Aerostructural Assessment Approach of a Transonic Morphing Wing in 3DExperience

Lizzy Middendorp

TU**Delft**

A one-way coupled aerodynamic - structural assessment approach of a span-wise trailing edge morphing wing on a transonic transport aircraft in 3DExperience

by

Lizzy Middendorp

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday, March 7th, 2024, at 9.30.

Student number: 4559649
Thesis committee: dr. ir. R. den Breuker,   TU Delft, Daily supervisor
                    D.M.J. Peeters   TU Delft, Chair
                    R. La Rocca   TU Delft, Examiner
                    J. Sodja   TU Delft, Additional

TUDelft

Summary

Innovative morphing wings technology has the capacity to greatly enhance aircraft capabilities. However, its intricate relationship between aerodynamics and structural integrity necessitates an aerostructural analysis. Dassault Systèmes' 3DExperience offers aComputer-Aided Design (CAD)-based software with integrated tools for both structural and aerodynamic analysis. This can streamline the design process by eliminating the need for separate input and output files for various programs.

This thesis delves into the aerostructural analysis of a transonic transport aircraft, specifically examining the span-wise trailing edge morphing wing within the 3DExperience software. Given the lack of an aerostructural framework within the software, this study focuses on a one-way coupled aerodynamic to structural analysis. To achieve this, the Common Research Model (CRM) and undeflected Common Research Model (uCRM) serve as benchmark models for the methodology.

The airfoil's trailing edges at four positions on the outboard wing are altered to achieve the desired morphing effect. This is accomplished by combining the PARSEC parameterisation method and 3DExperience's spline creation capabilities. The implemented Computational Fluid Dynamics (CFD) solver utilises the Reynolds-Averaged Navier-Stokes (RANS) equations and incorporates three different turbulence models. However, it should be noted that the student license of 3DExperience has a 5 million node limitation on the aerodynamic model, which affects the domain size and boundary layers in the mesh. A linear analysis is employed in terms of structural analysis, and the implemented solver is based on Abaqus. The wing fuel tank is divided into six sub-tanks to model the fuel accurately. Engine load is represented as a point load and is transferred to the surrounding ribs. Finally, spatial varying mapped data with search tolerance is utilised to implement the aerodynamic loads on the structural mesh.

The validity of the aerodynamic and structural models has been established through verification with the baseline geometry. However, the mesh convergence study has revealed that the current node limit is insufficient. This is supported by the normalised wall distance, which should ideally be around 1 for transonic conditions and the shortcomings in the automatic meshing. Initial analysis of the structural mesh with a basic tip load demonstrated some level of convergence, but it was not fully achieved when the node limit of 1 million nodes was reached. Furthermore, the mesh convergence study of all loads highlighted some unusual behaviour.

In order to evaluate the impact of a morphing trailing edge on an unmorphed baseline wing, two test cases were utilised to assess the methodology: a high-lift wing and an efficient wing in terms of lift over drag. The parameters selected for these tests were based on transonic design principles, but are not the final optimised solution.

The results show that the high-lift wing has an increased lift and drag coefficient and decreased lift-to-drag ratio than the unmorphed baseline wing. In contrast, The efficient wing shows a decreased lift and drag coefficient, but an increased lift-to-drag ratio. Upon closer examination of the aerodynamic outcomes, it was observed that the outboard changes had an impact on the flow of the inboard wing, and that the morphing of the trailing edge affected the flow in front of the aft spar.

The structural results show the correlation between the lift coefficient and the maximum displacement in z - direction. Overall, it is apparent that the engine and fuel tank have a more pronounced impact on the twist distribution at lower lift coefficient wings.

The present research exhibits the preliminary stages of an aerostructural analysis carried out within the 3DExperience environment. While a promising start, this effort represents a small contribution towards the practical implementation of span-wise trailing edge morphing wings in transonic aircraft.

Acknowledgements

Throughout my bachelor's degree in Aerospace Engineering, I never thought I would make it this far. However, with an inspiring and sometimes challenging mindset, I managed to persevere to the end. While I will miss the faculty's environment and the people, I am also excited about a new challenge in a different place. But I couldn't have done it without the help of the people who supported me along the way.

I would like to express my gratitude to Roeland den Breuker for providing me with the opportunity to work on this thesis. I would also like to thank him for sharing his knowledge with me during our meetings. Additionally, I want to thank Daniel Atherstone, who assisted me on numerous occasions with my 3DExperience struggles. Without your help, I wouldn't have come this far. I am also grateful to Panos Georgopoulos for his constructive feedback, tips, and a little emotional support during this thesis. Lastly, I want to thank Jurij Sodja for his flexibility and insightful feedback during the green-light meeting.

I would also like to thank the students from NB1.07 who kept me company and provided a listening ear during this time. Lastly, I want to express my appreciation to my partner, friends, and family for being patient with me throughout this period.

As I end this chapter, a new one awaits, and I can't wait to see what's out there.

Lizzy Middendorp
Delft, February 2024

Contents

List of Figures

List of Tables

List of Abbreviations

ACTE   Adaptive Compliant Trailing Edge
ADIF   Adaptive Wing
AFW   Active Flexible Wing
AMG   Algebraic Multi-grid
BiCGStab Biconjugate gradient stabilized
CAD   Computer-Aided Design
CFD   Computational Fluid Dynamics
CHANGE Combined morphing assessment software using flight envelope data and mission-based morphing prototype wing development
CRM   Common Research Model
CST   Class Shape Transformation
EC   European Commission
EKL   Enterprise Knowledge Language
EOS   Equation of State
EU   European Union
FEA   Finite Element Analysis
FFD   Free Form Deformation
FGMRES Flexible Generalised Minimum Residual
HDM   Hex-Dominant Mesher
MAS   Morphing Aircraft Structures
MAW   Mission Adaptive Wing
MAWS   Modelling of Adaptive Wing Structures
NACRE New Aircraft Concepts Research
NASA   National Aeronautics and Space Administration
RANS   Reynolds-Averaged Navier-Stokes
SA   Spalart-Allmaras
SADE   Smart High Lift Devices for Next Generation Wings
SARISTU Smart Intelligent Aircraft Structures
SIMPLE   Semi-Implicit Method for Pressure Linked Equations)
SST   Shear-Stress Transport
uCRM   undeflected Common Research Model
UDF   Unser Defined Feature

List of Symbols

A	Wing Area		p	stiffener pitch
C	Chord		q	local heat flux
C_D	Drag coefficient		r_{LE}	leading edge radius
C_L	lift coefficient		t	time
C_f	fricton coefficient		v	velocity
C_t	Tip Chord		v_τ	friction velocity
C_{SoB}	Side of Body chord		w_{st}	stiffener width
C_{r_0}	Root chord of the trapezoidal wing		$y+$	normalised distance from the wall
C_y	Yehudi Chord		y_w	wall-distance
E	total energy		Λ	1/4 chord sweep
L	Length of geometry at the symmetry axis		α_{TE}	trailing edge direction
L_r	Length of the rib between the spars		β_{TE}	trailing edge wedge angle
R	specific gas constant		δ	boundary layer thickness
S	strain rate		ϵ	turbulent dissipation rate
$U+$	friction velocity		γ	specific heat ratio
X_{LO}, Z_{LO}	lower crest location		κ	von Kármán constant
X_{UP}, Z_{UP}	upper crest location		λ	Taper Ratio
Z_{TE}	trailing edge coordinate		μ	dynamic viscosity
$Z_{XX_{UP}}, Z_{XX_{LO}}$	upper and lower curvature		μ_t	Eddy Viscosity
ΔZ_{TE}	trailing edge wedge thickness		ν	turbulent viscosity
b	span		ω	specific rate of dissipation
b_{SoB}	Side of Body span		ρ	density
h_{st}	stiffener height		σ	stress tensor
k	turbulent kinetic energy		τ_{ij}	Reynolds stress tensor
p	pressure		τ_w	wall shear stress
			δ_{ij}	Kronecker delta

<div style="text-align: right">

1

</div>

Introduction

The aviation industry, a vital component of the global economy, has long been recognised as a major emitter of greenhouse gases. These emissions, released directly into the upper atmosphere where their impact is amplified, play a crucial role in the complex web of climate change. Aviation is responsible for approximately 2,5% of global CO_2 emissions[1]. The European Commission (EC) has set ambitious goals to reduce net greenhouse gas emissions by 55 % by 2030 and aims to achieve a remarkable 90 % reduction in transport emissions by 2050[2]. This situation presents a compelling challenge: enhancing the efficiency of aircraft models to mitigate their climate impact effectively. One promising technology for aviation is morphing.

1.1. Wing Morphing

Morphing comes from the Greek word 'morphos', which means shape[3]. Wing morphing allows for dynamic shape and structural changes often in the wing, adapting to various flight conditions with conflicting requirements. This innovative approach has been present since the early days of aircraft development, with the Wright brothers using twist morphing for roll stability[4] and Otto Lilienthal utilising variable camber in his glider design[5]. Multiple aircraft with morphing capabilities were developed between 1985 and 1990, as shown in Table 1.1.

Table 1.1: Morphing concepts between 1895 and 1990[5, 6].

Year	Aircraft	Morphing Concept	Year	Aircraft	Morphing Concept
1895	Otto Lilienthal Glider	variable camber	1964	XB-70 Valkyrie	span morphing
1903	Wright Brothers' flyer	twist morphing	1966	SU 17 IG	variable sweep
1920	Parker variable wing	variable camber	1967	MiG-23	variable sweep
1931	Pterodactyl IV	variable sweep	1967	SU 24	variable sweep
1931	MAK-10	span morphing	1969	Tupolev TU-22M	variable sweep
1932	Nikitin-Schevchenko IS-1	bi- to monoplane	1970	F-14 Tomcat	variable sweep
1937	Bakshaev RK/LIG-7	variable chord	1972	FS-29 Glider	span morphing
1944	Messerschmitt P.1101	variable sweep	1974	B-1 Lancer	variable sweep
1947	MAK-123	span morphing	1974	Panavia Tornado	variable sweep
1951	Bell X 5	variable sweep	1981	TU 160	variable sweep
1952	Grumman XF10F Jaguar	variable sweep	1985	AFTI/F-111 MAW	variabie sweep & camber
1964	F111 Aardvark	variable sweep			

The research projects towards morphing wings began in the 1980s when National Aeronautics and Space Administration (NASA) initiated both the Active Flexible Wing (AFW)[7] and the Mission Adaptive Wing (MAW) program[8]. In 1995, the German Aerospace Center DLR launched research on morphing concepts in Europe with the Adaptive Wing (ADIF) project[9]. Numerous subsequent projects followed, including the Morphing Aircraft Structures (MAS) project, the Aircraft Morphing program[10], the Smart

Wing Program[11], the New Aircraft Concepts Research (NACRE) project[12], the Smart Fixed Wing Aircraft project[13], Smart High Lift Devices for Next Generation Wings (SADE) project[14], Modelling of Adaptive Wing Structures (MAWS) project, the Smart Intelligent Aircraft Structures (SARISTU) project and the Combined morphing assessment software using flight envelope data and mission-based morphing prototype wing development (CHANGE) project. The European Union (EU)'s most recent research program, Clean Sky 2, includes three morphing concepts for a regional aircraft: a morphing droop nose concept, a multifunctional flap, and an adaptive winglet[15]. In 2014, NASA collaborated with FlexSys Inc. to test their FlexFoil Adaptive Compliant Trailing Edge (ACTE) system on a Gulfstream GIII to demonstrate the technology's structural feasibility and resilience[16].

1.1.1. Morphing Concepts

Table 1.1 shows that morphing can occur on various aspects of the wing, affecting its performance. Breuker et al.[5] classified morphing into local and global morphing. Local morphing involves altering the wing's cross-sectional shape, primarily affecting the aerodynamic properties of the airfoil. Airfoil camber and thickness modifications are two examples of local morphing. Global morphing refers to alterations made throughout the entire wing span that impact the overall wing configuration. This is accomplished by segmenting the wing into multiple parts that can morph individually or in coordination. For instance, global morphing includes wing sweep or wing twist. Figure 1.1 shows the difference between local and global morphing.
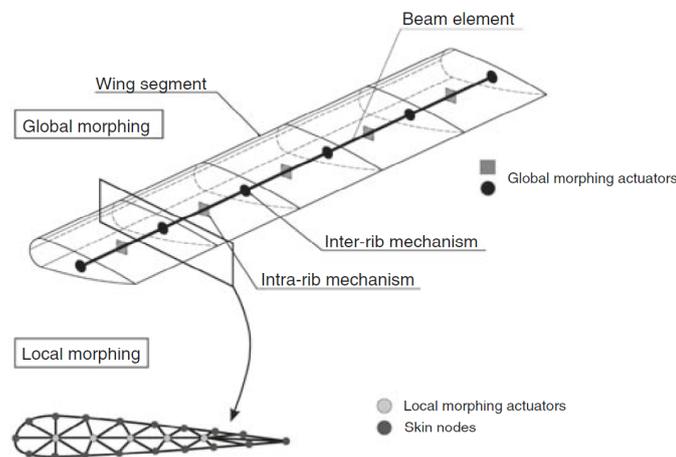
Figure 1.1: Local and Global morphing on a wing[5]

One can employ local morphing techniques, such as span-wise camber morphing, across the entire wing to merge the two morphing types. The trailing edge is particularly interesting for morphing in terms of aerodynamics and structure, as it is commonly used for flaps and ailerons. As a result, many studies have focused on developing and analysing trailing edge morphing concepts[17, 18, 19].

1.1.2. Analysis of a Morphing Wing

Two primary methods exist for developing a model capable of assessing the performance of a trailing edge morphing wing. The first method starts by creating a morphing mechanism, while the second technique uses a shape analysis. The shape analysis explores a broader design and enables the development of a tailored morphing mechanism. This method employs an iterative geometry process to determine the optimal design that meets the specifications. This requires the geometry to be easily modifiable.

The main goal of a morphing wing is to minimise fuel usage by enhancing the wing's aerodynamic efficiency. However, morphing wings establish an intricate relationship between aerodynamics and structural integrity. Consequently, an analysis of morphing wing performance demands integrating a coupled aerodynamic and structural analysis, also referred to as aerostructural analysis.

1.2. Aerostructural Analysis

As mentioned in the previous section, evaluating the performance of a trailing edge morphing wing requires conducting an aerostructural analysis. This type of analysis evaluates how the aerodynamic loads impact the structural design of an aircraft's wings while also considering how the structural response affects the aerodynamic loads. It is crucial to conduct a comprehensive analysis from both perspectives to establish the connection between aerodynamic forces and the system's structural response. This can be achieved by solving the governing equations simultaneously, or by coupling the aerodynamic and structural analysis. The latter method is commonly used as it allows for the integration of existing flow solvers and structural solvers.

To perform aerostructural analysis, solvers with different levels of fidelity are employed, ranging from low to high. Although low-fidelity solvers are generally faster and require less computational power than high-fidelity solvers, they may not be able to capture complex phenomena. Studies have shown that even minor changes in a wing's shape can significantly affect its aerodynamic performance, especially in the transonic regime[20]. Therefore, it is necessary to use aerodynamic solvers with sufficient fidelity to accurately capture the effects of design changes.

A commonly used high-fidelity method to analysing aerodynamics is using CFD. CFD employs mathematical models and numerical techniques to replicate fluid flows and resolve the Navier-Stokes equations that dictate fluid movement. These equations outline the preservation of mass, momentum, and energy in a fluid.

For structural analysis, Finite Element Analysis (FEA) is commonly used. FEA divides a complex system into smaller, simpler elements, allowing for simulation and analysis of its behaviour under various conditions. The number and size of the elements describe the level of detail of the simulation.

The coupling method is required when between the aerodynamic and structural results for the aerostructural analysis. Both aerodynamic and structural analysis use meshes to represent the geometry. When the meshes share nodes, connecting them is straightforward. However, as the meshes have different requirements, they often do not match. de Boer et al. [21] have reviewed coupling methods on non-matching meshes, such as nearest neighbour and radial basis function. Nonetheless, the compatibility and effectiveness of the coupling method are greatly influenced by the software and input-output files being used.

1.2.1. Software

Nowadays, multiple software programs can perform individual structural or aerodynamic analyses. However, this can make the process more complex as it requires manually rewriting output files and connecting various tools. A promising CAD-based software is 3DExperience by Dassault Systèmes. CAD-based modelling is the industry standard for creating geometric models and offers many advantages, such as automated documentation of changes and tracking of parameters. This is particularly relevant for this thesis, which focuses on changing the trailing edge shape. The software also includes both integrated structural FEA and aerodynamic CFD analysis tools. Kanaparthi et al.[22] have demonstrated the software's potential by performing a multidisciplinary optimization of a wing using 3DExperience.

1.3. Research Objective

The previous sections show that the 3DExperience software has the potential to assess the performance of trailing edge morphing wings. However, given the current lack of a fully coupled aerodynamic and structural analysis framework within the software, this thesis will instead focus on a one-way coupled aerostructural analysis, as shown in Figure 1.2. For the geometry, this thesis focuses a transonic transport aircraft due to its widely utilised in the aviation industry, as well as the accessibility of the CRM. This results in the following research objective :

"to develop a one-way coupled aerodynamic and structural assessment method in 3DExperience to analyse a span-wise trailing edge morphing wing on a transonic transport aircraft."
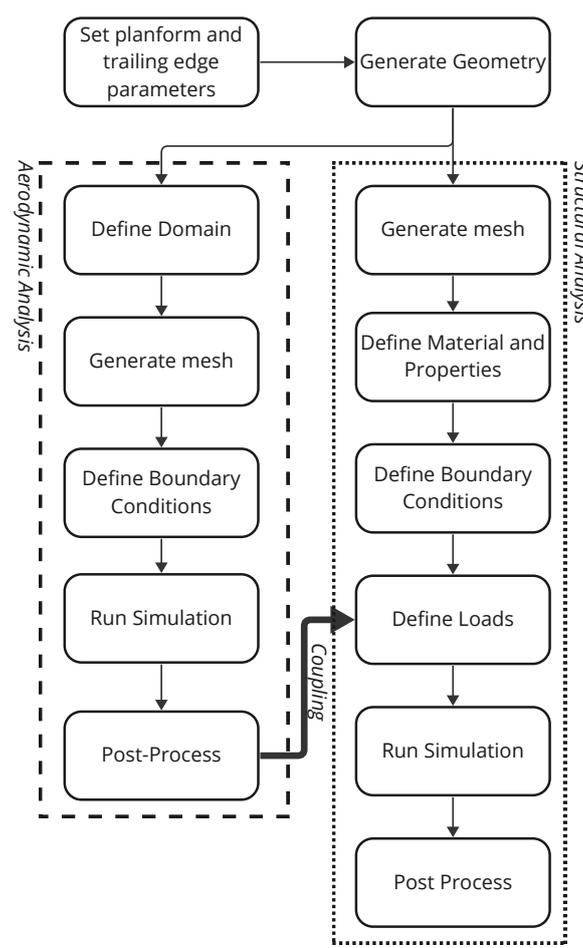
Figure 1.2: The one-way coupled aerostructural analysis framework

The research is divided into two main parts to achieve this objective. The first part focuses on creating a method capable of assessing the performance of a trailing edge morphing wing in 3DExperience. This results in the following research questions:

1. How can modifiable wing geometry be created in 3DExperience?
2. How can a span-wise trailing edge morphing wing be modelled in 3DExperience?
3. How can the 3DExperience platform be used to perform aerodynamic and structural analysis?
4. How can the structural and aerodynamic analysis be coupled?
5. What are the limitations of 3DExperience?

The second part focuses on the results of assessing a trailing edge morphing wing. The thesis uses two test cases of morphing trailing edge cases to compare the influence on the performance of the baseline wing. This results in the following research questions:

6. What are the effects of the morphing wing on the drag and aerodynamic efficiency?
7. What are the local effects of the morphing wing on the flow over the airfoil?
8. How does the morphing affect the deformation of the wing in terms of displacement and twist?

1.4. Report Outline

The report starts with the methodology. In chapter 2, the approach used to analyse the morphing wings in 3DExperience is outlined, with a focus on the first part of the research. This section explains the setup of the geometrical, aerodynamic, and structural models, including a discussion of the two test cases.

chapter 3 delves into the verification of the methodology, which involves a mesh convergence study of both the aerodynamic and structural model, as well as a quality analysis of the aerodynamic mesh. Moving forward, chapter 4 provides an in-depth analysis of the results, with a particular emphasis on the second part of the research. Finally, the thesis is concluded in chapter 5, where recommendations are also presented.

2

Methodology

This chapter discusses the methodology of the thesis. The methodology consists of a geometrical, aerodynamic, and structural model. The whole methodology is developed within the limits of 3DExperience with an academic license. The limits of this license will be discussed in the corresponding sections.

2.1. Geometrical Model

3DExperience Apps: Generative Shape Design, Engineering Rules Capture

3DExperience has been used as the industry standard to make CAD-based geometrical models. It has been utilised for its seamless transitions and robust capabilities in the creation of smooth, intricate geometries. With the ability to generate parameter sets, 3DExperience enables the expansion of designs into the realm of parametric design. This capability streamlines the creation of intricate geometries and facilitates dynamic modifications. This section outlines the methodology behind the geometrical model and starts with the benchmark model and planform. Utilizing a CAD-based modelling approach, the structural elements essential for structural analysis are explicitly modelled, which are outlined in this section. Finally, the airfoil parameterisation required to achieve the trailing edge morphing is discussed. The trailing edge morphing is only applied to the outboard wing due to the common presence of an inboard flap.

2.1.1. Benchmark Model

This thesis focuses on the analysis of a transonic transport aircraft. In the realm of aircraft design, opting for a benchmark model is a widespread strategy employed to sidestep the need to create an aircraft entirely from the ground up. The CRM is a geometric representation of a generic transport aircraft, specifically designed for research purposes[23]. The CRM, developed through a collaboration between NASA and Boeing, derives its geometry from the Boeing 777 and is specifically tailored for aerodynamic analysis. This means that the wing is deflected in its $1-g$ flight condition. Brooks et al.[24] introduced the uCRM to enhance its functionality for aeroelastic analysis. The uCRM features an undeflected wing that allows for in-depth exploration of the interaction between aerodynamics and structural dynamics. Due to the research objective, the uCRM is chosen as the benchmark model and will be used as a basis for developing the geometrical model.

The geometrical model in this thesis consists of a fuselage-wing-tail configuration without an engine geometry. The fuselage geometry consists of a half model and is directly imported into 3DExperience with a .stp file. The geometric configuration is modified to achieve a complete enclosure along the symmetry axis, thereby eliminating any holes or openings within the structure. The wing geometry is rebuilt in 3DExperience to be able to be easily modified. This process leverages the parameters and airfoil specifications inherent to the uCRM.

Figure 2.2: The Planform Parameterisation
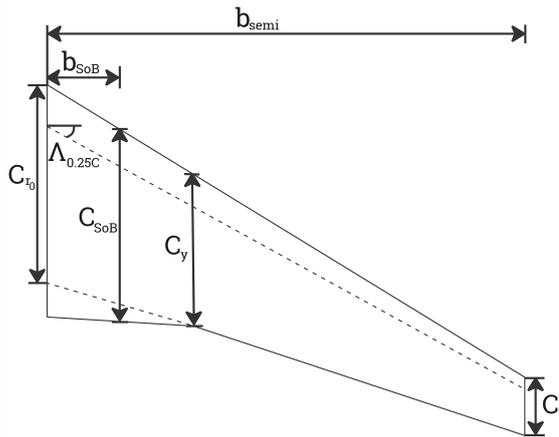
Table 2.1: uCRM planform specifications[24]

Parameter	Symbol	Value
span [m]	b	59.29
Side of Body chord [m]	C_{SoB}	11.92
Yehudi chord [m]	C_y	7.26
Tip chord [m]	C_t	2.736
Taper ratio [-]	λ	0.275
1/4 chord sweep [deg]	Λ	35

2.1.2. Planform parameterization

The wing planform is defined using the given specification parameters of the uCRM wing by Brooks et al.[24]. The resulting planform parameterisation is shown in Figure 2.2. By implementing the values presented in Table 2.1, the uCRM wing shape can be replicated. This method is versatile and can accommodate a range of other designs as well as shown in Figure 2.1.
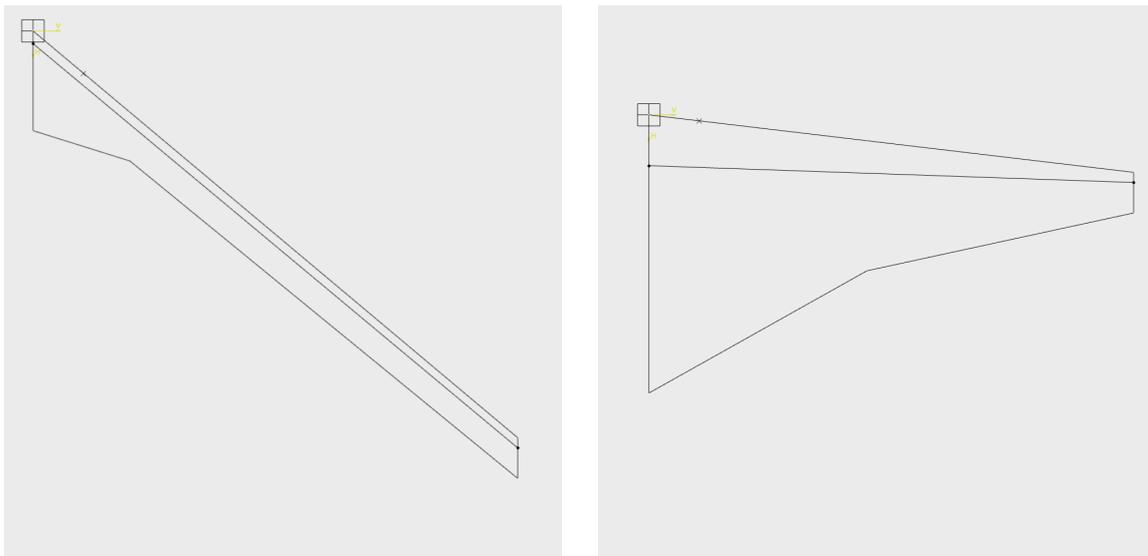
(a) A slender high sweep, low aspect ratio planform

(b) A high sweep, high aspect ratio planform

Figure 2.1: Demonstration of the modifiability of the planform parameterisation

The root chord of the trapezoidal wing,C_{r_0} is calculated with the taper ratio, λ, using:

$$C_{r_0} = \lambda \cdot C_t \tag{2.1}$$

The wing is modelled from the wing-fuselage intersection or Side of Body to the tip. The Side of Body span, b_{SoB} is determined using the geometry of the fuselage of the uCRM. This involves scaling the fuselage geometry using the Side of Body chord, C_{SoB}, at the wing-fuselage intersection. To ensure a smooth transition to the wing planform, b_{SoB} is increased by 20mm, and a blended surface is added between the fuselage and wing planform. When comparing the wing to the original uCRM wing, it

was found that the scaled model had a slightly increased span. As a result, the span is increased accordingly to achieve a comparable wing.

To create the shape of the wing, airfoils from the uCRM are placed at eight locations along the span, labelled as P1 to P8. The wing is then split at the Yehudi into an inboard and outboard section. The inboard wing is divided into two sections, while the outboard wing is divided into five sections. The intersections of these sections are used to implement the uCRM airfoils. Figure 2.3 shows wing planform and the eight positions where the airfoils are implemented.
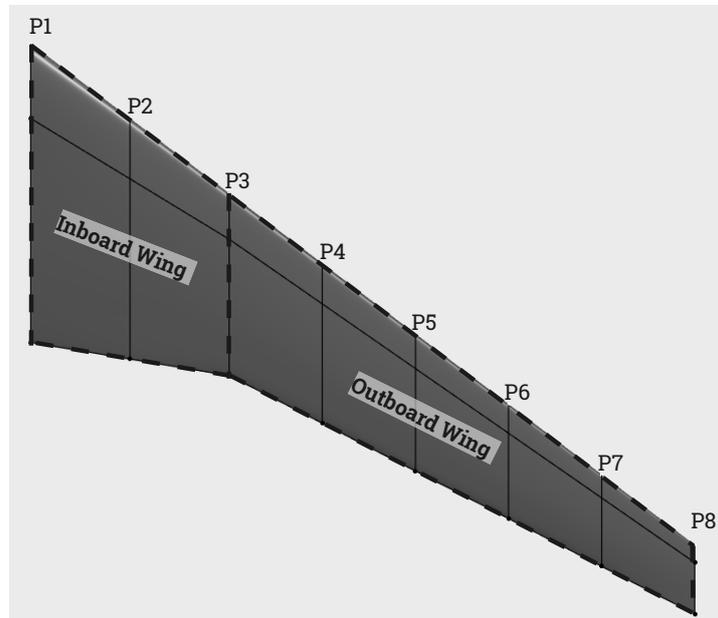
Figure 2.3: The planform sections

2.1.3. Structural Sizing

The geometrical model for structural analysis includes ribs, spars, and stiffeners, all of which are explicitly modeled using surface geometries. According to the design specifications of the uCRM[25], the front spar is located at 10% of the Side of Body and 35% of the tip chord, while the aft spar is situated at 60% of their respective chords at the Side of Body and tip.

Reoccurring structural components such as ribs and stiffeners can be efficiently modelled through the automation features available in 3DExperience. By utilising Unser Defined Feature (UDF) as templates, designers can create complex geometric features by combining and modelling existing features. The Engineering Rules Capture app allows for the creation of Knowledge Patterns using Enterprise Knowledge Language (EKL)-language. As the EKL-language is not a common used programming language, 3DExperience has an integrated support tool with examples and a search function. By using Knowledge Patterns to automatically generate UDFs, structural elements can be automatically generated.

Rib Generation

Wing ribs have a fundamental role in shaping and maintaining the cambered profile of the wing, transferring aerodynamic loads from the skin and stringers to the spars, and providing stability against panel buckling. Ribs also transfer local loads to the structure such as the fuel in the wings, engines, etc. The ribs are positioned continuously from the leading edge to the trailing edge within the inboard wing. In the outboard wing, trailing edge morphing capabilities are present, resulting in the extension of ribs from the leading edge to the aft spar. To ensure the proper alignment of ribs, they are configured to be perpendicular to the front spar, which serves as the guideline for the rib pitch. An UDF is created by using a point on the front spar, the rib start point, and by slicing the wing perpendicular to the front spar. Figure 2.4 shows the UDF of the rib. Automatic initiation of all ribs can be defined in a Knowledge Pattern by changing the rib start point. The uCRM wing consists of 43 ribs from the side of the body to the tip. [24]
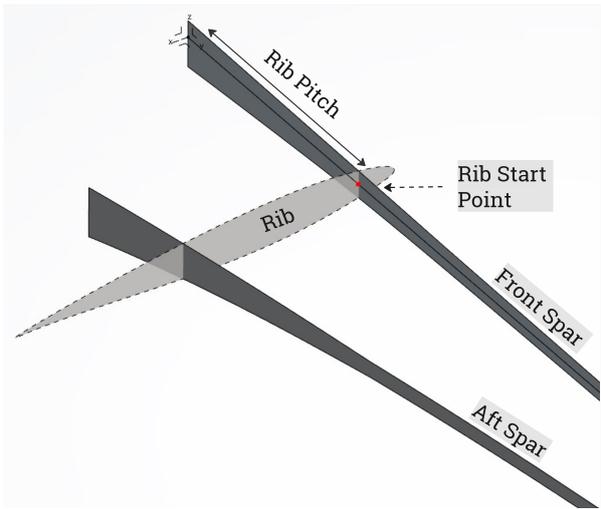
Figure 2.4: The UDF of the rib
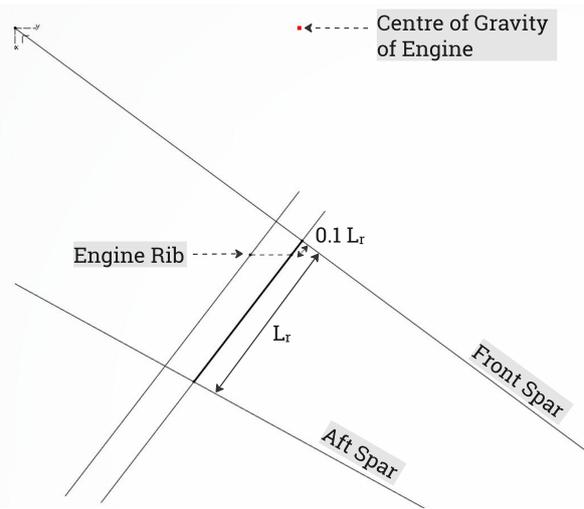
Figure 2.5: The UDF of the engine rib

An additional rib is created to support the engine and it is positioned between two existing ribs at the span-wise location of the engine. The engine is modelled with a point load at the estimated centre of gravity of the engine. The engine's placement aligns with the Boeing 777 specifications[1], situated at 32.6% of the wing semi-span[26]. The engine load is further discussed in subsection 2.3.3. Figure 2.5 shows the UDF of the engine rib. The reference for determining the two ribs is established from the start point of the rib. The engine rib, parallel to the span, is generated at 0.1% of the length of the rib between the spars of the outboard ribs.

Stiffener Generation

The structural model includes stiffeners at three locations within the wing box: at the skin(stringers), ribs, and spars. Stiffeners can be modelled through implicit and explicit methods. The uCRM used the implicit smeared stiffener approach[24]. In this approach, the influence of stiffeners is integrated into the skin's stiffness[27]. However, in the CAD environment, it is preferable to explicitly model the stiffeners.

The stiffeners used in the uCRM are T-stiffeners[24] as shown in Figure 2.6. The stiffener width, w_{st}, is set equal to the stiffener height, h_{st}. The rib and spar stiffeners are aligned with their respective surfaces and the stringers are parallel to the front spar. To emulate the attached stringer, the flange of the stringers is cut from the wing surface.
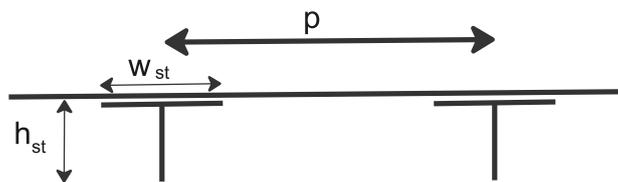
Figure 2.6: The parameterisation of the stiffeners and skin

To generate stiffeners, a unique UDF is created for each structural group, along with a Knowledge Pattern in which the stiffener pitch and height per structural group are defined. The results of uCRM by

---

[1]https://booksite.elsevier.com/9780340741528/appendices/data-a/table-4/table.htm

Brooks et al.[24] is utilised to determine these parameters. While the stiffener height in the uCRM can vary along the stiffener due to the smeared approach, a uniform stiffener height is assumed per stiffener for this thesis, with the maximum height observed in the uCRM being adopted. Table 2.2 shows the corresponding stiffener pitch and height values at each section.

Table 2.2: The values of the stiffeners and the shell section of the geometry

Parameter	Stiffener Pitch [mm]	Stiffener Height [mm]
Upper Stringers	180	75
Lower Stringers	220	45
Rib Stiffeners	230	68
Engine Rib Stiffeners	150	68
Spar stiffeners	150	68

Figure 2.7 shows the structural elements of the wing generated by the described UDFs and Knowledge Patterns. The detailed Knowledge Patterns code can be found in Appendix A.

(a) The top view of the wing skeleton including the rib and spar stiffeners

(b) The structural elements of wing near the side of body
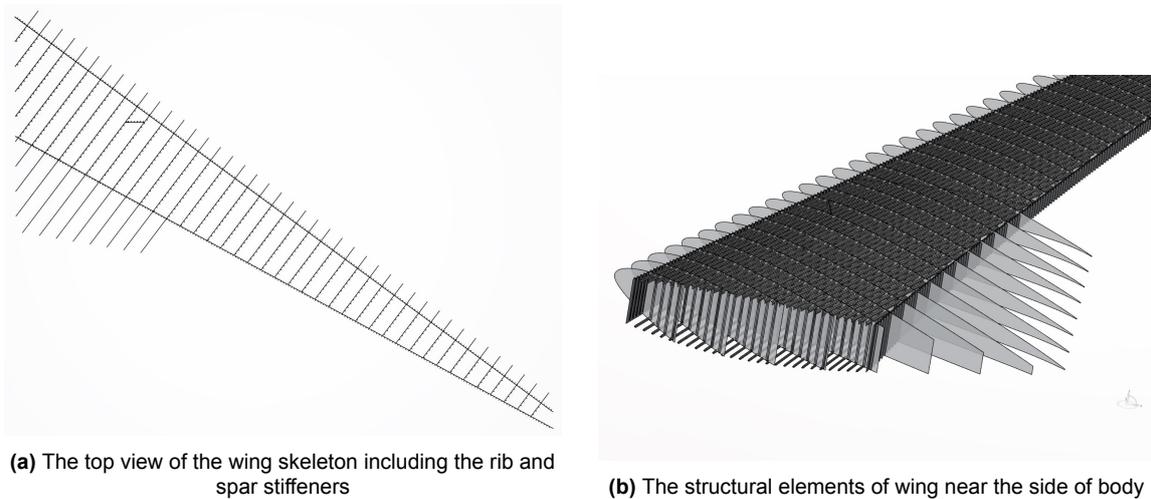
Figure 2.7: The structural elements of wing

2.1.4. Span-wise Trailing Edge Morphing Approach

Implementing span-wise trailing edge morphing on the geometric model requires the capability to modify the trailing edge of the implemented airfoil. The trailing edge is defined as the geometry after the aft spar. The aft spar of the CRM is placed at 60% of the chord at the Side of Body and tip[25]. The assumption is made that an inboard flap is present, limiting trailing-edge morphing exclusively to the outboard wing. In the selection of an appropriate methodology, two requirements must be met. The morphing approach under consideration should possess the capability to exclusively alter the trailing edge and should be compatible with the functionalities available within the CAD-based 3DExperience software.

An approach used by Werter et al.[28] utilises a database comprising NACA 4-series aerodynamic surfaces. While this methodology offers a systematic framework, it presents a challenge when modifications are focused solely on the trailing edge. Moreover, it imposes constraints on the design space by relying on a predetermined set of existing airfoils within the database.

A common approach is the use of airfoil parameterization. For example, Bézier curves can be used as a method for precise control over the trailing edge geometry. The mathematical curves, specified by strategically positioned control points, provide a framework for capturing the intricate contours characterising the trailing edge of the airfoil. In 2D cases, this methodology is effective. In 3D scenarios, Kenway et al. [29] observed that, due to the intricate geometry involved, Bézier curves require a substantial number of design variables. They propose using Free Form Deformation (FFD) for CAD - free

high-fidelity aero-structural optimisation. Burdette et al. [30] used this method to analyse adaptive morphing trailing edge on the uCRM. However, FFD is a CAD-free approach and therefore not suitable for this research.

Lépine et al. [31] discovered that employing airfoil base parameters reduces the convergence speed for aerodynamic optimisation. Two common airfoil base parameterization methods are the Class Shape Transformation (CST)[32] and the PARSEC method. The CST method utilises Class-Shape functions to mathematically characterise the airfoil's shape. Despite its extensive use in airfoil parameterization, the exclusion of the trailing edge from a designated point poses a challenge. Furthermore, the requirement to trace the trailing edge as a preliminary step introduces a potential source of error, thereby augmenting the margin of deviation from the original uCRM airfoils.

The PARSEC method uses 11 parameters to define an airfoil as shown in Figure 2.8[33] :

- leading edge radius (r_{LE})
- upper crest location (X_{UP},Z_{UP})
- lower crest location (X_{LO},Z_{LO})
- upper and lower curvature($Z_{XX_{UP}}$,$Z_{XX_{LO}}$)

- trailing edge coordinate(Z_{TE}) and direction(α_{TE})
- trailing edge wedge angle(β_{TE})
- thickness(ΔZ_{TE})
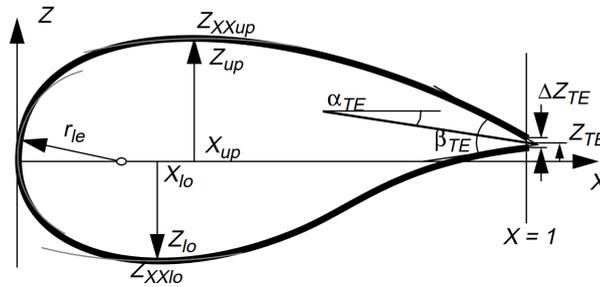
Figure 2.8: PARSEC airfoil geometry[33]

With these 11 parameters, Sobieczky similar to theCST uses a polynomial to create the airfoil:

$$Z_k = \sum_{n=1}^{6} a_{n,k} X_k^{\frac{n-1}{2}} \tag{2.2}$$

where a_n can be found using the geometric parameters, k takes a value of 1 and 2 for the upper and lower surface. The utilisation of this airfoil parameterization is widespread due to its incorporation of crucial airfoil features.

In the context of this research conducted within a CAD-based environment, the unique advantage arises from the ability to exclusively use the four trailing edge parameters (Z_{TE}, α_{TE}, β_{TE}, and ΔZ_{TE}) to modify the shape of the trailing edge. The original airfoils of the uCRM at the 5 positions(P4-8) along the span are split at the connection with the aft spar. This results in an upper and lower trailing edge spline. Utilising the four trailing edge parameters, two new splines are generated, maintaining tangency to the original airfoil at the aft spar. The assumption is made that the material will not exhibit elastic behaviour, implying that the lengths of the trailing edge legs will remain constant. The trailing edge parameterization by using the four trailing edge parameters of the PARSEC method is shown in Figure 2.9. The adjustment of trailing edge splines by altering parameter values is performed manually to derive a solution.
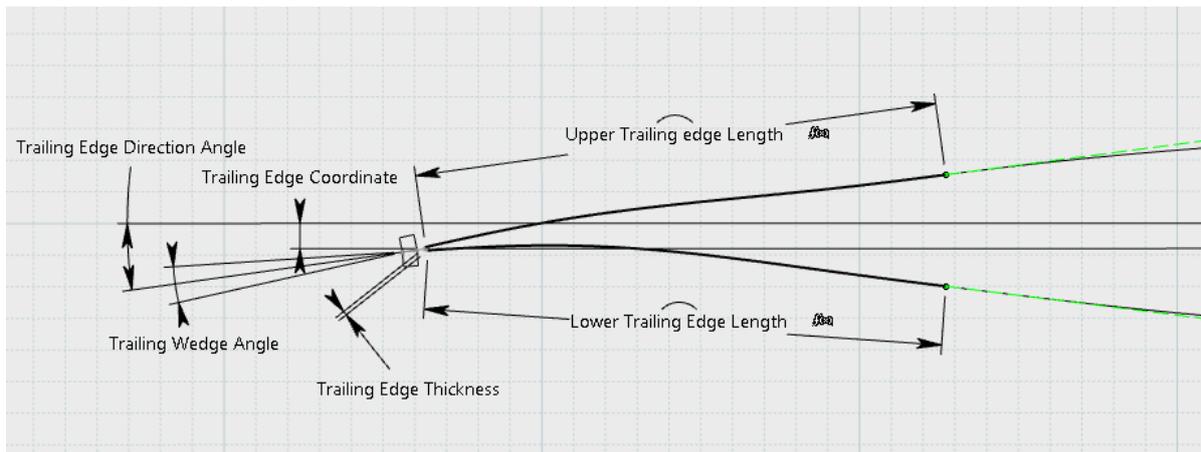
Figure 2.9: Trailing edge parameterization

Transonic Design Principles

By adjusting the trailing edge of the original wing, it is important to understand the interaction of the trailing edge with the flow. This supports the determination of the parameters used in the test sets. Therefore this section delves into the design principles of transonic wings. An important aspect of the wing is the aspect ratio. The aspect ratio can be calculated by $\frac{b^2}{A}$. The uCRM wing has an aspect ratio of 9, which is considered high. With a high aspect ratio wing, the two-dimensional effects predominantly influence the flow phenomena[34]. Therefore, a closer examination of the airfoil theory is conducted.

The CRM consist of supercritical airfoils[23]. Supercritical airfoils, developed in the 1960s and 1970s, were designed to demonstrate favourable transonic behaviour while retaining acceptable characteristics at low speeds[35]. The occurrence of shock waves often characterises the transonic flow regime. The formation of strong shock waves can cause the separation at the shock foot increasing the drag exponentially[34]. A supercritical airfoil uses geometrical features to weaken the shock wave. It is characterised by a large leading-edge radius, a flat middle part of the airfoil and a cusped trailing edge with a finite thickness[35]. The increased leading edge radius causes a strong expansion wave from the leading edge region. The expansion waves increase the velocity. These expansion waves are reflected as compression waves. When the compression contacts the airfoil, it decelerates the flow[35]. Given the diminished curvature in the midsection of the airfoil, the deceleration of compression and acceleration of expansion waves may counterbalance, leading to the absence of flow acceleration ahead of the shock. This reduces the shock strength. Figure 2.10 shows this effect.
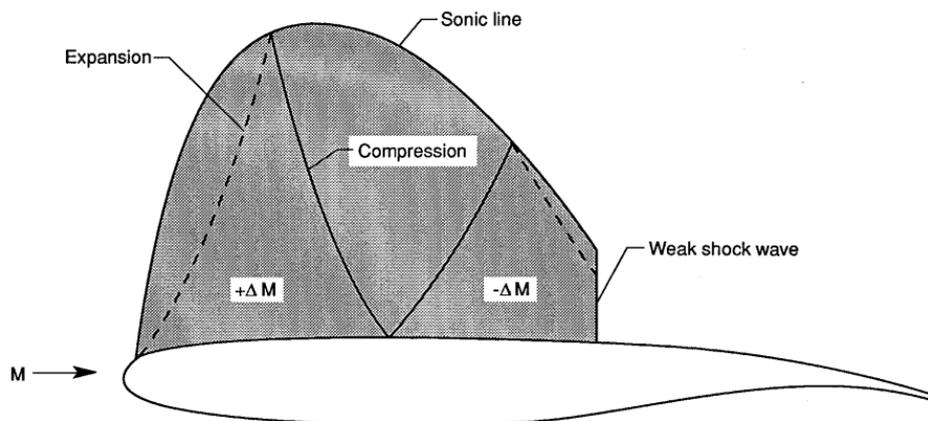
Figure 2.10: The effect of the increased leading edge radius and flat upper surface on the shock strength[35]

A supercritical airfoil exhibits a notable adverse pressure gradient at its trailing edge region, which makes the boundary layer more susceptible to separation[34]. To effectively manage the adverse pressure gradient, a supercritical airfoil incorporates both a cusp trailing edge and finite thickness at the trailing edge.

In incompressible, inviscid flow the Kutta condition should be satisfied[34]. The Kutta condition dictated that the flow around an airfoil must leave the trailing edge smoothly. This implies that the pressure on the upper surface of the airfoil is balanced by the pressure on the lower surface at the trailing edge, preventing the formation of a stagnation point and ensuring a physically realistic and consistent flow solution. This is shown in Figure 2.11[34] for two different trailing edge shapes.

(a) Trailing edge with a finite angle
at point a: $V_1 = V_2 = 0$

(b) A cusped trailing edge
at point a: $V_1 = V_2 \neq 0$
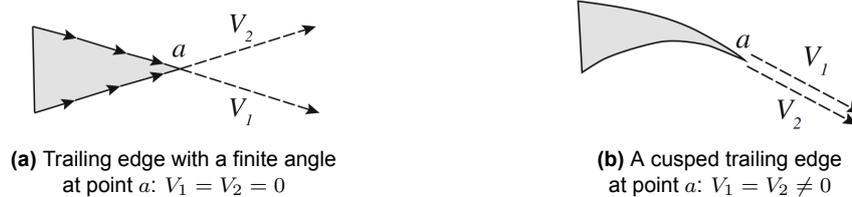
Figure 2.11: The Kutta condition applied on the trailing edge shape[34]

Figure 2.11a shows a trailing edge with a finite angle, while the upper and lower surfaces of Figure 2.11b are parallel at point a, a cusped trailing edge. The Kutta condition can only be satisfied for the trailing edge with a finite angle if the velocities are zero, leading to a stagnation point at point a. At the cusped trailing edge, no stagnation point is needed to satisfy the Kutta condition. This results in a pressure coefficient smaller than 1. In practice, this thesis does not assume incompressible, inviscid conditions. Nevertheless, it is observed that even in these conditions, the cusped trailing edge yields a lower pressure coefficient at the trailing edge and diminishes adverse pressure gradients[34].

Another advantage of the cusped trailing edge is the increase in pressure attributed to the concave shape of the lower surface. This phenomenon, known as aft loading, results in an increased total lift coefficient.

Incorporating a blunt trailing edge is a measure aimed at minimising weight while still achieving the necessary stiffness for the cusped trailing edge[34]. An additional benefit of a trailing edge with finite thickness is the reduction of adverse pressure gradients on the upper surface, achieved through off-surface pressure recovery in the wake.[36]. The original uCRM airfoils also show this blunt trailing edge.

The supercritical airfoil is designed to mitigate shock-induced separation, although complete avoidance may not always be achievable. Under a shock, a separation bubble can occur as shown in Figure 2.12. When the separation bubble extends to the end of the trailing edge, separation occurs[34]. The deliberate aft movement of the shock wave on supercritical airfoils heightens the probability of the separation bubble reaching the trailing edge. Therefore, the aft wards placement of the shock wave should be limited.
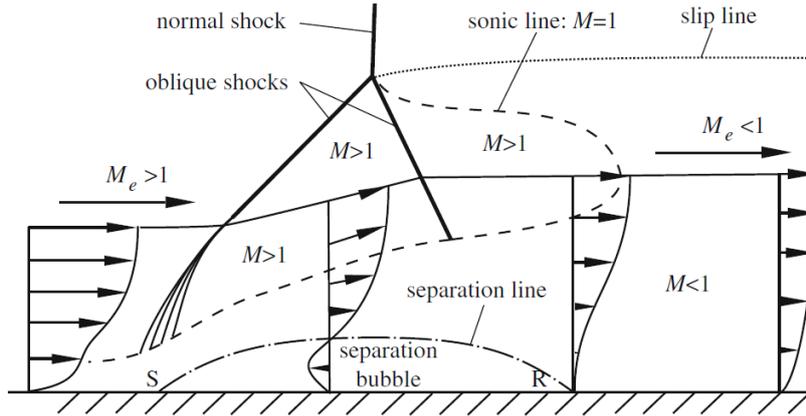
Figure 2.12: A shock-induced separation bubble[34]

Sizing Variables
This thesis uses two design cases to analyse the trailing edge morphing wing. The selected cases were intended to achieve a high-lift wing with a downward curvature of the original trailing edge, as well as an aerodynamically efficient wing with an upward curvature of the original trailing edge. The latter design aims to achieve a higher lift-to-drag ratio, resulting in increased aerodynamic efficiency. These designs are based on the transonic design principles and are not the optimized solution.

The first constraint before determining the parameter values is that there is no material elasticity. This results in that the length of the trailing edge legs and the trailing edge thickness, ΔZ_{TE}, is kept equal to the original airfoils. The second constraint is that the trailing edge legs do not cross each other. The third constraint is that the span-wise shape of the trailing edge does not contain steep gradients. Lastly, supercritical airfoil theory dictates that the wedge angle, β_{TE}, should be small to avoid adverse pressure gradients, especially for the high-lift wing. Therefore, β_{TE} should be smaller than 20 degrees. These constraints are used for determining the two design cases with the sizing variables: the trailing edge coordinate, Z_{TE}, direction, α_{TE}, and wedge angle, β_{TE} at span-wise positions 4-8. Table 2.3 shows the parameters of the original wing which are used as a baseline for the two design case parameters.

Table 2.3: The original Airfoil Characteristics at the positions P4-8.

Airfoil Position	Original Airfoil Characterestics			
	ΔZ_{TE} [mm]	Z_{TE} [mm]	β_{TE} [°]	α_{TE} [°]
P4	13,1	-142,0	5,3	22,6
P5	13,5	-151,0	5,2	24,9
P6	13,2	-173,2	5,2	26,3
P7	12,3	-159,5	5,3	22,0
P8	13,5	-69,6	5,9	3,0

The curvature of the airfoil is changed by a combination of Z_{TE} and α_{TE}. The initial airfoils of the uCRM have α_{TE}s ranging from 20 to 26 degrees. The high-lift design case with the downward curvature aims to increase α_{TE} to 28 degrees. In the efficient design, where the curvature of the trailing is upward, reduces α_{TE} to aim at 10 degrees. Some values within the test sets deviate to preserve the previously stated requirements. The test sets are outlined in Table 2.4 and shown in Figure 2.13.

Table 2.4: The test sets parameters of the morphing wings

Airfoil Position	Test Set 1:High Lift Wing				Test Set 2: Efficient Wing			
	ΔZ_{TE} [mm]	Z_{TE} [mm]	β_{TE} [°]	α_{TE} [°]	ΔZ_{TE} [mm]	Z_{TE} [mm]	β_{TE} [°]	α_{TE} [°]
P4	13.1	-220.0	2.0	28.0	13.1	-100.0	9.0	9.0
P5	13.5	-250.0	1.0	29.0	13.5	-50,0	15.0	7.0
P6	13.2	-250.0	6,0	28,0	13.2	-70,0	18.0	6.0
P7	12.3	-200.0	5.0	26.0	12.3	-80.0	20.0	8.0
P8	13.5	-120.0	6.0	12.0	13.5	-100.0	10.0	0.0
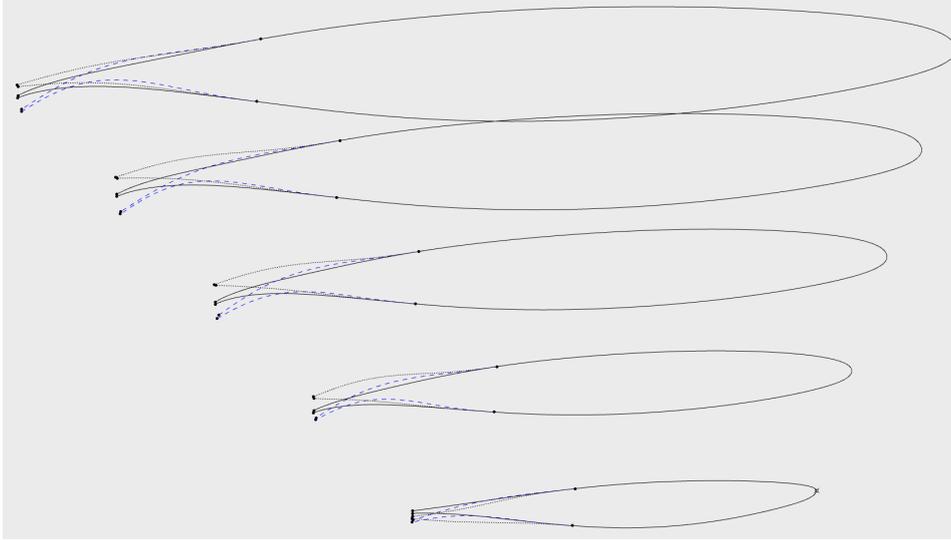
Figure 2.13: The resulting morphed airfoils at position 4-8. The black solid line shows the original airfoil. The blue striped line represents the high-lift design, and the gray dotted line represents the efficient design.

2.2. Aerodynamic Model

3DExperience Apps: Fluid Model Creation, Fluid Scenario Creation, Material Definition

The geometrical model forms the foundation for the aerodynamic model. The aerodynamic model is developed using the *Fluid Model Creation* and the *Fluid Scenario Creation* app. When opening the *Fluid Scenario Creation*, the assistant window pops up, which outlines the necessary steps for defining all components of the Fluid simulation.

This section starts with a numerical background for performing the CFD simulation. This encompasses details regarding the 3DExperience fluid solver. Subsequently, the CFD domain is discussed, where the flight conditions and mesh generation are explained. It is essential to highlight that the existing academic license is constrained by a limitation on the maximum number of nodes, which is capped at 5 million. This constraint restricts both the fluid domain and the mesh, a topic that will be discussed in detail in subsection 2.2.2. Lastly, the setup of the fluid solver is addressed.

2.2.1. Numerical Background

The fluid solver in 3DExperience is a RANS pressure-based finite volume solver. The Navier-Stokes equations used in CFD consist of the conservation of mass, momentum, and energy. These equations describe the motion of fluid substances, considering the effects of viscosity, pressure, and external forces[37]. To derive the Navier-Stokes equations for a Newtonian fluid of conservation of momentum, the continuity equation(Equation 2.3) and equation of motion(Equation 2.4)[38] are used:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{2.3}$$

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f} \tag{2.4}$$

Where ρ is the fluid density, $\mathbf{v}$ is the velocity vector, t is time, ∇ denotes the divergence operator, σ is the stress tensor, and f represents external forces acting on the fluid. By combining these equations, the compressible Navier-Stokes equation for the conservation of momentum can be derived[38]:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \frac{\mu}{3} \nabla \left(\nabla \cdot \mathbf{v} \right) + \mathbf{f} \tag{2.5}$$

Where p is the pressure, μ is the dynamic viscosity, and f represents external forces acting on the fluid.

With compressible flows, where density can vary with pressure, an additional equation is required to calculate the density accurately. This equation is derived from the principle of energy conservation by using Fourier's law and the equation of state[38]:

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho \mathbf{v} E) = \boldsymbol{\sigma} : \mathbf{S} - \nabla \cdot \mathbf{q} - \mathbf{f} \cdot \mathbf{v} + q"' \tag{2.6}$$

where E is the total energy and $\mathbf{S}$ is the strain rate tensor. $\mathbf{q}$ is the local heat flux which can be found using Fourier's law.

Solving the complete Navier-Stokes equations for turbulent flows can be computationally expensive. Applying time-averaging to the Navier-Stokes equations yields the RANS equations, which represent the mean behaviour of fluid flows. The RANS equations introduce additional terms in the momentum equation. This is also called the Reynolds stress tensor(τ_{ij}) and can be written as $\overline{-\rho v_i' v_j}{}^\backprime$. Additional equations are needed to model the Reynolds stresses to close the RANS equations. 3DExperience uses an eddy viscosity model. Eddy viscosity models aim to simplify the representation of turbulent viscosity, a measure of the internal friction in a turbulent flow, by introducing an effective viscosity term called the eddy viscosity (μ_t). Linear eddy viscosity models use the Boussinesq assumption for the constitutive relation[39, 40]:

$$\tau_{ij} = \overline{-\rho v_i' v_j'} = 2\mu_t \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \tag{2.7}$$

Where k represents the kinetic turbulent energy and δ_{ij} is the Kronecker delta. There are multiple ways to model the eddy viscosity.

Turbulence Models
3DExperience has implemented three eddy viscosity or turbulence models: Realizable $k - \epsilon$, Shear-Stress Transport (SST) $k-\omega$ and Spalart-Allmaras. Both SST $k-\omega$ and Spalart-Allmaras can be used to model external flow with sufficient accuracy in near wall flows[41, 42, 43]. The 3DExperience platform also provides a Fluid Verification Guide in the User Assistance[44]. This guide encompasses various test cases and verification details tailored to the Fluid Model Creation and Fluid Scenario Creation applications. Currently, there are no available airfoil test cases; the supersonic flat plate test scenario is the one most closely resembling the current setup. In this test case, the accuracy of the velocity and the temperature functions are assessed. This shows that the most accurate model is the Spalart-Allmaras (SA) model. The SA model is also used by the uCRM[24].

The SA turbulence model[45] is a one-equation model developed to provide accurate results with relatively low computational cost. This method solves for a single variable, turbulent viscosity ($\tilde{\nu}$), and is especially favoured in aerospace applications where computational efficiency is paramount. The model includes a transport equation for $\tilde{\nu}$ and incorporates various terms to model turbulent production, destruction, and turbulent diffusion. Despite its simplicity compared to multi-equation models, the SA model has been found to perform well in a wide range of turbulent flows, including those involving separation and complex geometry.

Flow Solver
In the 3DExperience platform, there are two methods for solving the RANS equations once they are closed: a segregated approach and a coupled approach. While the segregated solver can handle Mach

numbers up to about 2, it struggles with complexities at higher Mach numbers. On the other hand, the coupled solver boasts improved convergence rates, making it the preferred option, particularly for high Mach numbers. The only catch is that it requires twice as many memory resources. Given the current computer setup, the segregated solver was utilised for this thesis project.

The segregated approach solves the governing equations sequentially in a series of steps. To achieve this, 3DExperience employs the Semi-Implicit Method for Pressure Linked Equations) (SIMPLE) algorithm, as introduced by Patankar[46]. The steps of the SIMPLE algorithm are illustrated in Figure 2.14 [38].
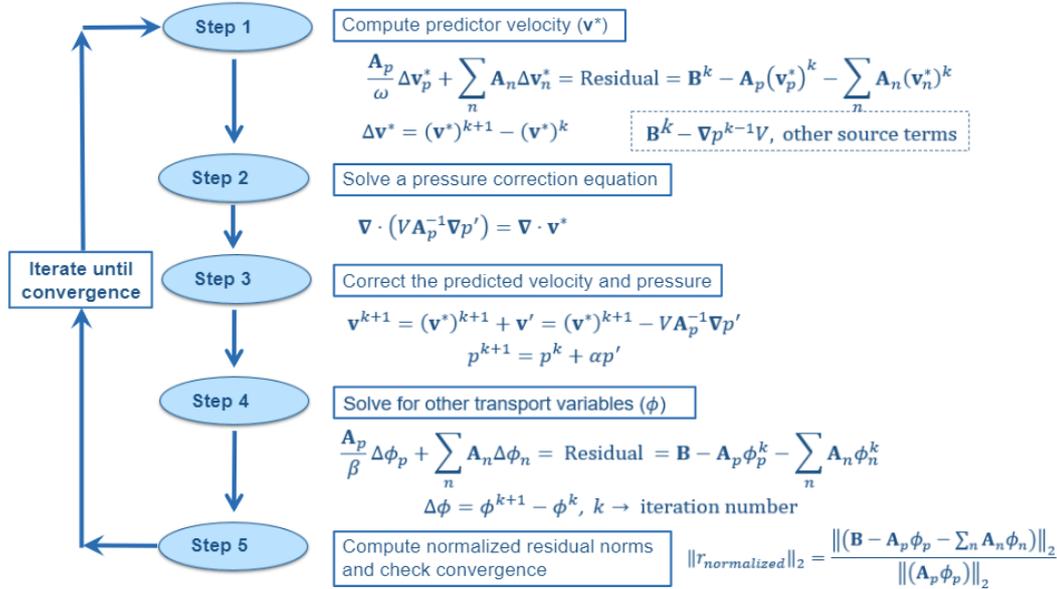
Figure 2.14: The steady-state SIMPLE alogrithm in 3DExperience where k is the iteration number and ω, α and β are under-relaxation factors[38]

This pressure-based segregated Solver operates by integrating the energy equation with an equation of state to compute density fluctuations. It uses under-relaxation factors to prevent overshooting and oscillations when the solver tries to converge.

There are a variety of methods available to improve the convergence rate when solving complex systems of equations. 3DExperience has implemented several of these methods for solving the Navier-Stokes equations. Linear solvers are commonly utilised to increase the convergence speed and stability of simulations, but there is a research gap on the impact of these solvers. The Algebraic Multigrid (AMG) method is the most frequently used pre-conditioner, coupled with the Biconjugate gradient stabilized (BiCGStab) linear solver, which 3DExperience states is superior to the Flexible Generalised Minimum Residual (FGMRES) solver [44].

By taking advantage of a hierarchical grid structure, AMG expertly navigates complex linear systems. This method expedites the iterative solution process by solving problems on coarse grids and then refining solutions on progressively finer grids. Its parallel processing capabilities provide a significant advantage. AMG is especially noteworthy as a pre-conditioner, guiding other iterative solvers towards efficient and accurate solutions.

The BiCGStab method is an iterative method designed for general non-symmetric systems including indefinite systems[47]. The BiCGStab solver adeptly handles the equation's advection and diffusion terms, accounting for particle positions and converting ϕ into matrix format for every cell within the mesh.

Wall Modelling

A boundary layer consists of two main layers. The inner layer is a small region near the wall, where most velocity variations take place. And the outer layer, which comprises most of the boundary layer. In this region, the velocity converges to the free stream velocity.

The inner layer can be subdivided into three regions: the viscous sub-layer, the buffer layer, and the log layer[48]. These layers can be distinguished by their normalised distance to the wall, $y+$. In the viscous layer($y+ < 5$), the flow is dominated by molecular diffusion. The buffer layer($5 < y+ < 30$) is the transition region where the flow transitions from laminar to turbulent flow. Lastly, the logarithmic layer($y+ > 30$) consists of a fully turbulent flow.

There are two approaches to achieving an accurate result near the wall. The first approach is the Low Reynolds Number approach. This approach requires the viscous sub-layer to be resolved, meaning a fine mesh near the wall($y+ \approx 1$). This is computationally expensive. Therefore wall function approach was developed. Based on experimental data a relation could be found between the normalised wall distance and the normalised velocity, $U+$.

In the viscous layer, the laminar law($U+ = y+$) is used to approximate the normalised velocity, $U+$. Due to the transition in the buffer law, it is difficult to approximate the normalised velocity so no relation could be found. In the logarithmic layer($y+ > 30$) the log-law($U+ = \frac{\ln y+}{k} + C+$) is used, where κ is the von Kármán constant (≈ 0.41), and $C+$ is a constant (≈ 5).

3DExperience blends both methods to be able to use coarse meshes as well as fine meshes and still get an accurate result. This is called the hybrid wall function approach[38].

2.2.2. CFD Domain

The foundation of CFD analysis lies in the CFD domain, which can be determined using various methods. Some common shapes include bullet-shaped and semi-circle[49]. The 3DExperience software has the automatic function to create a bounding box. As circular domain geometries can be created in 3DExperience, however, it is interpreted as complex shape, leading to an increases the node count in those areas. Due to the constraints of a student license, it was decided to utilize the bounding box approach.

As a general rule, a larger domain typically yields a more accurate result. Spalart et al.[50] recommended a flow domain of at least 50 body lengths. However, due to the 5 million node limit on the 3DExperience academic license, both the domain and mesh are significantly limited. To define the bounding box, the parameter "L" is used, which represents the length of the geometry at the symmetry axis. Figure 2.15 shows the CFD domain including the boundary conditions. In terms of the bounding box, it is restricted to $5L$ upstream and $10L$ downstream. In the span-wise direction, it is limited to 2.5 of the span. All planes, excluding the symmetry axis, utilise a free stream boundary condition to calculate Riemann invariants automatically, which ultimately determines the flow parameters at the boundary. The free stream Mach number and static conditions serve as the input parameters for the free stream boundary condition.
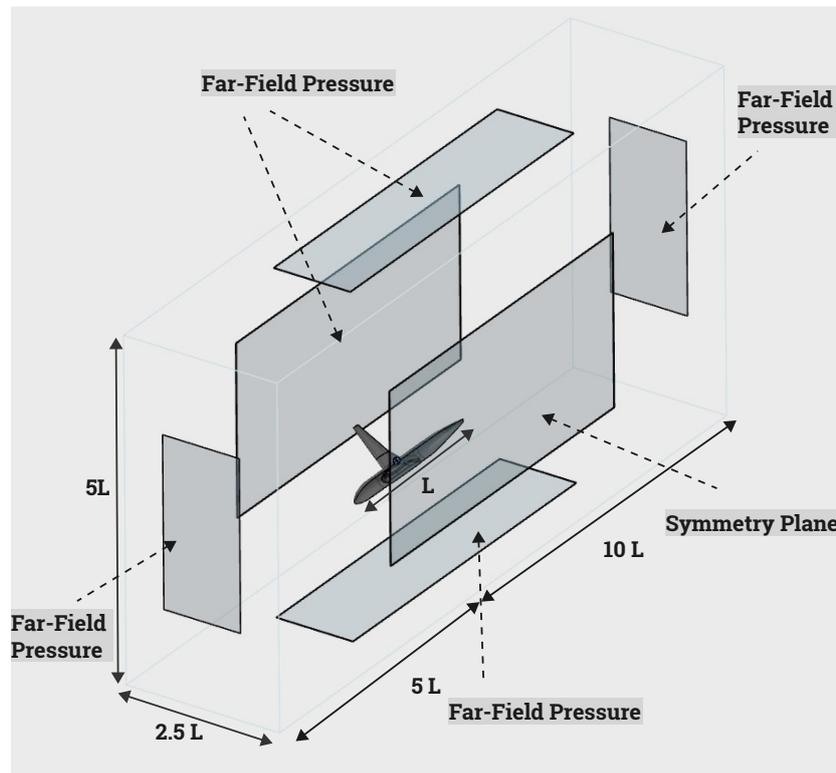
Figure 2.15: CFD Bounding Box

Flight Conditions
The steady-state aerodynamic analysis will be performed with a static model, which is inclined with an angle of attack with respect to the inflow. The flight conditions are at the cruise conditions of the uCRM and are shown in Table 2.5. The temperature and pressure are found for the simulation using the standard atmosphere calculations. In the *Material Definition* app, a material is created with the air properties corresponding to the flight conditions. The density is found using the calculated pressure and temperature. The viscosity is determined through Sutherland's law[51], and the thermal conductivity was found to be 0.02 W/m K[2]. Additionally, specific heat and Equation of State (EOS) characteristics are configured to maintain a constant specific heat ratio, denoted as γ, set at 1.4. The EOS is set to emulate a compressible ideal gas, specifying a specific gas constant, R, at 287 J/kg K. The specific heat at constant pressure is then calculated using the formula $\frac{\gamma R}{\gamma - 1}$.

Table 2.5: CFD Flight conditions

Definition	Value
Altitude	37000ft
Mach	0.85
Angle of Attack	2 degrees
Temperature	216.65 K
Pressure	22625.79 Pa
Density	0.3638 kg/m3
Dynamic Viscosity	$1.422 \cdot 10^{-5}$ Pa s
Thermal Conductivity	0.02 W/m K
Specific Heat a Constant Pressure	1.0045 kJ/kgK
Specific Gas Constant	287 J/kg K

---

[2]https://www.engineeringtoolbox.com/international-standard-atmosphere-d_985.html[visited on 29-11-2023]

The uCRM cruise conditions dictate a constrained lift coefficient at 0.5. Due to the time constraints of the thesis, it proves unfeasible to calibrate the simulation precisely to adhere to this specific lift coefficient. Consequently, an empirically determined fixed angle of attack is applied. Brooks et al. [24] found the angle of attack at cruise of 2.044 degrees for a coarse mesh. Therefore, an angle of attack of 2 degrees thesis uses this angle of attack for all simulations.

These initial conditions serve as an initial guess for the iterative solver. Hence, an initial condition is also needed for the turbulence models. The initial turbulence is defined by the turbulence intensity,I, and the turbulence viscosity ratio,$\frac{\mu_t}{\mu}$. Spalart et al.[50] developed a recommendation for the inflow conditions for most typical external aerodynamic applications. The turbulence intensity in external flows is low, generally between 0.05% and 1%. This thesis sets the turbulence ratio at 1%. The turbulence viscosity ratio can be estimated using $2E-7 \cdot Re$[50]. A Re of $4.5e7$ results in a turbulence viscosity ratio of 9. The initial values of variables of the two-equation eddy viscosity models can be calculated using Equation 2.8[38].

$$
\begin{aligned}
k &= \frac{3}{2}(vI)^2 \\
\epsilon &= C_\mu \frac{k^2}{\frac{\mu_t}{\mu}\mu} \\
\omega &= C_\mu \frac{k}{\frac{\mu_t}{\mu}\mu}
\end{aligned}
\tag{2.8}
$$

Mesh generation

A systematic approach is needed to discretise the computational domain to ensure numerical accuracy within the simulation framework. The mesh must have the capability to capture viscous effects accurately. The three main types of meshing are structured, unstructured and hybrid mesh.

In a structured mesh, the grid points are organised in a regular and systematic manner[52]. The internal cells are topologically similar and have the same number and type of connections. Structured meshes are commonly used in applications where the geometry permits a regular grid arrangement. Multi-block grids are used to improve the grid quality or to adapt to complex geometries[53]. The domain is split into multiple subdomains called blocks with different mesh topologies. The CRM uses a multi-block structured mesh with 1018 blocks[54].

In an unstructured mesh, no regularity exists between grid points, and there is no universal rule for connectivity with neighbours[52]. Structured meshes are often more efficient than unstructured grids because the flow generally aligns with the geometry[53]. Unstructured grids have increasingly become the predominant approach because generating block-structured grids automatically on arbitrary geometries is inherently challenging and requires a substantial amount of time[53]. The adaptability of unstructured grids allows them to conform to intricate geometries without requiring a predefined multi-block configuration, enabling localised mesh refinement.

The primary challenge associated with unstructured grids is meeting the boundary layer requirements of high Reynolds number flows[53]. In such scenarios, it becomes essential to adapt the grid density in the normal direction to align with the velocity profiles within the boundary layer. Figure 2.16[52] shows how the grid refinement near the wall influences the computed velocity profile where δ represents the boundary layer thickness. Hybrid grids have been developed to address this challenge. These grids represent a combination of structured and unstructured elements, incorporating layers of quadrilaterals or prisms specifically generated in the near-wall region[53]. Figure 2.17[55] shows a hybrid mesh around an airfoil.
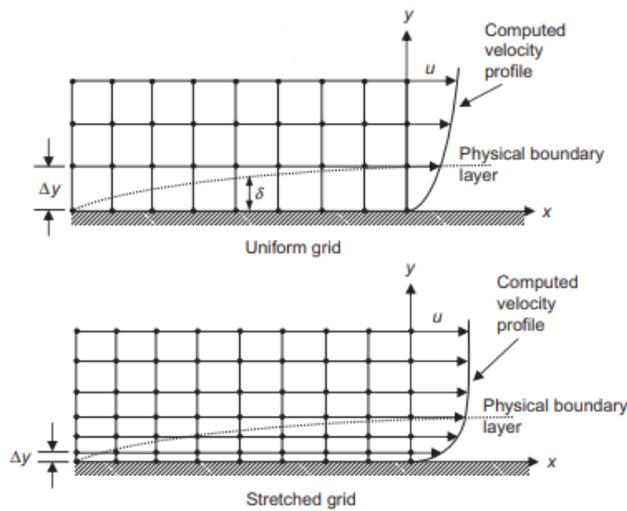
Figure 2.16: The difference in computed velocity profile due to the grid refinement[52]
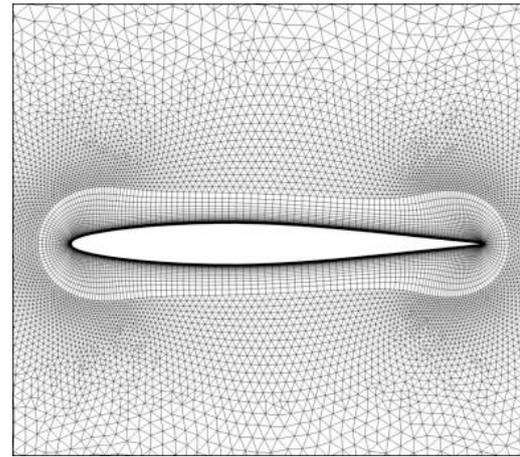
Figure 2.17: A hybrid mesh around an airfoil[55]

3DExpierence uses Hex-Dominant Mesher (HDM) technology to simplify the meshing process for CFD simulations. It extracts fluid domains and generates by setting the global and boundary layer mesh size an automatic unstructured finite volume mesh, complete with prism boundary layers directly on the native CAD geometry[44]. The automatic meshing procedure maintains the geometric complexity by implementing local mesh refinement. As shown in Figure 2.18, the aft section of the fuselage exhibits local mesh refinement at the wing-fuselage fairing, around the tail airfoil, and the fuselage tail cone. While this local mesh refinement enhances mesh accuracy, adhering to the 5 million node limit imposed by the academic license is crucial to avoid unnecessary refinement. The *Optimize surface treatment* option attempts an additional iteration over the initial mesh to reorganise and coarsen highly refined areas. Nevertheless, the inability to selectively target specific areas results in unavoidable unnecessary mesh refinement.
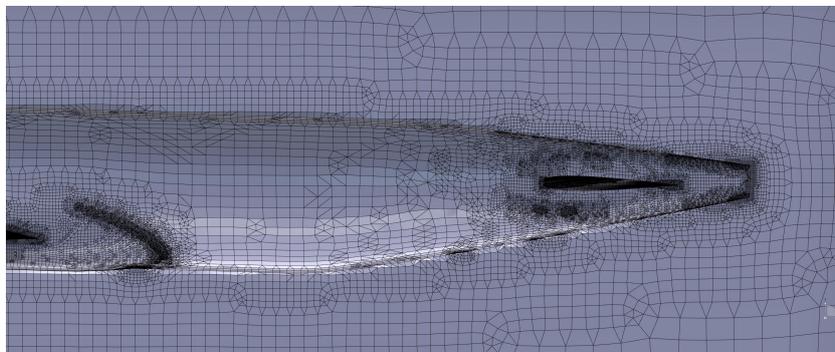
Figure 2.18: The mesh generated around the fuselage with automatic mesh refinement at complex geometry

The mesh can be refined by using a local surface refinement on the geometry. Goetten et al.[56] advise between 80-100 cells in a chord-wise direction. Therefore the 7 sections of the wing as shown in Figure 2.3 are used where the smallest chord is divided by 80. The fuselage mesh size is increased to decrease the amount of nodes to 400mm.

Another challenge which lays in the mesh generation is the near-wall modelling. At transonic conditions (Mach between 0.8 and 1.2), the flow could separate, and therefore, an accurate cell size near the wall is needed to capture the viscous effects. The first step to calculate the wall distance of the first cell, y_{w_1}, is to find the skin friction coefficient, C_f. For $Re < 10^9$, the Schlichting skin-friction correlation can be used[48]:

$$C_f = [2\log_{10}(Re_x) - 0.65]^{-2.3} \tag{2.9}$$

With the skin friction coefficient the wall shear stress, τ_w can be computed resulting in the friction velocity, v_τ:

$$\tau_w = C_f \cdot \frac{1}{2}\rho v_{freestream}^2 \tag{2.10}$$

$$v_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{2.11}$$

With the friction velocity, the $y+$ can be found for a given wall distance. With this correlation, the first cell distance can be estimated:

$$y+ = \frac{y_w v_\tau}{\nu} \tag{2.12}$$

With compressible flow, using a $y+$ 1 near the wall is advised to capture the viscous effects. However, with the 5 million node limit, this is not possible. The other approach is to have the first cell in the logarithmic layer ($y+$ between 30 and 300). The first layer thickness is chosen to be 2mm to target the logarithmic layer with 15 boundary layers.
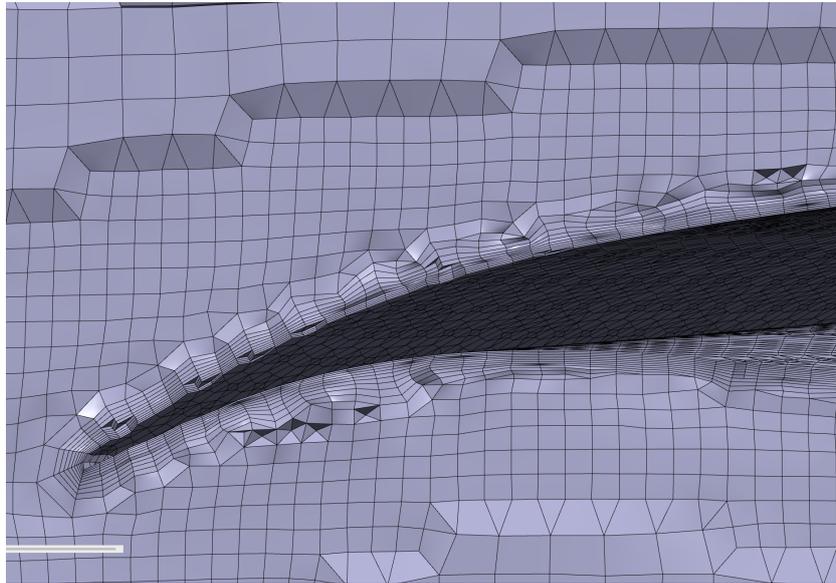
Figure 2.19: The boundary layer mesh around the trailing edge of the airfoil

2.2.3. Solver Setup

The CFD solver is set to have a maximum of 2000 iterations. The stopping criteria are based on the momentum, energy and turbulence residuals with an accuracy of 10^{-4}. Due to the memory limit of the current set-up, the SIMPLE segregated solver is used. The linear solvers and under-relaxation factor used for each equation are shown in Table 2.6.

Table 2.6: Settings of the CFD Solver

Equation Type	Linear Solver	Under-Relaxation Factor
Pressure	AMG-BiCGStab	0.2
Momentum	AMG-BiCGStab	0.8
Turbulence	AMG-BiCGStab	0.9
Energy	AMG-BiCGStab	0.9

2.3. Structural Model

3DExperience Apps: Structural Model Creation, Structural Scenario Creation, Material Definition

The *Structural Model Creation* and *Structural Scenario Creation* apps are used to create the structural model, and a linear analysis is performed. The structural solver of 3DExperience is similar to the well-known FEAsoftware Abaqus. The academic license includes a node limit of 1 million nodes for the structural mesh. Therefore, the structural model only consists of the wing, excluding the fuselage. The clamped boundary condition is used at the rib of the wing-fuselage connection. Brooks et al.[24] used the boundary conditions where the wing is clamped at the symmetry axis and allowed displacements in the span-wise direction and rotation. However, it was found that these displacements were small. Therefore, the clamped boundary condition at the fuselage wing connection is used.

2.3.1. Material and Section Properties

The first step is assigning the material properties to the structural geometry. The uCRM uses the 7000 series of aluminium alloy on all the structural elements and the properties are shown in Table 2.7. This material is created in *Material Definition* app.

Table 2.7: Properties of the material[24]

Parameter	value
Density	2780 kg/m^3
Young's modulus	$73.1 \cdot 10^9$ Pa
Poisson's ratio	0.3
Yield strength	$420 \cdot 10^6$ Pa

As the geometry is created using surfaces without thicknesses, the section properties are assigned in *Structural Model Creation* with shell sections. A shell section is a structure in which one dimension, in this case the thickness, is significantly smaller than the other two dimensions[44]. Taylor et al.[26] developed a characterization of the CRM wing for low-fidelity aerostructural optimization. In this research, the geometric and structural properties are based on the CAD geometries and finite-element models for the CRM wing and the uCRM. This is used in combination with the results of the uCRM[24] to define the thickness of the skin and the stiffeners. The skin and stiffener thickness is assumed to be constant per structural group except for the wing skin. The wing skin is divided into 7 span-wise sections as shown in Figure 2.20.
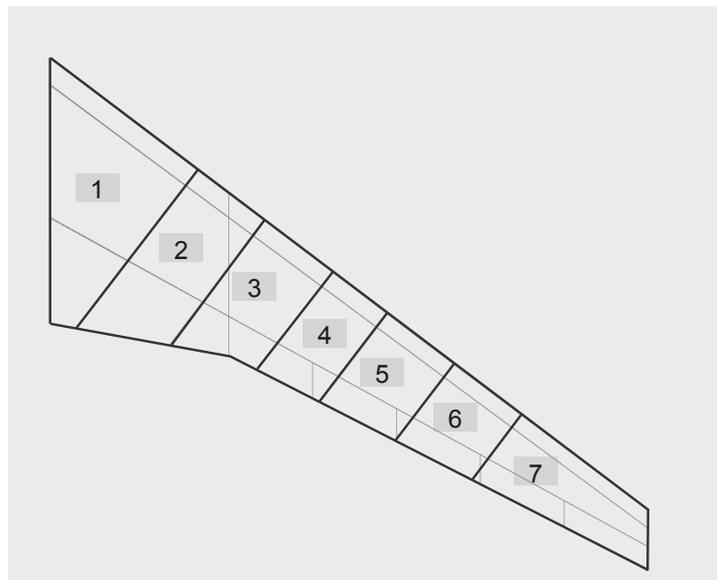
Figure 2.20: The planform division for the skin thickness.

To maintain model simplicity, the wing sections align with the sections created for the fuel tank. A comprehensive explanation of the sectioning is provided in subsection 2.3.3. Table 2.8 and Table 2.9

show the values of the thickness of the skin and stiffener shell sections of the geometry.

Table 2.8: The values of the skin thickness of the shell section per section

Parameter	Skin Thickness[mm]
Surface section 1	18mm
Surface section 2	20mm
Surface section 3	19mm
Surface section 4	15mm
Surface section 5	12mm
Surface section 6	10mm
Surface section 7	7mm
Ribs	5mm
Engine rib	10mm
Spars	5mm

Table 2.9: The values of the stiffener thickness of the shell section per section

Parameter	Stiffener Thickness[mm]
Upper stringers	13mm
Lower stringers	15mm
Rib stiffeners	5mm
Engine rib stiffeners	10mm
Spars	3mm

2.3.2. Mesh Generation

The mesh generation of the structural model can be selected per element and is generated automatically. This allows having a finer mesh for the smaller elements such as the stiffeners. Due to the complex shape of the wing, an unstructured mesh is chosen. To make sure the connected surfaces, such as the ribs and wing surface, are sharing nodes, the capture option is enabled. This option searches the selected mesh for nodes within a given tolerance and gives the option to coincide with these nodes. This simplifies the need for possible connections between the ribs and the wing surface.

The mesh size for the stiffeners is chosen to be 0.5 of the stiffener height. The mesh size of the wing surface, ribs and spars is determined in the mesh convergence study in section 3.2.

The stiffeners need to be connected to their respective surfaces. This can be done with the tie constraint which ties two surfaces together for the duration of a simulation[44]. The stiffeners are all tied to the respective surfaces.

2.3.3. Loads

After defining all structural elements and assigning materials, the next step involves applying loads to the model. This model incorporates four representative loads that signify the forces a wing experiences during cruise: the aerodynamic load, the fuel weight, the engine weight, and the intrinsic weight of the wing.

Aerodynamic Load

The aerodynamic load is derived directly from the outcomes of the CFD analysis. The absolute pressure data obtained from the analysis can be exported or stored on the cloud through an *Engineering Document*. Both these documents can then be imported into the structural scenario as a pressure load. Figure 2.21 shows the CFD results, which are represented by the coloured points implemented on the wing surface of the structural model. The directional arrows visually represent the aerodynamic pressure on the structural model. Given that absolute pressure data does not contain an orientation, the inclusion of directional arrows in the visualisation becomes imperative for an accurate representation of the compressive load on the wing. As the structural model only consists of a wing starting from the fuselage-wing intersection, the fuselage section shows no arrows, meaning no load.
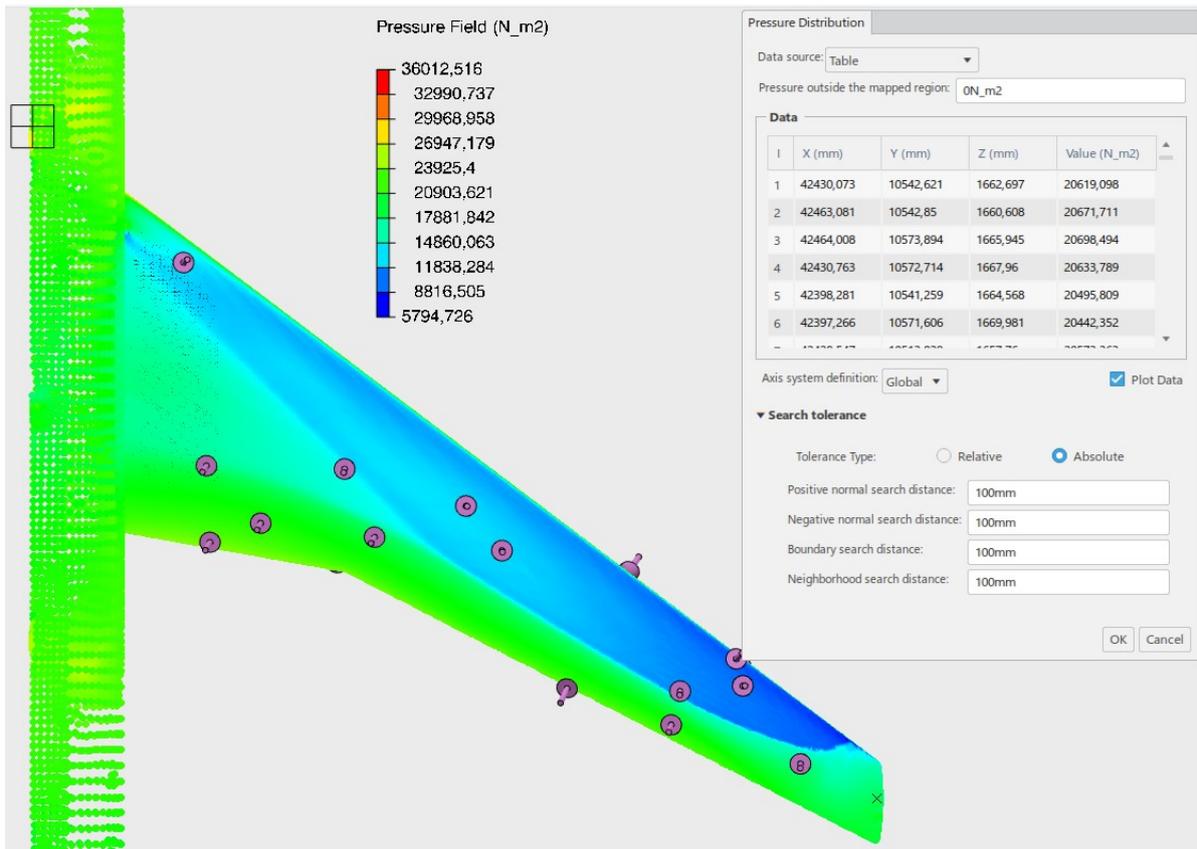
Figure 2.21: The CFD results on the structural model

In order to connect the aerodynamic data with the structural nodes, 3DExperience utilises search controls, consisting of four key components: positive and negative normal, boundary, and neighbourhood normal search distance tolerance. The positive and negative normal search distances determine which source points are included in the simulation based on their distance along the normal vector of the corresponding target surface geometry. The tolerance for in-plane distance specifies the acceptable region for source data points, while any source data points outside the neighbourhood search tolerance are disregarded and excluded from the analysis. Figure 2.22 shows how the boundary search controls are defined and how the source data is connected with the target mesh.
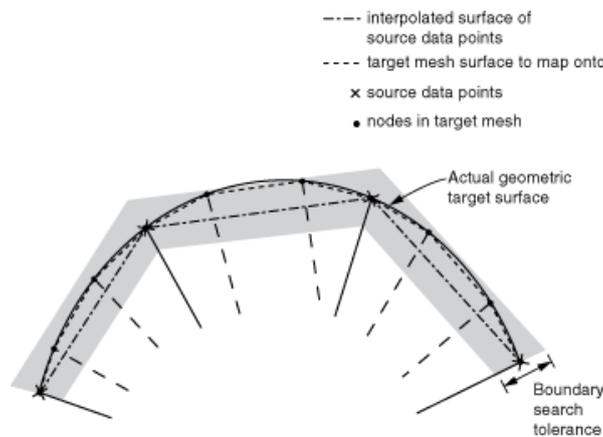
Figure 2.22: The boundary search control for a spatial varying mapping data in 3DExperience[44]

Fuel Load

Aircraft fuel is typically stored in the wings to counteract the lift and reduce displacement. Taylor et al.[26] calculated that the wing tank needs to be extended to at least 76% of the semi-span to be able to comply with the maximum fuel weight of the Boeing 777-200ER. To effectively model the fuel, the wing fuel tank is divided into six sub-tanks based on the location where the rib intersects with the front spar. The first sub-tank spans from the Side of Body connection to 0.26% of the semi-span, while the remaining tanks have a pitch of 0.1% of the semi-span. The result of the sub-tank division is shown in Figure 2.23. The fuel is distributed among the sub-tanks based on the volume percentage of each sub-tank.
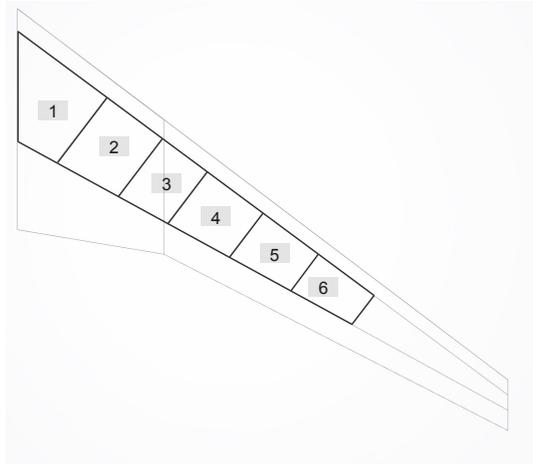
Figure 2.23: The fuel tank divided in 6 subtanks in the wing

According to Taylor et al. [26], the maximum usable fuel weight for the uCRM is determined to be $137,460$ kg. Under cruise conditions, the fuel quantity is approximated to be 50% of the wings' fuel capacity, resulting in $34,365$ kg for a semi-wing configuration. This estimation assumes that all fuel is stored in the wing fuel sub-tanks up to their maximum capacity. The maximum load per sub-tank is computed using the volume of the fuel sub-tank and a fuel density of 803.1 kg/m^3. This load is applied to the lower surface of the respective fuel sub-tank. Figure 2.24 illustrates the distribution of fuel tank loads on the model.

Engine Load

The engine also contributes as a bending relief for the wing. The engine load is represented as a point load at 32.6% of the semi-span at the height of the leading edge spar at the fuselage-wing connection. The weight of an engine is 7,893kg[24]. The load is transferred to the engine rib and the surrounding 4 ribs. Figure 2.24 shows the engine load with the connection to the structure.
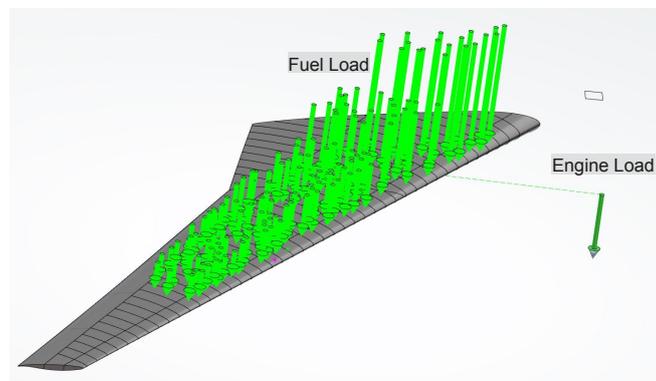
Figure 2.24: The fuel and engine loads on the structural model

<div style="text-align: right; font-size: 3em;">3</div>

Verification

Verification is a crucial part of a simulation-based analysis. This shows the reliability of the results. This chapter includes an analysis of the aerodynamic and structural model.

3.1. Aerodynamic Model

The verification process of the aerodynamic model is based on Tu et al.'s approach[52]. The process includes testing the mesh convergence to analyse the discretization error and quantifying the impact of the turbulence model. Further analysis is conducted on the mesh quality and residual convergence.

3.1.1. Mesh Convergence

In this section, a mesh convergence study is performed. This study uses the baseline unmorphed model with a Spalart-Allmaras turbulence model. The mesh is coarsed by increasing the minimum and maximum mesh sizes, keeping the first layer thickness and number of boundary layers the same. The results are shown in Figure 3.1.
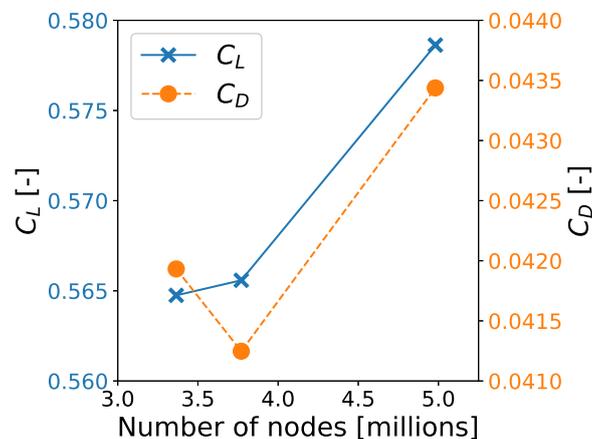
Figure 3.1: The mesh convergence study on the benchmark model

From Figure 3.1, it can be seen that the grid is not converged. The lift coefficient shows an increase with the increase in nodes. The drag coefficient shows first a decrease followed by a steep increase. This shows that a finer mesh is needed to have accurate results. The mesh with the highest node count is applied to the two test cases and is used in chapter 4.

3.1.2. Turbulence Models

As explained in the chapter 2, multiple turbulence models can be used in CFD simulations to close the Navie Stokes equations. The thesis uses the SA turbulence model, and to determine the uncertainty introduced by choosing the turbulence model, the results of the other turbulence models are analysed. Table 3.1 shows the resulting lift and drag coefficient calculated by the three different turbulence models in 3DExperience.

Table 3.1: The sensitivity analys of the turbulence model

Turbulence Model	C_L [-]	C_L compared to SA [%]	C_D [-]	C_D compared to SA [%]
Spalart Allmaras (SA)	0.579	0%	0.0434	0%
SST $k - \omega$	0.572	-1.12%	0.0430	-1.11%
Realizable $k - \epsilon$	0.590	2.05%	0.0434	0.00%

The SST $k-\omega$ turbulence model exhibits a deviation of approximately 1% in both lift and drag coefficients compared to the SA model. The Realizable $k - \epsilon$ turbulence model indicates a deviation of 2% in the lift but no deviation in the drag. It is known that the $k - \epsilon$ method has decreased reliability with complex geometries due to the lack of near wall accuracy[57], which can explain the increased deviation of the lift coefficient.

3.1.3. Mesh Quality

The mesh quality makes sure that the mesh is stable and converging. However, as explained in subsection 2.2.2, automatic meshing in 3D Experience is not a standard approach for aircraft meshing which can influence the quality of the mesh. To evaluate the quality of a mesh, it is necessary to consider its skewness, minimal angle, and y+ values. This assessment is carried out by employing the baseline unmorphed model with the highest node count. Moreover, two test cases were analysed, and similar results were obtained.

Skewness
Skewness is defined by the difference between a cell's actual and ideal shape. Cells with a high skewness can compromise accuracy and destabilize the solution. The desired value of the maximum skewness is 0.95 where the average is around 0.33[58]. Figure 3.2 shows the skewness distribution of the CFD mesh.
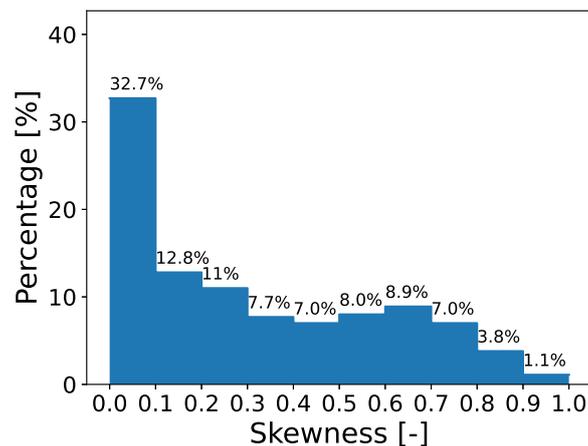
Figure 3.2: The skewness distribution

The average skewness of the mesh is 0.31 with the largest share between 0-0.1. This shows that the mesh quality is acceptable in terms of skewness.

Minimum angle

The minimum angle is defined as the angles between neighbouring cells' edges. As the mesh is a hex-dominant, it means that the mesh consists of different types of cells as shown in Figure 3.4. For convergence, the quadrilateral and triangular meshes have an angle of 30 and 60 degrees, respectively[58]. Figure 3.3 shows the minimum angle distribution.
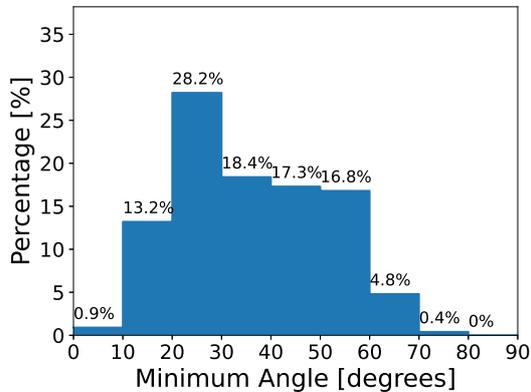
Figure 3.3: The minimum angle distribution
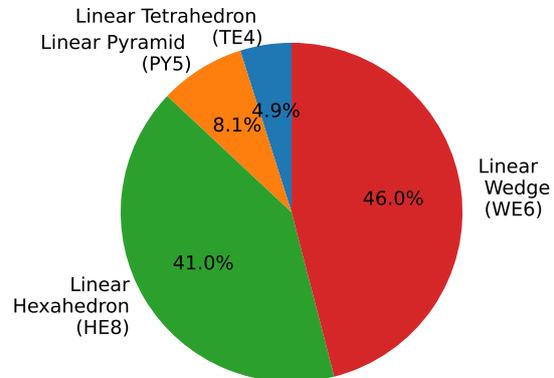
Figure 3.4: The cell type distribution of the mesh

Figure 3.3 shows that the majority of the cells have a minimum angle between 20 and 60 degrees. This also shows that the mesh is acceptable in terms of minimum angle

Normalised wall distance, y+

The last aspect of the mesh quality is the normalised wall distance, $y+$. During the mesh determination, the y+ values are approximated with the Schlichting skin-friction correlation. To target the logarithmic layer (y+ between 30-300), the first layer thickness was set at 2mm for the whole model. In the simulation, the actual y+ values per cell are calculated. Figure 3.5 shows the y+ values of the upper and lower part of the model. The grey area on the fuselage indicates that the y+ is above 1000.
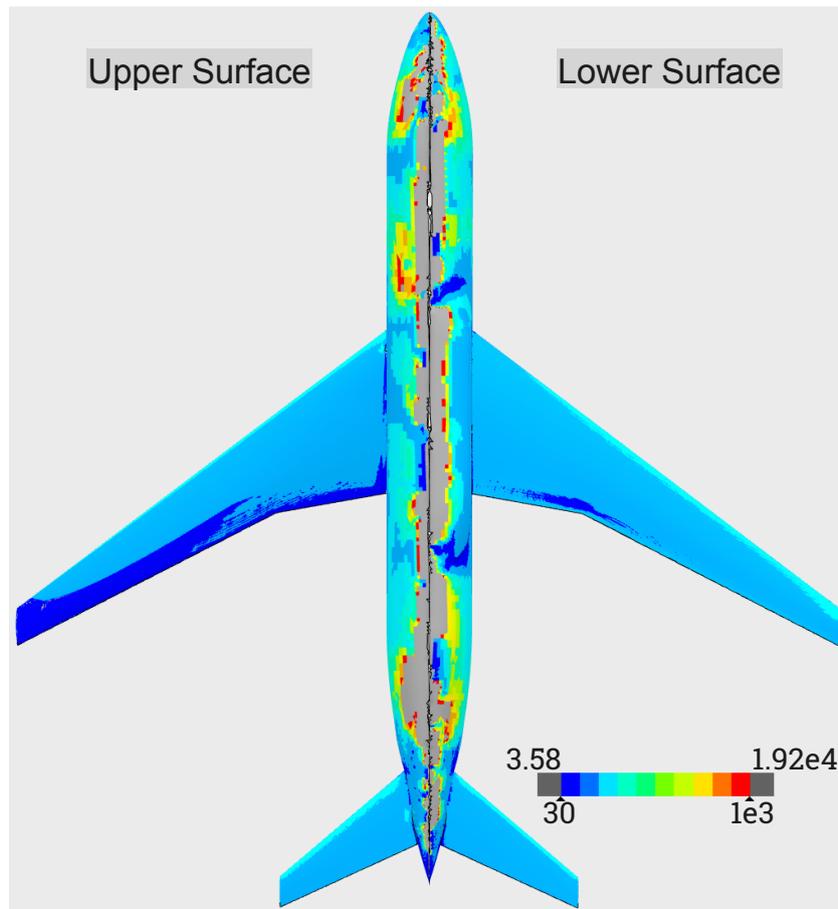
Figure 3.5: The y+ distribution of the baseline model

Figure 3.5 shows that the wing has a y+ over the wing between 30 and 300 as targeted. However, the fuselage shows higher y+ values, especially near the symmetry axis. Although the boundary layers are included in the geometry, they are excluded at the boundaries of the domain as shown in Figure 2.18. This results in high y+ values in these regions.

3.1.4. Residual Convergence
A crucial aspect of a CFD solution is the convergence of the residuals. 3DExperience uses the normalised residual, calculated by normalising the L-2 norm of the current iteration's residual by the L-2 norm of the residual obtained at the first iteration[44]. Figure 3.6 shows the residuals and Figure 3.7 shows the lift and drag force of the simulation of the baseline model.
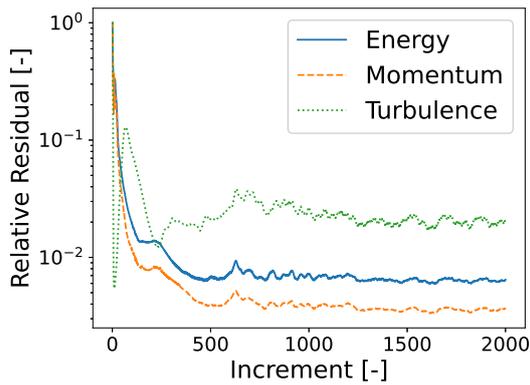
Figure 3.6: The convergence of the energy, momentum and turbulence residuals
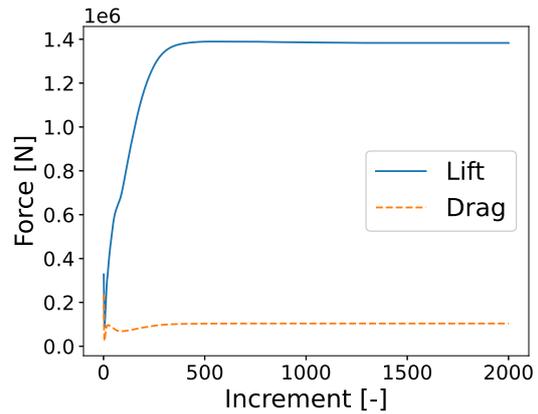
Figure 3.7: The stabilisation of the lift and drag force

A residual value of 1E-4 is deemed to be loosely converged for a CFD simulation. Figure 3.6 demonstrated that this was not reached in 2000 iterations. However, Figure 3.7 shows that the lift and drag force are stabilised. This required further investigation into the local residuals. Figure 3.8 shows the turbulence residual where the light blue border states where the residual is 1e-3.
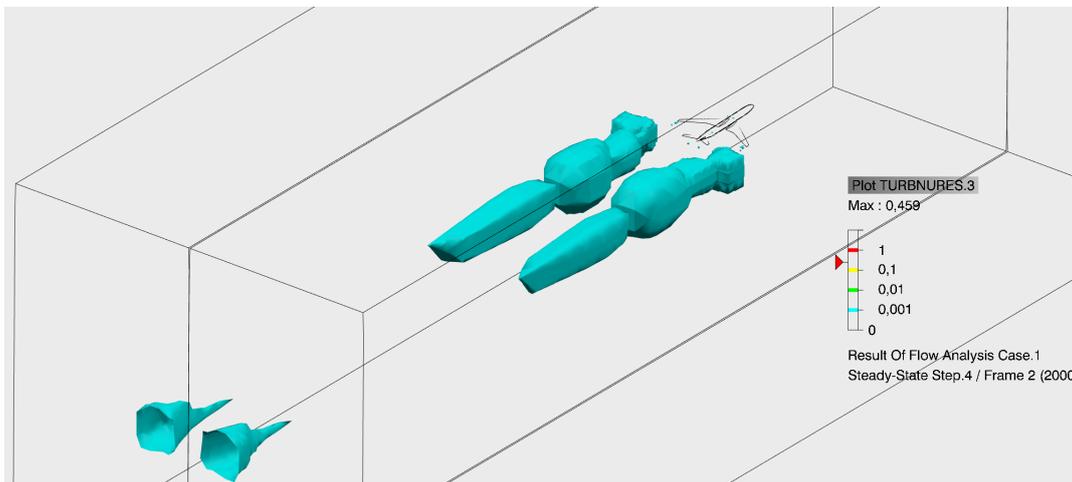
Figure 3.8: The turbulence residual in the domain

It can be seen in Figure 3.8 that the residual values higher than 1E-3 can be found around the tip vortex aft from the aircraft and at the outlet of the domain. It is necessary to incorporate local refinements in the mesh in these regions to enhance its performance. The elevated residual value towards the end of the domain indicates that the domain size needs to be expanded. Unfortunately, due to the imposed node limit, implementing these improvements in the domain and mesh is not feasible.

3.2. Structural Model

To verify the structural model, an extensive analysis is conducted wherein the mesh size and the impact of the loads are carefully evaluated. The process begins with a mesh convergence study using a basic tip load. Once the aerodynamic loads are incorporated into the structural model, the coupling is studied by varying search tolerances. Lastly, a comprehensive mesh convergence study of all loads is performed. Throughout the process, the mesh size of the stringers and stiffeners remains constant, while only the mesh size of the wing surface, ribs, and spar are subject to change.

3.2.1. Mesh Convergence with a Tip Load

As the model consists of multiple loads, the mesh is first tested with one simple tip load. Figure 3.9 shows the geometry with the tip load of 1e5N in the positive z-direction. 3DExperience also introduced a student license limit of 1 million nodes in the structural mesh, limiting the minimal mesh size to 30mm. Figure 3.10 shows the mesh convergence of the tip load.
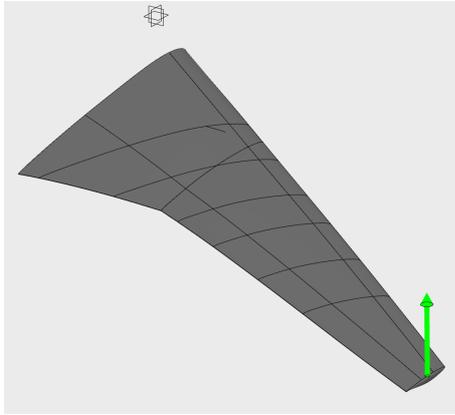
Figure 3.9: The tip load applied to the baseline wing
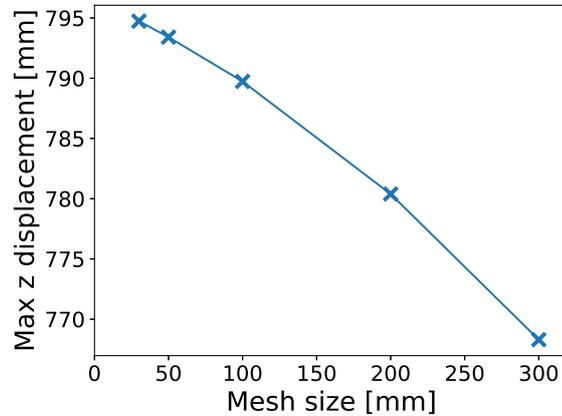
Figure 3.10: Mesh convergence study of a tip load

From Figure 3.10, it can be seen that the mesh slowly is converging towards a lower mesh size. Due to the node limit, the mesh size could not be lowered. Currently, the mesh maintains a uniform size across the entire model. Future investigations may explore alternative configurations, such as a decreasing mesh size towards the tip.

3.2.2. Aerodynamic Load Coupling

The aerodynamic loads are imported into the structural model as explained in subsection 2.3.3. A search tolerance was utilised to couple these loads, and this study examines the impact of varying tolerances. Figure 3.11 shows the result of the maximum z displacement with varying search tolerance for a mesh size of 50mm and 200mm. The four different search tolerances are kept constant.
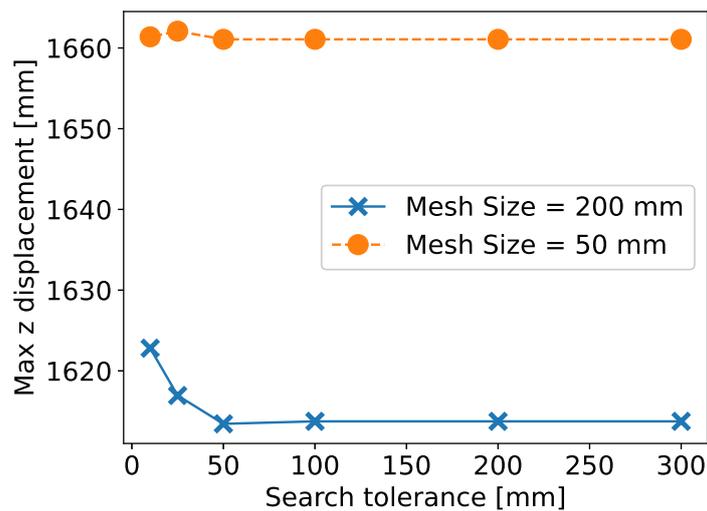
Figure 3.11: Mesh convergence study of the structural analysis

Figure 3.11 shows that the maximum z displacement stabilises after 100mm search tolerance for both the 200mm and 50mm mesh. Therefore, for the rest of the simulations, a search tolerance of 200mm is

used.

3.2.3. Mesh Convergence All Loads

Next, the mesh convergence is tested with all the loads described in chapter 2 with using a 200mm search tolerance. Figure 3.12 shows the results of the convergence study.
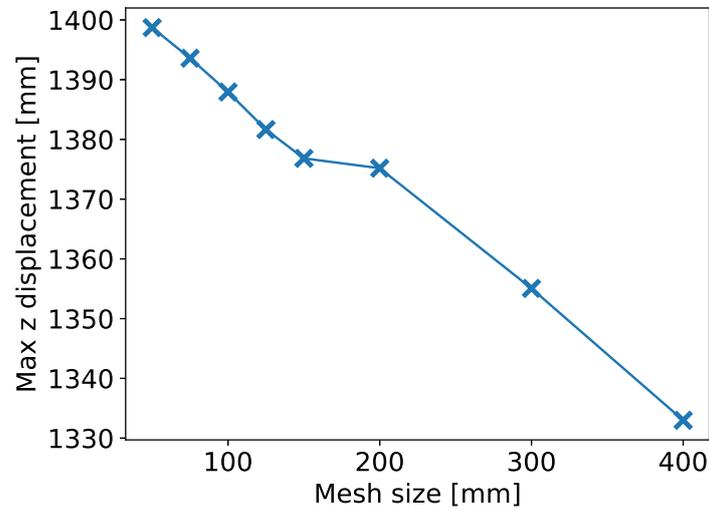
Figure 3.12: Mesh convergence study of the structural analysis

It can be seen that the mesh is not converged. An interesting point is between a mesh size of 150mm and 200mm. It shows some stabilisation; however, a steep gradient appears when the mesh is further decreased. Although the relative difference is small, this shows that the mesh should be further studied. As mentioned before, the current set-up has a constant span-wise mesh size. The setup can be changed such that the tip mesh can have a smaller mesh to account for the larger displacements in that region. For the results, the mesh size of 100mm is used.

Figure 3.12 shows that maximum z displacement is only $\approx 5\%$ of the semi-span. This shows that the assumption of a linear structural analysis is correct.

4

Results & Discussion

This chapter discusses the findings of the analysis. As outlined in our research questions, the baseline unmorphed wing is compared to the two test cases in terms of both aerodynamic and structural aspects. Both a global and local examination of the aerodynamic and structural results are discussed.

4.1. Aerodynamic results

To provide an overview of the global findings, the lift and drag coefficients of the wings are compared to assess the aerodynamic performance of the wing. The lift-to-drag ratio(L/D) is also included to assess the aerodynamic efficiency of the wings. Table 4.1 show the results for the unmorphed and the two morphing wings.

Table 4.1: Aerodynamic Results

	Unmorped Wing: Baseline	Morphed Wing 1: High-Lift	Morphed Wing 2: Efficient
Angle Of Attack [degrees]	2	2	2
C_L [-]	0.57	0.65	0.47
C_D [drag counts]	434	525	317
L/D	13.3	12.4	14.7

The high-lift morphing wing with the downward curvature of the trailing edge shows an increased lift coefficient. However, it also has a higher drag coefficient, which decreases the overall efficiency of the wing in terms of the lift-to-drag ratio. The upward efficient morphing wing shows a lower lift with a corresponding lower drag coefficient. This results in a higher lift-to-drag ratio. The local findings are presented with the pressure distribution and Mach distribution further to investigate the aerodynamic results of the three wings. The locations of the local findings are at the Yehudi (P3), outboard near the tip(P7) and at the tip(P8).

4.1.1. Yehudi (P3)

The first location for the local results is at the Yehudi, P3, to show the influence of the morphing trailing edge on the inboard wing. The airfoil at P3 is the same for all three wings. Figure 4.1 shows the pressure distribution at position 3.
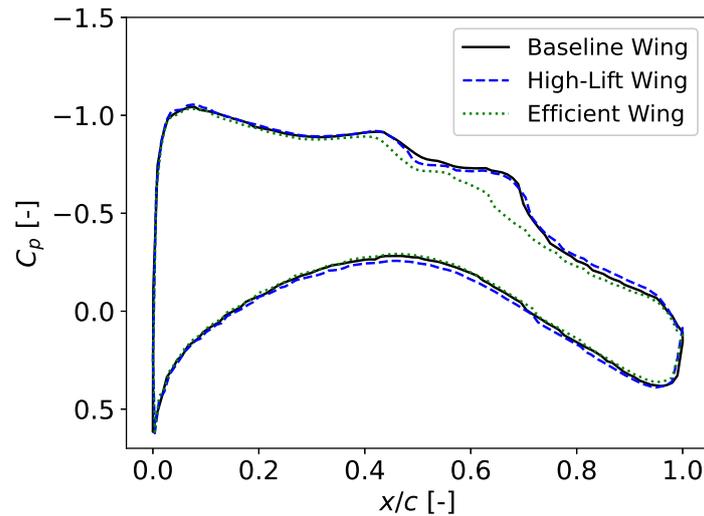
Figure 4.1: The pressure distribution at position 3

There is little to no difference in the pressure distribution between the unmorphed baseline and high-lift wing. However, the efficient wing displays a decline in pressure distribution around 0.6 c. To further illustrate this, Figure 4.2 displays the Mach distribution of both the baseline and efficient wing at P3.
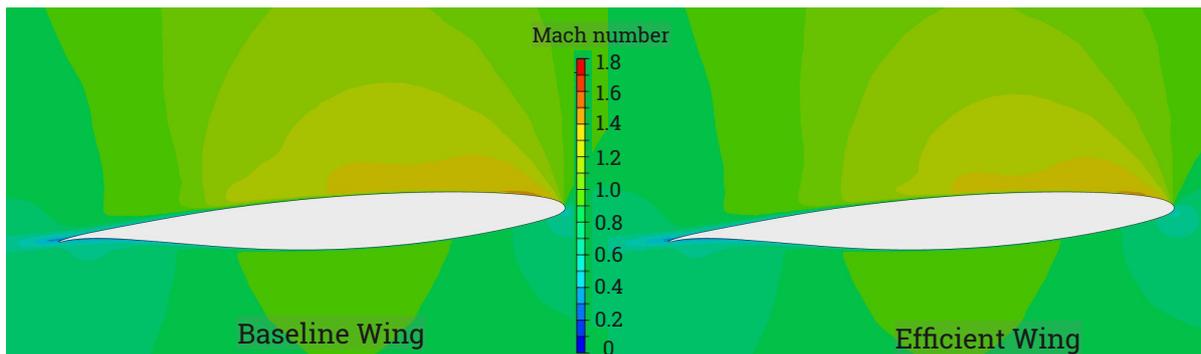
Figure 4.2: The Mach distribution at Yehudi, P3, of the baseline and the efficient wing

From Figure 4.2, it can be seen that the mach distributions exhibit a slight variation. Notably, the Mach number experiences an earlier decrease on the upper surface of the efficient wing compared to the baseline wing. This observation indicates that the alterations made to the outboard wing affect the flow on the inboard wing despite the airfoil remaining the same at that location.

4.1.2. Outboard wing near the tip (P7)

The local aerodynamic results are analysed at P7, near the tip, to analyse the influence of the trailing edge morphing. Similar to P3, the local aerodynamic results are shown in pressure distribution in Figure 4.3.
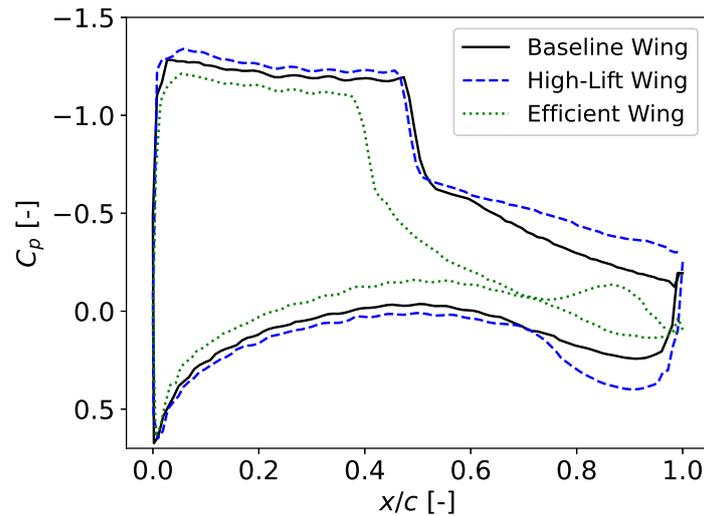
Figure 4.3: The pressure distribution at position 7

The pressure distribution shows a large variation between the different wings. The unmorphed baseline wing shows signs of a weak shock wave around 0.5 c, similar to the high lift morphed wing. Figure 4.4 shows the Mach contour plot around the three airfoils.
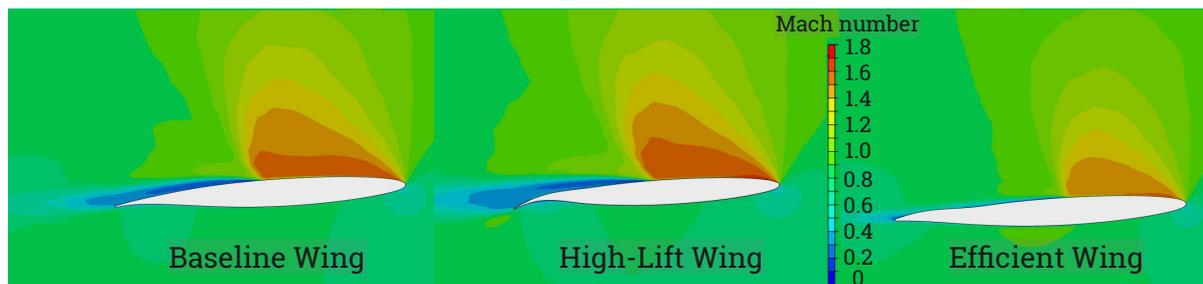
Figure 4.4: The Mach distribution at P7, of the all three wings

However, the high-lift wing has an improved pressure recovery. This can be linked to the shape of the high-lift trailing edge, as it is more flat than the baseline wing. Another prominent phenomenon in the pressure distribution is *aft loading*. The high lift wing shows an increase in pressure on the lower side around 0.7 c due to its concave shape. The efficient morphed wing shows that the shock wave is moved towards the leading edge around 0.4 c. The shock wave, in combination with the slightly concave shape of the trailing edge, causes the pressure coefficient to almost reach 0. Due to the convex shape at 0.9c, the pressure coefficient rises again.

4.1.3. Tip (P8)
Lastly, the flow is analysed at the tip, where Figure 4.5 shows the pressure distribution and Figure 4.6 shows the mach distribution around the three airfoils at position 8.
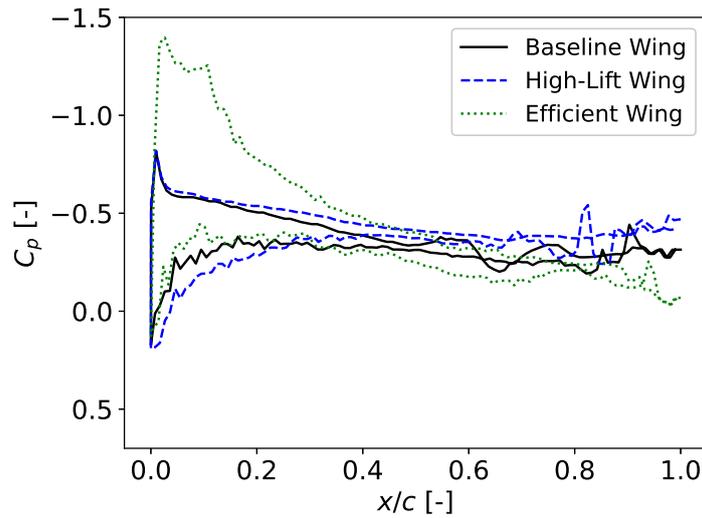
Figure 4.5: The pressure distribution at the tip
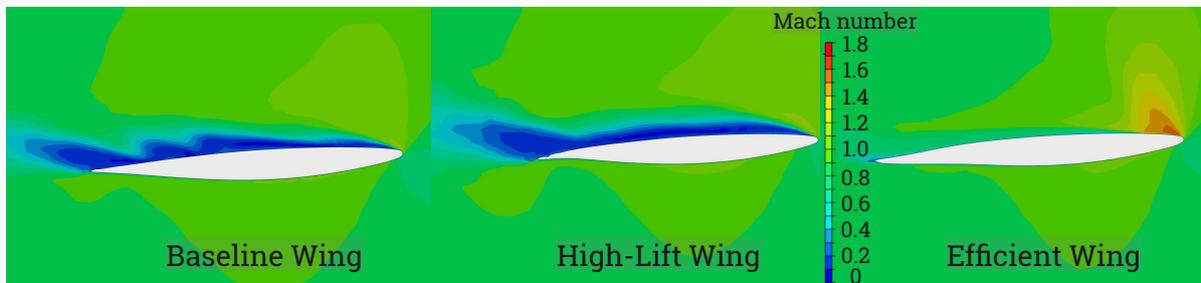
Figure 4.6: The Mach distribution at P8, of the all three wings

The pressure distribution of the tip airfoils is not similar to the previous two positions. The high-lift and unmorphed wings lack a sharp pressure peak. The Mach profile of both wings displays low Mach numbers along the upper portion of the airfoil, commencing at the leading edge, indicating separation. However, the efficient wing exhibits attachment with a pressure peak at the leading edge, which rapidly decreases, revealing the presence of a shock wave. Therefore, the efficient wing is the only one where the flow remains attached at the tip.

4.2. Structural results

The results of the CFD are used to evaluate the different structural responses to the different morphing settings. First, the global structural response is analysed. Table 4.2 shows the tip displacement of the different wings with the corresponding lift coefficient.

Table 4.2: Structural Results

	Unmorped Wing	Morphed Wing 1	Morphed Wing 2
C_L [-]	0.57	0.65	0.47
Tip Displacement [mm]	1388	1729	842

As expected, the higher the lift coefficient, the higher the tip displacement. Figure 4.7 shows the Von Mises stress distribution of the upper and lower geometry of the baseline model.
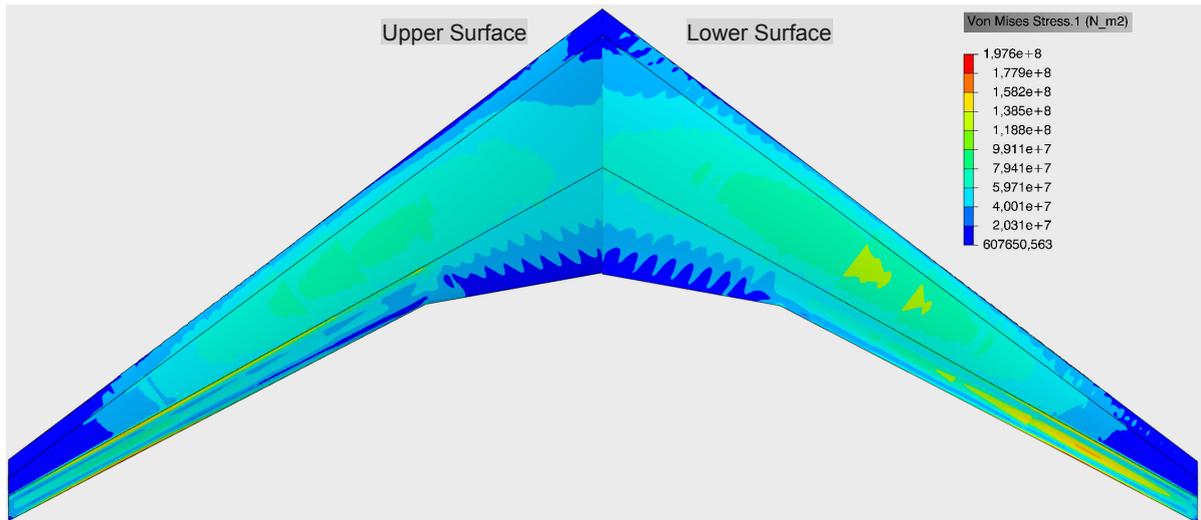
Figure 4.7: The Von Mises stress on the baseline geometry

Upon analysis, it is evident that the outboard trailing edge exhibits the highest Von Mises stress. Additionally, an examination of the high-lift wing revealed a collapse of the trailing edge, indicating a need for additional reinforcement to prevent such occurrences. Despite this, the maximum local stresses remain below the yield stress indicated in Table 2.7, thus preventing any plastic deformation. In addition, the evaluation of the structural response entails the assessment of the span-wise displacement and aeroelastic twist.

4.2.1. Displacements

The span-wise displacement shows the displacement distribution. Figure 4.8 shows the span-wise displacement for the three wings.
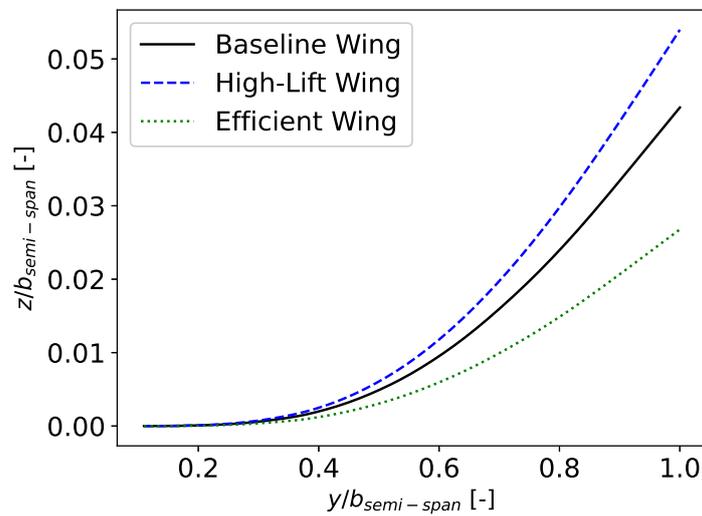
Figure 4.8: The span-wise displacement of the quarter chord

check axis It can be seen that all three wings show no z displacement until 0.3 $b_{y,semi-span}$. This can be explained through the fuel tanks and the engine weight counteracting the upward lift force. It can also be seen that the wing bending is most prominent between 0.4-0.7 $b_{y,semi-span}$. After 0.7 $b_{y,semi-span}$, the span-wise displacement seems almost linear.

4.2.2. Twist

Next, the span-wise twist deflection is shown in Figure 4.9 of the three wings compared to the original deflection.
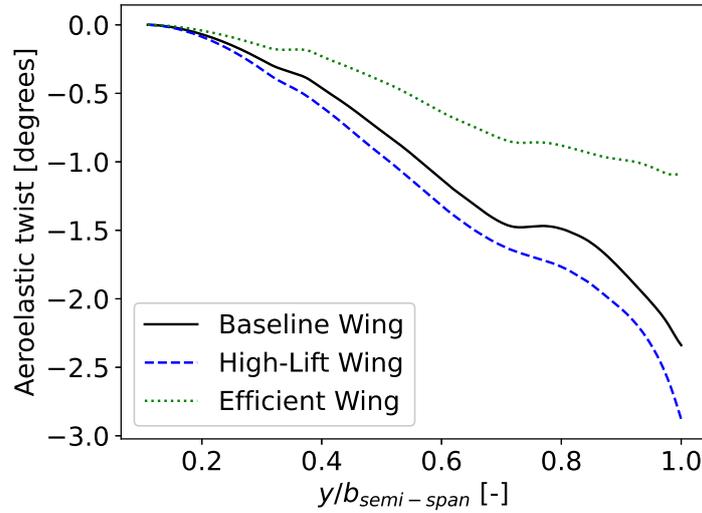
Figure 4.9: The span-wise twist deflection,θ_y

The high lift morphed wing exhibits greater twist deflection than the efficient morphed wing, similar to the tip displacement. At a distance of 0.4 $b_{y,semi-span}$, all three wings display a change in twist deflection, which could be attributed to the engine weight and engine rib countering the twist. Another shift in behaviour occurs at approximately 0.7 $b_{y,semi-span}$, which is due to the fuel weight as the fuel tank terminates around 0.75 $b_{y,semi-span}$. Overall, it is apparent that the engine and fuel tank have a more pronounced impact on the twist distribution at lower lift coefficient wings.

To compare the total twist of the three wings, the jig twist is added to the aeroelastic twist. Figure 4.10 shows the original jig twist and the total twist of the three wings.
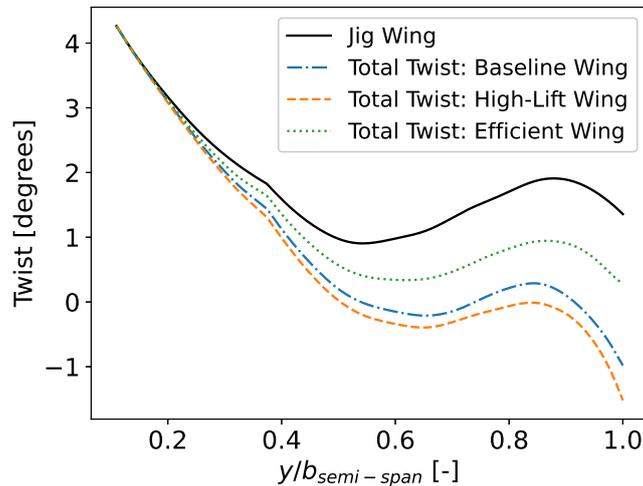
Figure 4.10: The span-wise total twist deflection

The jig twist shows a smooth profile with the exception around 0.4 $b_{y,semi-span}$. This is around the Yehudi, where the trailing edge sweep also changes which can explain the change in twist angle. Figure 4.10 shows that the shifts in behaviour can not be clearly found back in the total twist deflection.

5

Conclusion & Recommendations

The goal of this thesis was to develop a one-way coupled aerodynamic and structural assessment method in 3DExperience to analyse a span-wise trailing edge morphing wing on a transonic transport aircraft. Therefore the research was divided into two parts. The first part focused on the development of the assessment methodology within the 3DExperience platform.

The 3DExperience software effectively incorporates parameterisation and automation for the wing's design. With a parameterised planform as a starting point and the use of UDFs and Knowledge Patterns for repetitive geometrical elements, the wing's geometry is easily modifiable. Moreover, the software's capability to model span-wise trailing edge morphing is versatile, achieved through spline manipulation resulting in a smooth airfoil. Nevertheless, some limitations exist as the morphing trailing edge shapes need manual creation to fulfil the given requirements.

The aerodynamic CFD solver uses the RANS equations and integrates a range of settings for both turbulence models and linear solvers. Nevertheless, the node count restriction imposed by the academic license has had an adverse effect on the aerodynamic model's reliability. Furthermore, the automatic HDM-dominant feature has been found to over-refine in certain areas, while the boundary layers are not being applied to the symmetry axis.

The structural FEA solver is based on the Abaqus finite element analysis technology. The success of the analysis is heavily influenced by the geometric model, which requires careful consideration of the implementation of shell elements and necessary groups. With the knowledge pattern, the automatisation of creating geometrical groups showed its benefit. Additionally, importing aerodynamic loads can be easily accomplished using our spatial varying mapping function for pressure loads and adjustable search tolerances.

The second part of the thesis focused on the results of the one-way coupled aerostructural analysis. By creating a high lift and an aerodynamically efficient test case by applying transonic design principles, the effects of span-wise trailing edge morphing can be compared to the baseline wing. Results showed that the high-lift morphing wing increased both lift and drag coefficients, resulting in lower aerodynamic efficiency than the baseline unmorphed wing. Conversely, the efficient wing displayed an increase in aerodynamic efficiency but a decrease in lift and drag. Additionally, the study discovered that changes in the outboard trailing edge impacted the flow at the unmorphed Yehudi and also at the region before the influence region.

As a result of the varying aerodynamic forces, the wing's deformation was also affected. In the high-lift configuration, the tip displacement increased; in the efficient configuration, it decreased compared to the baseline model. Moreover, the aeroelastic twist became more pronounced with higher lift coefficients. Additionally, it was found that the influence of other loads on the aeroelastic twist decreased as the aerodynamic load increased.

5.1. Recommendations for future work

As this is one of the initial studies on utilising 3DExperience for aerostructural analysis, it is recommended that further research be conducted to explore the software's potential.

Regarding methodology, there are a few suggestions to enhance the approach. Firstly, manual changes to the trailing edge can be improved with the shape optimisation module available in 3DExperience. Unfortunately, this feature was not explored in this thesis due to time constraints.

The study showed that the structural mesh was not fully converged. Analysing a non-uniform mesh with a decreasing mesh size towards the tip is recommended. Overall, both the structural and aerodynamic model's meshes in 3DExperience exhibited drawbacks. It may be worthwhile to investigate using an external meshing program as an alternative. It is possible to import a mesh into 3DExperience for analysis.

Upon evaluation, the findings indicate that the trailing edge experienced a collapse due to the absence of any structural reinforcement in the current approach. To prevent this outcome in future research, it is recommended to incorporate additional stiffness to the trailing edge geometry. This will help to ensure the integrity of the design.

5.1.1. Aerostructural Analysis

The initial aim of the thesis was to conduct an aerostructural analysis that was fully coupled. However, due to constraints on time, this goal was downsized to a one-way coupled aerostructural analysis. Nonetheless, there has been some investigation into developing a fully connected system. A crucial aspect is the coupling that exports the deformed structural model back into the meshes. There are two methods to achieve this.

The first method involves using the extract tessellated shape function. In theory, this extracts the deformed mesh. This approach has the potential to be automated through a macro script. However, one drawback is that the smoothness of the airfoil decreases as the mesh, not the shape, is exported. This could lead to gaps in the geometry if the meshes are not entirely aligned, resulting in the aerodynamic solver ceasing to analyze.

The second method involves using the *Virtual to Real Shape Morphing* app. This app combines the original geometry with the vector displacement of the structural model to create a new shape. This ensures the wing profile remains smooth, but it cannot be employed for small components like stiffeners and stringers as they do not show, which are necessary for the second structural iteration. Further research can be done to determine a solution.

References

[1] H. Ritchie. "Climate change and flying: what share of global CO2 emissions come from aviation?" In: *Our World in Data* (2020). URL: `https://ourworldindata.org/co2-emissions-from-aviation`.

[2] European Comission. "Transport and The Green Deal". In: (). Visitd 5-10-2023. URL: `https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal/transport-and-green-deal_en`.

[3] D. Li, S. Zhao, A. Da Ronch, J. Xiang, J. Drofelnik, Y. Li, L. Zhang, Y. Wu, M. Kintscher, H. P. Monner, A. Rudenko, S. Guo, W. Yin, J. Kirn, S. Storm, and R. De Breuker. "A review of modelling and analysis of morphing wings". In: *Progress in Aerospace Sciences* 100 (June 2018), pp. 46–62. ISSN: 0376-0421. DOI: `10.1016/J.PAEROSCI.2018.06.002`.

[4] F. E. C. Culick. "The Wright Brothers: First Aeronautical Engineers and Test Pilots". In: *AIAA JOURNAL* 41 (6 2003), pp. 8–11.

[5] R. De Breuker, M. M. Abdalla, and Z. Gürdal. "A generic morphing wing analysis and design framework". In: *Journal of Intelligent Material Systems and Structures* 22 (10 July 2011), pp. 1025–1039. ISSN: 1045389X. DOI: `10.1177/1045389X11414958`.

[6] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman. "A review of morphing aircraft". In: *Journal of Intelligent Material Systems and Structures* 22 (9 June 2011), pp. 823–877. ISSN: 1045389X. DOI: `10.1177/1045389X11414084`.

[7] B. Perry, S. R. Cole, and G. D. Miller. "Summary of an active flexible wing program". In: *Journal of Aircraft* 32 (1 1995), pp. 10–15. ISSN: 00218669. DOI: `10.2514/3.46677`.

[8] S. G. Powers, L. D. Webb, E. L. Friend, and W. A. Lokos. "Flight Test Results From a Supercritical Mission Adaptive Wing With Smooth Variable Camber". In: *NASA Technical Memorandum 4415* (Nov. 1992).

[9] M. Sinapius, H. P. Monner, M. Kintscher, and J. Riemenschneider. "DLR's Morphing Wing Activities within the European Network". In: *Procedia IUTAM* 10 (2014). Mechanics for the World: Proceedings of the 23rd International Congress of Theoretical and Applied Mechanics, ICTAM2012, pp. 416–426. ISSN: 2210-9838. DOI: `10.1016/j.piutam.2014.01.036`.

[10] R. W. Wlezien, G. C. Horner, A. R. McGowan, S. L. Padula, M. A. Scott, R. J. Silcox, and J. S. Harrison. "Aircraft Morphing program". In: vol. 3326. 1998. DOI: `10.1117/12.310633`.

[11] J. N. Kudva, C. A. Martin, L. B. Scherer, A. P. Jardine, A. R. McGowan, R. C. Lake, G. P. Sendeckyj, and B. P. Sanders. "Overview of the DARPA/AFRL/NASA Smart Wing program". In: vol. 3674. 1999. DOI: `10.1117/12.351561`.

[12] J. Frota, K. Nicholls, J. Whurr, M. Müller, P. Gall, J. Loerke, K. Macgregor, P. Schmollgruber, J. Russell, M. Hepperle, S. Rodriguez, K. Taupin, J. Godard, S. Dron, K. Plötner, and G. Gallant. *NACRE: Final Activity Report 2005-2010*. Tech. rep. The European Commision, 2011.

[13] T. Hellstrom J. König. "The Clean Sky "Smart Fixed Wing Aircraft Integrated Technology Demonstrator": Technology targets and project status". In: *ICAS2010: 27th International congress of the aeronautical sciences*. 2010.

[14] M. Kintscher, M. Wiedemann, H. P. Monner, O. Heintze, and T. Kühn. "Design of a smart leading edge device for low speed wind tunnel tests in the European project SADE". In: *International Journal of Structural Integrity* 2 (4 2011). ISSN: 17579864. DOI: `10.1108/17579861111183911`.

[15] I. Dimino, F. Moens, R. Pecora, A. De Gaspari, S. Ricci, S. Ameduri, A. Concilio, U. Mercurio, and G. M. Carossa. "Morphing Wing Technologies within the Airgreen 2 Project". In: 2022. DOI: `10.2514/6.2022-0718`.

[16] S. Kota, P. Flick, and F. S. Collier. "Flight Testing of FlexFloil Adaptive Compliant Trailing Edge". In: American Institute of Aeronautics and Astronautics, Jan. 2016. ISBN: 978-1-62410-393-3. DOI: 10.2514/6.2016-0036.

[17] H. P. Monner. "Realization of an optimized wing camber by using formvariable flap structures". In: *Aerospace Science and Technology* 5 (2001), pp. 445–455. DOI: 10.1016/S1270-9638(01)01118-X.

[18] 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". In: Sept. 2015. DOI: 10.4271/2015-01-2565.

[19] X. Shi, Y. Yang, Z. Wang, S. Zhang, X. Sun, and W. Feng. "Design and Shape Monitoring of a Morphing Wing Trailing Edge". In: *Aerospace* 10.2 (2023). ISSN: 2226-4310. DOI: 10.3390/aerospace10020127.

[20] Z. Lyu, G. K. W. Kenway, and J. R. R. A. Martins. "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark". In: *AIAA Journal* 53.4 (2015), pp. 968–985. DOI: 10.2514/1.J053318.

[21] A. de Boer, A. H. van Zuijlen, and H. Bijl. "Comparison of conservative and consistent approaches for the coupling of non-matching meshes". In: *Computer Methods in Applied Mechanics and Engineering* 197 (49-50 2008). DOI: 10.1016/j.cma.2008.05.001.

[22] B. Kanaparthi, E. Vathsalan, D. Panda, G. Prasad, G. Umashankar, S. S. Chittur, G. Avula, A. Pashilkar, A. C. Pankaj, D. Singh, and Vijayakrishnan. "Multidisciplinary Design Optimization (MDO) of a Typical Civil Aircraft Wing Using 3DEXPERIENCE Platform from Dassault Systemés". In: *Advances in Multidisciplinary Analysis and Optimization*. Ed. by P. Pradeep Pratapa, G. Saravana Kumar, P. Ramu, and R. K. Amit. Springer Nature Singapore, 2023, pp. 97–112. ISBN: 978-981-19-3938-9.

[23] J. Vassberg, M. Dehaan, M. Rivers, and R. Wahls. "Development of a Common Research Model for Applied CFD Validation Studies". In: Aug. 2008. DOI: 10.2514/6.2008-6919.

[24] T. R. Brooks, G. K. W. Kenway, and J. R. R. A. Martins. "Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings". In: *AIAA Journal* 56.7 (2018), pp. 2840–2855. DOI: 10.2514/1.J056603.

[25] G. Kenway, G. Kennedy, and J. R. R. A. Martins. "Aerostructural optimization of the Common Research Model configuration". In: *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2014. DOI: 10.2514/6.2014-3274.

[26] J. D. Taylor and D. F. Hunsaker. "Characterization of the Common Research Model Wing for Low Fidelity Aerostructural Analysis". In: *AIAA 2021-1591, AIAA SciTech 2021 Virtual Forum*. 2021. DOI: 10.2514/6.2021-1591.

[27] G. Kennedy, J. R. R. A. Martins, and G. K. Kenway. "High Aspect Ratio Wing Design: Optimal Aerostructural Tradeoffs for the Next Generation of Materials". In: *52nd Aerospace Sciences Meeting*. 2014. DOI: 10.2514/6.2014-0596.

[28] N. P. M. Werter, R. De Breuker, C. S. Beaverstock, and M. I. Friswell. "Feasible conceptual design of morphing structures". In: *22nd AIAA/ASME/AHS Adaptive Structures Conference* (2014). DOI: 10.2514/6.2014-1260.

[29] G. K.W. Kenway, G. J. Kennedy, and J. R.R.A. Martins. "A CAD-free approach to high-fidelity aerostructural optimization". In: 2010. ISBN: 9781600869549. DOI: 10.2514/6.2010-9231.

[30] D. A. Burdette and J. R.R.A. Martins. "Design of a transonic wing with an adaptive morphing trailing edge via aerostructural optimization". In: *Aerospace Science and Technology* 81 (2018), pp. 192–203. ISSN: 1270-9638. DOI: 10.1016/j.ast.2018.08.004.

[31] J. Lépine, F. Guibault, J. Y. Trépanier, and F. Pépin. "Optimized nonuniform rational B-spline geometrical representation for aerodynamic design of wings". In: *AIAA Journal* 39 (11 2001). ISSN: 00011452. DOI: 10.2514/2.1206.

[32] B. M. Kulfan and J. E. Bussoletti. ""Fundamental" parametric geometry representations for aircraft component shapes". In: vol. 1. 2006, pp. 547–591. ISBN: 1563478234. DOI: 10.2514/6.2006-6948.

[33] H. Sobieczky. "Parametric Airfoils and Wings". In: *Recent Development of Aerodynamic Design Methodologies: Inverse Design and Optimization*. Ed. by K. Fujii and G. S. Dulikravich. 1998, pp. 71–87. DOI: 10.1007/978-3-322-89952-1_4.

[34] R. Vos and S. Farokhi. *Introduction to transonic aerodynamics*. English. Vol. Fluid mechanics and its applications. Germany: Springer, 2015. ISBN: 978-94-017-9746-7. DOI: 10.1007/978-94-017-9747-4.

[35] C.D. Harris. *NASA Supercritical Airfoils: A Matrix of Family-related Airfoils*. NASA technical paper. National Aeronautics, Space Administration, Scientific, and Technical Information Division, 1990.

[36] K. J. Standish and C. P. van Dam. "Aerodynamic Analysis of Blunt Trailing Edge Airfoils ". In: *Journal of Solar Energy Engineering* 125.4 (Nov. 2003), pp. 479–487. DOI: 10.1115/1.1629103.

[37] J. D. Jr. Anderson. *Computational Fluid Dynamics: The Basics with Applications*. New York, NY: McGraw-Hill Education, 2007.

[38] DS Portofolio. *Fluid Simulations Essentials Tutorial: A1 - Review of CFD Fundamentals*. Dassault Systems, 2022.

[39] J. H. Ferziger, M. Perić, and R. L. Street. *Computational Methods for Fluid Dynamics*. Fourth Edition. Springer, 2020.

[40] J. Dudek. "Evaluation of Full Reynolds Stress Turbulence Models in FUN3D for Axisymmetric Jet Flow". In: *AIAA Scitech Forum*. Jan. 2019. DOI: 10.2514/6.2019-2332.

[41] A. Aabid, L. N. B. Khairulaman, and S. A. Khan. "Analysis of Flows and Prediction of CH10 Airfoil for Unmanned Arial Vehicle Wing Design". In: *Advances in Aircraft and Spacecraft Science* 8 (2 2021), pp. 87–109. ISSN: 22875271. DOI: 10.12989/aas.2021.8.2.087.

[42] S. V. and I. S. R. A. "Comparative Study on the Prediction of Aerodynamic Characteristics of Mini - Unmanned Aerial Vehicle with Turbulence Models". In: *International Journal of Aviation, Aeronautics, and Aerospace* (2021). DOI: 10.15394/ijaaa.2021.1559.

[43] R. El Maani, S. Elouardi, B. Radi, and A. El Hami. "Study of the turbulence models over an aircraft wing". In: *Incertitudes et fiabilité des systèmes multiphysiques* 2 (2 2018). DOI: 10.21494/iste.op.2018.0306.

[44] *User Assistance R2023x*. Dassault Systems, 2023.

[45] P. Spalart and S. Allmaras. "A one-equation turbulence model for aerodynamic flows". In: *30th Aerospace Sciences Meeting and Exhibit*. 1992. DOI: 10.2514/6.1992-439.

[46] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemishere Publisching Corporation, 1980.

[47] H. A. van der Vorst. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems". In: *SIAM Journal on Scientific and Statistical Computing* 13.2 (1992), pp. 631–644. DOI: 10.1137/0913035.

[48] H. Schlichting and K. Gersten. *Boundary-Layer Theory*. 2016. DOI: 10.1007/978-3-662-52919-5.

[49] O. de Almeida and P. R. Souza. "Flow Characterization from a NACA Air Intake Assembled in a Canard Type Aircraft". In: *Engenharia Térmica (Thermal Engineering)* 16 (May 2017), pp. 03–10. DOI: 10.5380/reterm.v16i1.62183.

[50] P. R. Spalart and C. L. Rumsey. "Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations". In: *American Institute of Aeronautics and Astronautics* (2007).

[51] W. Sutherland. "The viscosity of gases and molecular force". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 36.223 (1893), pp. 507–531. DOI: 10.1080/14786449308620508.

[52] J. Tu, G. Yeoh, and C. Liu. *Computational Fluid Dynamics*. Third Edition. Butterworth-Heinemann, 2019. ISBN: 978-0-08-101127-0. DOI: 10.1016/B978-0-08-101127-0.00006-4.

[53] C. Hirsch. Second Edition. Oxford: Butterworth-Heinemann, 2007. ISBN: 978-0-7506-6594-0. DOI: 10.1016/B978-075066594-0/50039-4.

[54] 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". In: *Journal of Aircraft* 53.1 (2016), pp. 276–293. DOI: `10.2514/1.C033328`.

[55] T. Economon, F. Palacios, and J. Alonso. "An Unsteady Continuous Adjoint Approach for Aerodynamic Design on Dynamic Meshes". In: vol. 53. June 2014. DOI: `10.2514/6.2014-2300`.

[56] F. Goetten, D. Finger, M. Marino, C. Bil, M. Havermann, and C. Braun. "A Review of Guidelines and Best Practices for Subsonic Aerodynamic Simulations Using RANS CFD". In: *APISAT 2019: Asia Pacific International Symposium on Aerospace Technology*. Gold Coast, Australia: Engineers Australia, 2019, pp. 227–245.

[57] D. C. Wilcox. *Turbulence Modeling for CFD*. La Cañada Flintridge, CA: DCW Industries, 2006.

[58] *Ansys Fluent User's Guide*. "13th edition". Ansys Fluent Inc., 2010.

A
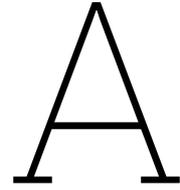
Knowledge Patterns

A.1. Rib and Rib Stiffeners Generation

```
1  let i,j,k,a,b,c,C, EngineRibPosition (Integer)
2  let R3,R4,K,EnginePosition(Real)
3  let temp1,temp3,temp4,temp5,temp6,temp7(Feature)
4  let Curve1,Curve2,Curve3(Curve)
5  let GEOSET1,GEOSET2,GEOSET3(OpenBodyFeature)
6  let Point1,Point2,Point3,Point4,Point5,Point6,Point7,Point8,Point9,Point10 (point)
7  let Rb,WingSurface,InboardRibs,EngineRibs,OutboardRibs,RibStiffenersJoin,EngineSupportRibs,
       Surface1,Surface2, EngineRibStiffenersJoin (Surface)
8  let EngPos(length)
9
10 //Wing Surface Elements
11 if MorphingSetting==0
12 {Surface1=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface\Upper Wing Surface`
13 Surface2=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface`}
14 else if MorphingSetting==1
15 {Surface1=`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1. Lower Surface\M1.
       Upper Surface`
16 Surface2=`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1. Lower Surface`}
17 else if MorphingSetting==2
18 {Surface1=`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2. Lower Surface\M2.
       Upper Surface`
19 Surface2=`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2. Lower Surface`}
20
21 i =1
22 for i while i<=`Rib Parameters\Number of Ribs`
23 {       GEOSET1=new("OpenBodyFeature","Rib"+i,`1. Wing Planform\1.3. Wingbox\1.3.1. Ribs`)
24         GEOSET2 = new("OpenBodyFeature","Guides Rib"+i,GEOSET1)
25
26         // Iniatite UDF
27         temp3=CreateOrModifyTemplate("UDF_Ribs",GEOSET2,Ribs,i)
28
29         temp3->SetAttributeObject("Front spar",`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\
               guideline for wing spars\Front spar position`)
30         temp3->SetAttributeObject("Aft Wingbox Connection",`1. Wing Planform\1.3. Wingbox
               \1.3.1. Ribs\Guidelines for Ribs\Aft Wingbox Connection`)
31         temp3 -> SetAttributeObject("Aft Rib Connection",`1. Wing Planform\1.3. Wingbox
               \1.3.1. Ribs\Guidelines for Ribs\Aft Rib Connection`)
32         temp3->SetAttributeObject("Upper Wing Surface",Surface1)
33         temp3->SetAttributeObject("Lower Wing Surface",Surface2)
34         temp3->SetAttributeObject("Start_Front_Spar",`1. Wing Planform\1.3. Wingbox\1.3.3.
               Spar\guideline for wing spars\Start Front Spar`)
35         temp3->SetAttributeObject("Rib Axis Guideline",`1. Wing Planform\1.3. Wingbox\1.3.1.
               Ribs\Guidelines for Ribs\Rib Axis Guideline`)
36         R3=i/(`Rib Parameters\Number of Ribs`+1)
37         temp3->SetAttributeReal("Ratio",R3)
38         //terminate template instantiation
39         EndModifyTemplate(temp3)
```

```
40
41
42
43
44
45        // Stringer Holes deactivated
46        //      a=1
47
48        //      for a while a<=UpperStringers.Size(){
49        //              Point1=UpperStringers.GetItem(a).GetAttributeObject("Connection to
          Aft Spar")
50        //              Point2=temp3.GetAttributeObject("Rib End Point")
51        //              Point3=UpperStringers.GetItem(a).GetAttributeObject("Stringer Start
          Point")
52
53        //              if Point1->coord(2)>Point2->coord(2)
54        //              {if Point3->coord(1)<Point2->coord(1)
55        //                      {temp5=CreateOrModifyTemplate("UDF_UpStrHo",GEOSET2,
          UpperStringerHoles,UpperStringerHoles.Size()+1)
56        //                              temp5->SetAttributeObject("Start Front Spar",`Wing
          Planform\Wingbox\Spar\guideline for wing spars\Start Front Spar`)
57        //                              temp5->SetAttributeObject("Rib Axis Guideline
          ",`Wing Planform\Wingbox\Ribs\Guidelines for Ribs\Rib Axis Guideline`)
58        //                              temp5-> SetAttributeObject("Web Guideline",
          UpperStringers.GetItem(a).GetAttributeObject("Web Guideline"))
59        //                              if a>1{temp5-> SetAttributeObject("Rib",
          UpperStringerHoles.GetItem(UpperStringerHoles.Size()-1).GetAttributeObject("New
          Rib"))}
60        //                              else{temp5-> SetAttributeObject("Rib",Ribs.GetItem(i)
          )}
61        //                              temp5 -> SetAttributeObject("Upper Rib Edge",Ribs.
          GetItem(i).GetAttributeObject("Upper Rib Edge"))
62        //                              temp5 ->SetAttributeDimension("Stiffener Height" ,`
          Skin parameters\Stiffener Height`,"length")
63        //                              temp5 ->SetAttributeDimension("Stiffener Thickness"
          ,`Skin parameters\Stiffener Thickness`,"length")
64        //                              temp5 ->SetAttributeDimension("Tolerance" ,`Skin
          parameters\Spar Hole Tolerance`,"length")
65        //      EndModifyTemplate(temp5)}}}
66
67
68        //      b=1
69        //      C=0
70        //      for b while b<=LowerStringers.Size(){
71        //              Point4=LowerStringers.GetItem(b).GetAttributeObject("Connection to
          Aft Spar")
72        //              Point5=temp3.GetAttributeObject("Rib End Point")
73        //              Point6=LowerStringers.GetItem(b).GetAttributeObject("Stringer Start
          Point")
74        //              if Point4->coord(2)>Point5->coord(2)
75        //              {if Point6->coord(1) <Point5->coord(1)
76        //                      {temp6=CreateOrModifyTemplate("UDF_LoStrHo",GEOSET2,
          LowerStringerHoles,LowerStringerHoles.Size()+1)
77        //                              temp6->SetAttributeObject("Start Front Spar",`Wing
          Planform\Wingbox\Spar\guideline for wing spars\Start Front Spar`)
78        //                              temp6->SetAttributeObject("Rib Axis Guideline
          ",`Wing Planform\Wingbox\Ribs\Guidelines for Ribs\Rib Axis Guideline`)
79        //                              temp6-> SetAttributeObject("Web Guideline",
          LowerStringers.GetItem(b).GetAttributeObject("Web Guideline"))
80        //                              if b>1
81        //                              {temp6-> SetAttributeObject("Rib",LowerStringerHoles.
          GetItem(LowerStringerHoles.Size()-1).GetAttributeObject("New Rib"))}
82        //                              else
83        //                              {temp6-> SetAttributeObject("Rib",UpperStringerHoles.
          GetItem(UpperStringerHoles.Size()).GetAttributeObject("New Rib"))}
84        //                              temp6 -> SetAttributeObject("Lower Rib Edge",Ribs.
          GetItem(i).GetAttributeObject("Lower Rib Edge"))
85        //                              temp6 ->SetAttributeDimension("Stiffener Height" ,`
          Skin parameters\Stiffener Height`,"length")
86        //                              temp6 ->SetAttributeDimension("Stiffener Thickness"
          ,`Skin parameters\Stiffener Thickness`,"length")
```

```
87    //                              temp6 ->SetAttributeDimension("Tolerance" ,`Skin
          parameters\Spar Hole Tolerance`,"length")
88    //                         C=C+1
89    //        EndModifyTemplate(temp6)}}}
90
91
92    //        Notify("HELLO #",C)
93    //split(`Relations\Rib+Stiffener Generation\LowerStringerHoles`.GetItem(1).
          GetAttributeObject("New Rib")),`Relations\Rib+Stiffener Generation\
          LowerStringerHoles`.GetItem(1),TRUE)
94    //        Rb=new("Surface","Final_Rib"+i,GEOSET1)
95    //        Rb = LowerStringerHoles.GetItem(LowerStringerHoles.Size()).GetAttributeObject
          ("New Rib")
96    //        if i <= 1
97    //        {Assemble1=LowerStringerHoles.GetItem(LowerStringerHoles.Size()).
          GetAttributeObject("New Rib")}
98    //        else
99    //        {Assemble1=assemble(Assemble1, LowerStringerHoles.GetItem(LowerStringerHoles.
          Size()).GetAttributeObject("New Rib"))}
100
101
102
103       //creating the stiffeners per rib
104       j=1
105       c=1
106       Curve1=Ribs.GetItem(i).GetAttributeObject("Rib Line")
107       K =  length(Curve1)/`Rib Parameters\Rib Stiffener Pitch` //calculating the ratio
             needed for stiffener placement
108       // Determine the amount of stiffeners on a rib taking into account the stiffener
             flange
109       if `Rib Parameters\Rib Stiffener Pitch`*floor(K)+`Rib Parameters\Rib Stiffener Height
          `/2- length(Curve1)<=0
110           k=floor(K)
111       else {k=floor(K) -1}
112       //Notify ("There are # occurrences of Reference '#' under the root product",k)
113       for j while j<=k
114       {      temp4=CreateOrModifyTemplate("UDF_Stiffeners",GEOSET1,RibStiffeners,
          RibStiffeners.Size()+1)
115           temp4->SetAttributeObject("Rib Line",temp3.GetAttributeObject("Rib Line"))
116           temp4->SetAttributeObject("Rib Start Point",temp3.GetAttributeObject("Rib
             Start Point"))
117           temp4->SetAttributeObject("Upper Wing Surface",Surface1)
118           temp4->SetAttributeObject("Lower Wing Surface",Surface2)
119           temp4->SetAttributeObject("Start_Front_Spar",`1. Wing Planform\1.3. Wingbox
             \1.3.3. Spar\guideline for wing spars\Start Front Spar`)
120               temp4->SetAttributeObject("Rib Axis Guideline",`1. Wing Planform\1.3.
                   Wingbox\1.3.1. Ribs\Guidelines for Ribs\Rib Axis Guideline`)
121               R4=j/K
122           temp4->SetAttributeReal("Ratio",R4)
123           temp4 ->SetAttributeDimension("H_Stiffener" ,`Rib Parameters\Rib Stiffener
             Height`,"length")
124           //temp4 ->SetAttributeDimension("Upper Hole Tolerance",0mm,"length")
125           //temp4 ->SetAttributeDimension("Lower Hole Tolerance",0mm,"length")
126           //        Point7=temp4.GetAttributeObject("Start Point Stiffener")
127           //        c=1
128           //        for c while c<=C
129           //        {Point10 = UpperStringerHoles.GetItem(UpperStringerHoles.Size
             ()-c+1).GetAttributeObject("Hole Start Point")
130           //              if sqrt((Point7.coord(1)-Point10.coord(1))**2+(Point7
             .coord(2)-Point10.coord(2))**2)<=`Skin parameters\Stiffener Height`/2+`
             Rib Parameters\Stiffener Height`/2
131           //              {temp4 ->SetAttributeDimension("Upper Hole Tolerance"
             ,`Skin parameters\Stiffener Height`+`Skin parameters\Stiffener Thickness
             `,"Length")
132           //              temp4 ->SetAttributeDimension("Lower Hole Tolerance",`Skin
             parameters\Stiffener Height`+`Skin parameters\Stiffener Thickness`,"
             Length")}}
133       EndModifyTemplate(temp4)}
134 }
135
136
```

```
137  //Engine Rib generation
138  i=1
139  EngPos=`Engine Position\CG Engine Y % of span`*`Wingspan, b`/2
140  for i while i<=Ribs.Size()-1
141  {if Ribs.GetItem(i).GetAttributeObject("Rib Start Point").coord(2)>EngPos
142          {EngineRibPosition=i
143  break}}
144
145
146
147  // Engine rib Generation
148  GEOSET3=new("OpenBodyFeature","Engine Rib",`1. Wing Planform\1.3. Wingbox\1.3.1. Ribs`)
149  temp1=CreateOrModifyTemplate("UDF_EngineRib",GEOSET3,EngineRib,1)
150  temp1->SetAttributeObject("Upper Wing Surface",Surface1)
151  temp1->SetAttributeObject("Lower Wing Surface",Surface2)
152  temp1->SetAttributeObject("Rib14 Line",Ribs.GetItem(EngineRibPosition).GetAttributeObject("
        Rib Line"))
153  temp1->SetAttributeObject("Rib13",Ribs.GetItem(EngineRibPosition-1).GetAttributeObject("Rib")
        )
154  EndModifyTemplate(temp1)
155
156  //creating the stiffeners per rib
157  j=1
158  Curve1=temp1.GetAttributeObject("Rib Engine Guideline")
159  //calculating the ratio needed for stiffener placement
160  K =  length(Curve1)/`Rib Parameters\Rib Engine Stiffener Pitch`
161  // Determine the amount of stiffeners on a rib taking into account the stiffener flange
162  if `Rib Parameters\Rib Engine Stiffener Pitch`*floor(K)+`Rib Parameters\Rib Engine Stiffener
        Thickness`/2- length(Curve1)<=0
163          k=floor(K)
164  else {k=floor(K) -1}
165
166
167  //Notify ("There are # occurrences of Reference '#' under the root product",k)
168  for j while j<=k
169  {       temp4=CreateOrModifyTemplate("UDF_Stiffeners",GEOSET3,EngineRibStiffeners,
        EngineRibStiffeners.Size()+1)
170          temp4->SetAttributeObject("Rib Line",temp1.GetAttributeObject("Rib Engine Guideline")
            )
171          temp4->SetAttributeObject("Rib Start Point",temp1.GetAttributeObject("Rib Engine
            Start Point"))
172          temp4->SetAttributeObject("Upper Wing Surface",Surface1)
173          temp4->SetAttributeObject("Lower Wing Surface",Surface2)
174          temp4->SetAttributeObject("Start_Front_Spar",`1. Wing Planform\1.1. Original Wing
            Airfoils\1.1.2. P2.\P2.Axis Origin`)
175          temp4->SetAttributeObject("Rib Axis Guideline",`1. Wing Planform\1.1. Original Wing
            Airfoils\1.1.2. P2.\P2.Chord`)
176          R4=1-j/K
177          temp4->SetAttributeReal("Ratio",R4)
178          temp4 ->SetAttributeDimension("H_Stiffener" ,`Rib Parameters\Rib Engine Stiffener
            Height`,"length")
179          EndModifyTemplate(temp4)}
180
181
182
183
184
185  // Create Join for Ribs for FEM Mesh
186  InboardRibs=Ribs.GetItem(1).GetAttributeObject("RibV2")
187  OutboardRibs=Ribs.GetItem(Ribs.Size()).GetAttributeObject("RibV2")
188  EngineSupportRibs = Ribs.GetItem(EngineRibPosition).GetAttributeObject("RibV2")
189  i=2
190  for i while i<=Ribs.Size()-1
191  {if i==EngineRibPosition
192          {continue}
193          else if i>=EngineRibPosition-2 and i<=EngineRibPosition+1
194          {EngineSupportRibs=assemble(EngineSupportRibs,Ribs.GetItem(i).GetAttributeObject("
            RibV2"))}
195          else if pointoncurveRatio(Ribs.GetItem(i).GetAttributeObject("Rib Line"),Ribs.GetItem
            (i).GetAttributeObject("Rib Start Point"),1,True).coord(2)<`1. Wing Planform\1.1.
             Original Wing Airfoils\1.1.3. P3.Yehudi\P3.Axis System\Origin\Y`
```

```
196          {InboardRibs=assemble(InboardRibs,Ribs.GetItem(i).GetAttributeObject("RibV2"))}
197          else
198 {OutboardRibs=assemble(OutboardRibs,Ribs.GetItem(i).GetAttributeObject("RibV2"))}}
199
200 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Inboard Rib Assemble w/o Engine
        Support Ribs`=InboardRibs
201 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Outboard Rib Assemble w/o
        Engine Support Ribs`=OutboardRibs
202 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Engine Support Ribs`=
        EngineSupportRibs
203 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Engine Rib `=EngineRib.GetItem
        (1)
204
205
206
207 // Rib stiffeners
208 RibStiffenersJoin=RibStiffeners.GetItem(1)
209 i=2
210 for i while i<=RibStiffeners.Size()
211 {RibStiffenersJoin=assemble(RibStiffenersJoin,RibStiffeners.GetItem(i))}
212
213 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Rib Stiffener Assemble`=
        RibStiffenersJoin
214
215 // Engine Rib Stiffeners
216 EngineRibStiffenersJoin=EngineRibStiffeners.GetItem(1)
217 i=2
218 for i while i<=EngineRibStiffeners.Size()
219 {EngineRibStiffenersJoin=assemble(EngineRibStiffenersJoin,EngineRibStiffeners.GetItem(i))}
220 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Engine Rib Stiffener Assemble`=
        EngineRibStiffenersJoin
```

A.2. Stringers

```
1 let l,m,s,t,i (Integer)
2 let S,T,R1,R2 (Real)
3 let temp1,temp2 (Feature)
4 let StringerJoin, UpperStringersJoin,LowerStringersJoin,Surface1,Surface2(Surface)
5 let GEOSET1(OpenBodyFeature)
6
7 Surface1=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface\Upper Wing Surface`
8 Surface2=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface`
9
10
11
12 // upper Stringers stiffeners
13 l=1
14 S = length(`1. Wing Planform\1.3. Wingbox\1.3.1. Ribs\Inboard Wingbox closing`)/`Skin
        parameters\Skin Upper Stiffener Pitch`
15 // Determine the amount of stiffeners on a rib taking into account the stiffener flange
16 if `Skin parameters\Skin Upper Stiffener Pitch`*floor(S)+`Skin parameters\Skin Upper
        Stiffener Height`/2-length(`1. Wing Planform\1.3. Wingbox\1.3.1. Ribs\Inboard Wingbox
        closing`)<=0
17         s=floor(S)
18 else {s=floor(S)-1 }
19
20 //Notify ("There are # occurrences of Reference '#' under the root product",s)
21
22 for l while l<=s
23 {        temp1=CreateOrModifyTemplate("UDF_UpStringers",`1. Wing Planform\1.3. Wingbox\1.3.2.
        Stringers\Upper Surface Stringers`,UpStringers,l)
24         temp1->SetAttributeObject("Start Front Spar",`1. Wing Planform\1.3. Wingbox\1.3.3.
            Spar\guideline for wing spars\Start Front Spar`)
25             temp1->SetAttributeObject("Upper Wing Surface",Surface1)
26         temp1->SetAttributeObject("Front spar",`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\
            guideline for wing spars\Front spar position`)
27             temp1->SetAttributeObject("Inboard Wingbox closing",`1. Wing Planform\1.3.
                Wingbox\1.3.1. Ribs\Inboard Wingbox closing`)
28         temp1->SetAttributeObject("Aft Stringer Connection",`1. Wing Planform\1.3. Wingbox
            \1.3.2. Stringers\Aft Stringer Connection`)
```

```
29          R1=l/S
30          temp1->SetAttributeReal("Ratio",R1)
31          temp1 ->SetAttributeDimension("Stiffener Height" ,`Skin parameters\Skin Upper
                Stiffener Height`,"length")
32          EndModifyTemplate(temp1)
33 }
34
35 // Lower Stringers stiffeners
36 m=1
37 T = length(`1. Wing Planform\1.3. Wingbox\1.3.1. Ribs\Inboard Wingbox closing`)/`Skin
      parameters\Skin Lower Stiffener Pitch`
38 // Determine the amount of stiffeners on a rib taking into account the stiffener flange
39 if `Skin parameters\Skin Lower Stiffener Pitch`*floor(T)+`Skin parameters\Skin Lower
      Stiffener Height`/2-length(`1. Wing Planform\1.3. Wingbox\1.3.1. Ribs\Inboard Wingbox
      closing`)<=0
40          t=floor(T)
41 else {t=floor(T)-1 }
42
43 //Notify ("There are # occurrences of Reference '#' under the root product",k)
44
45 for m while m<=t
46 {        temp2=CreateOrModifyTemplate("UDF_LoStringers",`1. Wing Planform\1.3. Wingbox\1.3.2.
      Stringers\Lower Surface Stringers`,LoStringers,m)
47          temp2->SetAttributeObject("Start Front Spar",`1. Wing Planform\1.3. Wingbox\1.3.3.
                Spar\guideline for wing spars\Start Front Spar`)
48              temp2->SetAttributeObject("Lower Wing Surface",Surface2)
49          temp2->SetAttributeObject("Front spar",`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\
                guideline for wing spars\Front spar position`)
50              temp2->SetAttributeObject("Inboard Wingbox closing",`1. Wing Planform\1.3.
                    Wingbox\1.3.1. Ribs\Inboard Wingbox closing`)
51          temp2->SetAttributeObject("Aft Stringer Connection",`1. Wing Planform\1.3. Wingbox
                \1.3.2. Stringers\Aft Stringer Connection`)
52          R2=m/T
53          temp2->SetAttributeReal("Ratio",R2)
54          temp2 ->SetAttributeDimension("Stiffener Height" ,`Skin parameters\Skin Lower
                Stiffener Height`,"length")
55          EndModifyTemplate(temp2)
56 }
57
58 //Stringers
59 UpperStringersJoin=UpStringers.GetItem(1)
60 i=2
61 for i while i<=UpStringers.Size()
62 {UpperStringersJoin=assemble(UpperStringersJoin,UpStringers.GetItem(i))}
63
64 LowerStringersJoin=LoStringers.GetItem(1)
65 i=2
66 for i while i<=LoStringers.Size()
67 {LowerStringersJoin=assemble(LowerStringersJoin,LoStringers.GetItem(i))}
68
69 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Upper Stringers Assemble`=
      UpperStringersJoin
70 `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Lower Stringers Assemble`=
      LowerStringersJoin
```

A.3. Spar Stiffeners

```
1 let i,k (Integer)
2 let P0(Point)
3 let P(Curve)
4 let Z1,Z2(LENGTH)
5 let V,K (Real)
6 let temp(Feature)
7 let GEOSET1,GEOSET2 (OpenBodyFeature)
8 let Join1, SparStiffenerJoin,Surface1,Surface2(Surface)
9
10 //Wing Surface Elements
11 if MorphingSetting==0
12 {Surface1=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface\Upper Wing Surface`
13 Surface2=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface`}
```

```
14  else if MorphingSetting==1
15  {Surface1=`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1. Lower Surface\M1.
        Upper Surface`
16  Surface2=`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1. Lower Surface`}
17  else if MorphingSetting==2
18  {Surface1=`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2. Lower Surface\M2.
        Upper Surface`
19  Surface2=`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2. Lower Surface`}
20
21  Z2= `Spar Parameters\Spar Stiffener Height`
22
23  // Front spar stiffeners
24  i=1
25  Z1= length(`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\guideline for wing spars\Front spar
        position`)
26         K = Z1/`Spar Parameters\Spar Stiffener Pitch`
27  k=floor(K)
28
29  // Determine the amount of stiffeners on a rib taking into account the stiffener flange
30  if `Spar Parameters\Spar Stiffener Pitch`*k+`Spar Parameters\Spar Stiffener Height`/2-Z1>=0
31  {k=k-1}
32
33  if `Spar Parameters\Spar Stiffener Pitch`*k+(`Spar Parameters\Spar Stiffener Height`+`Spar
        Parameters\Spar Stiffener Thickness`)*sin(`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\
        guideline for wing spars\Spar Parameters\LE Spar Sweep`)-Z1>=0
34  {k=k-1}
35
36  //Notify("Hello",`Spar Parameters\Stiffener Pitch`*floor(K)+`Spar Parameters\Stiffener Height
        `/2-Z1)
37  //Notify("Hello",Z1)
38  //Notify ("There are # occurrences of Reference '#' under the root product",k)
39
40  for i while i<=k
41  {       GEOSET1=new("OpenBodyFeature","Front Spar Stiffeners",`1. Wing Planform\1.3. Wingbox
        \1.3.3. Spar`)
42         temp=CreateOrModifyTemplate("UDF_SparStiffeners",GEOSET1,FrontSparStiffeners,i)
43         temp->SetAttributeObject("Aft spar position",`1. Wing Planform\1.3. Wingbox\1.3.3.
            Spar\guideline for wing spars\Front spar position`)
44             temp->SetAttributeObject("Start Aft Spar",`1. Wing Planform\1.3. Wingbox
                \1.3.3. Spar\guideline for wing spars\Start Front Spar`)
45             temp->SetAttributeObject("Aft Spar",`1. Wing Planform\1.3. Wingbox\1.3.3.
                Spar\guideline for wing spars\Front Spar Guide`)
46             temp->SetAttributeObject("Start Front Spar",`1. Wing Planform\1.3. Wingbox
                \1.3.3. Spar\guideline for wing spars\Start Front Spar`)
47             temp->SetAttributeObject("Guideline Aft Spar Stiffeners",`1. Wing Planform
                \1.3. Wingbox\1.3.3. Spar\guideline for wing spars\Guideline Front Spar
                stiffeners`)
48             temp->SetAttributeObject("Upper Wing Surface",Surface1)
49         temp->SetAttributeObject("Lower Wing Surface",Surface2)
50         V=i/K
51         temp->SetAttributeReal("Ratio",V)
52         temp ->SetAttributeDimension("H_Stiffener",`Spar Parameters\Spar Stiffener Height`,'
            length')
53         EndModifyTemplate(temp)
54  }
55
56  //Aft Spar Stiffeners
57
58  i=1
59  Z1= length(`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\guideline for wing spars\Aft spar
        position`)
60         K =  Z1/`Spar Parameters\Spar Stiffener Pitch`
61  k=floor(K)
62
63  // Determine the amount of stiffeners on a rib taking into account the stiffener flange
64  if `Spar Parameters\Spar Stiffener Pitch`*k+`Spar Parameters\Spar Stiffener Height`-Z1>=0
65  {k=k-1}
66
67  if `Spar Parameters\Spar Stiffener Pitch`*k+(`Spar Parameters\Spar Stiffener Height`+`Spar
        Parameters\Spar Stiffener Thickness`)*sin(`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\
        guideline for wing spars\Spar Parameters\TE Spar Sweep`)-Z1>=0
```

```
68  {k=k-1}
69
70  //Notify ("There are # occurrences of Reference '#' under the root product",(`Spar Parameters
        \Stiffener Height`+`Spar Parameters\Stiffener Thickness`)*sin(`Wing Planform\Wingbox\Spar
        \guideline for wing spars\Spar Parameters\TE Spar Sweep`/180*PI))
71
72  for i while i<=k
73  {       GEOSET2=new("OpenBodyFeature","Aft Spar Stiffeners",`1. Wing Planform\1.3. Wingbox
        \1.3.3. Spar`)
74          temp=CreateOrModifyTemplate("UDF_SparStiffeners",GEOSET2,AftSparStiffeners,i)
75          temp->SetAttributeObject("Aft spar position",`1. Wing Planform\1.3. Wingbox\1.3.3.
                Spar\guideline for wing spars\Aft spar position`)
76              temp->SetAttributeObject("Start Aft Spar",`1. Wing Planform\1.3. Wingbox
                    \1.3.3. Spar\guideline for wing spars\Start Aft Spar`)
77              temp->SetAttributeObject("Aft Spar",`1. Wing Planform\1.3. Wingbox\1.3.3.
                    Spar\guideline for wing spars\Aft Spar`)
78              temp->SetAttributeObject("Start Front Spar",`1. Wing Planform\1.3. Wingbox
                    \1.3.3. Spar\guideline for wing spars\Start Front Spar`)
79              temp->SetAttributeObject("Guideline Aft Spar Stiffeners",`1. Wing Planform
                    \1.3. Wingbox\1.3.3. Spar\guideline for wing spars\Guideline Aft Spar
                    Stiffeners`)
80              temp->SetAttributeObject("Upper Wing Surface",Surface2)
81          temp->SetAttributeObject("Lower Wing Surface",Surface1)
82          V=i/K
83          temp->SetAttributeReal("Ratio",V)
84          temp ->SetAttributeDimension("H_Stiffener",`Spar Parameters\Spar Stiffener Height`,'
                length')
85          EndModifyTemplate(temp)
86  }
87
88  SparStiffenerJoin=FrontSparStiffeners.GetItem(1)
89  i=2
90  for i while i<=FrontSparStiffeners.Size()
91  {SparStiffenerJoin=assemble(SparStiffenerJoin,FrontSparStiffeners.GetItem(i))}
92  i=1
93  for i while i<=AftSparStiffeners.Size()
94  {SparStiffenerJoin=assemble(SparStiffenerJoin,AftSparStiffeners.GetItem(i))}
95  `5. ForFEMSimulation\5.1. Wingbox Elements\5.1.1. Uncorrected\Spar Stiffener Assemble`=
        SparStiffenerJoin
```

A.4. FEM Preparation

```
1   let i,j,l,m,n,EngRib,Counter (Integer)
2   let P(Real)
3   let Join1, Join2, Surface1,Surface2, LEUp,LELo,TEUp,TELo, SparJoin (surface)
4   let RibStart, RibEnd, SurfaceUp, SurfaceLo, WingboxUp,WingboxLo, FuelTankAss, Spars (surface)
5   let FuelTankVol (volume)
6   let RibList(list)
7   let GEOSET2, GEOSET3 (OpenBodyFeature)
8
9   GEOSET2 = new("OpenBodyFeature","5.2.1. Uncorrected",`5. ForFEMSimulation\5.2. Wing Surface
        Elements`)
10  GEOSET3 = new("OpenBodyFeature","5.3.1. Fuel Tank Volumes",`5. ForFEMSimulation\5.3.
        Guidelines Wing Surface`)
11
12  //create a upper and lower surface.
13  //Wing Surface Elements
14  Join1 = new("Surface", "Uncorrected Trailing Clossing " , GEOSET2)
15  Join2 = new("Surface", "Uncorrected Tip Close Off " , GEOSET2)
16  if MorphingSetting==0
17  {Surface1=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface\Upper Wing Surface`
18          Surface2=`1. Wing Planform\1.3. Wingbox\Lower Wing Surface`
19          Join2=`1. Wing Planform\1.2. Wing Surface\1.2.1. Original Wing Surface\Tip Close Off`
20  Join1=assemble(`1. Wing Planform\1.2. Wing Surface\1.2.1. Original Wing Surface\Inboard TE
        Fill`,`1. Wing Planform\1.2. Wing Surface\1.2.1. Original Wing Surface\Outboard TE Fill`)
        }
21  else if MorphingSetting==1
22  {Surface1=`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1. Lower Surface\M1.
        Upper Surface`
23          Surface2=`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1. Lower Surface
```

```
                    `
24          Join2=`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1. Tip Close Off`
25          Join1=assemble(`1. Wing Planform\1.2. Wing Surface\1.2.1. Original Wing Surface\
                Inboard TE Fill`,`1. Wing Planform\1.2. Wing Surface\1.2.2. Morphing Wing 1\M1.
                Outboard TE Fill`)
26 }
27 else if MorphingSetting==2
28 {Surface1=`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2. Lower Surface\M2.
    Upper Surface`
29          Surface2=`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2. Lower Surface
                `
30          Join2=`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2. Tip Close Off`
31          Join1=assemble(`1. Wing Planform\1.2. Wing Surface\1.2.1. Original Wing Surface\
                Inboard TE Fill`,`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2.
                Outboard TE Fill 1`,`1. Wing Planform\1.2. Wing Surface\1.2.3. Morphing Wing 2\M2
                . Outboard TE Fill 2`)
32 }
33
34 // Leading edge and Trailing Edge Surfaces
35 LEUp=split(Surface1,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Front Spar`,False)
36 LELo=split(Surface2,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Front Spar`,False)
37
38 WingboxUp=split(split(Surface1,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Front Spar`,True)
      ,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Aft Spar`,False)
39 WingboxLo=split(split(Surface2,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Front Spar`,True)
      ,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Aft Spar`,False)
40
41 TEUp=split(Surface1,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Aft Spar`,True)
42 TELo=split(Surface2,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Aft Spar`,True)
43
44 SparJoin=assemble(`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Front Spar`,`1. Wing Planform
      \1.3. Wingbox\1.3.3. Spar\Aft Spar`)
45
46 // Find the ribs to enclose the fuel tanks.
47 j=1
48 l=1
49 P=0.26
50 Counter=0
51 for j while j<=Ribs.Size()-1
52 {if P<0.8
53          {if pointoncurveRatio(Ribs.GetItem(j).GetAttributeObject("Rib Line"),Ribs.GetItem(j).
                GetAttributeObject("Rib Start Point"),0.5,True).coord(2)>P*`Wingspan, b`/2
54                  {RibList.AddItem(j,l)
55                          P=P+0.1
56          l=l+1}}
57          if Counter==0
58          {if Ribs.GetItem(j).GetAttributeObject("Rib Start Point").coord(2)>`Engine Position\
                CG Engine Y % of span`*`Wingspan, b`/2
59                  {EngRib=j
60          Counter=1}}
61 }
62
63
64 //Surface and Volume of Fuel Tanks
65 l=1
66 m=RibList.GetItem(l)
67
68 Join1=new("Surface", "LE Upper Surface "+(l) , GEOSET2)
69 Join1= split(LEUp,Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),True)
70          Join1=new("Surface", "LE Lower Surface "+(l) , GEOSET2)
71 Join1=split(LELo,Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),True)
72
73 SurfaceUp=new("Surface", "Wingbox Upper Surface "+(l) , GEOSET2)
74 SurfaceUp= split(WingboxUp,Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),True)
75          SurfaceLo=new("Surface", "Wingbox Lower Surface "+(l) , GEOSET2)
76 SurfaceLo=split(WingboxLo,Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),True)
77
78 Join1=new("Surface", "TE Upper Surface "+(l) , GEOSET2)
79 Join1= split(TEUp,Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),True)
80          Join1=new("Surface", "TE Lower Surface "+(l) , GEOSET2)
```

```
81  Join1=split(TELo,Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),True)
82
83  RibStart = split(split(`1. Wing Planform\1.2. Wing Surface\1.2.1. Original Wing Surface\Body
        Close Off`,`1. Wing Planform\1.3. Wingbox\1.3.3. Spar\Front Spar`,True),`1. Wing Planform
        \1.3. Wingbox\1.3.3. Spar\Aft Spar`,False)
84  RibEnd = Ribs.GetItem(m).GetAttributeObject("Rib")
85  Spars=split(SparJoin, RibEnd,True)
86
87  FuelTankAss=new("Surface","Fuel Tank"+(l),GEOSET3)
88  FuelTankAss = assemble(RibStart, RibEnd, SurfaceUp,SurfaceLo, Spars)
89
90
91  for l while l<=RibList.Size()-1
92  {m=RibList.GetItem(l)
93          n=RibList.GetItem(l+1)
94
95          Join1=new("Surface", "LE Upper Surface "+(l+1) , GEOSET2)
96          Join1= split(split(LEUp,Ribs.GetItem(n).GetAttributeObject("Extrude for Split"),True)
                  , Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),False)
97              Join1=new("Surface", "LE Lower Surface "+(l+1) , GEOSET2)
98          Join1=split(split(LELo,Ribs.GetItem(n).GetAttributeObject("Extrude for Split"),True),
                  Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),False)
99
100         SurfaceUp=new("Surface", "Wingbox Upper Surface "+(l+1) , GEOSET2)
101         SurfaceUp= split(split(WingboxUp,Ribs.GetItem(n).GetAttributeObject("Extrude for
                  Split"),True), Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),False)
102             SurfaceLo=new("Surface", "Wingbox Lower Surface "+(l+1) , GEOSET2)
103         SurfaceLo=split(split(WingboxLo,Ribs.GetItem(n).GetAttributeObject("Extrude for Split
                  "),True),Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),False)
104
105         Join1=new("Surface", "TE Upper Surface "+(l+1) , GEOSET2)
106         Join1= split(split(TEUp,Ribs.GetItem(n).GetAttributeObject("Extrude for Split"),True)
                  , Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),False)
107             Join1=new("Surface", "TE Lower Surface "+(l+1) , GEOSET2)
108         Join1=split(split(TELo,Ribs.GetItem(n).GetAttributeObject("Extrude for Split"),True),
                  Ribs.GetItem(m).GetAttributeObject("Extrude for Split"),False)
109
110         RibStart = Ribs.GetItem(m).GetAttributeObject("Rib")
111         RibEnd = Ribs.GetItem(n).GetAttributeObject("Rib")
112         Spars=split(split(SparJoin, RibEnd,True),RibStart,False)
113
114         FuelTankAss=new("Surface","Fuel Tank"+(l+1),GEOSET3)
115         FuelTankAss = assemble(RibStart, RibEnd, SurfaceUp,SurfaceLo, Spars)
116 }
117
118 Join1=new("Surface","LE Upper Surface "+(l+1),GEOSET2)
119 Join1=split(LEUp,Ribs.GetItem(RibList.GetItem(RibList.Size())).GetAttributeObject("Extrude
        for Split"),False)
120         Join1=new("Surface","LE Lower Surface "+(l+1),GEOSET2)
121 Join1=split(LELo,Ribs.GetItem(RibList.GetItem(RibList.Size())).GetAttributeObject("Extrude
        for Split"),False)
122
123 Join1=new("Surface","Wingbox Upper Surface "+(l+1),GEOSET2)
124 Join1=split(WingboxUp,Ribs.GetItem(RibList.GetItem(RibList.Size())).GetAttributeObject("
        Extrude for Split"),False)
125         Join1=new("Surface","Wingbox Lower Surface "+(l+1),GEOSET2)
126 Join1=split(WingboxLo,Ribs.GetItem(RibList.GetItem(RibList.Size())).GetAttributeObject("
        Extrude for Split"),False)
127
128 Join1=new("Surface","TE Upper Surface "+(l+1),GEOSET2)
129 Join1=split(TEUp,Ribs.GetItem(RibList.GetItem(RibList.Size())).GetAttributeObject("Extrude
        for Split"),False)
130         Join1=new("Surface","TE Lower Surface "+(l+1),GEOSET2)
131 Join1=split(TELo,Ribs.GetItem(RibList.GetItem(RibList.Size())).GetAttributeObject("Extrude
        for Split"),False)
```