jameson [33] compared the performance of the continuous and discrete adjoint. They found that the discrete adjoint gradient had better agreement than the continuous adjoint gradient, which was expected as the discrete approach is applied directly to the discretised flow equations. At the time they concluded there was no particular benefit to using the discrete adjoint due to its greater computational cost. In the time since their paper, discrete adjoint optimisations are seen more frequently mostly due to their better agreement with finite difference gradients and that they can be directly applied to unstructured grids without special treatment [34]. Anderson and Venkatakrishnan [35] uncovered a significant difficulty with the continuous adjoint when applying it to an unstructured grid. Obtaining the shape sensitivity in viscous flows requires accurate second derivatives of the velocities to be calculated. For the continuous adjoint, these can be obtained by mapping the domain to a fixed computational coordinate system. This is doable for structured grids but not generally doable for unstructured grids. Without such a mapping, the sensitivity of the cost functions to the grid do not appear in the continuous adjoint. However, the grid sensitivities are critical for obtaining accurate gradients when geometries with singularities are present. For these reasons, they make the explicit recommendation to use the discrete adjoint.

A problem that arises when performing the discrete adjoint is the high levels of memory that the process requires. To solve the discrete adjoint, the full discrete flux Jacobian needs to be stored in memory [36]. A high-fidelity 3D aircraft model will consist of many millions of nodes. If a one equation turbulence model is used, there will be six equations to solve at each node and six state variables. The size of the matrix to be stored in memory will then be $(6 \times n_v) \times (6 \times n_v)$. Of course, not all of the matrix is non-zero. The number of non-zero elements is determined by the spatial stencil discretisation. The stencil of a cell refers to the number of surrounding cells information is taken from when solving the cell's governing equations. In the flow solver, this is often next nearest neighbour [37]. To calculate the governing equations for a given cell, a next nearest neighbour scheme uses the information of cells up to two away. Dwight and Brezillon [36] showed that the memory required by the flux Jacobian can be halved by assuming only the immediate neighbouring cells' information are required. Although still very large, modern high-performance computers are able to handle these large memory requirements. Another factor that affects the accuracy of the obtained gradient is the level of convergence of the flow solution [33]. In modern day CFD solvers, an exact match of the gradient will not be produced as CFD solvers use techniques such as shock switches which discontinuous and as such, these cannot properly be differentiated. However, a gradient with only small errors is sufficient for quickly optimising an aircraft.

2.1.1 Adjoint-Based Aeroelastic Optimisation

In the decades that have followed Jameson's original paper on adjoint-based aerodynamic optimisation, huge amounts of research has been dedicated to improving it and applying it to more and more complex configurations with excellent results [38]. A major innovation was pioneered by Martins [39] when he coupled the aerodynamic adjoint method with a structural adjoint to create a coupled adjoint approach for aerostructural optimisation. This enabled the effects of aeroelasticity to be included, allowing jig shapes rather than flight shapes to be optimised with the adjoint method. Prior to this, an optimisation of the jig shape would only be able to have a limited number of design parameters. It also opened the possibility for structural design parameters to be included in the optimisation. Critically, this enables the simultaneous optimisation of an aircraft's aerodynamic shape and its structural weight. Grossman *et al*. [40] showed that an integrated aerostructural optimisation gave higher performance designs than those found by sequential optimisations. This highlights the importance of the coupled-adjoint method for outputting optimal designs. The reason sequential optimisations fail to produce optimal results is that the optimiser does not have access to all the necessary information. In 1933, Prandtl demonstrated that when structural constraints were considered, the optimal lift distribution on a wing to minimise the drag was no longer elliptical [41]. A strictly aerodynamic optimisation does not account for the structural benefit of shifting the lift distribution inwards. On the other hand, a strictly structural optimisation does not manipulate the structure in a way that causes it to deflect to an aerodynamically favourable shape in flight.

Early aerostructural optimisations often used low fidelity aerodynamic models such as the lifting-line method. Haftka [42] produced one of the earliest aerostructural optimisations by combining the lifting line method with a finite-element analysis for the structure. He used the optimisation to quantify the trade-offs between weight and induced drag. Using high-fidelity solvers for aerodynamics is particularly important as they are capable of capturing important flow features that occur due to viscous and compressibility effects. In the field of aerodynamic and aeroelastic shape optimisations, the CFD solver is considered to be high-fidelity when the Reynolds-averaged Navier-Stokes (RANS) equations are used to model the flow physics. Ghazlane *et al.* [43] performed an aerostructural optimisation of the Airbus aircraft XRF1 using the coupled-adjoint method. They used a high-fidelity CFD solver but represented the structural model with a beam stick model instead of a full FEM. Abu-Zurayk [44] performed a coupled-adjoint aeroelastic optimisation of the XRF1 using both a high-fidelity CFD solver and a full FEM to represent the structural model. More recently, Abu-Zurayk *et al.* [45] were able to perform coupled-adjoint aeroelastic optimisations on the XRF1, that included the modelling of a powered engine, at a multitude of industrially relevant load cases. They found that engaging powered engines during the optimisation was essential as significantly different optimised geometries are found when the engines are powered. However, in order to find the gradient of the thrust with respect to the design variables, finite differences were required. A few other authors have performed aerostructural optimisations based on the coupled-adjoint approach with high fidelity solvers. Maute *et al*. [46], Zhang *et al*. [47], [48], Lei *et al*. [49] and Achard *et al*. [50] have all successfully performed similar coupled-adjoint optimisations.

Professor Martins at the University of Michigan and Professor Kennedy from the Georgia Institute of Technology have undertaken numerous aerostructural optimisations. Kenway *et al*. [51] created a jig shape variation of the common research model (CRM) and optimised the planform and airfoil shapes via the coupled-adjoint approach. Kenway and Martins [52] performed another aerostructural optimisation on the CRM to solve two optimisation problems. The mesh consisted of 2 million cells and the Euler governing equations were used. One optimisation sought to minimise the take-off gross weight and the other to minimise fuel burn. Later work by Martins and the University of Michigan research team has seen them target the coupled-adjoint approach to many different scenarios

involving various trade-offs. Brelje and Martins [53] performed an aerostructural optimisation with the interesting constraint of storing hydrogen fuel tanks, which adds complex geometry into the challenge. Gray and Martins [54] performed an aerostructural optimisation but did not assume the structure of the wing could be modelled by a linear elastic analysis as all others have. The use of a linear elastic model has a significant benefit when computing the adjoint because the stiffness matrix is symmetric. The symmetric nature of the stiffness matrix means the same structural solver can be used for the structural analysis and the structural adjoint. They note that as aspect ratios get higher, the structural behaviour becomes more non-linear making a non-linear structural solver more important. However, the inclusion of the non-linearity in the structure increased the computational cost of the structural solution by a factor of 20. It is clear that aerostructural optimisations based on the coupled-adjoint approach are gaining in maturity and are being utilised to investigate various new and exciting applications.

All optimisations that contain both aerodynamic shape parameters and structural parameters face a significant challenge. The adjoint technique allows a gradient to be determined independently from the number of design parameters. However, it is not independent from the number of constraints. Each constraint requires the calculation of a new gradient. The direct analytic approach enables a gradient to be found independent from the number of constraints. However, it is not independent from the number of design variables. A drag minimisation optimisation that concurrently optimises aerodynamic shape parameters and structural size parameters contains both a large number of design variables and a large number of constraints. The design variables are required to provide a large enough design space for manipulating the aerodynamic shape while the constraints occur because the changing structure must not exceed the maximum stresses within any element. Unfortunately, an algorithm that efficiently handles an optimisation case involving large numbers of design variables and constraints does not exist. The approach taken by Martins [55] and Ghazlane *et al*. [43] is to aggregate the constraints into a single or a small number of aggregation functions. The aggregation function that is most commonly used in aerostructural optimisations [58 - 59] is the Kreisselmeier-Steinhauser (KS) function. Kennedy and Hicken [58] found that the most common constraint-aggregation functions were mesh dependent. As the mesh spacing decreases, the aggregation function diverges. They also noted that there were no established methods to assess the impact of constraint aggregation on the final optimised design.

Aggregation functions do not have to be used if structural parameters are not in use as it can be presumed that the baseline stiffness matrix will satisfy the structural constraints. In this case, the gradient of the cost function is dependent on four factors: the flow state variables, the structural state variables, the mesh, and the shape of the aircraft. The flow state variables and structural state variables are handled by the flow adjoint and structural adjoint. To handle the mesh sensitivity, finite differences can be used. Alternatively, Nielsen and Park [59] proposed a mesh adjoint approach to eliminate the need for using finite differences to obtain the grid sensitivity. One benefit of the finite differences approach to the mesh sensitivity is that the dependency of the shape on the cost function does not have to be considered, as the only way the shape affects the cost function is through the mesh. However, finite differences have the problem of requiring $n_D + 1$ calculations and there is uncertainty over the appropriate step size to use when performing the finite differences. For large unstructured meshes with many design variables, obtaining the gradient through finite differences is expensive. The mesh adjoint enables the grid sensitivity calculation to be circumvented but the sensitivity of the surface to the design parameters must be calculated. Obtaining the gradient via finite differences for the surface is much less computationally expensive, but often it can be found analytically making the process even faster.

The speed of the mesh adjoint process is dependent on the mesh deformation algorithm that is used. Mesh deformation algorithms can be separated into two categories, explicit (non-iterative) and implicit (iterative) methods. The methods can be further categorised depending on how they work, do they work directly on a point cloud or do they need connectivity information between the points. Deformation methods that require connectivity information include Linear Elasticity analogy [60], elliptic smoothing [61] and quaternion [62] methods. The requirement of connectivity becomes challenging for unstructured grids due to the variation in the number of edges that the cells in the mesh have. Linear Elasticity analogy has successfully been used for mesh deformation in unstructured grids in a number of optimisations however [63], [64]. Point based methods are easier to implement on unstructured grids. Mavriplis and Yang [65] noted that point based methods are easily parallelised as care does not have to be taken regarding the index order, pre-conditioning and multigrid strategy unlike in connectivity based methods. Point based methods include radial basis function (RBF) mesh deformation [66], Delaunay Graph Mapping (DGM) [67] and inverse distance weighting (IDW) [68].

Iterative methods require the solving of a large linear system to find the relationship between the surface nodes and the volume nodes. Linear Elasticity analogy and the original RBF mesh deformation method fall under the category of iterative methods. They are iterative as they have the form $\boldsymbol{Kx} = \boldsymbol{f}$ where $\boldsymbol{x}$ is the unknown vector of volume displacements that is being sought and $\boldsymbol{K}$ is too large to invert. The explicit form unsurprisingly takes the form $\boldsymbol{Kf} = \boldsymbol{x}$, meaning that the volume displacements can be found via a matrix vector product. Typically, the matrix $\boldsymbol{K}$ is too large to store explicitly in memory and must be calculated on the fly. This is the case whenever the matrix is dense as is the case for RBF and IDW. For DGM however, the matrix is especially sparse, requiring only 4 elements per row to be non-zero. The ability to store the DGM matrix explicitly in memory combined with it being an explicit method enables it to perform a mesh deformation in a very efficient manner.

The trade-off that has to be made when deciding on a mesh deformation algorithm is between the robustness of the algorithm and the speed of the algorithm. DGM for instance is a fast algorithm, but it tends to struggle with mesh quality when the shape being deformed has changes in curvature between concave and convex. Wang *et al*. [69] combined the DGM method with RBF to improve the mesh quality near the surface. This enabled deformations such as rotations to be handled effectively but it was only applied to structured meshes. Mura *et al.* [70] successfully used the DGM method on unstructured meshes to optimise the ONERA M6 wing. Robustness was added to the DGM by defining a box far within the farfield boundary nodes to reduce the skewness of the Delaunay elements. This technique was successful, but it adds an implementation cost as the internal box must be defined for each mesh it is applied to. Rendall and Allen [71] developed a way to make the robust RBF mesh deformation technique more efficient through the use of data reduction algorithms. The data-reduction algorithm allows the RBF method to be used as an explicit mesh deformation method. This allows a deformation of the mesh to be performed efficiently and for the corresponding mesh adjoint to be calculated efficiently. Rendall and Allen proceeded to improve on their data reduction strategy [72]–[74] making it fully independent of the volume mesh, structural mesh and flow solver type. RBF with a data reduction strategy is a popular choice for many researchers and in industry due to its mesh quality preservation [75] and speed. The main issue with using a data reduction strategy is the loss of geometric integrity. A significant finding of their work is that the required size of the reduced data set to satisfactorily preserve the geometric integrity is almost independent of the mesh size. This means that RBF can be used as an explicit mesh deformation method for a mesh of any size. This is because the matrix to be inverted is the size of number of chosen surface points squared. The surface points chosen to define the displacement field are referred to as the base points of the deformation. The surface points that are not included in the primary deformation will not be exactly where the shape parameters intend them to be. Similarly, in the case of an FSI simulation, the surface points that are

not included as the base points will not be exactly where the structural displacements intend them to be.

The main focus when deforming a mesh is to preserve the quality of the mesh as the underlying geometry changes. This is not an important consideration when calculating the sensitivity of the mesh to changes in the underlying geometry. Mura *et al*. [76] showed that a different mesh deformation algorithm can be used in the mesh sensitivity calculation to the algorithm that was used for deforming the mesh without a meaningful loss in gradient accuracy. This is particularly useful if the algorithm used for mesh deformation is expensive. They used LE and RBF (without the data reduction algorithm) for the mesh deformation in their optimisations. They then compared the gradients when using a consistent algorithm for the mesh sensitivity against the gradient that was output when DGM was used for the mesh sensitivity. They concluded that a non-consistent approach could be used whenever deemed necessary.

2.2 Identification of Research Gaps in the Coupled-Adjoint Literature

An area of much interest to industry and academic researchers is the speed of convergence of the coupled-adjoint equations. Research has been undertaken studying the effects on convergence that different iteration methodologies. Kenway *et al.* [77] compared the efficiency of a block Gauss-Seidel approach against a Newton-Krylov approach and found the Newton-Krylov approach to be faster. Similarly, Y. Shi *et al.* [78] compared the gradient computation time of the coupled-adjoint using the block Gauss-Seidel approach and the Newton-Krylov approach. They found again that the Newton-Krylov approach was faster. However, there have been no studies detailing the effects of different interpolation methods on the gradient computation time. In particular, the effect of the chosen mesh deformation algorithm on the convergence of the coupled-adjoint has not been examined.

The effects of using a non-consistent algorithm in an adjoint-based shape optimisation for the mesh deformation and mesh sensitivity calculations has been explored by Mura *et al.* [76]. They found that the effect on the accuracy of the produced gradient was negligible. However, this was only studied in the case of a rigid aerodynamic optimisation. The effect on gradient accuracy of using a different mesh deformation algorithm in the coupled-adjoint to what was used in the FSI loop has not been examined. The accompanying effects on the efficiency of the coupled-adjoint of this approach has not been explored either.

Another important area for consideration is how to handle trim constraints within the optimisation. Trim is often handled by using a trim penalty or a fixed wing moment constraint. An alternative approach is to include a design parameter which is primarily used for trimming the aircraft, this will usually be the deflection of the horizontal tail. This approach is referred to as the direct optimisation strategy (DOS). Chen *et al*. [79] recommended using the DOS strategy for achieving trim as it found a better optimum when compared against a wing pitching moment constraint approach. However, they did not compare the DOS approach against an approach which trimmed the aircraft within the simulation. Merle *et al.* [64] was the first to examine the effect on optimisation efficiency of trimming an aircraft within the flow simulation. An approach they refer to as the trim corrected optimisation strategy (TCOS). They found that the optima produced by the TCOS and DOS were almost identical, although a 50% increase in wall clock time was required for each design candidate when the TCOS approach was used. However, they also found that the TCOS approach reduced the drag more

substantially in the first few design iterations relative to the DOS approach with the additional benefit that each design candidate was a feasible design. Very few studies have explored the approach of using a trim parameter within the flow simulation to trim the aircraft. The only works the author is aware of that employ a trim loop within an FSI simulation were performed by Abu-Zurayk *et al.* [45] and Merle *et al.* [80]. However, they did not compare the efficiency of the optimisation process of the TCOS approach was not compared against a DOS approach.

3. Aeroelastic Optimisation Methodology

3.1 Fluid-Structure Interaction

The design of an aircraft's external shape is not just an aerodynamic problem. The shape of a wing affects its aerodynamic loading. Different aerodynamic loading profiles on a wing will displace the wing into different shapes. One way of dealing with aeroelastic behaviour is to design the jig shape so it deforms into its intended design shape in cruise flight [81], [82]. An advantage of this approach is the decoupling of the aeroelastic system into one aerodynamic problem and one structural problem. This decreases the simulation time significantly, but it is not guaranteed to produce the optimum shape or even produce a feasible jig shape [83, p. 9], [84]. Additionally, this approach does not work for multi-point designs which are essential in ensuring acceptable off-design performance. A multi-point optimisation of a rigid wing outputs a flight shape wing that would perform well at the targeted flight conditions. However, the jig shape needed to produce this flight shape wing would be different at all the targeted flight conditions. As wings become less stiff to reduce the aircraft's weight, the aeroelastic effects are more pronounced. It is therefore important to accurately model the aeroelasticity so that the true optimum shape can be found. The way to achieve this is to take the coupled effects of structural mechanics and aerodynamics into account from the beginning of the design.

There are two approaches to solving a coupled aerodynamic and structural problem, either a monolithic or a partitioned approach [85]. The monolithic approach to FSI problems is one which solves the aerodynamic and structural equations at the fluid-structure interface synchronously. The discretised fluid and structural models are enclosed in one system. This approach avoids any time lag between structural movement and fluid flow as the whole system is solved implicitly. Blom [86] found that solving the fluid-structure interaction implicitly in this way can be advantageous when applied to dynamic FSI problems.

The partitioned approach to FSI problems is one where the aerodynamic and structural equations are solved successively. The information from one discipline is exchanged with the other after it converges. The exchanged information forms part of the starting conditions for the other discipline. In the static aeroelastic case, this iterative process is performed several times until the exchanged information becomes constant i.e. aeroelastic equilibrium is reached.

Michler *et al*. [87] examined the advantages and disadvantages of a monolithic approach. They found that a key advantage of the monolithic approach is an increase in the accuracy of the result while affording larger time-steps. On the other hand, they found the computational cost of the monolithic approach is considerably more expensive than the partitioned approach. There is a trade-off to be had then between computational time and accuracy. High-fidelity FSI simulations are expensive even with the partitioned approach. In an optimisation where many simulations are run, this extra time adds up considerably. The objective of this research project is to produce an optimisation tool that can directly optimise an aircraft's shape while considering aerodynamic and structural effects as efficiently as possible. The partitioned approach to solving FSI problems is the approach chosen in this research due to its lower computational expense relative to the monolithic approach. Another advantage of using the portioned approach is that tried and tested discipline specific adjoint codes can be employed. The partitioned approach to FSI problems formulated by DLR in FlowSimulator [88] has been used as the basis for this project to be built from.

3.1.1 The Navier-Stokes Equations

Computational fluid dynamics (CFD) is an integral part of aerodynamic design. While experimental testing in flight tunnels will always remain part of the design process, higher fidelity CFD simulations mean that less time needs to be spent in wind tunnel testing. This is because modern day CFD can accurately predict how air will behave as an aircraft travels through it. However, even with the advances made in the field of CFD, there is still a limit on the level of fidelity that can be captured by CFD. Specifically, approximations must be made about the behaviour of turbulence in order to make the simulations of fluid flow around an aircraft be computationally feasible.

The governing equations of fluid dynamics, the Navier-Stokes equations can be written in either an integral or differential form. In numerical solutions, the integral form is solved using the finite volume method while the differential form is solved using the finite difference method. The finite volume approach is conservative unlike the finite difference approach, and it therefore enables discontinuous solutions. This is important when modelling shocks. Shocks arise in the flow as they are the mechanism by which the air in the flow is recompressed after it has exceeded the sonic speed. It is important for the simulation to be able to model the effects of shocks in the flow as commercial aircraft fly in the transonic region. The transonic region is defined by having flow moving between subsonic and supersonic speeds. Commercial aircraft fly in this region as it is the most fuel-efficient flight regime. A metric used to measure the cruising efficiency is the Mach number times the lift over drag ratio. This stays roughly constant up to a certain Mach number which will be in the transonic regime [89].

Shocks occur on the upper surface of the wings on a commercial aircraft. They interact directly with the boundary layer of the flow. The boundary layer is the region of flow where viscous effects cannot be ignored. The viscous effects determine the skin friction drag, heat transfer and wing stall behaviour. As shocks are interacting directly with the boundary layer, it is important that viscous and turbulent effects are modelled accurately for an effective shape optimisation. For these reasons, the Navier-Stokes equations rather than the Euler equations are deemed necessary to model the flow physics.

The integral form of the Navier-Stokes equations can be derived by considering the conservation of mass, momentum, and energy through an arbitrary control volume Ω. For these quantities to be conserved, the rate of change or each quantity must equal the flux of each quantity through the control volume.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{3.1}$$

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \tau + \boldsymbol{S}_M \tag{3.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} H) = \nabla \cdot (\tau \cdot \boldsymbol{u}) + \nabla \cdot (\kappa \nabla T) + \boldsymbol{S}_E \tag{3.3}$$

In a single integral form, these conservation equations are written as:

$$\frac{\partial}{\partial t} \int_{\Omega} \boldsymbol{W} d\Omega + \oint_{\partial \Omega} \{\mathbf{F}^c(\boldsymbol{W}) - \mathbf{F}^v(\boldsymbol{W})\} \cdot \boldsymbol{n} d(\partial \Omega) + \int_{\Omega} S(\boldsymbol{W}) d\Omega = 0 \tag{3.4}$$

The boundary of the control volume Ω is denoted by $\partial\Omega$ with an outer normal direction of $\boldsymbol{n}$. The source term S in the third integral in Eq. (3.4) is any non-conservative source term. It refers to quantities that are being added to the flow field rather than quantities passing through the boundary. This could be active flow control devices on the wing or exhausts from the engine. In steady flow cases, the time derivative term becomes zero. For the cases presented in this work, the flow conditions will

always be a steady case and there will be no source terms. This leaves only the second integral term from Eq. (3.4) which is now defined by the residual vector $\boldsymbol{R}$.

$$\boldsymbol{R}(\boldsymbol{W}, \boldsymbol{n}, \partial\Omega, \Omega) = 0 \tag{3.5}$$

$\boldsymbol{W}$ represents the flow state variables, these are the quantities that are conserved in the flow. Appendix B describes all the terms associated with the Navier-Stokes and introduces the turbulence closure problem. The closure problem is handled with the use of the Reynolds Averaged Navier-Stokes equations and the Spalart-Allmaras turbulence model. A detailed derivation of this numerical model of airflow is provided in Appendix B.

The residual vector $\boldsymbol{R}$ is a function of the flow state variables and the shape of the control volume. This project is interested in simulating airflow around an aircraft hence the shape of the control volume is a function of the shape of the aircraft, which is defined by the design parameters $\boldsymbol{D}$. The aircraft's shape affects the state variables directly too. This is because air cannot pass through the aircraft surface in the normal direction, while in the tangential direction the air is sheared by the aircraft's surface due to the no-slip condition.

$$\boldsymbol{R}\left(\boldsymbol{W}\big(n(\boldsymbol{D}), \partial\Omega(\boldsymbol{D}), \Omega(\boldsymbol{D})\big), \boldsymbol{n}(\boldsymbol{D}), \partial\Omega(\boldsymbol{D}), \Omega(\boldsymbol{D})\right) = \boldsymbol{0} \tag{3.6}$$

The terms $\boldsymbol{n}$, $\partial\Omega$ and Ω only the represent the shape of the control volume. Hence the only variable they are dependent on is the design parameter vector.

$$\boldsymbol{R}(\boldsymbol{W}(\boldsymbol{D}), \boldsymbol{D}) = \boldsymbol{0} \tag{3.7}$$

For RANS simulations, it is necessary to spatially discretise the control volume and solve a finite volume problem. The mesh vector $\boldsymbol{X}$ now defines the control volume so the residual is implicitly dependent on the design variables through $\boldsymbol{X}$.

$$\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{D}) \tag{3.8}$$

$$\boldsymbol{R} = \boldsymbol{R}\big(\boldsymbol{W}(\boldsymbol{D}), \boldsymbol{X}(\boldsymbol{D})\big) \tag{3.9}$$

The aerodynamic governing equations are now related to the shape parameterisation of the aircraft. This relationship makes it possible for an aerodynamic quantity of the aircraft to be optimised with respect to the aircraft's shape. The next section will relate the design parameters to the structural governing equations, thus forming a basis for performing an aeroelastic shape optimisation.

3.1.2 Linear Elastic Analysis

To perform a partitioned aeroelastic simulation, a structural solver is needed. For the purposes of wing deformation in flight, a linear elastic solver is considered sufficient. This is because thin shell structures like wings typically display linear elastic behaviour when the applied aerodynamic forces do not take the structure past its elastic limit [90]. A benefit of using a linear elastic structural model is that the structural mesh does not need to be deformed for a singular FSI simulation. A force can be applied to an elastic structure in an undeformed state or a displaced state and both will displace to the same shape. This is because the displaced state of an elastic structure has no effect on how much a new input force will displace it. This assumes that the previous force that held the structure in its displaced state is replaced by the new one. This is particularly useful for FSI simulations as the aerodynamic forces from a deformed aerodynamic mesh can be directly transferred to the structural mesh without requiring any modifications due to the previously applied displacements.

The governing equation for the structural model derives from Newton's laws of motion which give rise to the Lamé equation. The Lamé equation is the underlying PDE which describes how the structure will respond to an applied load. The Linear Elasticity equations and the subsequent discretisation are derived in Appendix C.

The direct stiffness method was developed by Turner [91] whereby a stiffness matrix for a composite structure can be generated directly by the addition of stiffness matrices that correspond to individual elements of the structure. Each element's stiffness matrix is defined by its physical properties such as thickness.

$$\boldsymbol{F}^e = \boldsymbol{K}^e \boldsymbol{u}^e \tag{3.10}$$

$$\boldsymbol{K} = \sum_{e=1}^{n_e} (L_e)^T \boldsymbol{K}^e L_e \tag{3.11}$$

$$\boldsymbol{F} = \boldsymbol{K}\boldsymbol{u} \tag{3.12}$$

The variables L_e in Eq. (3.11) are Boolean localisation matrices. This means the relationship between the structural forces and the displacements is a linear system. The forces applied on the boundary of the structure are interpolated from the aerodynamic forces calculated by CFD. The remaining nodal forces not on the boundary are found by satisfying that all elements and nodes are in static equilibrium. To obtain the structural displacements for an input vector of forces, a linear system has to be solved as the stiffness matrix is too large to be directly inverted. The displacement values are considered found when the structural residual is sufficiently small, just like the CFD case.

$$\boldsymbol{R}_s = \boldsymbol{F} - \boldsymbol{K}\boldsymbol{u} \tag{3.13}$$

One small inaccuracy that is introduced during an optimisation is that the FEM mesh does not change regardless of the design parameter settings. It would be possible to update the stiffnesses with each design iteration, but the errors are small for the correspondingly small changes in the external shape.

As there are no structural parameters considered in this project, the stiffness matrix is considered constant for the whole optimisation despite a small dependence of the stiffness matrix on the shape of the elements. This of course does introduce error into the optimisation. However, so long as the jig shape changes remain small then the error introduced via a slightly inaccurate stiffness matrix will also be small. If the jig shape were to change significantly, it would be prudent to restart the optimisation with an updated stiffness matrix. Due to simplification of considering the stiffness matrix as constant throughout the optimisation, only the structural forces and displacement vectors are functions of the design parameters. The structural force vector is analytically related to the aerodynamic force vector through the interpolation. The relationship of the structural displacement vector to the design parameters cannot be found analytically. To consider this dependency within the optimisation, the coupled-adjoint method is used which is described in chapter 3.3.

3.1.3 Interpolation between the Aerodynamic and Structural Solvers

The surface meshes used for CFD and CSM will almost certainly have different fidelities as each discipline has different mesh requirements. Therefore, a method which can accurately interpolate information from the nodes on one mesh to the nodes of the other is required. Already at this point, the calculation of the gradient should be considered. It is possible to implement 2 separate interpolation methods, one for the transfer of information from CFD to CSM and a different interpolation method going in the other direction. However, if the interpolation in one direction is the

transpose of the interpolation in the other direction, then there is no extra development cost in the adjoint when transposing the interpolation processes.

In order to use the same technique for both interpolations, an interpolation method which is appropriate for both directions is needed. Interpolating the forces from CFD to CSM is simpler as the CFD mesh will typically be of a much higher fidelity than the CSM mesh. Even a simple procedure like nearest neighbour is acceptable as all that needs to be done is to create an equivalent system on the lower fidelity mesh. It is easy to create an equivalent system when going from high fidelity to low fidelity, but in the other direction it cannot be done as simply. Results from literature show that the best way to interpolate information from a sparser mesh to denser mesh is via radial basis function interpolation schemes [66], [92]–[94].

The starting point of the interpolation is a known set of displacements on the surface of the CSM mesh $\boldsymbol{u}_s$.

$$u_i(\boldsymbol{X}_i) = p(\boldsymbol{X}_i) + \sum_{j=1}^{n_s} a_j \phi\left(\left\|\boldsymbol{X}_i - \boldsymbol{X}_j\right\|\right) \tag{3.14}$$

$$p_x(\boldsymbol{X}_i) = \beta_{x,1} + \beta_{x,2} x_i + \beta_{x,3} y_i + \beta_{x,4} z_i \tag{3.15}$$

$$p_x(\boldsymbol{X}_i) = \beta_{y,1} + \beta_{y,2} x_i + \beta_{y,3} y_i + \beta_{y,4} z_i \tag{3.16}$$

$$p_x(\boldsymbol{X}_i) = \beta_{z,1} + \beta_{z,2} x_i + \beta_{z,3} y_i + \beta_{z,4} z_i \tag{3.17}$$

The polynomial p is added to ensure the linear system is uniquely solvable and to improve the accuracy of the RBF interpolation [95]. The RBF interpolant function is the thin plate spline method. The thin-plate spline basis function is:

$$\phi(r) = r^2 \log(r) \tag{3.18}$$

The coefficients a and β are obtained by ensuring the coefficients give the exact structural surface displacements when they are substituted in.

$$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ u_{\rho,1} \\ u_{\rho,2} \\ u_{\rho,3} \\ \vdots \\ u_{\rho,n_S} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & x_1 & x_2 & x_3 & \cdots & x_{n_S} \\ 0 & 0 & 0 & 0 & y_1 & y_2 & y_3 & \cdots & y_{n_S} \\ 0 & 0 & 0 & 0 & z_1 & z_2 & z_3 & \cdots & z_{n_S} \\ 1 & x_1 & y_1 & z_1 & 0 & \phi_{1,2} & \phi_{1,3} & \cdots & \phi_{1,n_S} \\ 1 & x_2 & y_2 & z_2 & \phi_{2,1} & 0 & \phi_{2,3} & \cdots & \phi_{2,n_S} \\ 1 & x_3 & y_3 & z_3 & \phi_{3,1} & \phi_{3,2} & 0 & \ddots & \phi_{3,n_S} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & x_{n_S} & y_{n_S} & z_{n_S} & \phi_{n_S,1} & \phi_{n_S,2} & \phi_{n_S,3} & \cdots & 0 \end{bmatrix} \begin{Bmatrix} \beta_{\rho,1} \\ \beta_{\rho,2} \\ \beta_{\rho,3} \\ \beta_{\rho,4} \\ a_{\rho,1} \\ a_{\rho,2} \\ a_{\rho,3} \\ \vdots \\ a_{\rho,n_S} \end{Bmatrix} \rho = x, y, z \tag{3.19}$$

If there are only a small number of surface points on the FEM, then this process would be perfectly adequate. However, once the number of surface points gets much higher than a few thousand then the inversion of the matrix becomes computationally expensive to perform. One of the main objectives of this project is to make optimisation as efficient as possible. To maintain the benefits of the RBF interpolation while keeping the speed of the interpolation fast, the approach proposed by Stickan *et al*. [96] is implemented. This approach is described in the section below.

Coupling Surface Domain Blending

To aid in the speed of the interpolation between the aerodynamic and structural solvers, the coupling surface is divided into domains. The domains are the same on both the CFD surface and the CSM surface. Each domain has its own spline technique which is used to interpolate displacements and forces between the two meshes. The user defines where the split domains on the aircraft should be, domains will typically correspond to physical features of the wing such as flaps and ailerons. After the domains have been defined, the FEM surface nodes are mapped with the CFD surface nodes which lie in the same domain. This is done prior to the optimisation, and it is only done once. This coupling approach allows for different components of the aircraft to use different interpolation methods. It also allows domains to be further subdivided to improve the numerical performance of the interpolation methods. A global interpolation matrix $\boldsymbol{G}$ is generated by combining the interpolations of each individual domain.

$$\boldsymbol{u}_{surf} = \boldsymbol{G}\boldsymbol{u}_S \tag{3.20}$$

$$\boldsymbol{F}_S = \boldsymbol{G}^T\boldsymbol{F}_{surf} \tag{3.21}$$

The global interpolation matrix is formed from a product of relaxation and blending matrices $\boldsymbol{M}_{relax}$ $\boldsymbol{M}_{blend}$, which are used to regulate the boundaries between the different domains, and a matrix consisting of the different splines $\boldsymbol{M}_{splines}$.

$$\boldsymbol{G} = \boldsymbol{M}_{relax} \cdot \boldsymbol{M}_{blend} \cdot \boldsymbol{M}_{splines} \tag{3.22}$$

$$\boldsymbol{M}_{splines} = \begin{bmatrix} \boldsymbol{M}_{spline_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{M}_{spline_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{M}_{spline_{nSplines}} \end{bmatrix} \tag{3.23}$$

Blending is managed by overlapping the domains and then averaging the displacements in the overlapped regions. Figure 3 shows the interpolation domains on the CFD coupling surface (left) and the CSM coupling surface (right). There are three domains defined by green, blue, and pink cells. To ensure a smooth displacement field is achieved by the interpolation, a region of blending between the adjacent domains is introduced. Here the domains overlap, the displacements in this region are calculated by the splines of both overlapping domains. As a result, the matrix $\boldsymbol{M}_{splines}$ is not block-diagonal. The matrix actually has the dimensions $3\left(n_{surf} + n_{surf,overlap}\right) \times 6n_S$. To ensure a unique displacement value for each CFD node after interpolation, the $\boldsymbol{M}_{blend}$ matrix averages the values calculated at the nodes in the overlapping domains. The relaxation matrix serves a similar purpose, it modifies the displacements at intersection of components on the CFD mesh such as the wing-fuselage interface.

The interpolation methods available for use in FlowSimulator [96] by individual spline domains are:

- Radial basis functions: When interpolating from a CSM surface to a CFD surface.
- Beam spline: When the structural component is represented by a line of nodes.
- Rigid body spline: When the structural component is represented by a single node.

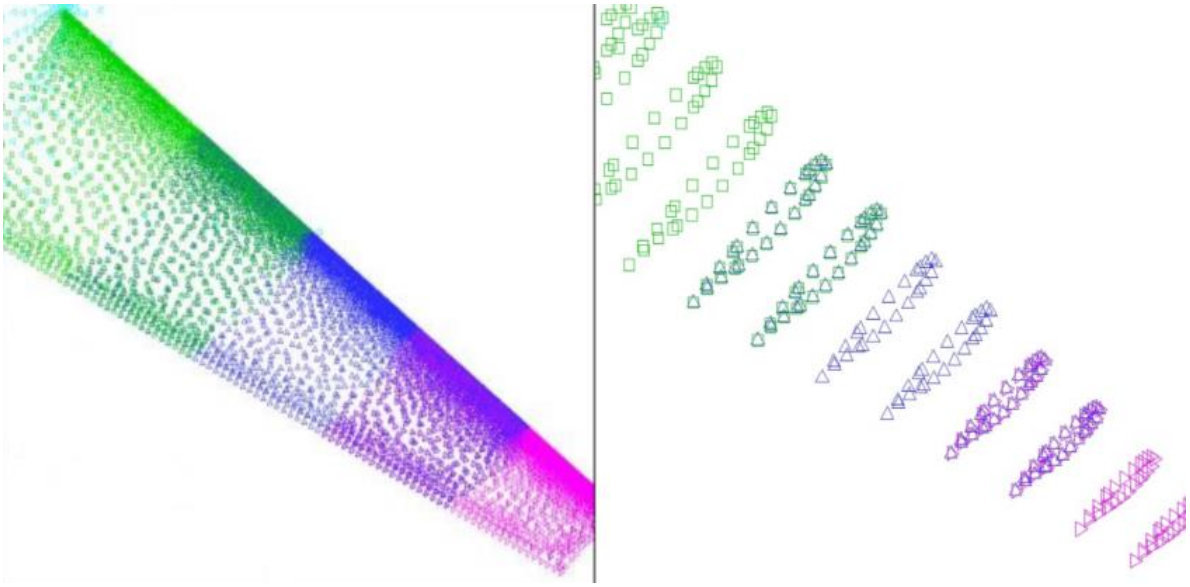

Fig. 3 Interpolation domains on the CFD and CSM coupling surface meshes.

An added benefit of this procedure is available when simulations include control surface rotations. When control surfaces are considered, the displacement profile of the control surface should not be blended with the rest of the wing. To achieve this, a spline domain pertaining to the control surface alone is created. To demonstrate this ability, a flap was generated at 35% of span to 75% of span.

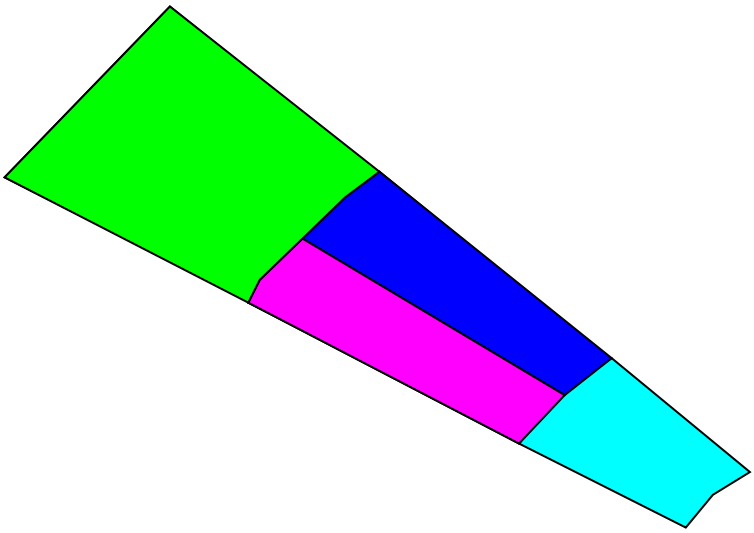

Fig. 4 The interpolation domain for the flap on the coupling surface of the LANN wing.

Heinrich [88] demonstrated that the use of individual splines for interpolating control surface deflections produces better agreement between the expected displacement profiles when compared with an interpolation that used only a single global spline. In Fig. 5, the author of this thesis validated this interpolation approach for the LANN wing test case. The expected displacement profile created on the CSM mesh after a rotation of the flap was compared against the displacements reported on

the CFD mesh after interpolation. It shows good agreement between the displacement profiles is achieved when using a separate interpolation spline without blending for the flap.

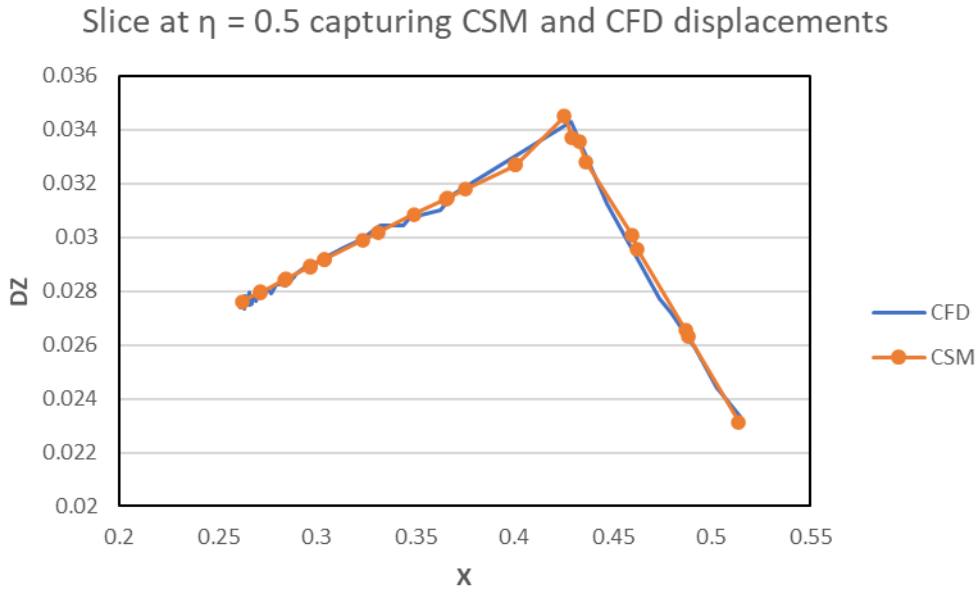

Fig. 5 Comparison of interpolated displacements on CFD surface against actual CSM displacements on the LANN wing.

This blending procedure between a control surface and the rest of the wing can also be used to approximate the deflection of a morphing wing actuator. A morphing wing surface can be approximated by blending the displacement profile created by the control surface's interpolation spline with the adjacent domains. A variable camber wing is a technology that is able to deform its shape whilst maintaining a smooth aerodynamic profile. Currently, this technology has not been applied to commercial aircraft but it is being actively researched by academia and industry [97], [98]. In this project, the elevator on the horizontal tail of the XRF1 has been modelled as having morphing wing characteristics, meaning it can be deflected while maintaining a smooth surface. This modelling of the variable camber horizontal tail is done by creating an interpolation domain for the elevator on the horizontal tail, but then adding a blending region with the rest of the horizontal tail.

An advantage of the multi-domain approach to interpolation taken by this project is the ability to use different interpolation techniques during the force and displacement transfers. This is useful when a full FEM representation is not available. High fidelity models are not always available in early phases of design, meaning the only available structural models are often basic models such as beam models [99]. Another important use case for using different interpolation techniques is when the designer would like to model a control surface deflection but does not yet have the accompanying structural model of the control surface. The control surface can simply be modelled as a beam of mass nodes in the FEM and a beam interpolation method will be able to produce control surface deflections in the simulation. The ability to use a different interpolation method for each domain offers a large degree of flexibility to the designer.

3.1.4 RBF Mesh Deformation

The next stage in an aeroelastic simulation after interpolating the structural displacements to the aerodynamic surface is to deform the rest of the CFD volume mesh. The same procedure is also performed at the start of each optimisation iteration. At the start of the optimisation iteration, the displacements on the surrogate CAD model are used to deform the CFD mesh. Within the FSI simulation, the interpolated displacements on the CFD surface are used to deform the mesh.

It is possible to deform the CFD volume mesh directly from the structural surface using an RBF mesh deformation method. So the question arises, why have an interpolation process and then a mesh deformation process? The reason is that the thin plane spline variant of RBFs is good at interpolating quantities that lie on a surface [100]. It also enables the option of using various interpolation methods between the FEM mesh and CFD mesh such as a beam interpolation. Regardless of what CSM to CFD interpolation method is used, the volume mesh deformation method can always be applied in the same way.

The mesh deformation strategy applied in this project is a specific variant of RBF interpolation. A scaling factor b is introduced to enhance the quality of the output mesh. The scaling factor is dependent only on the distance of the volume node from the surface, this is used to linearly suppress the deformation between R_F and R_Z.

$$b_i = \begin{cases} 0 & for\ d_i \geq R_Z \\ 1 & for\ d_i \leq R_F \\ \dfrac{R_z - d_i}{R_z - F_F} & for\ R_F < d_i < R_Z \end{cases} \tag{3.24}$$

$$\Delta \boldsymbol{X}_{\rho,i} = b_i \left(p(\boldsymbol{X}_i) + \sum_{j=1}^{n_S} \zeta_j \varphi\left(\left\|\boldsymbol{X}_i - \boldsymbol{S}_j\right\|\right) \right) \begin{matrix} i = 1 \dots n_V \\ \rho = x, y, z \end{matrix} \tag{3.25}$$

The interpolation coefficients are found directly through a matrix inversion.

$$\Delta \boldsymbol{X}_{surf_{\rho,i}} = p(\boldsymbol{X}_i) + \sum_{j=1}^{n_S} \zeta_j \varphi\left(\left\|\boldsymbol{X}_i - \boldsymbol{S}_j\right\|\right) \tag{3.26}$$

$$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Delta \boldsymbol{X}_{surf_{\rho,1}} \\ \Delta \boldsymbol{X}_{surf_{\rho,2}} \\ \Delta \boldsymbol{X}_{surf_{\rho,3}} \\ \vdots \\ \Delta \boldsymbol{X}_{surf_{\rho,1}} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & x_1 & x_2 & \cdots & x_{n_{surf}} \\ 0 & 0 & 0 & 0 & y_1 & y_2 & \cdots & y_{n_{surf}} \\ 0 & 0 & 0 & 0 & z_1 & z_2 & \cdots & z_{n_{surf}} \\ 1 & x_1 & y_1 & z_1 & 0 & \phi_{1,2} & \cdots & \phi_{1,n_{surf}} \\ 1 & x_2 & y_2 & z_2 & \phi_{2,1} & 0 & \ddots & \phi_{2,n_{surf}} \\ 1 & x_3 & y_3 & z_3 & \phi_{3,1} & \phi_{3,2} & \ddots & \phi_{3,n_{surf}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_{surf}} & y_{n_{surf}} & z_{n_{surf}} & \phi_{n_{surf},1} & \phi_{n_{surf},2} & \cdots & 0 \end{bmatrix} \begin{Bmatrix} \beta_{\rho,1} \\ \beta_{\rho,2} \\ \beta_{\rho,3} \\ \beta_{\rho,4} \\ a_{\rho,1} \\ a_{\rho,2} \\ a_{\rho,3} \\ \vdots \\ a_{\rho,n_{surf}} \end{Bmatrix} \tag{3.27}$$

In order for the matrix inversion to be computed without the need for iterative methods, the number of surface points that can be used to define the surface displacement profile are limited to 5000. While this doesn't capture the exact displacement profile, it still provides enough information to capture the

displacement profile to a sufficient level while producing a high quality deformed mesh in an efficient manner [101].

Similar to the splitting of domains in the interpolation between the CSM and CFD surface mesh, the aircraft surface mesh is split into different components. Each component has its displacement profile interpolated into the volume mesh separately. The final mesh deformation is the linear combination of all the RBF interpolations. The efficiency benefits of this method are clear, it is significantly faster to invert 5 matrices of a size 5000×5000 than it is to invert one matrix of size 25000×25000. This approach of using multiple RBF interpolations instead of a single global RBF interpolation could introduce a non-smooth displacement profile if the domain is split poorly. However, when the influence of the component's surface points have only a small effect on the mesh points surrounding another component it is a valid approach to use. For instance, the effect of the displacement profile around the horizontal tail would not cause the mesh points surrounding the wing to be disturbed by a large amount.

The speed of the mesh deformation can be further increased by approximating other components as fixed. In steady flight, it is reasonable to assume that the only components of the aircraft that will deflect significantly are the lifting surfaces. These are the wing and horizontal tail. The remaining components, such as the fuselage and vertical tail are therefore assumed to undergo no structural displacements. The aerodynamic forces will still be interpolated from the CFD surface to the structural mesh to ensure that when the linear elastic simulation is performed, the aircraft is in static equilibrium. The displacements that are interpolated back from the structural mesh to aerodynamic mesh at the fuselage are simply ignored and set to zero. The combination of using RBF with a data reduction algorithm, splitting the CFD surface mesh into components to interpolate them separately and only interpolating displacements from lifting surfaces reduces the computational cost of the mesh deformation greatly.

3.2 Parameterisation of the Aircraft

Early aerodynamic optimisation work was undertaken by Hicks, Murman and Vanderplaats [14 - 15] which showed encouraging signs that optimisation would be able to improve airfoil performance. Caughey *et al.* [23] investigated the effects that aerodynamic optimisations with large numbers of unconstrained design parameters had on aerodynamic performance. A critical effect they found was that an optimiser will manipulate flow at the smallest resolution available to it. A high-fidelity parameterisation scheme will modify the geometry at very small scales to significantly reduce the drag at the specified flow condition. The manipulation of geometry at such small scales has negative effects at off-design flow conditions. Multiple techniques such as multipoint and kriging-based optimisations [104] have emerged to ensure robust designs are output from optimisations. Another way to mitigate off-design losses is to limit the number of parameters and give those parameters a more global scope.

A typical shape optimisation is the drag minimisation of a three-dimensional wing belonging to a commercial aircraft. It inevitably takes a large number of design parameters in order to provide a sufficiently large design space that the optimiser can use to reduce the drag. Alongside the robustness challenge brought about by many design parameters, Chernukhin and Zingg [105] demonstrated that large numbers of design variables can lead to highly multimodal design spaces. A multimodal design space presents a problem for gradient-based optimisations as the optimiser will converge in a local minimum. In a general case, there is no guarantee that the local minimum is the global minimum. More recently however, Bons *et al.* [106] showed that multimodality is not a major limitation to

gradient-based optimisations as the application of practical design constraints can mitigate the effects of multimodality.

There are many approaches that can be taken to model the surface of an aircraft. One approach is to take all of the mesh points on the surface as the design parameters. This provides the largest design space and would allow an optimiser to return the largest performance gains. The issue with this approach is that there is no guarantee that the returned optimised shape is manufacturable [107]. Of most interest to this project is a parameterisation scheme that ensures manufacturability constraints are satisfied and one that returns a shape that is immediately useable by a designer. The motivation for this is to aid in producing a more integrated design process.

A parameterisation scheme that is increasing in popularity is one where the surface is defined by a series of modes, often obtained by proper orthogonal decomposition (POD). Allen *et al.* [108] used POD to derive a series of airfoil perturbation modes from a large training library of airfoils, enabling a large design space to be covered by a relatively small set of parameters. Another benefit of POD is that it can be applied to a design parameter set that has physical meaning to designers. This enables practical design constraints to be implemented easily. Bobrowski *et al.* [109] argue that an integrated approach to design is now essential to remove extensive trial and error iterations from the design process. For an integrated approach, high-fidelity models of the aircraft are required to be available at early stages in the design. To handle complex interference effects such as the engine installation and the integrated design process, they suggest using Computer Aided Design (CAD) parameters. Directly using CAD parameters present their own problems as the parameters are not typically differentiable and the CAD engine may not be available on a remote high-performance computing cluster. To address these problems, they created a surrogate model of a parameterised CAD surface using POD decomposition. The German Aerospace Centre (DLR) [21 - 22] ran a number of aeroelastic optimisations using the CAD-ROM approach demonstrating its suitability for high-fidelity optimisations of commercial aircraft. They found that CAD-ROM approach enabled complex interference effects to be effectively handled. They did find however that there is some discrepancy between the shape the optimised design parameters output for the CAD-ROM and the shape output by the CAD engine with the design parameters. For this reason, using POD decomposition as a parameterisation scheme is still an active area of research although this is not explored in this thesis.

3.2.1 Proper Orthogonal Decomposition Parameterisation

CAD engines are an important tool in geometry design. Their ability to handle complex geometries while satisfying prescribed geometric constraints makes them indispensable. For this reason, a parameterisation scheme that uses CAD parameters in the optimisation loop has been chosen. Using CAD parameterisation in an optimisation loop provides a number of challenges. The biggest challenge is that CAD systems are typically used on the designer's personal machine while the aerodynamic calculations are performed on remote high-performance computers. These HPCs often use different operating systems making it harder to use the same CAD software [110]. To maintain the benefits of CAD parameterisation while removing the dependence on calls to a CAD engine, a surrogate CAD model is employed as set out by K. Bobrowski *et al.* [109].

Bui-Thanh *et al.* [111] were the first to apply the POD method to aerodynamic shape optimisation. However, they demonstrated the method on 2D airfoils and they did not use CAD parameters as the basis for controlling the shape. In the approach proposed by Bobrowski, a number of CAD design variables are created on the designer's CAD model of the aircraft. A CAD model created by an application such as CATIA can be parameterised within the application. On the CAD surface, a

configurable number of curves can be defined which in turn can be discretised to define a coordinate field. Each setting of the CAD design parameters corresponds to a new set of surface coordinates $\mathbf{S}(\boldsymbol{D})$. The design space is sampled at many different settings according to Design of Experiments techniques [112] that are put together to form a snapshot matrix $\boldsymbol{Y}$.

$$\boldsymbol{Y} = [\mathbf{S}(\boldsymbol{D}^1), \cdots, \mathbf{S}(\boldsymbol{D}^{n_s})] \tag{3.28}$$

POD is a method which seeks to find interdependencies between the columns in a matrix. An interdependency in this context would be a basis vector of size n_p that captures how the coordinates move across the design space. POD produces a set of optimal basis vectors that, when used in some linear combination with each other, can best approximate the surface coordinate vector across the design space. To obtain the optimal set of basis vectors or modes ϕ, an optimisation which minimises the mean square error of an orthogonal projection of the snapshots is performed [113].

$$minimise: I(\phi) = \sum_{j=1}^{n_s} \left\| \mathbf{S}(\boldsymbol{D}^j) - \prod\nolimits_{\phi} \mathbf{S}(\boldsymbol{D}^j) \right\| \tag{3.29}$$

The operator $\prod_{\phi}$ refers to the orthogonal projection of some vector, in this case the coordinate vector, onto the subspace defined by the span of the basis vectors.

$$\boldsymbol{A} = \begin{bmatrix} \phi_1^1 & \cdots & \phi_1^{n_s} \\ \vdots & \ddots & \vdots \\ \phi_{n_p}^1 & \cdots & \phi_{n_p}^{n_s} \end{bmatrix} \tag{3.30}$$

$$\prod\nolimits_{\phi} \mathbf{S}(\boldsymbol{D}) = \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{S}(\boldsymbol{D}) \tag{3.31}$$

The solution to the optimisation problem in Eq. (3.29) is equivalent to solving an eigenvalue problem with the snapshot matrix $\boldsymbol{Y}$ [113], [114].

$$\boldsymbol{Y}\boldsymbol{Y}^T\phi^i = \lambda_i\phi^i \in \mathbb{R}^{n_p} i = 1, \dots, n_s \tag{3.32}$$

This eigenvalue problem can be reduced from an $n_p \times n_p$ problem to an $n_s \times n_s$ as seen below.

$$\boldsymbol{Y}^T\boldsymbol{Y}V^i = \lambda_i V^i \in \mathbb{R}^{n_s} i = 1, \dots, n_s \tag{3.33}$$

$$\phi^i = \frac{1}{\sqrt{\lambda_i}}\boldsymbol{Y}V^i \in \mathbb{R}^{n_p} i = 1, \dots, n_s \tag{3.34}$$

With the optimal modes found, an equation for the projection of the surface coordinates as a function of the design parameters can be written.

$$\mathrm{a}(\boldsymbol{D}) = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{S}(\boldsymbol{D}) \tag{3.35}$$

$$\prod\nolimits_{\phi} \mathbf{S}(\boldsymbol{D}) = \boldsymbol{A}\mathrm{a}(\boldsymbol{D}) \tag{3.36}$$

$$\prod\nolimits_{\phi} \mathbf{S}(\boldsymbol{D}) = \sum_{i=1}^{n_s} \mathrm{a}_i(\boldsymbol{D})\phi^i \tag{3.37}$$

From now on, the surface coordinates will be defined by the RHS of Eq. (3.37), so the projection operator is dropped from this point on.

$$\mathbf{S}(\boldsymbol{D}) = \sum_{i=1}^{n_s} \mathrm{a}_i\,(\boldsymbol{D})\phi^i \tag{3.38}$$

All of the modes and coefficients are stored in memory. If a large number of samples have been taken, this can be quite substantial. Some modes capture the interdependencies between different design settings more strongly than others. These modes are highlighted by having a large corresponding eigenvalue. To reduce storage costs, only the n_m most significant modes are kept.

$$\mathbf{S}(\boldsymbol{D}) = \sum_{i=1}^{n_m} \mathrm{a}_i\,(\boldsymbol{D})\phi^i \tag{3.39}$$

In order to make this parameterisation scheme useful, the parameterisation must be continuous across the design space. The CAD surface coordinates at the design parameter settings that were not sampled must be found via interpolation. Specifically, an analytic interpolation equation for the POD coefficients $a_i(D)$ must be found so that the grid coordinates with any design parameters within the design space can be found. The chosen method for this task is RBF interpolation with thin plate spline as defined in Eq. (3.42).

$$\tilde{\mathbf{S}}(\boldsymbol{D}) = \sum_{i=1}^{n_m} \tilde{\mathrm{a}}_i\,(\boldsymbol{D})\phi^i \tag{3.40}$$

$$\varphi(r) = \begin{cases} r^2 \log(r), & for\ r\ \neq 0 \\ 0, & for\ r = 0 \end{cases} \tag{3.41}$$

$$\tilde{\mathrm{a}}_i(\boldsymbol{D}) = \beta^i \cdot \begin{Bmatrix} 1 \\ \boldsymbol{D} \end{Bmatrix} + \sum_{k=1}^{n_s} \gamma_k^i \varphi\left(\left\|\boldsymbol{D} - \boldsymbol{D}^k\right\|\right), i = 1, \dots, n_m \tag{3.42}$$

The first expression in Eq. (3.42) is a first order polynomial that is added to the interpolation equation to improve the accuracy of derivative approximations [95]. The prerequisite step to having the analytic equation is to calculate the constant vectors β^i and γ^i for each mode. This is done by finding the values of the coefficients at the known design settings, so they produce the same surface coordinates. An additional condition for the RBF coefficient, as seen in Eq. (3.43), is also added to allow unique interpolation.

$$0 = \sum_{k=1}^{n_s} \gamma_k^i \varphi\left(\left\|\boldsymbol{D} - \boldsymbol{D}^k\right\|\right) \tag{3.43}$$

The constants $\beta^1, \dots, \beta^{n_m}$ and $\gamma^1, \dots, \gamma^{n_m}$ are then obtained by introducing an interpolation matrix H_D.

$$\boldsymbol{A}_D = \begin{bmatrix} \varphi(\|\boldsymbol{D}^1 - \boldsymbol{D}^1\|) & \cdots & \varphi(\|\boldsymbol{D}^1 - \boldsymbol{D}^{n_s}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|\boldsymbol{D}^{n_s} - \boldsymbol{D}^1\|) & \cdots & \varphi(\|\boldsymbol{D}^{n_s} - \boldsymbol{D}^{n_s}\|) \end{bmatrix} \tag{3.44}$$

$$\boldsymbol{P}_D = \begin{bmatrix} 1 & \cdots & 1 \\ \boldsymbol{D}^1 & \cdots & \boldsymbol{D}^{n_s} \end{bmatrix} \tag{3.45}$$

$$\boldsymbol{H}_D = \begin{bmatrix} \mathbf{0} & \boldsymbol{P}_D \\ \boldsymbol{P}_D^T & \boldsymbol{A}_D \end{bmatrix} \tag{3.46}$$

$$\mathrm{a}^i = \begin{Bmatrix} \mathrm{a}_i(\boldsymbol{D}^1) \\ \vdots \\ \mathrm{a}_i(\boldsymbol{D}^{n_s}) \end{Bmatrix} \tag{3.47}$$

$$\boldsymbol{H}_D \begin{Bmatrix} \beta^i \\ \gamma^i \end{Bmatrix} = \begin{Bmatrix} 0 \\ \mathrm{a}^i \end{Bmatrix} \tag{3.48}$$

After the linear system is solved in Eq. (3.48) to obtain the constants for each mode, Eq. (3.40) can be used to analytically obtain the surface coordinates across the whole design space. Thus, the need for a CAD engine in the loop has been removed whilst maintaining the benefits of a CAD parameterisation.

3.3 Adjoint Method for Aeroelastic FSI Simulations

The focus of this thesis is to perform high-fidelity aeroelastic optimisations of aircraft in an efficient way. The only feasible way of performing a fast optimisation for a multi-physics high-fidelity problem is to use gradient-based optimisation, especially when there are a large number of design variables. In aerodynamic applications, often 100s of design variables are required to produce a large enough design space capable of producing a significant improvement in performance [115]. This section focuses on how to obtain the gradient after a completing an aeroelastic simulation. There are two major ways of obtaining a gradient of an objective function with respect to an aircraft's shape. These are the finite differences (FD) approach and the adjoint method. The finite differences approach finds the sensitivity of the cost-function to each design parameter by perturbing the design parameter and re-evaluating the cost-function. The difference between the perturbed and non-perturbed value divided by the perturbation amount gives the sensitivity of the cost-function to the design parameter. This is succinctly shown by Eq. (3.49).

$$\frac{dI}{dD_i} = \frac{I(D_i + \Delta D_i) - I(D_i)}{\Delta D_i} \tag{3.49}$$

The major problem with the finite differences approach is that it requires $n_D + 1$ cost-function evaluations. This is prohibitive for high-fidelity aerodynamic optimisations that use a large number of design parameters. The adjoint method has been used in aerodynamic optimisation literature since 1988 [32]. The use of the adjoint method for aerodynamic optimisation has matured over the last few decades and it is now widely used within the commercial aviation industry [116].

The adjoint method has the major advantage of only requiring one evaluation of the objective function. This is essential for any practical high-fidelity gradient-based optimisation. The adjoint method therefore has a negligible dependence on the number of design variables employed, allowing the optimiser to explore a large design space without adding significant computational overhead. Finite differences still have their place in aerodynamic optimisation, although mostly for validation purposes.

3.3.1 Flow Adjoint

Two simple concepts are employed by the adjoint method to eliminate the dependency on the number of design variables. First, if you add a constant to a function, the gradient at all points will remain unchanged. Second, multiplying anything by zero will result in a product equal to zero.

$$I = C_D \tag{3.49}$$

In Eq. (3.49), the cost function has been set to the value of the drag coefficient. In order to find the sensitivity of the cost-function to the design parameters efficiently, the equation will be modified by adding a constant to the right-hand-side (RHS) of Eq. (3.49) that is also a function of the design parameters.

$$\mathcal{L} = I + \boldsymbol{\lambda} \cdot \boldsymbol{R} \tag{3.50}$$

$$\boldsymbol{R}\big(\boldsymbol{W}(\boldsymbol{D}), \boldsymbol{X}(\boldsymbol{D})\big) = \boldsymbol{0}, \qquad for\ all\ \ \boldsymbol{D} \in \mathbb{R} \tag{3.51}$$

This simple modification to the equation is a powerful one. $\boldsymbol{R}$ is the residual of the CFD simulation, for a sufficiently converged solution it can be considered to be a vector of zeros for all of the design space. At this point, the value of the adjoint vector $\boldsymbol{\lambda}$ is arbitrary as the term $\boldsymbol{\lambda} \cdot \boldsymbol{R}$ will be equal to zero for all of the design space. This is why the residual must be strongly converged for the adjoint method to work, the derivation depends on it. This allows the adjoint vector to be defined in a very favourable way.

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \frac{dI}{d\boldsymbol{D}} + \boldsymbol{\lambda} \cdot \frac{d\boldsymbol{R}}{d\boldsymbol{D}} + \frac{d\boldsymbol{\lambda}}{d\boldsymbol{D}} \cdot \boldsymbol{R} \tag{3.52}$$

Both $\boldsymbol{R}$ and $\frac{d\boldsymbol{R}}{d\boldsymbol{D}}$ are equal to zero so it can be seen how the sensitivity of the Lagrangian is equal to the sensitivity of the cost function. The term $\frac{d\boldsymbol{\lambda}}{d\boldsymbol{D}} \cdot \boldsymbol{R}$ is equal to zero and cannot be manipulated for any gain so it is dropped from the equation. The adjoint vector will not be constant for all of the design space, instead it will meet the criteria that allows for the elimination of the state variables' sensitivity to the design parameters $\frac{d\boldsymbol{W}}{d\boldsymbol{D}}$. With this term eliminated, the need for multiple simulations to be run is eliminated too as every other term can be calculated from the results of one converged solution.

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \frac{dI}{d\boldsymbol{D}} + \boldsymbol{\lambda}^T \frac{d\boldsymbol{R}}{d\boldsymbol{D}} \tag{3.53}$$

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \frac{\partial I}{\partial \boldsymbol{W}}\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \frac{\partial I}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{D}} + \boldsymbol{\lambda}^T\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}}\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{D}}\right) \tag{3.54}$$

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \left(\frac{\partial I}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}}\right)\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \left(\frac{\partial I}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right)\frac{d\boldsymbol{X}}{d\boldsymbol{D}} \tag{3.55}$$

The partial derivatives $\frac{\partial I}{\partial \boldsymbol{W}}$, $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}}$, $\frac{\partial I}{\partial \boldsymbol{X}}$ and $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}$ are all be provided by the CFD solver TAU. The partial differentiation of an aerodynamic cost function or flow residual is a challenging task. For the TAU solver this was addressed by Dwight in [37]. Due to the complexities involved in the derivation of these terms they will not be discussed here; it is sufficient to know that they are available. The remaining unknowns are $\boldsymbol{\lambda}$, $\frac{d\boldsymbol{W}}{d\boldsymbol{D}}$ and $\frac{d\boldsymbol{X}}{d\boldsymbol{D}}$. As seen in the derivation, the adjoint vector is arbitrary and thanks to this, the troublesome term $\frac{d\boldsymbol{W}}{d\boldsymbol{D}}$ can be eliminated while retaining the ability to calculate the gradient.

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}}\right]^T \boldsymbol{\lambda} = -\left\{\frac{\partial I}{\partial \boldsymbol{W}}\right\}^T \tag{3.56}$$

Equation (3.56) is known as the adjoint equation. The adjoint vector that satisfies Eq. (3.56) can be plugged into Eq. (3.55) and the gradient equation will no longer require the calculation of $\frac{d\boldsymbol{W}}{d\boldsymbol{D}}$.

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \left(\frac{\partial I}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right)\frac{d\boldsymbol{X}}{d\boldsymbol{D}} \tag{3.57}$$

There only unknown left to calculate in order to obtain the gradient is the mesh sensitivity to the design parameters $\frac{d\boldsymbol{X}}{d\boldsymbol{D}}$.

Solving the linear system in Eq. (3.56) is not a trivial exercise. Two important techniques, Krylov subspaces and GMRES need to be understood to see how large non-symmetric linear systems are solved in modern solvers. These two techniques are detailed in Appendix D.

3.3.2 Mesh Adjoint

The mesh sensitivity can of course be calculated by finite differences. This is not a ridiculous prospect, as calculating mesh deformations is significantly less time consuming than a full RANS simulation.

$$\frac{d\boldsymbol{X}}{d\boldsymbol{D}_i} = \frac{\boldsymbol{X}(\boldsymbol{D}_i + \Delta\boldsymbol{D}_i) - \boldsymbol{X}(\boldsymbol{D}_i)}{\Delta\boldsymbol{D}_i} \tag{3.58}$$

Obtaining the mesh sensitivity through finite differences obtains the final unknown term and the gradient can be calculated. However, as mesh sizes get larger, the mesh deformation will still take a significant time to compute. To further improve the efficiency of the gradient calculation, a mesh adjoint can be applied using the same principles as described in the previous flow adjoint section. The mesh adjoint was first used by Nielson and Park [117] to eliminate the need to obtain the mesh sensitivity via finite differences. The mesh adjoint adds a mesh residual to the Lagrangian equation to provide another mesh sensitivity term. This term can then be used to remove the mesh sensitivity from the gradient equation.

$$\mathcal{L} = I + \boldsymbol{\lambda}_a^T \boldsymbol{R}_a + \boldsymbol{\lambda}_m^T \boldsymbol{R}_m \tag{3.59}$$

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \frac{dI}{d\boldsymbol{D}} + \boldsymbol{\lambda}_a^T \frac{d\boldsymbol{R}_a}{d\boldsymbol{D}} + \boldsymbol{\lambda}_m^T \frac{d\boldsymbol{R}_m}{d\boldsymbol{D}} \tag{3.60}$$

The mesh residual is defined by the mesh deformation process, i.e. the relationship between the surface mesh and the volume mesh.

$$\boldsymbol{R}_m = \boldsymbol{X} - \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{S}}\boldsymbol{S} \tag{3.61}$$

$$\boldsymbol{R}_m = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{X}}\boldsymbol{X} - \boldsymbol{S} \tag{3.62}$$

The relationship is obviously determined by the choice of mesh deformation strategy. An explicit mesh deformation strategy, one where the volume mesh points can be determined by a simple matrix vector product, will have the residual defined by Eq. (3.61). An implicit mesh deformation strategy, one which requires the solution of a large linear system, will have the residual defined by Eq. (3.62).

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \frac{\partial I}{\partial \boldsymbol{W}}\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \frac{\partial I}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{D}} + \boldsymbol{\lambda}_a^T\left(\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{D}}\right) + \boldsymbol{\lambda}_m^T\left(\frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{S}}\frac{d\boldsymbol{S}}{d\boldsymbol{D}} + \frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{D}}\right) \tag{3.63}$$

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \left(\frac{\partial I}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}}\right)\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \left(\frac{\partial I}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}_m^T \frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{X}}\right)\frac{d\boldsymbol{X}}{d\boldsymbol{D}} + \boldsymbol{\lambda}_m^T \frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{S}}\frac{d\boldsymbol{S}}{d\boldsymbol{D}} \tag{3.64}$$

It can be seen in Eq. (3.64) that the addition of a mesh adjoint has no effect on the flow adjoint equation. The flow adjoint vector can therefore be carried out in exactly the same way. This leaves the gradient equation in the following form:

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \left(\frac{\partial I}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}_m^T \frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{X}}\right)\frac{d\boldsymbol{X}}{d\boldsymbol{D}} + \boldsymbol{\lambda}_m^T \frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{S}}\frac{d\boldsymbol{S}}{d\boldsymbol{D}} \tag{3.65}$$

Now the mesh adjoint vector can be chosen so that the mesh sensitivity is multiplied with a vector of zeros, thus eliminating it from the gradient equation.

$$\left[\frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{X}}\right]^T \boldsymbol{\lambda}_m = -\left\{\frac{\partial I}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}}\right\}^T \tag{3.66}$$

Equation (3.66) highlights an advantage of an explicit mesh deformation scheme over and implicit one. For an explicit scheme, the Jacobian $\frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{X}}$ is the identity matrix making the calculation of the mesh adjoint vector a simple addition problem. An implicit scheme would require solving a large linear system that will naturally take a significant time to do.

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \boldsymbol{\lambda}_m^T \frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{S}}\frac{d\boldsymbol{S}}{d\boldsymbol{D}} \tag{3.67}$$

The Jacobian $\frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{S}}$ is the identity matrix multiplied by minus 1 for implicit schemes. It does have to be calculated for explicit schemes, however. In this work, the mesh deformation scheme employed is an explicit version of RBF mesh deformation. The mesh adjoint terms for the employed RBF mesh deformation scheme $\frac{\partial \boldsymbol{R}_m}{\partial \boldsymbol{S}}$ are derived in the next chapter.

The only unknown left in the gradient equation is the surface mesh sensitivity to the design variables $\frac{d\boldsymbol{S}}{d\boldsymbol{D}}$. This is much simpler than the volume mesh sensitivity and it often has an analytic relationship. An analytic relationship means there is no need for any finite differences and can be computed very quickly. The use of the mesh adjoint and flow adjoint has rendered the relationship between the gradient calculation time and the number of design variables to be completely negligible.

Mesh Sensitivity for the RBF Mesh Deformation Method

The mesh deformation scheme employed in this project is an explicit RBF mesh deformation. As this is an explicit scheme, it enables the gradient to be calculated faster. Two terms need to be found, the sensitivity of the mesh to the surface $\frac{\partial \boldsymbol{X}}{\partial \mathbf{S}}$ and the sensitivity of the surface to the design parameters $\frac{d\mathbf{S}}{d\boldsymbol{D}}$. In this implementation, the surface is approximated by a surrogate CAD model that is defined by the design parameters. This means $\frac{d\mathbf{S}}{d\boldsymbol{D}}$ is defined analytically as seen in chapter 3.2.1. The sensitivity of the volume mesh to the surrogate CAD surface $\frac{\partial \boldsymbol{X}}{\partial \mathbf{S}}$ is derived in this section.

$$\Delta \boldsymbol{X}_{\rho,i} = \boldsymbol{b}_i \left(\begin{Bmatrix} \eta_{\rho,1} \\ \eta_{\rho,2} \\ \eta_{\rho,3} \\ \eta_{\rho,4} \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ x_i \\ y_i \\ z_i \end{Bmatrix} + \sum_{j=1}^{n_S} \zeta_j \varphi\left(\left\|\boldsymbol{X}_i - \mathbf{S}_j\right\|\right) \right) \begin{matrix} i = 1 \dots n_V \\ \rho = x, y, z \end{matrix} \tag{3.68}$$

The interpolation coefficients η and ζ are different for x, y, z but the RBF function φ is the same for each direction.

$$\boldsymbol{b}_i = \begin{cases} 0 & for\ d_i \geq R_Z \\ 1 & for\ d_i \leq R_F \\ \dfrac{R_z - d_i}{R_z - R_F} & for\ R_F < d_i < R_Z \end{cases} \tag{3.69}$$

The volume points $\boldsymbol{X}$ and base points $\mathbf{S}$ used in the RBF function are the undeformed points defined by the first set of design parameters. When the design parameters are updated, the displacements required to move the undeformed base points to their new position are calculated, this vector is $\Delta\boldsymbol{S}$.

$$\Delta\mathbf{S} = \mathbf{S}(\boldsymbol{D}) - \mathbf{S}(\boldsymbol{D}_0) \tag{3.70}$$

$$\frac{d\Delta\mathbf{S}}{d\boldsymbol{D}} = \frac{d\mathbf{S}}{d\boldsymbol{D}} \tag{3.71}$$

The displacements are from the undeformed base points rather than the previous design iteration. This makes the mesh adjoint easier to implement as the RBF function φ and scaling factor $\boldsymbol{b}_i$ are not dependent on the design variables. If the RBF function were dependent on the design variables, then a matrix of the size $(n_V \times n_S \times \boldsymbol{D})$ would need to be calculated during each gradient equation. The enormous term $\frac{d\varphi}{d\boldsymbol{D}}$ can be omitted with no loss of mesh quality simply by deforming from the undeformed condition.

This means the only terms in Eq. (3.68) dependent on the design parameters are η and ζ. These are implicitly dependent on the design parameters through the known deformation of the surrogate base points $\Delta\mathbf{S}_D$. Equation (3.71) means that the interpolation coefficients' sensitivity to the displacements are equal to their sensitivity to the surface points.

$$\Delta\mathbf{S}_{\rho,i}(\boldsymbol{D}) = \begin{Bmatrix} \eta_{\rho,1} \\ \eta_{\rho,2} \\ \eta_{\rho,3} \\ \eta_{\rho,4} \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ x_i \\ y_i \\ z_i \end{Bmatrix} + \sum_{j=1}^{n_S} \zeta_j \varphi\left(\left\|\boldsymbol{X}_i - \mathbf{S}_j\right\|\right) \begin{matrix} i = 1 \dots n_S \\ \rho = x, y, z \end{matrix} \tag{3.72}$$

The distance from the surface for the i^{th} node $\boldsymbol{d}_i$ is obviously 0 for all base points. It can be seen from Eq. (3.69) that the value of $\boldsymbol{b}_i$ will be 1 for all of the base points, hence why $\boldsymbol{b}_i$ has disappeared from the interpolation equation. To aid in the derivation of the mesh adjoint, Eq. (3.72) is put in a matrix format.

$$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Delta S_{\rho,1} \\ \Delta S_{\rho,2} \\ \Delta S_{\rho,3} \\ \vdots \\ \Delta S_{\rho,n_S} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & x_1 & x_2 & x_3 & \cdots & x_{n_S} \\ 0 & 0 & 0 & 0 & y_1 & y_2 & y_3 & \cdots & y_{n_S} \\ 0 & 0 & 0 & 0 & z_1 & z_2 & z_3 & \cdots & z_{n_S} \\ 1 & x_1 & y_1 & z_1 & 0 & \varepsilon_{1,2} & \varepsilon_{1,3} & \cdots & \varepsilon_{1,n_S} \\ 1 & x_2 & y_2 & z_2 & \varepsilon_{2,1} & 0 & \varepsilon_{2,3} & \cdots & \varepsilon_{2,n_S} \\ 1 & x_3 & y_3 & z_3 & \varepsilon_{3,1} & \varepsilon_{3,2} & 0 & \ddots & \varepsilon_{3,n_S} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & x_{n_S} & y_{n_S} & z_{n_S} & \varepsilon_{n_S,1} & \varepsilon_{n_S,2} & \varepsilon_{n_S,3} & \cdots & 0 \end{bmatrix} \begin{Bmatrix} \eta_{\rho,1} \\ \eta_{\rho,2} \\ \eta_{\rho,3} \\ \eta_{\rho,4} \\ \zeta_{\rho,1} \\ \zeta_{\rho,2} \\ \zeta_{\rho,3} \\ \vdots \\ \zeta_{\rho,n_S} \end{Bmatrix} \rho = x, y, z \tag{3.73}$$

The sensitivity of the interpolation coefficients to the surface displacements $\frac{\partial \zeta}{\partial \Delta \mathbf{S}}$ is the inverse of the matrix in Eq. (3.73). The first four rows in Eq. (3.73) set conditions to ensure that the interpolation gives a unique result.

The sensitivity of the deformed volume mesh points to the interpolation coefficients are found in a similar way.

$$\boldsymbol{X}(\boldsymbol{D}) = \Delta \boldsymbol{X}(\boldsymbol{D}) - \boldsymbol{X}(\boldsymbol{D}_0) \tag{3.74}$$

$$\frac{d\boldsymbol{X}}{d\boldsymbol{D}} = \frac{d\Delta \boldsymbol{X}}{d\boldsymbol{D}} = \frac{\partial \Delta \boldsymbol{X}}{\partial \boldsymbol{\zeta}} \frac{\partial \boldsymbol{\zeta}}{\partial \Delta \mathbf{S}} \frac{d\Delta \mathbf{S}}{d\boldsymbol{D}} \tag{3.75}$$

Equation (3.75) shows that only the sensitivity of the mesh coordinate displacements to the interpolation coefficients $\frac{\partial \Delta \boldsymbol{X}}{\partial \boldsymbol{\zeta}}$ need to be found.

$$\begin{Bmatrix} \Delta X_{\rho,1} \\ \Delta X_{\rho,2} \\ \Delta X_{\rho,3} \\ \vdots \\ \Delta X_{\rho,n_V} \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 & 0 & \varepsilon_{1,2} & \varepsilon_{1,3} & \cdots & \varepsilon_{1,n_S} \\ 1 & x_2 & y_2 & z_2 & \varepsilon_{2,1} & 0 & \varepsilon_{2,3} & \cdots & \varepsilon_{2,n_S} \\ 1 & x_3 & y_3 & z_3 & \varepsilon_{3,1} & \varepsilon_{3,2} & 0 & \ddots & \varepsilon_{3,n_S} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & x_{n_V} & y_{n_V} & z_{n_V} & \varepsilon_{n_V,1} & \varepsilon_{n_V,2} & \varepsilon_{n_V,3} & \cdots & \varepsilon_{n_V,n_S} \end{bmatrix} \begin{Bmatrix} \eta_{\rho,1} \\ \vdots \\ \eta_{\rho,4} \\ \zeta_{\rho,1} \\ \vdots \\ \zeta_{\rho,n_S} \end{Bmatrix} \rho = x, y, z \tag{3.76}$$

In practical applications, the matrix in Eq. (3.75) is too large to be stored. It has the dimensions $(3n_v) \times (3n_s)$. It must be calculated on the fly each iteration which is unfortunate as it does not change from one iteration to another. The smaller matrix in Eq. (3.73) can be stored however, as it only contains information pertaining to a reduced set of the surface points. When the size of the matrix being inverted is kept small, the matrix can be inverted on one core in only a number of seconds as seen in Fig. 6. This keeps the time needed for the interpolation down. The author of this thesis performed the numerical experiment shown in Fig. 6 by inverting dense matrices of different sizes using the Python scripting language.

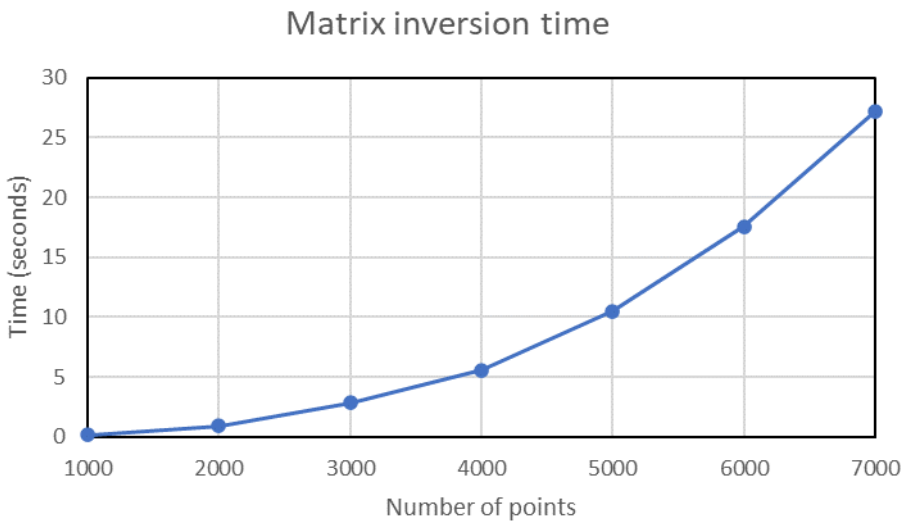

Fig. 6 The inversion time of a matrix compared with the size of the matrix.

Selim and Koomullil [101] performed a literature review of mesh deformation strategies and concluded that the RBF approach with a data reduction algorithm was the best in terms of efficiency, robustness and mesh quality. As seen in this section, it also makes the calculation of the mesh adjoint easier which is another advantage. The only remaining term required for the gradient calculation not to be derived is the sensitivity of the base points to the design parameters $\frac{d\mathbf{S}}{d\boldsymbol{D}}$.

Surface Sensitivity of the Surrogate CAD Model

The parameterisation model used in this research project is a CAD-based surrogate model as proposed by Kamil Bobrowsi *et al*. [109]. A description of this technique and the motivation for using it have been provided in chapter 3.2. The sensitivity of the surface to the design parameters is derived below.

$$\mathbf{S}(\boldsymbol{D}) = \sum_{i=1}^{n_m} \tilde{a}_i(\boldsymbol{D})\phi^i \tag{3.77}$$

$$\tilde{a}_i(\boldsymbol{D}) = \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n_D+1} \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ \boldsymbol{D}_1 \\ \vdots \\ \boldsymbol{D}_{n_D} \end{Bmatrix} + \sum_{k=1}^{n_s} \gamma_k^i \varphi\left(\left\|\boldsymbol{D} - \boldsymbol{D}^k\right\|\right) \; i = 1, \dots, n_m \tag{3.78}$$

$$\varphi(r) = \begin{cases} r^2 \log(r), & for\ r \neq 0 \\ 0, & for\ r = 0 \end{cases} \tag{3.79}$$

The constants β and γ are calculated by satisfying that the surrogate model produces the exact displacements taken from the samples. The POD modes ϕ are a reduced set of constant eigenvectors resulting from the snapshot matrix described in chapter 3.2. The POD modes are not functions of the design parameters, only the POD coefficients $\tilde{a}$ are functions of the design parameters.

$$\frac{d\mathbf{S}}{d\boldsymbol{D}} = \sum_{i=1}^{n_m} \frac{d\tilde{a}_i}{d\boldsymbol{D}} \phi^i \tag{3.80}$$

$$\frac{d\boldsymbol{S}}{d\boldsymbol{D}} = \sum_{i=1}^{n_m} \left[\delta_i + \sum_{k=1}^{n_s} \gamma_k^i \frac{d\varphi(\|\boldsymbol{D} - \boldsymbol{D}_k\|)}{d\boldsymbol{D}} \right] \phi^i \tag{3.81}$$

$$\frac{d\boldsymbol{S}}{d\boldsymbol{D}} = \sum_{i=1}^{n_m} \left[\delta_i + \sum_{k=1}^{n_D} \gamma_k^i (\boldsymbol{D} - \boldsymbol{D}_k)(2\log(\|\boldsymbol{D} - \boldsymbol{D}_k\|) + 1) \right] \phi^i \tag{3.82}$$

$$\boldsymbol{\delta}_i = \begin{Bmatrix} \beta_2^i \\ \beta_3^i \\ \vdots \\ \beta_{n_D+1}^i \end{Bmatrix} i = 1, \dots, n_m \tag{3.83}$$

All the terms in the surface sensitivity equation have now been calculated. The gradient of the cost function with respect to the design parameters can now be found without the need for finite differences.

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \boldsymbol{\lambda}_m^T \frac{\partial \boldsymbol{R}_m}{\partial \mathbf{S}} \frac{d\mathbf{S}}{d\boldsymbol{D}} \tag{3.84}$$

All the terms on the RHS of Eq. (3.84) can be calculated analytically once the flow adjoint vector has been computed. The computational cost of $\frac{d\mathbf{S}}{d\boldsymbol{D}}$ is similar to the cost of calculating the displacements from a new set of design parameters. This is only weakly dependent on the number of design variables making this mesh adjoint implementation very efficient.

3.4 Aeroelastic Adjoint Derivation

The purpose of this research project has been to enable fast aerodynamic shape optimisation while considering the structural effects of the aerodynamic loading. Up to this point, only aerodynamics has been considered when deriving the gradient. In an aeroelastic simulation, the mesh coordinates $\boldsymbol{X}$ are no longer at a constant position throughout the simulation.

The mesh coordinates are a function of two things. First, the base points $\boldsymbol{S}$ which are defined at the start of the simulation. Second, the structural displacements that change throughout the simulation. It is useful for the purpose the gradient derivation to define two meshes. The jig mesh $\boldsymbol{X}_J$, that is independent of the structural displacements, and the converged mesh $\boldsymbol{X}_\Delta$ that is a function of the structural displacements.

$$\boldsymbol{X}_J = \boldsymbol{X}_J(\boldsymbol{S}_J) \tag{3.85}$$

$$\boldsymbol{X}_\Delta = \boldsymbol{X}_\Delta(\boldsymbol{X}_J, \boldsymbol{u}_{vol}) \tag{3.86}$$

It is important to be clear on the exact relationship that the displaced mesh has with the structural displacements as different implementations can result in the aeroelastic adjoint equation being harder to solve. In this implementation, the initial mesh $\boldsymbol{X}_0$ is used to define the interpolation coefficients and these remain constant throughout the optimisation.

$$\boldsymbol{X}_\Delta = \boldsymbol{X}_J + \boldsymbol{u}_{vol} \tag{3.87}$$

$$\boldsymbol{X}_\Delta = \boldsymbol{X}_0 + \boldsymbol{u}_J + \boldsymbol{u}_{vol} \tag{3.88}$$

All the displacement values are defined relative to the undeformed coordinates that are defined in the first optimisation iteration. In other words, the mesh deformation due to the structural displacements is independent of the design parameters. This is important as it means matrix variables are not dependent on the design variables. This prevents the need to compute three-dimensional matrices when obtaining the gradient. A downside of this coupling-surface approach is that the output from the interpolations will lose their accuracy if the jig shape changes significantly during the optimisation. If the jig shape does change significantly, the optimisation should be restarted with new interpolation coefficients.

$$\boldsymbol{u}_{vol} = \boldsymbol{u}_{vol}(\boldsymbol{u}_{surf}) \tag{3.89}$$

$$\boldsymbol{u}_{surf} = \boldsymbol{u}_{surf}(\boldsymbol{u}_s) \tag{3.90}$$

$$\boldsymbol{u}_s = \boldsymbol{u}_s(\boldsymbol{D}) \tag{3.91}$$

$\boldsymbol{u}_{vol}$ is related to the aerodynamic surface displacements $\boldsymbol{u}_{surf}$ through the mesh deformation procedure. Then the aerodynamic surface displacements are related to the structural displacements $\boldsymbol{u}_s$ through the interpolation procedure which is discussed in the next section. The structural displacements are the state variables of the structural solver and their sensitivity to the design variables $\frac{d\boldsymbol{u}_s}{d\boldsymbol{D}}$ cannot be found analytically. Therefore, to enable fast gradient-based optimisation, the Lagrangian must be extended to include a term that can eliminate $\frac{d\boldsymbol{u}_s}{d\boldsymbol{D}}$. Just like with the flow adjoint and mesh adjoint, the residual of the structural solver is added to the Lagrangian.

$$\mathcal{L} = I + \boldsymbol{\lambda}_a^T \boldsymbol{R}_a + \boldsymbol{\lambda}_s^T \boldsymbol{R}_s + \boldsymbol{\lambda}_m^T \boldsymbol{R}_m \tag{3.92}$$

$$\mathcal{L} = I + \boldsymbol{\lambda}_a^T \boldsymbol{R}_a + \boldsymbol{\lambda}_s^T \boldsymbol{R}_s \tag{3.93}$$

To reduce clutter in the derivation, the mesh adjoint terms will be omitted.

$$\boldsymbol{R}_s = \boldsymbol{F}_s - \boldsymbol{K}\boldsymbol{u}_s \tag{3.94}$$

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \frac{dI}{d\boldsymbol{D}} + \boldsymbol{\lambda}_a^T \frac{d\boldsymbol{R}_a}{d\boldsymbol{D}} + \boldsymbol{\lambda}_s^T \frac{d\boldsymbol{R}_s}{d\boldsymbol{D}} \tag{3.95}$$

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \frac{\partial I}{\partial \boldsymbol{W}}\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \frac{\partial I}{\partial \boldsymbol{X}_\Delta}\frac{d\boldsymbol{X}_\Delta}{d\boldsymbol{D}} + \lambda_a^T\left(\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\frac{d\boldsymbol{W}}{d\boldsymbol{D}} + \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta}\frac{d\boldsymbol{X}_\Delta}{d\boldsymbol{D}}\right) + \boldsymbol{\lambda}_s^T\left(\frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{F}_s}\frac{d\boldsymbol{F}_s}{d\boldsymbol{D}} + \frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{u}_s}\frac{d\boldsymbol{u}_s}{d\boldsymbol{D}}\right) \tag{3.96}$$

A significant change in the gradient equation can be seen in Eq. (3.96). The cost function and aerodynamic residual are now dependent on the converged displaced mesh $\boldsymbol{X}_\Delta$ and not directly on the jig mesh. The converged mesh $\boldsymbol{X}_\Delta$ along with the structural force vector $\boldsymbol{F}_s$ are the variables through which the aerodynamic and structural solvers are coupled. In order to tackle this gradient, each term must be understood. To start this process, each term is expanded so anything which has not yet been derived can be.

$$\frac{d\boldsymbol{X}_\Delta}{d\boldsymbol{D}} = \frac{d\boldsymbol{X}_J}{d\boldsymbol{D}} + \frac{d\boldsymbol{u}_{vol}}{d\boldsymbol{D}} \tag{3.97}$$

$$\frac{d\boldsymbol{u}_{vol}}{d\boldsymbol{D}} = \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}\frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s}\frac{d\boldsymbol{u}_S}{d\boldsymbol{D}} \tag{3.98}$$

$$\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{W}} = \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}} \tag{3.99}$$

$$\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{X}_\Delta} = \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta} \tag{3.100}$$

$$\frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{F}_s} = \boldsymbol{I} \tag{3.101}$$

$$\frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{u}_s} = -\boldsymbol{K} \tag{3.102}$$

$$\begin{aligned}\frac{d\mathcal{L}}{d\boldsymbol{D}} = &\left(\frac{\partial I}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_a^T\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_s^T\frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{F}_s}\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{W}}\right)\frac{d\boldsymbol{W}}{d\boldsymbol{D}} \\ &+ \left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T\frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{F}_s}\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{X}_\Delta}\right)\frac{d\boldsymbol{X}_\Delta}{d\boldsymbol{D}} + \left(\boldsymbol{\lambda}_s^T\frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{u}_s}\right)\frac{d\boldsymbol{u}_s}{d\boldsymbol{D}}\end{aligned} \tag{3.103}$$

Substituting Eq. (3.97) to Eq. (3.102) into Eq. (3.103) produces the gradient equation that the aeroelastic adjoint equations can be formed from. As the displaced mesh has simply been defined as

the addition of the jig mesh coordinates and the structural displacements, the Jacobian $\frac{\partial \boldsymbol{X}_\Delta}{\partial \boldsymbol{X}_J}$ is simply the identity matrix.

$$\frac{d\mathcal{L}}{d\boldsymbol{D}} = \left(\frac{\partial I}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}}\right) \frac{d\boldsymbol{W}}{d\boldsymbol{D}}$$

$$+ \left(\left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s} - \boldsymbol{\lambda}_s^T \boldsymbol{K}\right) \frac{d\boldsymbol{u}_s}{dD}$$

$$+ \left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{d\boldsymbol{X}_J}{d\boldsymbol{D}} \tag{3.104}$$

$\frac{\partial I}{\partial \boldsymbol{W}}$, $\frac{\partial I}{\partial \boldsymbol{X}_\Delta}$, $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}$ and $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta}$ are found in the same way as they were in the regular aerodynamic adjoint equations. $\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}$ and $\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}}$ are new, but they can also be calculated within the flow solver TAU [118]. This leaves the following terms: $\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}$, $\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}$ and $\frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s}$.

$\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}$ is similar to the term $\frac{\partial \Delta \boldsymbol{X}}{\partial \Delta \boldsymbol{S}}$ that was calculated in chapter 3.2.1. It uses exactly the same mesh deformation method, RBF with reduced a data reduction algorithm. However, there is one subtle difference to do with the surface points. The surface points used in the jig shape calculation are not actually on the CFD mesh, they are points defined on the surrogate CAD model surface. Whereas, in the interpolation of the structural displacements to the volume mesh, there is an intermediary step of calculating all the displacements on the surface of the actual CFD mesh. A subset of these displacements is then interpolated to the volume mesh. This means the interpolation coefficients used in the mesh deformation of the structural displacements are different from those used in the jig shape calculation. However, as will be explained in chapter 4.2, the mesh deformation term used in the coupled-adjoint is not consistent with the actual mesh deformation used in the simulation.

3.4.1 Derivatives of the Cross Discipline Interpolation Terms

The two remaining terms that have not been discussed so far in this section, $\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}$ and $\frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s}$, are known as cross-derivatives. They represent the sensitivity between vectors from two separate disciplines. In this case, one vector is from aerodynamics and the other is from structural mechanics. They are used to interpolate the information from one discipline to the other in the FSI simulation.

It would be possible to use entirely separate interpolation processes in each direction. For instance, a simple nearest neighbour interpolation could be used to calculate the forces on the structural mesh. While RBF interpolation could be used to calculate the displacements on the aerodynamic mesh. However, there is an implementation benefit to having the interpolation process in one direction be the transpose of the process in the other.

$$\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} = \left[\frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s}\right]^T \tag{3.105}$$

$$\left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}\right]^T \boldsymbol{\lambda}_s = \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s} \boldsymbol{\lambda}_s \tag{3.106}$$

$$\left[\frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s}\right]^T \left[\left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}\right]^T$$

$$= \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \left[\left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}\right]^T \tag{3.107}$$

The implementation benefit is demonstrated in Eq. (3.107). There is no new code that needs to be written for the coupled-adjoint when performing the transpose interpolation process. This is not ground-breaking, but it makes life easier for the developer.

3.5 Solving the Coupled System

The coupled-adjoint equations can be solved in two different ways. Barcelos *et al*. [119] developed a number of Newton-Krylov-Schur methods for solving the FSI problem. Kenway *et al*. [77] introduced a coupled Newton-Krylov (NK) approach for solving the coupled-adjoint equations, and also implemented an Aitken acceleration [120] into the block Gauss-Seidel (BGS) approach. They found that the coupled NK approach improved the convergence of the coupled-adjoint relative to the BGS approach. The coupled NK approach is described in Appendix E. The difficulty with employing the coupled NK approach for either the primal solution or adjoint solution is the memory overhead of the method. Additionally, the FlowSimulator package used in this work does not enable the retrieval of the flow and structural Jacobians that are required to form the full matrix used by the NK approach. For this reason, the BGS approach was chosen.

3.5.1 Block Gauss-Seidel Method

Rather than create a single linear system of equations to be solved, a lagged iteration scheme is used. The major benefit of this approach is that the discipline specific solvers can be reused with only minor modifications.

$$\frac{\partial I}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}} = \boldsymbol{0} \tag{3.108}$$

$$\left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s} - \boldsymbol{\lambda}_s^T \boldsymbol{K} = \boldsymbol{0} \tag{3.109}$$

In the case of the flow-adjoint, the right-hand side of the equation must be modified by the interpolated structural term. The modification is a simple addition. In the case of the structural adjoint, the gravity switch needs to be turned off before reusing the structural solver. This is because Eq. (3.109) has no reference to the gravity force like the structural problem in the FSI simulation has. In other words, the force due to gravity is not dependent on the shape design parameters. The stiffness matrix for a linear-elastic simulation is symmetric, meaning the structural solver can be used directly to obtain the structural adjoint vector by supplying the RHS of Eq. (3.111) as the force.

$$\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\right]^T \boldsymbol{\lambda}_a^k = -\left(\left[\frac{\partial I}{\partial \boldsymbol{W}}\right]^T + \left[\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}}\right]^T \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s} \boldsymbol{\lambda}_s^{k-1}\right) \tag{3.110}$$

$$\boldsymbol{K}\boldsymbol{\lambda}_s^k = \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \left[\left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \{\boldsymbol{\lambda}_a^k\}^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \{\boldsymbol{\lambda}_s^{k-1}\}^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}\right]^T \tag{3.111}$$

$$\left\{\frac{d\mathcal{L}}{d\boldsymbol{D}}\right\}^k = -\left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \{\boldsymbol{\lambda}_a^k\}^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \{\boldsymbol{\lambda}_s^k\}^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{R}_m}{\partial \mathbf{S}} \frac{d\mathbf{S}}{d\boldsymbol{D}} \tag{3.112}$$

Block Gauss-Seidel Solution procedure

1) The solution is initialised for $\{\boldsymbol{\lambda}_s\}^0 = 0$
2) Equation (3.110) is solved to find $\{\boldsymbol{\lambda}_a\}^k$
3) Equation (3.111) is solved to find $\{\boldsymbol{\lambda}_s\}^k$
4) Equation (3.112) is solved to find the k^{th}gradient evaluation.
5) Steps 2, 3 and 4 are cycled through until the gradient converges.

The convergence criteria selected in this project is that the difference between the L2 norm of the gradient of the last iteration must be under 5% of the previous iteration's L2 norm. The Jacobians $\frac{\partial \boldsymbol{R}_m}{\partial \mathbf{S}}$ and $\frac{d\mathbf{S}}{d\boldsymbol{D}}$ as seen in Eq. (3.112) are exactly the same in this aeroelastic adjoint as they would be in a regular aerodynamic adjoint evaluation. This is because the mesh residual $\boldsymbol{R}_m$ is independent of the structural displacements.

4. Development of Aeroelastic Optimisation Methods

4.1 Hybrid Mesh Deformation Approach to Adjoint Aeroelastic Optimisation

To solve an FSI simulation and coupled-adjoint system of equations with a block Gauss-Seidel approach, a solver for both the aerodynamic and structural discipline is required. For the flow solution and accompanying flow adjoint, the DLR TAU solver was employed. To solve the structural and structural adjoint problems, the open source finite element solver MYSTRAN [121] was used. The parameterisation scheme used in both optimisation cases was the method described in chapter 3.2. This parameterisation method creates a reduced order model of a parametric CAD model; hence the parameterisation scheme is continuous and analytically differentiable.

There are three distinct mesh deformation operations that occur in an aeroelastic optimisation. There is the deformation from the base shape to the design shape according to the design parameters, this sets the undeformed shape of the aircraft for the given design iteration of the optimisation. The second deformation occurs in the FSI simulation when the structural displacements are propagated to the aerodynamic mesh. These two deformations make use of the RBF with a data reduction algorithm mesh deformation strategy. The third mesh deformation occurs within the coupled-adjoint loop, this is a transpose mesh deformation operation, the DGM algorithm is used here instead of RBF to improve the convergence properties of the coupled-adjoint matrix. The use of DGM in the coupled-adjoint loop, in place of the RBF algorithm used in the FSI simulation, to improve the diagonal dominance of the coupled-adjoint matrix is the primary novelty of this research project. A flow diagram of the overall optimisation workflow is shown in Fig. 7.

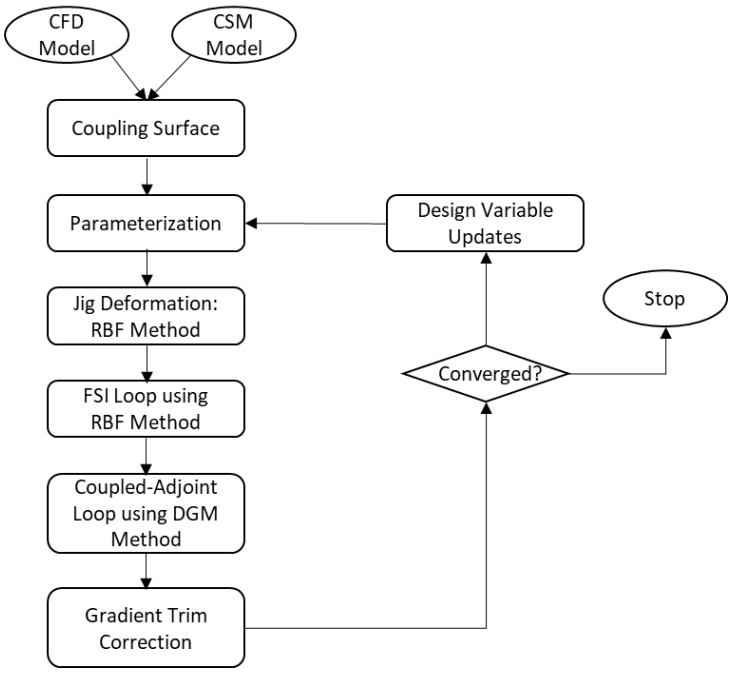

Fig. 7 Workflow for the hybrid mesh deformation approach to coupled-adjoint optimisations.

The workflows of the FSI simulation loop and coupled-adjoint loop are shown in Fig. 8 and Fig. 9 respectively. The parametric sensitivity for a given design iteration is found via the mesh adjoint approach. Although it would be possible to substitute the RBF mesh adjoint with a DGM mesh adjoint, it is not done as the efficiency savings are minimal as the data reduction algorithm makes the RBF mesh adjoint a fast process anyway.

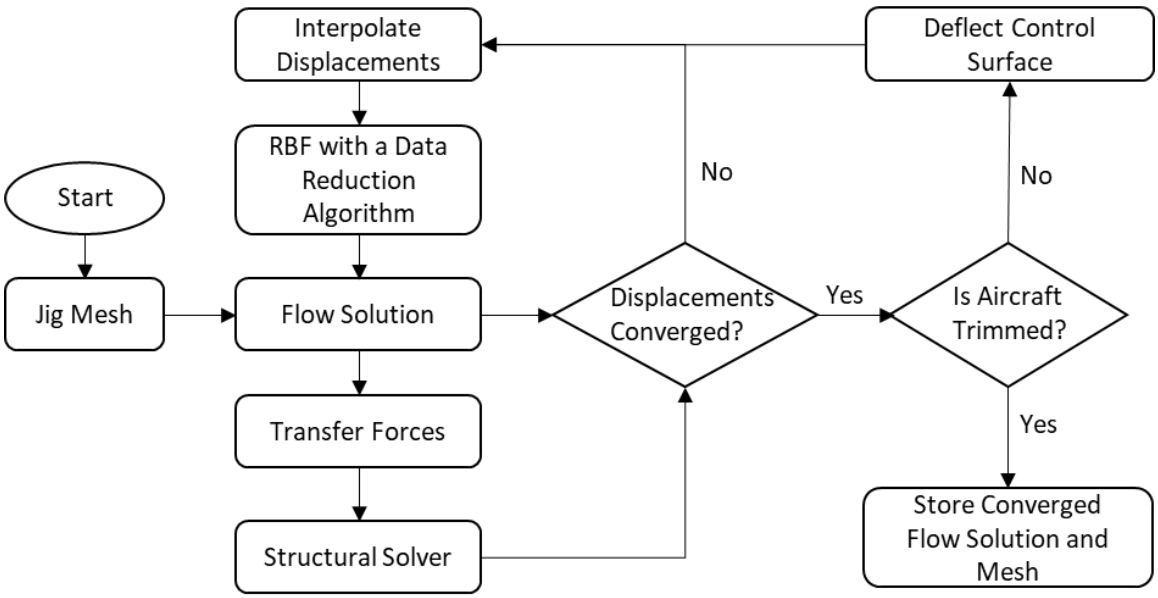

Fig. 8 The FSI simulation loop with an outer loop for handling trim.

The strategy employed to converge the full FSI simulation with an outer trim loop took the form of iterative convergence. The optimisation of the LANN wing did not require a trim loop as only the lift was held constant, and this was achieved through TAU updating the angle of attack during the flow simulation. For the XRF1 however, longitudinal trim was achieved by deflecting the elevator on the HTP which in this project is considered to be a variable camber surface. The elevator deflection angle was found rapidly by making the minimum flow residual required for convergence in the trim loop 10^{-3}. This allowed many outer trim loop loops to be performed quickly, allowing a deflection angle which trimmed the aircraft longitudinally to be found quickly by use of the Broyden algorithm [122]. Once this deflection angle was found, the flow solver's minimum residual value for convergence was set to 10^{-6}. A side-effect of this strategy is that a large number of FSI inner iterations were required which results in a correspondingly large number of mesh deformations. This emphasised the need for a robust and efficient mesh deformation algorithm in the FSI loop, particularly if the TCOS approach is employed.

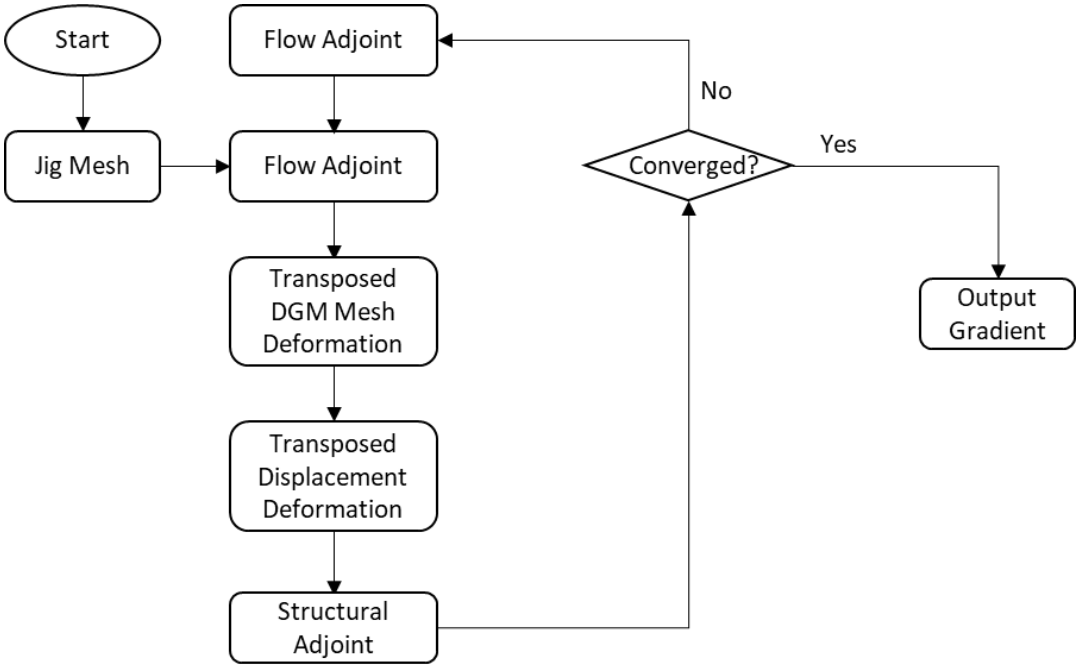

Fig. 9 The coupled-adjoint loop using DGM for the transposed mesh deformation operation.

After a converged FSI simulation, a coupled-adjoint problem must be solved for each cost function. In this case, it must be solved for the drag, lift and pitching moment. In the LANN wing optimisation, the pitching moment was not constrained so only the drag and lift were solved for. The coupled-adjoint loop is similar to a reordered inner loop of an FSI simulation. The DGM algorithm holds an advantage over many other mesh deformation strategies as it can be stored in memory even for very large meshes due to its sparse nature. This makes implementing the transposed operation even easier as the transpose mesh deformation operation is found with a straightforward matrix vector multiplication in memory.

4.2 Three-field Coupled-Adjoint Approach for an Explicit Mesh Deformation Strategy

The coupled-adjoint formulation derived in chapter 3.4 is often called a two-field formulation. The fields refer to the coupled flow adjoint and structural adjoint problems within the coupled-adjoint procedure. In the two-field formulation, the interpolation of quantities between the aerodynamic and structural disciplines are included in the two adjoint problems. It is possible to include a third adjoint problem so the interpolation can be considered separately. Maute *et al.* [12], [46], [123], [124] adopted this method and concluded it was more suitable for problems with large structural deformations despite its higher computational cost. The Lagrangian for a three-field approach includes another adjoint term to handle the interpolation. The relationship between the surface displacements on the aerodynamic mesh and structural mesh are not explicitly considered, instead the extra adjoint equation handles their relationship implicitly.

$$\mathcal{L} = I + \boldsymbol{\lambda}_a^T \boldsymbol{R}_a + \boldsymbol{\lambda}_\Delta^T \boldsymbol{R}_\Delta + \boldsymbol{\lambda}_s^T \boldsymbol{R}_s \tag{4.1}$$

An implicit mesh deformation strategy will cause the interpolation residual to take a different form than an explicit mesh deformation strategy. An implicit strategy will be:

$$\boldsymbol{R}_{\Delta} = \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_{vol}} \boldsymbol{u}_{vol} - \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_{s}} \boldsymbol{u}_{s} \tag{4.2}$$

An explicit strategy will take the form:

$$\boldsymbol{R}_{\Delta} = \boldsymbol{u}_{vol} - \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_{s}} \boldsymbol{u}_{s} \tag{4.3}$$

$$\frac{\partial \boldsymbol{R}_{\Delta}}{\partial \boldsymbol{u}_{s}} = \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_{s}} \tag{4.4}$$

A 3-field strategy modifies the coupled-adjoint equation and produces the following equation:

$$\begin{aligned} \frac{d\mathcal{L}}{d\boldsymbol{D}} = & \left(\frac{\partial I}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_{a}^{T} \frac{\partial \boldsymbol{R}_{a}}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_{s}^{T} \frac{\partial \boldsymbol{F}_{s}}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}} \right) \frac{d\boldsymbol{W}}{d\boldsymbol{D}} \\ & + \left(\left(\frac{\partial I}{\partial \boldsymbol{X}_{\Delta}} + \boldsymbol{\lambda}_{a}^{T} \frac{\partial \boldsymbol{R}_{a}}{\partial \boldsymbol{X}_{\Delta}} + \boldsymbol{\lambda}_{s}^{T} \frac{\partial \boldsymbol{F}_{s}}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_{\Delta}} \right) + \boldsymbol{\lambda}_{\Delta}^{T} \frac{\partial \boldsymbol{R}_{\Delta}}{\partial \boldsymbol{u}_{vol}} \right) \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} \frac{d\boldsymbol{u}_{surf}}{d\boldsymbol{D}} \\ & + \left(\boldsymbol{\lambda}_{\Delta}^{T} \frac{\partial \boldsymbol{R}_{\Delta}}{\partial \boldsymbol{u}_{s}} - \boldsymbol{\lambda}_{s}^{T} \boldsymbol{K} \right) \frac{d\boldsymbol{u}_{s}}{d\boldsymbol{D}} + \left(\frac{\partial I}{\partial \boldsymbol{X}_{\Delta}} + \boldsymbol{\lambda}_{a}^{T} \frac{\partial \boldsymbol{R}_{a}}{\partial \boldsymbol{X}_{\Delta}} + \boldsymbol{\lambda}_{s}^{T} \frac{\partial \boldsymbol{F}_{s}}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_{\Delta}} \right) \frac{d\boldsymbol{X}_{J}}{d\boldsymbol{D}} \end{aligned} \tag{4.5}$$

The key difference of the three-field approach vs the two-field approach is that the Jacobian $\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}$ has been moved outside of the 2nd term's brackets. If an implicit mesh deformation strategy is used, the Jacobian $\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}$ will not have to be calculated. The 2nd term can be solved by performing a mesh adjoint calculation with the input vector being the term $\left(\frac{\partial I}{\partial \boldsymbol{X}_{\Delta}} + \lambda_{a}^{T} \frac{\partial \boldsymbol{R}_{a}}{\partial \boldsymbol{X}_{\Delta}} + \boldsymbol{\lambda}_{s}^{T} \frac{\partial \boldsymbol{F}_{s}}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_{\Delta}} \right)$.

In the case of an explicit mesh deformation strategy, like DGM, the three-field approach turns out to be the same as the two-field approach. In the two-field approach, the term $\boldsymbol{\lambda}_{\Delta}^{T}$ is equal to the expression $\left(\frac{\partial I}{\partial \boldsymbol{X}_{\Delta}} + \boldsymbol{\lambda}_{a}^{T} \frac{\partial \boldsymbol{R}_{a}}{\partial \boldsymbol{X}_{\Delta}} + \boldsymbol{\lambda}_{s}^{T} \frac{\partial \boldsymbol{F}_{s}}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_{\Delta}} \right)$. By substituting this value of $\boldsymbol{\lambda}_{\Delta}^{T}$ into the three-field gradient equation, the two-field gradient equation is retrieved. The use of an explicit mesh deformation strategy therefore ensures that the number of coupled expressions required for the coupled-adjoint is only two.

4.3 Non-Consistent Mesh Deformation

In the derivation of the gradient, the sensitivity of the volume displacements to the surface displacements $\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}$ is assumed to be the mesh deformation operation used in the fluid-structure interaction simulation. This ensures the calculated gradient is exact. An interesting observation was noted during the course of this project. The convergence of the coupled-adjoint is significantly affected by a mesh deformation strategy that only uses some of the surface points. The convergence of the FSI simulation is largely independent of the mesh deformation algorithm provided that the algorithm produces a suitably high-quality mesh. To demonstrate this, the LANN wing was used to run an FSI simulation at Mach 0.8 and at an angle of attack of 2° using three different mesh deformation approaches.

Table 1 Convergence comparison of LANN wing FSI simulations.

Mesh deformation technique	Maximum structural displacement delta from previous iteration					
	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6
RBF 1000 points	1.892e-01	8.948e-02	2.239e-02	8.842e-03	2.376e-03	8.735e-04
RBF 4000 points	1.892e-01	8.949e-02	2.240e-02	8.840e-03	2.372e-03	8.676e-04
RBF All points	1.892e-01	8.951e-02	2.243e-02	8.858e-03	2.384e-03	8.737e-04

Table 1 shows that LANN wing converged to its final displaced shape in six iterations regardless of how many base points were used in the RBF mesh deformation. The independence of the rate of convergence to the mesh deformation algorithm demonstrated for FSI simulations does not hold for the coupled-adjoint equations. A key observation of this thesis is that a mesh deformation algorithm that uses only a subset of the surface points to interpolate from introduces a destabilising effect into the coupled-adjoint Jacobian. The mesh deformation operation in the FSI simulation acts on a much different vector field than the vector field that the transpose mesh deformation operation acts on in the coupled-adjoint.

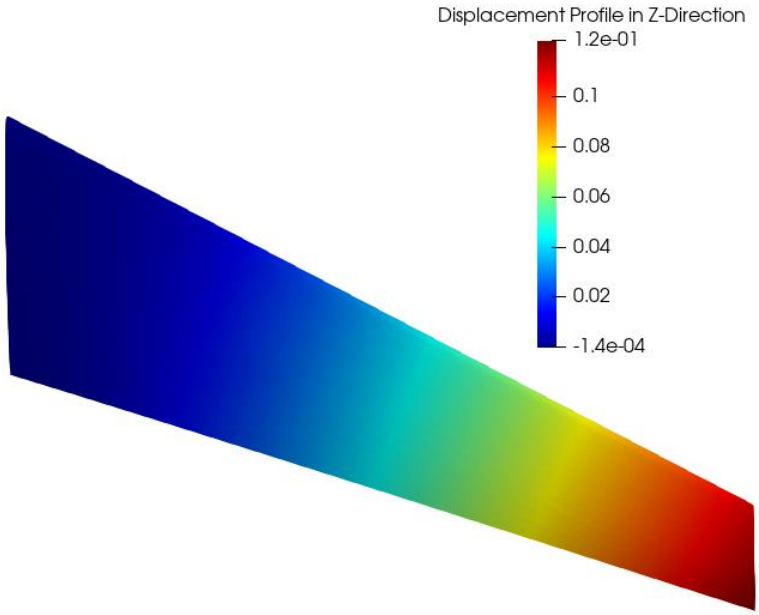

Fig. 10 Displacement profile on LANN wing in z-direction.

Figure 10 shows the displacement profile on the CFD surface of the LANN wing before the volume mesh deformation takes place. The bluer regions represent areas that have small displacement values while the redder regions represent areas with large displacement values. In the FSI simulation, the mesh deformation chooses a subset of points on the smooth displacement field and interpolates those displacements to every other point in the volume mesh. The effect of not using every surface point on the mesh as a base point is that there is a loss of geometric integrity after each deformation. However, as the displacement field is smooth, the use of a subset of points is valid as the interpolation of

displacements will accurately approximate the displacements of surface points which are not chosen as base points in the deformation.

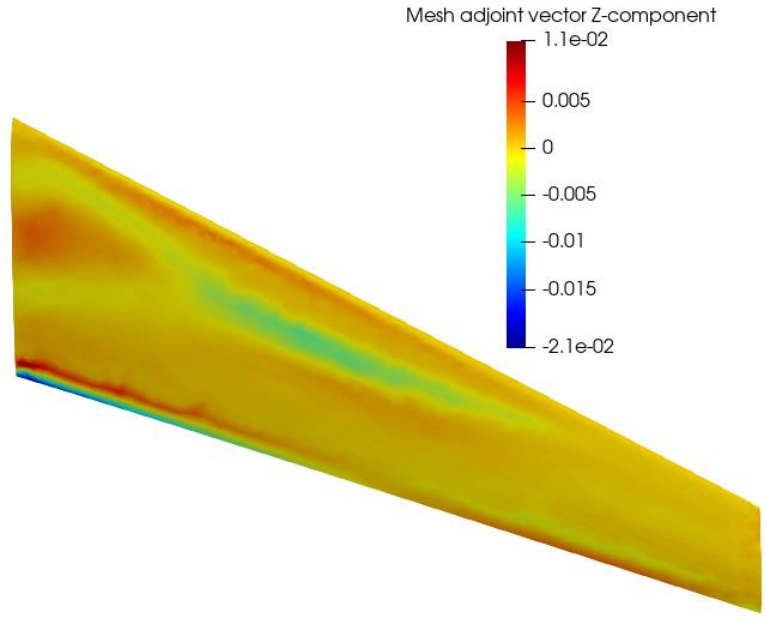

Fig. 11 Mesh adjoint profile on the LANN wing in the z-direction.

Figure 11 shows the mesh adjoint profile on the CFD surface of the LANN wing before the transpose volume mesh deformation takes place. In a block Gauss-Seidel approach to the coupled-adjoint, the transpose mesh deformation operates on the adjoint vector before it is interpolated to the CSM surface. The use of a data-reduction algorithm for the transposed operation artificially reduces the fidelity of the CFD surface. The effect of this reduction in fidelity is that a significantly modified adjoint profile will be transferred to the CSM mesh after the transposed mesh deformation.

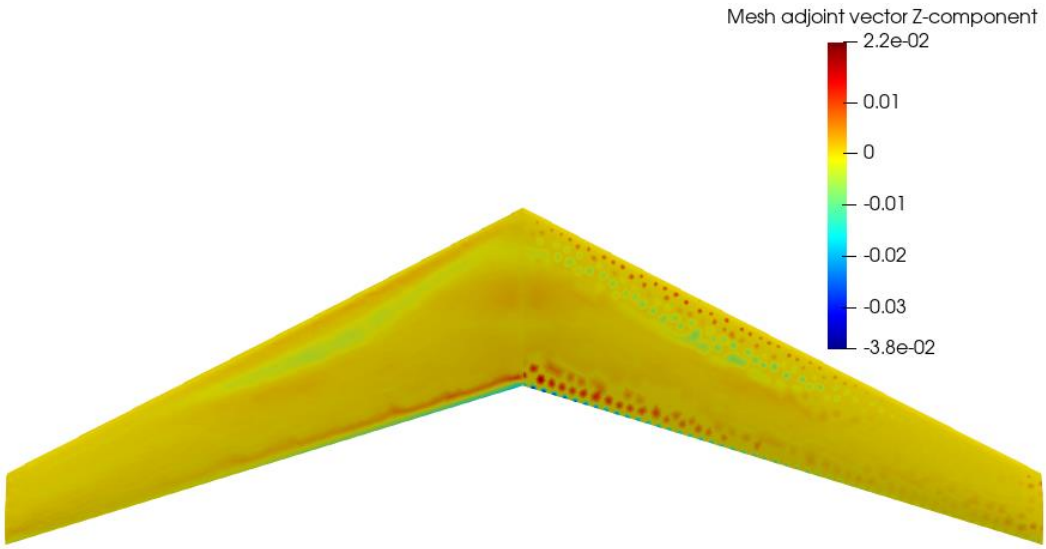

Fig. 12 Comparison of adjoint vectors created after transpose mesh deformation operation.

Figure 12 shows the adjoint field on the surface of the CFD mesh after the transposed mesh deformation for two different mesh deformation algorithms. The wing on the left of Fig. 12 is the adjoint field when the RBF mesh deformation algorithm uses all the surface points as base points. The wing on the right of Fig. 12 is the adjoint field when the RBF mesh deformation algorithm only uses 1000 surface points as base points. The use of a data-reduction algorithm for the transposed mesh deformation results in the concentration of data at the chosen base points. This concentration of data is why dots with large magnitudes appear on the right wing in Fig. 12. Crucially, the surface points that were omitted as base points in the mesh deformation have their contribution to the mesh adjoint concentrated at the chosen base points. The magnitude of the values at the base points corresponds directly with the number of nodes that were omitted from the mesh deformation. This effect has important implications for both the accuracy of the output gradient and the rate of convergence of the coupled-adjoint.

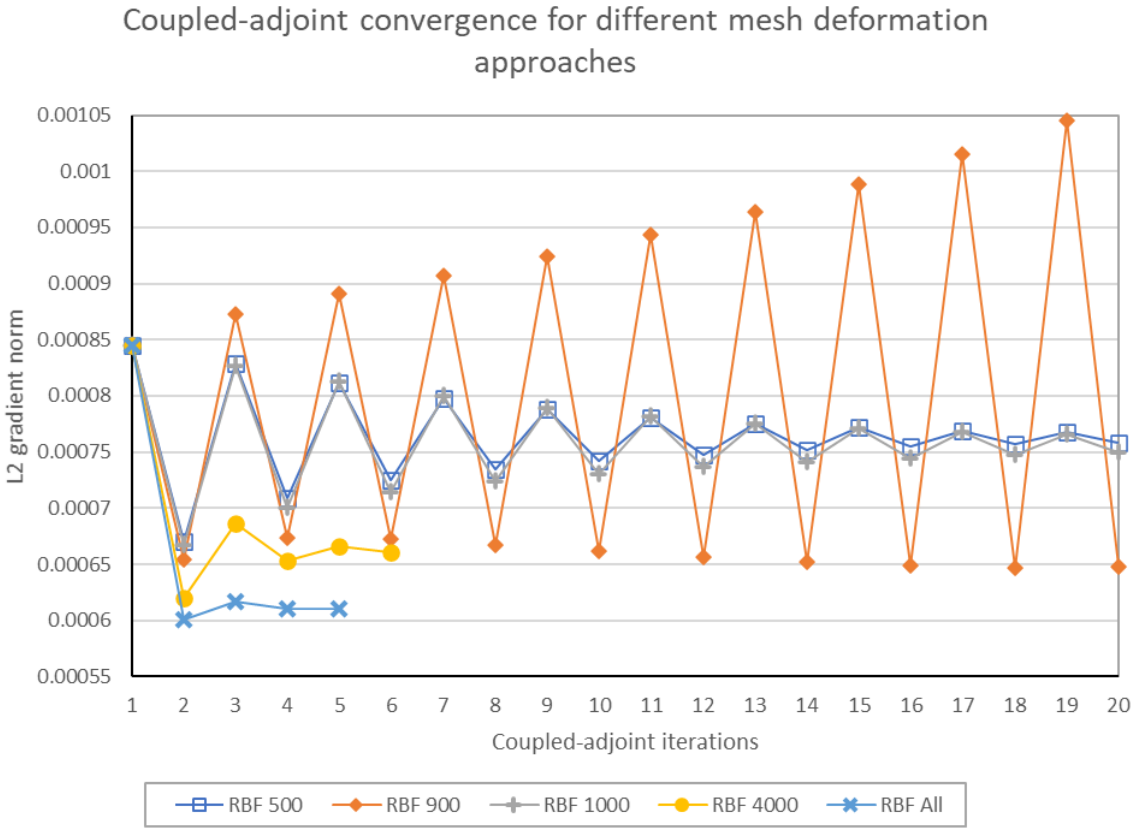

Fig. 13 Convergence of the L2 gradient norm using RBF with different levels of data-reduction on the base points.

The converged FSI solution that was used as the starting point for all the coupled-adjoint procedures in Fig. 13 was the simulation that used 4000 surface points as the base points for the mesh deformation. The coupled-adjoint was considered converged when the change in the L2 gradient norm was less than 1% different relative to the value of the previous iteration. The trend shown by Fig. 13 is that the less surface points that are used as base points in the mesh deformation, the worse the rate of convergence is. Importantly, the case where the mesh deformation algorithm matches the one used in the FSI simulation takes another iteration to converge when compared to the mesh deformation algorithm which uses all the surface points. The coupled-adjoint diverges when only 900

surface points are used. A surprising result was observed when the points were reduced further. When the mesh deformation algorithm used 500 points, the coupled-adjoint converged again. This indicates that the effect of the mesh deformation on the convergence of the coupled-adjoint is not simply related to the number of surface points used. Although it can be stated that the effect of using less surface points in the mesh deformation in general has a damaging effect on the convergence of the coupled-adjoint.

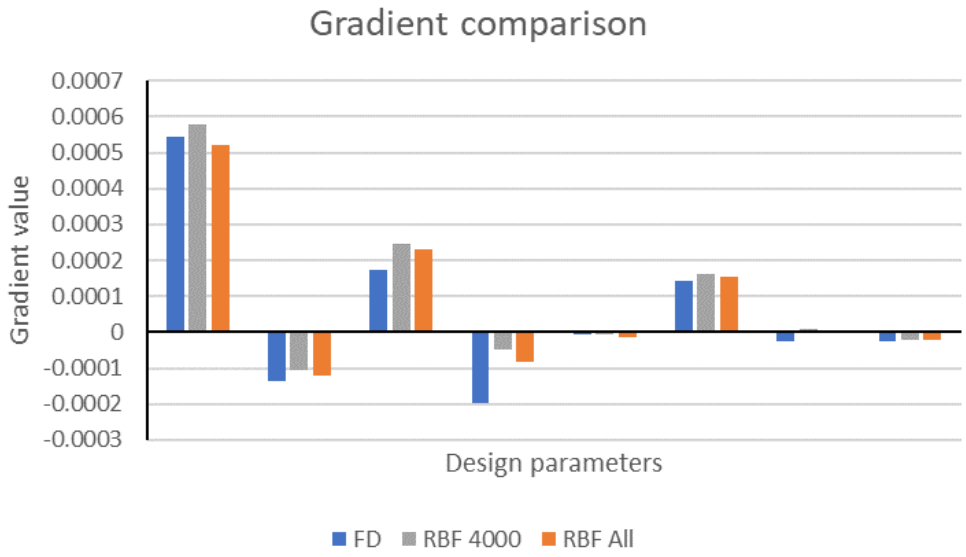

Fig. 14 Comparison of output gradients when using different mesh deformation techniques.

Figure 14 shows how the coupled-adjoint gradients compare against the finite difference gradient. It shows that the unexpected result that RBF with 4000 base points is less accurate than the RBF approach that used all the surface points as base points. The third and fourth parameters show that the output gradient has a noticeable level of error present. This is not due to the mesh deformation procedure however, but due to the complexity of the discretised governing equations. Even so, the produced gradient is sufficiently accurate to make an optimisation using the coupled-adjoint possible.

4.3.1 RBF Data-Reduction Effect on the Coupled-Adjoint Matrix

In this project the coupled adjoint is solved using a lagged block Gauss-Seidel approach. The block Gauss-Seidel algorithm is known to converge if the spectral radius ρ of the matrix has a value less than 1 [125].

$$\begin{bmatrix} \left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\right]^T & \left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{W}}\right]^T \\ \left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}\right]^T & \left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s} - \boldsymbol{K}\right]^T \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda}_a \\ \boldsymbol{\lambda}_s \end{Bmatrix} = \begin{Bmatrix} \left[\frac{\partial I}{\partial \boldsymbol{W}}\right]^T \\ \left[\frac{\partial I}{\partial \boldsymbol{u}_s}\right]^T \end{Bmatrix} \tag{4.6}$$

The block Gauss-Seidel approach to coupled-adjoint equations written in Eq. (4.6) can be written as:

$$\begin{Bmatrix} \boldsymbol{\lambda_a^{(i+1)}} \\ \\ \boldsymbol{\lambda_s^{(i+1)}} \end{Bmatrix} = \begin{bmatrix} \mathbf{0} & -\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\right]^{-T}\left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{W}}\right]^{T} \\ \mathbf{0} & \left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s} - \boldsymbol{K}\right]^{-T}\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}\right]^{T}\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\right]^{-T}\left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{W}}\right]^{T} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda_a^{(i)}} \\ \\ \boldsymbol{\lambda_s^{(i)}} \end{Bmatrix}$$
$$+ \begin{Bmatrix} \left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\right]^{-T}\left[\frac{\partial I}{\partial \boldsymbol{W}}\right]^{T} \\ \\ \left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s} - \boldsymbol{K}\right]^{-T}\left(\left[\frac{\partial I}{\partial \boldsymbol{u}_s}\right]^{T} - \left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}\right]^{T}\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\right]^{-T}\left[\frac{\partial I}{\partial \boldsymbol{W}}\right]^{T}\right) \end{Bmatrix} \tag{4.7}$$

The iteration matrix of the block Gauss-Seidel algorithm is the matrix in Eq. (4.7). The scheme therefore converges if the following equation holds true:

$$\rho\left(\left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s} - \boldsymbol{K}\right]^{-T}\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}\right]^{T}\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}\right]^{-T}\left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{W}}\right]^{T}\right) < 1 \tag{4.8}$$

Unfortunately, the flow solver TAU does not expose any APIs that make the retrieval of the Jacobians $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}}$, $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta}$, $\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}$, $\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}}$ possible. This makes quantifying the effect on the convergence of the mesh deformation algorithm by use of the spectral radius impossible at this time. Another criteria that guarantees the convergence of a Gauss-Seidel iterative scheme is to check whether the matrix is strictly diagonally dominant [126]. A matrix is diagonally dominant if the absolute values of all the diagonal elements are greater than the sum of the absolute values of the non-diagonal elements in their respective rows.

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad for\ all\ i \tag{4.9}$$

The structural force vector $\boldsymbol{F}_s$ is the one defined from the structural residual meaning it is only a function of $\boldsymbol{u}_s$ through the aerodynamic force. It is not equal to the stiffness matrix $\boldsymbol{K}$.

$$\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s} = \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}}\frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_s} \neq \boldsymbol{K} \tag{4.10}$$

In fact, $\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s}$ is a small term that captures the effect of a change in the CFD surface normals on the aerodynamic force and therefore the interpolated structural force. This means the matrix $\left[\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s} - \boldsymbol{K}\right]^{T}$ effectively still has the same level of diagonal dominance as the stiffness matrix. The mesh deformation operator is only prevalent in the bottom block row of Eq. (4.6).

$$\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}\right]^{T} = \begin{bmatrix} \frac{\partial \boldsymbol{R}_{a,1}}{\partial \boldsymbol{u}_{s,1}} & \cdots & \frac{\partial \boldsymbol{R}_{a,n_{vol}}}{\partial \boldsymbol{u}_{s,1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{R}_{a,1}}{\partial \boldsymbol{u}_{s,n_s}} & \cdots & \frac{\partial \boldsymbol{R}_{a,n_{vol}}}{\partial \boldsymbol{u}_{s,n_s}} \end{bmatrix} \tag{4.11}$$

To shed some light on how the convergence of the coupled-adjoint is affected by the mesh deformation algorithm, the off-diagonal matrix $\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}\right]^{T}$ is examined. In particular, the impact the mesh deformation algorithm has on the diagonal dominance of the coupled-adjoint matrix is explored.

To make the analysis of the mesh deformation clearer, a simplification is made about the interpolation in order to isolate the mesh deformation effects. The simplification is to assume a special case whereby the CSM and CFD surface meshes align perfectly thus making the force and displacement transfer a simple one for one transfer between corresponding nodes.

$$\frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{u}_{s,j}} = \frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{X}_{\Delta}} \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} \frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_{s,j}} \tag{4.12}$$

$$\frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{u}_{s,j}} = \frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{X}_{\Delta}} \frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf,j}} \tag{4.13}$$

All the terms in the matrix $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}$ are off-diagonal in the coupled-adjoint matrix. This means for any given column in $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}$, the larger the sum of the column's absolute values, the worse the coupled-adjoint's convergence characteristics will be. When using a mesh deformation operation with reduced points, some rows in $\left[\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}\right]^T$ become a row of zeros while others become denser. Specifically, the rows related to the subset of surface points used in the mesh deformation become denser.

The reduction in diagonal dominance becomes a serious problem when many elements in $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s}$ are large. An element in $\frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{u}_{s,j}}$ will be large when the i^{th} volume node is close to the j^{th} surface node, unless the i^{th} node is also a surface node that is used as a base point in the volume deformation. An RBF mesh deformation with a data reduction algorithm introduces a large number of volume nodes that are very close to surface nodes. This creates large non-zero elements at off-diagonal positions in the coupled-adjoint matrix. The term $\frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{X}_{\Delta}}$ is complex even for a 2D Euler case [127]. However, some things can be asserted qualitatively. The value will be larger when the air speed is higher simply because the change in flux over the affected control volume face will be larger, thus changing the residual by more. Secondly, the presence of shock switches on the scalar dissipation in TAU which uses an unstructured generalisation of the JST scheme [37], [129]. The scalar dissipation is also a function of the grid points meaning that the presence of shock waves on the surface will reduce the diagonal dominance in the coupled-adjoint matrix.

$$\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} = \boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1} \tag{4.14}$$

$$\frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{u}_{surf,j}} = \frac{\partial \boldsymbol{R}_{a,i}}{\partial \boldsymbol{X}_{\Delta}} \boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]_j^{-1}, \qquad i = 1, \dots, n_{vol} \tag{4.15}$$

The values in column j of the matrix are determined by evaluating Eq. (4.15) for each value of i. Therefore, the test for the effect on coupled-adjoint convergence due to the mesh deformation algorithm is to check if some columns in the matrix $\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1}$ have larger element values when less surface points are used. The matrix $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_{\Delta}}$ is independent of the mesh deformation algorithm but the values of its elements are more significant at nodes closer to the surface. The values are larger closer to the surface as cells are smaller, meaning that any mesh perturbation affects the flux through these cells more significantly. Therefore, elements in the $\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1}$ matrix that are relevant to nodes close to the surface have a stronger effect on the convergence of the coupled-adjoint procedure.

The effect of a data reduction algorithm on the coupled-adjoint matrix is quantified in the section below.

4.3.2 Quantification of RBF Data-Reduction Effect on the Coupled-Adjoint Matrix

It is the columns in the matrix $\frac{\partial R_a}{\partial \boldsymbol{u}_s}$ that affect the diagonal dominance of the coupled-adjoint matrix. Assume again that the interpolation is one-to-one from the CSM mesh to the CFD mesh, but the RBF mesh deformation algorithm does not use a polynomial for interpolation. Figure 15 is used to consider two scenarios. The first scenario will use points 1 and 3 as the surface points, the second will use points 1, 2 and 3 as the surface points. The two scenarios will produce a different mesh deformation matrix.

For an RBF mesh deformation algorithm which only uses 2 surface points, the matrix $\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1}$ takes the form:

$$\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1} = \frac{1}{{\varepsilon_{12}}^2}\begin{bmatrix} {\varepsilon_{12}}^2 & 0 \\ 0 & {\varepsilon_{12}}^2 \\ \varepsilon_{12}\varepsilon_{i2} & \varepsilon_{12}\varepsilon_{i1} \\ \vdots & \vdots \end{bmatrix} \tag{4.16}$$

For an RBF mesh deformation algorithm which only uses 3 surface points, the matrix $\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1}$ takes the form:

$$\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1} = \frac{1}{2\varepsilon_{12}\varepsilon_{13}\varepsilon_{23}}\begin{bmatrix} 2\varepsilon_{12}\varepsilon_{13}\varepsilon_{23} & 0 & 0 \\ 0 & 2\varepsilon_{12}\varepsilon_{13}\varepsilon_{23} & 0 \\ 0 & 0 & 2\varepsilon_{12}\varepsilon_{13}\varepsilon_{23} \\ a & b & c \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{4.17}$$

$$a = -\varepsilon_{i1}{\varepsilon_{23}}^2 + \varepsilon_{i2}\varepsilon_{13}\varepsilon_{23} + \varepsilon_{i3}\varepsilon_{12}\varepsilon_{23} \tag{4.18}$$

$$b = \varepsilon_{i1}\varepsilon_{13}\varepsilon_{23} - \varepsilon_{i2}{\varepsilon_{13}}^2 + \varepsilon_{i3}\varepsilon_{12}\varepsilon_{23} \tag{4.19}$$

$$c = \varepsilon_{i1}\varepsilon_{12}\varepsilon_{23} + \varepsilon_{i2}\varepsilon_{12}\varepsilon_{23} - \varepsilon_{i3}{\varepsilon_{12}}^2 \tag{4.20}$$

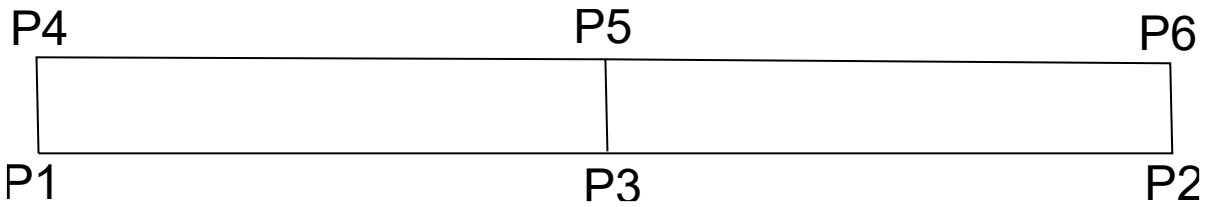

Fig. 15 A 6-point two-dimensional grid.

Using the mesh from Fig. 15, the matrices in Eq. (4.16) and Eq. (4.17) can be computed and compared. For the matrix that uses 2 surface points, the surface points are assumed to be point 1 and point 2. For the matrix that uses 3 surface points, the surface points are assumed to be points 1, 2 and 3.

$$\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0.5 & 0.5 \\ 1 & 0.1 \\ 0.51 & 0.51 \\ 0.1 & 1 \end{bmatrix} \tag{4.21}$$

$$\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.962 & 0.057 & 0.085 \\ 0.1 & 0.1 & 0.820 \\ 0.057 & 0.962 & 0.085 \end{bmatrix} \tag{4.22}$$

Just by visually comparing the matrices in Eq. (4.21) and Eq. (4.22) it is obvious that the addition of another surface point has significantly reduced the size of the elements in the columns of the matrix $\boldsymbol{\phi}^{vol}\left[\boldsymbol{\phi}^{surf}\right]^{-1}$. This effect is significant when changing from 3 surface points to 2, but the concept applies when changing from 50,000 surface points to 10,000. It is clear that the columns of the matrix $\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_{surf}}$ will be significantly affected, which in turn will significantly affect the rows in the coupled-adjoint matrix.

4.3.3 Coupled-Adjoint Convergence with Different Fidelity CFD and CSM Meshes

It is usually not the case that the surface of the CSM mesh will have the same fidelity as the CFD mesh, meaning a one for one interpolation between the two meshes will not happen. Two test cases were generated to examine how the fidelity of the CSM mesh affected the convergence of the coupled-adjoint and FSI simulations. The two generated FEMs were wing boxes which assumed the wing was a solid material with a Young's modulus of 4×10^{10} and a Poisson ratio of 0.3. The high-fidelity FEM consisted of 4541 nodes on the upper surface and 4564 nodes on the lower surface. The low-fidelity FEM consisted of 120 nodes on the upper surface and 120 nodes on the lower surface.

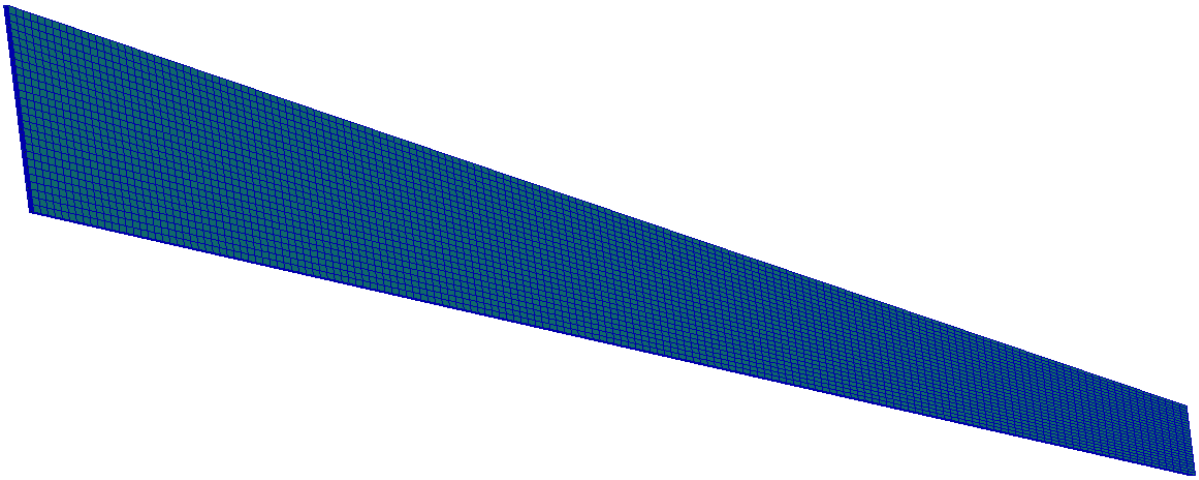

Fig. 16 The generated high-fidelity FEM for the LANN wing.

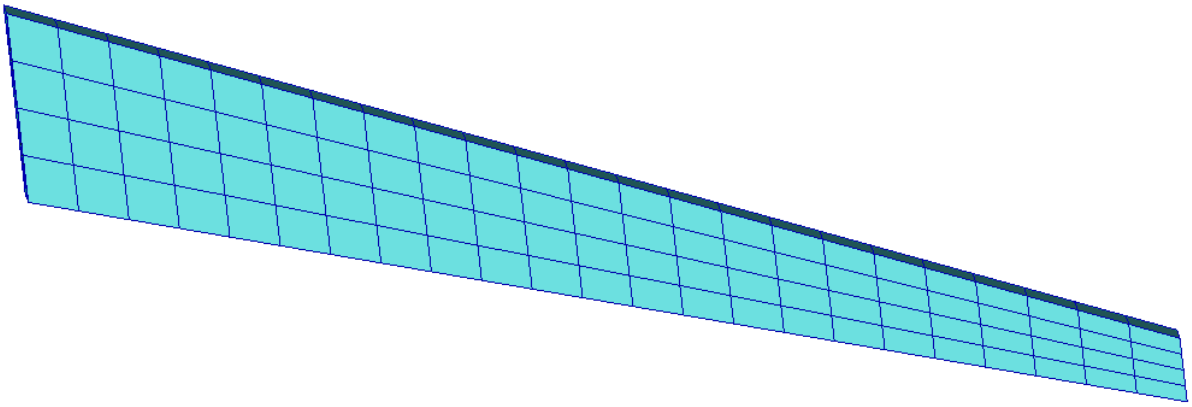

Fig. 17 The generated low-fidelity FEM for the LANN wing.

The two FEMs were generated to determine if the fidelity of the FEM was a factor in how much the mesh deformation algorithm affected the convergence of both the FSI simulations and the coupled-adjoint equations.

Table 2 Convergence comparison of LANN wing FSI simulations with a high-fidelity FEM.

Mesh deformation technique	Maximum structural displacement delta from previous iteration				
	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
RBF 1000 points	8.513e-02	2.142e-02	5.065e-03	1.367e-03	3.076e-04
RBF 4000 points	8.513e-02	2.185e-02	5.322e-03	1.456e-03	3.402e-04
RBF All points	8.513e-02	2.182e-02	5.304e-03	1.450e-03	3.378e-04

Table 3 Convergence comparison of LANN wing FSI simulations with a low-fidelity FEM.

Mesh deformation technique	Maximum structural displacement delta from previous iteration					
	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6
RBF 1000 points	6.787e-02	2.564e-02	9.954e-03	3.798e-03	1.475e-03	5.695e-04
RBF 4000 points	6.787e-02	2.640e-02	1.055e-02	4.148e-03	1.656e-03	6.541e-04
RBF All points	6.787e-02	2.638e-02	1.053e-02	4.138e-03	1.649e-03	6.495e-04

The convergence comparison of the FSI simulations reported in Table 2 and Table 3 reveal that the mesh deformation algorithm has a minimal effect on the convergence regardless of the fidelity of the CSM mesh. All the high-fidelity simulations converged in 5 iterations while all the low-fidelity simulations converged in 6 iterations. This also demonstrates the unsurprising result that the fidelity of the FEM will have an effect on the end result of the FSI simulation and how fast it converges. The key point of interest for this research is whether the fidelity of the FEM affects the convergence of the coupled-adjoint.

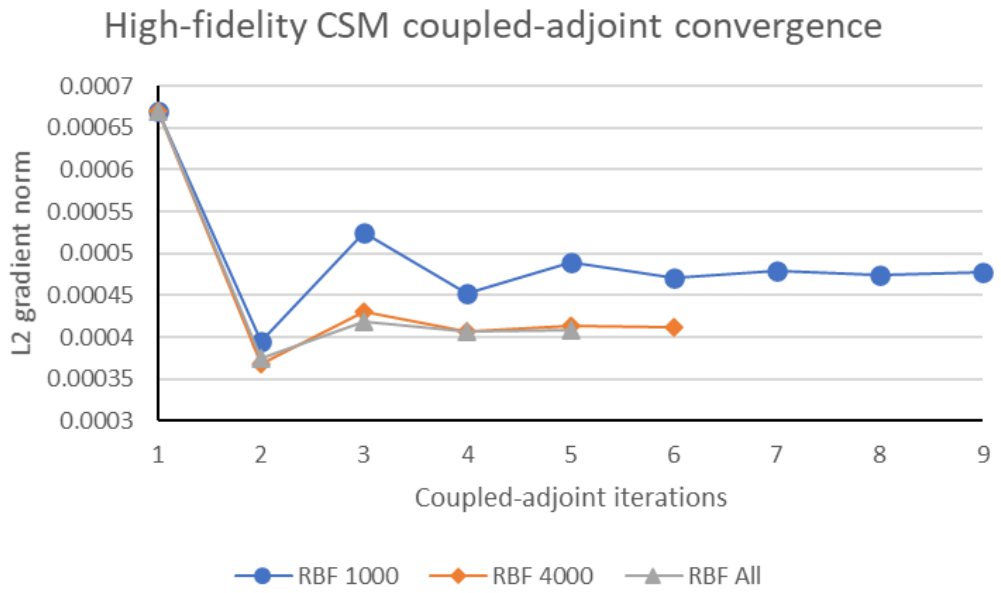

Fig. 18 The convergence of the coupled-adjoint with different mesh deformation algorithms with a high-fidelity mesh.

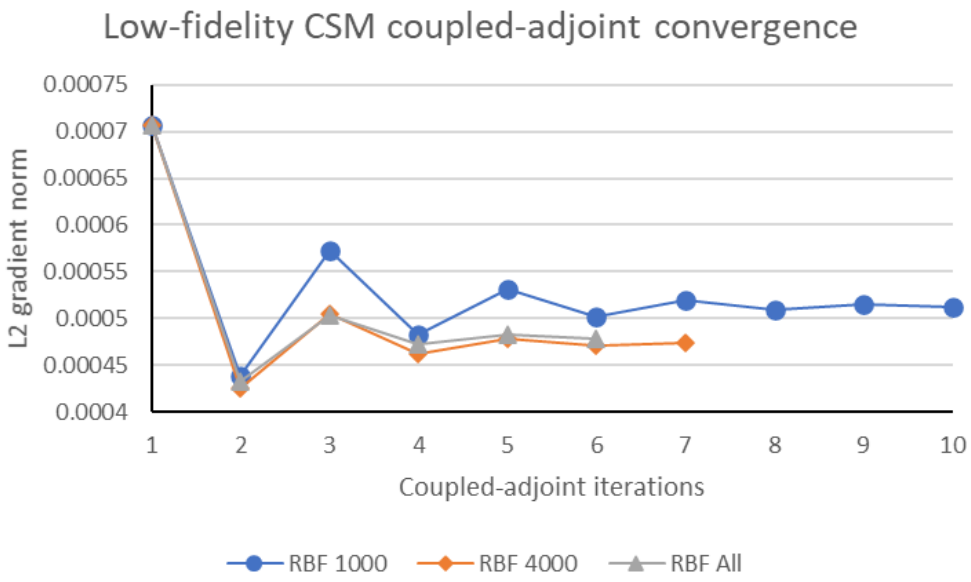

Fig. 19 The convergence of the coupled-adjoint with different mesh deformation algorithms with a low-fidelity mesh.

Figures 18 and 19 show that increasing the fidelity of the CSM mesh does not affect how the mesh deformation algorithm affects the convergence of the coupled-adjoint. The main impact of a higher fidelity CSM mesh is the accuracy of the final FSI solution and an accompanying increase in the FSI simulation's rate of convergence. This increased accuracy and rate of convergence occurs because the force on the surface of the CSM mesh after interpolation more accurately represents the force calculated by the CFD calculation. This is similar to the reason why the mesh deformation with an aggressive data-reduction algorithm lowers the rate of convergence of the coupled-adjoint. The aggressive data-reduction algorithm artificially lowers the fidelity of the CFD surface and thus some of the information in the vector field that is to be interpolated is lost in the process.

4.3.4 Hybrid Mesh Deformation Approach to Maximise Optimisation Process Efficiency

A simple solution to the convergence of the coupled-adjoint that comes from the data-reduction algorithm is available. Just use a different mesh deformation algorithm in the coupled-adjoint that uses all of the surface points. Mura *et al.* [76] made use of a hybrid mesh deformation strategy in a rigid aerodynamic adjoint-based optimisation to eliminate the mesh sensitivity. They used an iterative RBF method for deforming the mesh and they used Delaunay Graph Mapping (DGM) when calculating the mesh adjoint. They found that the non-consistent approach for the mesh sensitivity and mesh deformation had a negligible effect on the accuracy of the gradient and the overall optimised result. They used DGM as it is a very efficiency mesh deformation algorithm as the deformation matrix is sparse enough that it can be stored in memory. Thus reducing the mesh deformation and transposed mesh deformation problem down to a simple matrix vector multiplication problem. The DGM method is derived in Appendix F.

Two mesh deformation strategies were considered due to their use of all the surface points while also being direct mesh deformation methods. These strategies were Delaunay Graph Mapping (DGM) [67] and Inverse Distance Weighting (IDW) [68]. The IDW method is also derived in Appendix F. Iterative mesh deformation methods were not considered as the overriding purpose of this research is to make

the efficiency of an adjoint-based aeroelastic optimisation more efficient. By avoiding iterative methods, the transposed mesh deformation operation in the coupled-adjoint will be an efficient one.

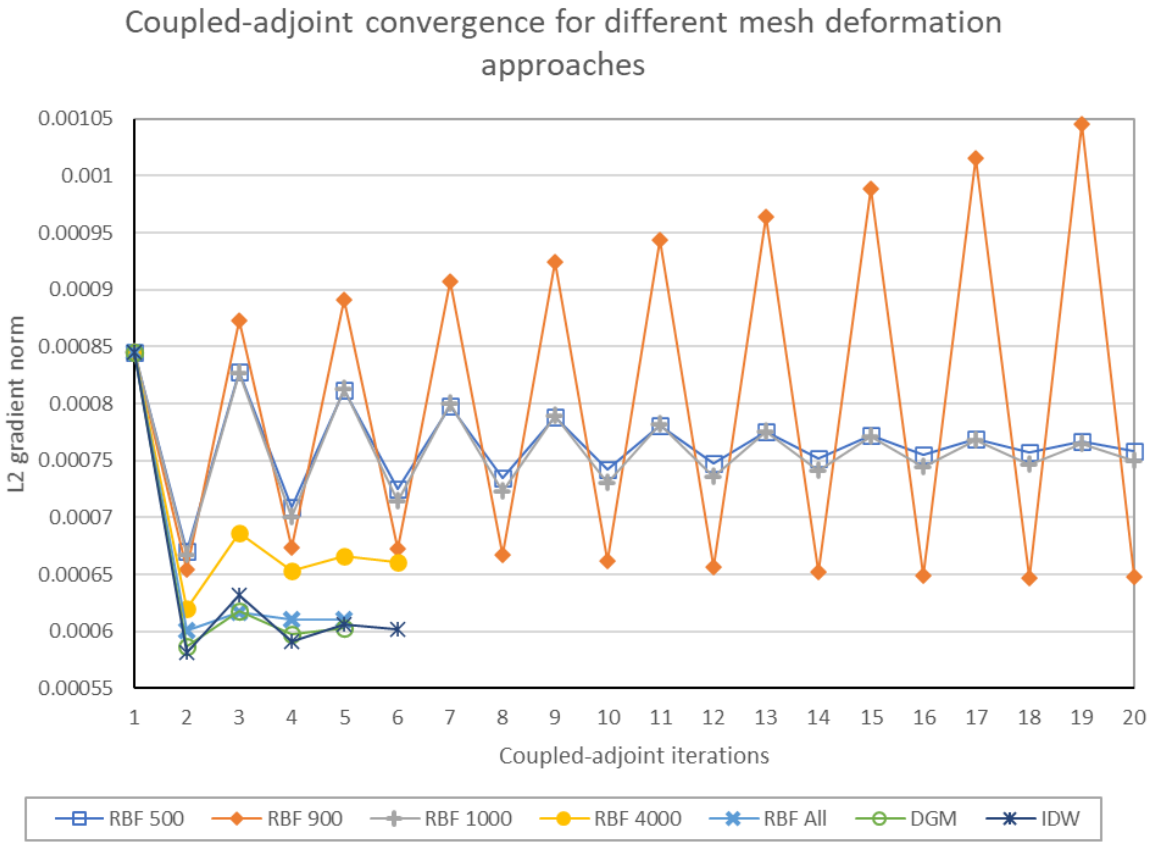

Fig. 20 Convergence comparison of the L2 gradient norm for different mesh deformation algorithms.

To decide which mesh deformation algorithm to use in the coupled-adjoint, a comparison of the rates of convergence for different mesh deformation algorithms was performed. Figure 20 is an extended version of Fig. 13 from chapter 4.3. The coupled-adjoint was deemed to be converged when the L2 gradient norm changed less than 1% relative to the previous iteration. The DGM algorithm and RBF without a data-reduction algorithm converged the fastest taking 5 iterations in total. The next fastest was IDW and RBF with a data-reduction algorithm that used 4000 surface points.

The other consideration to be looked at when deciding which mesh deformation algorithm to choose is the efficiency of the algorithm itself. Table 4 compares the efficiency of a number of mesh deformation operations. Each mesh deformation algorithm was performed using 1 core on the same displacement field on the LANN wing.

Table 4 Efficiency comparison of different mesh deformation algorithms.

Method	RBF	IDW	RBF	RBF	DGM
Surface points used	7420	7420	4000	1000	7420
Time taken (S)	39.6286	27.5769	9.79322	1.40692	0.21151

Table 4 reveals that DGM is the fastest mesh deformation algorithm by quite some distance. The time taken for the RBF algorithms to complete is strongly dependent on how aggressive the data-reduction algorithm is. It is important to note that the majority of time the time taken by the RBF algorithm is involved in inverting the matrix that produces the RBF interpolation coefficients. This means the major advantages of the RBF approach with a data-reduction get more pronounced as the size of CFD mesh increases as the matrix inversion time will not increase. The IDW algorithm was already the second slowest, even for a mesh with only 7200 surface nodes. The time taken by the IDW is proportional to the number of surface nodes so it will get considerably more expensive as the size of the CFD mesh increases. There are undoubtedly improvements that can be made to the author's implementation of IDW to improve its efficiency but there are no clear advantages to using IDW in the aeroelastic optimisation chain in the author's opinion.

In terms of efficiency, DGM is the clear winner. It is the fastest algorithm as shown in Table 4 and it enabled the coupled-adjoint to converge in the joint fastest number of iterations. Efficiency is not the only measure for whether an algorithm is a good choice however, it must also produce an accurate gradient.

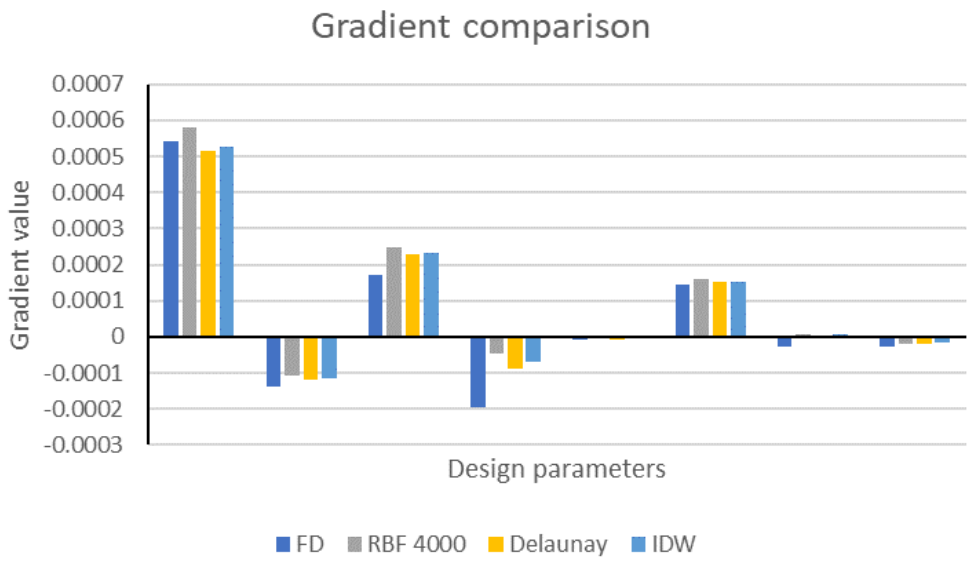

Fig. 21 Gradient comparison of two non-consistent mesh deformation approaches against the consistent approach.

Figure 21 demonstrates that both IDW and DGM produce gradients that closely match the gradient produced by the consistent RBF approach. Due to this similarity of the produced gradients from the coupled-adjoint and the already demonstrated more efficient nature of the DGM approach, the DGM method was chosen for use in the proposed hybrid mesh deformation procedure. The hybrid mesh deformation approach to coupled-adjoint optimisations is fully developed in the proceeding sections.

4.4 Satisfying the Pitching Moment Constraint

An aircraft must be trimmed in flight. An aircraft is said to be trimmed when the lift it produces is equal to its weight and the net pitching moment of the aircraft is zero. The fuselage, wing and

horizontal tail all contribute to the pitching moment. The pitching moment is typically addressed by rotating the horizontal tail to create a force to ensure there is no net moment. The rotation of the horizontal tail produces a drag of its own as the rotated shape of the horizontal tail will not be as aerodynamic as the un-rotated shape. This is known as trim drag, the drag that comes from deflecting control surfaces in flight to balance the aircraft.

Trim drag should be considered when designing an aircraft and when performing a performance optimisation. There are several methods used to address trim drag when optimising an aircraft. These involve constraining the pitching moment of the wing so that the tail would not need to produce a different force than the pre-optimised aircraft requires. This has been a very popular approach in adjoint-based aerodynamic shape optimisations [130] [131] [132] [133].

Another approach is the Direct Optimisation Strategy (DOS), this is where the rotation of the horizontal tail is added to the design parameters and letting the optimiser solve the trim problem simultaneously with the drag reduction problem. Chen *et al.* [79] employed this approach and they found that this approach produced better overall drag reductions when compared with alternative approaches of a trim penalty or a fixed wing pitching moment constraint.

A less commonly used approach is the Trim-Corrected Optimisation Strategy (TCOS). This strategy provides the FSI solver with trim parameters that it changes within the simulation so that the output solution from the solver is trimmed. Merle *et al*. [64] used this approach in an aerodynamic shape optimisation of a full aircraft configuration. They found that the drag reduction achieved was comparable with the DOS method, but the TCOS approach reduced the drag more significantly in the first few iterations, it also guarantees that each design iteration produces a feasible design. Abu-Zurayk *et al.* [45] extended this approach for use in an aerostructural adjoint optimisation. The TCOS approach means the optimiser does not have to know about the trim constraints as they will always be satisfied, enabling an unconstrained optimisation algorithm to be used.

To trim the aircraft within the FSI simulation, two loops are required. The first loop, or inner loop, iterates until the displacements for that configuration converge to within one millimetre and the flow density residual has reached its defined convergence criterion. The second loop, or trim loop, iterates until the pitching moment coefficient falls below a defined criterion. The trim loop deflects the control surface depending on the previous values of the pitching moment coefficient. It updates the deflection value according to the Broyden algorithm [122] which is described in chapter 4.3.2. In this work, the pitching moment coefficient was considered sufficiently small if it fell below 10^{-2} as this was the best precision TAU was able to achieve for converging the lift coefficient. It is only at the end of the last iteration of the trim loop that the flow residual and displacements need to be fully converged. To ensure a fast FSI simulation, the trim iterations that search for a control surface deflection angle that satisfies the trim constraint only converge the flow residual to 5×10^{-4}. Once the deflection angle has been found, the remaining iterations converge the residual to 10^{-6}.

4.4.1 Inclusion of Control Surface Deployment in Coupled-Adjoint Optimisation

In the TCOS and DOS optimisation cases mentioned prior, the pitching moment constraint has been constrained by parameterising the rotation of the horizontal tail. This is understandable as it is the way commercial aircraft trim themselves when in flight. Campos and Marques [134] assessed various methods for achieving a pitching moment constraint via control surface deployments. They determined that the best method for achieving trim was via the deflection of multiple control surfaces

rather than just using one. It would be useful for optimisation purposes to be able to deflect an arbitrary control surface. One reason for this is that it would allow a way of examining performance differences for differently sized control surfaces. Another is that the method could be applied to any provided aircraft configuration, without the control surface needing to be parameterised beforehand. A number of optimisations that use control surface deflection have been performed, these are often referred to as aeroservoelastic optimisations. Stanford [135] performed an aeroservoelastic optimisation on a wingbox of the CRM to minimise the mass of the wingbox given different manoeuvre loads. Aeroservoelastic optimisations involve finding the optimum actuator deflector values to optimise a specific objective. This project is interested in deflecting the control surfaces to achieve longitudinal trim while also optimising the shape of the aircraft to minimise drag. It would be possible to run an optimisation at each design iteration to determine the optimum control surface deflections that achieves trim at that design iteration. However, this would substantially increase the time of the overall optimisation and remove the ability to use the TCOS approach as the TCOS approach requires one trim variable for each trim constraint. To minimise the time required of the optimisation in this work, only one control surface is used for achieving trim.

The method used in this project takes advantage of the interpolation between the CSM and CFD coupling surface. Where the control surface is desired to be, a new domain is defined on the coupling surface. This domain defines the size of the control surface and the hinge line about which the control surface rotates.

$$\mathbf{a} = \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix} \tag{4.23}$$

$$\boldsymbol{R} = \begin{bmatrix} \cos\delta + a_x^2(1-\cos\delta) & a_x a_y(1-\cos\delta) - a_z\sin\delta & a_x a_z(1-\cos\delta) + a_y\sin\delta \\ a_y a_x(1-\cos\delta) + a_z\sin\delta & \cos\delta + a_y^2(1-\cos\delta) & a_y a_z(1-\cos\delta) - a_x\sin\delta \\ a_z a_x(1-\cos\delta) - a_y\sin\delta & a_x a_y(1-\cos\delta) + a_x\sin\delta & \cos\delta + a_z^2(1-\cos\delta) \end{bmatrix} \tag{4.24}$$

$$\boldsymbol{u}_{s_{CS}} = -\boldsymbol{X}_{s_{CS}}\boldsymbol{R}^T \tag{4.25}$$

The control surface points, defined by $\boldsymbol{X}_{s_{CS}}$ are simply multiplied by the negative transpose of the rotation matrix. This procedure outputs the displacements of the control surface points caused by a rotation of angle δ. A control surface deflection can then be input onto the mesh by adding the displacements caused by the deflection with the displacements caused by aerodynamic forces. The sum of these deflections is interpolated from the CSM surface onto the CFD surface. With blending between the control surface domain and adjacent domains turned off, this method provides a basic representation of a deflected control surface.

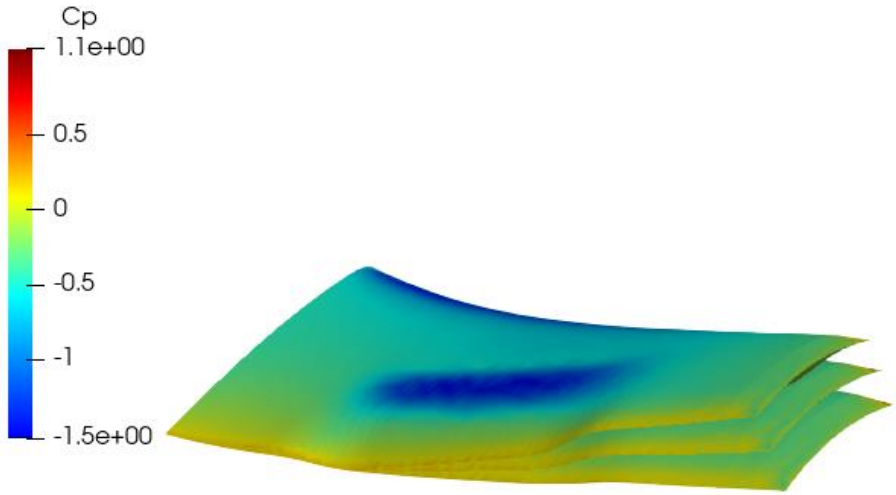

Fig. 22 The aeroelastically converged LANN wing with its flap deflected at 5°, 10° and 15°.

This approach to control surface deflection does not capture the airgaps that would appear between the control surface and adjacent surfaces during a real deflection. To fully capture the effects of a control surface deployment, a new mesh would have to be generated, or the Chimera technique [136] could be employed. These techniques would involve a different number of mesh points for a simulation with no control surface deflection and a simulation with a deflection. The changing number of mesh points make these techniques unsuitable for an adjoint optimisation [116]. With this limitation in mind, the method used should be limited to situations where either the deflections are small or where there are not surfaces adjacent to the deflected control surface e.g. elevators. Alternatively, a morphing wing could be modelled via this approach. Airgaps between surfaces adjacent to the deflected control surface do not occur with morphing wings. The deflection of a control surface on a morphing wing can be modelled by appropriately blending the displacements at the interfaces of the control surface. For this project, the elevators on the horizontal tail are used to trim the aircraft at each design iteration.

4.4.2 Broyden Algorithm

The Broyden algorithm is a member of the quasi-Newton family of methods used for solving $\mathbf{F}(\boldsymbol{x}) = \mathbf{0}$. The functional $\mathbf{F}(\boldsymbol{x})$ could be a set of functions but for this project it is only one function, and that function is the pitching moment constraint. Newton's method involves computing the Jacobian of the function at each iteration, and then using that Jacobian to determine the next estimation of $\boldsymbol{x}$.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4.26}$$

With multiple functions to solve, Newton's method becomes:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{J}^{-1}\boldsymbol{F}(\boldsymbol{x}_n) \tag{4.27}$$

Computing the Jacobian can be an expensive operation however, Broyden's idea was to compute the Jacobian only once at the first iteration and after that, update the Jacobian with the following equation.

$$\boldsymbol{J}_n = \boldsymbol{J}_{n-1} + \frac{\boldsymbol{F}(\boldsymbol{x}_n) - \boldsymbol{F}(\boldsymbol{x}_{n-1}) - \boldsymbol{J}^{-1}(\boldsymbol{x}_n - \boldsymbol{x}_{n-1})}{\|\boldsymbol{x}_n\|^2}(\boldsymbol{x}_n - \boldsymbol{x}_{n-1})^T \tag{4.28}$$

The Jacobian is then inverted and substituted back into Eq. (4.27) to find the next value or values of $\boldsymbol{x}$.

The Broyden algorithm is re-evaluated after each converged inner loop iteration of an FSI simulation to determine the next deflection angle for the control surface. After the Broyden algorithm finds the rotation angle of the control surface which satisfies the pitching moment constraint, one last inner loop is completed. The converged flow solution of the last iteration is then input as the starting condition for the aeroelastic coupled-adjoint.

4.4.3 Gradient of the TCOS Method when employing a Control Surface for Trim

The gradient equation that was derived in the previous sections is true only for the case of a simulation at a constant angle of attack. The FSI simulations in this research project have been trimmed in the loop. There are two implicit trim parameters, one for satisfying the lift constraint and one for satisfying the pitching moment constraint. These trim parameters are adjusted during the FSI simulation to ensure that the final result occurs at trim. Trim means that the target lift-coefficient C_L is reached while the pitching moment C_{my} is zero. The trim parameters are the deflection angle of the horizontal tail δ and the angle of attack α. The full dependencies of the terms in the drag gradient are defined below:

$$I = I\big(\boldsymbol{W}(\boldsymbol{D},\alpha), \boldsymbol{X}_\Delta(\boldsymbol{D},\delta), \alpha(\boldsymbol{D})\big) \tag{4.29}$$

$$\boldsymbol{R}_a = \boldsymbol{R}_a\big(\boldsymbol{W}(\boldsymbol{D},\alpha), \boldsymbol{X}_\Delta(\boldsymbol{D},\delta), \alpha(\boldsymbol{D})\big) \tag{4.30}$$

$$\boldsymbol{R}_s = \boldsymbol{R}_s\left(\boldsymbol{F}_S\big(\boldsymbol{W}(\boldsymbol{D},\alpha), \boldsymbol{X}_\Delta(\boldsymbol{D},\delta)\big), \boldsymbol{u}_S(\boldsymbol{D})\right) \tag{4.31}$$

These extra terms obviously make a difference to value of the gradient.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{D}} = \left.\frac{\partial \mathcal{L}}{\partial \boldsymbol{D}}\right|_{\alpha,\delta=const}$$
$$+\left(\left(\frac{\partial I}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_a^T\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}} + \boldsymbol{\lambda}_s^T\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{W}}\right)\frac{\partial \boldsymbol{W}}{\partial \alpha} + \frac{\partial I}{\partial \alpha} + \boldsymbol{\lambda}_a^T\frac{\partial \boldsymbol{R}_a}{\partial \alpha}\right)\frac{d\alpha}{d\boldsymbol{D}}$$
$$+\left(\left(\frac{\partial I}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T\frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T\frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}}\frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right)\frac{\partial \boldsymbol{X}_\Delta}{\partial \delta}\right)\frac{d\delta}{d\boldsymbol{D}} \tag{4.32}$$

A useful observation can be made about Eq. (4.32). The term that multiplies $\frac{\partial \boldsymbol{W}}{\partial \alpha}$ is equal to zero. Also, as α and δ are one-dimensional, the terms that multiply it must also be one-dimensional. The values $\boldsymbol{\lambda}_a^T$ and $\boldsymbol{\lambda}_s^T$ are already known as they have been calculated while finding $\left.\frac{\partial \mathcal{L}}{\partial \boldsymbol{D}}\right|_{\alpha,\delta=const}$. The unknown terms in the brackets are $\frac{\partial I}{\partial \alpha}$, $\frac{\partial \boldsymbol{R}_a}{\partial \alpha}$ and $\frac{\partial \boldsymbol{X}_\Delta}{\partial \delta}$. The terms $\frac{\partial I}{\partial \alpha}$, $\frac{\partial \boldsymbol{R}_a}{\partial \alpha}$ are available from the TAU solver so they do not need to be derived. The sensitivity of the mesh to the horizontal tail deflection angle does need to be derived, however.

$$\frac{\partial \boldsymbol{X}_\Delta}{\partial \delta} = \frac{\partial \boldsymbol{X}_\Delta}{\partial \boldsymbol{u}_{surf}}\frac{\partial \boldsymbol{u}_{surf}}{\partial \boldsymbol{u}_S}\frac{\partial \boldsymbol{u}_S}{\partial \delta} \tag{4.33}$$

The only new term here is $\frac{\partial \boldsymbol{u}_S}{\partial \delta}$.

$$\frac{\partial \boldsymbol{u}_S}{\partial \delta} = \frac{\partial \boldsymbol{u}_S}{\partial \boldsymbol{u}_{s_{CS}}} \frac{\partial \boldsymbol{u}_{s_{CS}}}{\partial \delta} \tag{4.34}$$

The matrix $\frac{\partial \boldsymbol{u}_S}{\partial \boldsymbol{u}_{s_{CS}}}$ is a sparse matrix that consists of ones and zeros as $\boldsymbol{u}_{s_{CS}}$ is simply a subset of $\boldsymbol{u}_S$. The control surface deflections are calculated by multiplying a rotation matrix against the control surface's CSM surface coordinates as derived in chapter 4.3.1.

$$\frac{\partial \boldsymbol{R}}{\partial \delta} = \begin{bmatrix} -sin\,\delta + a_x^2\,sin\,\delta & a_x a_y\,sin\,\delta - a_z\,cos\,\delta & a_x a_z\,sin\,\delta + a_y\,cos\,\delta \\ a_y a_x\,sin\,\delta + a_z\,cos\,\delta & -sin\,\delta + a_y^2\,sin\,\delta & a_y a_z\,sin\,\delta - a_x\,cos\,\delta \\ a_z a_x\,sin\,\delta - a_z\,cos\,\delta & a_x a_y\,sin\,\delta + a_x\,cos\,\delta & -sin\,\delta + a_z^2\,sin\,\delta \end{bmatrix} \tag{4.35}$$

$$\boldsymbol{u}_{s_{CS}} = -\boldsymbol{X}_{s_{CS}} \boldsymbol{R}^T \tag{4.36}$$

$$\frac{\partial \boldsymbol{u}_{s_{CS}}}{\partial \delta} = -\boldsymbol{X}_{s_{CS}} \left[\frac{\partial \boldsymbol{R}}{\partial \delta}\right]^T \tag{4.37}$$

Equation (4.131) provides the sensitivity of the translational displacements on the structural mesh to the control surface's deflection. The sensitivity of the rotational displacements to the deflection is simply:

$$\frac{\partial \boldsymbol{u}_{s_{CS}}}{\partial \delta} = \boldsymbol{a} sec^2 \delta \tag{4.38}$$

Now all the partial derivative terms have been found, a way to deal with the trim parameters sensitivity to the design parameters is also needed. This can be done without resorting to finite differences. Instead, the fact that the gradients of both the pitching moment and lift coefficient are zero is taken advantage of.

$$-\frac{\partial \mathcal{L}_{C_L}}{\partial \boldsymbol{D}}\bigg|_{\alpha,\delta=const}$$

$$= \left(\frac{\partial I_{C_L}}{\partial \alpha} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \alpha} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\alpha}\right) \frac{d\alpha}{d\boldsymbol{D}}$$

$$+ \left(\left(\frac{\partial I_{C_L}}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{X}_\Delta}{\partial \delta}\right) \frac{d\delta}{d\boldsymbol{D}} \tag{4.39}$$

A similar gradient equation can be formed for the pitching moment.

$$-\frac{\partial \mathcal{L}_{C_{my}}}{\partial \boldsymbol{D}}\bigg|_{\alpha,\delta=const}$$

$$= \left(\frac{\partial I_{C_{my}}}{\partial \alpha} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \alpha} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\alpha}\right) \frac{d\alpha}{d\boldsymbol{D}}$$

$$+ \left(\left(\frac{\partial I_{C_{my}}}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_a^T \frac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{X}_\Delta} + \boldsymbol{\lambda}_s^T \frac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{F}_{surf}} \frac{\partial \boldsymbol{F}_{surf}}{\partial \boldsymbol{X}_\Delta}\right) \frac{\partial \boldsymbol{X}_\Delta}{\partial \delta}\right) \frac{d\delta}{d\boldsymbol{D}} \tag{4.40}$$

The coupled-adjoint procedure is performed for both the lift gradient and pitching moment gradient at constant alpha and horizontal tail deflection angle. The gradients output from this procedure allows a simultaneous equation to be setup. The simultaneous equation is solved to obtain $\frac{d\alpha}{d\boldsymbol{D}}$ and $\frac{d\delta}{d\boldsymbol{D}}$.

$$-\frac{\partial \mathcal{L}_{C_L}}{\partial \boldsymbol{D}}\bigg|_{\alpha,\delta=const} = A_{C_L}\frac{d\alpha}{d\boldsymbol{D}} + B_{C_L}\frac{d\delta}{d\boldsymbol{D}} \tag{4.41}$$

$$-\frac{\partial \mathcal{L}_{C_{my}}}{\partial \boldsymbol{D}}\bigg|_{\alpha,\delta=const} = A_{C_{my}}\frac{d\alpha}{d\boldsymbol{D}} + B_{C_{my}}\frac{d\delta}{d\boldsymbol{D}} \tag{4.42}$$

$$\frac{d\alpha}{d\boldsymbol{D}} = \frac{\left(\frac{B_{C_{my}}}{A_{C_{my}}}\frac{\frac{\partial \mathcal{L}_{C_L}}{\partial \boldsymbol{D}}\Big|_{\alpha,\delta=const}}{B_{C_L}} - \frac{\frac{\partial \mathcal{L}_{C_{my}}}{\partial \boldsymbol{D}}\Big|_{\alpha,\delta=const}}{A_{C_{my}}}\right)}{\left(1-\frac{B_{C_{my}}}{A_{C_{my}}}\frac{A_{C_L}}{B_{C_L}}\right)} \tag{4.43}$$

$$\frac{d\delta}{d\boldsymbol{D}} = \frac{\left(\frac{A_{C_{my}}}{B_{C_{my}}}\frac{\frac{\partial \mathcal{L}_{C_L}}{\partial \boldsymbol{D}}\Big|_{\alpha,\delta=const}}{A_{C_L}} - \frac{\frac{\partial \mathcal{L}_{C_{my}}}{\partial \boldsymbol{D}}\Big|_{\alpha,\delta=const}}{B_{C_{my}}}\right)}{\left(1-\frac{A_{C_{my}}}{B_{C_{my}}}\frac{B_{C_L}}{A_{C_L}}\right)} \tag{4.44}$$

Every term in the drag gradient is now known and the gradient can be found. It is worth reiterating at this point the benefit of trimming within the FSI loop. This procedure allows the use of an unconstrained optimisation algorithm. This enables the local minimum to be found faster by the optimiser and it ensures that each design returned by the optimiser is feasible. This ties in with the use of the surrogate CAD model parameterisation, no volume constraints need to be applied as that constraint is satisfied within the CAD engine ensuring a fully feasible design space. This optimisation tool can be more directly integrated into the design process due to these features, making the added complexity of the optimisation process worth it.

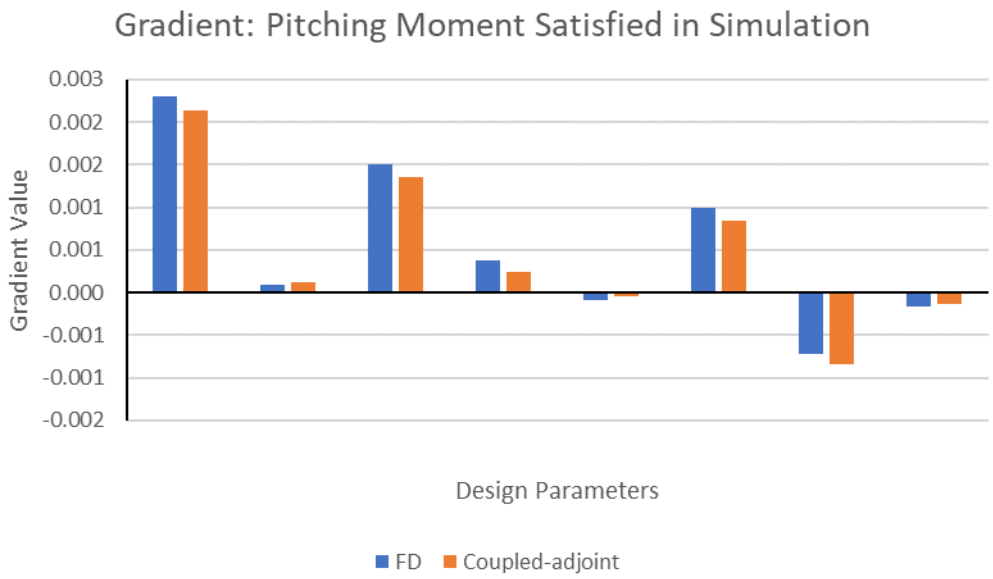

Fig. 23 Verification of coupled-adjoint drag gradient with pitching moment held constant.

Figure 23 shows the output of a coupled-adjoint gradient for the drag of the LANN wing where the gradient has been corrected for a constant pitching moment simulation while the angle of attack has been held constant. The numerical experiment was performed with flow conditions at Mach 0.8 while the angle of attack was held at 2°.

5. Application of Hybrid Mesh Deformation Strategy to Aeroelastic Optimisations

Two aeroelastic optimisations are presented in this chapter. The first optimisation is performed on a wing only case, the LANN wing [30]. The LANN wing is a moderate aspect ratio wing that was designed in the 1980s to study the effects of unsteady pressures at transonic speeds. The second optimisation is performed on the XRF1 aircraft. The XRF1 is a full aircraft model created by Airbus that is intended to represent a modern commercial aircraft. It is a twin engine aircraft and multiple coupled-adjoint optimisations have already been performed on it [43], [44]. In both optimisations, the hybrid mesh deformation procedure described in the previous chapter are used.

5.1 Lift Constrained Aeroelastic Optimisation of the LANN Wing

The first optimisation was performed on the LANN wing. The actual design condition for the wing is intended to cause oscillation of the wing in the pitching direction. This occurs at a Mach number of 0.82 and at a lift coefficient of 0.53. In fact, simulating the wing with Euler equations in TAU showed that convergence behaviour of this wing was poor in the transonic region, most likely due to unsteady effects. Due to this behaviour, a design point at Mach 0.73 and a lift coefficient of 0.15 was chosen.

Eight design parameters were generated for this model, two adjusted the spanwise twist profile and the other six controlled the camber profile along the wing.

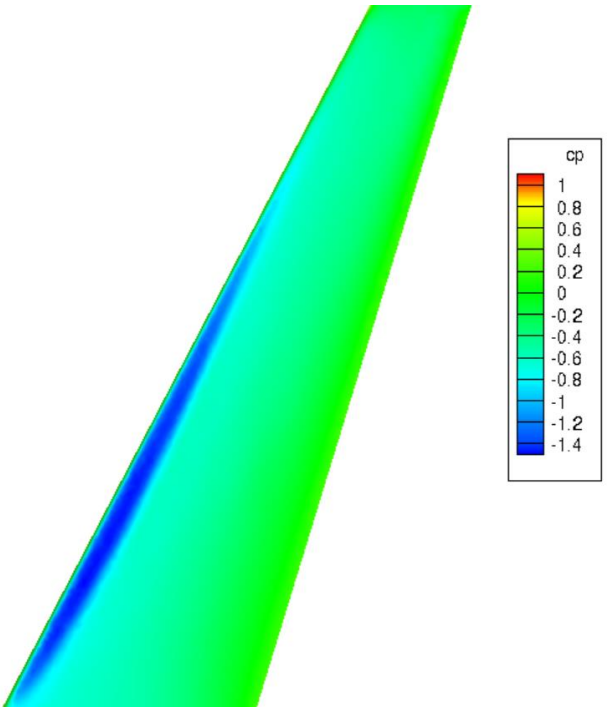

Fig. 24 The pressure distribution on the baseline configuration of the LANN wing.

It can be seen from Fig. 24 that the chosen design point is not one that the baseline wing was designed for. There is a large shock wave along over half the span near the leading edge of the wing. The shock will cause a significant amount of wave-drag and by extension, the optimiser will be able to significantly reduce the drag.

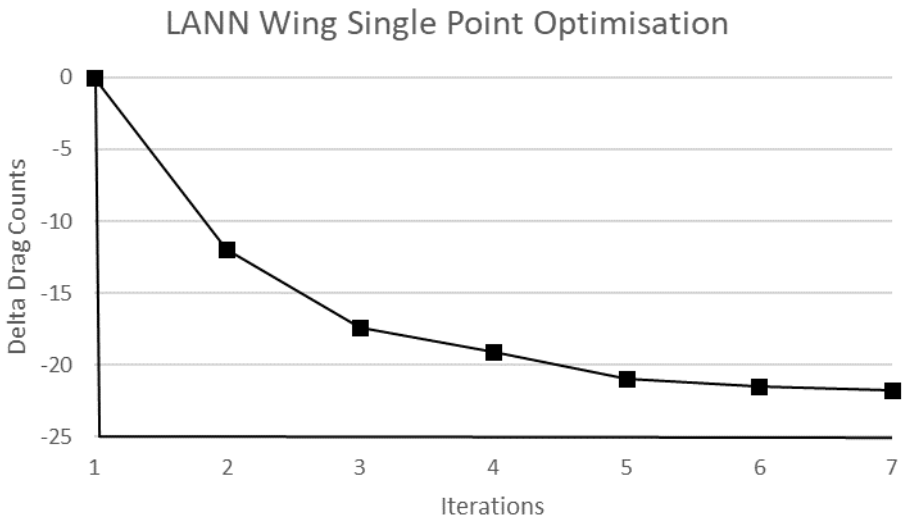

Fig. 25 The drag history of the LANN wing in the aeroelastic optimisation.

The LANN wing was designed to investigate flutter, minimising drag was not a consideration. In fact, the flight condition that the LANN wing is being optimised for in this instance was not considered during the design of the wing. Therefore, in the transonic region of flight a significant amount of wave drag is unsurprising. As a result of this design, there is a large amount of drag that can be removed by the optimiser despite there only being a few design parameters available to modify.

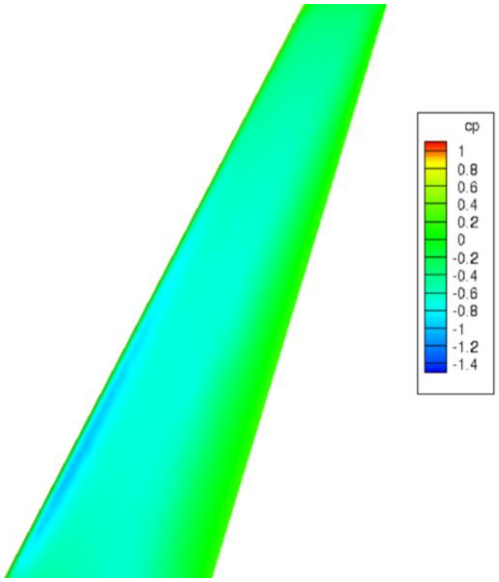

Fig. 26 The pressure distribution on the optimised configuration of the LANN wing.

The limited set of design parameters was able to adjust the shape of the wing so the strength of the shock across the inboard leading edge was significantly reduced. As shown in Table 5, the optimiser increased the camber and twist profile at all sections of the wing.

Table 5 Design parameter changes for the optimised LANN wing.

Design Variables	Twist Root	Twist Tip	Cb1a	Cb2a	Cb3a	Cb1b	Cb2b	Cb3b
Baseline Value	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Optimised Value	1.1682	0.3	0.7125	1.3697	0.0079	0.3285	0.5541	0.0090

The design variables in Table 5 are directly related to twist and camber values on the wing. The design variable changes in Table 5 show that the angle of incidence at the root was twisted up more substantially than the tip. The upward twisting at the root acts to push the centre of lift further inboard which lowers the magnitude of the pressure drop experienced at the outboard section of the wing. The camber was controlled at three sections by the remaining design parameters. The sections were defined on the wing near the root, near the middle, and near the tip of the wing. These are controlled by the parameters whose names start with Cb1, Cb2, and Cb3 respectively. The tip of the optimised LANN wing saw small camber changes only, while the middle and root of the wing had significant positive increases in its camber. Both the changes in tip angle and camber would act to increase the total lift produced if the angle of attack were to remain constant. To maintain the lift constraint, the angle of attack was decreased. The net effect of all these changes was to significantly reduce the strength of the shock that was present across the inboard leading edge thus significantly reducing the drag.

While this approach successfully reduced the shock strength and the drag, it increased the pitching moment. A feasible design must ensure that the longitudinal trim of the aircraft is achieved. In the optimisation presented in the following section, the XRF1 aircraft is optimised while ensuring that longitudinal trim is achieved at each design iteration.

5.2 Lift and Trim Constrained Aeroelastic Optimisation of the XRF1

The XRF1 is a full aircraft model that representing a typical configuration for a modern commercial long-range wide-bodied aircraft. The author was only able to obtain a mesh representing the flight shape of the XRF1 rather than its jig shape, as a result the FSI simulation does not accurately capture the aeroelastic response of the real XRF1. However, the aircraft model remains a viable candidate for verifying the effectiveness of the proposed hybrid mesh deformation approach to aeroelastic optimisations.

The wing of the XRF1 was parameterised with the surrogate CAD model method described in chapter 3.2. Using the surrogate CAD model approach, 74 parameters controlling the local camber and local twist along the span of the wing were produced. As the parameters only effect the local camber and twist, the wing volume will remain constant thus removing the need for an additional constraint in the optimisation. The flow condition selected for a single point optimisation was at a Mach number of 0.73 and a target lift coefficient of 0.15. To represent the structural model of the XRF1, a simple beam model was created in the open-source structural solver MYSTRAN [121]. MYSTRAN was integrated into the FlowSimulator environment to enable the aeroelastic response of the XRF1 to be captured. The trim parameters were the angle of attack and the deflection angle of the elevator on the

horizontal tail. The angle of attack was varied to satisfy the lift-constraint and the elevator was deflected to satisfy the pitching moment-constraint. The horizontal tail was modelled as a variable camber surface. This was represented by blending the interpolated displacements around the elevator hinge line thus producing a smoother deflected surface.

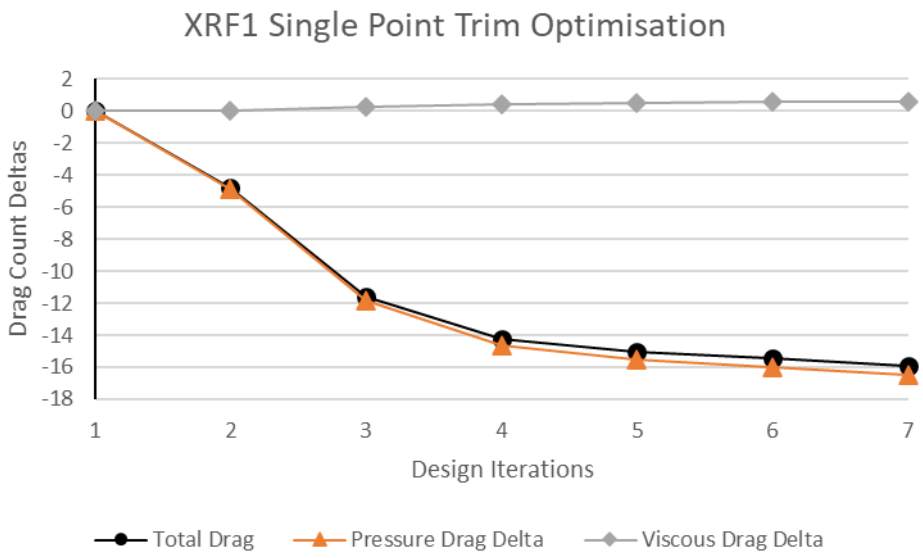

Fig. 27 The drag history of the XRF1 in the optimisation.

The TCOS optimisation of the XRF1 ran for 82 hours on 192 cores. The optimised configuration of the XRF1 produced 16 fewer drag counts than the baseline configuration. Figure 27 shows the reduction of the total drag through the optimisation and how the contributions of the viscous and pressure components changed through the optimisation. It reveals that the total drag was reduced entirely by a reduction in the pressure drag. In fact, the viscous drag actually increased on the optimised configuration albeit only by a single drag count.

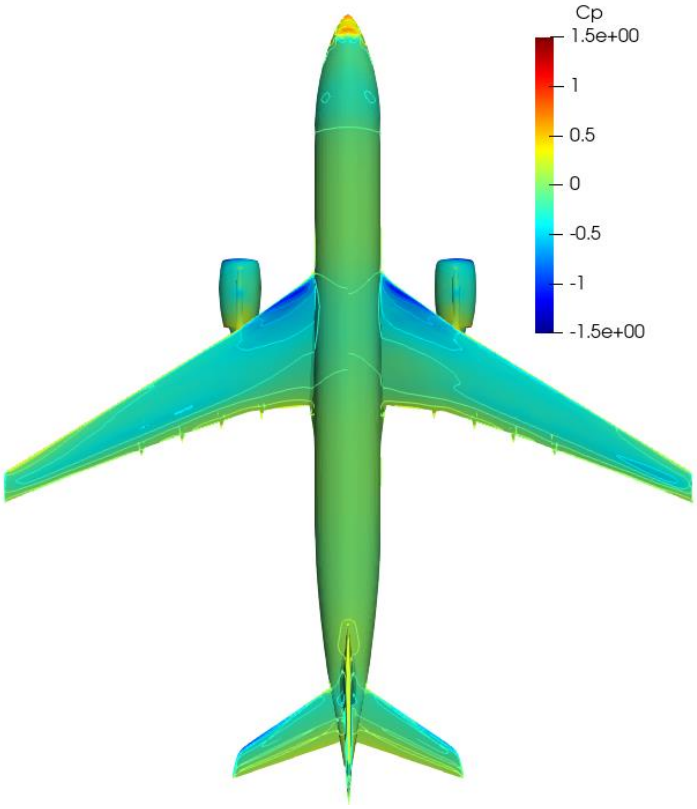

Fig. 28 Pressure coefficient comparison between the baseline and optimised configuration of the XRF1.

Figure 28 shows the pressure distribution of the baseline configuration on the left and the optimised configuration on the right. The pressure distribution on the optimised wing is not significantly different to the pressure distribution on the baseline wing particularly regarding the inboard leading-edge shock. This suggests that the reduction in drag on the optimised configuration has not been achieved primarily be a reduction in wave drag. The inboard region of the wing aft of the leading-edge shock does show the optimised configuration has a region of higher pressure here than the baseline configuration. This indicates that the drag reduction has primarily come from a larger pressure recovery at the inboard section of the wing.

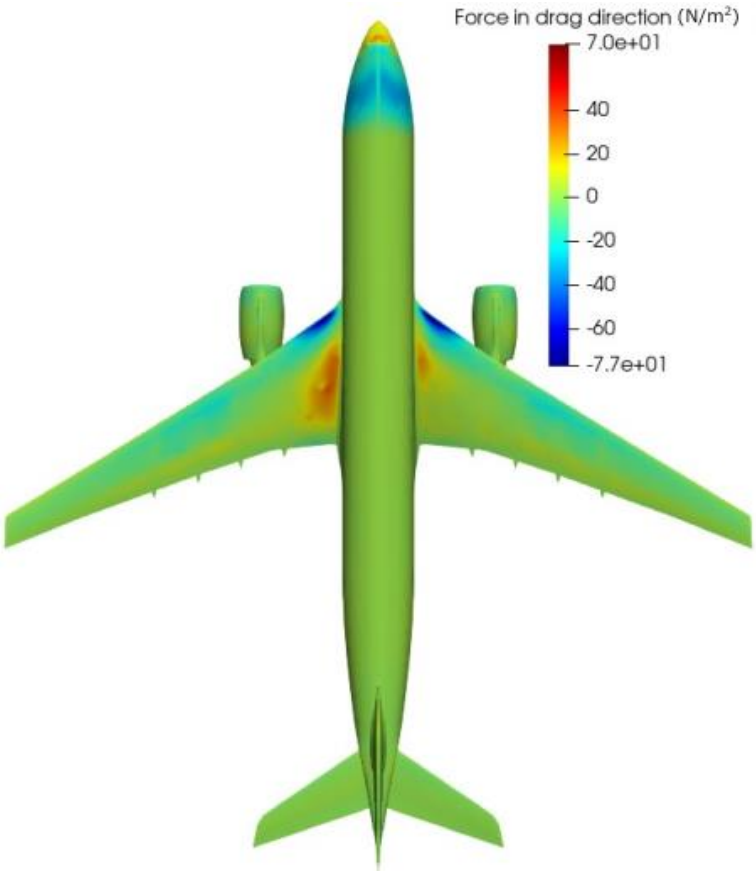

Fig. 29 Surface force per unit area in the drag direction on the baseline and optimised XRF1 configuration.

The comparison of the surface forces in the drag direction in Fig. 29 make it clear that the inboard region of the wing is the source of the drag reduction. In Fig. 29, a region of dark red signifies a region that is producing a large amount of drag while a blue region indicates a region with a local thrust contribution. The baseline configuration has a large inboard region with a local drag contribution, the optimised configuration has a significantly smaller region contributing to the local drag due to the improved pressure recovery.

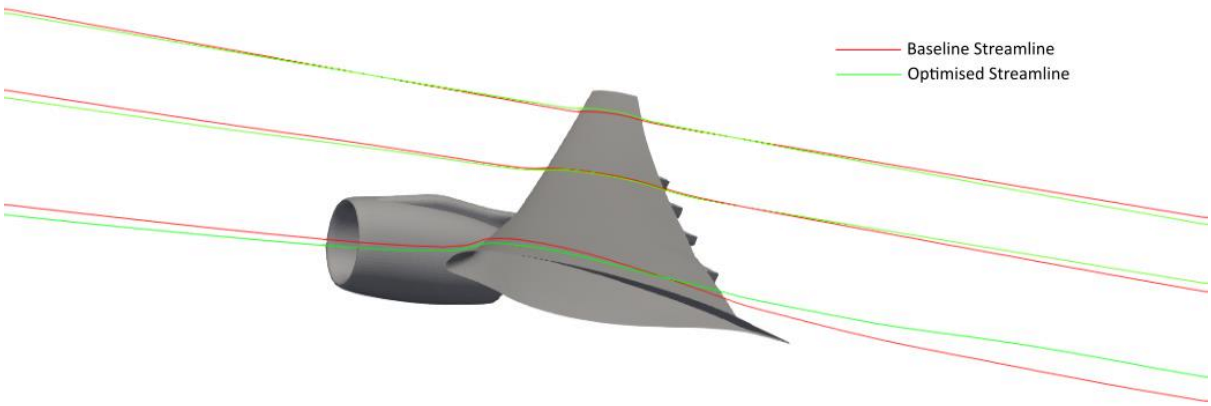

Fig. 30 Comparison of the baseline and optimised streamlines at three spanwise locations.

The streamline comparison of the baseline and optimised configuration of the XRF1 in Fig. 30 reveals the source of the heightened drag on the baseline configuration. The streamlines of the optimised configuration are shown in green, and the streamlines of the baseline configuration are shown in red. The velocity profile at the inboard section of the baseline configuration has a large amount of downwash present. A large amount of downwash in the flow field is accompanied by a large amount of induced drag. Downwash is always present in the flow field of a lift-generating object. The downwash indicates the presence of wing vortices that are shed from the wing when the sectional lift of adjacent regions is different. The strength of the wing vortex, and therefore the magnitude of the downwash, is proportional to the size of the difference in sectional lift. Typical explanations of induced drag tend to focus on the wing tips of an aircraft. However, induced drag will occur wherever there is a downward component in the velocity field of the flow. The lift force acts perpendicular to the direction of the flow. An increase in the downward component of the velocity field results in the aircraft's lift force having a rearward component. The size of the induced drag depends on two factors. The magnitude of lift being produced and the magnitude of the downwash at a given section. The magnitude of the lift being produced on a wing will be largest at its inboard section. This means that if the downwash is present at the inboard section, more induced drag will be generated. For the baseline XRF1 configuration at the target flow condition, there is both a large amount of lift being generated and a large amount of downwash. The result of this is a large amount of induced drag at the target flow condition.

Induced drag is caused by the lift-distribution alone. A well-known condition in aerodynamics is that the optimum lift distribution for minimising the induced drag on a wing with a constrained span is the elliptical lift distribution. The large amount of downwash in the baseline configuration reveals that a large proportion of the lift generation was being produced inboard. The optimiser has significantly changed the lift distribution on the optimised wing by pushing the centre of lift outboard, thus moving the lift distribution toward the elliptical lift distribution.

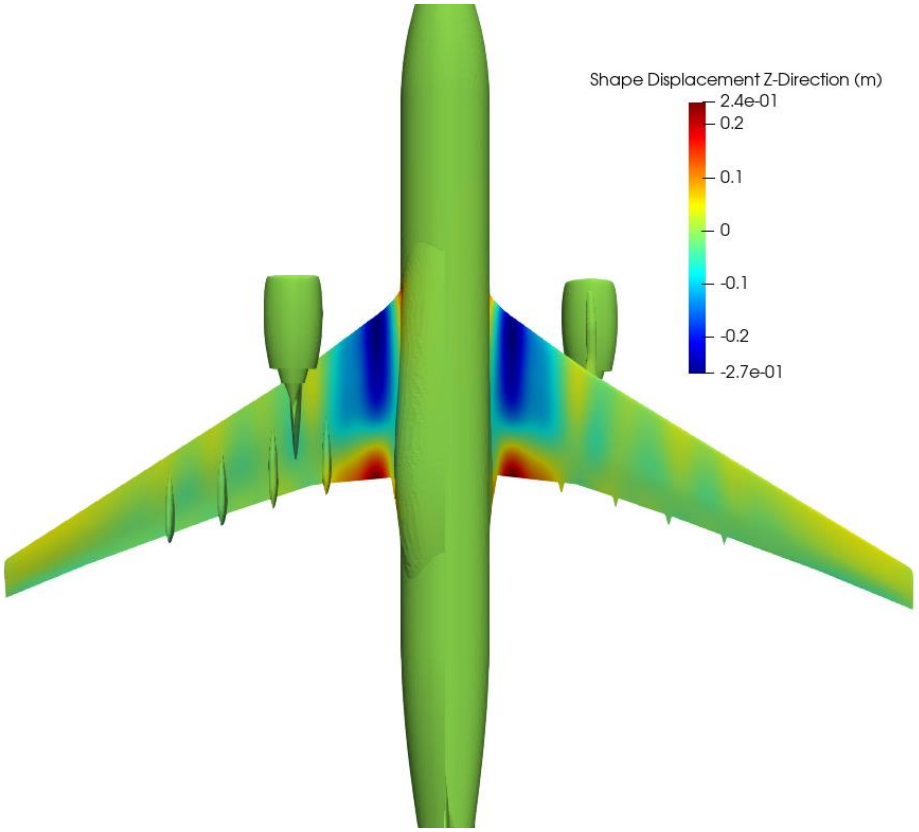

Fig. 31 The shape changes of the optimised jig shape in the z-direction.

Figure 31 shows how the wing shape of the optimised configuration was modified from the baseline configuration. Figure 31 shows that the jig shape has been displaced substantially at the inboard region in the z-direction. This amounts to a significant reduction of the optimised wing's angle of incidence. The optimiser did this to address the excess lift being generated inboard at this flight condition. The jig shape changes alone would amount to an overall reduction in lift and a modification to the pitching moment. To address this, the optimised trim variables were modified significantly. The baseline angle of attack was $2.702°$, the optimised value was $3.218°$. The overall effect of decreasing the lift produced inboard and increasing the angle of attack is a lift-distribution that is shifted outwards. The net-effect of the shape changes would have resulted in the overall pitching moment becoming more negative. To ensure longitudinal trim was achieved, the elevator deflection angle was changed from the baseline value of $-1.083°$ to the optimised value of $-3.151°$.

The optimisation of the XRF1 performed in this section made use of a trim-corrected optimisation strategy (TCOS) and the novel hybrid mesh deformation approach. The approach enabled a significant drag count reduction to be achieved in an efficient manner and has therefore shown the employed strategy to be an effective approach to performing aeroelastic optimisations. The RBF method with a data-reduction algorithm allowed the XRF1 with a complex aircraft geometry to be deformed robustly and efficiently. The use of the DGM method in the coupled-adjoint ensured that the block Gauss-Seidel approach to the coupled-adjoint would converge quickly while simultaneously allowing the transposed mesh deformation operation to be performed very efficiently. The TCOS was made possible by employing a variable camber parameterisation of the horizontal tail. The TCOS strategy ensured that the configuration produced at each iteration was feasible while also enabling the use of an unconstrained optimisation algorithm. The use of the TCOS approach, the coupled-adjoint and the

hybrid mesh deformation approach have allowed a complex optimisation with large numbers of design parameters to be performed efficiently.

5.3 Limitations of the Performed Optimisations

The TCOS approach to optimisation assumes that the gradients of the lift coefficient and pitching moment coefficient with respect to the design variables are equal to zero everywhere in the design space. The accuracy of the trim-corrected gradient depends on how precisely these coefficients reach their target values in each FSI run. Currently, the 2019 version of TAU has a limitation in that the minimum step size it can increment the angle of attack by is 10^{-5}. In the performed optimisations, this translated to the lift coefficient only converging to a precision of around 10^{-2}. Due to this, the convergence level specified for the pitching moment coefficient in the Broyden algorithm was also 10^{-2}. Some error will enter the gradient due to this but as is seen from the optimisations, the gradient was sufficiently accurate to enable a successful optimisation of both the LANN wing and XRF1 to be performed.

A second limitation was that the provided XRF1 mesh was in flight shape rather than an undeformed jig shape. Due to a number of factors, a jig shape of the XRF1 could not be obtained for either a meshed version or just the CAD shape. It was therefore decided that it would still be beneficial to perform an aeroelastic optimisation on the XRF1 to prove the viability of the hybrid mesh deformation approach. However, as the base shape is already deflected but the implemented FSI algorithm assumes it is not deflected, the resulting FSI simulation adds displacements to an already displaced mesh. As a result, the converged shape that comes out of the FSI simulation is not representative of how the XRF1 would actually behave. In effect, a different aircraft to the XRF1 has been optimised. By extension, the output shape of the optimisation is not an optimum shape for the XRF1. It is instead an optimised shape for an aircraft whose jig shape was the same as the XRF1's flight shape at a cruise condition. Even so, the optimisation of the XRF1 in this manner does prove the capability of the implemented algorithm which is the purpose of this thesis.

The last limitation was due to unforeseen circumstances of the last year which meant the optimisations could not be run on Airbus HPCs as was initially intended. Instead, the Centre for Modelling and Simulation (CFMS) was used as a backup and the author was only given a limited number of CPU hours. This meant there was only enough resources available for a single aeroelastic optimisation of the XRF1, instead of the desired multi-point optimisation.

6. Conclusions and Future Work

Aerostructural optimisation of aircraft will be an increasingly important aspect in the design of the next generation of aircraft. The strongly coupled effects between aerodynamics and structures must be considered early in the design process to reduce the time spent iterating between optimised internal structures and optimised aerodynamic shapes. The major finding of this thesis was the observation that the choice of mesh deformation algorithm has a significant impact on the convergence characteristics of the coupled-adjoint matrix. A solution was then sought that would enable the mesh deformation algorithm of choice to be used for FSI but also ensure strong convergence characteristics for the coupled-adjoint. The algorithm used in the FSI simulation was the RBF approach with a data reduction algorithm. The algorithm used in the coupled-adjoint loop was the DGM approach. Used together, these algorithms tackle the dual challenges of efficiently and robustly deforming geometrically complex meshes and also producing an accurate gradient from the coupled-adjoint procedure.

A second novelty was introduced to obtain the sensitivity of an arbitrarily sized control surface. This gave a flexible manner in which a control surface could be included in an adjoint-based aeroelastic optimisation. It also provided a convenient manner to employ the TCOS strategy, allowing the aircraft to be trimmed within the FSI simulation and thus enabling an unconstrained optimisation algorithm to govern the optimisation.

The novelties introduced in this work were then implemented into two optimisations. First, a lift-constrained optimisation of the medium aspect-ratio LANN wing was performed. Second, an optimisation with longitudinal trim enforced within the simulation was performed on the XRF1. Both optimisation cases successfully achieved significant drag reductions.

6.1 Conclusions

There are a number of conclusions reached in this thesis:

The major finding is that the mesh deformation algorithm strongly affects the convergence of the coupled-adjoint linear system. An algorithm that does not use the full displacement field on the surface when propagating the displacements into the volume mesh risks causing the coupled-adjoint to converge in a significantly slower time, or for it to not converge at all.

The second finding is that this problem can easily be circumvented by employing a non-consistent mesh deformation algorithm, such as the DGM, in the coupled-adjoint that does use the full displacement field on the surface. This dilutes the strength of off-diagonal elements across all the rows in the second block row of the coupled-adjoint matrix, thus improving its convergence characteristics.

The third finding is that the use of a non-consistent mesh deformation algorithm in the coupled-adjoint loop has only a negligible effect on the accuracy of the produced gradient. Gradients obtained in this manner can be used in aeroelastic optimisations to successfully increase the performance of an aircraft.

The use of an explicit mesh deformation algorithm like the DGM method in the coupled-adjoint means a mesh adjoint system does not need to be solved on each coupled-adjoint iteration. An implicit system like Linear Elasticity Analogy will ensure the coupled-adjoint matrix has good convergence

characteristics but it will increase the time of the coupled-adjoint loop as it is a large linear system that takes a long time to solve.

The direct rotation of a control surface on the CSM mesh in combination with a blended interpolation scheme between the coupling surfaces produces an easy way to represent a morphing surface. As the procedures are both analytical their sensitivities can also be found analytically thus making the parameterisation method a suitable candidate for use in adjoint-based optimisations. It is compatible with regular FEMs and beam models and their accompanying displacement and force interpolation methods.

6.2 Future Work

Two single point optimisations were performed in this work using the TCOS approach. Chapter 5.2 highlighted limitations that were present during this study, particularly regarding the accuracy of gradients produced when using the TCOS approach. In the 2019 version of TAU, the enforced precision of a lift constraint cannot be specified. It would be worthwhile to compare the efficiency of the TCOS approach with DOS approach for enforcing constraints. It is common to see optimisations in literature where the lift is constrained by the flow solver, but moment constraints are handled by the optimiser via the use of an algorithm such as SQP. An interesting question then is, when is it beneficial to enforce a constraint within the simulation and when is it not? What level of precision is required when enforcing a constraint in the solver for the gradient to still be sufficiently accurate? A follow-on question is, are there significant differences in convergence time between an optimisation that constrains no quantities in the solver, one that constrains some quantities in the solver and one that constrains all quantities in the solver? To address another limitation, a coupled-adjoint optimisation of a full aircraft configuration should be performed using its jig shape as the baseline configuration.

The horizontal tail of the XRF1 was modelled as a variable camber surface in this thesis. High-fidelity coupled-adjoint optimisations using these variable camber technologies are rare. One optimisation of interest would be to constrain the shape parameterisation of the aircraft so that these morphing devices could feasibly be placed into the wing. Perform a cruise condition multi-point optimisation to obtain a baseline shape, then perform secondary optimisations at off-design points using the morphing devices as parameters to investigate what drag savings these would achieve. A morphing technology such as the VCCTEF could be effectively parameterised by making use of the surrogate model approach described in chapter 3.2. An application such as Blender would be able to parameterise a wing with VCCTEF installed and capture the complex geometric changes that would occur. To make the integration of a Blender based parameterisation with an HPC cluster easier, a surrogate model of the Blender parameters could be made on a local computer using the method described in chapter 3.2 to produce a file with the POD parameters. This would remove the need for the Blender application to be installed on an HPC and it would make the sensitivity of the surface to the design parameters readily available.

An area that was discussed in this thesis but not implemented was the idea of including structural variables. Structural variables introduce a number of challenges due to the number of constraints that become involved, but their inclusion is essential if significant improvement of commercial aircraft is to be achieved. A specific investigation of interest would be to modify the FEM to fully represent the structure of the morphing wing devices and then perform an aerostructural optimisation to determine the true effect their inclusion has on the performance of a commercial aircraft.

Appendix A Gradient-Based Optimisation Algorithms

There are many gradient-based optimisation algorithms available to choose from. The simplest is the steepest descent method which just chooses the search direction to be the negative of the gradient. Another popular method to obtain the search direction is the conjugate gradient algorithm [137]. This is a simple modification to the steepest descent method, but the previously calculated gradients are taken into account in an attempt to make it converge faster. A method that improves the rate of convergence is Newton's method. Newton's method obtains the inverse of the Hessian matrix at each design iteration and multiplies it with the gradient to determine the search direction. The Hessian matrix is the derivative of the gradient with respect to the design parameters. Calculating the Hessian matrix at each design iteration is computationally expensive so more frequently, a method to obtain an approximation of the Hessian matrix is used. These methods fall under the Quasi-Newton family [26 – 28].

The category of algorithm used depends on whether the optimisation problem has constraints that the optimiser needs to handle. An aerodynamic drag optimisation will need to have constraints such as holding the lift constant, or otherwise the optimiser would just lower the lift to reduce the drag. Another common constraint regards the pitching moment. An optimised design that has reduced the drag would not be useful if the aircraft is so far off-balance that the control surface deflection required to balance it increases the drag above that of the original design. These constraints can either be handled directly by the optimiser, implicitly by the flow solver, or a mix of both.

In both constrained and unconstrained algorithms, a Hessian matrix is often employed to improve the optimisation's rate of convergence. If a Hessian matrix is required, a Quasi-Newton method will typically be used to obtain an approximation to it. The approximation to the Hessian gets built up after each iteration of the optimisation.

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \mathbf{B}_k^{-1} \nabla \boldsymbol{f}(\boldsymbol{x}_k) \qquad (A.1)$$

Equation (A.1) shows the general form of quasi-Newton methods. Where α_k is a line search parameter and $\mathbf{B}_k$ is an approximation to the Hessian. If α_k is 1 and $\mathbf{B}_k$ is the exact Hessian matrix, then Eq. (A.1) would give Newton's method for optimisation. Quasi-Newton methods typically lose the quadratic convergence of Newton's method but often superlinear convergence is achieved.

After the search direction has been found, the next task is to determine the step size to take in that direction. This is known as a line search. It is a one-dimensional search that attempts to find the minimum of the objective function with respect to the provided search direction. A popular approach to the line search is one that satisfies the Wolfe Conditions [141]. The Wolfe conditions are met when a sufficient decrease along with a curvature condition occur. These conditions can be met without being close to minimising the line search, showing there is not a one size fits all approach to optimisation. In the case where line searches are considerably more expensive than the gradient evaluation, a more suitable approach is the trust region strategy [141]. A model function is formed from information gathered from the objective function. The candidate step is then found by minimising the model function, which is typically a quadratic function.

Appendix B Derivation of the Navier-Stokes Equations

The integral form of the Navier-Stokes equations can be derived by considering the conservation of mass, momentum, and energy through an arbitrary control volume Ω. For these quantities to be conserved, the rate of change or each quantity must equal the flux of each quantity through the control volume.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{B.1}$$

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \tau + \boldsymbol{S}_M \tag{B.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} H) = \nabla \cdot (\tau \cdot \boldsymbol{u}) + \nabla \cdot (\kappa \nabla T) + \boldsymbol{S}_E \tag{B.3}$$

In a single integral form, these conservation equations are written as:

$$\frac{\partial}{\partial t} \int_{\Omega} \boldsymbol{W} d\Omega + \oint_{\partial \Omega} \{\mathbf{F}^c(\boldsymbol{W}) - \mathbf{F}^v(\boldsymbol{W})\} \cdot \boldsymbol{n} d(\partial \Omega) + \int_{\Omega} S(\boldsymbol{W}) d\Omega = 0 \tag{B.4}$$

The flow conditions in this work will always be a steady case and there will be no source terms. This leaves only the second integral term from Eq. (B.4). The boundary of the control volume Ω is denoted by $\partial \Omega$ with an outer normal direction of $\boldsymbol{n}$. The second integral in Eq. (B.4) is denoted by the residual term $\boldsymbol{R}$.

$$\boldsymbol{R}(\boldsymbol{W}, \boldsymbol{n}, \partial \Omega, \Omega) = 0 \tag{B.5}$$

$\boldsymbol{W}$ represents the flow state variables, these are the quantities that are conserved in the flow.

$$\boldsymbol{W} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix} \tag{B.6}$$

The density of the fluid is signified by ρ. The velocity in the x, y, z directions are u, v and w respectively. The variable E is the total energy within the control volume. The flux of quantities passing through the control volume are particularly important for CFD as these are used to determine whether the flow solution produced by the CFD solver is converged. There is a convective (inviscid) flux tensor and a viscous flux tensor.

$$\mathbf{F}^{\boldsymbol{c}} = \begin{bmatrix} \mathbf{F}_x^{\boldsymbol{c}} \\ \mathbf{F}_y^{\boldsymbol{c}} \\ \mathbf{F}_z^{\boldsymbol{c}} \end{bmatrix} \tag{B.7}$$

$$\mathbf{F}^{\boldsymbol{v}} = \begin{bmatrix} \mathbf{F}_x^{\boldsymbol{v}} \\ \mathbf{F}_y^{\boldsymbol{v}} \\ \mathbf{F}_z^{\boldsymbol{v}} \end{bmatrix} \tag{B.8}$$

Each element of the flux tensors in Eq. (B.7) and Eq. (B.8) is a vector.

$$\mathbf{F}_x^c = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho Hu \end{bmatrix} \tag{B.9}$$

$$\mathbf{F}_y^c = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \end{bmatrix} \tag{B.10}$$

$$\mathbf{F}_z^c = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho u^2 + p \\ \rho Hw \end{bmatrix} \tag{B.11}$$

$$\mathbf{F}_x^v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix} \tag{B.12}$$

$$\mathbf{F}_y^v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix} \tag{B.13}$$

$$\mathbf{F}_z^v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix} \tag{B.14}$$

The static pressure is represented by p, H represents the total enthalpy, τ_{ij} is the viscous stress tensor for a given direction at specific location and q_i is the heat transfer flux vector. The viscous stress tensors that arise due to viscous effects are given by:

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \boldsymbol{\lambda}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \tag{B.15}$$

$$\tau_{yy} = 2\mu\frac{\partial v}{\partial y} + \boldsymbol{\lambda}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \quad (B.16)$$

$$\tau_{zz} = 2\mu\frac{\partial w}{\partial z} + \boldsymbol{\lambda}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \quad (B.17)$$

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad (B.18)$$

$$\tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \quad (B.19)$$

$$\tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \quad (B.20)$$

The variable μ is the dynamic viscosity of the fluid and $\boldsymbol{\lambda}$ is the second coefficient of viscosity which can be computed by Stokes' hypothesis:

$$\boldsymbol{\lambda} = -\frac{2}{3}\mu \quad (B.21)$$

The dynamic viscosity can be calculated via Sutherland's law.

$$\mu = \mu_0\left(\frac{T}{T_0}\right)^{3/2}\frac{T_0 + \mathrm{S}}{T + \mathrm{S}} \quad (B.22)$$

The constants μ_0, T_0 and S (Sutherland's temperature constant) are constant values that are defined for each fluid. The heat transfer flux vector is calculated using the thermal conductivity relation.

$$q_x = -\kappa\frac{\partial T}{\partial x} \quad (B.23)$$

$$q_y = -\kappa\frac{\partial T}{\partial y} \quad (B.24)$$

$$q_z = -\kappa\frac{\partial T}{\partial z} \quad (B.25)$$

The variable κ is the coefficient of thermal conductivity and is obtained with the following relation.

$$\kappa = \mu\frac{C_p}{Pr} \quad (B.26)$$

The variable C_p is the specific heat at constant pressure and Pr is the Prandtl number. The total enthalpy H of the control volume is related to the total energy within the control volume E.

$$H = E + \frac{p}{\rho} \quad (B.27)$$

$$E = \frac{p}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2 + w^2) \quad (B.28)$$

The ratio of specific heats is γ and in air this is equal to 1.4. The final equation needed for closure of the Navier-Stokes equations is the ideal gas law. This relates the pressure of the gas to its density and temperature.

$$p = \rho RT \quad (B.29)$$

The variable R is the gas constant and in air it has a value of $287\ J\ Kg^{-1}K^{-1}$.

Reynolds Averaged Navier-Stokes Equations

The Navier-Stokes equations are nonlinear partial differential equations which at the present time cannot be solved analytically, meaning they must be solved numerically. Discretising the Navier-Stokes equations and attempting to solve them through numerical brute force is known as the Direct Numerical Solution (DNS) approach. This discretisation requires the entire range of spatial and time turbulence scales to be resolved. The smallest length and time scales are known as the Kolmogorov microscales [142]. In order to resolve these scales, a very dense mesh along with a very small time-step would be required. The Reynolds number to the third power is proportional to the total number of discrete mesh points (spatial resolution) multiplied by the time step (time resolution) that needs to be used. The Reynolds number of a flow over a commercial aircraft at cruise typically has a value of many millions. Parviz and Kim [143] estimated that a DNS simulation for a commercial aircraft using a teraflop computer would therefore take thousands of years.

To make the problem of aerodynamic simulation tractable, the turbulence must be modelled differently so that the spatial and time scales no longer have to be so small. There are a number of techniques available to do this, all with different levels of fidelity and therefore different levels of accuracy. At the high-fidelity end are detached eddy simulations (DES), large eddy simulations (LES) and Reynolds Averaged Navier-Stokes (RANS) simulations. RANS is the lowest fidelity of the three as it can only give time averaged values for the velocity field, whereas DES and LES resolve large scale eddies. DES and LES are produce more accurate results than RANS but they take a significantly longer time. For this reason, DES and LES are considered too expensive to be used to simulate the flow, meaning RANS is the aerodynamic model of choice.

The Reynolds' averaging method involves finding the mean value of a time dependent quantity over time.

$$\boldsymbol{W} = \overline{\boldsymbol{W}} + \boldsymbol{W}' \tag{B.30}$$

$$\overline{\boldsymbol{W}} = \frac{1}{\Delta T}\int_{\Delta t} \boldsymbol{W} dt \tag{B.31}$$

However, the Reynolds average would not take into account the compressibility effects that occur at transonic flow conditions, so the Favre average is used instead.

$$\boldsymbol{W} = \widetilde{\boldsymbol{W}} + \boldsymbol{W}'' \tag{B.32}$$

The instantaneous flow state variable vector $\boldsymbol{W}$ is decomposed into $\widetilde{\boldsymbol{W}}$, which is the mean Favre averaged value of the state variables, and $\boldsymbol{W}''$ which is the fluctuating part of Eq. (B.32).

$$\widetilde{\boldsymbol{W}} = \frac{1}{\bar{\rho}\Delta T}\int_{\Delta t} \rho \boldsymbol{W} dt \tag{B.33}$$

The variable $\bar{\rho}$ is the Reynolds averaged density. This averaging procedure is only an accurate approximation when there is little to no detachment of the flow, otherwise a higher fidelity scheme should be employed. Substituting the new Favre averaged state variables back into the governing equations creates two new unknowns. The first unknown is referred to as the Reynolds stresses, the second term is known as the turbulent heat fluxes. These are defined in Eq. (B.34) and Eq. (B.35), respectively.

$$\bar{\tau}_{ij} = -\overline{\rho u_i'' u_j''} \tag{B.34}$$

$$\bar{q}_i = -C_p \overline{\rho u_i'' T''} \tag{B.35}$$

There are six new unknown variables due to the introduction of Reynolds stresses. This creates a closure problem as there are more unknowns than equations. The Navier-Stokes equations represent the conservation of mass, momentum, and energy respectively. The Favre averaged conservation of mass equation looks almost the same as the original equation.

$$\frac{\partial \bar{\rho}}{\partial t} + \nabla \cdot (\bar{\rho} \tilde{\boldsymbol{u}}) = 0 \tag{B.36}$$

The Favre averaged conservation of mass equation can therefore be solved in the same way as the original, this does not hold true for the conservation of momentum or energy equations. Averaging of the conversation of momentum equation yields:

$$\frac{\partial (\bar{\rho} \tilde{\boldsymbol{u}})}{\partial t} + \nabla \cdot (\bar{\rho} \tilde{\boldsymbol{u}} \otimes \tilde{\boldsymbol{u}}) = -\nabla \bar{p} + \nabla \cdot \bar{\tau} - \nabla \cdot \overline{\rho \boldsymbol{u}'' \otimes \boldsymbol{u}''} \tag{B.37}$$

This modified conservation of momentum equation has an additional term that is not present in the original equation. The Reynolds stress tensor has appeared and must somehow be determined. The Reynolds stress tensor is symmetric meaning that six unknowns have been added to the three-dimensional conservation of momentum equations. This is the cause of the turbulence closure problem.

Similarly, the Favre averaged conservation of energy equation yields two new terms that are not present in the original.

$$\frac{\partial (\bar{\rho} \tilde{E})}{\partial t} + \nabla \cdot (\bar{\rho} \tilde{\boldsymbol{u}} \tilde{H}) = \nabla \cdot \left[\bar{\tau} \cdot \tilde{\boldsymbol{u}} + \kappa \nabla \tilde{T} + \overline{\rho \boldsymbol{u}'' \otimes \boldsymbol{u}''} \cdot \tilde{\boldsymbol{u}} - \overline{\rho \boldsymbol{u}'' \mathrm{h}''} \right] \tag{B.38}$$

The two new terms on the right-hand side of Eq. (B.38) represent the work done by the Reynolds stresses and the turbulent transport of the heat. The three unknowns that arise from the modified conservation of energy equation are the components of the heat flux vector $\overline{\rho \boldsymbol{u}'' \mathrm{h}''}$. A Reynolds analogy is typically used to model the turbulent heat flux.

$$\overline{\rho \boldsymbol{u}'' \mathrm{h}''} = -C_p \frac{\mu_T}{Pr_T} \nabla \tilde{T} \tag{B.39}$$

The variable Pr_T is the turbulent Prandtl number. For air, it is assumed to have the constant value of 0.9. To simplify the equations, effective components for the viscosity and thermal conductivity are formed.

$$\mu_{eff} = \mu + \mu_T \tag{B.40}$$

$$\kappa_{eff} = C_p \left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \tag{B.41}$$

A model needs to be created so the new Reynolds stress terms can be approximated. A model proposed by Boussinesq in 1877 approximated the Reynolds stresses as being linearly related to the Favre averaged velocity gradients. The constant of proportionality is called the eddy viscosity term μ_t.

$$\overline{\rho u_i'' u_j''} = \mu_t \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \tilde{u}_k}{\partial x_k} \right) - \frac{2}{3} \delta_{ij} \bar{\rho} k \tag{B.42}$$

The variables δ_{ij} and k in Eq. (B.42) are the Kronecker's delta and turbulent kinetic energy, respectively. The Favre-averaged Navier-Stokes equations can now be rewritten in the following way. The average symbols have been removed as every quantity is an averaged quantity going forward. The source terms have also been dropped.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{B.43}$$

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \overline{\rho u_i'' u_j''} \tag{B.44}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} H) = \nabla \cdot \left(\overline{\rho u_i'' u_j''} \cdot \boldsymbol{u}\right) + \nabla \cdot \left(\kappa_{eff} \nabla T\right) \tag{B.45}$$

To close the equations, only a new relationship to obtain the eddy viscosity term is needed. It is worth remembering that the eddy viscosity is not a physical quantity, it is an artificial variable used to model the production and decay of eddies that are formed in the flow field.

Spalart-Allmaras Turbulence Model

The Spalart-Allmaras [144] turbulence model ignores the last term in Eq. (B.42) as the turbulent viscosity is considered to be an isotropic scalar quantity. The assumption of isotropic turbulent viscosity is violated in the case where a flow experiences a sudden change of mean strain rate, secondary flows, boundary layer separation and reattachment, or significant streamline curvature. In this work, the approximation of isotropic turbulent viscosity is considered valid. The Spalart-Allmaras (SA) turbulence model was specifically designed for aerodynamic flows, particularly around an airfoil or wing. Despite it being a one equation model, it has been shown to give accurate predictions for attached boundary layer flow. An obvious benefit is that is provides a simpler model to use than two equation models. The SA scheme was differentiated by Dwight [37] and the accompanying adjoint code for the scheme is available within TAU. For these reasons, it is a good fit for this work, and it is the turbulence model that will be proceeded with.

A turbulence model is needed in RANS simulations to close the system of equations. The SA model allows for the computation of the state variables without requiring the time-dependent flow field to be calculated.

There are a number of constants used in the SA derivation, these are listed below.

$$\sigma = \frac{2}{3}$$

$$\kappa = 0.41$$

$$c_{b1} = 0.1355$$

$$c_{b2} = 0.622$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma} \tag{B.46}$$

$$c_{w2} = 0.3$$

$$c_{w3} = 2$$

$$c_{t3} = 1.2$$

$$c_{t4} = 0.5$$

$$c_{v1} = 7.1$$

The SA model develops a transport equation in order to find the artificial variable called the modified eddy viscosity term $\hat{v}_t$. It uses this to close the RANS equations by relating the Reynolds stresses to the strain tensor.

$$-\overline{u_i' u_j'} = 2\hat{v}_t S_{ij} \tag{B.47}$$

The modified eddy viscosity term is related to the eddy viscosity through a simple algebraic relation.

$$\mu_t = \rho \hat{v}_t f_{v1} \tag{B.48}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{B.49}$$

$$\chi = \frac{\hat{v}_t}{\mathrm{v}} \tag{B.50}$$

The variable v is the kinematic viscosity. The variable d is the distance to the nearest solid wall. Additional definitions used in the transport equation are defined below.

$$f_{t2} = c_{t3} e^{-c_{t4}\chi^2}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$

$$W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

$$\Omega = \sqrt{2 W_{ij} W_{ij}} \tag{B.51}$$

$$\widehat{\boldsymbol{S}} = \Omega + \frac{\hat{v}_t}{\kappa^2 d^2} f_{v2}$$

$$r = min\left[\frac{\hat{v}_t}{\widehat{\boldsymbol{S}} \kappa^2 d^2}, 10\right]$$

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

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

The term f_{t2} is used to delay the transition of the flow from laminar flow to turbulent flow. There are many variants of the SA model, a common variant neglects the f_{t2} as it makes only a small difference at high Reynolds numbers [145]. The variable Ω is the magnitude of the vorticity, $\widehat{\boldsymbol{S}}$ is a modification to the vorticity term that is used to maintain log-law behaviour all the way to the wall. Difficulties can arise if the value of $\widehat{\boldsymbol{S}}$ goes to zero or below so some variants, such as the negative SA model [146] provide a different transport equation to be used in regions of negative $\hat{v}_t$.

The modified eddy viscosity is governed by the transport equation in Eq. (B.52).

$$\frac{\partial \hat{v}_t}{\partial t} + \boldsymbol{u} \cdot \nabla \hat{v}_t =$$

$$c_{b1}(1 - f_{t2})\widehat{\boldsymbol{S}}\hat{v}_t - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\hat{v}_t}{d}\right)^2 + \frac{1}{\sigma}\nabla \cdot [(\mathrm{v} + \hat{v}_t)\nabla \hat{v}_t] + \frac{c_{b2}}{\sigma}(\nabla \hat{v}_t)^2 \qquad (B.52)$$

The boundary conditions for the transport equation are:

$$\hat{v}_{wall} = 0 \qquad (B.53)$$

$$\hat{v}_{farfield} = [3v_\infty, 5v_\infty] \qquad (B.54)$$

Eq. (B.52) is the standard form of the SA model. This equation has similarities with the Navier-Stokes equations. The left-hand side consists of the rate of change of a quantity with respect to time along with the advection of the same quantity. While the terms on the right-hand side consist of the production, destruction, diffusion, and anti-diffusion of the same quantity, respectively. In solving Eq. (B.52) for the modified eddy viscosity, the set of equations for modelling the air flow becomes closed.

Appendix C The Linear Elasticity Equations

Elasticity is the material property that causes an object to return to its original shape after the load that displaced it has been removed. The relationship between the loading force and elastic behaviour gets broken when the loading becomes too large and plastic displacement occurs. Plastic displacement occurs if the loading applied to the material takes the object past its yield point. If this occurs the object will not return back to its original shape. In aeroelastic simulations of an aircraft, it is assumed that this does not happen for the given wing structure. However, if an aerostructural optimisation with structural parameters is undertaken, it must be ensured that the changed structure never allows the stresses in the wing to exceed the yield point stress.

The simplest approach to modelling elasticity is to assume that the elastic behaviour of an object is linear. At small strain values of a material, the elastic behaviour does act linearly. For the wings of aircraft deflecting under aerodynamic loads, the linear elastic model holds reasonably well although for high aspect-ratio wings the behaviour becomes less linear [54]. For this work, it is assumed that a linear elastic model of deformation for the wing and horizontal tail is an appropriate model.

A cartesian coordinate system is introduced with base vectors $\mathbf{e}_i$ where one of the base vectors is normal to the surface. The origin of the coordinate system is located at a point where a surface traction acts. A surface traction is the force per unit area in a specified direction applied on a surface from an external source. A traction sets up a system of stresses in the body.

Consider the traction in Fig. 32. The applied traction is given by the expression in Eq. (C.1).

$$\mathbf{t}^i = t_1\mathbf{e}_1 + t_2\mathbf{e}_2 + t_3\mathbf{e}_3 \qquad (C.1)$$

$$\mathbf{t}^i = \sigma_{i1}\mathbf{e}_1 + \sigma_{i2}\mathbf{e}_2 + \sigma_{i3}\mathbf{e}_3 \qquad (C.2)$$

The superscript of the traction vector represents the normal direction from the surface and the traction. For the example in Fig 3.2, this direction is axis 3. The traction will consist of two shear stress components and one normal stress component.

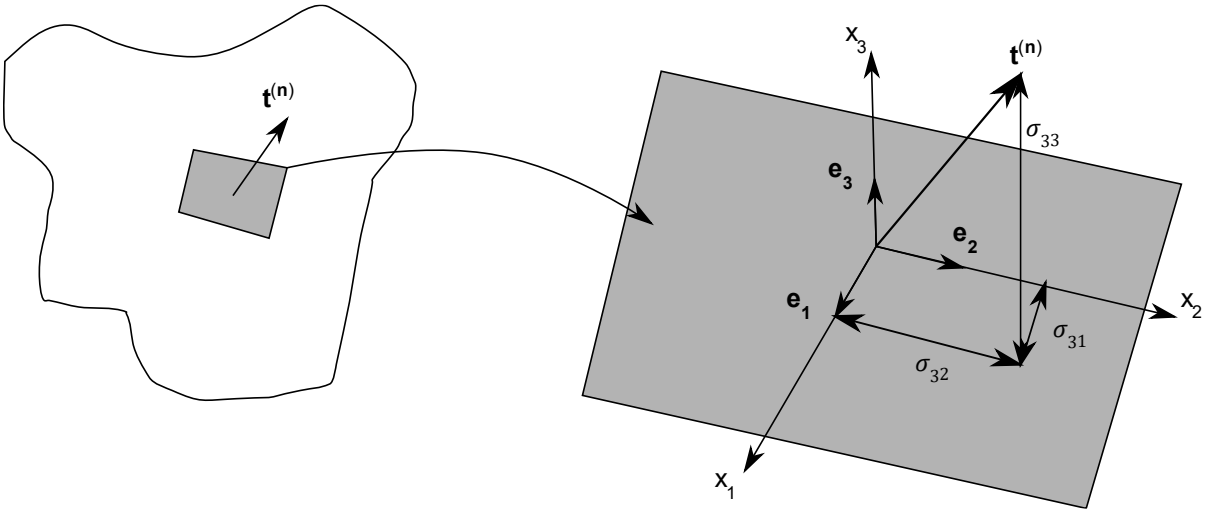

Fig. 32 A system of stresses created by a surface traction.

A three-dimensional infinitesimal volume can be subjected to three surface tractions: $\mathbf{t}^1$, $\mathbf{t}^2$ and $\mathbf{t}^3$. The surface tractions applied to an infinitesimal volume can be written in tensor format.

$$\mathbf{t}^i = \sigma_{ij}\mathbf{e}_j \tag{C.3}$$

The stress components caused by the surface tractions in the volume element can be represented in a 3×3 matrix.

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{C.4}$$

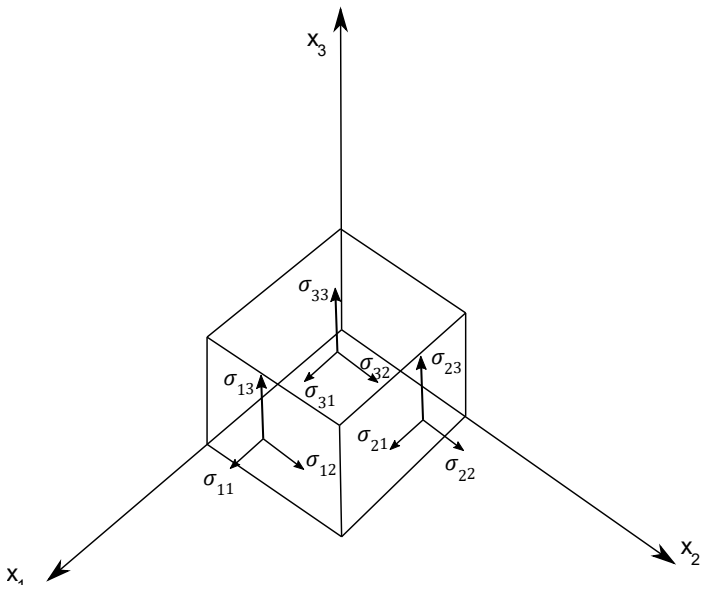

Fig. 33 The nine stress components with respect to a Cartesian coordinate system.

If another volume element from the same material particle was taken, but its normal directions were defined by x_1', x_2' and x_3', then the nine stress values would be different also.

$$\boldsymbol{\sigma}' = \begin{bmatrix} \sigma_{11}' & \sigma_{12}' & \sigma_{13}' \\ \sigma_{21}' & \sigma_{22}' & \sigma_{23}' \\ \sigma_{31}' & \sigma_{32}' & \sigma_{33}' \end{bmatrix} \tag{C.5}$$

Therefore, it is useful to be able to determine the tractions that act on a plane when that plane is not normal to the coordinate axis.

Cauchy's Law states that the vector normal to a surface, $\boldsymbol{n} = n_i\boldsymbol{e}_i$, is related to the traction vector acting on that surface. In other words, given the stresses and the normal to a plane, the traction vector acting along the plane can be determined. $\boldsymbol{n}$ is a linear combination of the base vectors.

$$t_i = \sigma_{ji}n_j = \sigma_{ij}n_j \tag{C.6}$$

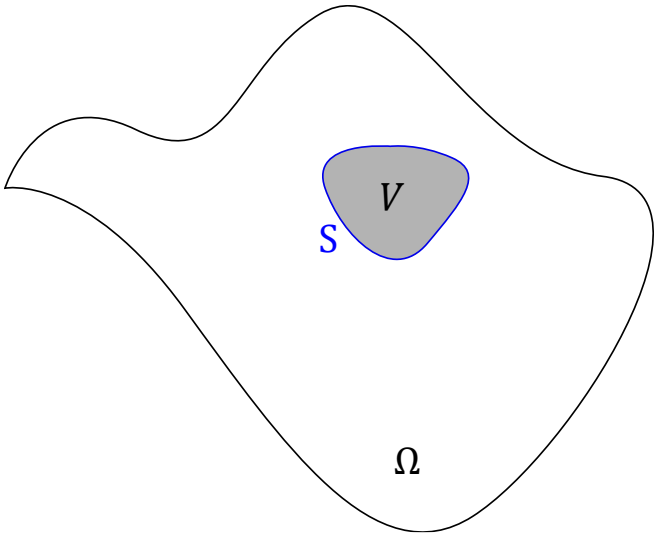

Fig. 34 A continuum body Ω containing an arbitrary volume V.

Now consider an object with an arbitrary volume V with a closed surface boundary S as shown in Fig. 34. The object has a uniform density ρ, body forces $\boldsymbol{f}$ and tractions $\mathbf{t}$ applied to it. This displaces the body by $\boldsymbol{u}$ and the body experiences an acceleration of $\ddot{\boldsymbol{u}}$. Applying Newton's second law gives:

$$\int_V \rho\ddot{\mathbf{u}}\, dV = \int_V \rho\boldsymbol{f}\, dV + \int_S \mathbf{t}\, dS \tag{C.7}$$

$$\int_V (\rho\ddot{\mathbf{u}} - \rho\boldsymbol{f})\, dV = \int_S t_i\, dS, \qquad i = 1,2,3 \tag{C.8}$$

$$\int_V (\rho\ddot{\mathbf{u}} - \rho\boldsymbol{f})\, dV = \int_S \sigma_{ij} n_j\, dS \tag{C.9}$$

$$\int_S \sigma_{ij} n_j\, dS = \int_S \sigma_{ij,j}\, dV, \qquad By\ divergence\ theorem \tag{C.10}$$

$$\int_V \left(\rho\ddot{u}_i - \sigma_{ij,j} - \rho f_i\right) dV = 0 \tag{C.11}$$

As the volume V is arbitrary, the integrand must equal zero. As the case of interest to this project is the case of static equilibrium, $\ddot{\boldsymbol{u}} = \mathbf{0}$. This gives the equation of equilibrium.

$$-\sigma_{ij,j} = \rho f_i \tag{C.12}$$

Rewriting Eq. (C.12) in vector notation gives the Lamé equation.

$$-\nabla \cdot \boldsymbol{\sigma} = \rho\boldsymbol{f} \tag{C.13}$$

Hooke's Law

The relationship between stress and strain is described by Hooke's law for linear elastic materials. In three-dimensions Hooke's law takes the form:

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon} + \lambda \mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I} \tag{C.14}$$

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^2) \tag{C.15}$$

It is convenient to write the stress and strain expressions in tensor form.

$$\sigma_{ij} = 2\mu\varepsilon_{ij} + \boldsymbol{\lambda}\varepsilon_{kk}\delta_{ij} \tag{C.16}$$

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) \tag{C.17}$$

The variables μ and $\boldsymbol{\lambda}$ are the Lamé parameters, tr is the trace operator, $\boldsymbol{I}$ is the identity matrix and δ_{ij} is the Kronecker delta function. The variable strain $\boldsymbol{\varepsilon}$ is the strain tensor. It is convenient to flatten the expression in Eq. (C.14) into a matrix vector format so it can more easily be implemented into the finite element method.

$$\mu = \frac{\mathrm{E}}{2(1+\nu)} \tag{C.18}$$

$$\boldsymbol{\lambda} = \frac{\mathrm{E}\nu}{(1+\nu)(1-2\nu)} \tag{C.19}$$

Where E is the Young's modulus of the material and ν is the Poisson's ratio of the material, both variables are constant for a linear elastic material along an axis. The Young's modulus is the ratio of stress over strain along an axis. Poisson's ratio is the ratio of transverse to axial strain, or how much does stretching the material in one direction cause it to compress in the others.

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yx} \\ \sigma_{yy} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{zy} \\ \sigma_{zz} \end{Bmatrix} = [9 \times 9\ coefficient\ matrix] \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yx} \\ \varepsilon_{yy} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \\ \varepsilon_{zy} \\ \varepsilon_{zz} \end{Bmatrix} \tag{C.20}$$

Alternatively in tensor form it is:

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl} \tag{C.21}$$

The variable c_{ijkl} is known as the stiffness tensor.

Linear Elasticity Discretisation

The continuous model described by the Lamé equation needs to be discretised so that it can be computed. A finite difference or finite volume approach could be pursued but the most common approach for a structural problem is to use the finite element method. The core idea behind the finite element (FE) method is that if the infinite dimensional continuous problem cannot be solved, attempt to represent the problem in a finite dimensional subspace that can be solved. More simply, it sets up a system of algebraic equations which can be solved numerically.

The FE method has the following steps:

1. Weak reformulation of the problem (handle the continuous differential operator)
2. Spatially discretise the domain into a number of elements
3. The type of elements with associated interpolation functions must be chosen, resulting in a local system of equations for each element
4. Assemble a global system of equations from each elements' local system of equations
5. Apply a boundary condition
6. Numerically solve the system

Weak reformulation

Weak formulations allow the transfer of concepts from linear algebra to other fields such as partial differential equations. A weak formulation of an equation means it is no longer required to hold absolutely and has weak solutions with respect to test functions w. It also allows for redistribution of derivatives and the incorporation of boundary conditions.

The weak form is obtained by first integrating over the volume domain where the problem is being solved and the integrand is multiplied by the test function. Now, instead of requiring the Lamé equation to hold at every point in the domain, only a weighted sum of integrated terms is required to hold. The weak form of the linear elasticity problem is written as:

$$\int_V (\sigma_{ij,j} - \rho f_i) w_i \, dV = 0 \tag{C.22}$$

Using a corollary of the divergence theorem, the derivatives can be redistributed to give:

$$\int_V \sigma_{ij} w_{i,j} \, dV = \int_V \rho f_i w_i \, dV + \int_S t_i w_i \, dS \tag{C.23}$$

Discretisation

The continuous domain is discretised by creating a mesh to represent the domain. A mesh consists of nodes which then define finite elements within the mesh. An element in a 3D representation is some geometric 3D shape such as a tetrahedron or prism. The element's definition will be determined by the number of nodes that make up the element and the element's chosen interpolation (or basis) functions. Interpolation functions approximate the value any quantity will have at any position inside the element.

A basis function φ^{ne} is only non-zero within its associated element e and it is a scalar function that depends on the location. The basis function is bounded by the node n on element e.

$$\varphi^{ne}(\boldsymbol{x}^m) = \delta_{nm} \tag{C.24}$$

Where $\boldsymbol{x}^m$ is the location of a node on the element e. The continuous unknown quantities such as the displacements are approximated by interpolation functions for each element.

$$u_i^e \approx \varphi^{ne} u_i^{ne} \tag{C.25}$$

The variable u_i^{ne} is the displacement of a node n within the element e in the direction i. There is an implicit summation over the nodes n within the element to output the interpolation function of the element. The spatial derivative of displacement function within the element can be approximated as:

$$u_{i,j}^e \approx \varphi_j^{en} u_i^{ne} \equiv \frac{\partial \varphi^{ne}}{\partial x_j} u_i^{ne} \tag{C.26}$$

The test functions introduced when reformulating the problem into a weak form are the interpolation functions.

$$w_i^{en} \equiv \varphi^{ne} W_i^{en} \tag{C.27}$$

The variable W represents a term known as virtual displacements. The Galerkin method is used to convert the weak form of the linear elasticity problem into a discretised form.

$$\sum_e \int_{V_e} \left(\sigma_{ij} \varphi_j^{en} dV\right) W_i^{en} = \sum_e \left(\int_{V_e} \rho f_i \varphi^{en} dV + \int_{S_e} t_i \varphi^{en} dS \right) W_i^{en} \tag{C.28}$$

For each element, the stress can be approximated by the following expression.

$$\sigma_{ij}^e = \mu\left(\varphi_j^{en} u_i^{en} + \varphi_i^{en} u_j^{en}\right) + \lambda \varphi_k^{en} u_k^{en} \delta_{ij} \tag{C.29}$$

As the equality in Eq. (C.28) will hold for all virtual displacements, the equation can be rewritten as a system of equations with each element satisfying the following expression:

$$\int_{V_e} \sigma_{ij}^e \varphi_j^{en} dV = \int_{V_e} \rho f_i^e \varphi^{en} dV + \int_{S_e} t_i \varphi^{en} dS \tag{C.30}$$

For simplicity, it is assumed that the tractions are zero everywhere and the body forces do not depend on their location. Therefore, the discretised linear elasticity equation for an element becomes:

$$\int_{V_e} \sigma_{ij}^e \varphi_j^{en} dV = \rho f_i \int_{V_e} \varphi^{en} dV \tag{C.31}$$

To relate the nodal displacements to the forces, Eq. (C.29) can be substituted into Eq. (C.31).

$$\int_{V_e} \left(\mu\left(\varphi_j^{en} u_i^{en} + \varphi_i^{en} u_j^{en}\right) + \lambda \varphi_k^{en} u_k^{en} \delta_{ij}\right) \varphi_j^{en} dV = \rho f_i \int_{V_e} \varphi^{en} dV \tag{C.32}$$

Form local element stiffness matrix

The expression in Eq. (C.31) is generic for any element type. Before a global stiffness matrix can be obtained a local element stiffness matrix must be obtained. The simplest element in 3D is the linear tetrahedron element. This will be used to show how a local element's stiffness matrix is formed.

A linear tetrahedra element has four nodes and the interpolation functions are all linear.

$$\varphi(x_1, x_2, x_3) = a + bx_1 + cx_2 + dx_3 \tag{C.33}$$

To rewrite Eq. (C.31) for the linear tetrahedra element, the interpolation coefficients for each interpolation function need to be found. The coefficients are found by satisfying the basis function boundary condition defined in Eq. (C.24). Typically this is done by solving a number of linear systems, one linear system for each node in the element.

$$\begin{bmatrix} 1 & x_1^1 & x_2^1 & x_3^1 \\ 1 & x_1^2 & x_2^2 & x_3^2 \\ 1 & x_1^3 & x_2^3 & x_3^3 \\ 1 & x_1^4 & x_2^4 & x_3^4 \end{bmatrix} \begin{Bmatrix} a^1 \\ b^1 \\ c^1 \\ d^1 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

$$\vdots \qquad (C.34)$$

$$\begin{bmatrix} 1 & x_1^1 & x_2^1 & x_3^1 \\ 1 & x_1^2 & x_2^2 & x_3^2 \\ 1 & x_1^3 & x_2^3 & x_3^3 \\ 1 & x_1^4 & x_2^4 & x_3^4 \end{bmatrix} \begin{Bmatrix} a^4 \\ b^4 \\ c^4 \\ d^4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix}$$

For implementation purposes, it is useful to convert from tensor notation, which was useful in the derivation, to matrix vector notation. The discretised formulation reached in Eq. (C.32) is linear in U so the left-hand side of the equation can be reorganised as a matrix vector product, with the vector being the nodal displacements of the element.

$$\boldsymbol{K}^e \boldsymbol{u}^e = \boldsymbol{f}^e \qquad (C.35)$$

The variable $\boldsymbol{K}^e$ is the element stiffness matrix and it can be decomposed into a geometric matrix component $\boldsymbol{B}^e$ and a material matrix component $\mathbf{D}^e$.

$$\boldsymbol{K}^e \equiv \int_{V_e} [\boldsymbol{B}^e]^T \mathbf{D}^e \boldsymbol{B}^e dV \qquad (C.36)$$

The material matrix for a linear elastic material takes the form:

$$\boldsymbol{D}^e = \begin{bmatrix} 2\mu + \boldsymbol{\lambda} & \boldsymbol{\lambda} & \boldsymbol{\lambda} & 0 & 0 & 0 \\ \boldsymbol{\lambda} & 2\mu + \boldsymbol{\lambda} & \boldsymbol{\lambda} & 0 & 0 & 0 \\ \boldsymbol{\lambda} & \boldsymbol{\lambda} & \mu + \boldsymbol{\lambda} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \qquad (C.37)$$

The geometric matrix component depends on the element type, for the linear tetrahedra element it takes the form:

$$\boldsymbol{B}^e = \begin{bmatrix} b^1 & 0 & 0 & b^2 & 0 & 0 & b^3 & 0 & 0 & b^4 & 0 & 0 \\ 0 & c^1 & 0 & 0 & c^2 & 0 & 0 & c^3 & 0 & 0 & c^4 & 0 \\ 0 & 0 & d^1 & 0 & 0 & d^2 & 0 & 0 & d^3 & 0 & 0 & d^4 \\ c^1 & b^1 & 0 & c^2 & b^2 & 0 & c^3 & b^3 & 0 & c^4 & b^4 & 0 \\ d^1 & 0 & b^1 & d^2 & 0 & b^2 & d^3 & 0 & b^3 & d^4 & 0 & b^4 \\ 0 & d^1 & c^1 & 0 & d^2 & c^2 & 0 & d^3 & c^3 & 0 & d^4 & c^4 \end{bmatrix} \qquad (C.38)$$

The integrations required for solving the element stiffness equation in Eq. (C.36) are typically done numerically. In the case where linear interpolation functions are used for an element, the left-hand side of the equation is not dependent on the location so the integration can be replaced by a multiplication with the volume of the element.

Appendix D GMRES and Krylov Subspaces

Large linear systems cannot be solved by direct inversion. Instead, they must be solved using an iterative procedure. Consider the large linear system in Eq. (D.1).

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \qquad \boldsymbol{A} \epsilon \mathbb{R}^{m \times m} \tag{D.1}$$

A Krylov subspace is constructed by repeated applications of the $\boldsymbol{A}$ matrix on the vector $\boldsymbol{b}$. The matrix should not be multiplied with itself as this would damage the sparsity of the matrix and as a result make the computation more expensive. The Krylov matrix of this linear system is defined in Eq. (D.3).

$$\boldsymbol{\kappa}_n = \langle \boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \dots, \boldsymbol{A}^{n-1}\boldsymbol{b} \rangle, \qquad \boldsymbol{n} \leq \boldsymbol{m} \tag{D.2}$$

$$\mathbf{K}_n = [\boldsymbol{\kappa}_n], \qquad \mathbf{K}_n \in \mathbb{R}^{m \times n} \tag{D.3}$$

The column space as defined in Eq. (D.2) of the Krylov matrix is known as the Krylov subspace. The more Krylov iterations, the bigger the Krylov space. If the vector $\boldsymbol{x}$ is in the span the Krylov subspace, then the vector $\boldsymbol{x}$ can be defined Eq. (D.4).

$$\boldsymbol{x} = \mathbf{K}_n \boldsymbol{z}, \qquad \boldsymbol{z} \in \mathbb{R}^n \tag{D.4}$$

The vector $\boldsymbol{z}$ is simply some vector of size $n \times 1$. The big idea of Krylov subspace methods it to replace the action of the large dimensional matrix $\boldsymbol{A}$ with the action of a much smaller dimensional matrix $\mathbf{K}_n$**.** Where $\mathbf{K}_n$ approximates the behaviour of $\boldsymbol{A}$. This problem becomes a minimisation problem.

$$\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \mathbf{0} \tag{D.5}$$

$$min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2 \tag{D.6}$$

The optimisation problem in Eq. (D.6) is equivalent to finding the vector $\boldsymbol{x}$ as defined in Eq. (D.4) which solves the optimisation problem.

$$min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{b} - \boldsymbol{A}\mathbf{K}_n \boldsymbol{z}\|_2 \tag{D.7}$$

Now, Eq. (D.7) is just an ordinary least squares problem that will return $\boldsymbol{z}$. The approximate solution to the linear system can be retrieved by plugging the obtained value of $\boldsymbol{z}$ into Eq. (D.4). This is the GMRES method. The GMRES method, uses Krylov subspaces to minimise the residual at each iteration by finding the linear combination of Krylov subspaces that minimise the residual.

It would be reasonable to expect that increasing the dimensionality of the Krylov subspace would improve the accuracy of the approximation. However, as the dimensionality is increased by repeatedly applying the $\boldsymbol{A}$ matrix to the previous Krylov subspace vector, the vectors created become increasingly parallel. Ideally for the minimisation problem, the Krylov subspace would consist of orthogonal vectors as this will avoid cancellation errors when solving the problem. Mathematically, adding a new Krylov vector to the matrix will increase its rank by one, but numerically due to the subtractive cancellation errors, the rank does not increase after a certain number of Krylov iterations. To make the subspace consist of a more suitable set of vectors for the minimisation problem, a technique such as QR decomposition is often employed.

$$\mathbf{K}_n = \boldsymbol{Q}_n \boldsymbol{R}_n \tag{D.8}$$

Here, $\boldsymbol{Q}_n$ is an orthogonal matrix while $\boldsymbol{R}_n$ is an upper triangle matrix which will be discarded for the purposes of solving the minimisation problem. Only the orthogonal vectors in the column space of $\boldsymbol{Q}_n$ are of interest. Instead of finding the linear combination of Krylov subspace vectors that minimises

the problem, a linear combination of the orthogonal vectors defined by the column space of $\boldsymbol{Q}_n$ is used.

$$\boldsymbol{x} = \boldsymbol{Q}_n \boldsymbol{y} \tag{D.9}$$

$$min_{\boldsymbol{y}\in\mathbb{R}^n} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{Q}_n \boldsymbol{y}\|_2 \tag{D.10}$$

The GMRES method simply increments the value of n, increasing the size of the Krylov subspace until the residual of the linear system reaches a sufficient level of convergence. For the purposes of the flow adjoint to produce an accurate gradient, it is considered sufficient for the flow adjoint to have an order of magnitude less convergence [147] than the flow's primal solution residual. Nadarajah and Jameson [33], and Kim *et al*. [148] suggested that it is good practice to start from relaxed convergence values for the primal and adjoint solution and progressively converge them until the gradient stops changing.

Appendix E Coupled Newton-Krylov Method

The major difference between the Newton-Krylov (NK) approach and the block Gauss-Seidel (BGS) approach is the convergence criterion. In the BGS approach, a convergence criterion is based on the change in structural displacements between iterations. In the NK approach, the residuals of the aerodynamic and structural problems are combined into a single residual function. When this residual function is small enough the coupled system is considered to be converged.

$$\boldsymbol{R} = \begin{Bmatrix} \boldsymbol{R}_a \\ \boldsymbol{R}_s \end{Bmatrix} = \boldsymbol{0} \tag{E.1}$$

The procedure of an NK iteration is similar to an iteration of BGS. However, there are two parts to an NK iteration. The first part is a computation of the non-linear residual, here only the residuals of the flow and structural governing equations are evaluated. One iteration of the non-linear residual computation of the NK method follows the same procedure as a single BGS iteration.

Single Newton-Krylov non-linear residual iteration

1. **Function** $\boldsymbol{R}(\boldsymbol{W}, \boldsymbol{u}_s)$: *State variables* $\boldsymbol{W}$ *and* $\boldsymbol{u}_s$ *are passed into the non-linear computation.*
2. $\boldsymbol{X}_\Delta = \boldsymbol{X}_\Delta(\boldsymbol{u}_s)$: *CFD mesh is deformed by* $\boldsymbol{u}_s$
3. $\boldsymbol{R}_a = \boldsymbol{R}_a(\boldsymbol{W}, \boldsymbol{X}_\Delta)$: *CFD residual is updated on new mesh*
   A new flow solution is not run. Just the residuals with these parameters are evaluated.
4. $\boldsymbol{F}_a = \boldsymbol{F}_a(\boldsymbol{W}, \boldsymbol{X}_\Delta)$: *Aerodynamic forces are evaluated on new mesh.*
5. $\boldsymbol{F}_s = \boldsymbol{F}_s(\boldsymbol{F}_a)$: *Forces are transferred to structural mesh.*
6. $\boldsymbol{R}_s = \boldsymbol{R}_s(\boldsymbol{F}_s)$: *CSM residual is updated.*
7. $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{R}_a, \boldsymbol{R}_s)$: *New total residual is updated.*

In order to start the Newton-Krylov method off, an initial set of state variables need to be provided. These are often found by performing one iteration or multiple iterations [77] with the BGS approach. After the initial BGS iterations are performed, new state variables are found by solving the large linear system in Eq. (E.2).

$$\begin{bmatrix} \dfrac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{W}} & \dfrac{\partial \boldsymbol{R}_a}{\partial \boldsymbol{u}_s} \\ \dfrac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{W}} & \dfrac{\partial \boldsymbol{F}_s}{\partial \boldsymbol{u}_s} - \boldsymbol{K} \end{bmatrix} \begin{Bmatrix} \Delta \boldsymbol{W} \\ \Delta \boldsymbol{u}_s \end{Bmatrix} = - \begin{Bmatrix} \boldsymbol{R}_a(\boldsymbol{W}) \\ \boldsymbol{R}_s(\boldsymbol{u}_s) \end{Bmatrix} \tag{E.2}$$

The linear system in Eq. (E.2) that is used to find the converged state variables takes the same form as the coupled-adjoint linear system, except the matrix is transposed. This linear system is solved, and the flow state variables and structural state variables are updated with $\Delta \boldsymbol{W}$ and $\Delta \boldsymbol{u}_s$ respectively. The method used for solving the non-symmetric linear system is the popular Generalized Minimum Residual Algorithm for solving Non-symmetric Linear Systems [149] (GMRES).

Newton-Krylov Algorithm

1. Initial state variables calculated through some iterations of BGS method.
2. Solve the linear system in Eq. (E.2) to get the state variable updates.
3. Calculate the non-linear residuals for the updated state variables.
4. Check if non-linear residuals have met convergence criteria. If not, go to step 2.

One downside with the NK approach, like most Newton-type methods is that they may not converge for some initial conditions. On the other hand, the NK method eliminates the need for an under-

relaxation parameter which is sometimes needed in the BGS approach for flexible structures. As a result, this can lead to faster convergence when handling aircraft with flexible structures [77].

Appendix F Mesh Deformation Equations

Delaunay Graph Mapping

Delaunay Graph Mapping starts by subdividing the mesh via Delaunay triangulation. Delaunay triangulation is a method of connecting a cloud of points in a unique way by ensuring all joined points meet the Delaunay criteria. This criteria states that "The circumcircle of each triangle must not include other points outside of those constructing the triangle" [150]. This concept is extended to 3D by creating tetrahedra elements and ensuring the elements' circumsphere does not include points other than those in the element.

The Delaunay triangulation is conducted on all the surface points and some farfield points so that all mesh points are enclosed within a subdomain. With each mesh node within a Delaunay element, an explicit relationship between the node and the nodes that make up the Delaunay element can be made. Figure 35 shows a mesh node p within its Delaunay element ABC.

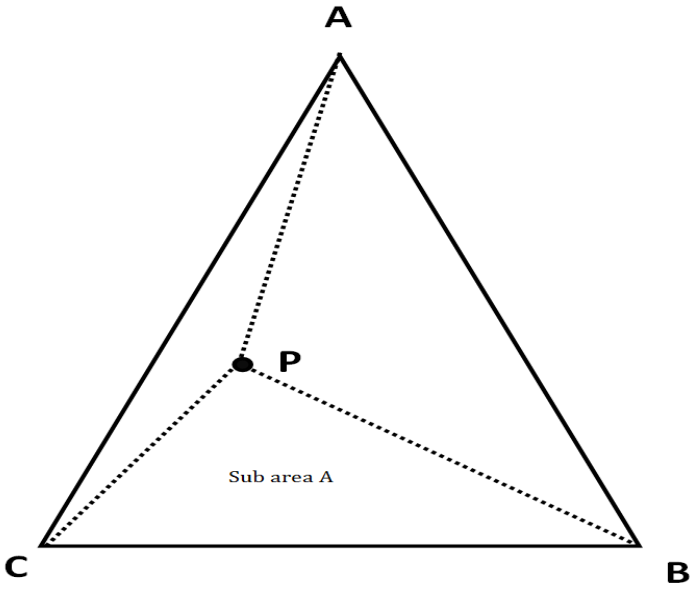

Fig. 35 A mesh node within its Delaunay element.

In 2D the mesh node is related to the Delaunay element vertices via area ratios, in 3D it is related via volume ratios. The area ratios e_i are the ratio of the sub-area a_i and the total element area A.

$$e_i = \frac{\mathrm{a}_i}{\mathrm{A}} \tag{F.1}$$

Using Fig. 35 as a reference, sub-area a_1 refers to the triangle CPB. A, B and C can be replaced by the notation $\boldsymbol{x}_{D_1}$, $\boldsymbol{x}_{D_2}$ and $\boldsymbol{x}_{D_3}$ respectively. Now the mesh location can be specified in terms of the Delaunay vertex locations.

$$\boldsymbol{X}_p = \sum_{i=1}^{3} e_i \, \boldsymbol{x}_{D_i} \tag{F.2}$$

$$\boldsymbol{X} = \boldsymbol{EB} \tag{F.3}$$

$\boldsymbol{B}$ is made up of both surface nodes and farfield nodes so only a section of the Delaunay deformation matrix is needed.

$$\boldsymbol{B} = \begin{Bmatrix} \boldsymbol{S} \\ \boldsymbol{X}_{ff} \end{Bmatrix} \tag{F.4}$$

$$\frac{\partial \boldsymbol{u}_{vol}}{\partial \boldsymbol{u}_{surf}} = \boldsymbol{E_S} \tag{F.5}$$

Inverse Distance Weighting

The convergence of the Delaunay method was obtained by improving the diagonal dominance of the coupled-adjoint matrix. The non-consistent mesh deformation approach for FSI and the coupled-adjoint was tested with another explicit mesh deformation algorithm, IDW. IDW deforms the mesh in a similar way to RBF in that each surface point effects every volume point. The deformation is also similar in that the mesh movement is explicitly dependant on the radial distance from the surface points. It has also been shown to have a similar accuracy to RBF mesh deformation [68] when used for interpolation purposes.

$$\Delta \boldsymbol{X}_i = \begin{cases} \dfrac{\sum_{j=1}^{n_{surf}} \mathrm{w}_j(\boldsymbol{X}_i)\boldsymbol{u}_{surf,j}}{\sum_{j=1}^{n_{surf}} \mathrm{w}_j(\boldsymbol{X}_i)}, & if\ \boldsymbol{X}_i \neq \boldsymbol{X}_j\ for\ all\ j \\ \boldsymbol{u}_{surf,j}, & if\ \boldsymbol{X}_i \neq \boldsymbol{X}_j\ for\ any\ j \end{cases} \tag{F.6}$$

$$\mathrm{w}_j(\boldsymbol{X}_i) = \frac{1}{\left\|\boldsymbol{X}_i - \boldsymbol{X}_j\right\|^3} \tag{F.7}$$

An advantage of IDW is that it does not need to be solved iteratively meaning just like the Delaunay method. One disadvantage it has when compared to the Delaunay method is that it cannot be stored explicitly. It has n_{vol} rows and n_{surf} columns in its interpolation matrix. For high fidelity CFD meshes, the interpolation matrix that represents this operation will be too large to store.

References

[1] European Commission, “Flightpath 2050 Europe’s Vision for Aviation, Report of the High Level Group on Aviation Research,” 2011, doi: 10.2777/50266.

[2] Airbus, “Global Market Forecast 2019-2038,” 2019.

[3] F. Daoud, S. Deinert, R. Maierl, and Ö. Petersson, “Integrated Multidisciplinary Aircraft Design Process Supported by a Decentral MDO Framework,” *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015, doi: 10.2514/6.2015-3090.

[4] G. Schuhmacher, I. Murra, L. Wang, A. Laxander, O. O’Leary, and M. Herold, “Multidisciplinary Design Optimization of a Regional Aircraft Wing Box*,” 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, doi: 10.2514/6.2002-5406.

[5] A. Jameson, “Aerodynamic shape optimization using the adjoint method,” *VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics*, 2003, doi: 10.1108/02644401311329334.

[6] M. Vázquez, A. Dervieux, and B. Koobus, “Multilevel optimization of a supersonic aircraft,” *Finite Elements in Analysis and Design*, 2004, pp. 2101-2124, doi: 10.1016/j.finel.2004.01.010.

[7] J. Sobieszczanski-Sobieski and R. T. Haftka, “Multidisciplinary aerospace design optimization: Survey of recent developments,” *Structural Optimization*, 1997, pp. 1-23, doi: 10.1007/BF01197554.

[8] J. F. Wang and J. Periaux, “Multi-point optimization using GAs and Nash/Stackelberg games for high lift multi-airfoil design in aerodynamics,” *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, Vol. 1, 2001, pp. 552-559, doi: 10.1109/CEC.2001.934440.

[9] J. Wang, Y. Wu, and J. Periaux, “Multi-point optimization with DDM, genetic algorithms and game theory for high lift configuration in aerodynamics,” *41st Aerospace Sciences Meeting and Exhibit*, 2003, doi: 10.2514/6.2003-294

[10] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther, “A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design,” *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 33–62, doi: 10.1023/B:OPTE.0000048536.47956.62.

[11] J. Reuther, J. Alonso, J. Martins, and S. Smith, “A coupled aero-structural optimization method for complete aircraft configurations,” *37th Aerospace Sciences Meeting and Exhibit*, 1999, doi: 10.2514/6.1999-187.

[12] K. Maute, M. Nikbay, and C. Farhat, “Coupled Analytical Sensitivity Analysis and Optimization of Three-Dimensional Nonlinear Aeroelastic Systems,” *AIAA Journal*, Vol. 39, No. 11, 2001, pp. 2051–2061, doi: 10.2514/2.1227.

[13] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski, “Multidisciplinary optimization methods for aircraft preliminary design,” *5th Symposium on Multidisciplinary Analysis and Optimization*, 1994, doi: 10.2514/6.1994-4325.

[14] J. Dudley, X. Huang, P. E. MacMillin, B. Grossman, R. Haftka and W. H. Mason, "Multidisciplinary optimization of the high-speed civil transport," *33rd Aerospace Sciences Meeting and Exhibit*, 1995, doi: 10.2514/6.1995-124.

[15] R. Heinrich and N. Kroll, "Fluid-Structure Coupling for Aerodynamic Analysis and Design – A DLR Perspective," *46th AIAA Aerospace Sciences Meeting*, 2008, doi: 10.2514/6.2008-561.

[16] T. Belytschko, "Methods and programs for analysis of fluid-structure systems," *Nuclear Engineering and Design*, Vol. 42, No. 1, 1977, pp. 41-52, doi: 10.1016/0029-5493(77)90060-7.

[17] K. J. Bathe and W. F. Hahn, "On transient analysis of fluid-structure systems," *Computers and Structures*, Vol. 10, No. 1–2, 1979, pp. 383-391, doi: 10.1016/0045-7949(79)90109-3.

[18] J. Donea, S. Giuliani, and J. P. Halleux, "An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 33, No. 1, 1982, pp. 689–723, doi: 10.1016/0045-7825(82)90128-1.

[19] U. P. Farias Filho, A. R. Alessandro, S. M. Silvana, and P. R. Paulo, "Minimization of vortex induced vibrations using Surrogate Based Optimization," *Structural and Multidisciplinary Optimization*, Vol. 52, No. 4, 2015, pp. 717-735, doi: 10.1007/s00158-015-1264-6.

[20] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, Vol. 45, No. 1–3. 2009, pp. 50-79, doi: 10.1016/j.paerosci.2008.11.001.

[21] A. Bhosekar and M. Ierapetritou, "Advances in surrogate based modeling, feasibility analysis, and optimization: A review," *Computers and Chemical Engineering*, Vol. 108, 2018, pp. 250-267, doi: 10.1016/j.compchemeng.2017.09.017.

[22] L. Laurent, R. Le Riche, B. Soulier, and P. A. Boucard, "An Overview of Gradient-Enhanced Metamodels with Applications," *Archives of Computational Methods in Engineering*, Vol. 26, No. 1, 2019, pp. 61-106, doi: 10.1007/s11831-017-9226-3.

[23] D. A. Caughey, M. M. Hafez, and M. Drela, "Pros & Cons of Airfoil Optimization," *Frontiers of Computational Fluid Dynamics 1998*, 1998, pp. 363–381, doi: 10.1142/9789812815774_0019.

[24] A. Jahangirian and A. Shahrokhi, "Aerodynamic shape optimization using efficient evolutionary algorithms and unstructured CFD solver," *Computers and Fluids*, Vol. 46, No. 1, 2011, pp. 270-276, doi: 10.1016/j.compfluid.2011.02.010.

[25] A. Hashimoto, S. Jeong, and S. Obayashi, "Aerodynamic Optimization of Near-future High-wing Aircraft," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 58, No. 2, 2015, pp. 73–82, doi: 10.2322/tjsass.58.73.

[26] E. Jonsson, C. Riso, C. A. Lupp, C. E. S. Cesnik, J. R. R. A. Martins, and B. I. Epureanu, "Flutter and post-flutter constraints in aircraft design optimization," *Progress in Aerospace Sciences*, Vol. 109. 2019, doi: 10.1016/j.paerosci.2019.04.001.

[27] J. Iott, R. Haftka, and H. Adelman, “Selecting step sizes in sensitivity analysis by finite differences,” *NASA Technical Memorandum 86382*, 1985.

[28] L. S. Pontryagin, “Optimal regulation processes,” *American Mathematical Society Translations*, Vol. 18, 1961, pp. 321-339,

[29] A. E. Bryson and Y. Ho, “Applied optimal control: Optimization, Estimation, and Control”, Vol. 1, 1975, Routledge, doi: 10.1201/9781315137667.

[30] S. Ruo, J. Malone, J. Horsten, and R. Houwink, “The LANN program - An experimental and theoretical study of steady and unsteady transonic airloads on a supercritical wing,” 1983, doi: 10.2514/6.1983-1686.

[31] Firth, G. C., “LANN Wing Design.,” *NASA Conference Publication 2262 Cryogenic Wind Tunnel Models. Design and Fabrication*, 1983, pp. 83–85.

[32] A. Jameson, “Aerodynamic design via control theory,” *Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260, doi: 10.1007/BF01061285.

[33] S. Nadarajah and A. Jameson, “A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization,” *38th Aerospace Sciences Meeting and Exhibit*, 2000, doi: 10.2514/6.2000-667.

[34] A. Jameson, S. Shankaran, and L. Martinelli, “Continuous adjoint method for unstructured grids,” *AIAA Journal*, Vol. 46, No. 5, 2008, 1226-1239, doi: 10.2514/1.25362.

[35] W. K. Anderson and V. Venkatakrishnan, “Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation,” *Computers and Fluids*, Vol. 28, No. 4–5, 1999, 443-480, doi: 10.1016/S0045-7930(98)00041-3.

[36] R. P. Dwight and J. Brezillon, “Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization,” *AIAA Journal*, Vol. 44, No. 12, 2006, pp. 3022–3031, doi: 10.2514/1.21744.

[37] R. P. Dwight, “Efficiency Improvement of RANS-Based Analysis and Optimization using Implicit and Adjoint Methods on Unstructured Grids,” *Ph.D. Dissertation*, University of Manchester, 2006.

[38] J. E. V Peter and R. P. Dwight, “Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches,” *Computers and Fluids*, Vol. 39, No. 3. 2010, pp. 373-391, doi: 10.1016/j.compfluid.2009.09.013.

[39] J. R. R. A. Martins, “A coupled-adjoint method for high-fidelity aero-structural optimization,” *Ph.D. Dissertation*, Stanford University, USA, 2002.

[40] B. Grossman, Z. Gurdal, R. T. Haftka, G. J. Strauch, and W. M. Eppard, “Integrated aerodynamic/structural design of a sailplane wing,” *Journal of Aircraft*, Vol. 25, No. 9, 1988, pp. 855-860, doi: 10.2514/3.45670.

[41] A. H. Bowers, O. J. Murillo, R. R. Jensen, B. Eslinger, and C. Gelzer, “On Wings of the Minimum Induced Drag: Spanload Implications for Aircraft and Birds,” *NASA Technical Publication 219072*, 2016.

[42] R. T. Haftka, “Optimization of flexible wing structures subject to strength and induced drag constraints,” *AIAA Journal*, Vol. 15, No. 8, 1977, pp. 1101-1106, doi: 10.2514/3.7400.

[43] I. Ghazlane, G. Carrier, and A. Dumont, “Aerostructural Adjoint Method for Flexible Wing Optimization Torsion constant*,” 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2012, pp. 1–13, doi: 10.2514/6.2012-1924.

[44] M. Abu-Zurayk, “An Aeroelastic Coupled Adjoint Approach for Multi-Point Designs in Viscous Flows,” *Ph.D. Dissertation*, Institut für Aerodynamik und Strömungstechnik, Deutsches Zentrum für Luft und Raumfahrt, Braunschweig, Germany, 2017.

[45] M. Abu-Zurayk, A. Merle, C. Ilic, S. Keye, S. Görtz, M. Schulze, T. Klimmek, C. Kaiser, D.Q. Martin, J. Häßy, R.G. Becker, B. Fröhler and J. Hartmann, “Sensitivity-based multifidelity multidisciplinary optimization of a powered aircraft subject to a comprehensive set of loads,” *AIAA Aviation 2020 Forum*, Vol. 1 PartF, 2020, pp. 1–26, doi: 10.2514/6.2020-3168.

[46] K. Maute, M. Nikbay, and C. Farhat, “Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method,” *International Journal for Numerical Methods in Engineering*, Vol. 56, No. 6, 2003, pp. 911-933, doi: 10.1002/nme.599.

[47] Z. J. Zhang and D. W. Zingg, “Efficient monolithic solution algorithm for high-fidelity aerostructural analysis and optimization,” *AIAA Journal*, Vol. 56, No. 3, 2018, pp. 1251-1265, doi: 10.2514/1.J056163.

[48] Z. J. Zhang, S. Khosravi, and D. W. Zingg, “High-fidelity aerostructural optimization with integrated geometry parameterization and mesh movement,” *Structural and Multidisciplinary Optimization*, Vol. 55, No. 4, 2017, pp. 1217-1235, doi: 10.1007/s00158-016-1562-7.

[49] R. Lei, J. Bai, D. Xu, Y. Zhang, and H. Wang, “Speciality assessment of sequential and concurrent aerostructural optimization based on coupled adjoint technique,” *Hangkong Dongli Xuebao/Journal of Aerospace Power*, Vol. 34, No. 5, 2019, pp. 1036–1049, doi: 10.13224/j.cnki.jasp.2019.05.009.

[50] T. Achard, C. Blondeau, and R. Ohayon, “High-fidelity aerostructural gradient computation techniques with application to a realistic wing sizing,” *AIAA Journal*, Vol. 56, No. 11, 2018, pp. 4487-4499, doi: 10.2514/1.J056736.

[51] G. K. W. Kenway, J. R. R. A. Martins, and G. J. Kennedy, “Aerostructural optimization of the common research model configuration,” *AIAA Aviation 2014 -15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2014, doi: 10.2514/6.2014-3274.

[52] G. K. W. Kenway and J. R. R. A. Martins, “Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration,” *Journal of Aircraft*, 2014, pp. 144-160, doi: 10.2514/1.C032150.

[53] B. J. Brelje and J. R. R. A. Martins, “Aerostructural wing optimization for a hydrogen fuel cell aircraft,” *AIAA Scitech 2021 Forum*, 2021, doi: 10.2514/6.2021-1132.

[54] A. C. Gray and J. R. R. A. Martins, “Geometrically nonlinear high-fidelity aerostructural optimization for highly flexible wings,” *AIAA Scitech 2021 Forum*, 2021, doi: 10.2514/6.2021-0283.

[55] J. R. R. A. Martins and N. M. K. Poon, “On Structural Optimization Using Constraint Aggregation,” *Proceedings of 6th World Congress on Structural and Multidisciplinary Optimization*, 2005.

[56] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther, “High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet,” *Journal of Aircraft*, 2004, pp. 523-530, doi: 10.2514/1.11478.

[57] M. A. Akgun, R. T. Haftka, and K. C. Wu, “Sensitivity of lumped constraints using the adjoint method,” *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Vol. 2, 1999, doi: 10.2514/6.1999-1314.

[58] G. J. Kennedy and J. E. Hicken, “Improved constraint-aggregation methods,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 289, 2015, pp. 332-354, doi: 10.1016/j.cma.2015.02.017.

[59] E. J. Nielsen, M. A. Park, “Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design,” *AIAA Journal*, Vol. 44, No.5, 2006, pp. 948–953, doi: 10.2514/1.16052.

[60] K. Stein, T. Tezduyar, and R. Benney, “Mesh moving techniques for fluid-structure interactions with large displacements,” *Journal of Applied Mechanics, Transactions ASME*, Vol. 70, No. 1, 2003, pp. 58-63, doi: 10.1115/1.1530635.

[61] B. T. Helenbrook, “Mesh deformation using the biharmonic operator,” *International Journal for Numerical Methods in Engineering*, Vol. 56, No. 7, 2003, pp. 1007-1021, doi: 10.1002/nme.595.

[62] D. Maruyama, D. Bailly, and G. Carrier, “High-quality mesh deformation using quaternions for orthogonality preservation,” *AIAA Journal*, Vol. 52, No. 12, 2014, pp. 2712-2729, doi: 10.2514/1.J052954.

[63] R. P. Dwight, “Robust mesh deformation using the linear elasticity equations,” *Computational Fluid Dynamics 2006 - Proceedings of the Fourth International Conference on Computational Fluid Dynamics, ICCFD 2006*, 2009, pp. 401–406, doi: 10.1007/978-3-540-92779-2-62.

[64] A. Merle, A. Stück, and A. Rempke, “An adjoint-based aerodynamic shape optimization strategy for trimmed aircraft with active engines,” *35th AIAA Applied Aerodynamics Conference*, 2017, doi: 10.2514/6.2017-3754.

[65] Z. Yang and D. J. Mavriplis, “Unstructured dynamic meshes with higher-order time integration schemes for the unsteady Navier-Stokes equations,” *43rd AIAA Aerospace Sciences Meeting and Exhibit - Meeting Papers*, 2005, doi: 10.2514/6.2005-1222.

[66] A. Beckert and H. Wendland, “Multivariate interpolation for fluid-structure-interaction problems using radial basis functions,” *Aerospace Science and Technology*, Vol. 5, No. 2, 2001, pp. 125-134, doi: 10.1016/S1270-9638(00)01087-7.

[67] X. Liu, N. Qin, and H. Xia, “Fast dynamic grid deformation based on Delaunay graph mapping,” *Journal of Computational Physics*, Vol. 211, No. 2, 2006, pp. 405–423, doi: 10.1016/j.jcp.2005.05.025.

[68] J. A. S. Witteveen and H. Bijl, “Explicit mesh deformation using inverse distance weighting interpolation,” *19th AIAA Computational Fluid Dynamics Conference*, 2009, doi: 10.2514/6.2009-3996.

[69] Y. Wang, N. Qin, and N. Zhao, “Delaunay graph and radial basis function for fast quality mesh deformation,” *Journal of Computational Physics*, Vol. 294, 2015, pp. 149-172, doi: 10.1016/j.jcp.2015.03.046.

[70] G. L. Mura, B. L. Hinchliffe, N. Qin, and J. Brezillon, “Efficient method to eliminate mesh sensitivity in adjoint-based optimization,” *AIAA Journal*, Vol. 55, No. 4, 2017, pp. 1140-1151, doi: 10.2514/1.J055212.

[71] T. C. S. Rendall and C. B. Allen, “Efficient mesh motion using radial basis functions with data reduction algorithms,” *Journal of Computational Physics*, Vol. 228, No. 17, 2009, pp. 6231-6249, doi: 10.1016/j.jcp.2009.05.013.

[72] T. C. S. Rendall and C. B. Allen, “Reduced surface point selection options for efficient mesh deformation using radial basis functions,” *Journal of Computational Physics*, Vol. 229, No. 8, 2010, pp. 2810-2820, doi: 10.1016/j.jcp.2009.12.006.

[73] T. C. S. Rendall and C. B. Allen, “Improved radial basis function fluid-structure coupling via efficient localized implementation,” *International Journal for Numerical Methods in Engineering*, Vol. 78, No. 10, 2009, pp. 1188-1208, doi: 10.1002/nme.2526.

[74] T. C. S. Rendall and C. B. Allen, “Evaluation of RBFs for volume data interpolation in CFD,” *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 7, 2013, pp. 748-769, doi: 10.1002/fld.3766.

[75] S. Jakobsson and O. Amoignon, “Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization,” *Computers and Fluids*, Vol. 36, No. 6, 2007, pp. 1119-1136, doi: 10.1016/j.compfluid.2006.11.002.

[76] G. L. Mura, B. L. Hinchliffe, N. Qin, and J. Brezillon, “Nonconsistent Mesh Movement and Sensitivity Calculation on Adjoint Aerodynamic Optimization,” *AIAA Journal*, Vol. 56, No. 4, 2018, pp. 1541–1553, doi: 10.2514/1.J055904.

[77] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins, “Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations,” *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951, doi: 10.2514/1.J052255.

[78] Y. Shi, C. A. Mader, S. He, G. L. O. Halila, and J. R. R. A. Martins, “Natural laminar-flow airfoil optimization design using a discrete adjoint approach,” *AIAA Journal*, Vol. 58, No. 11, 2020, pp. 4702-4722, doi: 10.2514/1.J058944.

[79] S. Chen, Z. Lyu, G. K. W. Kenway, and J. R. R. A. Martins, “Aerodynamic Shape Optimization of Common Research Model Wing–Body–Tail Configuration,” *Journal of Aircraft*, Vol. 53, No. 1, 2016, pp. 276–293, doi: 10.2514/1.C033328.

[80] A. Merle, and C. Ilic, M. Abu-Zurayk, J. Häßy, R.G. Becker, M. Schulze and T. Klimmek, “High-Fidelity Adjoint-based Aircraft Shape Optimization with Aeroelastic Trimming and Engine Coupling,” *EUROGEN 2019*, 2019, doi: 10.5281/zenodo.4656053.

[81] P. M. Flick, M. H. Love, and P. S. Zink, “The Impact of Active Aeroelastic Wing Technology on Conceptual Aircraft Design,” RTO AVT Specialists’ Meeting, Canada, 1999.

[82] M. Vázquez, A. Dervieux, and B. Koobus, “A methodology for the shape optimization of flexible wings,” *Engineering Computations*, Vol. 23, 2006, pp. 344-367, doi: 10.1108/02644400610661145.

[83] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther, “Complete configuration aero-structural optimization using a coupled sensitivity analysis method,” *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, doi: 10:2514/6.2002-5402.

[84] P. Roehl, D. Mavris and D. Schrage. "Combined aerodynamic and structural optimization of a high-speed civil transport wing," *AIAA 1995-1222. 36th Structures, Structural Dynamics and Materials Conference*, 1995.

[85] F. Sotiropoulos and X. Yang, “Immersed boundary methods for simulating fluid-structure interaction,” *Progress in Aerospace Sciences*, Vol. 65, 2014, pp. 1-21, doi: 10.1016/j.paerosci.2013.09.003.

[86] F. J. Blom, “A monolithical fluid-structure interaction algorithm applied to the piston problem,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 167, No. 3–4, 1998, pp. 369–391, doi: 10.1016/S0045-7825(98)00151-0.

[87] C. Michler, S. J. Hulshoff, E. H. van Brummelen, and R. de Borst, “A monolithic approach to fluid-structure interaction,” *Computers and Fluids*, Vol. 33, No. 5–6, 2004, pp. 839–848, doi: 10.1016/j.compfluid.2003.06.006.

[88] R. Heinrich, “Development and Application of TAU-ANSYS Coupling Procedure,” Springer, Berlin, Heidelberg, 2009, pp. 151–167, doi: 10.1007/978-3-642-04093-1_11.

[89] A. Jameson, “Computational aerodynamics for aircraft design,” *Science*, Vol. 245, No. 4916, 1989, pp. 361-371, doi: 10.1126/science.245.4916.361.

[90] C. D. Coman, “Continuum Mechanics and Linear Elasticity: An Applied Mathematics introduction,” Springer, Dordrecht, 2019, doi: 10.1007/978-94-024-1771-5.

[91] M. J. Turner, C. H. Martin, and R. C. Weikel, “Further Development and Applications of the Stiffness Method,” *14th meeting of the Structures and Materials Panel*, *AGARD*, Paris, 1962.

[92] H. Wendland, “Scattered data approximation”, Cambridge University Press, 2004, doi:10.1017/CBO9780511617539.

[93] A. Beckert, “Coupling fluid (CFD) and structural (FE) models using finite interpolation elements,” *Aerospace Science and Technology*, Vol. 4, No. 1, 2000, pp. 13-22, doi: 10.1016/S1270-9638(00)00111-5.

[94] M. Cordero-Gracia, M. Gómez, and E. Valero, “A radial basis function algorithm for simplified fluid-structure data transfer,” *International Journal for Numerical Methods in Engineering*, Vol. 99, No. 12, 2014, pp. 888-905, doi: 10.1002/nme.4708.

[95] N. Flyer, B. Fornberg, V. Bayona, and G. A. Barnett, “On the role of polynomials in RB-FD approximations: I. Interpolation and accuracy,” *Journal of Computational Physics*, Vol. 321, 2016, pp. 21–38, doi: 10.1016/j.jcp.2016.05.026.

[96] B. Stickan, H. Bleecke, and S. Schulze, “NASTRAN based static CFD-CSM coupling in flowsimulator,” *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Vol. 123, 2013, pp. 223–234, doi: 10.1007/978-3-642-38877-4-15.

[97] N. Nguyen, S. Lebofsky, E. Ting, U. Kaul, D. Chaparro, and J. Urnes, “Development of Variable Camber Continuous Trailing Edge Flap for Performance Adaptive Aeroelastic Wing,” *SAE 2015 AeroTech Congress & Exhibition*, 2015, doi: 10.4271/2015-01-2565.

[98] G. E. C. Fujiwara, N. T. Nguyen, E. Livne, and M. B. Bragg, “Aerostructural design optimization of flexible wing aircraft with continuous morphing trailing edges,” *2018 Multidisciplinary Analysis and Optimization Conference*, 2018, doi: 10.2514/6.2018-3571.

[99] L. Wiart and G. Carrier, “Accounting for wing flexibility in the aerodynamic calculation of transport aircraft using equivalent beam model,” *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference 2010*, 2010, doi: 10.2514/6.2010-9135.

[100] J. Duchon, “Splines minimizing rotation-invariant semi-norms in Sobolev spaces,” *Lecture Notes in Mathematics*, Vol. 571, Springer, Berlin, Heidelberg, 1977, doi: 10.1007/BFb0086566.

[101] M. M. Selim and R.P. Koomullil, “Mesh Deformation Approaches – A Survey,” *Journal of Physical Mathematics*, Vol. 7, No. 2, 2016, pp. 1-9, doi: 10.4172/2090-0902.1000181.

[102] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats, “*An Assessment of Airfoil Design by Numerical Optimization*,” *NASA Technical Memorandum X-3092*, 1974.

[103] R. M. Hicks and G. N. Vanderplaats, “Application of numerical optimization to the design of supercritical airfoils without drag-creep,” *SAE Technical Paper 770440*, 1977.

[104] S. U. Rehman, M. Langelaar, and F. van Keulen, “Efficient Kriging-based robust optimization of unconstrained problems,” *Journal of Computational Science*, Vol. 5, No. 6, 2014, pp. 765-773, doi: 10.1016/j.jocs.2014.04.005.

[105] O. Chernukhin and D. W. Zingg, “Multimodality and global optimization in aerodynamic design,” *AIAA Journal*, Vol. 51, No. 6, 2013, pp. 1342-1354, doi: 10.2514/1.J051835.

[106] N. P. Bons, X. He, C. A. Mader, and J. R. R. A. Martins, “*Multimodality in aerodynamic wing design optimization*,” AIAA Journal, Vol. 57, No. 3, 2019, pp. 1004-1018, doi: 10.2514/1.J057294.

[107] G. R. Anderson, M. J. Aftosmis, and M. Nemec, “Constraint-based Shape Parameterization for Aerodynamic Design,” *Seventh International Conference on Computational Fluid Dynamics*, 2012.

[108] C. B. Allen, D. J. Poole, and T. C. S. Rendall, “Efficient modal design variables applied to aerodynamic optimization of a modern transport wing,” *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, doi: 10.2514/6.2016-3215.

[109] K. Bobrowski, E. Ferrer, E. Valero, and H. Barnewitz, “Aerodynamic Shape Optimization Using Geometry Surrogates and Adjoint Method,” *AIAA Journal*, Vol. 55, 2017, pp. 1–14, doi: 10.2514/1.J055766.

[110] H. Barnewitz, “Flexible Wing Optimisation Based on Shapes and Structures,” *MEGADESIGN and MegaOpt - German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design. Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Springer, Berlin, Heidelberg, Vol. 107, 2009, pp. 287-305, doi: 10.1007/978-3-642-04093-1_20.

[111] T. Bui-Thanh, M. Damodaran, and K. Willcox, “Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition,” *AIAA Journal*, Vol. 42, No. 8, 2004, pp. 1505–1516, doi: 10.2514/1.2159.

[112] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and Analysis of Computer Experiments,” *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423, doi: 10.1214/SS/1177012413.

[113] R. Pinnau, “Model Reduction via Proper Orthogonal Decomposition,” *Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry*, Springer, Berlin, Heidelberg, Vol. 13, 2008, pp. 95–109, doi: 10.1007/978-3-540-78841-6_5.

[114] R. Zimmermann and S. Görtz, “Non-linear reduced order models for steady aerodynamics,” *Procedia Computer Science*, Vol. 1, No. 1, 2010, pp. 165–174, doi: 10.1016/j.procs.2010.04.019.

[115] J. Brezillon, O. Brodersen, R. Dwight, A. Ronzheimer, and J. Wild, “Development and Application of a Flexible and Efficient Environment for Aerodynamic Shape Optimisation.” *Proceedings of the ONERA-DLR Aerospace Symposium*, 2006.

[116] J. Brezillon, R. P. Dwight, and M. Widhalm, “Aerodynamic Optimization for Cruise and High-Lift Configurations,” *Megadesign and Megaopt - German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, 2009, pp. 249-262, doi: 10.1007/978-3-642-04093-1_18.

[117] E. J. Nielsen and M. A. Park, “Using an adjoint approach to eliminate mesh sensitivities in computational design,” AIAA Journal, Vol. 44, No. 5, 2006, pp. 948–953, doi: 10.2514/1.16052.

[118] M. Widhalm, J. Brezillon, Č. Ilić, and T. Leicht, “Investigation on adjoint based gradient computations for realistic 3d aero-optimization,” *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference 2010*, 2010, doi: 10.2514/6.2010-9129.

[119] M. Barcelos, H. Bavestrello, and K. Maute, “A Schur-Newton-Krylov solver for steady-state aeroelastic analysis and design sensitivity analysis,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 17–18, 2006, pp. 2050-2069, doi: 10.1016/j.cma.2004.09.013.

[120] B. M. Irons and R. C. Tuck, “A version of the Aitken accelerator for computer iteration,” *International Journal for Numerical Methods in Engineering*, Vol. 1, No. 3, 1969, pp. 275-277, doi: 10.1002/nme.1620010306.

[121] “GitHub - dr-bill-c/MYSTRAN: MYSTRAN is a general purpose finite element analysis solver.” [online]. Available: https://github.com/dr-bill-c/MYSTRAN.

[122] C. G. Broyden, “A class of methods for solving nonlinear simultaneous equations,” *Mathematics of Computation*, Vol. 19, No. 92, 1965, p. 577, doi: 10.1090/s0025-5718-1965-0198670-6.

[123] K. Maute, M. Nikbay, and C. Farhat, “Analytically based sensitivity analysis and optimization of nonlinear aeroelastic systems,” *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, doi: 10.2514/6.2000-4825.

[124] K. Maute and M. Allen, “Conceptual design of aeroelastic structures by topology optimization,” *Structural and Multidisciplinary Optimization*, Vol. 27, No. 1–2, 2004, pp. 27-42, doi: 10.1007/s00158-003-0362-z.

[125] Y. Saad, “Iterative Methods for Sparse Linear Systems,” SIAM, Philadelphia, 2003, doi: 10.1137/1.9780898718003.ch4.

[126] R. Bagnara, “Unified proof for the convergence of Jacobi and Gauss-Seidel methods,” SIAM Review, Vol. 37, No. 1, 1995, pp. 93–97, doi: 10.1137/1037008.

[127] A. Taylor III, V. Korivi, and G. Hou, “Sensitivity analysis applied to the Euler equations - A feasibility study with emphasis on variation of geometric shape,” *AIAA 1991-173. 29th Aerospace Sciences Meeting*, 1991, doi: 10.2514/6.1991-173.

[128] A. Jameson, W. Schmidt, F. R. G. Fredrichshafen, and E. Turkel, “Numerical Solution of the Euler Equations by Finite Volume Methods Schemes,” *AIAA 14th Fluid and Plasma Dynamic Conference*, 1981, doi: http://dx.doi.org/10.2514/6.1981-1259.

[129] G. Carpentieri, B. Koren, and M. J. L. van Tooren, “Adjoint-based aerodynamic shape optimization on unstructured meshes,” *Journal of Computational Physics*, Vol. 224, No. 1, 2007, pp. 267-287, doi: 10.1016/j.jcp.2007.02.011.

[130] Z. Lyu, G. K. W. Kenway, and J. R. R. A. Martins, “Aerodynamic shape optimization investigations of the common research model wing benchmark,” *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968-985, doi: 10.2514/1.J053318.

[131] S. N. Skinner and H. Zare-Betash, “Aerodynamic design optimisation of non-planar lifting surfaces,” *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016, doi: 10.2514/6.2016-0164.

[132] P. Iannelli, F. Moens, M. Minervino, R. Ponza, and E. Benini, “Comparison of optimization strategies for high-lift design,” *Journal of Aircraft*, Vol. 54, No. 2, 2017, pp. 642-658, doi: 10.2514/1.C033648.

[133] L. M. B. C. Campos and J. M. G. Marques, “On a Method of Lagrange Multipliers for Cruise Drag Minimization,” *Journal of Aerospace Engineering and Mechanics*, Vol. 5, No. 1, 2021, pp. 348-366, doi: 10.36959/422/450.

[134] B. K. Stanford, “Optimization of an aeroservoelastic wing with distributed multiple control surfaces,” *Journal of Aircraft*, Vol. 53, No. 4, 2016, pp. 1131-1144, doi: 10.2514/1.C033613.

[135] H. S. Tang, S. C. Jones, and F. Sotiropoulos, “An overset-grid method for 3D unsteady incompressible flows,” *Journal of Computational Physics*, Vol. 191, No. 2, 2003, pp. 567-600, doi: 10.1016/S0021-9991(03)00331-0.

[136] W. W. Hager and H. Zhang, “A Survey of Nonlinear Conjugate Gradient Methods,” *Pacific Journal of Optimization*, Vol. 2, 2006, pp. 35-58.

[137] P. E. Gill and W. Murray, “Quasi-Newton Methods for Unconstrained Optimization,” *Journal of the Institute of Mathematics and its Applications*, Vol. 9, pp. 91–108, 1972.

[138] P. E. Gill and W. Murray, “Newton-type methods for unconstrained and linearly constrained optimization,” *Mathematical Programming*, Vol. 7, No. 1, 1974, pp. 311-350, doi: 10.1007/BF01585529.

[139] W. C. Davidon, “Variable Metric Method for Minimization,” *SIAM Journal on Optimization*, Vol. 1, No. 1, 1991, pp. 1–17, doi: 10.1137/0801001.

[140] J. Nocedal and S. J. Wright, “Numerical Optimization”, Springer, New York, 1999.

[141] M. T. Landahl and E. Mollo-Christensen, “Turbulence and Random Processes in Fluid Mechanics,” *Turbulence and Random Processes in Fluid Mechanics*, Cambridge University Press, 1992, doi: 10.1017/9781139174008.

[142] P. Moin and J. Kim, “Tackling turbulence with supercomputers,” *Scientific American*, Vol. 276, No. 1, 1997, pp. 62-68.

[143] P. R. Spalart and S. R. Allmaras, “One-equation turbulence model for aerodynamic flows,” *Recherche aerospatiale*, 1994, pp. 5–21, doi: 10.2514/6.1992-439.

[144] C. L. Rumsey, “Apparent transition behavior of widely-used turbulence models,” *International Journal of Heat and Fluid Flow*, Vol. 28, No. 6, 2007, pp. 1460-1471, doi: 10.1016/j.ijheatfluidflow.2007.04.003.

[145] S. R. Allmaras, F. T. Johnson, and P. R. Spalart, “Modifications and clarifications for the implementation of the spalart-allmaras turbulence model,” *7th International Conference on Computational Fluid Dynamics*, ICCFD 2012, 2012.

[146] G. L. Mura, “Mesh Sensitivity Investigation in the Discrete Adjoint Framework,” *Ph.D. Dissertation*, University of Sheffield, 2017.

[147] S. Kim, J. J. Alonso, and A. Jameson, “A gradient accuracy study for the adjoint-based navier-stokes design method,” *37th Aerospace Sciences Meeting and Exhibit*, 1999, doi: 10.2514/6.1999-299.

[148] Y. Saad and M. H. Schultz, “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856-869, doi: 10.1137/0907058.

[149] M. Lamot and B. Zalik, “An overview of triangulation algorithms for simple polygons,” *Proceedings of the International Conference on Information Visualisation*, 1999, doi: 10.1109/IV.1999.781552.