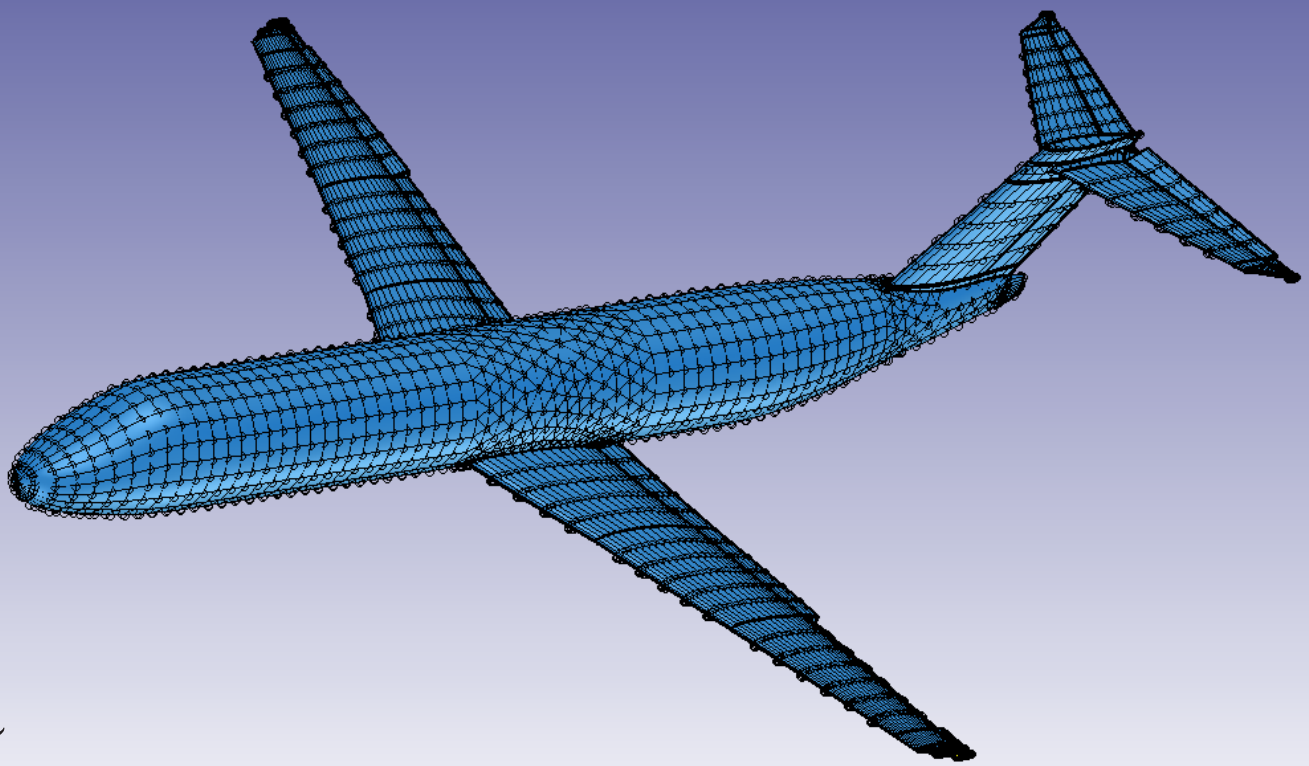
Parametric modelling for determining aircraft stability & control derivatives

J.H. Wei

Technische Universiteit Delft



TU Delft

Parametric modelling for determining aircraft stability & control derivatives

by

J.H. Wei

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on October 24, 2016.

Student number: 4006828
Project duration: January 10, 2015 – October 24, 2016

Thesis committee:
Chairman: Prof. dr. ir. L.L.M. Veldhuis
Daily supervisor: Dr. ir. M. Voskuijl
Supervisor: Ir. R.E.C. van Dijk
Dr. ir. W.A.A.M. Bierbooms

Thesis registration number = 090#16#MT#FPP
An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Summary

This thesis is part of a project that focuses on developing an optimisation framework for dynamically scaled flight testing. The optimisation framework must design the scale model and the flight test such that the performance of the scale model is representative to the full-scale aircraft. The similarity in the performances between the scale model and the full-scale aircraft is achieved by altering the geometry, the mass distribution and the structure of the model, which are related to the aerodynamic, flight dynamic and structures disciplines respectively. This thesis contributes to the aerodynamics optimisation framework by focusing on the development of a parametric model that is capable of deriving the stability and control derivatives.

The research goals of the thesis are mostly related to the construction of a parametric aircraft model for the aerodynamic solvers that are based on a first order panel method like VSAERO. The most important goal is how to model the trailing edge moveables for a first order panel method for deriving the control derivatives. This thesis will investigate three options for creating the moveables. The first option ('normal rotation') is a mathematical operation in VSAERO that rotates the normal vectors of the body panels that represents the moveable. The second option ('transition surface') is a wing with a moveable model that has transition surfaces between the wing and the moveable in the spanwise direction. The third option ('gap') is the same model as method two, but instead of transition surfaces, there is a gap between the wing and the moveable. The second research goal is to compute the stability & control derivatives of the parametric model and investigate the accuracy of these derivatives.

Out of the three modelling options, the wing with a moveable model that has a gap between the sides of the moveable and the wing was useless in VSAERO. The model was unstable due to the presence of the gap. The gap creates a very low pressure locally, which accelerates the surrounding airflow to a ridiculously high value. The difference between the 'normal rotation' and the 'transition surface' models is the location of the moveable suction peak. The suction peak of the 'normal rotation' model is located on the wing in front of the start of the moveable, while the suction peak of the 'transition surface' is located at the start or slightly behind the start of the moveable. The effect of different locations of the suction peaks is that a suction peak on the wing will generate less induced drag than a suction peak located on the deflected moveable.

The results of stability & control derivatives for the 'normal rotation' and the 'transition surface' modelling options were quite acceptable with the static derivatives as the most accurate derivatives with an average error of seven percent. The control derivatives, on the other hand, was the least accurate with a mean error of 40 percent. The overall performance of the two moveable modelling options is that they were performing equally well when only the accuracy of the derivatives was observed. The 'transition surface' modelling option was more accurate in the prediction of lateral stability derivatives and the longitudinal control derivatives than the 'normal rotation' option, while the 'normal rotation' option was more accurate in predicting the longitudinal stability derivatives and the lateral control derivatives. But the deflection of the rudder produces side force, yaw moment and roll moment coefficients that are less accurate for the 'normal rotation' option then for the 'transition surface' option.

The 'transition surface' modelling option is the better option for modelling the moveable because it produced more accurate aerodynamic results than the 'normal rotation' modelling option. Both moveable modelling options have its strengths and weakness when concerning the prediction of the stability and control derivatives, and neither of the two was considerably better. Thus based on the accuracy of the aerodynamic characteristics, the 'transition surface' modelling option is the best option for modelling the moveables of a first order panel method solvers like VSAERO.

Acknowledgements

First of all, I want to thank my daily supervisor Mark Voskuijl for guiding my research and I appreciate the advises he gave me for my thesis. Also thanks to Durk Steenhuizen who was one of my initial daily supervisors, but unfortunately he does not work here anymore. Special thanks go to Reinier van Dijk, who has supervised me around the development of the parametric model for the new Multi-Model Generator on the ParaPy platform.

J.H. Wei
Delft, October 2016

Contents

Nomenclature

Roman Symbols

a	Linear acceleration [m/s^2]
b	Wingspan [m]
c	Chord length [m]
C_D	Drag coefficient [-]
C_L	Lift coefficient [-]
C_l	Roll moment coefficient [-]
C_m	Pitch moment coefficient [-]
C_n	Yaw moment coefficient around the Z-axis [-]
C_p	Pressure coefficient [-]
C_X	Force coefficient in direction of X-axis [-]
C_Y	Force coefficient in direction of Y-axis [-]
C_Z	Force coefficient in direction of Z-axis [-]
E	Young's modulus [N/m^2]
Eu	Euler number [-]
F	Force [N]
Fr	Froude number [-]
G	Shear modulus of elasticity [N/m^2]
g	Gravitational acceleration [m/s^2]
I	Mass moment of inertia [kg·m^2]
I'	Area moment of inertia [m^4]
I_p	Incidence angle engine trust [deg]
J'	Polar moment of inertia [m^4]
k	Radius of gyration [m]
L	Lift force [N]
l	Length [m]
L	Roll moment [Nm]
l_h	Length from center of gravity to aerodynamic center of horizontal tail plane in direction of X-axis [m]
M	Mach number [-]
m	Mass [kg]

M	Pitch moment [Nm]
N	Number of cycles [-]
n	Scale factor [-]
N	Yaw moment [Nm]
N_h	Normal force of horizontal tail plane [N]
P	Period [N/m^2]
p	Pressure [N/m^2]
q	Dynamic pressure [N/m^2]
R	Gas constant [-] or radius [m]
Re	Reynolds number [-]
S	Wing area [m^2]
s	Displacement distance [m]
S_h	Horizontal tail plane area [m^2]
Str	Strouhal number [-]
T	Temperature [K]
t	Time [s]
T_c	[-] Engine thrust coefficient [-]
u	Velocity in the direction of the x-axis [m/s]
V	Velocity [m/s]
V_S	Speed of sound [m/s]
W	Weight [N]
X	Aerodynamic force along the X-axis [N]
Y	Aerodynamic force along the Y-axis [N]
Z	Aerodynamic force along the Z-axis [N]

Greek Symbols

α	Angle of attack [deg]
β	Angle of sideslip [deg]
γ	Adiabatic constant or isentropic expansion factor [-]
δ	Deflection angle [deg]
δ_a	Aileron deflection [deg]
δ_e	Elevator deflection [deg]
δ_r	Rudder deflection [deg]
δ_t	Trim tab deflection [deg]
ζ	Damping ratio [-]

η	Response damping [rad/s]
λ	Eigenvalue [-]
ν	Kinematic viscosity [m^2/s]
ρ	Mass density of air [kg/m^3]
φ	Bank angle [deg]
Ω	Angular rate [rad/s]
$\dot{\Omega}$	Angular acceleration [rad/s^2]
ω	Frequency of oscillation or damped natural frequency [rad/s]

Subscripts

0	Initial condition
2	... to doubling amplitude
$\frac{1}{2}$	... to halving amplitude
a	Full-scale aircraft
m	Scale model
n	Natural
p	Derivatives w.r.t. to the angular velocity about X-axis [s/rad]
q	Derivatives w.r.t. to the angular velocity about Y-axis [s/rad]
r	Derivatives w.r.t. to the angular velocity about Z-axis [s/rad]
r	Ratio between scale model and full-scale aircraft w.r.t. to the parameter
u	Derivatives w.r.t. to the velocity component along the X-axis [s/m]
α	Derivatives w.r.t. to angle of attack [1/rad]
α	Derivatives w.r.t. to angle of sideslip [1/rad]
δ_a	Derivatives w.r.t. to aileron deflection [1/rad]
δ_e	Derivatives w.r.t. to elevator deflection [1/rad]
δ_r	Derivatives w.r.t. to rudder deflection [1/rad]
δ_t	Derivatives w.r.t. to trim tab deflection [1/rad]

Abbreviations

CST	Class Shape Transformation
KBE	Knowledge Based Engineering
MAC	Mean Aerodynamic Chord [m]
MMG	Multi-Model Generator
$NVPI$	Number of viscous potential iterations
$NWIT$	Number of wake iterations
$SOLRES$	Solution Residual
UML	Unified Modelling Language

1

Introduction

New aircraft configurations like the blended wing body aircraft are predicted to have lower fuel consumption than traditional aircraft configurations by reducing the drag of the aircraft. The blended wing body, for example, eliminates the fuselage and thus a large portion of a non-lifting component of the aircraft.

Numerous design studies have been carried out for analysing the design of unconventional aircraft configurations. These design studies focus on three key characteristics of the aircraft, which are aerodynamic performance, flight performance and the handling qualities. Most design studies performed on the unconventional configurations are 'paper' studies that are being supported by computer aided analysis. Validating these studies require actual flight testing and a full-scale flight test is infeasible due to its high costs and development time. A potential solution to this problem is dynamically scaled flight testing.

Dynamically scaled flight testing is a flight test with a dynamically scaled model to simulate the flight characteristics of the full-scale aircraft. Dynamically scaled flight testing can significantly reduce the test preparation, test time, costs and risks.

Within the Flight Performance and Propulsion group, a project has been started to develop an optimisation framework for designing and predicting the flight mechanics of a dynamically scaled model. The framework analyses the behaviour of a dynamically scaled model in response to atmospheric disturbances and/or pilot inputs. The primary usage of this framework will be the optimisation of the scale model or the flight conditions of the scaled model to minimise the difference between the flight behaviour of the scaled model and the full sized aircraft. The framework will be especially useful for unconventional designs, where the flight response is mostly unknown.

The thesis focuses on the generation of the parametric aircraft model that will be used by the aerodynamic optimisation framework. The questions that arise are: How do you model the aircraft and its control surfaces such that it can be used to determine the flight mechanics? How do you link the model to an first order panel method tool like VSAERO? The subject of this thesis is thus to develop a parametric aircraft model for determining the stability and control derivatives.

The report is structured as follows. The background information about dynamically scaled flight testing and thesis goals are presented in chapter 2. Chapter 3 presents the parametric aircraft tool for modelling the aircraft. Chapter 4 explains the different methods for modelling the moveable/control surfaces. In chapter 5, the coupling between the aerodynamic tool and the model, will be explained. The validation of the aerodynamic results of the model will be presented in chapter 6. Chapter 7 contains the data about the stability and control derivatives and validate these derivatives. The conclusion and recommendations will form the last chapter of this thesis.

2

Context and Thesis Goals

Aircraft scale models have played a significant role in the development of aircraft throughout the history of human flight. Model testing is a crucial tool for understanding the aerodynamic flow properties and the flight mechanics of aircraft especially for unconventional and novel designs like the blended wing body aircraft. Aircraft model testing can either be static testing or dynamic testing. The goal of a static test is to analyse the aerodynamic properties of the scale model, while the focus of a dynamic test is to observe the flight behaviour of the scale model.

Developing and certifying a new aircraft requires a lot of full-scale flight test to predict the flight characteristics of the new aircraft. The potential of scaled model research is to reduce the amount of full-scale testing. Dynamically scaled model flight research, in particular, reduces the number of required flight tests of the full-scale aircraft, which in turn increases the safety and significantly reduces the risks and the costs in the development process. Also, the efficiency of the design process is improved since model testing can be commenced earlier in the design process than full-scale tests.

2.1. Concept of Scale Model Testing

The basic concept behind scale model testing is similitude in which the test data of the model can be used to predict the behaviour to the full-scale aircraft. The main issue behind model testing is how to ensure that the model results apply to the real life situation of the aircraft. So which flow conditions and model properties are required for model testing to achieve similitude?

The similarity between the model and prototype can be divided into two categories. The first category is model similitude and is related to the geometry of the model and its properties such as its moments of inertia. The second category is flow similitude. This category is related to the properties of the flow and specifies the required testing conditions for the model to achieve flow similarity.

2.1.1. Model Similitude

Model similitude is achieved by scaling or altering the model itself to ensure that the models characteristic can be correlated to the characteristics of the full-scale counterpart. This concept provides some scaling conditions for the model itself so that the model exhibits the same dynamic behaviour as the full-scale aircraft when it is loaded with the same scaled loads. Model similitude encompasses three scaling principles, which are geometric, dynamic and elastic scaling.

Geometric Scaling

Geometric scaling is the simplest scaling principle. It requires that the model and the full-scale aircraft to be scaled such that they have a similar geometry. Similarity in geometry implies that the ratios between corresponding lengths of the model and prototype are the same. The requirement for geometric scaling is illustrated in equation 2.1.

$$\frac{c_m}{c_a} = \frac{b_m}{b_a} = n \tag{2.1}$$

Dynamic Scaling

"Dynamically scaled free-flight models are not only geometrically scaled replicas; they are specially designed to ensure motion similitude between the subscale model and the full-scale subject. The model and the aircraft are referred to as dynamically similar, when the geometrically similar model reacts to external forces and moves in such a manner that the relative positions of its components are geometrically similar to those of a full-scale airplane after a proportional period of time." as defined by Chambers [1].

In short, a dynamically scaled model will have a similar flight path and angular displacement as the full-scale aircraft. The difference resulting from dynamic scaling is the time duration for each particular movement. Evaluating the test results requires a mathematical factor (equation 2.12) to compensate for the different time scale ratio.

Besides the geometric requirements, dynamic scaling gives additional requirements for achieving dynamic similitude. These requirements set the required mass and mass moment of inertia of the scale model. The requirements are the relative density factor and the relative mass moment of inertia [2] of equations 2.2 and 2.3 respectively [3].

$$\left(\frac{m}{\rho l^3}\right)_m = \left(\frac{m}{\rho l^3}\right)_a \tag{2.2}$$

$$\left(\frac{I}{\rho l^5}\right)_m = \left(\frac{I}{\rho l^5}\right)_a \tag{2.3}$$

The mass and mass moment of inertia ratios between the model and the aircraft can be derived using the above equation and is shown in equation 2.4 and 2.5 [3].

$$\frac{m_m}{m_a} = \frac{\rho_m}{\rho_a} n^3 \tag{2.4}$$

$$\frac{I_m}{I_a} = \frac{\rho_m}{\rho_a} n^5 \tag{2.5}$$

Equation 2.4 and 2.5 shows that both the mass and the mass moment of inertia ratios are dependent on the scale factor and the air density ratio. For example, a 1/10th scale model flying at the same altitude as the full-scale aircraft needs to have a mass that is 1/1000th of the full-scale aircraft, while the mass moment of inertia needs to be 1/100000th of the full-scale aircraft. Thus a dynamically scaled model will, therefore, require a mass and a mass moment of inertia that is smaller than the full-scale aircraft. The mass and the mass and mass moment of inertia of the scale model is also affected by the test altitude of the scale model and the flight altitude of the aircraft. Lowering the test altitude of the scale model below the flight altitude of the aircraft will increase the required mass and mass moment of inertia of the scale model.

Elastic Scaling

Elastic scaling is used for matching the elastic structural properties such as aeroelastic bending and torsion modes or for matching the flutter properties between the scale model and the full-scale aircraft [2]. Elastic scaling is only needed if the aeroelastic interaction between the fluid flow and elastic body has a significant impact on the aerodynamic performance and hence on the flight mechanics.

The similitude conditions for achieving aeroelastic similarity between the scale model and the aircraft are the aeroelastic bending and the aeroelastic torsion equations given below respectively [3].

$$\left(\frac{EI'}{\rho V^2 l^4}\right)_m = \left(\frac{EI'}{\rho V^2 l^4}\right)_a \tag{2.6}$$

$$\left(\frac{GJ'}{\rho V^2 l^4}\right)_m = \left(\frac{GJ'}{\rho V^2 l^4}\right)_a \tag{2.7}$$

The bending and torsional stiffness ratios between the scale model and the aircraft that are used for achieving aeroelastic similitude are derived from equation 2.6 and 2.7. The stiffness ratios are given in equations 2.8 and 2.9.

$$\frac{EI'_m}{EI'_a} = \frac{q_m}{q_a} n^4 \tag{2.8}$$

$$\frac{GJ'_m}{GJ'_a} = \frac{q_m}{q_a} n^4 \tag{2.9}$$

The bending and torsional stiffness ratios are dependent on the scale factor ratio and the dynamic pressure ratio. Assuming the dynamic pressure ratio to be one, a 1/10th scale model will, therefore, require a bending and torsional stiffness that is ten thousand times smaller than the stiffness of the full-scale aircraft for achieving elastic similitude of the structure. The required stiffness of the scale model can, however, be increased by flying at a lower altitude or increasing the flight velocity.

2.1.2. Flow Similitude

The essence of flow similitude or similarity is to achieve the same fluid flow behaviour between the scale model and the full-scale aircraft. Flow similarity is achieved by setting up the required atmospheric test conditions of the scale model for achieving similitude. There are two types of flow similarities and these are the kinematic similarity and the dynamic similarity.

Kinematic Similarity

Kinematic similarity is the similarity of the fluid motion between the model and the aircraft. The similarity of fluid motion means that an air particle at any point in the airflow of the scale model must have a velocity that is *n* times smaller or bigger than the velocity of the air particle at a similar point in the airflow of the aircraft. *n* is a constant factor and it never changes when the location in the airflow is changed. Kinematic similarity ensures that the airflow between the scale model and the aircraft has the following properties:

1. The paths of homologous moving particles are geometrically similar (i.e. fluid streamlines are similar) [4].
2. The ratios of the velocities of homologous particles are equal [4].

The motion of an object or particle is described by velocity and acceleration. The velocity and the acceleration of an object are written in terms of distance and time. See equations 2.10 and 2.11 [4].

$$\frac{V_m}{V_a} = \frac{l_m}{t_m}\frac{t_a}{l_a} = \frac{n}{t_r} \tag{2.10}$$

$$\frac{a_m}{a_a} = \frac{l_m}{{t_m}^2}\frac{{t_a}^2}{l_a} = \frac{n}{{t_r}^2} \tag{2.11}$$

The velocity and the acceleration of an air particle are dependent on time and can, therefore, be used to derive time ratio between the model and the aircraft. Rewriting equation 2.10 and the time ratio can be expressed as follows:

$$\frac{t_m}{t_a} = \frac{V_a}{V_m} n \tag{2.12}$$

The time ratio of equation 2.12 has a direct impact on the duration of the movement between the scale model and the full-scale aircraft. A scale model that is for example smaller than the full-scale aircraft will, therefore, experience the duration of a movement to be shorter than a similar movement of the full-scale aircraft. If it is an oscillating movement than the frequency of the movement is higher for smaller scale model than for the full-scale aircraft.

Dynamic Similarity

Dynamic similarity is the similarity of forces. The similarity of forces between dynamic similar systems means that the ratio of forces between the two dynamic similar system is fixed. This of course only applicable if the forces are located at points that are similar for both systems. This similarity is shown in equation 2.13.

$$\frac{F_{m_{pointA}}}{F_{a_{pointA}}} = \frac{F_{m_{pointB}}}{F_{a_{pointB}}} = \frac{F_m}{F_a} \tag{2.13}$$

A particle in a flow may experience a lot of different forces with different causes for each force. Each of these forces will provide a condition for achieving its own 'dynamic similitude'. The most significant forces that are relevant for dynamically scaled flight testing are viscous, pressure and gravity forces. The conditions for each of these forces are explained below. For the derivation of these conditions please consult the following papers [3] and [4].

- **Viscous force** is the resistance of a fluid to flow. This viscous force is the result of the shear stress or friction between the layers of a fluid flow. The parameter that describes this viscosity of the fluid is the Reynolds number. For achieving similarity of viscous forces between the different cases, the Reynolds number must be the same. The similarity condition for viscosity is given in equation 2.14.

$$\left(\frac{Vl}{\nu}\right)_m = \left(\frac{Vl}{\nu}\right)_a \rightarrow Re_m = Re_a \tag{2.14}$$

  Equation 2.14 can be rewritten to the Reynolds number ratio given in equation 2.15 and this ratio must be equal to 1 for achieving similarity in viscous forces between the scale model and the full-scale aircraft.

$$\frac{Re_m}{Re_a} = \frac{\nu_a}{\nu_m}\frac{V_m}{V_a}n \tag{2.15}$$

  Equation 2.15 shows us that a 1/10th scale model has Reynolds number that is ten times smaller than the Reynolds number of the full-scale aircraft. The viscous effects are thus larger on the scale model than on the full-scale aircraft. Satisfying the similarity condition for viscous forces requires the scale model to be tested at a condition, where either the flight velocity of the scale model is ten times higher, or the kinematic viscosity of the atmospheric test condition must be ten times smaller or a combination of both.

- **Pressure force** in a fluid is a force that arises from different pressures in a flow field. Fluid can be incompressible like liquids or compressible like gases. For incompressible fluids, the parameter that is used for achieving the similarity of pressure forces is the Euler number. The similarity condition for incompressible fluids is given in equation 2.16. The similarity condition for compressible fluids can be derived by introducing equation 2.17. Combining equation 2.17 with equation 2.16 and the similarity condition for compressible fluids can be derived. The similarity condition for compressible fluids is given in equation 2.18.

$$\left(\frac{p}{\rho V^2}\right)_m = \left(\frac{p}{\rho V^2}\right)_a \rightarrow Eu_m = Eu_a \tag{2.16}$$

$$V_S = \sqrt{\gamma RT} = \sqrt{\gamma\frac{p}{\rho}} \tag{2.17}$$

$$\left(\frac{V}{V_S}\right)_m = \left(\frac{V}{V_S}\right)_a \rightarrow M_m = M_a \tag{2.18}$$

  Equation 2.18 can be rewritten to the Mach number ratio given in equation 2.19. This ratio must be equal to 1 for achieving similarity in pressure forces between the scale model and the full-scale aircraft.

$$\frac{M_m}{M_a} = \frac{V_m}{V_a}\frac{V_{S,a}}{V_{S,m}} \tag{2.19}$$

According to equation 2.19, a scale model flying at a lower velocity than the full-scale aircraft can still satisfy the similarity condition for the pressure force of the compressible fluid. Satisfying this similarity condition is achieved by lowering the test altitude of the scale model, hence lowering the speed of sound.

- **Gravity force** is another force that is acting upon the flow of fluid particles. The parameter for achieving the similarity condition for the gravity force is called the Froude number. Froude number is the ratio of inertial forces to gravitational forces, see equation 2.20. The similarity condition for the gravity force is given in equation 2.21 [3].

$$\frac{ma}{mg} = \frac{\rho l^2 V^2}{\rho l^3 g} = \frac{V^2}{lg} \tag{2.20}$$

$$\left(\frac{V^2}{lg}\right)_m = \left(\frac{V^2}{lg}\right)_a \rightarrow Fr_m = Fr_a \tag{2.21}$$

Equation 2.21 can be rewritten to the Froude number ratio given in equation 2.22 and this ratio must be equal to 1 for achieving similarity in gravity forces between the scale model and the full-scale aircraft.

$$\frac{Fr_m}{Fr_a} = \frac{g_a}{g_m}\frac{V_m}{V_a}^2\frac{1}{n} \tag{2.22}$$

Equation 2.22 shows us that a 1/10th scale model has a Froude number that is ten times bigger than the Froude number of the full-scale aircraft. The effect of a larger Froude number can be explained by using the following example [3], where an aircraft is performing a banked turn at a constant altitude as shown in figure 2.1. The bank angle of the aircraft is given in equation 2.23 and is defined as the inverse tangens of the centrifugal force divided by the aircraft weight.

$$\tan(\phi) = \frac{F_c}{W} \tag{2.23}$$

Filling the formula of the centrifugal force and the aircraft weight into equation 2.23 gives us the following equation for the bank angle:

$$\tan(\phi) = \frac{mV^2}{R}\frac{1}{mg}$$

$$\phi = \tan^{-1}\left(\frac{V^2}{gR}\right) \tag{2.24}$$

Equation 2.24 shows that the bank angle is dependent on the Froude number. A 1/10th scale model that has a higher Froude number than the full-scale aircraft requires a bigger bank angle to perform a banked turn at constant altitude.

The above items all gives similarity conditions that are directly related to the forces that are acting upon the fluid. **Another important parameter for achieving similitude is the Strouhal number**. Strouhal number, or reduced frequency parameter, is used to establish similitude for the unsteady flow effects caused by the oscillatory perturbations of the aircraft [2]. The periodic similitude condition for unsteady flow effects is defined as follows [3]

$$\left(\frac{\omega l}{V}\right)_m = \left(\frac{\omega l}{V}\right)_a \rightarrow Str_m = Str_a \tag{2.25}$$

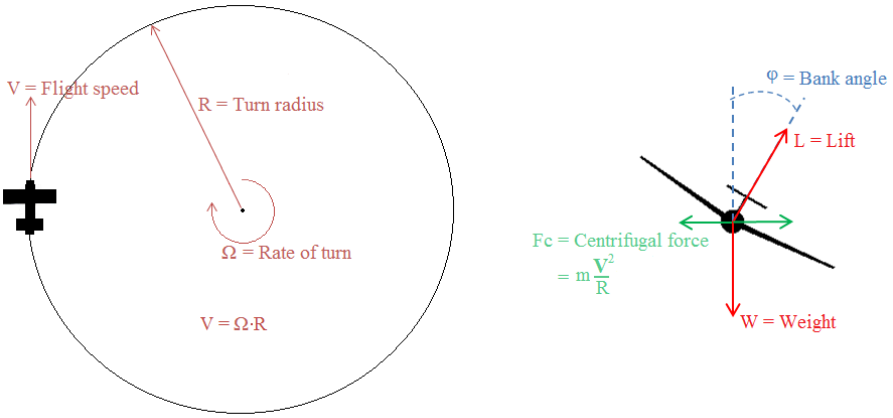


Figure 2.1: Constant altitude banked turn

2.1.3. Scaling Principles

Accurate predictions of the performance of the full-scale aircraft require the model to satisfy a set of similitude conditions. Table 2.1 shows the similitude conditions for designing the scale models. These conditions are from a NASA technical paper [3], which derives the conditions from a dimensional analysis of the relevant parameters for the forces and the moments. The result of the dimensional analysis is fourteen dimensionless parameters that represent the requirements for static and dynamic similitude between the model and the aircraft.

Table 2.1: Similitude conditions for designing scale models [2]

Airstream attitude relative to free stream	α, β	
Control deflection	δ	
Mach number	V/V_S	
Reynolds number	$\rho V l/\mu$	Static model
Reduced angular velocity	$\Omega l/V$	Rotary balance
Strouhal number	$\omega l/V$	Forced oscillation
Reduced linear acceleration	al/V^2	
Reduced angular acceleration	$\dot{\Omega} l^2/V^2$	
Froude number	V^2/lg	
Relative density factor	$m/\rho l^3$	
Relative mass moment of inertia	$I/\rho l^5$	Dynamic model
Aero-elastic bending parameter	$EI'/\rho V^2 l^4$	
Aero-elastic torsion parameter	$GJ'/\rho V^2 l^4$	
Reduced time parameter	tV/l	

Table 2.1 shows that accurate predictions of full-scale characteristics require correlation of many, sometimes conflicting, similitude parameters. This is especially true for dynamically scaled flight testing, where the models are tested under atmospheric conditions. For dynamically scaled flight testing,

Froude, Mach and Reynolds numbers are rarely satisfied simultaneously. Therefore one of the three conditions is usually satisfied and the other conditions will be designed such that the discrepancy is as small as possible. Three scaling principles are thus available for designing a dynamically scaled flight test. The three scaling principle are Froude scaling, Mach scaling and Reynolds scaling. Froude scaling and Mach scaling procedures are widely used for designing dynamically scaled models. Reynolds scaling, however, is rarely used for designing dynamically models. The reasons for this will be explained below. The scale factors of the three scaling procedures are given in appendix A.

Froude Scaling
The Froude scaling procedure is often applied in designing the flight test when the airflow around the scale model and the aircraft is assumed to be incompressible. Froude scaling preserves the kinematic properties between the model and the full-scale aircraft by satisfying the Froude number similitude requirement. Meeting the Froude number requirement ensures the similitude of the attitude angles and the load factors between the scale model and the full-scale aircraft. Ensuring the similarity of attitude angles and load factors allows the motion dynamics of the model to be properly correlated to the full-scale aircraft.

Mach Scaling
Testing conditions involving compressibility effects are scaled according to Mach scaling procedure. In Mach scaling, the compressibility effects between the model and the full-scale aircraft are the same through the adherence of Mach number requirement. This scaling procedure is often preferred above Froude scaling when the compressibility effects are expected to have a significant effect on the results. The downside of Mach scaling (unless Froude number requirement is also matched) is that the lift coefficient and hence the attitudes between the model and full-scale aircraft are not the same.

Reynolds Scaling
The advantage of Reynolds scaling is that the Reynolds number between the model and full-scale aircraft is the same, which means that flow structure and its behaviour will be identical. The downside of Reynolds scaling is the Mach and Froude number dissimilarity. According to table A.1, Froude and Mach number scale factors increase when the model size is decreased.

The effect of Mach number dissimilarity is apparent. A smaller model will have a higher Mach number than the full-scale aircraft and it will thus experience more compressibility effects than full-scale aircraft. If the model scale is small enough then the model will experience supersonic. The other downside of Reynolds scaling is the Froude number dissimilarity. The Froude number scaling factor scales to the power of three on the model scale and the Froude number dissimilarity becomes very big very quickly when the scale factor becomes smaller. The resulting effect is that the lift coefficient ratio between the model and the aircraft will be very high.

Conclusion
Dynamically scaled flight testing is often designed according to the Froude or Mach scaling. Reynolds scaling is seldom used to design dynamically scaled models, due to disadvantages mentioned above. Froude scaling is often applied for flight conditions, where the aircraft is performing most of its manoeuvres and that is during the take-off and landing phase. Mach scaling is often applied for the cruise conditions of the aircraft, where the compressibility has a huge effect on the aircraft performance.

2.2. Dynamically Scaled Flight Testing

Dynamically scaled flight testing is an actual flight test that uses dynamically scaled replicas of an aircraft to predict the aerodynamics and the flight dynamics of the full-scale counterpart. These flight tests are usually performed under atmospheric conditions by remotely flying the scaled models in the air. The primary usage of dynamically scaled flight tests is for analysing the flight characteristics and the handling qualities of the full-scale aircraft.

As shown in section 2.1.3, a dynamically scaled flight test must be designed according to one of the scaling principles and these scaling principles are the Froude scaling, Mach scaling or Reynolds scaling. The scaling principles provide the required atmospheric conditions for performing the dynamically scaled flight test to satisfy a certain similitude condition. Each scaling principle will, therefore, provide a different set of atmospheric test conditions since each scaling principle satisfies a different set of similitude conditions.

The scaling principle that is often used for designing the dynamically scaled flight test is the Froude scaling. Froude scaling is used because it ensures that the ratio of inertial forces to gravitational forces is similar to both the scaled model and the full-scale aircraft. A similar ratio of inertial forces to gravitational forces ensures that a certain manoeuvre like a 1g turn will take place at the same attitude angle for the scaled model and the full-scale aircraft.

Link of thesis to dynamically scaled flight testing

This thesis is part of the project for designing an optimisation framework dynamically scaled flight testing. This optimisation framework can be divided into three disciplines, which are the aerodynamics, the flight mechanics and structures.

- **Aerodynamics:** A dynamically scaled flight test, that is scaled according to the Froude scaling principle, will hence create a flight test that has a different Reynolds number and Mach number. A different Reynolds number and Mach number will, therefore, decrease the accuracy of the scaled model to predict the aerodynamic performance of the scaled model. This accuracy can be increased by optimising the geometry of the model such as changing the curvature of the airfoil to improve the lift curve slope.
- **Flight mechanics:** To ensure the similitude of the flight characteristics of the scale model and the full-scale aircraft, one must satisfy the relative density factor and the relative mass moment of inertia of table A.2. The relative density factor and the relative mass moment of inertia provide the required mass and mass moment of inertia of the scale model for ensuring the similitude of the flight characterises between the scale model and full-scale aircraft. The provided mass and mass moment of inertia are only valid for a specific atmospheric test condition. A different test condition means a different set of mass and mass moment of inertia. An optimisation process is thus required to find a feasible mass distribution for the scale model that produces the required mass and mass moment of inertia.
- **Structures:** The wing of an aircraft is not rigid, but it can bend upwards when the wing is loaded with the lift force. A scale model with similar structure as the full-scale aircraft will have different structural properties. The aeroelastic bending and torsion similitude parameters are given in table A.2 must, therefore, be satisfied to ensure a similar bending and twisting behaviour between the structure of scale model and the full-scale aircraft. The two similitude parameters give the required bending and torsion stiffness of the structure to achieve a similar structural behaviour. An optimisation process for the structural discipline is thus required to find a structure that gives both the required bending and torsional stiffness.

The dynamically scaled flight testing project revolves around three disciplines, which were the aerodynamics, the flight dynamics and the structures. Each discipline requires an optimisation process to increase the accuracy of the flight dynamics data of the scaled model. The work that has been performed in this thesis will contribute to the aerodynamic optimisation framework of the project.

This thesis will focus on developing a parametric model that can be used for performing the aerodynamic optimisation of dynamically scaled models. An optimisation procedure requires a model that can be changed easily by just changing a few parameters. A parametric modelling code that is properly coded can also be used to create different aircraft configurations such as a conventional aircraft with T-tail or a blended wing body aircraft.

Aerodynamic solver
The scale models of dynamically scaled flight testing are scaled according to the Froude scaling principle, which ensures that Froude number between the scale model and the full-scale aircraft are similar. Applying Froude scaling means dissimilarity in the Mach and Reynolds numbers between the scale model and the full-scale aircraft. The effect of Mach number dissimilarity is neglected in this thesis since most aircraft manoeuvres are performed during the take-off and landing phases of the flight envelope. The only issue remaining is the Reynolds number dissimilarity between the model and the aircraft, where the Reynolds number of the scale model is lower than the Reynolds number of the full-scale aircraft. Therefore the scale model will experience a lot more viscous effects that the full-scale aircraft.

The aerodynamic solver that will be coupled to the parametric model must, therefore, have a method that is capable of predicting the viscous effects of the scale models. The aerodynamic solver that has been chosen for computing the aerodynamic performance of the scale models is VSAERO. VSAERO is a 3D panel method that uses potential flow model for computing the aerodynamic characteristics. It is also capable of calculating the viscous effects by using an integral method for the boundary layer. VSAERO is chosen due to its capability to compute viscous effects and it uses a first order panel method for solving the aerodynamics. Since a simulation based on a first order potential based panel method are very fast when compared to a simulation of a Computational Fluid Dynamics solver.

2.3. Control Surfaces Modelling Options

The main purpose of the dynamically scale flight testing is to predict the flight characteristics and handling qualities of the full-scale aircraft by using the flight characteristics and the handling qualities of the dynamically scaled model. Determining the flight characteristics requires the stability and control derivatives to be determined first.

Calculating the control derivatives of an aircraft for determining the handling qualities of the aircraft requires the modelling of the control surface deflection. The modelling of the control surfaces or moveables for a 3D panel method can be modelled in two ways, see figure 2.2. The first approach creates a slotted moveable with two separate airfoils, one for the main wing and one for the moveable. The second approach, on the other hand, creates one airfoil that contains the moveable and the wing, while the deflection of the moveable is represented as a 'bent' in the aft part of the airfoil.

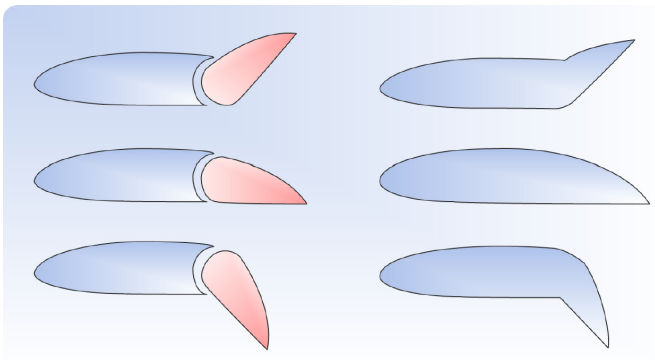


Figure 2.2: Slotted moveables (left) and non-slotted moveables (right)[5].

Modelling the wing moveable as a slotted moveable is physically more accurate than modelling it as a not-slotted moveable. The downside of the slotted moveable approach is that constructing such a model for a first order panel method is more complex and requires more work than creating a model for the not-slotted moveable approach. The complexity originates from the fact that a panel method requires the wake of the wing to be defined beforehand. This means that the slotted approach requires three wake lines to be modelled beforehand. Two wake line starts from the top and the bottom trailing edge of the wing, while the third wake line is attached to the trailing edge of the moveable. Generating the two wake lines of the wing adds additional complexity and it is because two wake lines from the wing must be modelled around the curvature of the moveable. Senden[6] has used this method to create a wing with a slotted flap for his research on the performance of an elliptical flap.

The main advantage of the not-slotted approach is that the wing and moveable forms one entity. This

advantage requires only one wake to be attached to the wing-moveable combination and is, therefore, a lot easier and less time consuming to develop a useable model than the slotted approach. The not-slotted approach is therefore chosen for this thesis work. This approach is widely used and yields good results as demonstrated by the following thesis works of van Ginneken[5], Grotenhuis[7] and Arntzen[8].

Non-Slotted Modelling Options

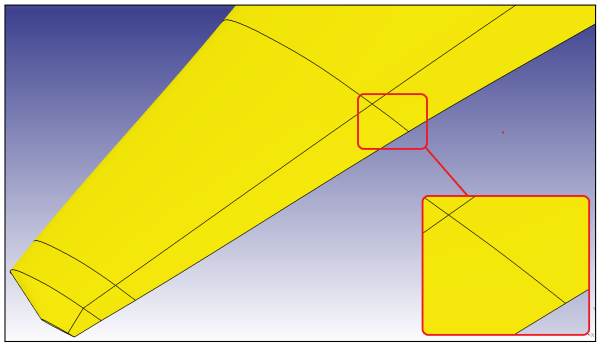
There are many different options to create a wing with not-slotted moveables for aerodynamics solvers that are based on the first order panel method. Three of these modelling options are being investigated in this thesis work to find the most optimal modelling option. The three options are enumerated in the list below:

- **Normal rotation** is a mathematical method in the aerodynamic solver VSAERO that allows the user to rotate the normal vector of the moveable panels by only defining the rotation vector and the deflection angle, see figure 2.3a.
- **Transition surface** is a method that physically models the moveable deflection. Deflecting the moveable creates a discontinuity in the wing surface in the spanwise direction. This discontinuity is smoothed out by creating a transition surface from the wing to the moveable, see figure 2.3b.
- **Gap** is a similar modelling method as the transition surface but instead of creating a transition surface a gap is created between the wing and moveable, see figure 2.3c.

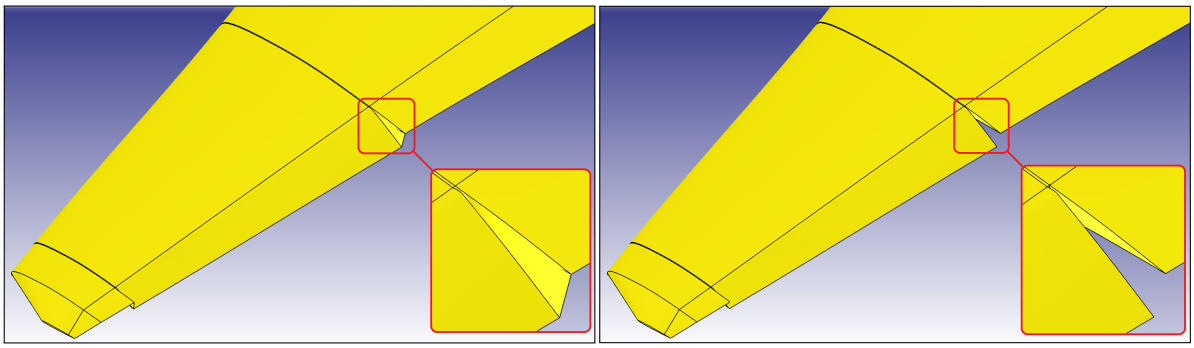
The main advantage of the 'normal rotation' option is its simplicity. This option does not require any steps to physically model the deflected moveable in the wing of the aircraft, which makes it very easy to implement when compared to the two other options.

The 'transition surface' and 'gap' options require a lot more work to create the wing with moveables when compared to the 'normal rotation' option. The 'transition surface' modelling option is the most labour intensive option out of three modelling options. The models, created by the 'transition surface' and the 'gap' approach, is expected to produce a more accurate result than the 'normal rotation' option since these two options are geometrically more accurate.

The difference in the results between the 'transition surface' and the 'gap' options is expected to be



(a) Normal rotation option



(b) Transition surface option

(c) Gap option

Figure 2.3: Moveable modelling options

noticeable when the models are experiencing a side slip angle with its moveables deflected. An aircraft in a side-slip experience different airflow over the right and the left wing. The extra vertical surfaces of the transition surface method are expected to create different side forces on the right and left wing due to asymmetrical flow over the wings.

2.4. Thesis Objective and Goals

This thesis is part of a project that focuses on developing an optimisation framework for dynamically scaled flight testing. The optimization framework must design the scale model and the flight test such that the performance of the scale model is representative to the full-scale aircraft. The similarity in the performances between the scale model and the full-scale aircraft can be achieved by altering the geometry, the mass distribution and the structure of the model, which are related to the aerodynamic, flight dynamic and structures disciplines respectively.

This thesis contributes to the aerodynamics optimisation framework by focusing on the development of a parametric model that is capable of deriving the stability and control derivatives. A parametric aircraft model is a model that defines a finite set of parameters for modelling the aircraft model. A parametric model is needed since an optimisation process requires a model that can be changed easily by varying the modelling parameters. Also, a properly designed parametric modelling code can be used for creating different aircraft configurations like a blended wing body aircraft or T-tail aircraft. The objective of this thesis can thus be defined as follows:

> Create a generic parametric model of an aircraft with trailing edge moveables that is coupled with a first order potential based panel method with an integral boundary layer method for deriving the stability and control derivatives.

The research goals of the thesis are mostly related to the construction of an aircraft model for the aerodynamic solvers that are based on a first order panel method like VSAERO. The most important goal is how to model the trailing edge moveables for a first order panel method for deriving the control derivatives. This thesis will investigate three methods for creating the moveables. The first option is a mathematical operation in VSAERO that rotates the normal vectors of the body panels that represents the moveable. The second option is creating a model with moveables that is not slotted and has transition surfaces in the spanwise direction. The third option is also a wing with not-slotted moveable but without transition surfaces in the spanwise direction. The research goals are:

- Determine the best option out of the three proposed moveable options to model the moveables of a wing for an aerodynamic solver that is based on the first order panel method.
- Compute the stability & control derivatives of the parametric model and investigate the accuracy of these derivatives.

3

Parametric Aircraft Tool

The Multi-Model Generator or MMG is an aircraft modelling tool that has been conceived by La Rocca and others [9] in 2003. The original MMG was created on the ICAD design platform, which employs knowledge based engineering or KBE method for modelling the aircraft.

The MMG models the aircraft on an abstract level, which it calls High Level Primitives or HLP. The HLPs can be distinguished into four distinct categories, see figure 3.1. As shown in the figure the wing trunk HLP can be used to create the main wing, horizontal tail, winglet etc. The MMG allows the engineer to combine different HLPs to create its idea of the new aircraft such as a conventional aircraft with a cruciform tail or an unconventional aircraft such as a blended wing body aircraft.

The primary function of the MMG is to adapt the geometric model that is being created with the HLPs such that it results in a model that can be used for discipline-specific analysis tools such as VSAERO(aerodynamic) and NASTRAN(structural).

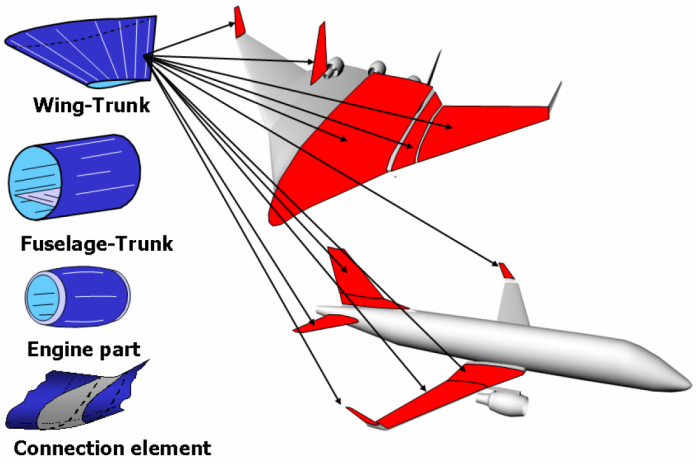


Figure 3.1: The High Level Primitives. Example of the wing-trunk multiple instantiation in different aircraft configurations [10].

The original Multi-Model Generator was created on the ICAD platform, which was acquired by Dassault Systemès in 2002. After the acquisition, the ICAD platform was taken off the market. Due to unavailability of the ICAD platform, a new MMG tool was created by van den Berg[11], Koning[12] and van Hoek [13]. The new MMG was created on a different KBE system named Genworks GDL. Genworks GDL is a similar program as ICAD since both are KBE systems based on the LISP coding language.

The original idea of this thesis was to use the MMG that has been written on the Genworks GDL platform, to create an aircraft model with moveables that can be exported to the aerodynamic solver VSAERO. Since this version of the MMG tool was already capable of generating a wing with slotted moveables and has a mesh application for meshing various aircraft models that have a clean wing. This application was created by Abedian Kalkhoran[14]. The author has used this mesh application for

the MMG tool and has concluded that this application was not working properly in the version GDL1588 of Genworks GDL. The application does not function properly because some operations involving the geometry of the model produces results that were incorrect. Due to this bug and some other reasons the author has decided to create another version of the MMG that runs on the ParaPy platform. The reason for creating another version of the MMG was:

- There was some issues revolving around the robustness of the geometry models in Genworks GDL version GDL1588, which could not be solved by the author.
- The mesh application of Abedian Kalkhoran[14] was constructed for wings without moveables. Adapting the code to work with moveables was a daunting task and required a lot changes in the structure of the mesh application. Luckily the ParaPy platform has a dedicated toolbox that automates the meshing procedure for complex geometry models[15]. The user of this toolbox only needs to define the rules that control the mesh settings of various parts of the geometry[15]. Using this toolbox will greatly reduce the development time of a new mesh application for the new MMG tool on the ParaPy platform.
- ParaPy is a new KBE system that is built on the popular Python language[15]. Using ParaPy is thus very easy and intuitive, while this not the case for Genworks GDL that uses the outdated and hard to learn LISP language.
- The ParaPy platform has a unique feature that tracks all the individual edges and faces of a geometrical model. This feature allows the user to easily extract a new edge or face from a modified geometrical model by simply specifying the old edge or face and insert it into this feature.

3.1. Multi-Model Generator on Genworks GDL Platform

The modules for generating the wing trunk and the fuselage trunk are the only High Level Primitives that has been modelled in the Multi-Model Generator. The engine and the connection element modules are still lacking in the MMG. The module that is responsible for modelling the wing trunk is called DARwing. The fuselage, on the other hand, is modelled by the module called DARfuse.

The DARwing module is the code module in the MMG that is used for creating lifting surfaces of the aircraft. The definition of the DARwing module is shown in the aircraft model tree of figure 3.2. The DARwing module is capable of producing a clean wing or a multi-elements wing including the structural components of the wing. The clean wings produced by the DARwing module can then be exported to VSAERO for aerodynamic analysis. For more information about DARwing, consult the thesis work of van den Berg[11] and Koning[12].

The fuselage of the MMG is modelled by DARfuse code module. DARfuse models the fuselage by defining the cross sections of the fuselage. These cross sections are generated by using CST-coefficients for defining the various cross sections of the fuselage. These cross sections are then sized and positioned using four guiding rails that stretch from the nose to tail. The four rails are placed at top, bottom, left and right of the fuselage. Besides the generation of the outer skin of the fuselage, DARfuse is also capable of designing the cabin and performing a class II weight estimation analysis. For more information about DARfuse then read the thesis of Brouwers[16].

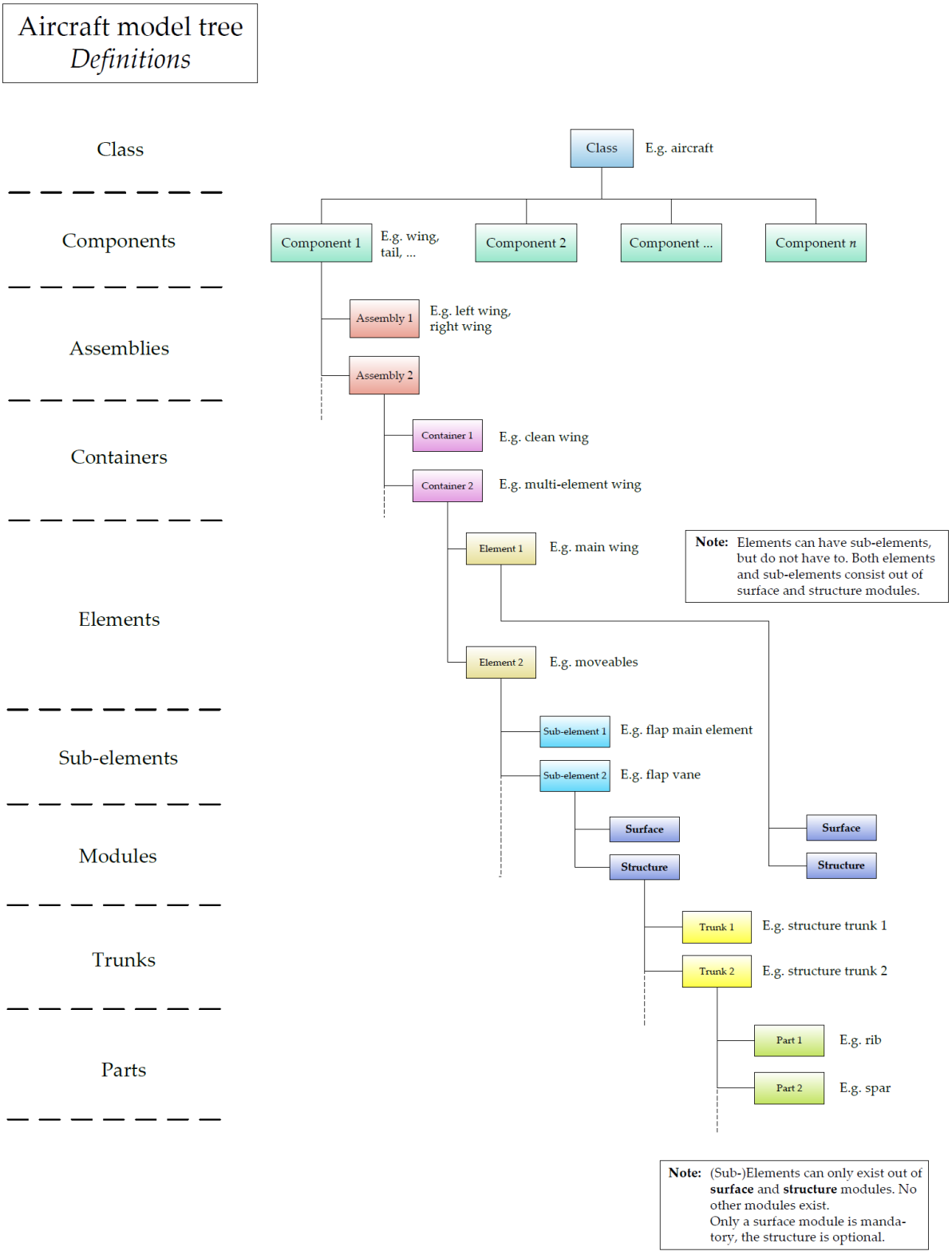


Figure 3.2: Aircraft object tree definition of the DARwing MMG[11] written on the Genworks GDL platform

3.2. Multi-Model Generator on ParaPy Platform

The new MMG tool written on the ParaPy platform does not yet have all the functionalities of the old MMG tool written on the Genworks GDL platform. Only some of the functionalities of the DARwing and the DARfuse modules has been recreated in the new version of the MMG. The functionalities that have been recreated are explained in the subsection below.

3.2.1. Clean Wing Generation

For this thesis, the module for generating the clean wings of the new MMG has been recreated by using the old clean wing module of the old MMG as reference [11]. Other functionalities like generating the multi-elements wing are not yet available in this new version of the MMG.

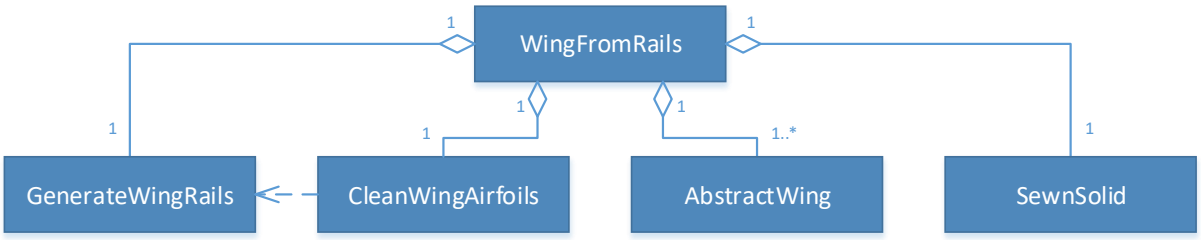


Figure 3.3: UML class diagram of the clean wing module

The structure of clean wing module is shown in the Unified Modelling Language diagram or UML-diagram of figure 3.3. The clean wing model is generated in the *WingFromRails* class, which contains four subclasses for modelling the components of the clean wing. The four subclasses represent the four steps of the module for creating the clean wing model. These four steps are illustrated in figure 3.4. The first step is to generate the leading edge rail and the trailing edge rail of the clean wing. These

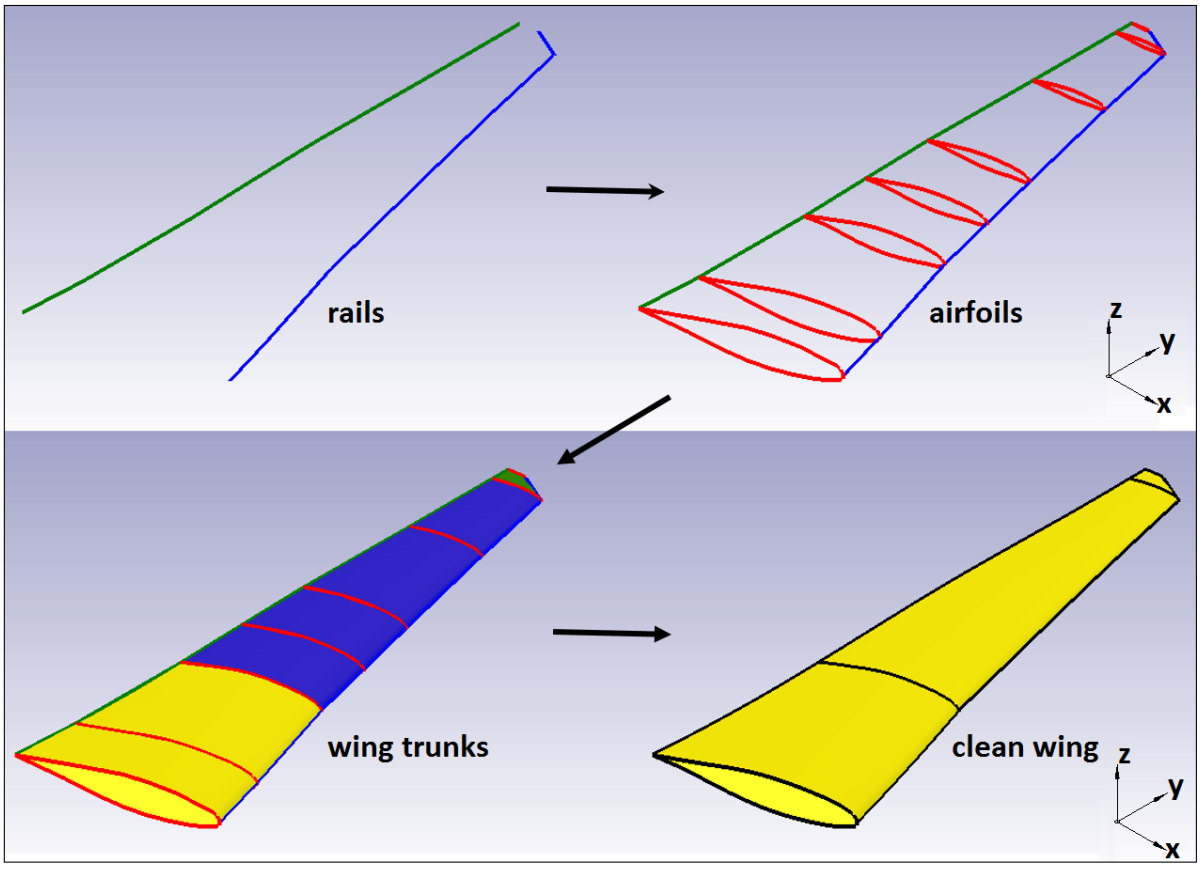


Figure 3.4: Clean wing generation procedure

rails contain all the information of the clean wing such as the dihedral and the twist angles distributions of the clean wing. The second step creates the airfoil curves of the clean wing model and these airfoil curves are then positioned and orientated using the leading edge and trailing edge rails of step 1. The positioned airfoil curves are used in step 3 for creating the wing trunk models by lofting the airfoil curves. The last step is to collect the wing trunk models and sew it together into one model.

The procedures for creating the wing rails and the airfoil curves will not be explained in this thesis since a good explanation can be found in the thesis of van den Berg [11]. Instead, the input parameters for creating the clean wing will be explained in this remaining section.

Rails input parameters

The leading edge and trailing edge rails of the clean wing module is created from five curves, which are the flat leading edge rail, flat trailing edge rail, dihedral rail, twist axis and the twist line. These five curves are created from the five input parameters listed below:

- **le_point_list** is a list of XYZ-coordinates for creating the flat leading edge rail of the clean wing model. A flat leading edge rail does not have any information about the dihedral and the twist distributions of the clean wing. A flat leading edge rail is a two-dimensional curve in the XY-plane.
- **te_point_list** is a list of XYZ-coordinates for creating the flat trailing edge rail of the clean wing model. A flat trailing edge rail does not have any information about the dihedral and the twist distributions of the clean wing. A flat trailing edge rail is a two-dimensional curve in the XY-plane.
- **dihedral_point_list** is a list of XYZ-coordinates for creating the dihedral rail of the clean wing model. The dihedral rail is a two-dimensional rail in the YZ-plane, where the Z-coordinate represents the local height of the wing at the spanwise location of the Y-coordinate. The dihedral rail describes the dihedral distribution of the wing and it is used for applying the dihedral distribution on the flat leading edge rail and the trailing edge rail.
- **twist_axis_point_list** is a list of XYZ-coordinates for creating the twist axis of the clean wing model. This axis functions as the rotation axis for applying the twist distributions on the flat leading edge rail and the flat leading edge rail.
- **twist_points** is a list of XYZ-coordinates for creating the twist line of the wing model. The twist line is a two-dimensional curve in the YZ-plane, where the Z-coordinate represents the twist angles of the wing in degrees at the spanwise location of the Y-coordinate. This line describes the twist angle distribution of the wing and it is used for applying the twist distribution on the flat leading edge rail and the flat trailing edge rail.

Airfoil curves input parameters

The airfoil curves of the clean wing are created and positioned using the following input parameters:

- **airfoils** is a list of strings. These strings indicate the file names of the text files that contains the airfoil coordinates. The length of the *airfoils* input list determines the number of airfoil curves that will be created.
- **span_positions** is a list of normalized Y-coordinates ranging from 0 to 1. These normalised Y-coordinates determine the spanwise position of the airfoil curves.
- **kink_indices** is a list of integers, where each integer represents the index for the *span_positions* input list. This index can be used to retrieve the spanwise position of the kink.
- **airfoil_thickness** is a list of thickness percentages for each airfoil curve. These percentages are used for scaling the thickness of the airfoil curves.
- **airfoil_cant** is a list of strings. These strings are used for choosing the orientation options of the airfoil curves when observed from above the wing. The available options for orienting the airfoil curves are shown in figure 3.5.
- **follow_dihedral** is a list of 0 and 1. The 0 or 1 indicates the option for orientating the airfoil curves when observed from the front. The orientation options are illustrated in figure 3.6.

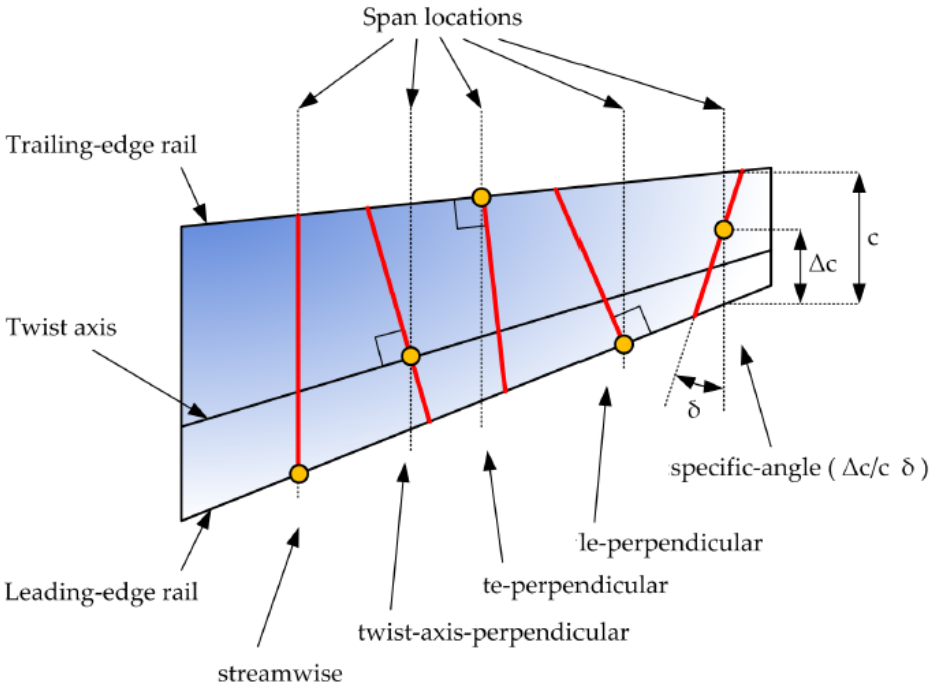


Figure 3.5: Options for the cant angle of an airfoil [11]

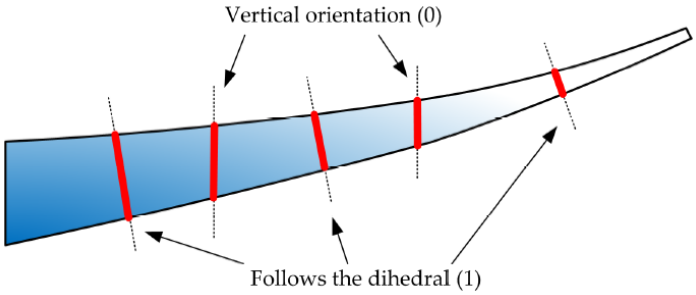


Figure 3.6: Options for the frontal orientation of an airfoil [11]

3.2.2. Fuselage Generation

The fuselage module for generating the fuselage model has been recreated by using the DARfuse module of the old MMG as reference [16]. Other functionalities such as modelling the structural components of the fuselage are not yet available in this new version of the MMG.

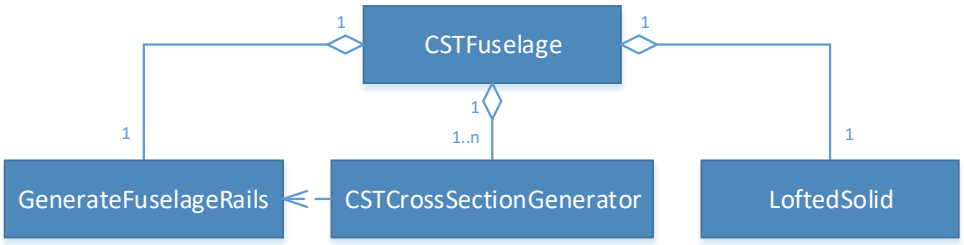


Figure 3.7: UML class diagram of the fuselage module

The structure of the fuselage module is shown in the UML-diagram of figure 3.7. The fuselage is generated by the *CSTFuselage* class, which contains three subclasses for modelling the components of the fuselage model. The three subclasses represent the three steps of the module for creating the fuselage model. These steps are illustrated in figure 3.8. The first step of the modelling process is to

create the four rails of the fuselage. These rails are used for scaling the fuselage cross section curves of step 2. The second step of the modelling process is to generate the normalised cross sections curves of the fuselage. The width of these normalised curves are scaled using the two green rails of figure 3.8 and the height of these normalised curves are scaled using the two yellow rails of figure 3.8. The last step of the modelling process is to generate the fuselage model by making a loft from the fuselage cross sections curves.

The procedures for creating the fuselage rails and the fuselage cross section curves will not be explained in this thesis since a good explanation can be found in the thesis of Brouwers[16]. Instead, the input parameters for creating the fuselage model will be explained in this remaining section.

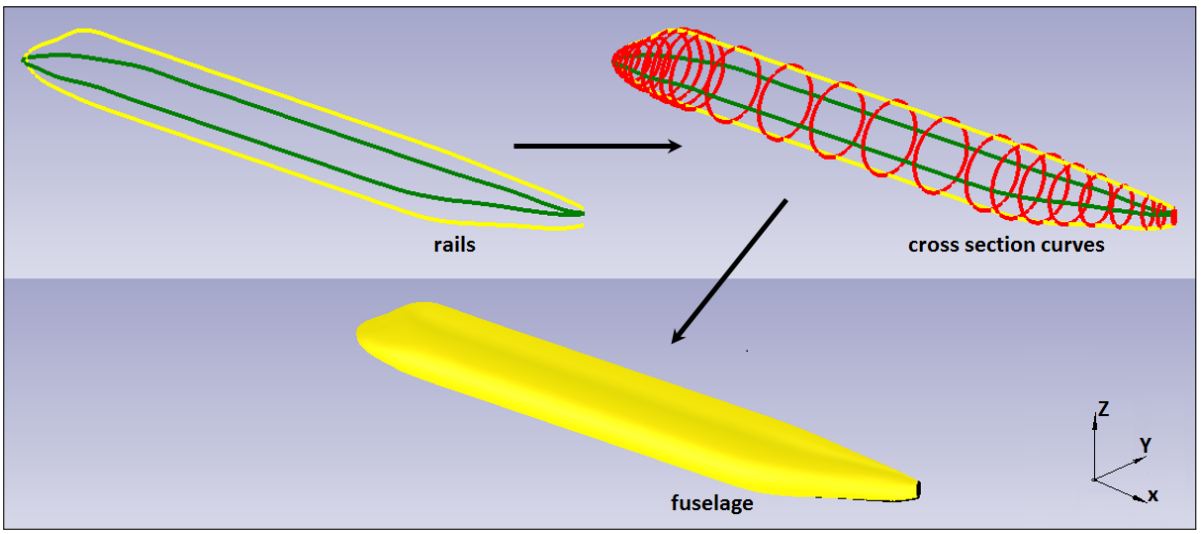


Figure 3.8: Fuselage generation procedure

Rails input parameter

The two yellow rails of figure 3.8 are the top and belly curves of the fuselage. The top and belly curves are two-dimensional curves in the XZ-plane. These curves are created as B-spline curves using the following input parameters:

- **top_extremes_points_list** is a list of two points which are the start point and the end point of the top fuselage curve.
- **top_control_points_list** is a list of XYZ-coordinates of the control points for the top fuselage curve excluding the start and the end points.
- **top_weights_list** is a list of floats. These floats are the weights for the control points of the top fuselage curve.
- **belly_extremes_points_list** is a list of two points, which are the start point and the end point of the belly fuselage curve.
- **belly_control_points_list** is a list of XYZ-coordinates of the control points for the belly fuselage curve excluding the start and end points.
- **belly_weights_list**is a list of floats. These floats are the weights for the control points of the belly fuselage curve.

The two green rails of figure 3.8 are the left and the right side curves of the fuselage, which are called the side fuselage curve and the mirrored side fuselage curve. The mirrored side fuselage curve is a mirrored version of the side fuselage curve and it is mirrored in the XZ-plane. The side fuselage curve is a three-dimensional curve and it is created from two B-spline curves. The two B-spline curves are two dimensional, which is called the flat side curve and the side dip curve. The flat side curve is a flat curve in the YZ-plane, while the side dip curve is a flat curve in the XZ-plane. The flat side curve is a two-dimensional version of the three-dimensional side curve. The three-dimensional side curve is created by projecting the flat side curve into a plane that is created by extruding side dip curve into the direction of the global Y-axis. The two B-spline curves are created from the following input parameters:

- **side_extremes_points_list** is a list of two points, which are the start point and the end point of the flat side fuselage curve.
- **side_control_points_list** is a list of XYZ-coordinates of the control points for the flat side fuselage curve excluding the start and end points.
- **side_weights_list** is a list of floats. These floats are the weights for the control points of the flat side fuselage curve.
- **side_dip_extremes_points_list** is a list of two points, which are the start point and the end point of the side dip fuselage curve.
- **side_dip_control_points_list** is a list of XYZ-coordinates of the control points for the side dip fuselage curve excluding the start and end points.

Cross section curves input parameters
The cross-sectional curves of the fuselage are created by using the Class Shape Transformation method or in short the CST method [17]. A good explanation of the CST-method with examples is shown in the thesis of Brouwers[16]. The parameters for creating the fuselage cross sections are:

- **long_location** is a list X-coordinates indicating the X-positions of each fuselage cross section.
- **N_up** is a list of a list of two class function exponents for each fuselage cross section. The class function exponents of this input list are used for creating the top fuselage curves of each fuselage cross section.
- **N_low** is a list of a list of two class function exponents for each fuselage cross section. The class function exponents of this input list are used for creating the bottom fuselage curves of each fuselage cross section.
- **A_up** is a list of a list of scaling factors for each fuselage cross section. The scaling factors of this input list are used for scaling the upper section of the fuselage cross section.
- **A_low** is a list of a list of scaling factors for each fuselage cross section. The scaling factors of this input list are used for scaling the lower section of the fuselage cross section.

4

Moveable Modelling

This chapter informs the reader about the design of the modelling code to create the moveables for a wing of an aircraft. The moveable modelling code is designed such that the wing with moveable model is compatible with an aerodynamic solver that is based on a first order panel method like VSAERO. There are many different options for creating a wing with moveable and three of these modelling options were chosen in section 2.3. The three modelling options are:

- **Normal rotation** is a mathematical method in the aerodynamic solver VSAERO that allows the user to rotate the normal vector of the moveable panels by only defining the rotation vector and the deflection angle, see figure 2.3a.
- **Transition surface** is a method that physically models the moveable deflection. Deflecting the moveable creates a discontinuity in the wing surface in the spanwise direction. This discontinuity is smoothed out by creating a transition surface from the wing to the moveable, see figure 2.3b.
- **Gap** is a similar modelling method as the transition surface but instead of creating a transition surface a gap is created between the wing and the moveable, see figure 2.3c.

The first section shows the structure of the code for generating the moveables of a wing. The second section explains the input parameters for modelling the three modelling options. The sizing procedure of the moveables is explained in section 4.3. The modelling process of the moveables curves is explained in section 4.4, while the last section describes the creation of the wing with moveable model.

4.1. Moveable Modelling Structure

The parametric aircraft modelling tool (Multi-Model Generator) uses an object-oriented approach for creating the aircraft models as shown in chapter 3. It is thus logical to use the same approach to design the code for modelling the wing with non-slotted moveables. The main structure of the moveable modelling tool for generating the three different moveable models is shown in the UML-diagram of figure 4.1. The UML-diagram shows that the main structure is the same for the three moveable modelling options. The differences between the options are explained in sections 4.3, 4.4 and 4.5.

The top level class for generating the wing with moveables is the *WingMoveablesFromRails* class. The *WingMoveablesFromRails* uses three subclasses for modelling the wing with moveables model. These three subclasses are the *WingFromRails* class, the *CutShapes* class and the *WingModelling-NotSlotted* class. The *WingFromRails* class is the class that generates the clean wing model. This class is the same DARwing module of the old MMG that has been recreated on the ParaPy platform, for more information see chapter 3.2. The function of the CutShapes class is to create the airfoil cross sections of the moveable sides and to determine the position of the rotation/hinge points of the moveable sides. The *WingModellingNotSlotted* class, on the other hand, initiates the objects for creating the moveables of the wing.

The modelling of the moveables starts by initiating *n* objects of the *MoveableAirfoils* class with *n* the number of moveables. The *MoveableAirfoils* class divides each airfoil cross section from the *CutShapes* class into two sections. One section representing the airfoil curves of the wing and the other section representing the movable curves. The *MoveableAirfoils* class will also model the deflected moveable curves if required. Step two of the modelling process is performed by the *SpanwiseSplitLines* class.

The *SpanwiseSplitLines* class divides the surface of the wing with moveables model into blocks that contain the outline of the moveables. The blocking strategy of the *SpanwiseSplitLines* class is to create the spanwise split curves for splitting the wing model. The last step is to create the physical model of the wing with moveable and this step is executed by the *WingSolid* class.

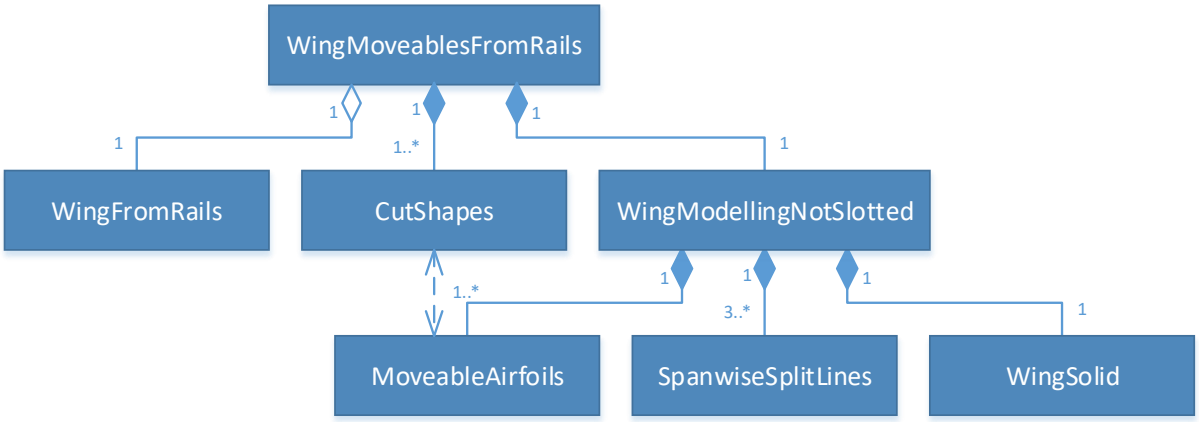


Figure 4.1: UML-diagram containing the main classes for modelling the wing with moveables

4.2. Input parameters

Creating a parametric wing model with moveables requires some parameters for modelling the moveables. The moveable of a wing is parametrised using the following input parameters:

- **span_positions** is a list of spanwise positions, that indicate the spanwise positions of the inboard and the outboard hinge/rotation points of the moveable. The input values for spanwise positions of the hinge/rotation points are percentages that are relative to the wing root with 0 for the root and 1 for the tip.
- **frontal_orientation** input parameter is used to orient the side planes of the moveable ends as shown in figure 4.2. Two options are available which are $follow\_diheral$ and $vertical$. The $follow\_diheral$ option positions the side plane such that it is perpendicular to the local dihedral angle of the wing, while the $vertical$ option positions the side plane such that it is perpendicular to the global y-axis. This input parameter is based on a similar parameter of the old MMG.

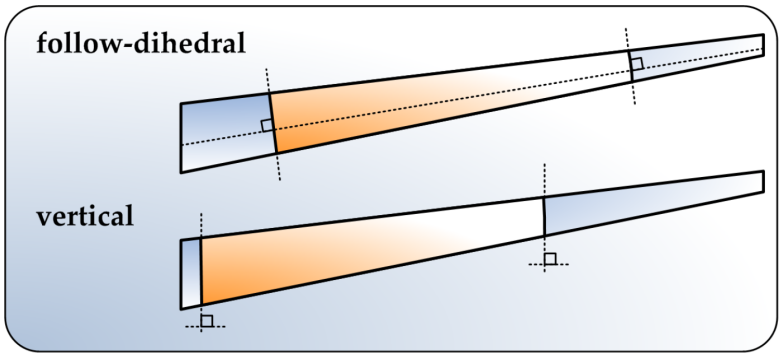


Figure 4.2: Moveable input: $frontal\_orientation$ [11]

- **edge_cant** orients the side edges of the moveable when viewed from the top of the wing, see figure 4.3. Figure 4.3 shows the possible options to orient the side planes of the moveable edges. The $streamwise$ option orient the side plane such that it is perpendicular to the freestream direction of the air. The $te\_perpendicular$, $le\_perpendicular$ and $twist\_axis$ options on the other hand orient the side plane such that they are perpendicular to the trailing edge, leading edge and twist axis respectively. Also, the side plane can be oriented by indicating an angle. This angle is

the angle between the side plane of the moveable and the global XZ-plane. This input parameter is based on a similar parameter of the old MMG.

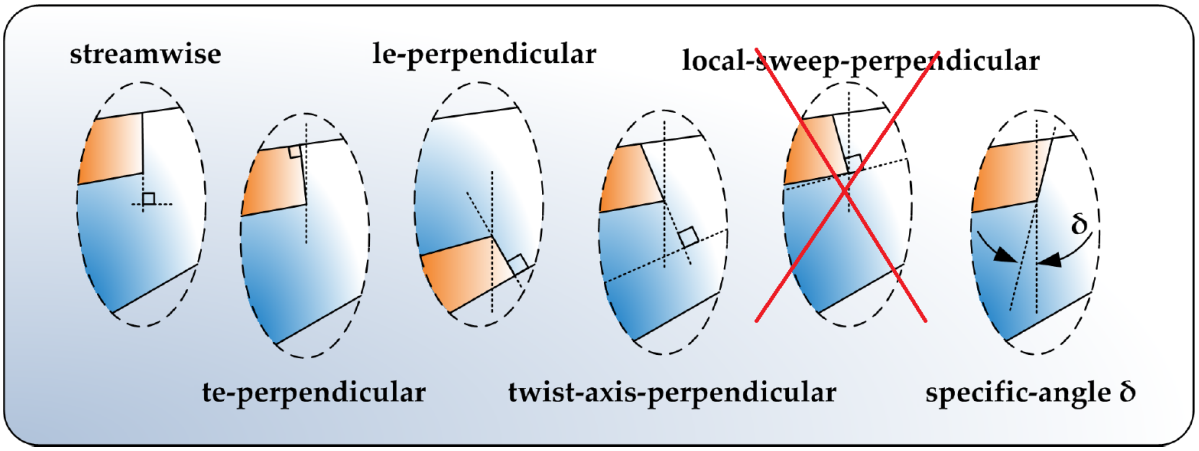


Figure 4.3: Moveable input: $edge\_cant$ [11]

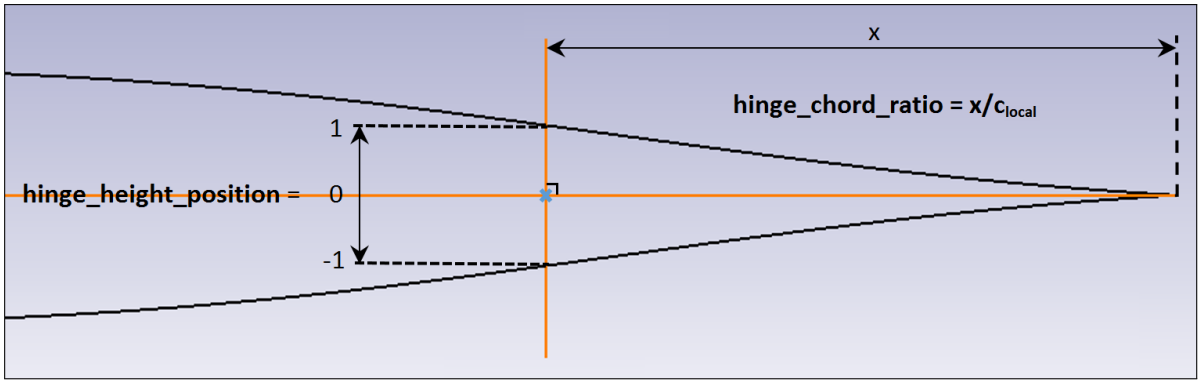


Figure 4.4: Moveable input: $hinge\_chord\_ratios$ and $hinge\_height\_position$

- **hinge_chord_ratio** is a list of normalised chord ratios, that determine the chordwise length of the inboard and the outboard side of the moveable. The input parameter indicates the chordwise position of the hinge/rotation point of the moveable with a ratio that is defined by the moveable chord length x divided by local chord length c_{local}, see figure 4.4. This ratio is then used to determine the chordwise position of the hinge point starting from the trailing edge.

- **hinge_height_position** positions the hinge points of the moveable sides in the height direction of the moveable cross-sections, see figure 4.4. This input parameter positions the hinge point along the vertical line of figure 4.4, which can have a value ranging from -1 to 1. This hinge point is positioned relative to local chord line with 0 value on the chord line. A value of 1 will position the hinge point on the upper curve of the airfoil, while -1 will position the hinge point on the lower curve of the airfoil.

- **deflection_angles** is a list of two deflection angles in degrees for modelling the deflected moveable of a wing. The first angle is used for modelling the deflected moveable of the original wing model, while the second angle is used for modelling the deflection moveable of a mirrored wing model.

- **deflection_limits** is a list of two deflection angles, which represent the maximum deflection angles of a moveable. The maximum deflection angles are used for determining the intersection points of the maximum deflected moveable curves with the local airfoil curves of the moveable sides, see figure 4.5. These intersection points are used to determine the trim points of the top

and bottom airfoil curves of the moveable sides. The trim points are used for dividing the local airfoil curve into a section representing the wing and a section representing the moveable.

- **clearance_chordwise** is the clearance distance in millimetres between the intersection points and the trim points of figure 4.5. The chordwise clearance distance is used to determine the trim points of the top and bottom airfoil curves of the moveable sides. The trim points are used for dividing the local airfoil curve into a section representing the wing and a section representing the moveable.
- **clearance_spanwise** indicates the minimal distance between a maximum deflected moveable and the wing as shown in figure 4.6. This clearance distance is used for determining the cove distance between the moveable and the wing. The cove distance is the clearance distance in millimetres plus the additional distance caused by the deflection of the moveable.

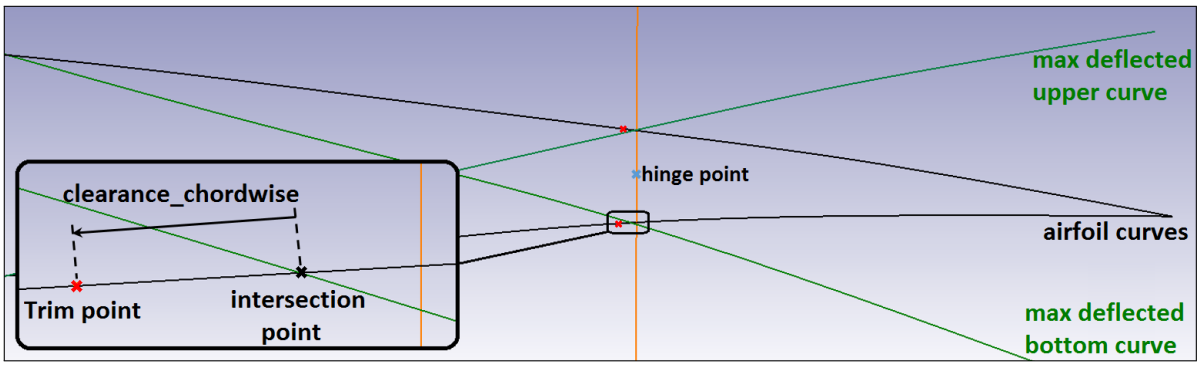


Figure 4.5: Moveable input: $deflection\_limits$ and $clearance\_chordwise$

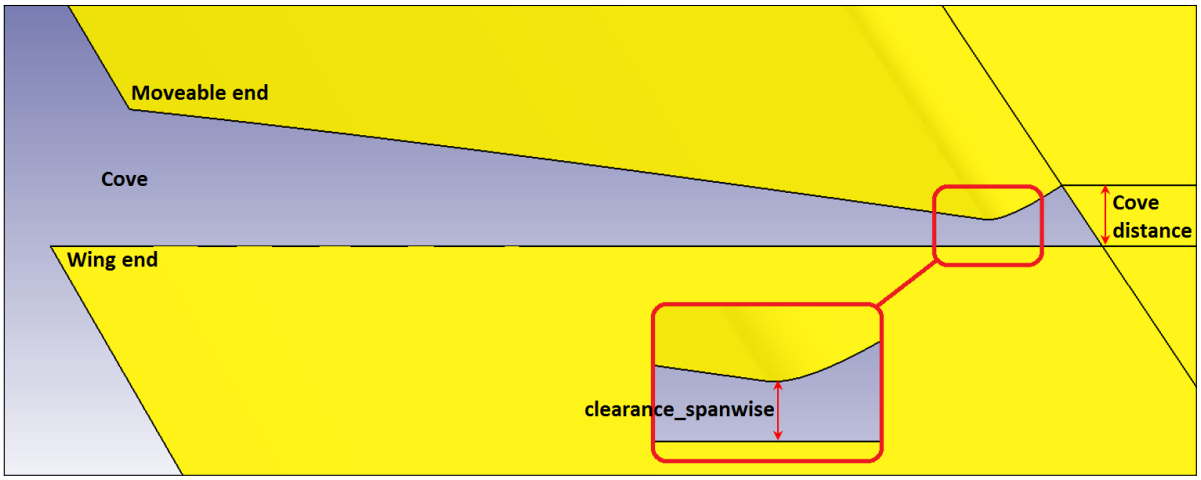


Figure 4.6: Moveable input: $clearance\_spanwise$

4.3. Sizing

The purpose of the moveable sizing module is to produce the chordwise and the spanwise split curves of figure 4.7 (green and red curves). This figure shows the split curves of the 'transition surface' and the 'gap' modelling options. The 'normal rotation' modelling option only has the same split curves as the other options but then without the green chordwise split curves of figure 4.7. These green split curves are used for modelling the coves of the moveable. The chordwise and the spanwise split curves are used for physically modelling the wing with moveable by dividing the surface of the clean wing model into blocks that contain the outline of the moveable. For example, the split curves of figure 4.7 are used to model the wing with moveable of figure 4.8. Figure 4.8 also shows some names that is often used in this chapter.

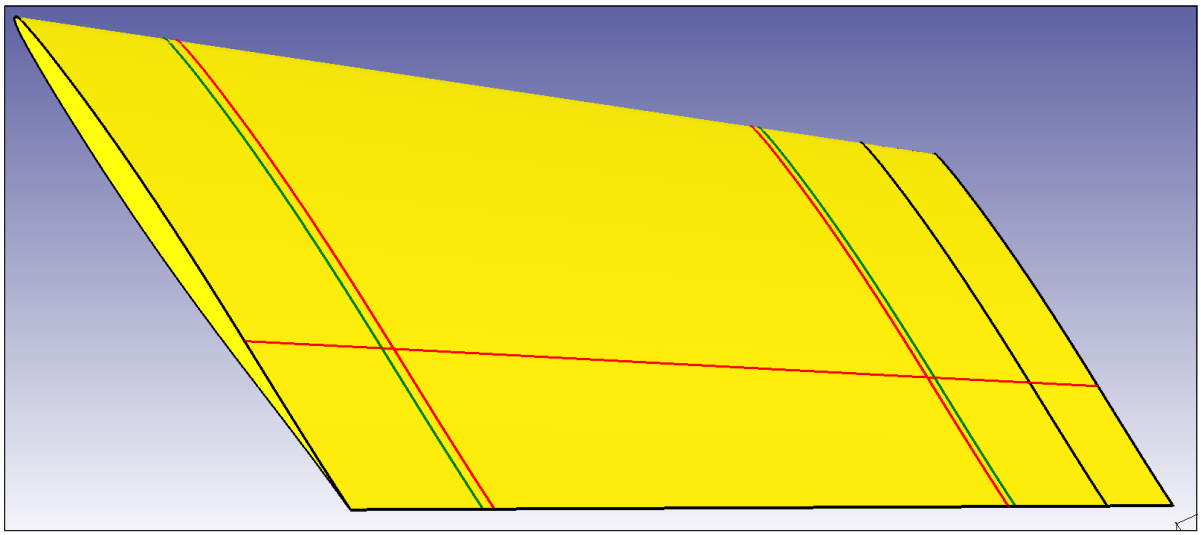


Figure 4.7: Wing model showing the chordwise and the spanwise split curves for moveable sizing ('transition surface' and 'gap' options)

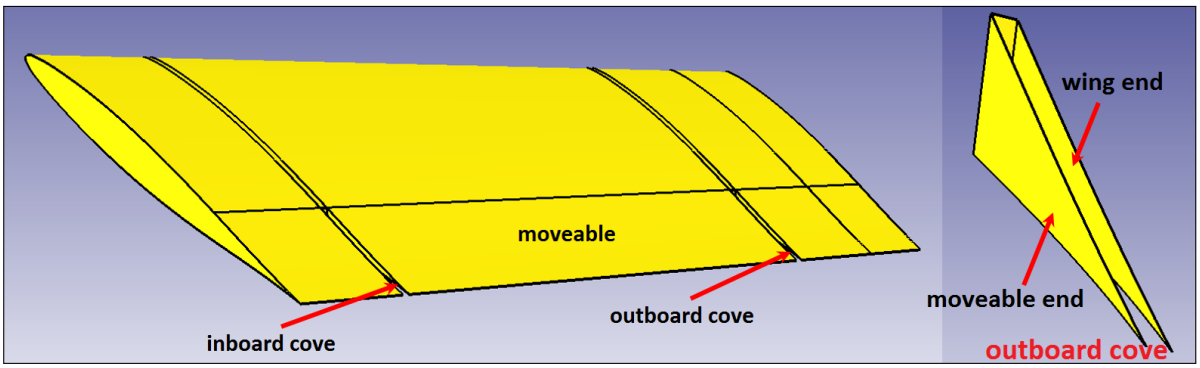


Figure 4.8: Meaning of names for moveable modelling ('gap' modelling option)

The sizing process of the moveable is shown in the activity diagram of figure 4.9. The sizing process of the moveable involves three classes, which are the *CutShapes* class, *MoveableAirfoils* class and *SpanwiseSplitLines* class. The sizing process of the moveable starts with determining the spanwise position and the chordwise position of the hinge points on the moveable ends. The chordwise hinge point (black point) is the position of the hinge point on the local chord line (green line), see figure 4.10. The position of this chordwise point is determined by using the $span\_positions$ and $hinge\_chord\_ratios$ input parameters. After that the moveable side plane is oriented using the $frontal\_orientation$ and $edge\_cant$ input parameters, which is represented as the green square in figure 4.10. This side plane is then used to create a vertical line (green line) that is perpendicular the local chord line and it is embedded in the side plane. The position of the hinge point can finally be determined by moving the chordwise hinge

point along the vertical line of figure 4.10 with the amount that is defined by $hinge\_height\_position$. The airfoil cross-section of the moveable end can then be retrieved by using the green side plane of figure 4.10 as a cut plane to cut/intersect the clean wing model. The intersection of the cut plane and the clean wing model will produce the red airfoil curve of figure 4.10.

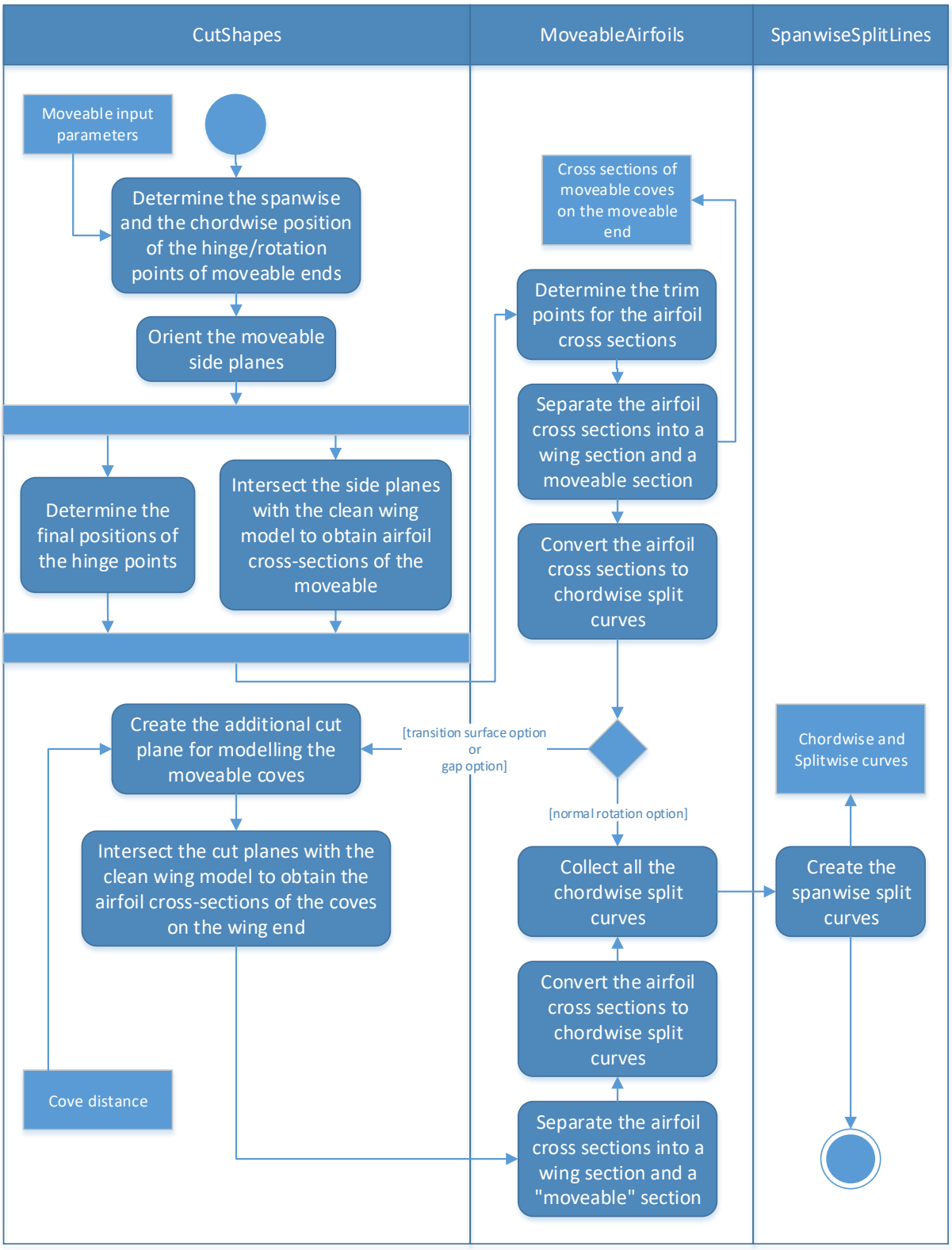


Figure 4.9: UML activity diagram for sizing a moveable

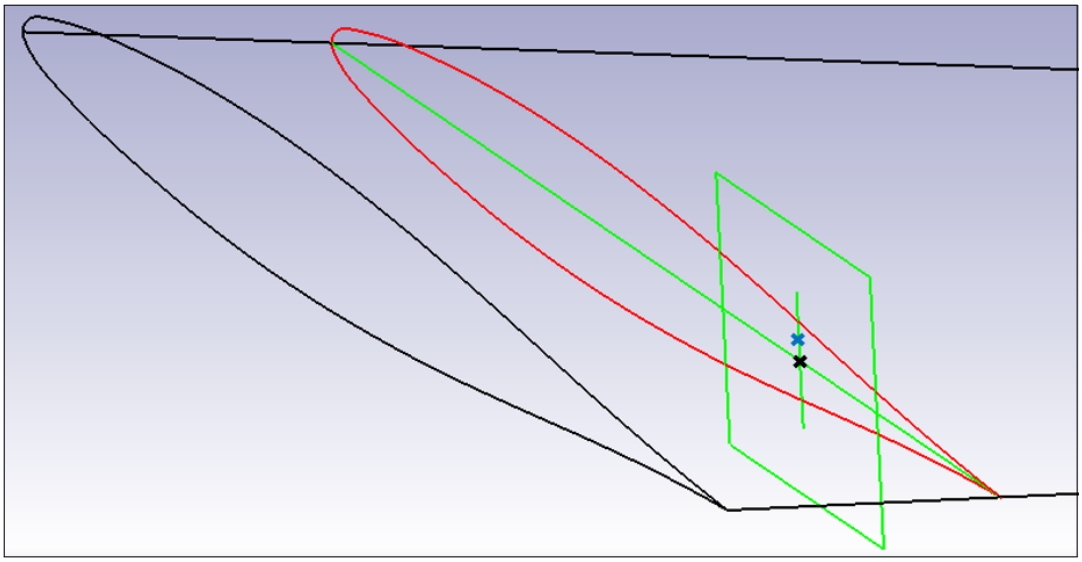


Figure 4.10: Wireframe of wing showing hinge point and moveable curves of the inboard moveable end

The next step in the moveable sizing process is to determine the chordwise length of the moveable by determining the two red trim points of figure 4.5. The derivation of the position of these trim points was explained in section 4.2. The function of these trim points is to trim the chordwise cross-section of the moveable into two sections. The front section will represent the local curve of the wing, while the aft part represents the local curve of the moveable.

The sizing process needs to model the coves between the wing and the moveable if the moveable is modelled according to the 'transition surface' and 'gap' options. A cove can be modelled by intersecting the clean wing model with an additional cut plane shown in figure 4.11. Figure 4.11 shows both the wing end cut-plane (green plane) and the moveable end cut-plane (red plane). The green cut plane has the same orientation as the red cut plane and it is essentially the same plane, but it is repositioned by moving the plane in the normal direction of the moveable cut plane with a distance specified by the cove distance parameter. This cove distance parameter was explained in section 4.2.

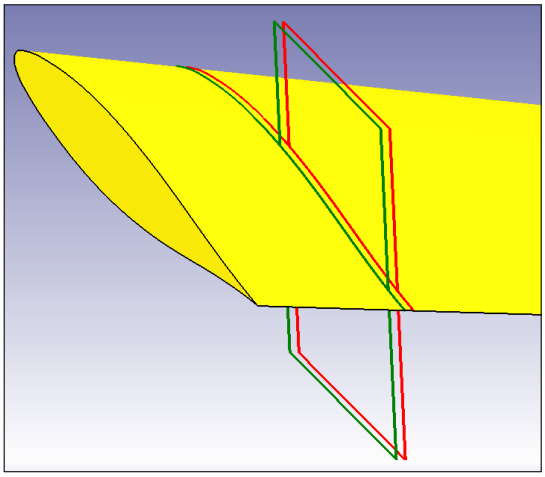


Figure 4.11: Cut-planes of both the wing and moveable side of the inboard cove

The final step of the moveable sizing process is to create the spanwise split curves of figure 4.7. The spanwise split curves of figure 4.7 between the red chordwise split curves are created by connecting the inboard trim points with the outboard trim points of the moveable. The remaining spanwise split curves are created by first creating some additional trim points on the green split curves and on the chordwise edges of the wing solid. Figure 4.12 shows these additional trim points for the inboard side

of the moveables as the blue points. These blue trim points are positioned on the curves such that it has the same chordwise position (x/c) as the trim points of the moveable curves (black points). The remaining spanwise split curves are then created by connecting these blue trim points with each other and with trim points of the moveable.

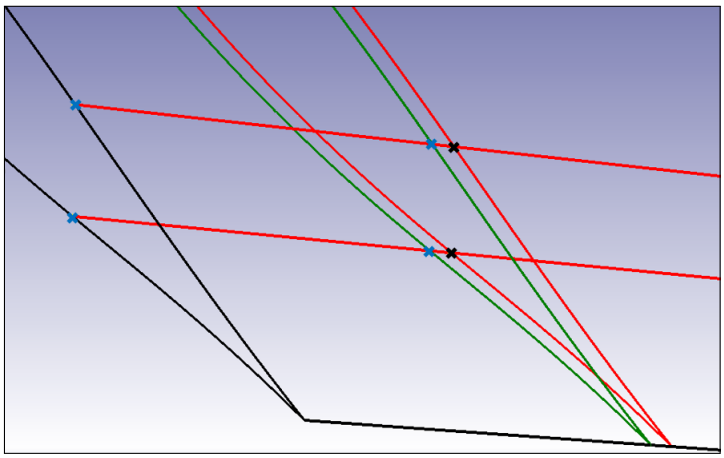


Figure 4.12: Wireframe of wing showing the trim points

4.4. Deflected Moveable Curves Modelling

The deflected moveable models of the 'transition surface' and the 'gap' options are modelled by the *WingSolid* class by creating a new bottom surface and a new top surface for the moveable. These surfaces are created by using the deflected moveable curves that represent the moveable cross sections of the inboard side and the outboard side of the moveable. The modelling of the deflected moveable curves is performed by the *MoveableAirfoils* class and this modelling process is illustrated by the activity diagram of figure 4.13.

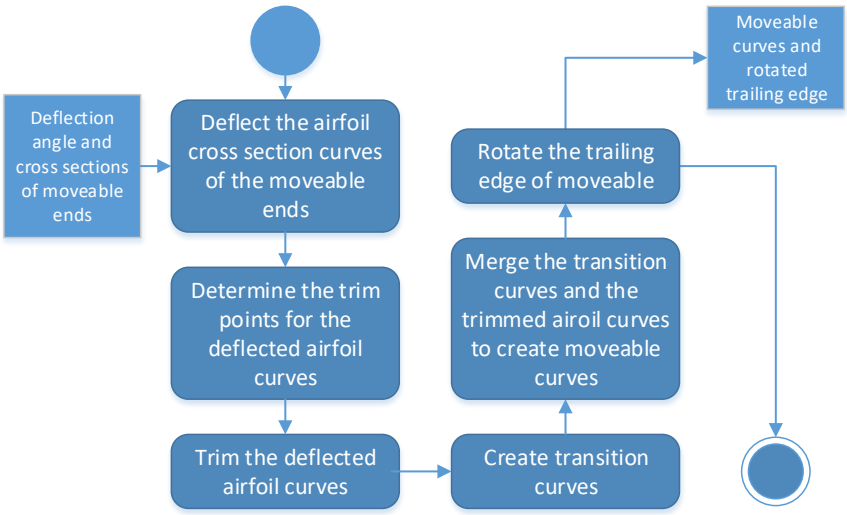


Figure 4.13: UML activity diagram for modelling the deflected moveable curves

The modelling of the deflected moveable curves starts with rotating the airfoil cross sections of the moveable sides. The next step of the modelling process is to determine the trim points for each airfoil cross section of a moveable side. These trim points are the black points in figure 4.14. Figure 4.14 shows the undeflected cross section curves of the moveable as the black lines and the deflected cross sections of moveable as the green lines. The figure also shows two red trim points $w1$ and $w2$. The red trim points are the same red trim points as in figure 4.5, which are used for sizing the moveable.

The deflected moveable curves are created by using the trim points $m2$ and $m3$ to trim the deflected cross-section of the moveable.

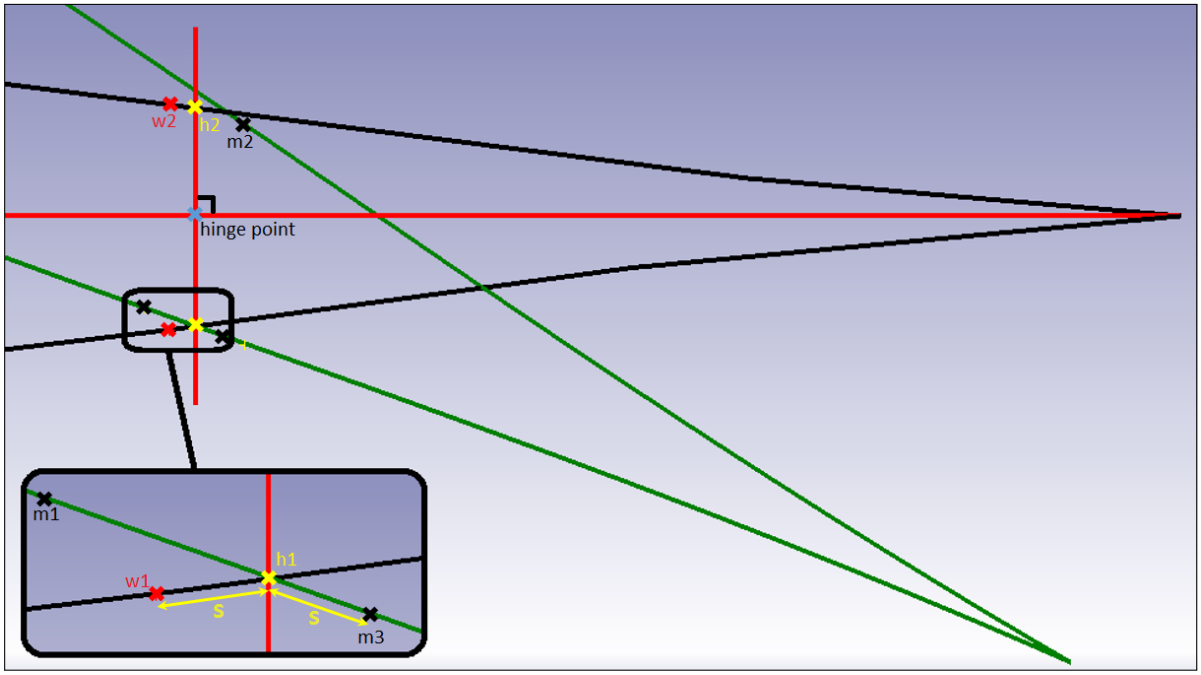


Figure 4.14: Deflected and undeflected airfoil curves with trim points.

The trim point $m2$ is calculated by rotating point $h2$ with the deflection angle around the hinge point. A different method was used to derive the trim point $m3$ since rotating the point $h1$ around the hinge point will result in the point $m1$, which is not usable for creating a transition curve between the red trim point and black trim point. The trim point $m3$ is determined by first calculating the intersection point between the black bottom curve and the green bottom curve, which is almost the same as point $h1$. The next step is to determine the distance s between the intersection point and the trim point $w1$. This distance is then used to move the intersection point along the green bottom line toward the trailing edge to obtain the trim point $m3$.

The final step in creating the deflected moveable curves is to create transition curves to connect the trimmed deflected moveable curve with the trimmed curve of the wing section. Figure 4.15 shows the transition curves as the yellow curve which connects the green moveable curves with the black curves of the wing section. The transition curves are then merged with the trimmed deflected moveable curves to create the deflected moveable curves.

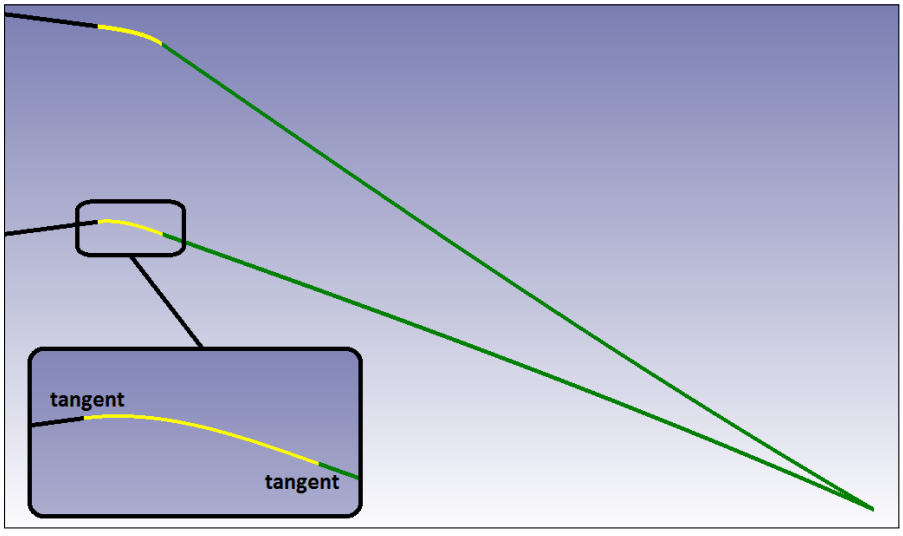


Figure 4.15: Deflected moveable-airfoil curve with transition curve

4.5. Wing with Moveable Model

The physical wing with moveable model is created by the *WingSolid* class and the process to generate this model is shown in the activity diagram of figure 4.16.

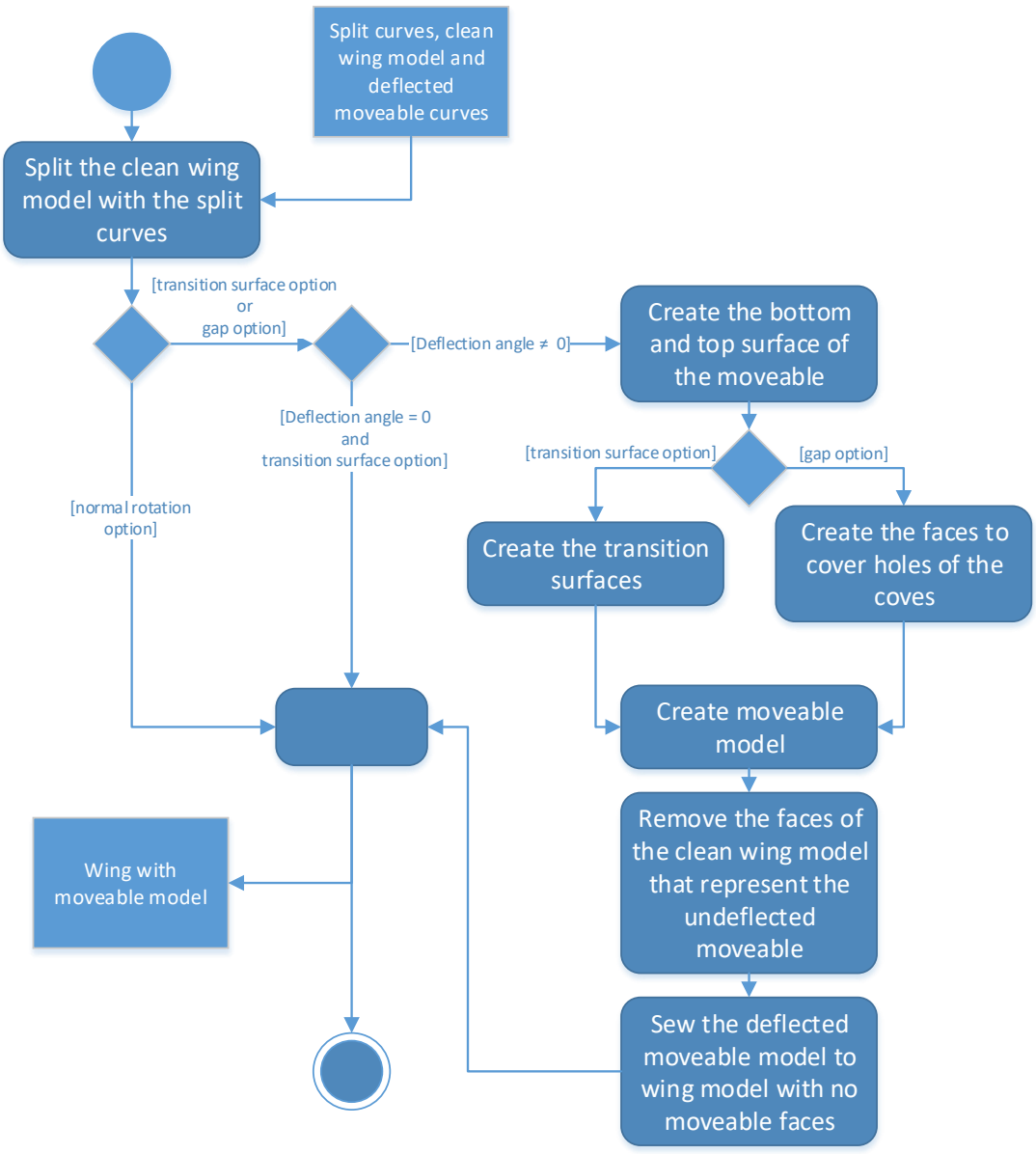


Figure 4.16: UML activity diagram for creating the physical wing with moveable model

The modelling process starts by splitting the clean wing model. The clean wing model is splitted by using the chordwise and the spanwise split curves of the sizing process, see section 4.3. The result of the splitting process is a wing model with an outline of a moveable that has a deflection angle of zero. The modelling process of the wing with moveable model is finished if the moveable is modelled according to the ”normal rotation” option. The wing with moveable model for the ’normal rotation’ option is shown in figure 4.21.

If the deflected moveable is modelled according to the ’transition surface’ and the ’gap’ options then the deflected moveable model needs to be created. The first step of modelling the deflected moveable model is to create the bottom and top surfaces of the moveable, see figure 4.17. These surfaces are created by using the deflected moveable curves of figure 4.15, the spanwise split curves of figure 4.7 and the rotated trailing edge of the moveable as a frame. The next step in creating the deflected moveable model is to add some additional surfaces to the moveable faces of figure 4.17. Both the

'transition surface' and the 'gap' options use the green chordwise split curves and small red spanwise split curves of figure 4.7 to create the additional surfaces.

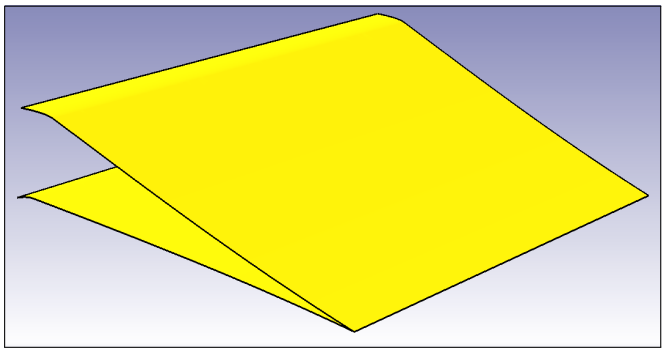


Figure 4.17: Moveable surface

The 'transition surface' option uses these split curves to create four transition faces to close the gap between the moveable and the wing, see figure 4.18. The 'gap' option, on the other hand, uses these split curves to create two quadrilateral faces and four triangular faces to close the gaps of the inboard cove and the outboard cove, see figure 4.19.

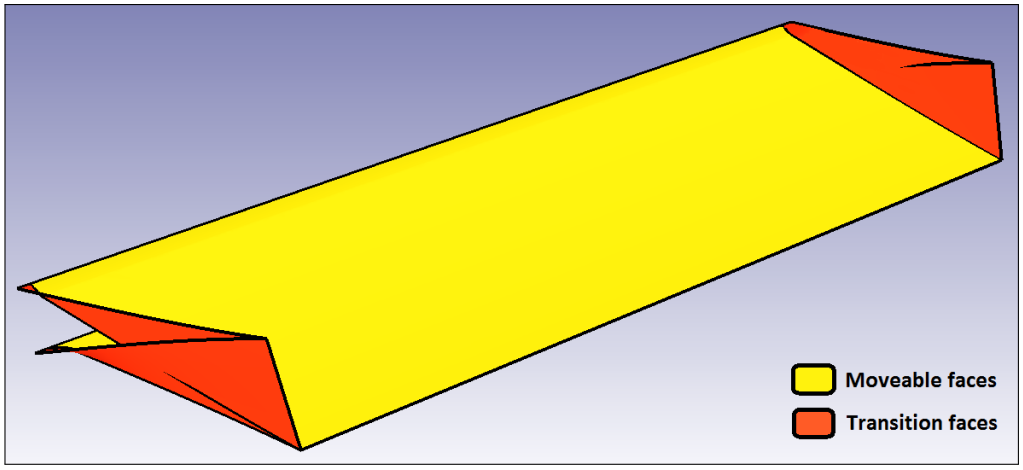


Figure 4.18: Moveable shell for 'transition surface' modelling option

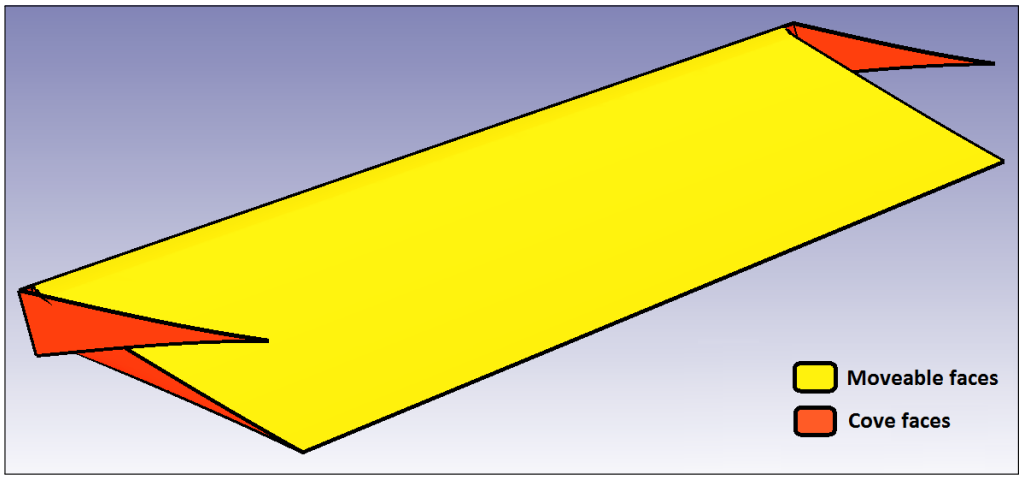


Figure 4.19: Moveable shell for 'gap' modelling option

The next step in creating the wing with moveable deflection is to remove the moveable surfaces of the splitted wing model that contains the moveable with zero deflection. The result of the removal is shown in figure 4.20. The final step is to sew the moveable shells of figure 4.18 and 4.19 to the wing shell of figure 4.20. The results of the sewing is shown in figures 4.22 and 4.23.

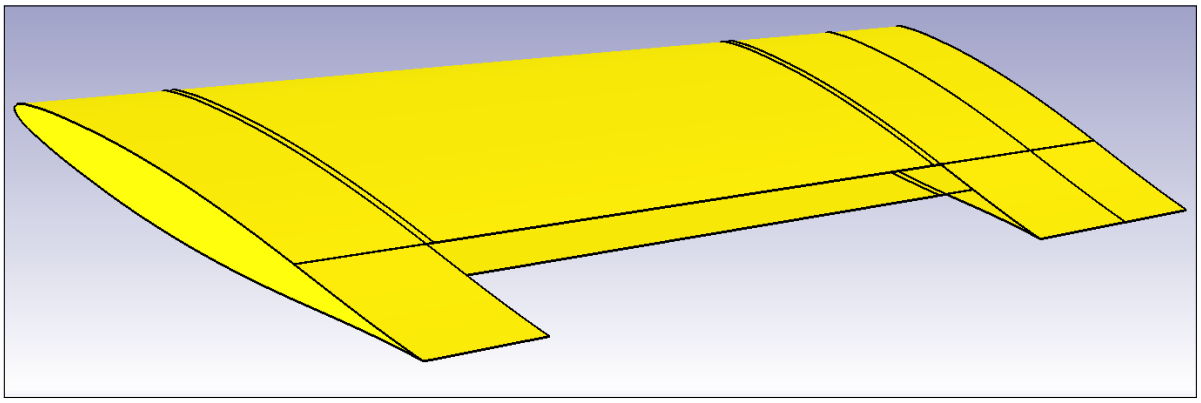


Figure 4.20: Wing shell without moveable

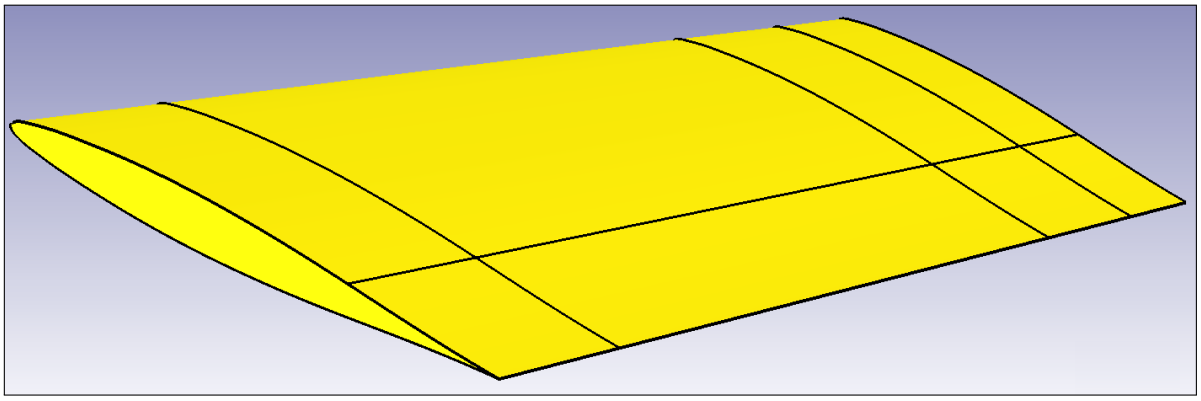


Figure 4.21: Wing with deflected moveable for 'normal rotation' option

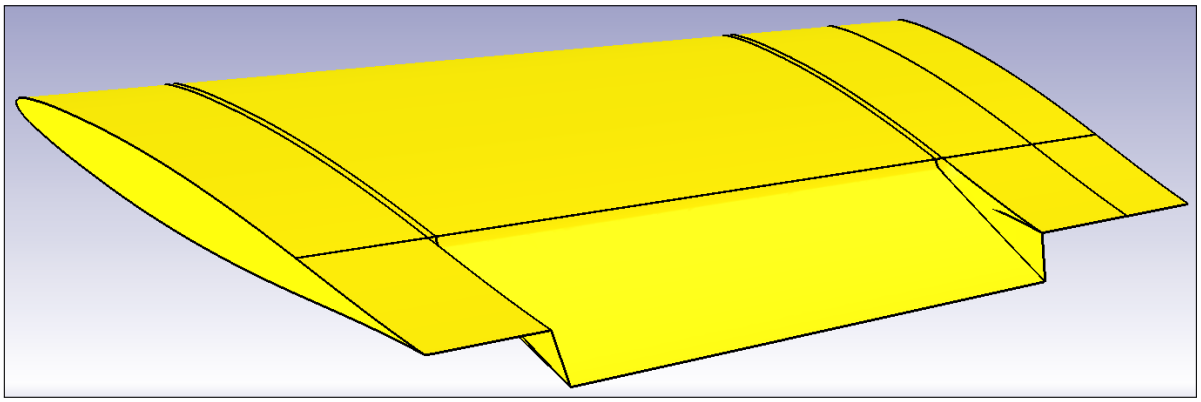


Figure 4.22: Wing with deflected moveable for 'transition surface' option

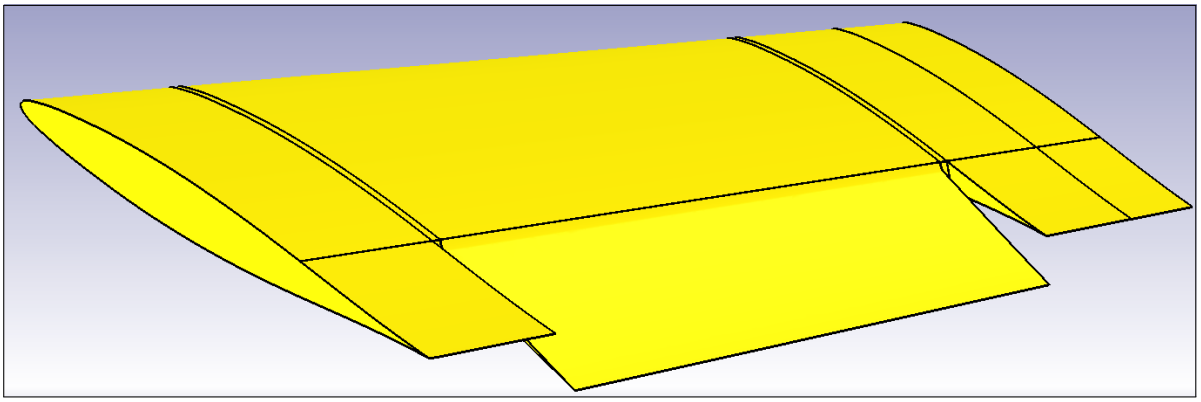


Figure 4.23: Wing with deflected moveable for 'gap' option

5

Mesh Generation

The basic model of the Multi-Model Generator or MMG uses a boundary representation method or BRep to represent the aircraft model. Connecting the MMG to the aerodynamic solver VSAERO requires the discretization of the BRep model. The discretization of the model is necessary since VSAERO is a 3D panel method and it requires a model that consist of squared and/or triangular panels.

The aircraft mesher consist of three modules, which are called the DARwingMesher, the DARfuseMesher and the MMGmesher. DARwingMesher is a wing-moveable mesher that discretizes the wing with moveables models into panels. The DARfuseMesher module on the other meshes the fuselage. The MMGmesher is the mesher that is created for meshing the entire aircraft and this module reuses the subclasses of the DARwingMesher and the DARfuseMesher modules.

The chapter starts by explaining the built-in mesher of ParaPy that has been used for created the mesher.

5.1. ParaPy Built-In Mesher

The aircraft meshing module created during this thesis uses a built-in mesher tool of ParaPy to discretize the BRep model. This built-in mesher tool is a library of high-level, declarative classes in ParaPy that wrap around low-level C++ kernel functionality in Salome [15]. The mesher tool uses a set of meshing algorithms and their corresponding conditions (hypotheses) to compute meshes on the boundaries of the models. Meshing a BRep model requires three inputs and these inputs are:

- **shape_to_mesh** requires the shape that needs to be meshed. This shape can be a solid, a shell, a face or an edge of a model.
- **controls** require inputs that define the rules for the mesh of the input shape. The rules that have been used in this thesis for meshing the models are the edge controls and the face controls. Edge controls define the number of mesh points on the edges, while the face controls define the mesh type of the faces. There are two mesh types and these are structured mesh and unstructured mesh. Besides these rules, there are plenty of other rules for defining the controls of a mesh, such as edge controls that are based on characteristic length or controls based on a certain type of distribution [15].
- **groups** input is used to create a link between the sub-grids of a mesh with the sub-shapes of the input shape. Linking the sub-grids to the sub-shapes allows it the user to retrieve the mesh information of the sub-shapes.

Figure 5.1 shows the meshing capability of the built-in mesher tool that has been used for creating the aircraft mesher in this thesis. The figure shows a BRep of a box in ParaPy. The input for the **shape_to_mesh** parameter is the box shape. The **groups** input is left empty for now. The **controls** input is a list of the following settings:

- Quadrilateral panels on all the faces of the box
- Triangular panels on face3
- 11 equal distanced mesh points on edges e3, e4 and e5

- 11 mesh points with full cosine distribution on edges e1, e2, e6 and e7
- 6 equal distanced mesh points on edges e8 and e9

Using these inputs a hybrid mesh of the box can be created. This mesh is shown in figure 5.2.

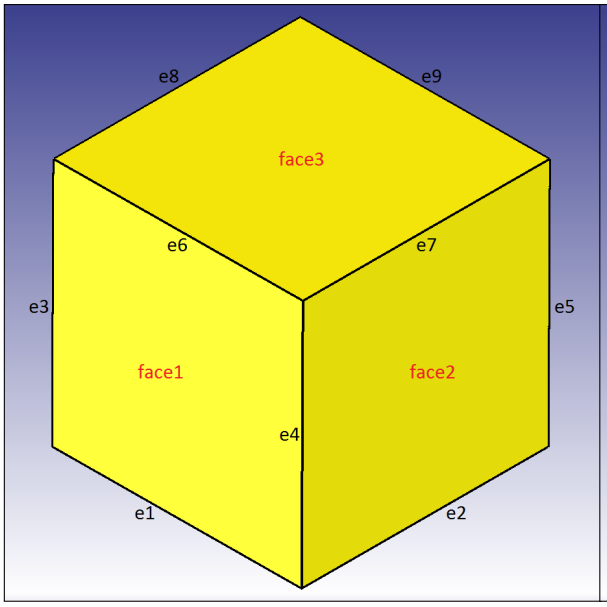


Figure 5.1: BRep of a box in ParaPy

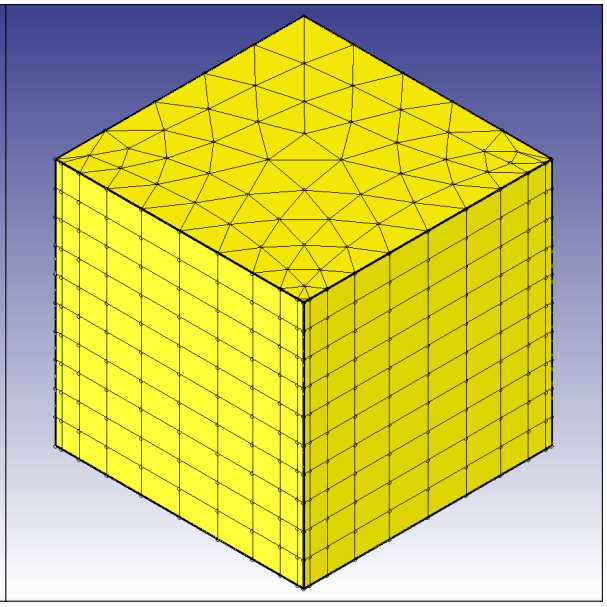


Figure 5.2: Hybrid mesh of a box in ParaPy

The unstructured mesher of this built-in mesher is very versatile and can mesh any BRep models. The structured mesher, on the other hand, has some restrictions on the models. The most important restriction is that the each face with a structured mesh must have four distinct sides. This means that each of these faces must have four edges, where the start and end vectors of each edge are not parallel to start and end vectors of the adjacent face edges.

More than four edges are allowed as long as there are four distinct sides, which means that two or more edges must have C1-continuity. An example of a face that cannot be meshed with quadrilateral panels is shown in figure 5.3. This face has five edges, but none of the edges has C1-continuity. Figure 5.4 show the face with five edges that can be meshed with quadrilateral panels. This face has five edges but the start vector of edge e2 is equal to the end vector of edge e1. The resulting mesh is given in figure 5.5.

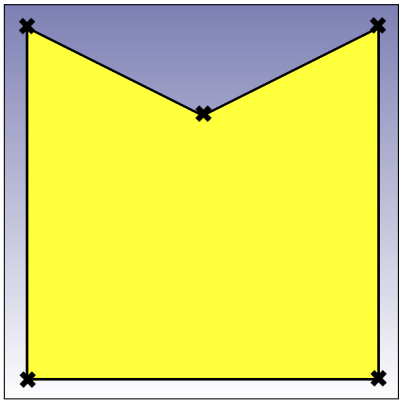


Figure 5.3: Face with 5 edges with no parallel start and end vectors

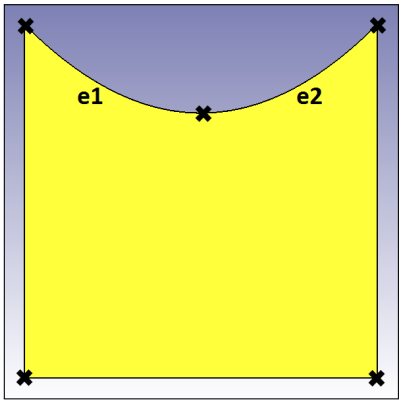


Figure 5.4: Face with 5 edges with one parallel start and end vectors

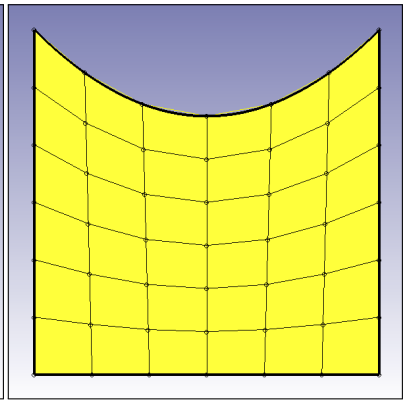


Figure 5.5: Mesh of face with 5 edges with parallel start and end vectors

5.2. Structure of Aircraft Mesher

The aircraft mesher tool and the VSAERO preprocessing tool are created such that it can mesh the whole aircraft or it can be used on the component level such as meshing an isolated wing or an isolated fuselage. To do this, three different classes were created and these classes are the *DARwingMesher*, *DARfuseMesher* and *MMGMesher*. *DARwingMesher* is used for meshing the wing components, while the *DARfuseMesher* is designed for meshing the fuselage component. Both mesher classes are then combined to form the *MMGMesher* class for meshing the entire aircraft.

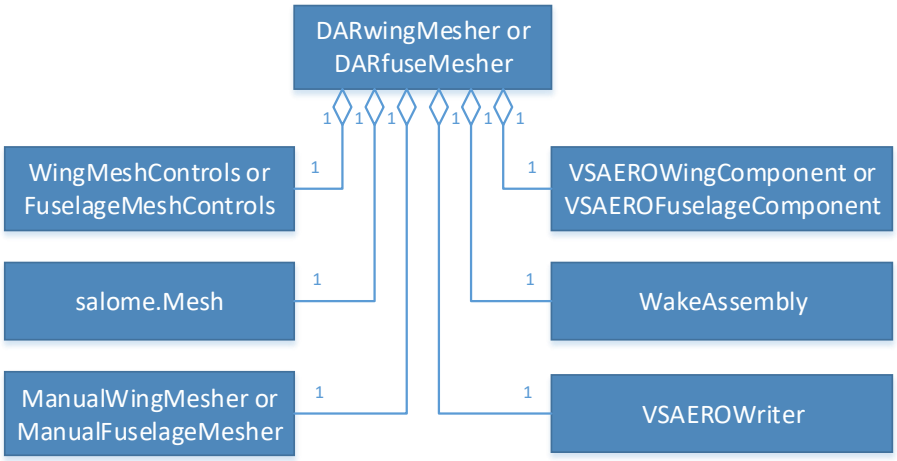


Figure 5.6: UML-diagram of the DARwingMesher class and the DARfuseMesher class

The UML-diagram of the wing and the fuselage meshers is displayed in figure 5.6. The wing and the fuselage meshers are constructed in a similar fashion and has in total of six classes for performing various tasks to model the wing and to process the mesh data for creating the VSAERO input file. The *WingMeshControls*, *salome.Mesh* and *ManualWingMesher* classes are responsible for discretizing the wing component, while the *FuselageMeshControls*, *salome.Mesh* and *ManualFuselageMesher* classes are responsible for discretizing the fuselage component. The *VSAEROWingComponent*, *VSAERO-FuselageComponent*, *WakeAssembly* and *VSAEROWriter* classes are responsible for processing the mesh data and to create the input file for the aerodynamic solver VSAERO.

The meshing of the component starts with the *WingMeshControls* and the *FuselageMeshControls* classes, which are responsible for defining the rules for the *salome.Mesh* class. The *salome.Mesh* class uses the meshing rules to create the mesh of the wing or the fuselage by using the built-in mesher tool. The *salome.Mesh* class is not capable of meshing all the faces of the wing or fuselage component. These faces are the root, tip and cove faces of the wing component. The nose and the tail faces of the fuselage component can also not be meshed by the built-in mesher tool. The root, tip, cove, nose and tail faces are ”manually” meshed by the *ManualWingMesher* class or the ManualFuselageMesher class.

After the meshing phase, the mesh data needs to be processed so that it can be used for creating the input file. The processing starts with the *VSAEROWingComponent* or the *VSAEROFuselageCompo-nent* class, which divides the component into patches and arranges the mesh data of the *salome.Mesh* class. The next step in creating the VSAERO input file is to model the wake of the component, which is the responsibility of the *WakeAssembly* class. The last class in the chain is the *VSAEROWriter*. The function of this class is to write the input file for VSAERO by using the preprocessed data of the *VSAEROWingComponent* or *VSAEROFuselageComponent* class and the *WakeAssembly* class.

The aircraft mesher has a similar design as the wing and fuselage meshers and this similarity can be observed in the UML-diagram of figure 5.7. The UML-diagram shows the aircraft mesher and it combines the classes of both the wing mesher and the fuselage mesher for meshing the aircraft and for creating the input file. The aircraft mesher has an additional class that is not present in the wing and fuselage meshers. This class is the *Patcher* class if the mesh of two components does line up with each other at the intersection between the two components. If the mesh of two components does not line up with each other at the intersection between the two components then the*IncongruentPatcher* class is used. The function of the *Patcher* and the *IncongruentPatcher* classes is to divide faces of the

aircraft model into blocks that can be meshed by the built-in mesher tool for meshing the aircraft.

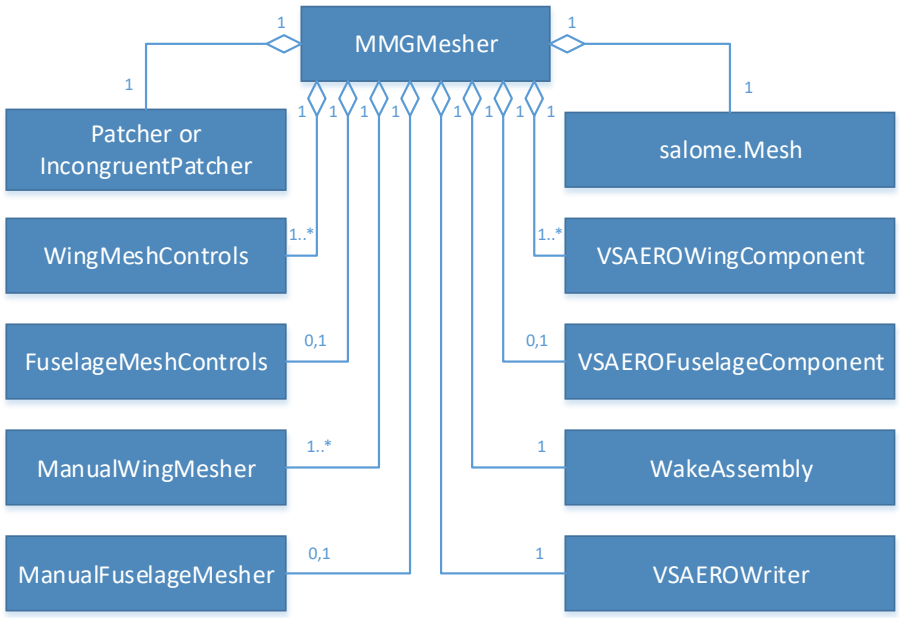


Figure 5.7: UML-diagram of the MMGMesher class

5.3. Blocking Strategy

Figure 5.8 shows an aircraft model that is generated by the MMG. This model can be meshed if the fuselage and the vertical tail are meshed with triangular panels since these components contain faces that have more than four distinct sides. This means that a large part of the aircraft will have an unstructured mesh. A fully unstructured mesh for the fuselage and the vertical tail are undesirable since VSAERO requires a least a stroke of quadrilateral panels at the tail of the fuselage and at the trailing edge of a wing for attaching the wakes on the aircraft model. Also, unstructured grid calculation usually requires more computation time than a structured grid. The strategy to divide the aircraft model into blocks will be explained in section 5.3.1. Section 5.3.2 will show the creation of the split curves to divide the aircraft model into blocks.

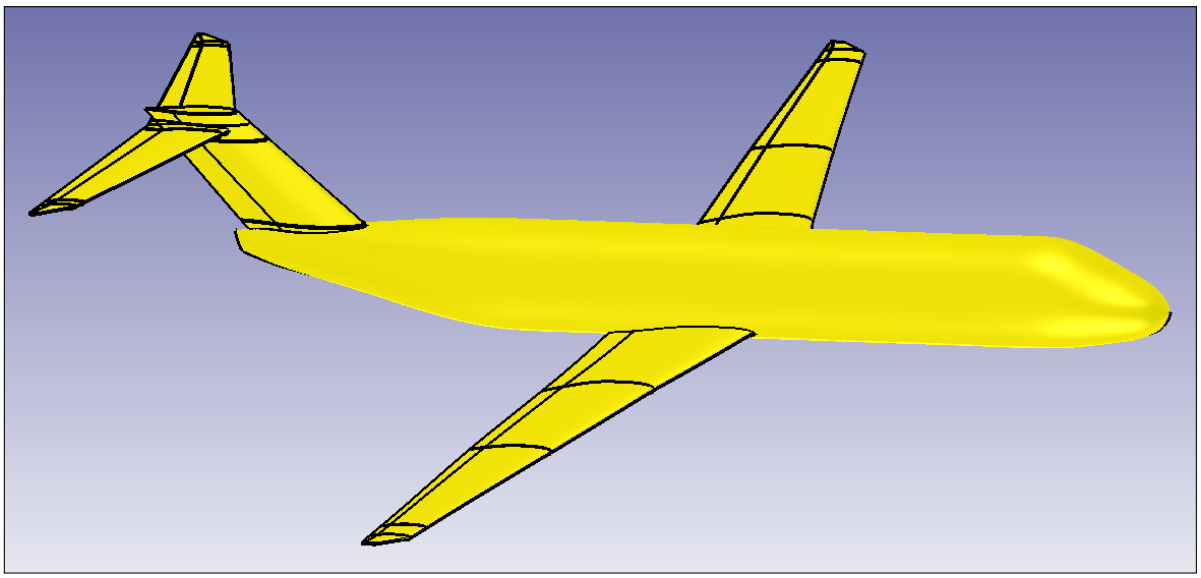


Figure 5.8: Aircraft model before applying blocking strategy

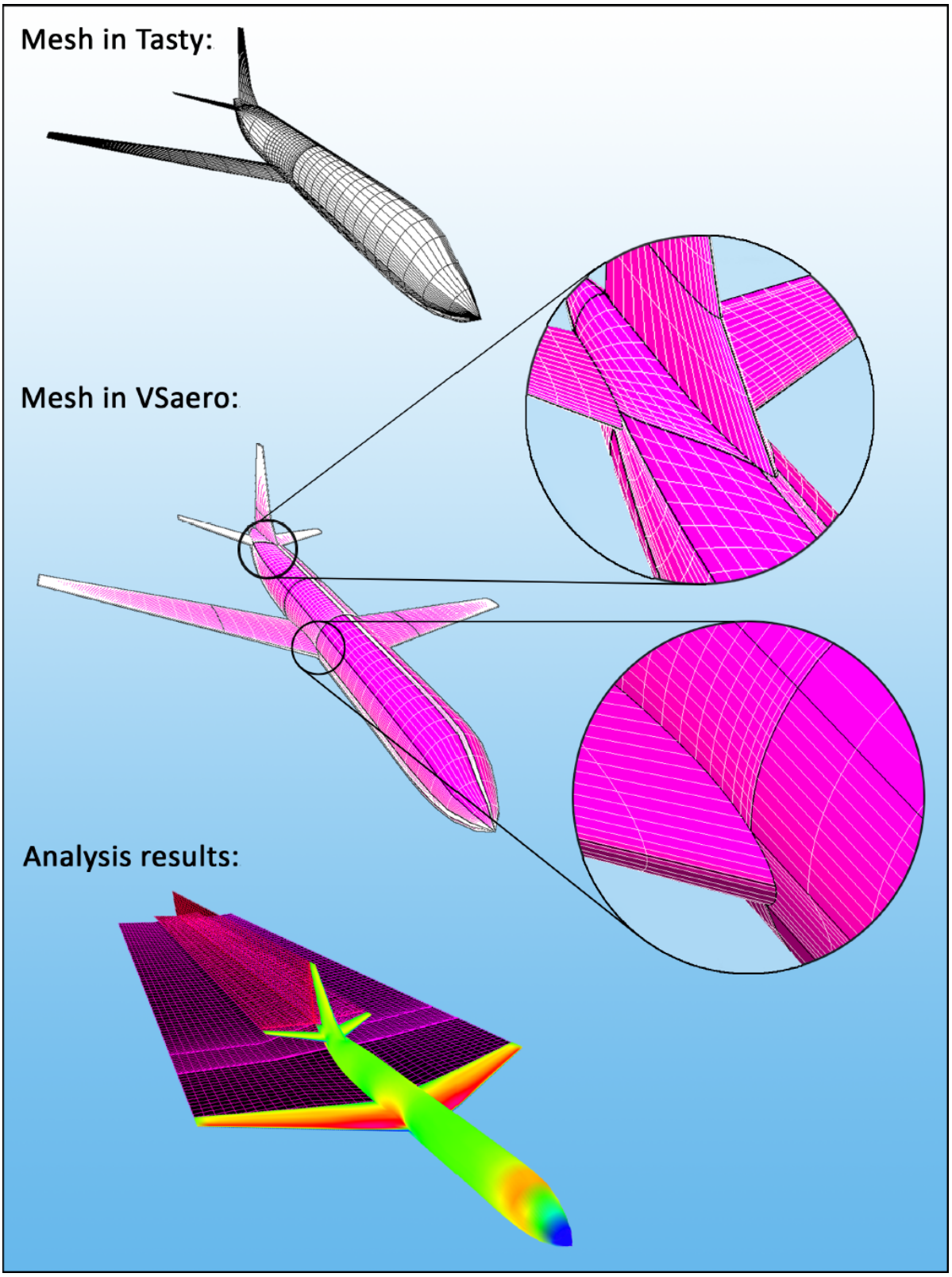


Figure 5.9: Mesh and analysis result of the work of Abedian for a conventional aircraft [14]

5.3.1. Strategy

Attaching a wake to the trailing edge of a component requires at least a stroke of quadrilateral panels. This problem of attaching the wake to the component can be solved if the mesh of the aircraft model is completely structured with quadrilateral panels. Abedian [14] has created a blocking strategy that divides the whole aircraft model into blocks. The result of this blocking strategy is shown in figure 5.9 for a conventional aircraft. The blocking strategy of Abedian to create a fully structured mesh for the aircraft model is not reused in this thesis. Instead, a blocking strategy for creating a hybrid mesh is used.

The first reason for not using the blocking strategy of Abedian is that his strategy will create a panel distribution for the fuselage that has many different sizes. The different sizes of the panels are caused by the dense distribution of mesh points around the leading edge of the wing. This dense distribution is also needed for the fuselage mesh around the wing. This uneven distribution of the panels is not effectively using the panels that are placed too close each other. The second reason for not using the blocking strategy of Abedian is that it is more complex and requires a lot more steps to create a fully structured mesh than the blocking strategy for creating a hybrid mesh. The complexity originates from the fact that a fully structured approach require more split curves for splitting the fuselage than the hybrid approach.

The aircraft mesher of the new MMG uses the blocking strategy that creates a hybrid mesh for the fuselage component. This blocking strategy also creates a hybrid mesh for the wing component that has been intersected by another wing component. This blocking strategy for the fuselage component is shown in figure 5.10. This figure shows that an unstructured mesh will be created around the holes of the main wing and an unstructured mesh will be created around the hole(s) of the tail of the aircraft. The propagation of the dense mesh distribution around the leading edge of the wing to the rest of fuselage is thus prevented by creating triangular panels around the holes. This blocking strategy is thus very simple and requires the fuselage to be divided into five sections by only using the split curves that are oriented in the width direction of the fuselage.

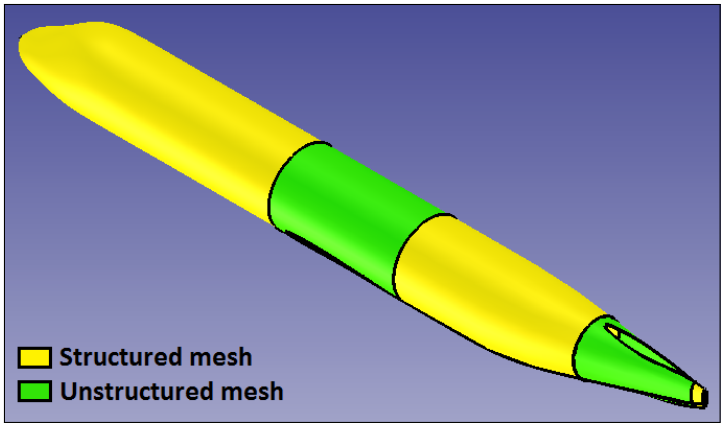
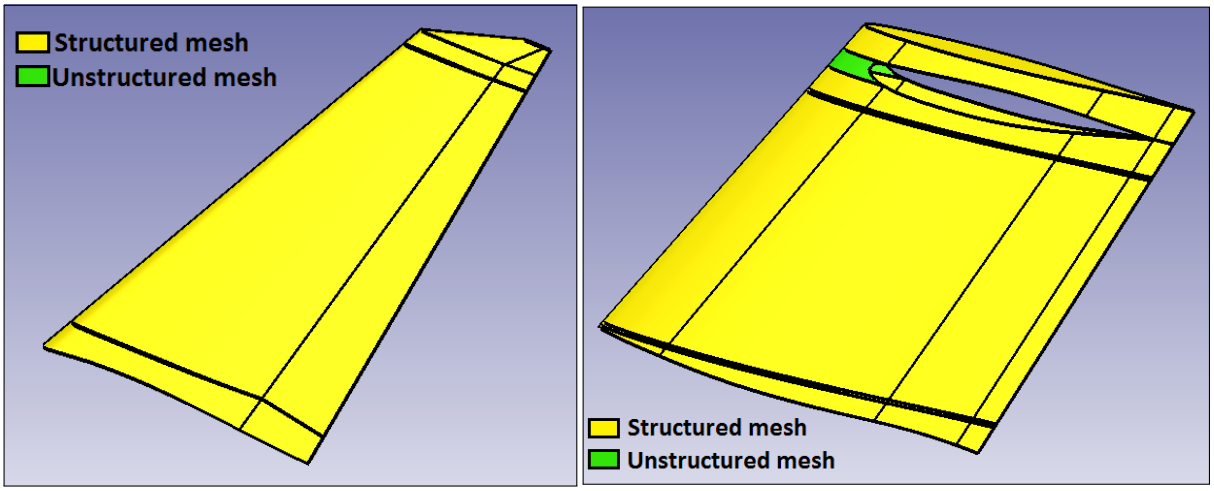


Figure 5.10: Hybrid mesh for fuselage component

Figures 5.11a and 5.11b shows the blocking strategy of the wing components with moveables. Figure 5.11a shows a wing component with no holes and this wing component is fully meshed with quadrilateral panels. Figure 5.11b shows the blocking strategy of the wing component with holes. The wing component with holes is blocked such that the most of the faces of the wing are meshed with quadrilateral panels. Only the faces in front of the leading edge of the wing holes are meshed with triangular panels. The reason for creating an unstructured mesh on these faces is that a structured mesh on these faces will have a curved panel distribution. This curved panel distribution is caused by the curvature of the leading edge of the wing hole. The reason for not applying the same blocking strategy as the fuselage component is that the blocking strategy of the fuselage component will create an unstructured mesh over the most part of the wing surface. The panel distribution in the chordwise direction cannot be controlled for the faces that are meshed with triangular panels.

An additional blocking strategy has been implemented in the aircraft mesher tool. This additional strategy removes all the holes of the components and meshes all the components of the aircraft separately. The advantage of this strategy is it does not require any split curves to create faces that are

suitable for creating structured meshes. It only needs split curves for stitching the wake lines to other components of the aircraft. This strategy will create an aircraft mesh, where the mesh of the wing does not line up with the mesh of the fuselage. This mesh will be referred as incongruent mesh in this thesis.



(a) Structured mesh for wing component without holes (b) Hybrid mesh for wing component with holes

Figure 5.11: Blocking strategy for wing component

5.3.2. Split Curves

The split curves are the curves created to split the faces of an aircraft component into blocks so that it can be meshed by the built-in mesher tool with quadrilateral panels. Also, some split curves are created for attaching root wake lines of the wing components such as on the fuselage to attach root wake line of the main wing.

Fuselage split curves

The blocking strategy of the fuselage component divides the fuselage into five sections as shown in figure 5.10. The fuselage component is divided by using the lateral split curves of figure 5.12 to split the faces of the fuselage. The sections containing the holes of the wings will be meshed with triangular panels, while the remaining the sections will be meshed with quadrilateral panels. Each section, which has holes in it, requires two lateral split curves to create the section. A conventional aircraft will have in total four lateral split curves. Two split curves for the section that has the holes of the main wing and two split curves for the section that has the hole(s) of the tail of the aircraft, see figure 5.12.

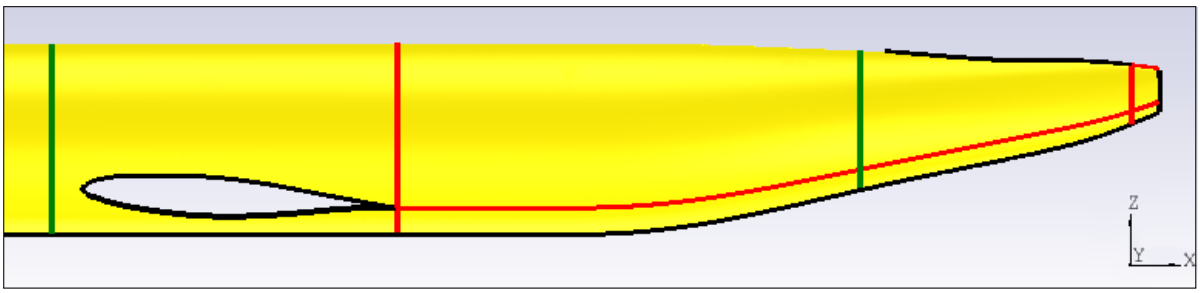


Figure 5.12: Longitudinal and Lateral split curves for congruent mesh

The lateral split curves are closed curves and these curves are created by using the isolines of the fuselage face. The x-coordinates of the split curves are determined by first collecting the root leading edge points and the root trailing edges points of all the wing holes of a fuselage section. The x-coordinate of the trailing edge split curve is determined by the largest x-coordinate of all the root trailing edge points. The x-coordinate of the leading edge split curve, however, is computed by first determining the smallest

x-coordinate of all the root leading edge points. The next step is to determine the offset distance that will be subtracted from the smallest x-coordinate of the leading edge points. This offset distance is computed by taking the smallest chord length of the wing holes and multiply it with **fraction_cut_plane** input parameter. The x-coordinate of the leading edge split curve is then computed by subtracting the offset distance from the smallest x-coordinate of the leading edge points.

The longitudinal split curves are created for stitching the root wake lines of the wing components. Figure 5.12 shows these curves as red lines and these curves start at trailing edge of the wing component and stops at trailing edge of the fuselage. Each wing component that is attached to the fuselage has its own longitudinal split curve for stitching its root wake line. The longitudinal split curves are created by using the isolines of the fuselage face.

The second blocking strategy that meshes all the aircraft components separately also requires longitudinal split curves for stitching the root wake lines of the wing components. These longitudinal split curves are created by following the same creation procedure as the longitudinal split for the hybrid mesh strategy. The difference is that the longitudinal split curves are extended. Instead of starting at the trailing edge points of the wing components, the longitudinal split curves will start at the nose of the fuselage, see figure 5.13. Moving the starting points of the longitudinal split curves to the nose of the fuselage will create split curves that span the entire length of the fuselage. These split curves can then be used to divide the fuselage face into faces that have four distinct sides. These splitted faces with four distinct sides can be meshed by the built-in mesher tool.

Figure 5.13: Longitudinal split curves for wake stitching for incongruent mesh

Wing split curves

The split curves for a wing component will only be created if the wing component has a hole that is created by an intersection with another wing components. This means that the wing component of figure 5.11a does not have any split curves, while the wing of figure 5.11b does contain split curves. The split curves of the wing of figure 5.11b is shown in figure 5.14 as yellow curves. The red points represent the root mesh points of a wing component that is attached to this wing. Three sets of spanwise and chordwise split curves are thus required for splitting one side of the wing surface. Two sets of the split curves are located around the leading edge of the wing hole, while one set of split curves are located on the trailing edge of wing hole.

The chordwise and the spanwise split curves at the leading edge are created such that it always starts at one of the red mesh points of figure 5.14. The starting mesh points of the leading edge split curves are determined by the **no_split_point** input settings. The **no_split_point** input parameter gives a number, which says that nth mesh point from the leading edge point shall be chosen for creating the chordwise and spanwise split curves at the leading edge. On the other hand, the start point for the trailing edge split curves is always fixed and this start point is the trailing edge mesh point of the wing hole.

The spanwise split curves of figure 5.14 are created such that it always follows the spanwise isoline of the wing faces. The spanwise split curves start at one of the mesh points of the wing hole. The inboard split curves will end at the root of the wing component, while the outboard split curves will end at the tip of the wing component.

The chordwise split curves of figure 5.14 start at one of the mesh points of the wing hole. The leading edge split curves will end at a point on the leading edge of the wing, while the trailing edge split curves will end at a point on the trailing edge of the wing. The end point of each chordwise split curve must coincide with the end point of a similar split curve on the opposite side of the wing. The end points of the two similar split curves are synchronised by averaging the chordwise isolines that go through the start points of the split curves.

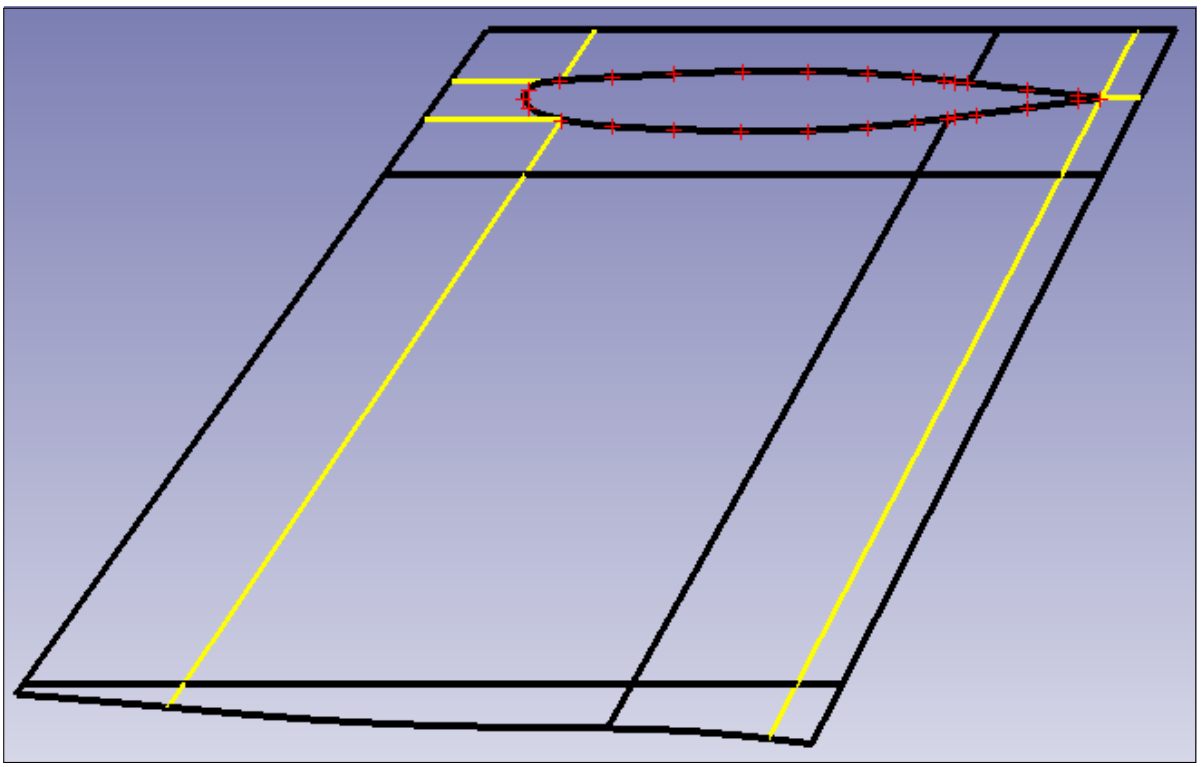


Figure 5.14: Chordwise and spanwise split curves for a congruent mesh

The second blocking strategy that meshes all the aircraft components separately also requires chordwise split curves for stitching the root wake lines of the wing components that are attached to this wing. The same wing component of figure 5.14 is shown in figure 5.15 but without the wing hole of the other wing component. Figure 5.15 shows a red point, which represents the root trailing edge point of figure 5.14 for the wing component that is attached to this side of the wing. The red trailing edge point will be used to create a chordwise split curve that spans the entire chordwise length of the wing surface and this curve follows the chordwise isolines of the wing surface. This split curve spans the entire chordwise

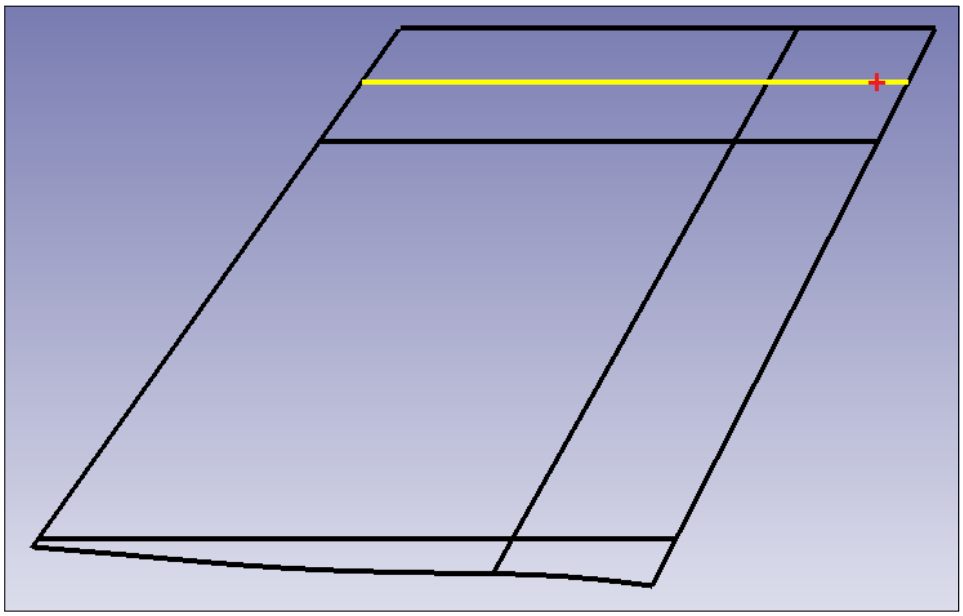


Figure 5.15: Chordwise split curves for wake stitching for incongruent mesh

length because this split curve is used to divide some of the wing faces into several faces that have four distinct sides. These splitted faces with four distinct sides can be meshed by the built-in mesher tool. The chordwise split curve will be used for stitching the root wake line of the wing component that is attached to this wing component.

5.4. Controls

The *WingMeshControls* and *FuselageMeshControls* classes define the rules for creating the mesh of the aircraft model. The *WingMeshControls* is used for defining the rules of the wing component, while the *FuselageMeshControls* is used for defining the rules of the fuselage component. Those rules are, for example, to determine the number of mesh points and its distribution on all the edges of the wing component or the fuselage component. Another rule is to determine the faces that are meshed with quadrilateral panels and the faces that are meshed with triangular panels.

5.4.1. Wing Controls

The wing with moveable model of figure 5.11a has a fully structured mesh and it has no holes on its surface. The first meshing rule for this wing model is that all the faces should meshed with quadrilateral panels. A structured mesh requires that the opposite edges of the mesh faces to have the same number of mesh points. The same number of mesh points on the opposite edge are required to create a mesh with only quadrilateral panels. It is thus very important to find the chains of opposite edges in both the chordwise and the spanwise directions of the wing. Figure 5.16 shows the edge chains for the wing with moveable model of figure 5.11a. The top image shows all edges of wing model, while the middle image shows the edge chains of the model that contains the chordwise edges of the model. The bottom

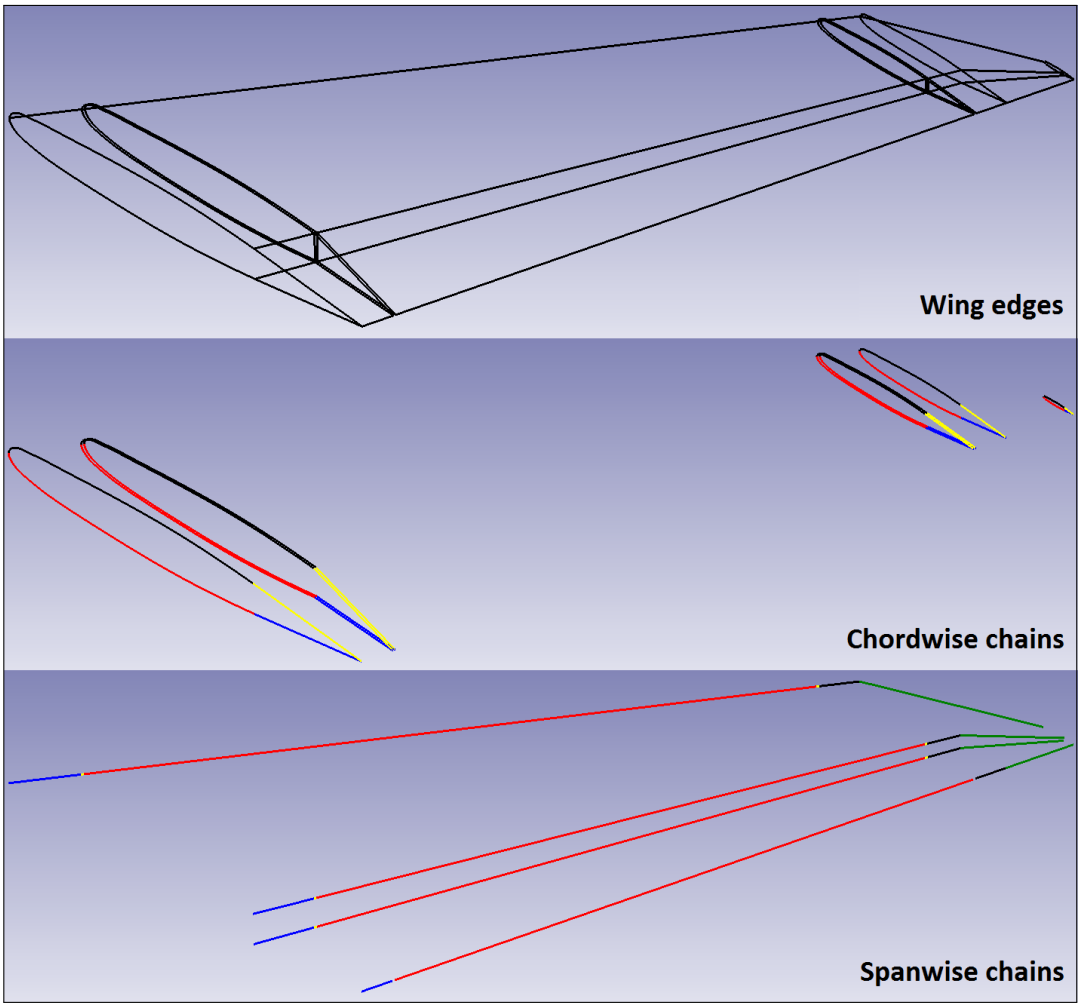


Figure 5.16: Chordwise and spanwise edge chains of a wing with moveable model

image shows the edge chains of the model that contains the spanwise edges of the model. Each chain in figure 5.16 is represented by a different colour.

The chordwise and spanwise chains of figure 5.16 are used to create rules that define the number of mesh points and its distribution on the edges of the wing model. The mesh points distribution of each chordwise chain are distributed according to the full-cosine distribution. A full-cosine distribution is desired since a full-cosine distribution will place more mesh points at the start and at the end of the chordwise edges. The start and the end part of the chordwise edges represent most of the time the trailing edge or the leading edge or the deflected curvature of a moveable. The mesh points of the spanwise chains are distributed evenly along the length of the edges. The number of mesh points on the edge chains is determined by the follow input parameters.

- **no_chordwise_pts** determines the number of mesh points on the chordwise edge chains that do not contain the moveable edges.
- **no_chordwise_pts_aft** determines the number of mesh points on the chordwise edge chains that do contain the moveable edges.
- **spanwise_pitch** specifies the distance in millimetres between the consecutive mesh points of the spanwise edge chains. The length of the longest edge of each chain is used for determining the number of mesh points.
- **spanwise_pitch_trunks** is a list of **spanwise_pitch** for each wing trunk. These pitch values will override the global **spanwise_pitch** value

A wing with holes on its surface uses a different method for determining the number of mesh points and for determining the distribution of mesh points on the edge chains. Figure 5.17 shows the chordwise and spanwise chains of a wing surface with a hole. The left image shows all the edges of the wing component, while the middle image shows the chordwise chains. The right image shows the spanwise chains of the wing with moveable component. The intersection/hole edges impose a requirement on the number of mesh points on the chordwise chains. To be able to create a structured mesh, the number of mesh points on the chordwise chains must have the same number as the number mesh points on the intersection edges. The mesh points on the intersection edges represent the root mesh points of another wing component that is attached to this wing component. So the number of mesh points and the distribution of the mesh points for the red, blue, orange and green chordwise chains of figure 5.17 are derived from the intersection edges.

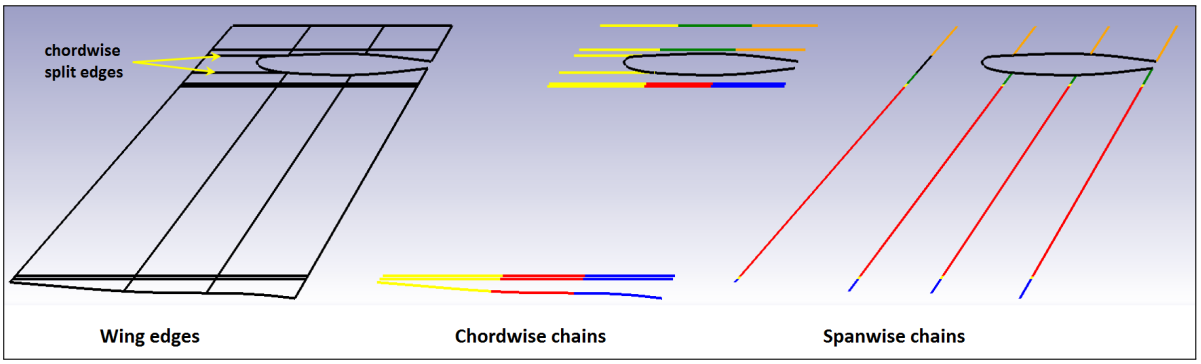


Figure 5.17: Chordwise and spanwise edge chains of a wing with a moveable and has holes on the wing surface

The yellow chordwise chain of figure 5.17 is not dependent on the intersection curves since the start and the end edges of this chain are not located on a wing face that has the intersection edges. Two of the chordwise edges in this chain are located on a face that contains the intersection edges, but the intersection edges are not located on the opposite side of the chordwise edges. The yellow chordwise chain of figure 5.17 has thus the freedom to determine the number of mesh points on the edges of the chain. The number of mesh points on the edges of this chain is determined by the **chordwise_pitch** input parameter. The **chordwise_pitch** input parameter specifies the distance in millimetres between two consecutive mesh point of the longest edge in the chordwise chain.

The determination of the number of mesh points on the spanwise edges chains is the same as for the wing with no holes, except for spanwise chain/edge shown in black, see figure 5.17. The number

of mesh point on this chain/edge is determined by the **spanwise_pitch_tri** input parameter. This edge gets a different input parameter because the face that is enclosed by this edge and chordwise split edges will be meshed with triangular panels. This enclosed face is shown in figure 5.11b as the green face of the wing. A smaller spanwise pitch for the black spanwise chain/edge will produce a more consistent unstructured mesh than using the same spanwise pitch as for the other spanwise chains. The max size of the triangular panels for the unstructured mesh is determined by the **tri_max_pitch** input parameter. This parameter limits the sizes of the triangular panels of the unstructured mesh and an increase in the value of this parameters causes the triangular panels of the unstructured mesh to become larger. A large panel means fewer panels and fewer panels means the curvature of the wing surface is modelled less accurately.

Another job of the *WingMeshControls* class is to create the rules that link the sub-grids of the aircraft mesh with the sub-shapes of the aircraft model. Linking the sub-grids to the sub-shapes is very useful for retrieving the mesh data of the sub-shapes. These rules are stored in the parameters that contain the word groups. These ”groups” are:

- *groups_face_pairs* links the bottom and top faces of the wing to the sub-grids. These groups are used for creating the bottom and top wing patches of VSAERO.
- *groups_root_edges* links the root edges to the sub-grids. These groups are used for creating the mesh of the root face ”manually”.
- *groups_tip_edges* links the tip edges to the sub-grids. These groups are used for creating the mesh of the tip face ”manually”.
- *groups_cove_edges* links the cove edges to the sub-grids. These groups are used for creating the mesh of the cove faces ”manually”.
- *groups_gap_edges* links the gap edges to the sub-grids. These groups are used for creating the mesh of the gap faces ”manually”.

5.4.2. Fuselage Controls

The mesh of a fuselage component is a combination of structured and unstructured mesh as shown in figure 5.10. The faces, which contains the wing holes, are meshed with triangular panels, while the remaining faces are meshed with quadrilateral panels. For an isolated fuselage without any wings, the mesh of the fuselage will be a structured mesh without any triangular panels.

A structured mesh requires that the opposite edges of the mesh faces to have the same number of mesh points. The same number of mesh points on the opposite edge are required to create a mesh with only quadrilateral panels. For the fuselage, two kinds of chains are defined and these chains are the lateral chains and the longitudinal chains. The chains of the fuselage component are shown in figure 5.18. The top image shows all the edges of the fuselage component, while the middle image shows the lateral edge chains of the fuselage. The bottom image shows the longitudinal chains of the fuselage component.

All the input parameters that determine the number of mesh elements on the fuselage mesh are enumerated below.

- **tri_max_pitch** specifies the maximum size of the triangular panels of the unstructured mesh. This parameter limits the sizes of the triangular panels of the unstructured mesh and an increase in the value of this parameters causes the triangular panels of the unstructured mesh to become larger. A large panel means fewer panels and fewer panels means the curvature of the fuselage surface is modelled less accurately.
- **longitudinal_pitch** gives the maximum allowed distance in millimetres between two consecutive mesh points of the longitudinal edge chains. The longest edge in the chain will be used for determining the number of mesh points.
- **lateral_pitch** gives the maximum allowed distance in millimetres between two consecutive mesh points for the fuselage cross-section edges that is the longest in terms of the circumferential length. The cross section edges is a list of lateral edges that represents a cross-section of the fuselage. The resulting number of mesh points will then be used for deriving the number of mesh points for each lateral edge chain.

The number of mesh points of a lateral chain is determined by identifying the longest edge in the chain. The longest edge is used to determine the circumferential length of the fuselage cross-section to which

this edge belongs to. The circumferential length is then used for determining the pitch of the lateral chain by dividing this length with the number of mesh points of the longest fuselage cross-section.

The pitch of the **lateral_pitch** input parameter is not used for determining the number of mesh point of each lateral chains. The reason is that the edges of the lateral chains at the tail of the fuselage has lengths that are very small when compared to the lengths of the edges of the lateral chain at the centre of the fuselage. This unequal length will produce a mesh that has very few panels at the tail of the fuselage, while there are a lot of panels at the centre of the fuselage.

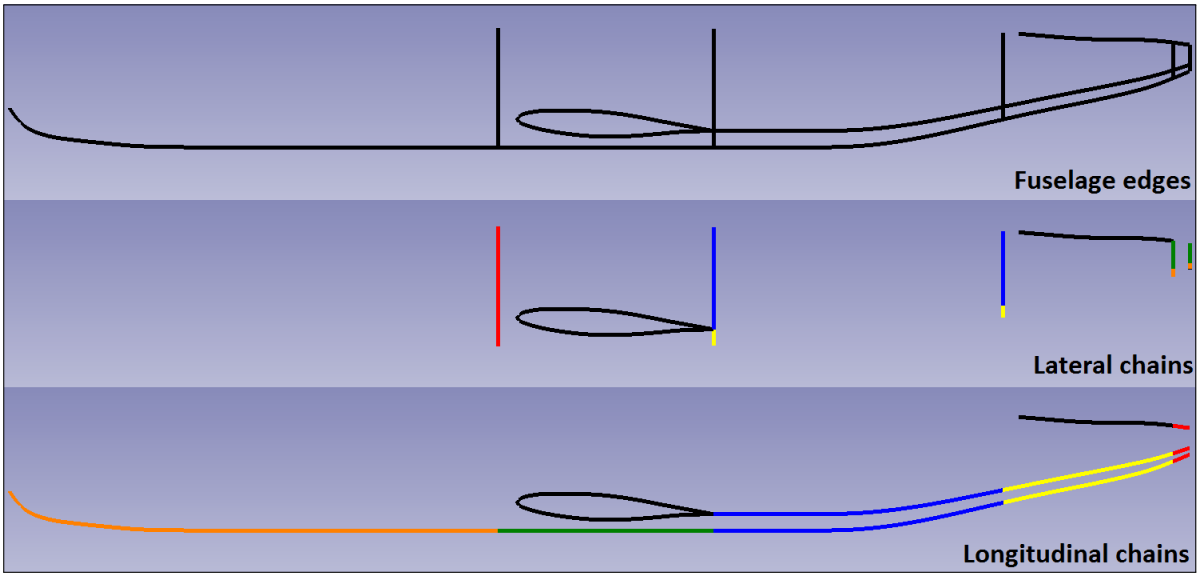


Figure 5.18: Longitudinal and lateral edge chains of a fuselage component

5.5. Mesh Results

The *mesh.salome* class uses the built-in mesher tool to mesh the aircraft model. The rules for controlling the mesh of the models are extracted from the *WingMeshControls* class and/or the *FuselageMeshControls* class. Below a few examples of the mesh results for different aircraft models.

Figure 5.19 shows the mesh result of an isolated fuselage without wings. The figure shows a fully structured mesh for the fuselage.

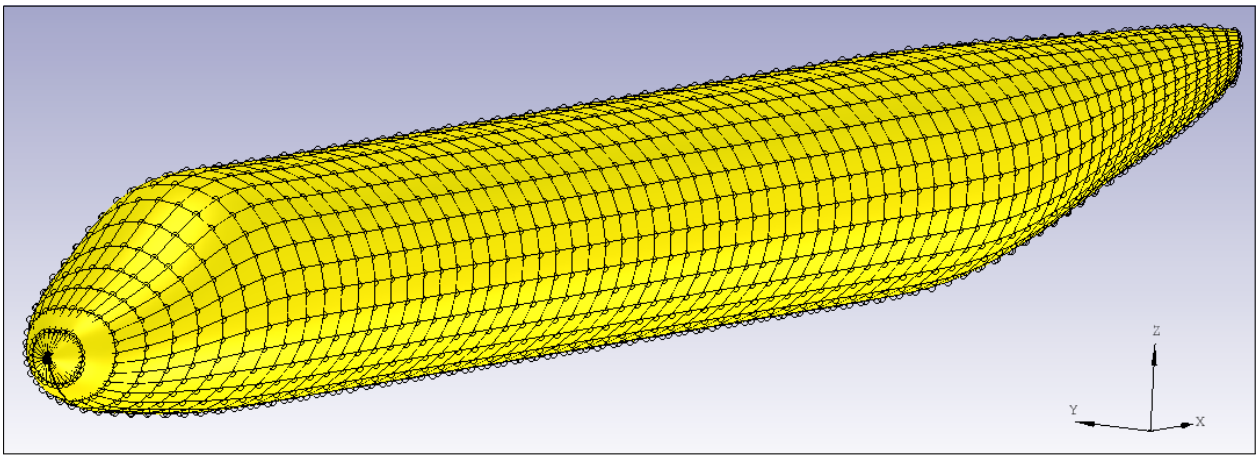


Figure 5.19: Mesh of an isolated fuselage without wings

Figure 5.20 shows a mesh of a wing that has three moveables. The moveables are modelled by using the 'transition surface' option. This wing has a mesh that is fully structured.

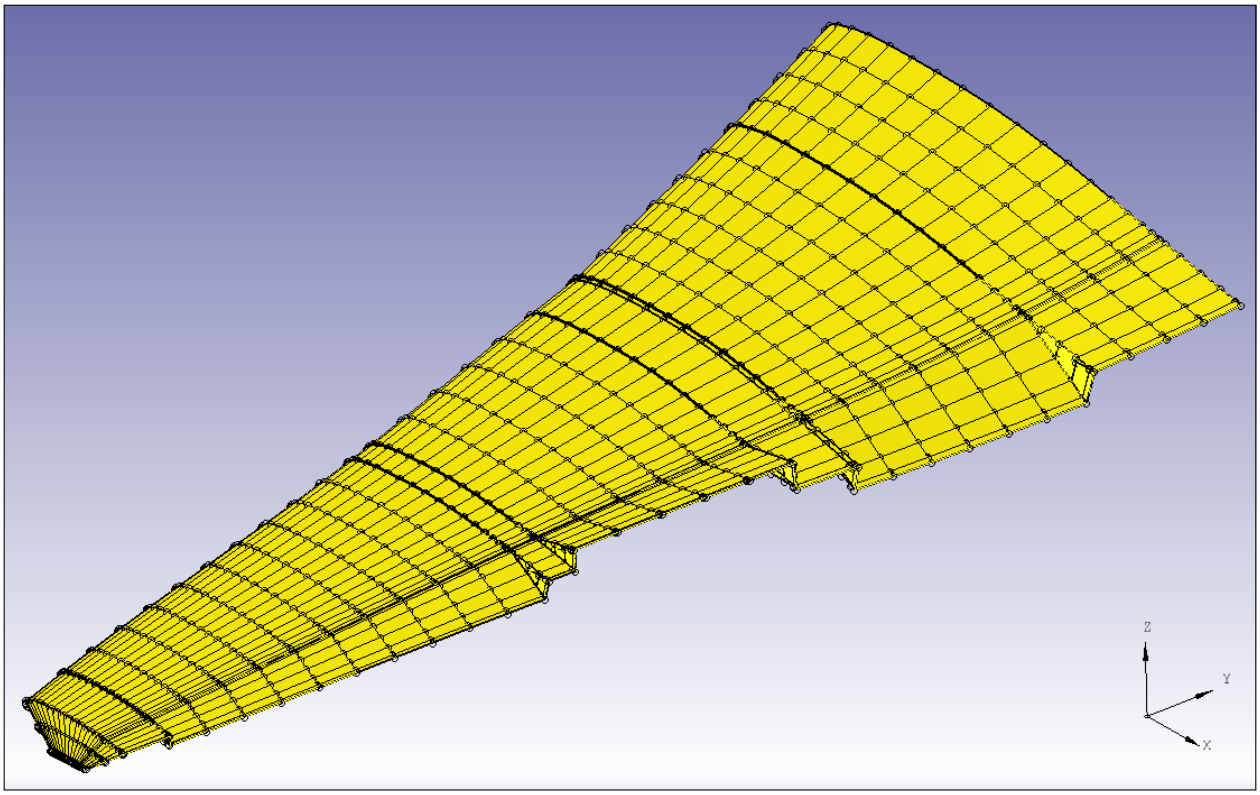


Figure 5.20: Mesh of a wing with moveables modelled by using the 'transition surface' option

Figure 5.21 shows the mesh of a conventional T-tail aircraft that has two moveables on each wing. The moveables are modelled by using the 'gap' option. The mesh of the conventional aircraft is a hybrid mesh. The unstructured mesh is shown at the fuselage sections, where the wings of the aircraft are attached to it. The unstructured mesh at the vertical tail is shown in figure 5.22. Figure 5.23 shows the mesh at the gap between the moveable and the wing. This figure shows the cove faces are not meshed. These faces could not be meshed by the built-in mesher tool because these faces have only three distinct sides. These faces are meshed "manually".

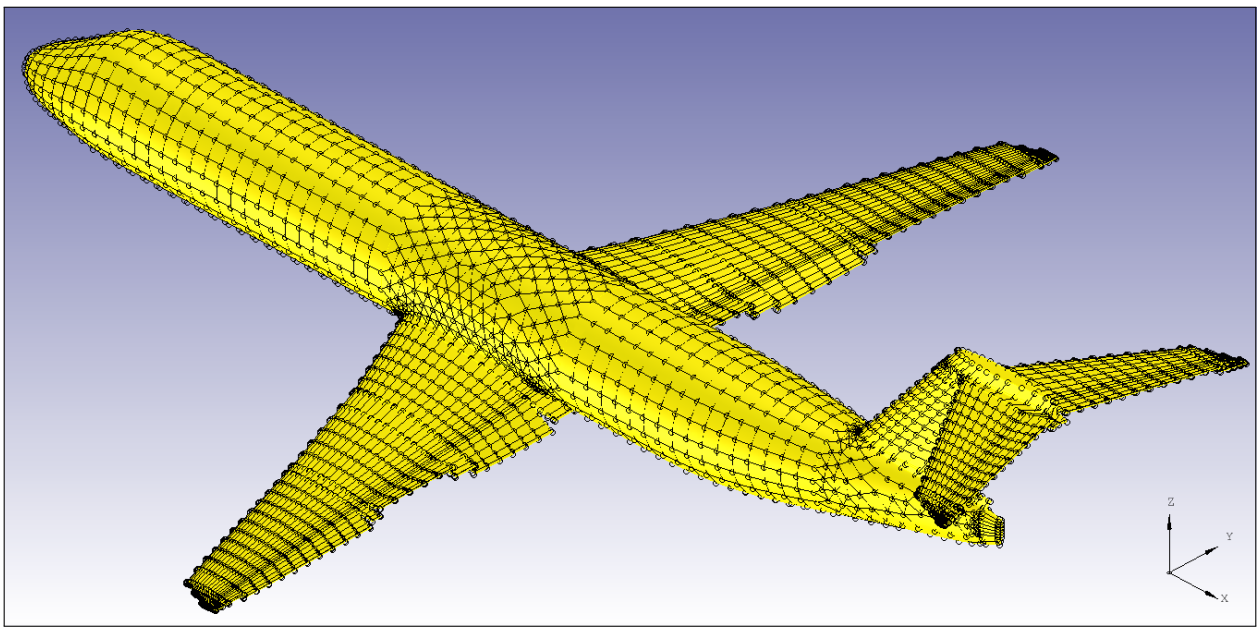


Figure 5.21: Mesh of a Fokker100 with moveables modelled by using the 'gap' option

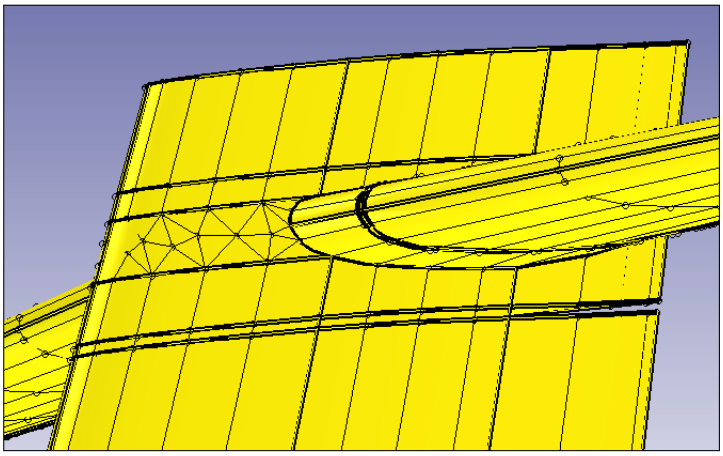


Figure 5.22: Mesh of the T-tail of figure 5.21

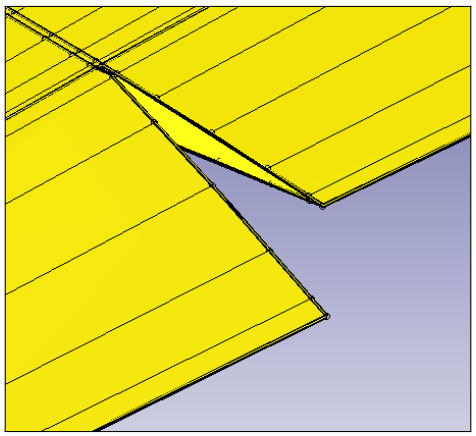


Figure 5.23: Mesh of wing at the moveable gap of the left wing of figure 5.21

Figure 5.24 shows the mesh of a blended body aircraft that has two moveables on each wing. The moveables are modelled by using the 'transition surface' option. The mesh of the blended body aircraft is a fully structured mesh.

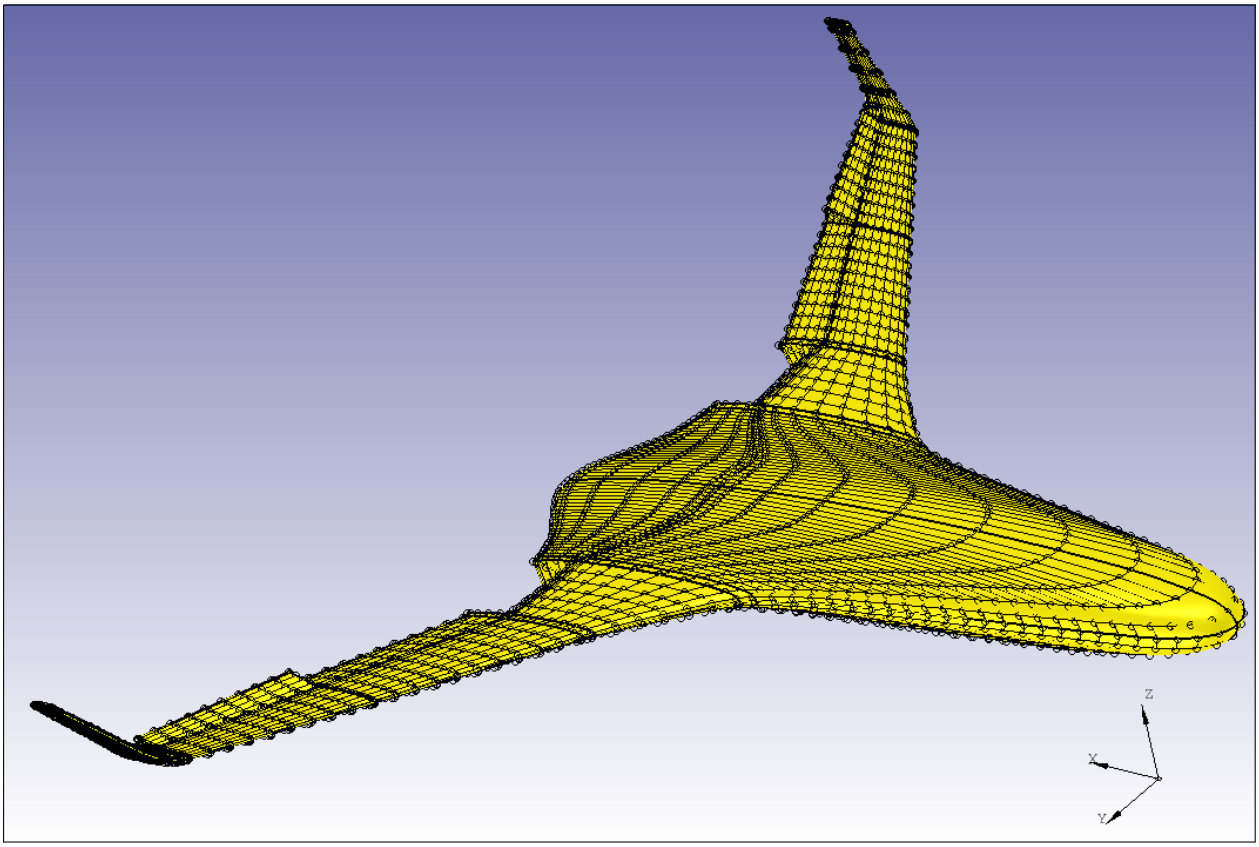


Figure 5.24: Mesh of a blended wing body aircraft with moveables modelled by using the 'transition surface' option

5.6. Manual Mesh Generation

The built-in mesher tool cannot mesh all the faces of the aircraft with quadrilateral panels, because there are faces on the aircraft that have less than four distinct sides. These faces, however, can be meshed such that it has a structured mesh. These faces are for example the root face, tip face, gap faces and cove faces of a wing component as shown in figure 5.25. Beside those faces the nose face and tail face of the fuselage component can also be meshed such that they have a structured mesh. The root face, tip face, gap faces, cove faces, nose face and the tail faces will thus be ”manually” meshed such that they have a structured mesh. The reason for choosing a structured mesh instead of an unstructured triangular mesh is that an unstructured mesh in VSAERO requires more computation time than a structured mesh.

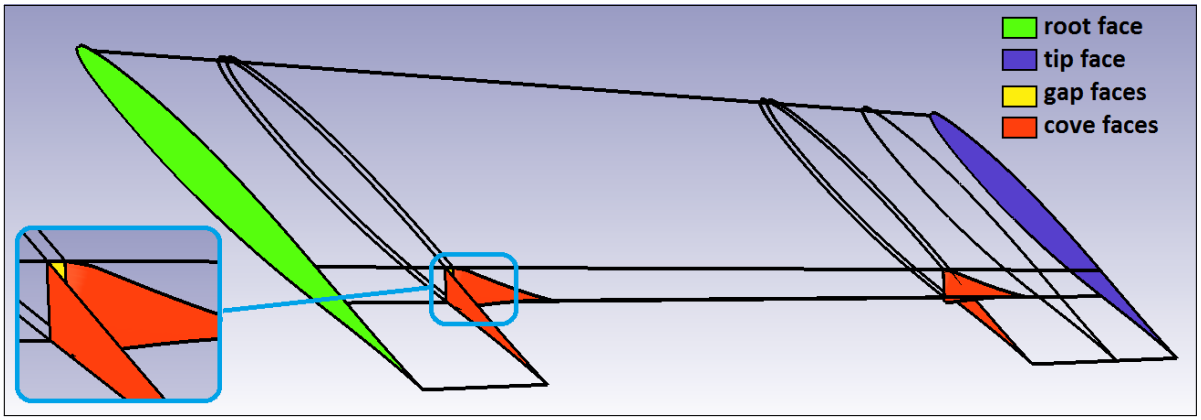


Figure 5.25: Root, tip, cove and gap faces of a wing component

The mesh for the nose, root and cove faces are shown in figure 5.26. The black plus signs are the mesh points that has been generated by the built-in mesher tool. These mesh points are retrieved from the mesh grid by using the red and green edges of the nose, root and cove faces. These mesh points (black plus sign) are used to derive the intermediate mesh points given as the yellow plus signs. These intermediate mesh points and the mesh point from the mesh grid must be collected and sorted into one array of mesh points. The procedure for creating this array of mesh points for the nose, root and cove faces can be explained using figure 5.26.

The creation of the array of mesh points starts by retrieving the mesh points of ’edge 1’ and ’edge 2’. The mesh points are retrieved by making use of the groups attributes that has been mentioned in section 5.4. The retrieved mesh nodes of each edge are then sorted such that the mesh nodes of both the ’edge 1’ and ’edge 2’ start at the same side of the face and progress in the same direction to the other side of the face. For the nose mesh, the ’edge 1’ does not exist. So instead of ’edge 1’ three mesh points are created at the centre point of the nose face.

The sorted nodes are then used to determine the mesh points between ’edge 1’ and ’edge 2’ of the face. The position of these computed mesh points is determined by first connecting the retrieved mesh points of ’edge 1’ with the retrieved mesh points of ’edge 2’. These connections are represented in figure 5.26 as black lines that connect mesh points of ’edge 1’ with the mesh points of ’edge 2’. Depending on the settings *n* numbers of equal distanced mesh points are generated on these black lines, these mesh points are the yellow computed mesh points of figure 5.26. The *n* number is determined by the **no_intermediate_pts** input parameter. These computed points are then collected and they are combined with the retrieved mesh points to create the arrays of mesh points for the nose, root and cove faces.

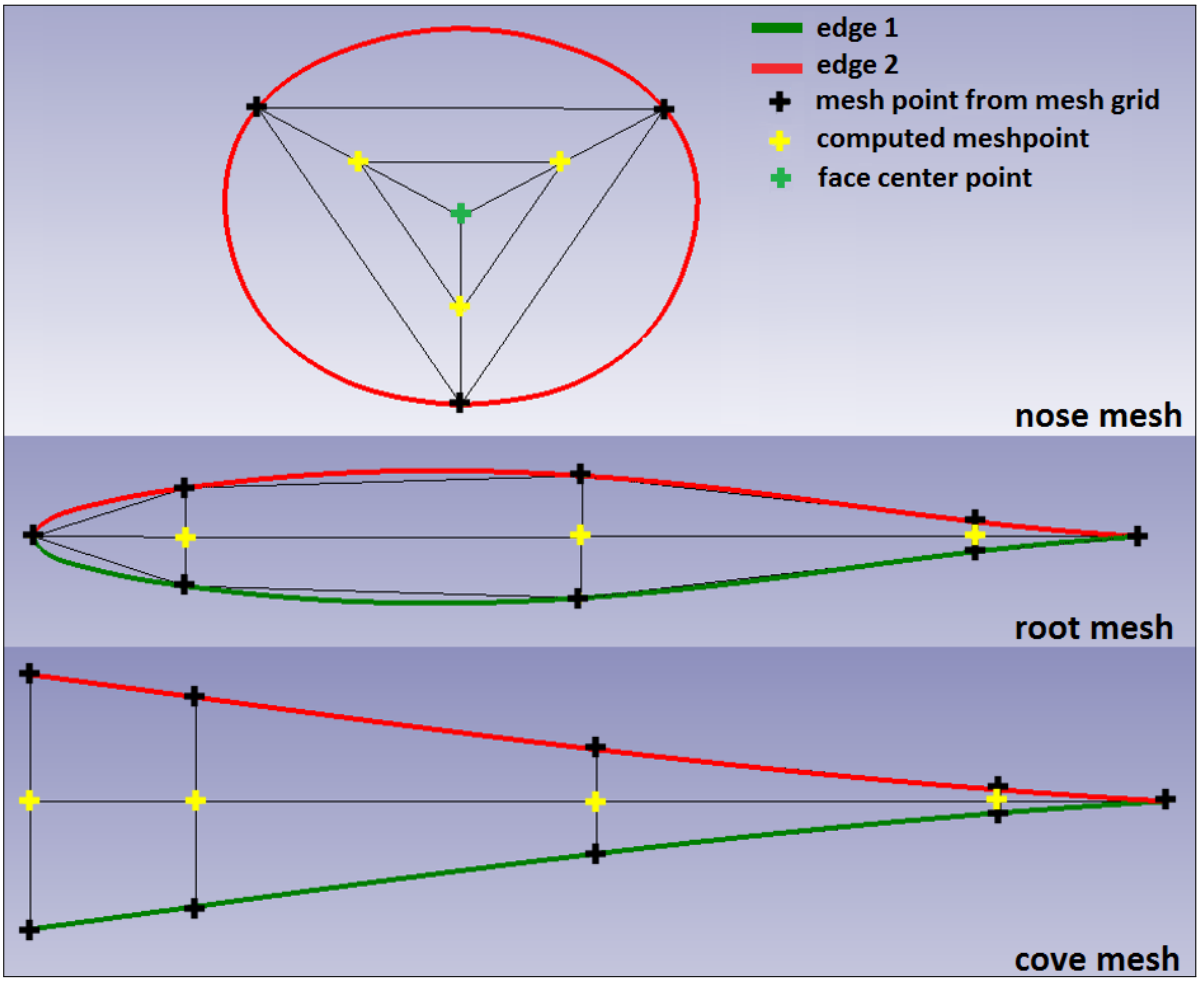


Figure 5.26: Nose, root and cove mesh

5.7. Mesh Processing

The mesh grid of the aircraft, which is being generated by the built-in mesher tool, needs to be processed first. After the processing, the mesh data can then be used to generate the panel model for the aerodynamic solver VSAERO. VSAERO requires each component of the aircraft model to be divided into patches with each patch representing one or multiple wing or fuselage faces. The shape of each patch must be defined such that there are no large kink angles of around 90 degrees present. A patching strategy that is recommended by Nathman[18] for a slotted wing is shown in figure 5.27. He recommends that the wing should be divided into three patches instead of one patch for the whole wing.

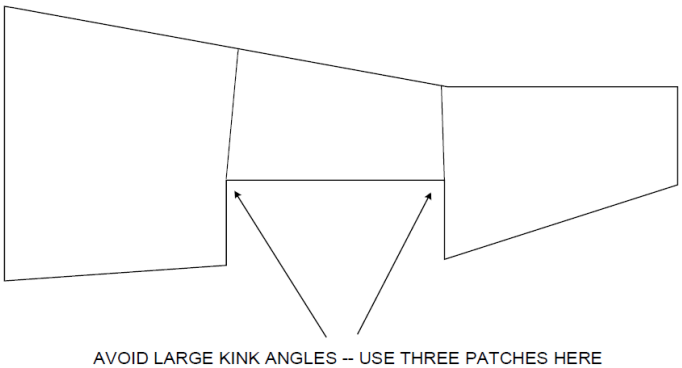


Figure 5.27: Shape of structured patches [18]

The patching strategy of Nathman combines all the faces that can be linked to each other in the chordwise direction while avoiding the very large kink angles as illustrated by figure 5.27. Figure 5.28 shows this patching strategy for a wing component that has a moveable with gaps between the wing and the moveable in the spanwise direction. This strategy will produce five patches for the top surface. The patching strategy that has been implemented in the aircraft mesher tool is different than the one that is proposed by Nathman. The patching strategy that has been implemented in the aircraft mesher tool is shown in figure 5.29. This patching strategy combines all the faces that can be linked to each other in the spanwise direction. This strategy will produce four patches for the top surface. The reason for choosing the strategy that combines the faces in the spanwise direction is that it produces fewer patches than the other patching strategy.

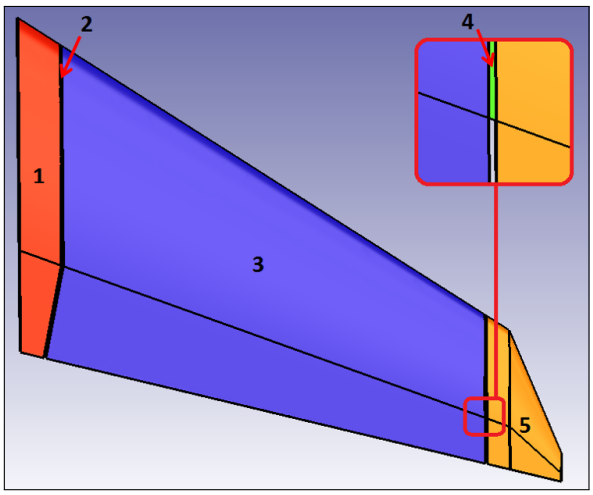


Figure 5.28: Chordwise patching strategy

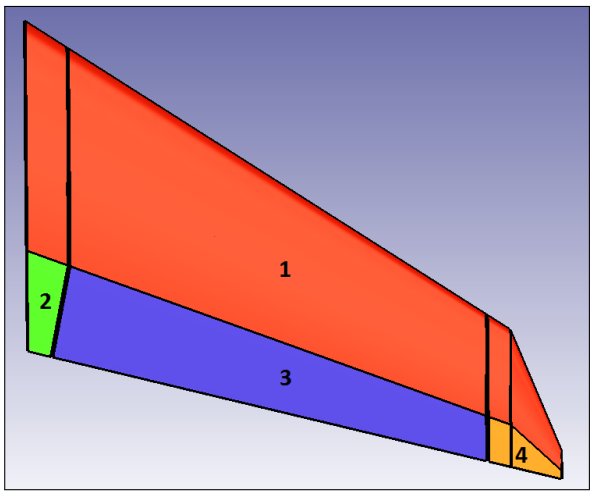


Figure 5.29: Spanwise patching strategy

The patching strategy for the fuselage component is very simple. The fuselage of a conventional aircraft is divided into section five sections as shown in figure 5.10. The figure shows the different sections of the fuselage that have a structured mesh and sections that have an unstructured mesh. Each section of the fuselage depicted in this figure also represents a patch for the fuselage component. The patching strategy of the fuselage is thus that each consecutive cross-sections of the fuselage will form a patch.

After the definition of the patches for the wing and the fuselage components, the array of mesh points representing the patch needs to be created. The creation of the array of mesh points of the patch is performed by the *VSAEROPatchContainer* class. The *VSAEROPatchContainer* class starts by extracting the mesh points of the input faces that represent the patch. The mesh points of each face are then sorted. The sorted mesh points of each array are then collected and the collected arrays are put into a list. The arrays in this list are then merged with each other to form one single array of mesh points. The sorting and merging step is skipped if the mesh points are from the manual mesh or if the mesh is unstructured. The mesh points of the manual mesh do not require sorting because they were already sorted when they were created ”manually”. The mesh points of a unstructured mesh do not need sorting because an unstructured patch does not require the mesh points to be ordered in a specific arrangement.

Another function of the *VSAEROPatchContainer* class is to search if there are moveable faces present in the input faces. If there are moveable faces present in the input faces and the moveable of the model is created according to the ’normal rotation’ option then the following steps are taken.

- Identify the row and column numbers of the patch that forms the moveable.
- Create four attributes stating the start and end numbers of the rows and columns of the patch.
- Extract the hinge/rotation line vector from the moveable modelling tool and use this as input for the rotation vector.
- Extract the deflection angle from the inputs

These steps creates the required attributes for writing the inputs to rotate the moveable panels in VSAERO.

5.8. Wake Assembly

The wake model of the aircraft consists of multiple wakes, where each component of the aircraft has its own wake model. A wake model for an aircraft component in VSAERO is described by a set wake lines that are aligned with the local flow direction. Figure 5.30 shows these wake lines for a wake of a wing component. In VSAERO the wake panels are created by intersecting the wake lines of the wing component with the wake grid planes that are parallel to global YZ-plane. These wake grid planes are used to distribute the wake panels in the flow direction. These planes are defined once and this means that the wakes of all the aircraft components use the same set of wake grid planes to define the distribution of the wake panels in the flow direction.

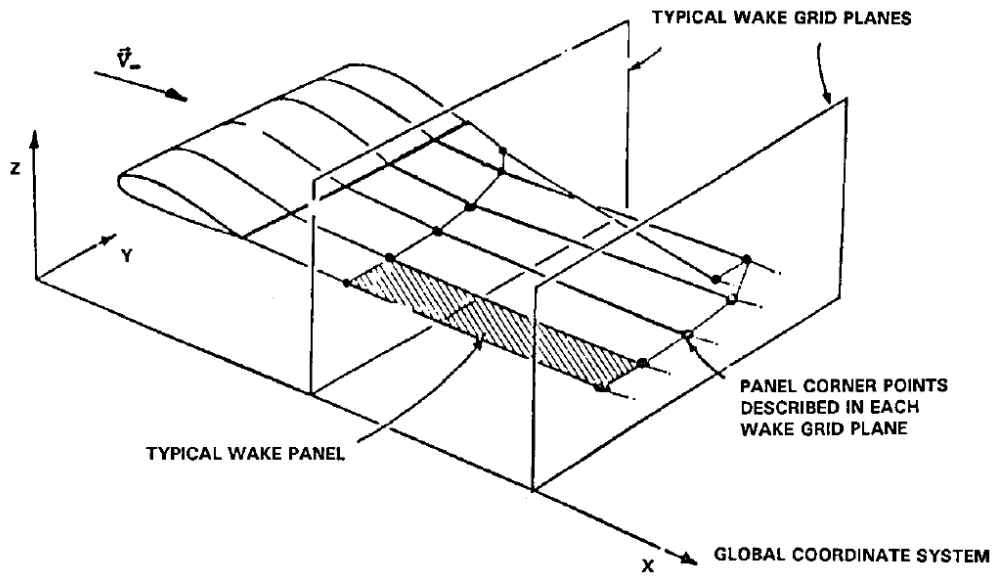


Figure 5.30: Wake grid planes[18]

The wake grid planes are infinite planes and they are parallel to global YZ-plane. The only variable that remains for creating the wake grid planes are the x-coordinates of all the wake grid planes. The spacing between the x-coordinates of the different wake grid panels should be small at positions, where there are intense vortex roll-up. The x-coordinates of the wake grid planes are determined as follows:

- Collect the root trailing edge x-coordinates of all the wing components that are not mirrored thus excluding the right wings of the aircraft model. Also, the x-coordinate of the aft-most point of the fuselage component is added to the list of x-coordinates, which is called ”fixed stations”.
- Add another x-coordinate to the ”fixed stations” list. This x-coordinate has the value of the largest x-coordinate of the ”fixed stations” list plus the wingspan of the aircraft.
- The number of wake planes between the x-coordinates of the ”fixed stations” list is determined by the **N** input setting. The *N* number of wake planes between each pair of ”fixed stations” are distributed according to the cosine distribution.
- After the last ”fixed station” additional wake grid planes are added. These additional planes are also distributed according to the cosine distribution with the amount that is specified by **N_last** input setting. The last x-coordinate of these additional stations is determined to be 10 wingspans or 100 chords behind the last ”fixed station”.
- Remove the x-coordinates that are too close to each other. The minimum distance between two consecutive x-coordinates is specified by the **min_pitch** input setting.

5.8.1. Wing Wake

The wake of a component is built up of one or multiple wake sheets. A wake sheet is a wake that has been shed by one of the patches of the component. It is, therefore, important to know that one wake sheet can never be shed by two patches. An example is given in figure 5.31, where the wake of the right wing of the aircraft is constructed out of five wake sheets. The patch outlines of the wing component are shown as black lines in the figure and each patch has one wake sheet attached to it.

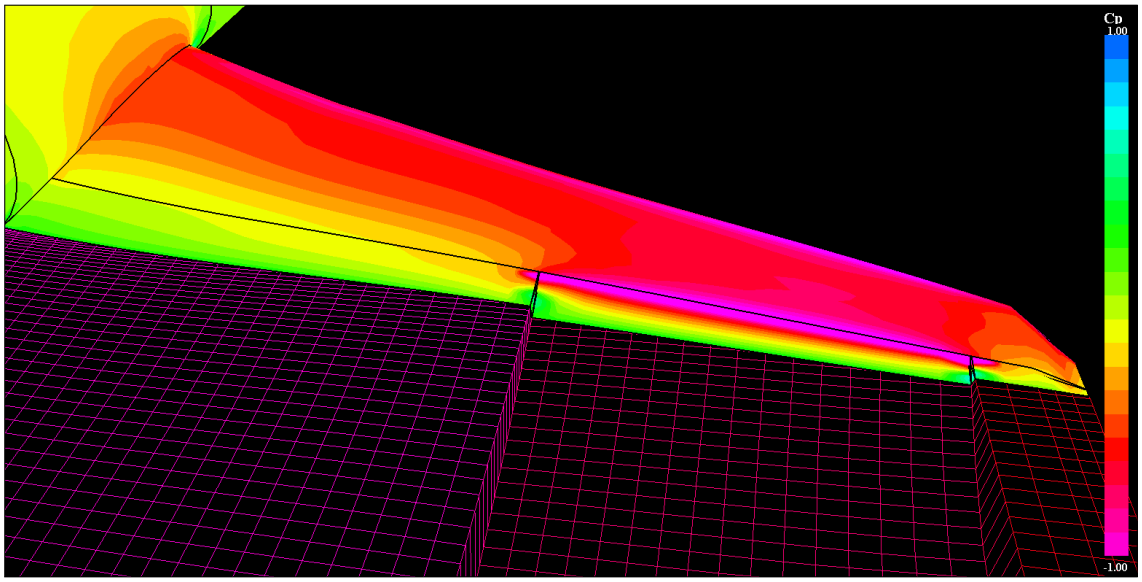


Figure 5.31: Wakes of the right wing component with transition surfaces

A wake sheet of a wing component is constructed out of two, three or four wake lines. Each of these wake lines is then attached to a corner point of the wake shedding panels of the patch. If there are more wake shedding panels than wake lines then VSAERO will automatically create the missing wake lines for the wake shedding panels that do not have a wake line attached to it.

Four wake lines are used for constructing a wake sheet if both ends of the wake sheet need to be stitched to other components of the aircraft. Three wake lines are required if only one side of the wake sheet is stitched to another component of the aircraft, see figure 5.32. Two wake lines are sufficient for modelling the wake sheet if both ends of the wake sheet do not require any stitching. The difference between a stitched wake line and a flexible wake line is that the flexible wake line is free to move during wake relaxation, while this is not the case for a stitched wake line.

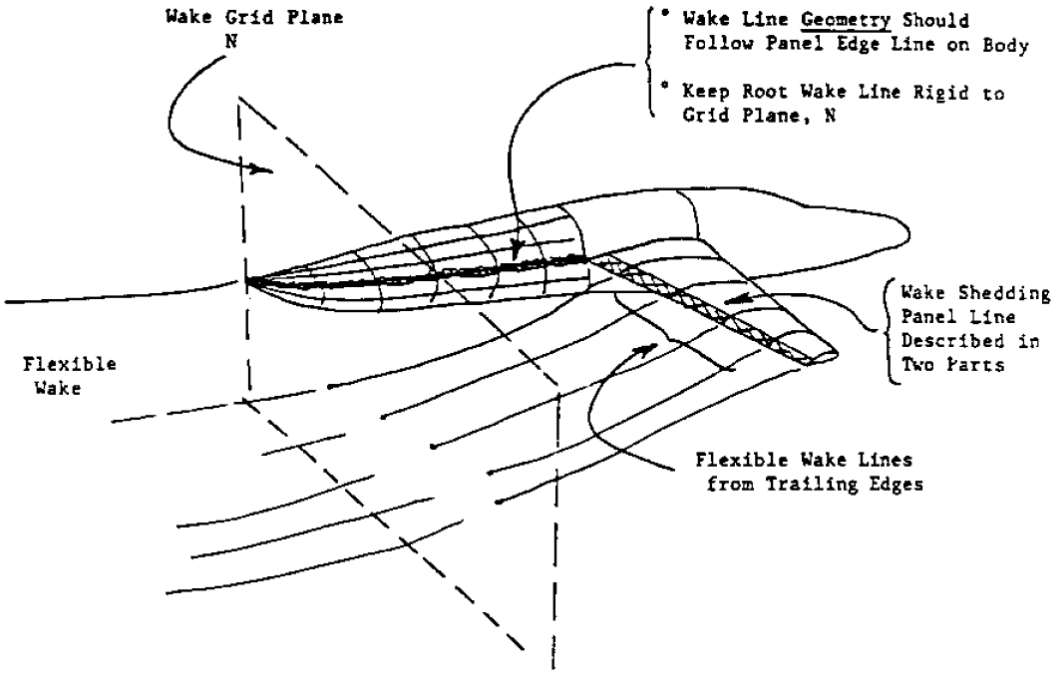


Figure 5.32: Stitching of the root wake line to the fuselage[18]

Stitching of the wake lines to another aircraft component is recommended by Nathman[18]. Stitching the wake lines to the other components will prevent the wake line from moving away from the surface of the other component and hence preventing the production of a strong vortex by the gap between the wake sheet and the surface of the other aircraft component.

A wing with moveable model that is modelled using the 'gap' option has gaps between the wing and the moveable in the spanwise direction, see figure 5.33. Two wake sheets are created to cover the gap between the wing and moveable. One wake sheet for the top surface and one wake sheet for the bottom surface of the wing. This "gap" wake sheet is shed from the blue spanwise edge as shown in the figure. Each "gap" wake sheet is constructed of two wake lines, which are attached to the chordwise edges of the moveable and the wing to prevent it from moving away from the wing surface.

A wing with moveable model that is modelled using the 'transition surface' option has transition surfaces that connect the wing surface and the moveable surface, see figure 5.34. The wake sheet of the transition surface is created as a flexible wake sheet by attaching two wake lines at the corners of "transition surface" patch.

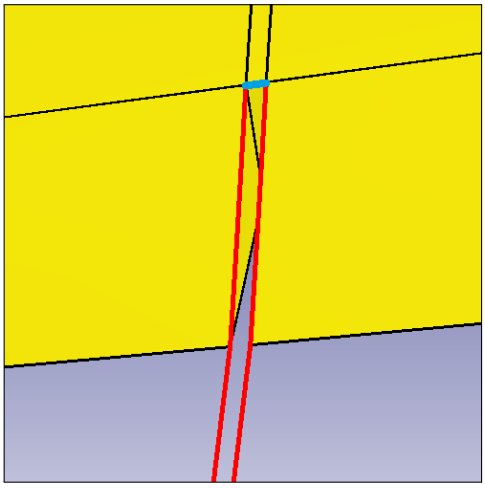


Figure 5.33: Two wake lines for attaching the gap wake sheet to the wing surface

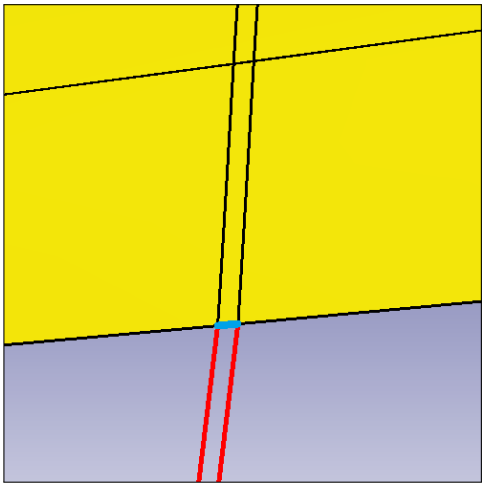


Figure 5.34: Two wake lines for creating the wake sheet of the transition surface

Wake lines

As described above, a wake sheet can be modelled from stitched wake lines and/or flexible wake lines. The modelling of a stitched wake line is very simple and it is demonstrated by figure 5.35. The figure shows the chordwise root edges of the main wing and the mesh points of the fuselage with the yellow plus signs. The fuselage mesh points are used for stitching the root wake line of the main wing to the fuselage. The red line in figure 5.35 is created for visualising the stitched wake line. The fuselage mesh points are sufficient for defining the stitched wake line.

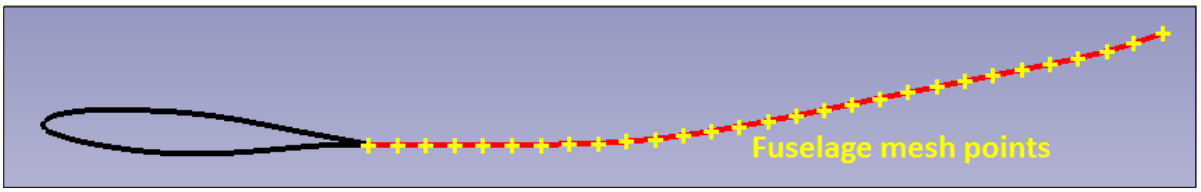
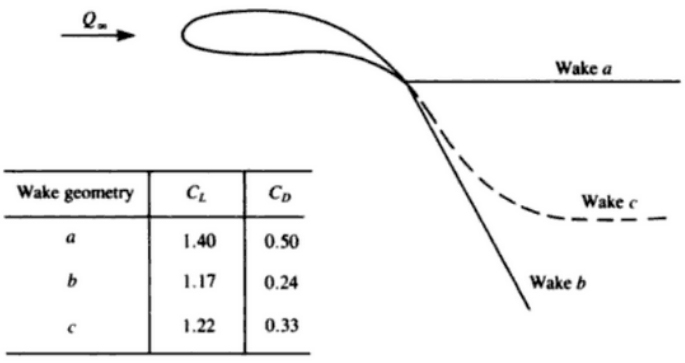


Figure 5.35: Root airfoil edges with a stitched wake line

The modelling of the flexible wake line is a bit more complicated than the modelling of the stitched wake line. The shape of the flexible wake line can be modelled in three different ways as presented in figure 5.36. Shape *a* models the wake line as a straight line that is parallel to freestream airflow, while the wake line of shape *b* is unaffected by the freestream airflow and leaves the trailing edge of the airfoil with an angle. Shape *c* is a compromise between shapes *a* and *b*. Shape *c* leaves the trailing edge of the airfoil at an angle but this angle decreases till the wake line is parallel to freestream airflow.



Wake geometry	C_L	C_D
a	1.40	0.50
b	1.17	0.24
c	1.22	0.33

Figure 5.36: Effect of prescribed wake geometry on the aerodynamics of an AR = 1.5 wing[19].

The modelling of the flexible wake line is explained by using figure 5.37 as an example. The figure shows an airfoil with a deflected moveable and a red BSpline curve representing the wake line. The BSpline curve is created by using the yellow points as control points. The yellow control points *C* and *D* are determined by computing the angle β and point *A*.

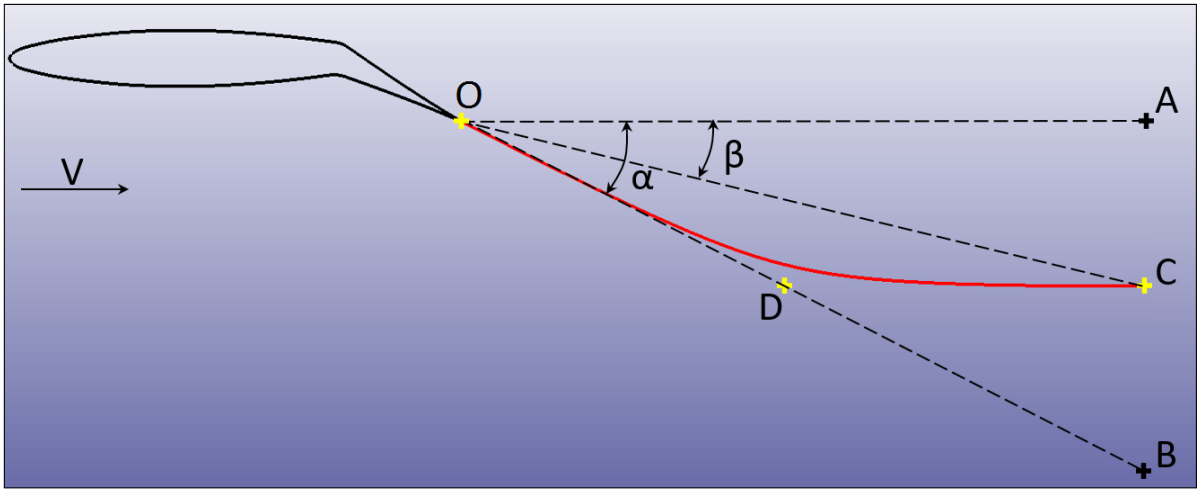


Figure 5.37: Modelling of wake line geometry.

Point *A* is determined by translating the trailing edge point in the freestream flow direction with a distance that is computed by equation 5.1. The distance is thus computed by multiplying the local chord length with a factor specified by the **distance_factor** input parameter. The angle β, on the other hand, is computed by multiplying the angle α with a factor specified by the **angle_factor** input parameter, see equation 5.2. The angle α is the angle between the freestream flow direction and the averaged tangent vector at the trailing edge point.

$$distance\,OA = chord * distance\_factor \tag{5.1}$$

$$\beta = \alpha * angle\_factor \tag{5.2}$$

After determining the position of point *A* and the angle β, the positions of points *C* and *D* can be determined. Point *C* is determined by translating point *A* along a vector that is perpendicular to the freestream flow direction and it is pointing in the same direction as the tangent vector of the trailing edge point. The distance of this translation is determined by equation 5.3. Point *D*, on the other hand, is determined by translating the trailing edge point along the tangent vector of the trailing edge point with a distance that is specified by equation 5.4

$$distance\,AC = distance\,OA * tan(\beta) \tag{5.3}$$

$$distance\,OD = \frac{distance\,AC}{sin(\alpha)} \tag{5.4}$$

As the last step in the wake line modelling, the stitched wake line and the free moving wake line are extended with a straight line that is parallel to the freestream airflow. The length of this straight line is the x-coordinate of the last wake grid plane minus the x-coordinate of the last point of the wake line.

Figure 5.36 shows three shapes for modelling the wake lines. Shape *a* is created by setting the **angle_factor** parameter to 0, while shape *b* is created by setting the **angle_factor** parameter to 1. Shape *c* is created if **angle_factor** is larger than 0 and smaller than 1. Figure 5.36 also shows the performance of the different wake shapes. It is clear from the figure that the performance is affected by the shape of the wake, but note that the results of this figure are a simulation in VSAERO with no wake iterations. The effect of wake iterations on wake shapes *a* and *c* are presented in section 6.3.2.

5.8.2. Fuselage Wake

The wake of the fuselage component is modelled as a closed wake where the start of the wake coincides with the end of the wake as shown in figure 5.38. The blocking strategy of section 5.3.2 ensures that aft part of the fuselage is meshed as one mesh grid. This mesh grid is then converted to one patch and this patch is used for shedding the fuselage wake sheet. This fuselage patch can be observed in figure 5.38 as black outlines.

The fuselage wake sheet is modelled by generating two straight and identical wake lines to represent the start and end shape of the wake sheet. The two wake lines are positioned at the same location at the bottom of the fuselage, where the aft fuselage patch starts and ends. All the intermediate wake lines between the start and end panels of the aft fuselage patch are automatically generated by VSAERO.

The distribution of the fuselage wake panels in the flow direction was specified at the start of this section and it was distributed according to the cosine distribution. A cosine distribution creates a lot of panels at the start of the wake. A dense panels distribution at the start of the wake is desired since the wake of the fuselage will move during the wake relaxation process of VSAERO. The movement of the wake is the largest just after the fuselage and a dense panel distribution can model this movement more accurately than a course panel distribution.

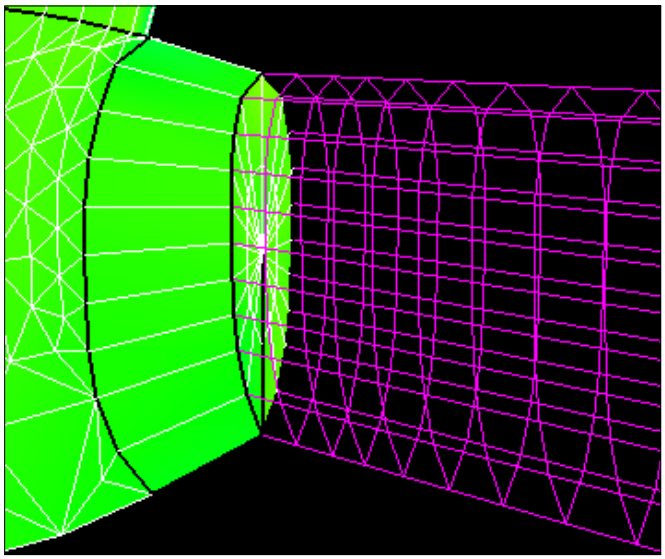


Figure 5.38: Closed wake of the fuselage component

5.9. Models in VSAERO

This section shows some examples of the panel model and the wake model of some aircraft models in VSAERO that have been created using the MMG and the aircraft mesher tool.

Figures 5.39, 5.40 and 5.41 shows the wings with moveables created according to the 'normal rotation', 'transition surface' and 'gap' options respectively.

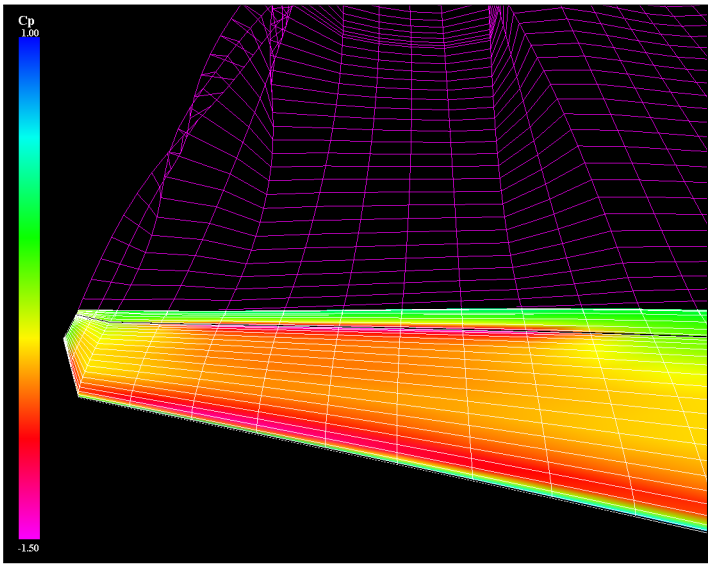


Figure 5.39: VSAERO result of wing with 'normal rotation' moveable

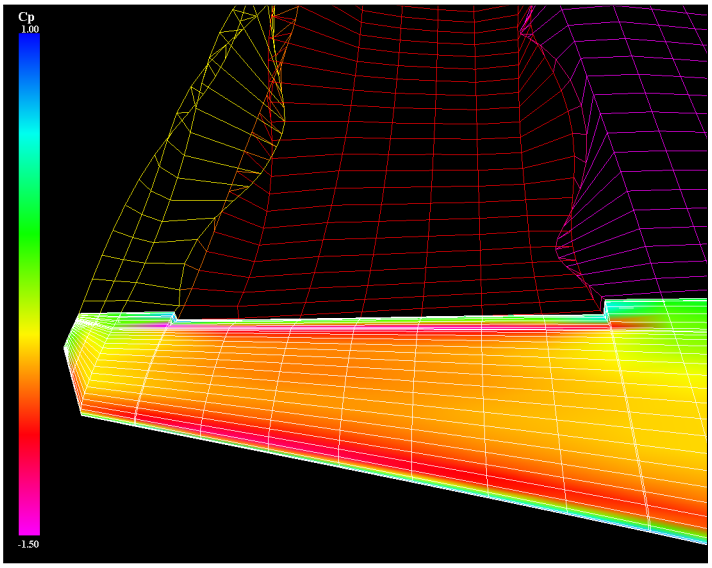


Figure 5.40: VSAERO result of wing with 'transition surface' moveable

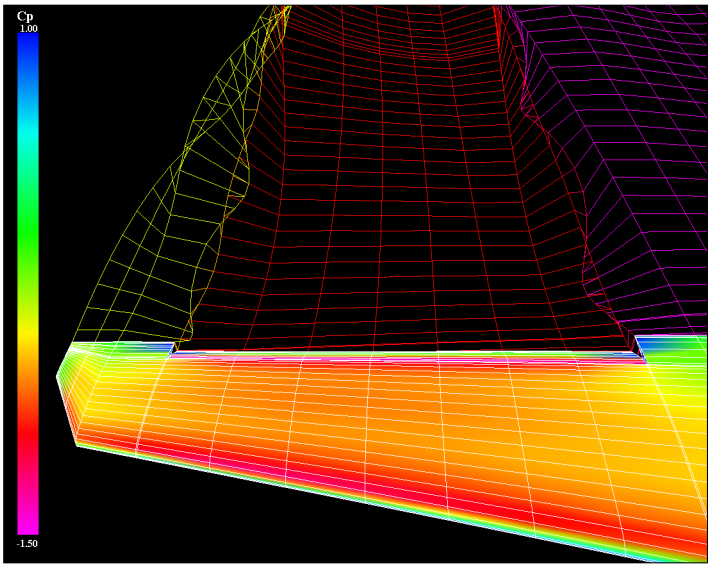


Figure 5.41: VSAERO result of wing with 'gap' moveable

Figure 5.42 shows the Fokker100 model that has been created using the MMG, while figure 5.43 shows a blended body aircraft.

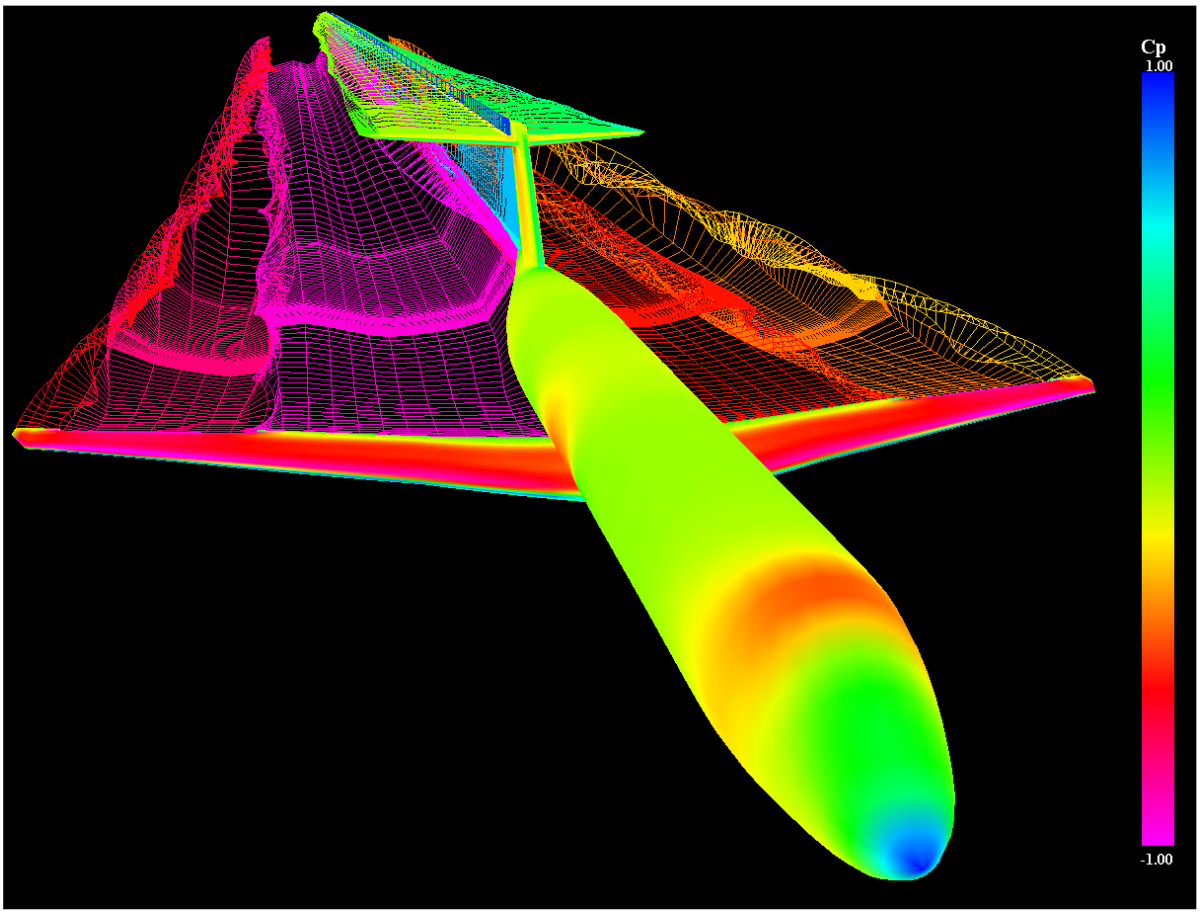


Figure 5.42: VSAERO result of Fokker100 model

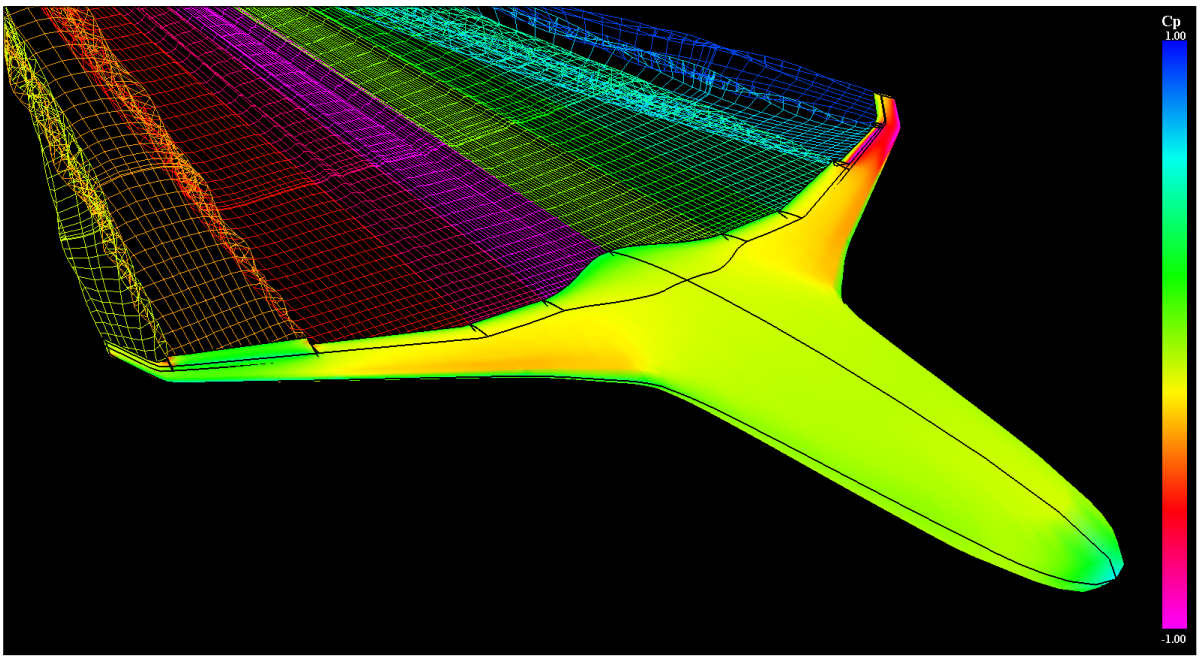


Figure 5.43: VSAERO result of a blended wing body aircraft

6

Validation of Aerodynamic Characteristics

The aerodynamic characteristics of the parametric aircraft model are validated in this chapter. The validation of the stability and control derivatives is performed in chapter 7. The validation of the aerodynamic characteristics is performed by comparing the aerodynamic data of the Fokker100 with the aerodynamic data that has been generated by the aerodynamic solver VSAERO. The aerodynamic data of the Fokker100 are from a Fokker100 report [20] and this data will be labelled in the figures as **'reference'**.

Section 6.1 will describe the Fokker100 model and the test condition that has been used for generating the aerodynamic data of this chapter. Section 6.2 analyse the performance of VSAERO such as accuracy and computation time by varying some input settings. Section 6.3 investigate the impact of the model variations on the accuracy of the results by changing the panel density and the shape of the wake line. Finally, the model will be validated in the last two sections by analysing the longitudinal and lateral-directional characteristics.

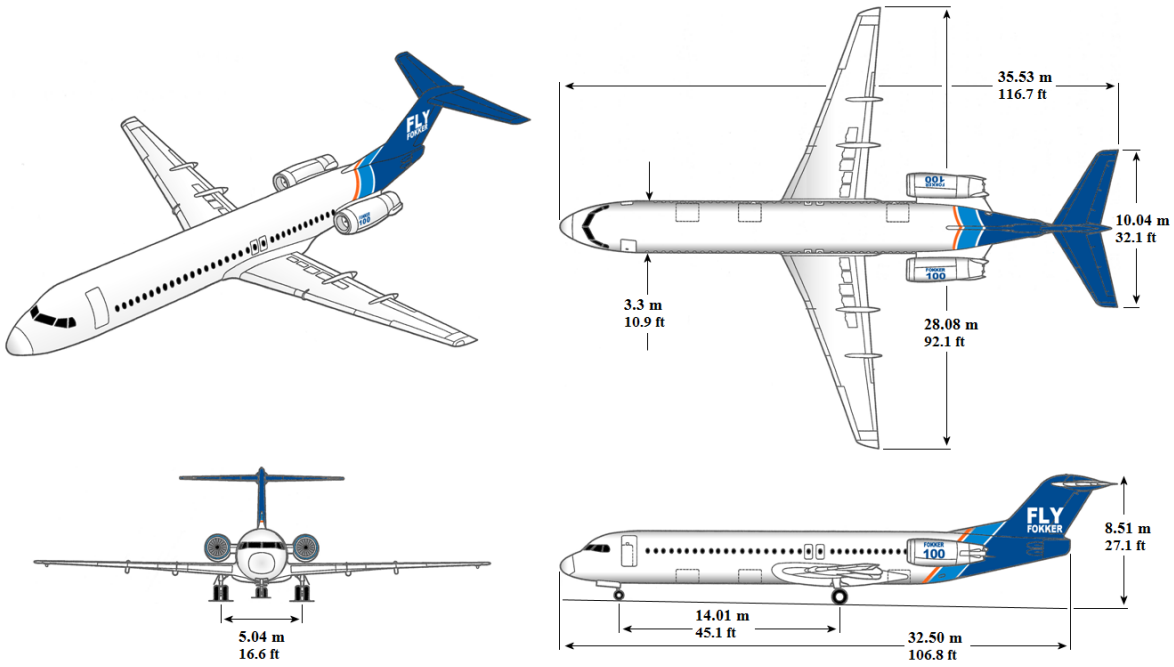


Figure 6.1: Fokker100 aircraft. Source [http://www.flyfokker.com/Basics](Last visisted: 18-11-2014)

6.1. Fokker100 Test Case

The aircraft, which has been chosen for validating the parametric model, is the Fokker100 aircraft. The Fokker100 aircraft is chosen due to the vast amount of available aerodynamic data that can be used to derive the stability and control derivatives of the aircraft [20]. This aerodynamic data can be used for validating the simulated aerodynamic data and the simulated stability and control derivatives. The Fokker100 was also used as the basis for the development of the parametric model with moveables and for creating the VSAERO model to determine the aerodynamic characteristics.

The Fokker100 aircraft is a medium-sized commercial aircraft with a twin turbofan engine mounted on the rear of the fuselage. Fokker100 is a conventional T-Tail aircraft with low mounted wings. The dimensions of the aircraft are illustrated in figure 6.1. The size and the position of the lifting surfaces of the Fokker100 aircraft are measured from an engineering drawing. The airfoil profiles of the lifting surfaces are from the MMG repository of Genworks GDL version.

Figure 6.2 shows the Fokker100 model that is constructed using the new MMG on the ParaPy platform. Please note that this model is not an exact replica of the Fokker100. This model is based on the measurement of an engineering drawing of the Fokker100. The exact values of for example the twist distribution of the wing is unknown. Also, the twin turbofan engines, the pylons of the engines and the dorsal fin of the Fokker100 aircraft are not modelled. The impact of the lack of the items as mentioned above will especially be observed in the zero-lift drag of the aircraft. The lack of the dorsal fin has an impact on the side force coefficient since the model has less wing area than the Fokker100 aircraft.

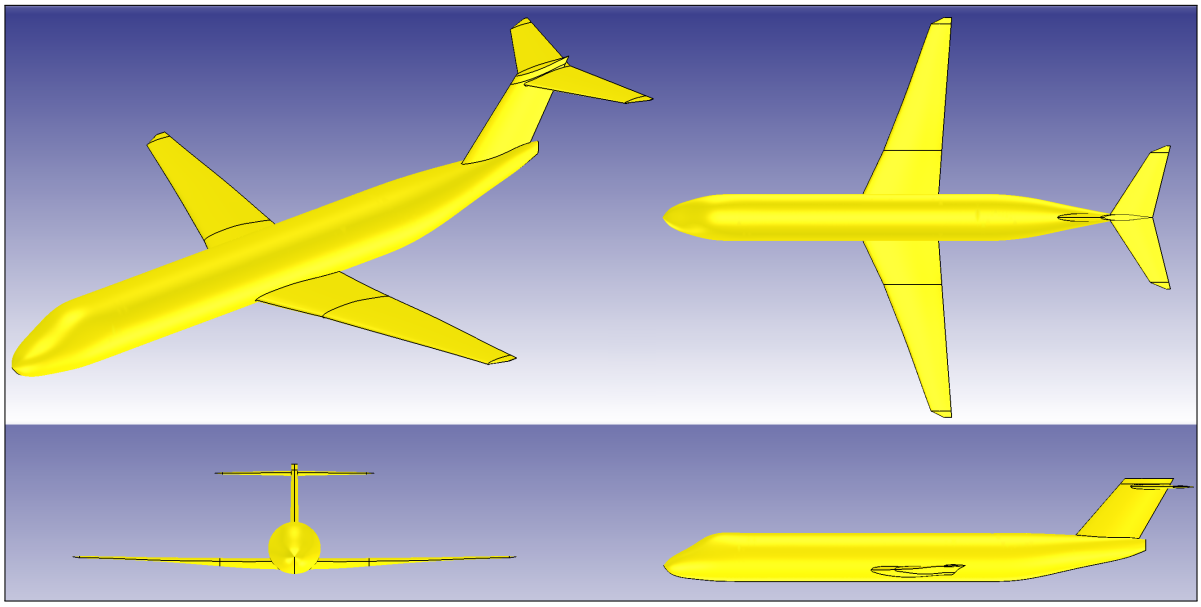


Figure 6.2: Fokker100 model in MMG ParaPy version.

All the aerodynamic data presented in this chapter are computed using the flight conditions of table 6.1. The reason for choosing this test condition is that the manoeuvring of an aircraft is mostly performed at low altitudes during take-off or landing phase. During these phases, the control surfaces are used the most for controlling the attitude of the aircraft.

The aerodynamic coefficients given in this thesis are determined using the reference parameters of table 6.2. The reference point or centre of gravity for computing the moment coefficients are given in the global axis. The global axis is located on the nose of the aircraft at the fuselage centerline. The X-axis points towards the tail of the aircraft. The Y-axis is perpendicular to the plane of symmetry and is positive in the direction of the left wing. The Z-axis is perpendicular to the XY plane and is positive upwards.

Table 6.1: Test condition

Variable	Value	Unit
Altitude	3000	[m]
Velocity	98.6	[m/s]
MAC	3.83	[m]
Mach number	0.3	[-]
Reynolds number	$20,3 \cdot 10^6$	[-]

Table 6.2: Reference parameters for the aerodynamic coefficients

Variable	Value	Unit
Chord	3.83	[m]
Area	93.5	[m^2]
Span	28.08	[m]
Center of gravity, X-coordinate	16.95	[m]
Center of gravity, Y-coordinate	0.00	[m]
Center of gravity, Z-coordinate	-0.25	[m]

6.2. VSAERO Settings

The most important settings, which affect the performance of VSAERO, are the type of the solver and the number of iterations. The type of matrix solver determines the computational speed and the convergence of the results. The number of iterations, on the other hand, determines the number of wake and/or viscous iterations that is being used for computing the aerodynamic data.

6.2.1. Matrix Solvers

The matrices resulting from the first order panel method can be solved by either using a direct solver or by using an iterative solver. A direct solver solves the matrices directly while the iterative solvers make an educated guess and iterate this until it converges. Direct solvers are more robust than the iterative solver, but the iterative solver, on the other hand, is significantly faster regarding computation time. This thesis will analyse one iterative solver and one direct solver of VSAERO. The direct solver that is chosen for comparison is the LAPACK solver. For the iterative solver, Nathman(VSAERO) advises to use BLOCKED GS since it was tested as the most reliable iterative solver. Instead of BLOCKED GS, LAPACK BGS is used as the iterative solver. LAPACK BGS uses a similar method as BLOCKED GS but then without restriction on the block size (number of panels in a patch).

The lift curve, the drag polar and the side force coefficient of the Fokker100 model for the different solvers are presented in figures 6.3, 6.4 and 6.5 respectively. These curves are simulated with 9493 body panels, 10642 wake panels, 10 wake iterations and 10 viscous iterations. Figures 6.3 and 6.4 show that there are no notable differences in the lift curve and in the drag polar, but there are some differences between the reference data and the simulated data. The lift curve is slightly overpredicted, while the drag is under predicted. The drag polar shows that the zero lift drag is under predicted,

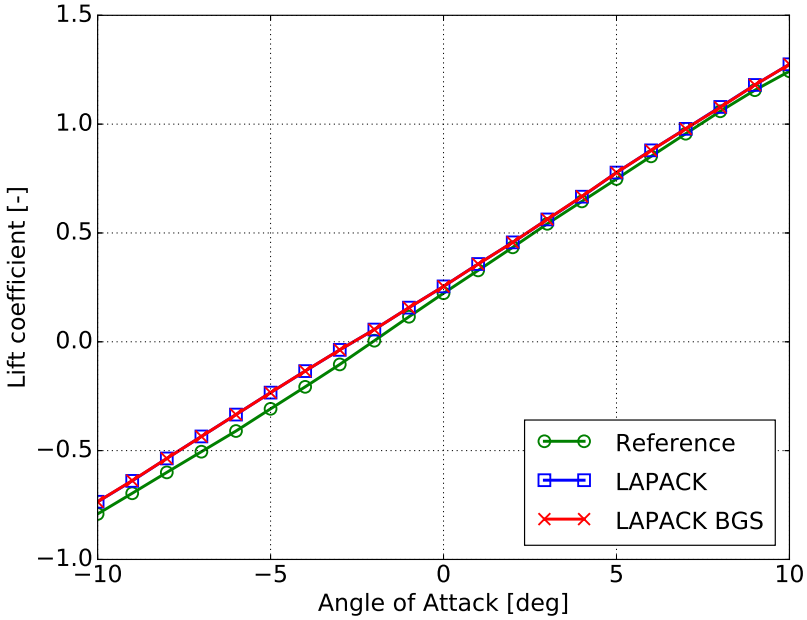


Figure 6.3: Lift curves with different solvers

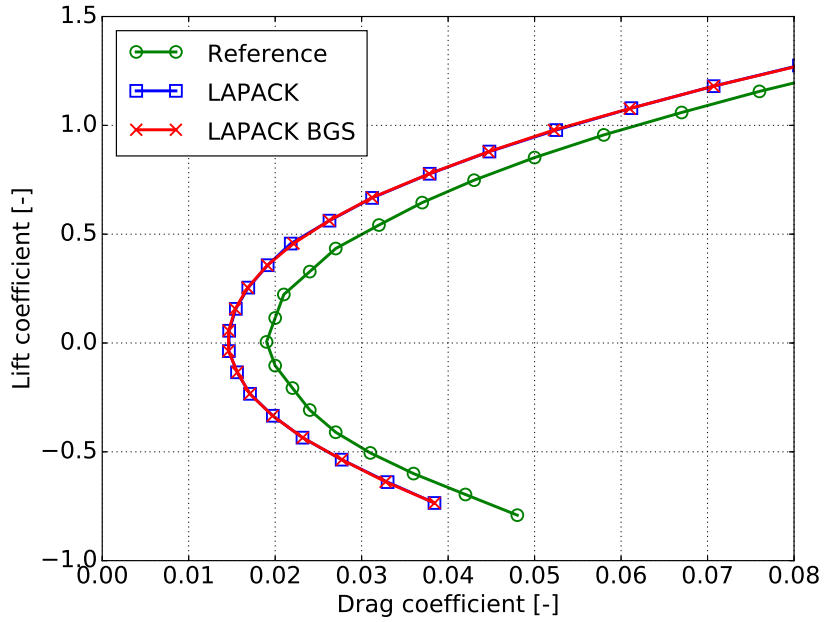


Figure 6.4: Drag polars with different solvers

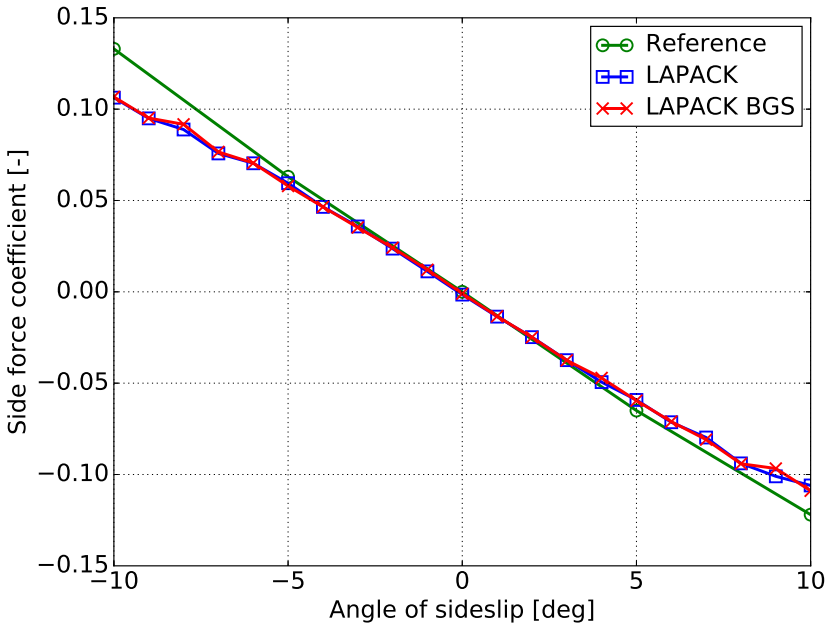


Figure 6.5: Sideforce coefficient versus angle of side-slip with different solvers

while the trend line of the simulated data seems to follow the trend line of the reference data. The underprediction of the zero lift drag is attributed to the lack of engines, engine pylons and the dorsal fin for the Fokker100 model. Figure 6.5 shows that for low slip angles the side force coefficient are almost the same for the two solvers, but both solvers begin to diverge after a slip angle of 4 degrees. The probable cause of this problem is depicted in section 6.5.1.

The most notable difference between the solvers is the required computation time to obtain the aerodynamic data. Figure 6.6 shows the runtime of the two solvers for different angle of attacks and slip angles. The iterative LAPACK BGS solver is around two to three times faster than the direct LAPACK solver. Figure 6.6 also shows that LAPACK BGS does not converge for slip angles higher than 6 degrees and lower than -7 degrees.

The conclusion is that the iterative LAPACK BGS solver is the best option for the current Fokker100

model due to its superior performance in runtime, while there are no notable differences in the results.

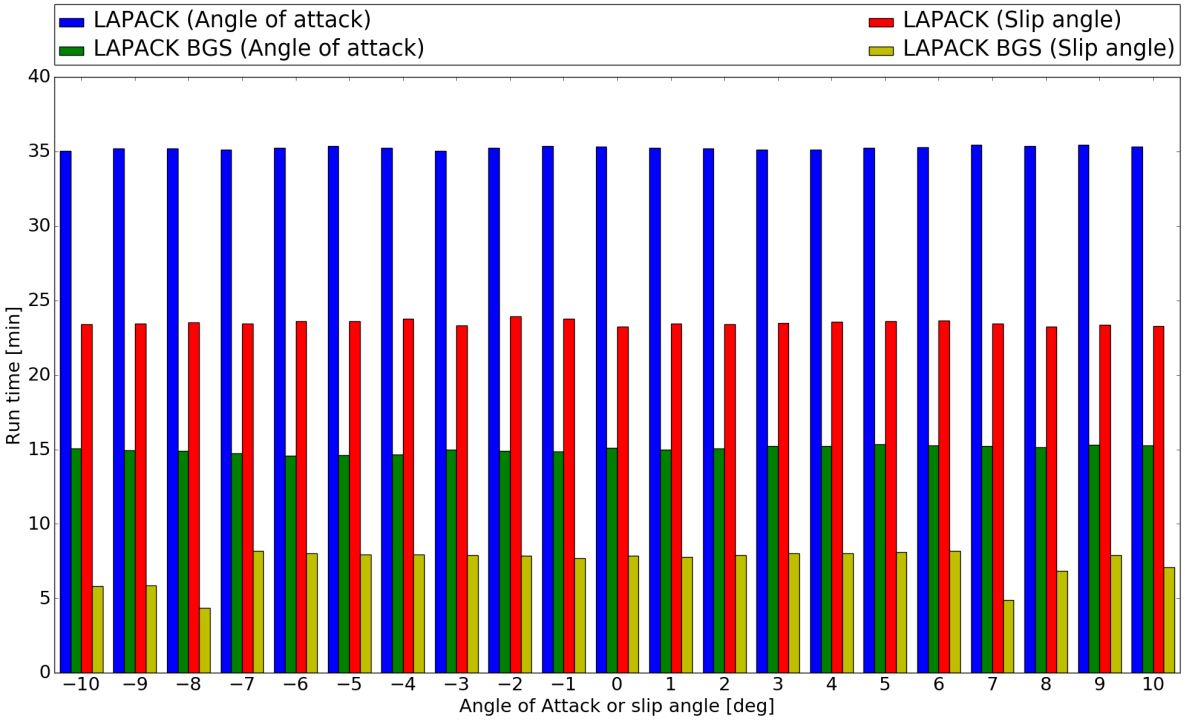


Figure 6.6: Run times for Fokker100 model with different solvers.

6.2.2. Iterations

The first order panel method tools like VSAERO requires that the body and the wake of the aircraft to be modelled. The initial wake model is most of the time a rough estimation of the shape of the wake with no wake relaxation. The solution produced by VSAERO while using this model can be improved by specifying multiple wake iterations NWIT and/or viscous potential iterations NVPI. Note that wake relaxation is computationally less expensive than viscous calculation [18].

Wake Iterations

A wake iteration improves the shape and the coefficients of the wake panels by changing the positions of the streamwise edges of the wake panels. The re-positioning of the streamwise edges are conducted by re-aligning the edges with the local flow directions.

According to Nathman [18], a rigid wake(NWIT=0) results are in many cases very close to the relaxed wake results. He suggests that if a relaxed wake is desired, NWIT may be set to 1, 2, or 3. If convergence is not indicated by the first four iterations (including the rigid wake iteration), higher values of NWIT are not recommended. Instead, the underlying reasons for oscillation in the solution must be pursued.

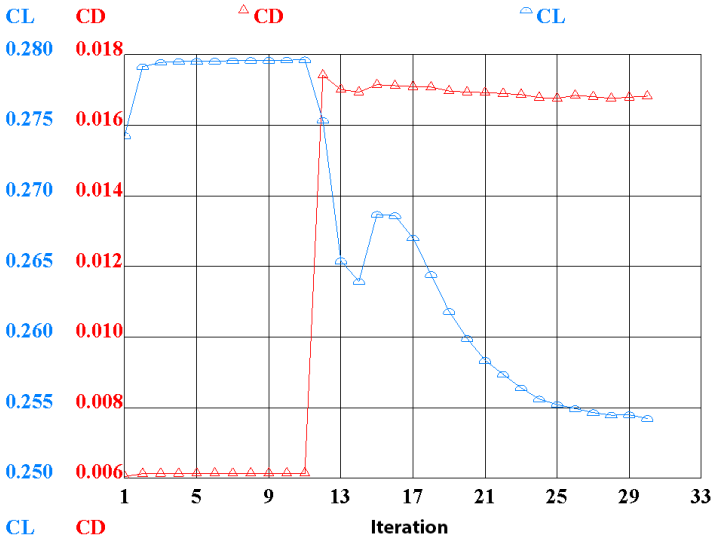
Viscous Iterations

The viscous iteration uses the integral boundary layer calculations to compute the boundary layer properties. For each iteration, the surface transpiration velocities are computed and are returned to the potential flow code. Viscous calculation can only be performed if the streamlines around the surface of the aircraft are specified. The viscous calculation utilises the velocity and convergence along the length of the streamlines to compute the boundary layer properties.

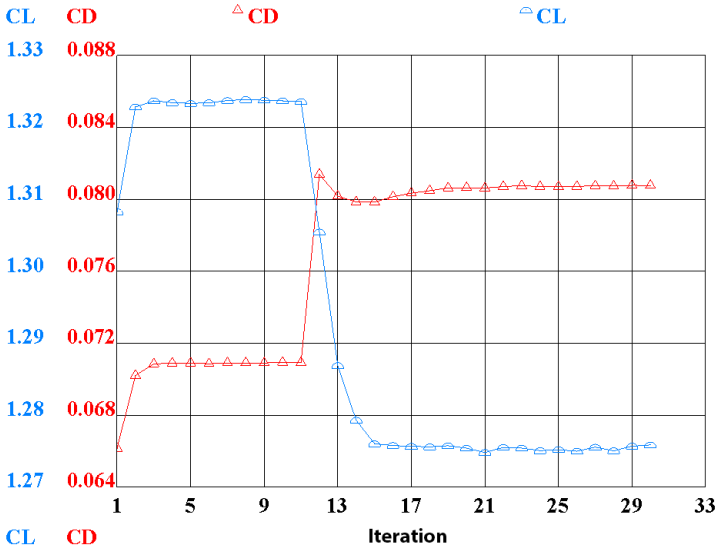
Nathman [18] says that the viscous iterations usually converges quite quickly and that NVPI of 3 or 4 is a good initial value. Any oscillation beyond NVPI of 4 signals a problem in the streamline tracing or has significant amounts of separated flow on the body.

Results

Figures 6.7 and 6.8 show the lift and the drag coefficients for four different cases. The four sub-figures are created by varying the angle of attack and the BLWIT input parameters. The BLWIT parameter indicates if the viscous iterations are performed after the wake iterations (BLWIT = 0) or must it alternate between the viscous and the wake iterations (BLWIT = 1). The results of figures 6.7 and 6.8 are generated using the LAPACK BGS solver. Note that the SOLRES or the solution residual input parameter for LAPACK BGS affects the convergence rate of the wake iterations.



(a) Angle of attack = 0, BLWIT = 0

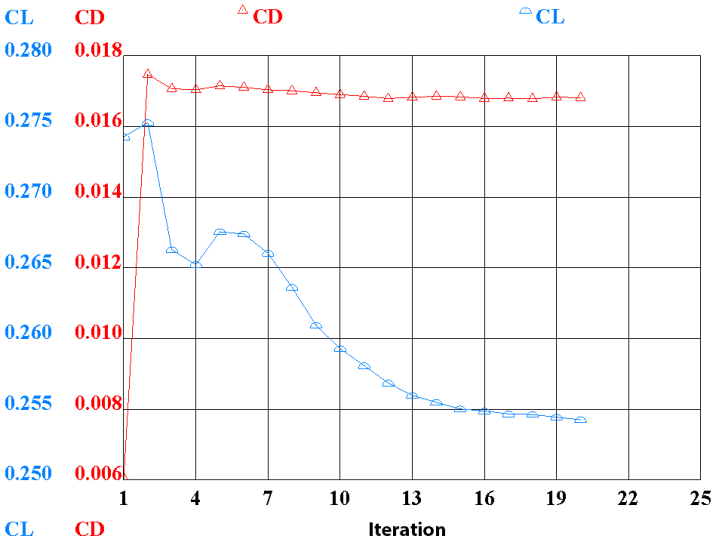


(b) Angle of attack = 10, BLWIT = 0

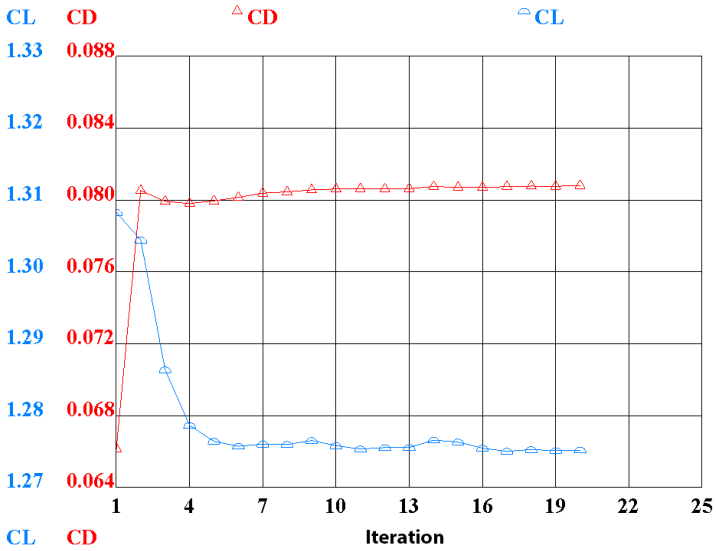
Figure 6.7: Lift and drag coefficient versus number of iterations.
BLWIT = 0, Body panels = 9493, Wake panels = 10642, NWIT = 10, NVPI = 20

Figures 6.7a and 6.7b show the convergence rate of the lift and the drag coefficients. The first eleven iterations show the convergence for the wake iterations, while the remaining twenty iterations show the convergence of the viscous iterations. The figures show that the lift and drag coefficients are indeed converged within four iterations for the wake iterations, while the convergence of the viscous iterations is very dependent on the angle of attack. The number of iterations needed for convergence of the viscous iterations decreases when the angle of attack is increased.

Comparison of figures 6.7a and 6.7b with figures 6.8b and 6.8b show that the final values of the lift and drag coefficients are almost the same. This means that the impact of BLWIT is negligible. BLWIT is



(a) Angle of attack = 0, BLWIT = 1



(b) Angle of attack = 10, BLWIT = 1

Figure 6.8: Lift and drag coefficient versus number of iterations.
BLWIT = 1, Body panels = 9493, Wake panels = 10642, NWIT = 10, NVPI = 20

equal to 0 is chosen for further computations due to the separation of the wake and viscous iterations. The convergence of the wake and viscous iterations can then be checked independently of each other.

The total run time of 10 wake iteration is 17.2 minutes, which is approximately 103 seconds for one wake iteration. The total run time of 20 viscous iterations is 4.8 minutes, which is approximately 14.5 seconds for one viscous iteration. This shows that viscous iteration is significantly faster.

The conclusion is that wake iterations converge very quickly, while the convergence of the viscous iteration is dependent on the attitude of the aircraft. Also, the run time is predominantly determined by the number of wake iterations.

6.3. Model Settings

A major contribution to the accuracy of the aerodynamic results is the method for constructing the VSAERO model. There are various ways to change construction method or settings of the model. This section will look at the effect of the mesh density, wake line curvature and congruent versus incongruent mesh.

6.3.1. Panel Density

A VSAERO model consists of two type of panels. The first panel type is called the body panels and these panels represent the geometry or the skin of the aircraft. The second panel type is called the wake panels and these panels represent the wake that has been shed of the aircraft.

The panel density of the aircraft affects the computation time and the accuracy of the results. A model with too few panels will produce poor aerodynamic data, while a model with too many panels will increase the run time for computing the aerodynamic data of the aircraft.

Table 6.3 shows the body panel budget for a typical conventional aircraft. The panel budget given in table 6.3 is for half an aircraft. The panel budget for the whole aircraft is thus the double number of panels.

Table 6.3: Body panel budget on one side of symmetry for a typical conventional aircraft in cruise condition[18].

Component	Chordwise	Spanwise	Panels
Wing	80	20 - 25	1600 - 2000
Horizontal Tail	60	15	900
Fin	30 (one side only)	12 - 15	400
Body	40 (x direction)	18 - 24	2000
Pylon	40	15	600
Nacelle	60 (x direction)	24	1250

The total amount of wing panels can be increased or decreased by changing the panel distribution in the chordwise and/or in the spanwise direction. The effects of the panels density in the chordwise and the spanwise direction will be investigated further below by only changing the density of the main wing.

Chordwise panel density

The chordwise panel density represents the number of panels that has been distributed over the local chordwise cross section of the wing to model the curvature of the airfoil. The chordwise panel density determines how accurate the airfoil curvature can be represented.

Figure 6.9 shows three different chordwise pressure distribution of the same model with different amount of chordwise panels. A comparison between figure 6.9a and figure 6.9c shows that the high density model produces a lot smoother pressure distribution than the low density model. The most significant differences are the prediction of the suction peak. The high-density model shows a steep suction peak, while the low-density model has a shallow suction peak.

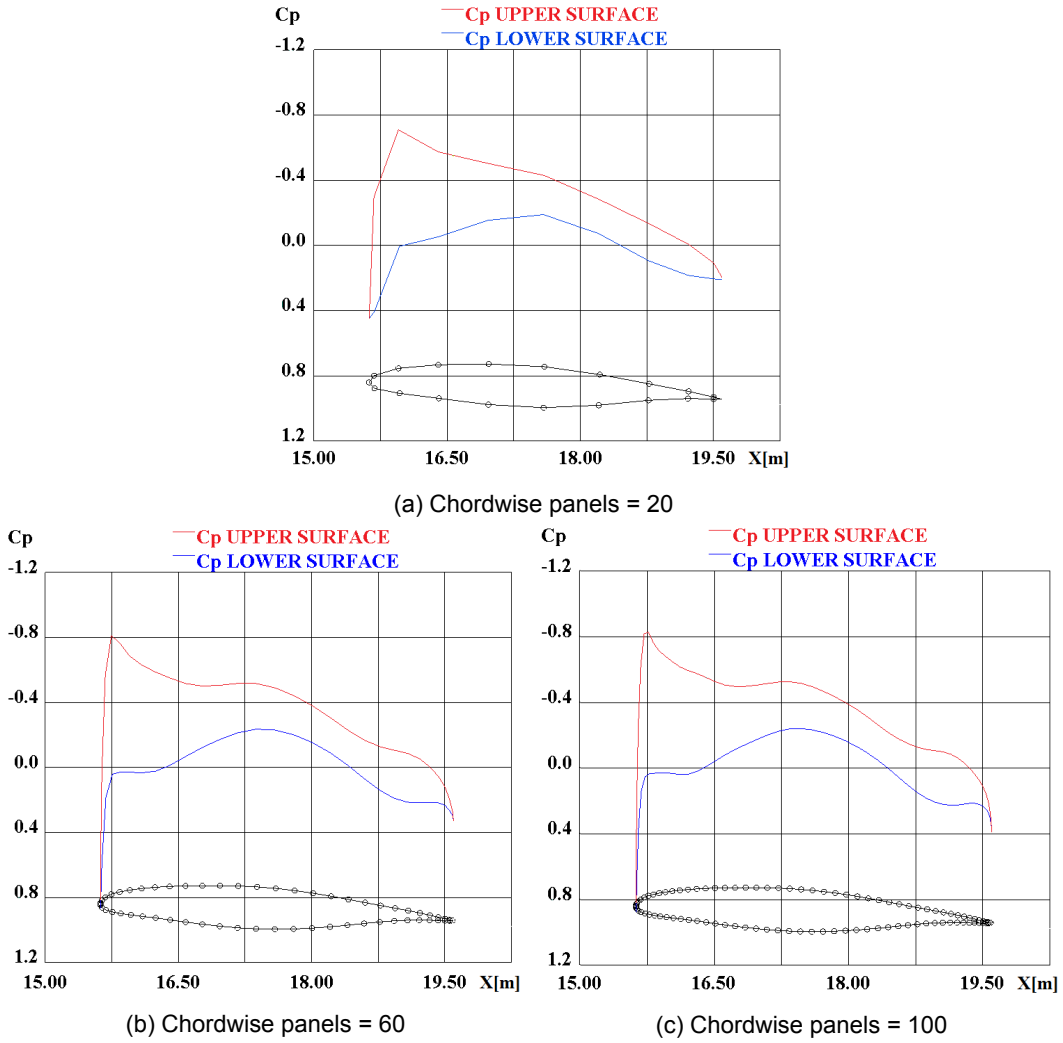
The effects of the different pressure distribution on the performance of the Fokker100 model can be clearly seen in the drag polar of figure 6.10. For each drag polar curve of figure 6.10, a curve fitting has been made using equation 6.1. C_D^* and C_L^* represents the drag and lift coefficients at minimum drag, while k is the coefficient that represents the shape of drag polar curve. The values of C_D^*, C_L^* and k for the drag polar curves are shown in table 6.4.

$$C_D = C_D^* + k(C_L - C_L^*)^2 \tag{6.1}$$

Table 6.4: Drag polar trend lines data of figure 6.10

	Chordwise panels			Reference
	20	60	100	
C_D^*	0.0187	0.0149	0.0141	0.0193
k	0.0412	0.0421	0.0421	0.0441
C_L^*	0.0934	0.0308	0.0214	0.0170

Table 6.4 shows that the drag polars of the models become more similar to the drag polar of the reference data when the chordwise panel density is increased (increase in k). It also shows that the C_L^* becomes more accurate when the panel density is increased. But the minimum drag coefficient, however, decreases, when the panel density is increased. The effects of the increasing panel density can



(a) Chordwise panels = 20

(b) Chordwise panels = 60

(c) Chordwise panels = 100

Figure 6.9: Chordwise pressure distribution of main wing at $b = 5m$ and at $\alpha = 0deg$

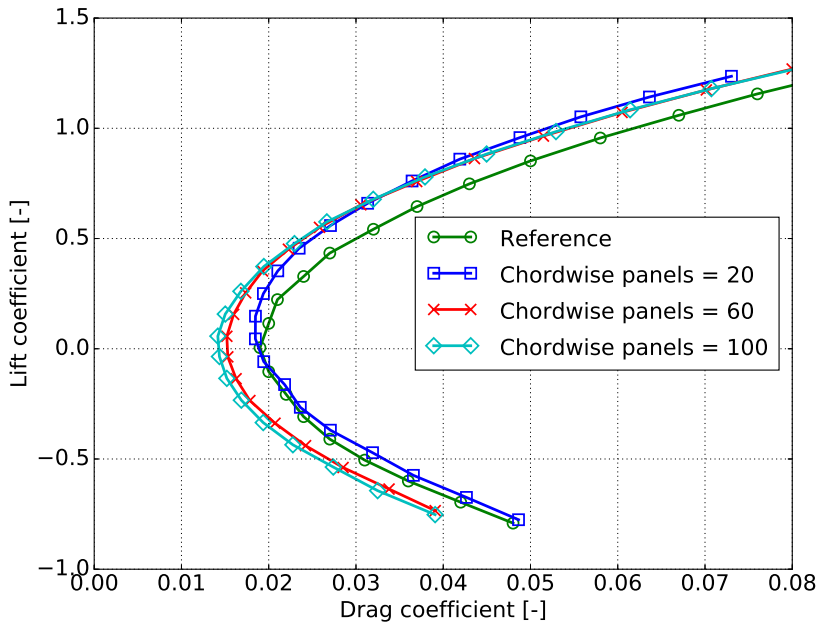


Figure 6.10: Drag polar for varying chordwise panel density

be explained due to the different suction peaks of the models as shown in figure 6.9. The increase in the panel density of the models will produce a pressure distribution with a higher suction peak at the nose. A higher suction peak means an increase in the lift coefficient and decrease in the drag coefficient.

It can be concluded that the chordwise panel density affects the pressure distribution over the airfoil especially the suction peak at the leading edge. The increase in panel density means that the model can represent airfoil curvature more accurately. A more accurate airfoil means that the shape of the pressure distribution can be predicted more accurately. The number of chordwise panels that is being used for creating the Fokker100 models in this thesis is 100 chordwise panels.

Spanwise panel density

The spanwise panel density determines the number of panels in the spanwise direction of the wing. This density also influences the number of wake panels in the spanwise direction indirectly since the wake is shed by the wing and thus the number of wake panels in the spanwise direction is equal to the number of spanwise body panels.

The effect of varying spanwise panel density predominantly affects the root and tip of the wing. Figures 6.11 and 6.12 shows the spanwise lift distribution of the right wing at the angle of attack of 0 and 10 degrees respectively. The root lift distribution is significantly affected at low angle attack, but the effect is very small at high angle of attack. The tip effect is exactly the opposite. The effect is very small at low angle of attack, while the effect is significant at high angle of attack.

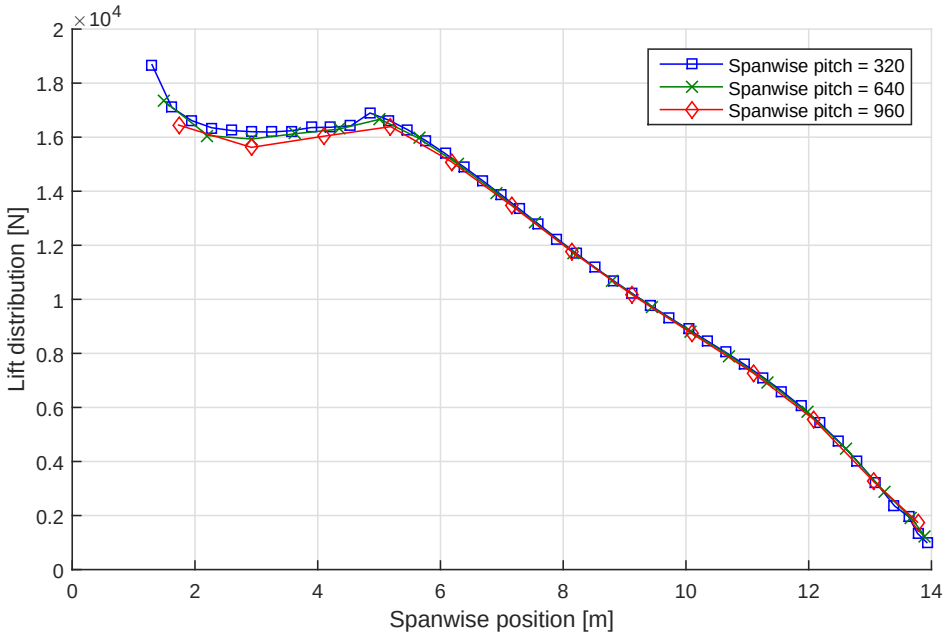


Figure 6.11: Spanwise lift distribution of right wing at $\alpha = 0$ deg, pitch in mm

The impact of varying spanwise panel density on the aircraft performance is greater when the aircraft is slipping at an angle. Figure 6.13 shows this impact on the side force coefficients. Table 6.5 shows the corresponding C_{Y_β} for the curves of figure 6.13. This table shows that the first order derivative C_{Y_β} becomes more accurate when the number of spanwise panels is increased (decrease in pitch). The cause of the increase of the accuracy for C_{Y_β} is shown in figure 6.14. According to the figure, the impact of different pitches between the mesh points is very different for the right and the left wing. The wing that is partially shielded by the fuselage is heavily influenced by the varying panel density. This wing has a smoother lift distribution when the spanwise panel density is increased.

The conclusion is that the spanwise panel density affects the root and tip performance of the wing. The impact of the panel density on the root lift distribution is only relevant at low angle of attack or at side slip angles, while the tip performance is only relevant at high angle of attack. A higher spanwise density at the root and at the tip produces a smoother spanwise lift distribution with higher lift at the root and lower lift at the tip. The spanwise pitch of 480 millimetres is being used for creating the Fokker100 models in this thesis.

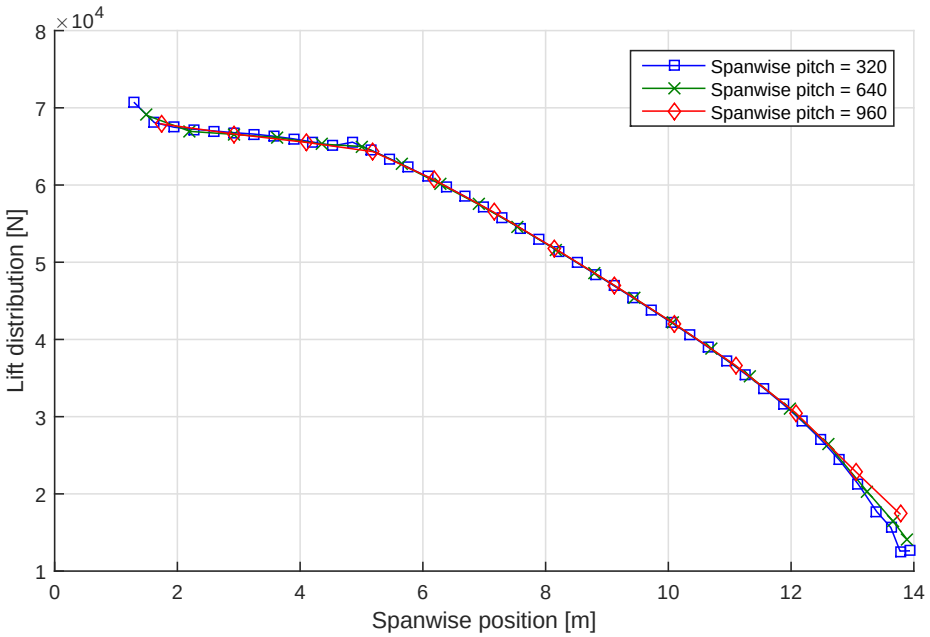


Figure 6.12: Spanwise lift distribution of right wing at $\alpha = 10$ deg, pitch in mm

Table 6.5: C_{Y_β} in [1/rad] for curves of figure 6.13

	Spanwise pitch			Reference
	320	640	960	
C_{Y_β}	-0,688	-0,670	-0,596	-0,733

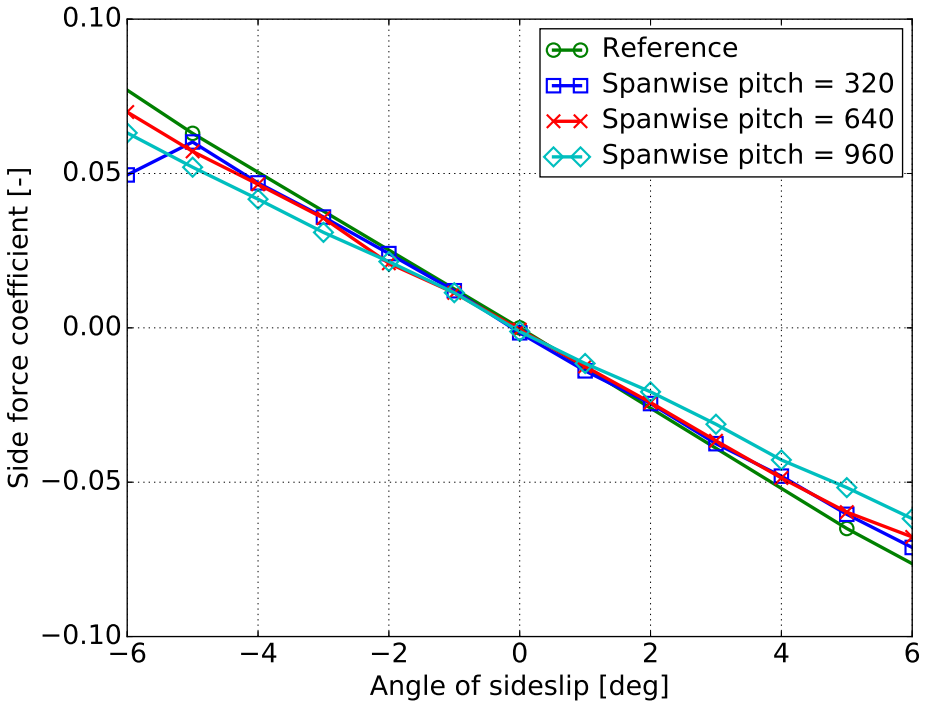
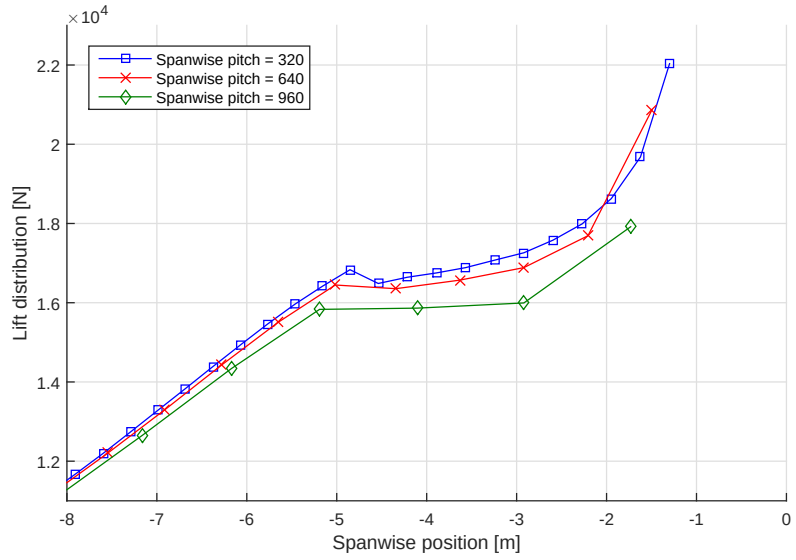
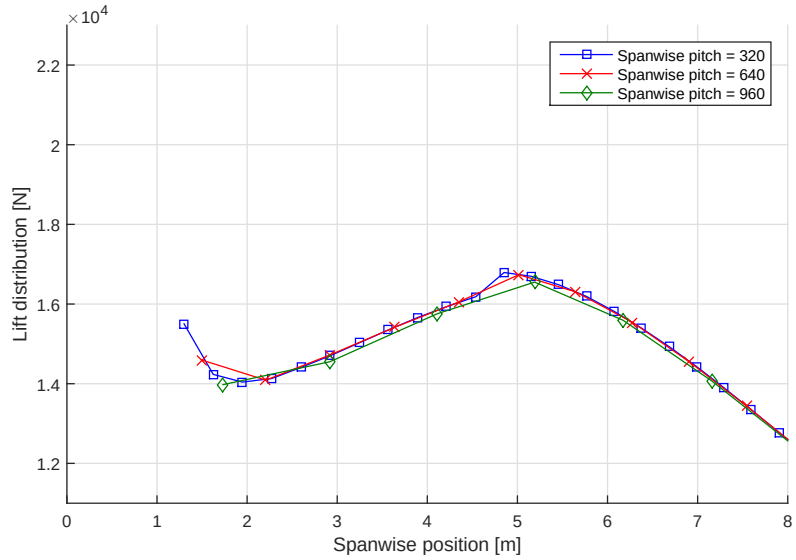


Figure 6.13: Side force curve for varying spanwise panel density at $\alpha = 0$ degree, pitch in mm



(a) Left wing



(b) Right wing

Figure 6.14: Spanwise lift distribution at $\alpha = 0$ and $\beta = 5$ degree, pitch in mm

6.3.2. Wake lines

The wake model for a first order panel method like VSAERO is modelled by defining the wake lines. A wake line can be modelled in multiple ways as presented in figure 5.36. Two of the three modelling methods for creating the wake lines are investigated further. The first wake line model is a straight line from trailing edge parallel to the freestream direction, which corresponds to wake 'a' of figure 5.36. The second model is a curved line starting from the trailing edge in the average direction of the upper and lower trailing direction vectors. After the start point, the direction of the line will change until it is parallel to freestream direction.

Figures 6.15 and 6.16 show the convergence of the lift and the drag coefficients for a model with straight wake lines and for a model with curved wake lines respectively. The most striking aspect of these figures is that after five wake iterations, the differences in the lift and the drag coefficients between the two models are negligible. The differences in the coefficients between the two wake lines models can only be seen in the first calculation where there is no wake relaxation.

These result may suggest that if you have enough wake iteration, the shape of the wake lines becomes irrelevant, but be aware that this is not always true. A good example is a canard aircraft, where

the wake of the canard can intersect the main wing when wake relaxation is applied. The aerodynamic solver will become unstable if the wake is intersecting the main wing. To avoid this instability one must approximate the final wake line such that the wake line shape is not changed too much during the wake relaxation.

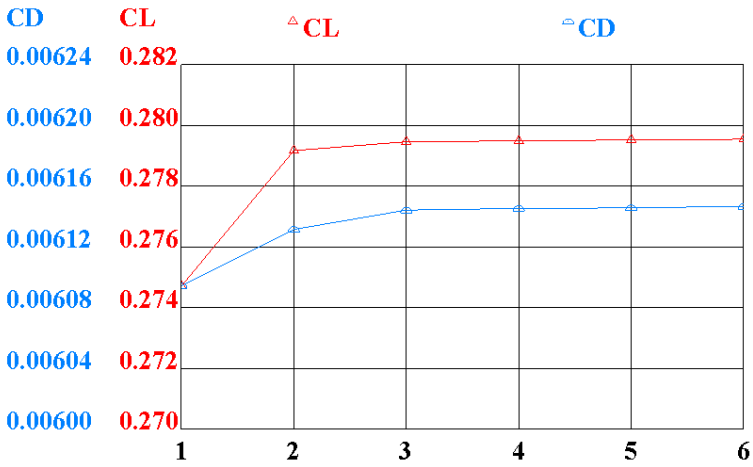


Figure 6.15: Straight wake lines for $\alpha = 0$ with 5 wake iterations

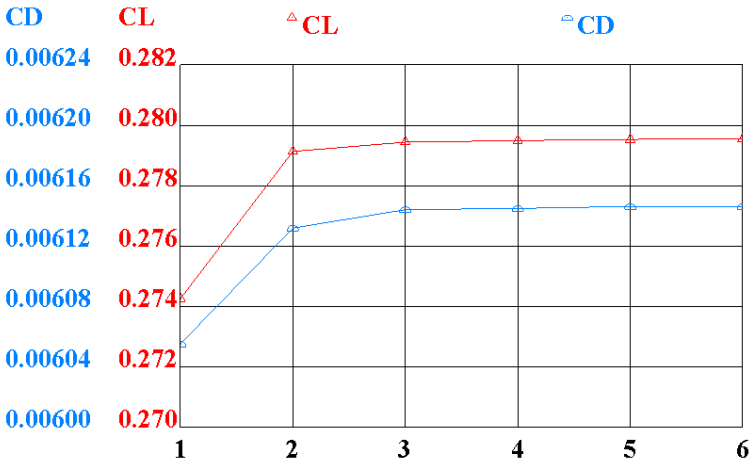


Figure 6.16: Curved wake lines for $\alpha = 0$ with 5 wake iterations

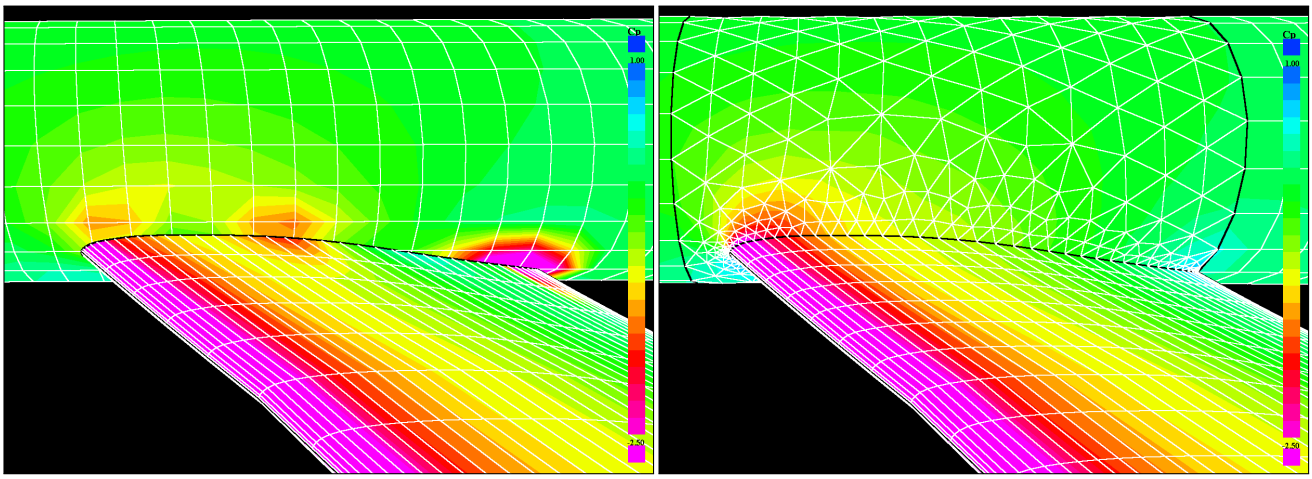
6.3.3. Congruent versus Incongruent

Discretizing an aircraft to a panel model can be performed in many ways. Section 5.3.1 has explained the two meshing methods that have been implemented in the MMG tool for discretizing an aircraft model.

The first method is the incongruent mesh method that meshes all components of the aircraft separately. All the individual mesh grids are then written away to the input file of VSAERO. This will result in a panel model that has an incongruent mesh between the components. Or in other words, the meshes of the two neighbouring components are not aligned with each other at the intersection between the two components, see figure 6.17a.

The second method is the congruent mesh method that meshes all the aircraft components as one model. The resulting model is a model, where the meshes of two neighbouring components are aligned at the intersection between the two components, see figure 6.17b.

The difference between the two methods is clearly shown in the pressure plot of figure 6.17. The figure shows that the difference between the two methods is related to the interaction between the two



(a) Incongruent mesh (b) Congruent mesh

Figure 6.17: Pressure distribution at the intersection between wing and fuselage at $\alpha = 10$ deg

components. The incongruent meshing method does a very poor job in predicting the wing-fuselage interaction, while the results of the congruent meshing method seem plausible.

The lift and drag curve of the two methods are shown in figures 6.18 and 6.19. The lift curve shows that lift curve slope of the incongruent mesh is lower than the lift curve slope of the congruent mesh. The figure also suggests the lift curve of the congruent mesh is approximating the reference curve way better than the lift curve of the incongruent mesh. The underestimation of the lift at high angle of attack is also reflected in the drag curve as a lower drag value since lower lift means lower induced drag.

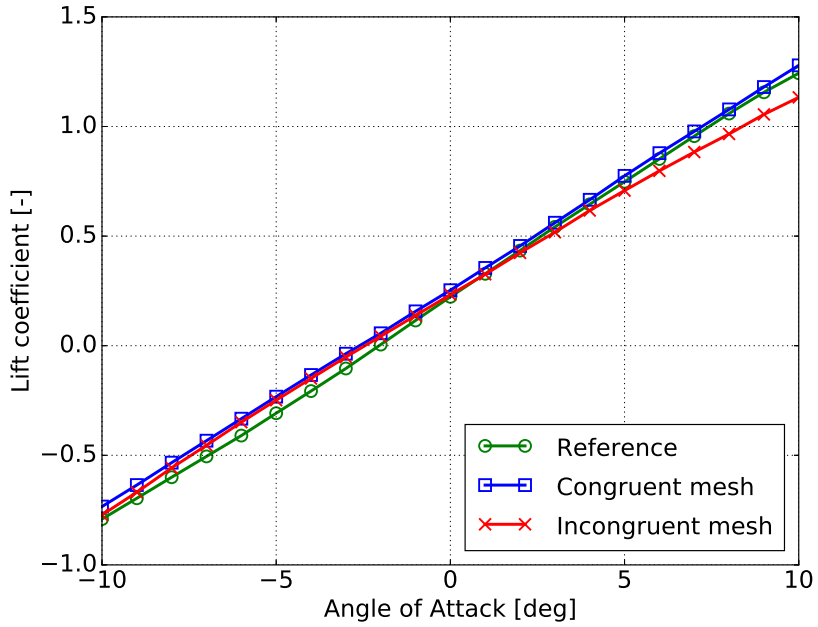


Figure 6.18: Lift curve of congruent and incongruent mesh

The cause of the lower lift can be observed in the chordwise pressure distribution at the root of the wing, see figure 6.20a. The weird pressure fluctuation at wing root of figure 6.20a is also reflected in the pressure distribution. The weird pressure fluctuation causes the pressure to drop temporarily after the first suction peak. After the pressure drop, the pressure rises to such a level that the local area of the wing produces negative lift. This negative lift is the cause of lower lift and drag at high angle of attack.

It can be concluded that the congruent meshing method is the better method to discretize a aircraft model to a panel model since the incongruent meshing method is incapable in predicting the interaction of the two neighbouring components such as the wing and fuselage.

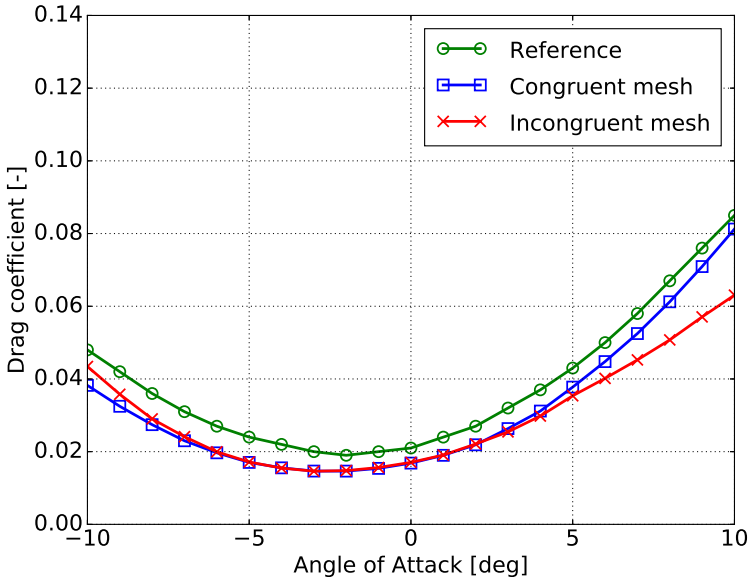
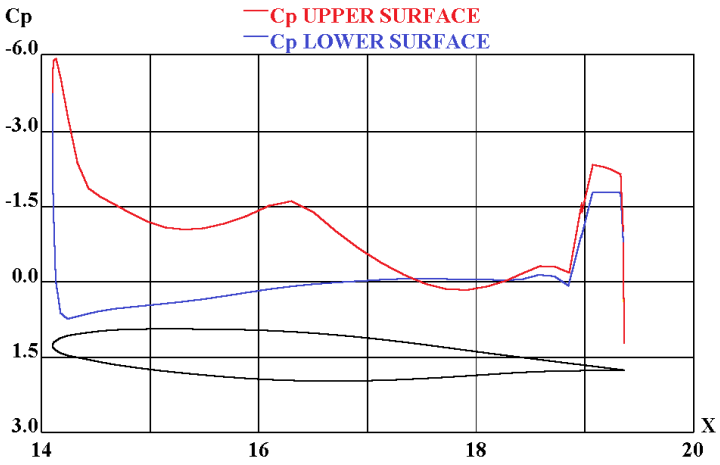
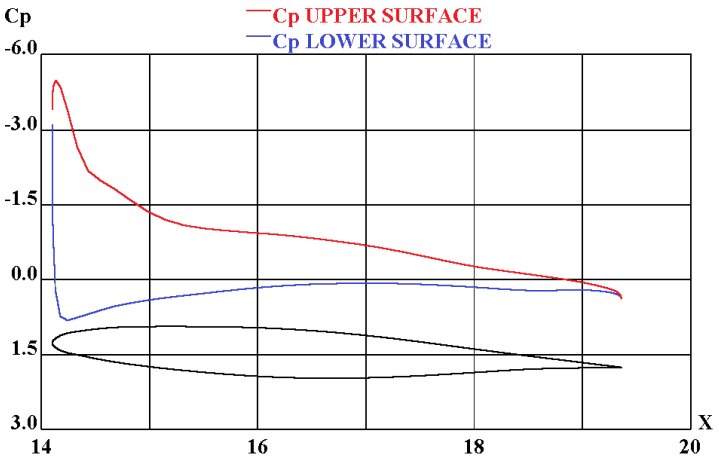


Figure 6.19: Drag curve of congruent and incongruent mesh



(a) Incongruent mesh



(b) Congruent mesh

Figure 6.20: Pressure distribution near root of main wing, α = 10 deg and b = 1.75m

6.4. Longitudinal Characteristics

The validation of the models is performed in this section and in section 6.5. This section analyses the longitudinal characteristics of four different Fokker100 models. One of the models is a clean aircraft with no moveables, while the other models have wings with moveables. The characteristics of the models are validated against the reference data from the Fokker report [20].

The moveable models used for the simulations has two ailerons, two elevator and one rudder. Figure 6.21 shows the locations and the size of the moveables on the aircraft as well as the mesh panels. The mesh of the models is created using the settings specified in table 6.6. These models are tested under the conditions specified in table 6.1.

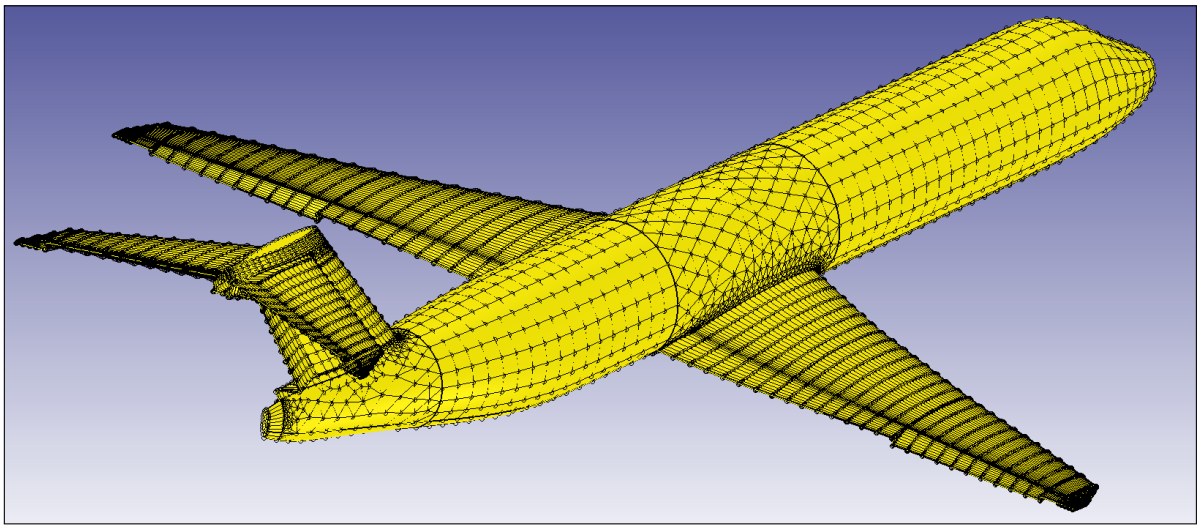


Figure 6.21: Meshed Fokker100 with not-slotted moveables and transition surfaces

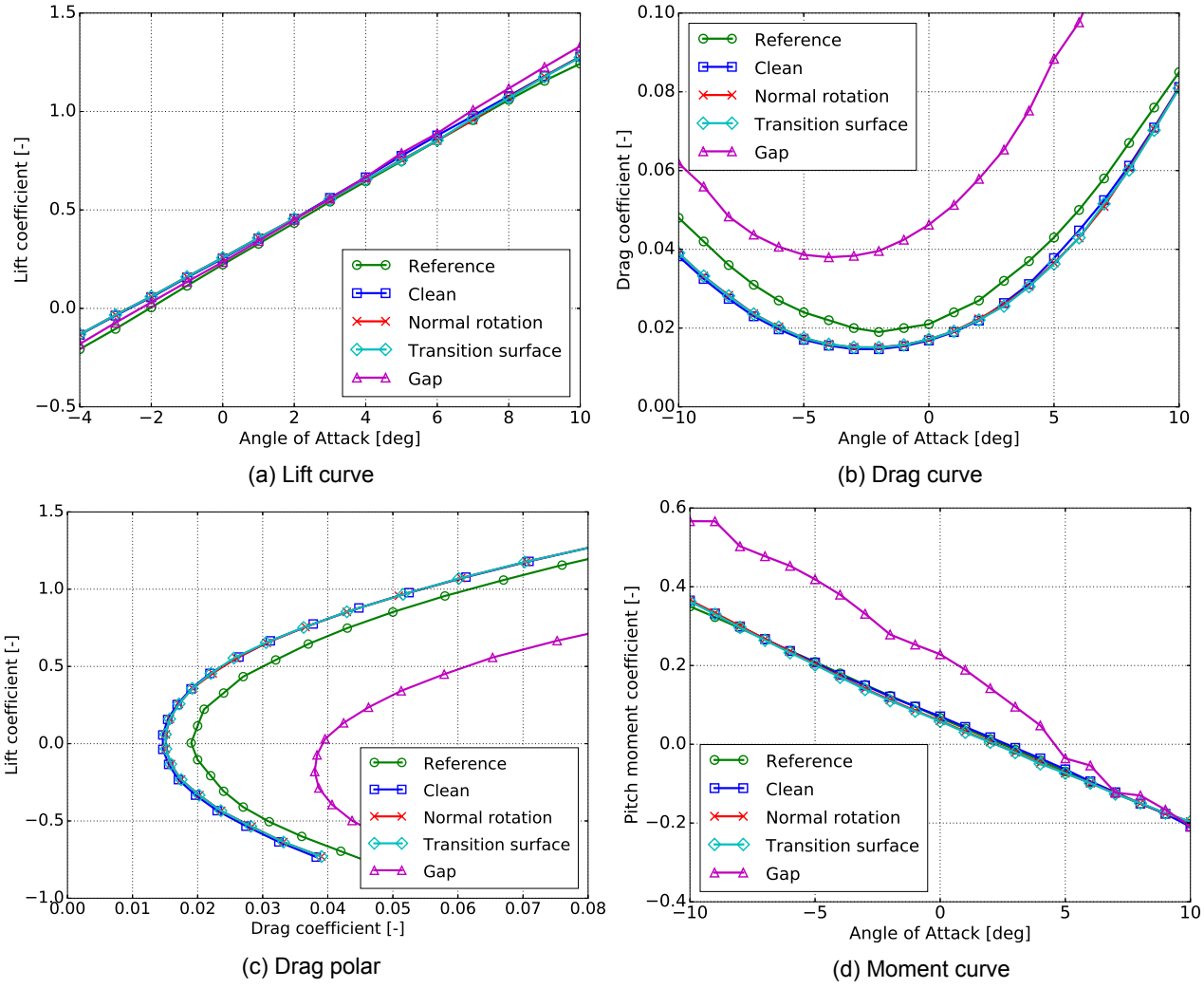
Table 6.6: Mesh and VSAERO settings for the four Fokker100 models.

	Clean	Moveable modelling method		
		Normal rotation	Transition surface	Gap
Main wing				
Chordwise panels	40	42	42	42
Spanwise pitch	480	480	480	480
Horizontal tail plane				
Chordwise panels	30	31	31	13
Spanwise pitch	400	400	400	400
Vertical tail plane				
Chordwise pitch	130	130	130	130
Spanwise pitch	400	400	400	400
Fuselage				
Chordwise pitch	130	130	130	130
Spanwise pitch	400	400	400	400
Aircraft				
Body panels		11270	11964	11974
Wake panels		12818	13778	16695
Solver	LAPACK BGS			
NWIT	5			
NVPI	20			

6.4.1. Zero Deflection

The first step in the validation process is to simulate the results of the Fokker100 models with zero deflection for the moveables. By doing this first, one can check the feasibility of the different modelling methods for the moveables by comparing the results of the moveables models with the results of the clean wing model.

Figure 6.22 shows the longitudinal characteristics of the models such as the lift curve, drag curve, drag polar and the moment curve. The trend lines of the lift curve, drag polar and the moment curve is shown in table 6.7. C_D^* represents the minimum drag or the zero-lift drag of the model, while k is the coefficient that represents the shape of drag polar curve. C_D^* and k represents the parameters of equation 6.1. Table 6.7 and figure 6.22 show that the difference between the different models is negligible except for the model with the 'gap' option.



(a) Lift curve

(b) Drag curve

(c) Drag polar

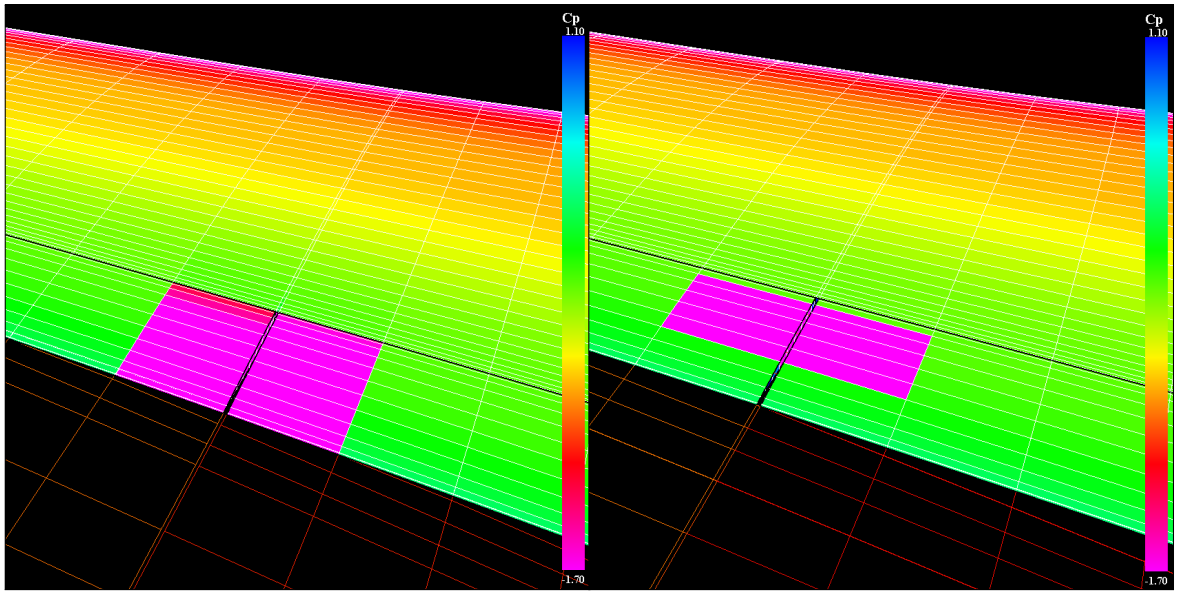
(d) Moment curve

Figure 6.22: Longitudinal characteristics of Fokker100 model with zero deflections

Table 6.7: C_{L_α}, C_D^*, k and $C_{m\alpha}$ for curves of figure 6.22, derivatives are in [1/rad]

	Clean	Normal rotation	Transition surface	Reference
C_{L_α}	5,770	5,724	5,724	5.936
C_D^*	0.0143	0.0147	0.0146	0.0195
k	0.0422	0.0423	0.0424	0.0436
C_{m_α}	-1,587	-1,604	-1,604	-1,593

The 'gap' moveable option models the moveables as part of the wing with gaps between the sides of the moveable and the wing. The presence of these gaps creates a low pressure area around the gap as shown in figure 6.23a. According to Katz [19], the potential flow model does not account for the viscosity of the flow. This lack of viscosity of the flow will increase the velocity inside the gap to unrealistic values, while in reality the velocity increase is countered by the friction of viscous flow with the skin of the wing. An attempt has been made to solve this problem by covering the gap with two wake sheets. One wake sheet covers the top side of the gap, while the second wake sheet covers the bottom side of the gap. The result of this attempt is shown in figure 6.23b. This attempt is not very successful since there is still a low pressure area present around the gap. The presence of the low pressure area around the gap creates a suction force; that significantly increases the drag of the aircraft as shown in the drag curve of figure 6.22b. **The increase of drag is so significant that the author concludes that the "gap" moveable option is not a good approach to model the moveables of an aircraft for a first order panel method.**



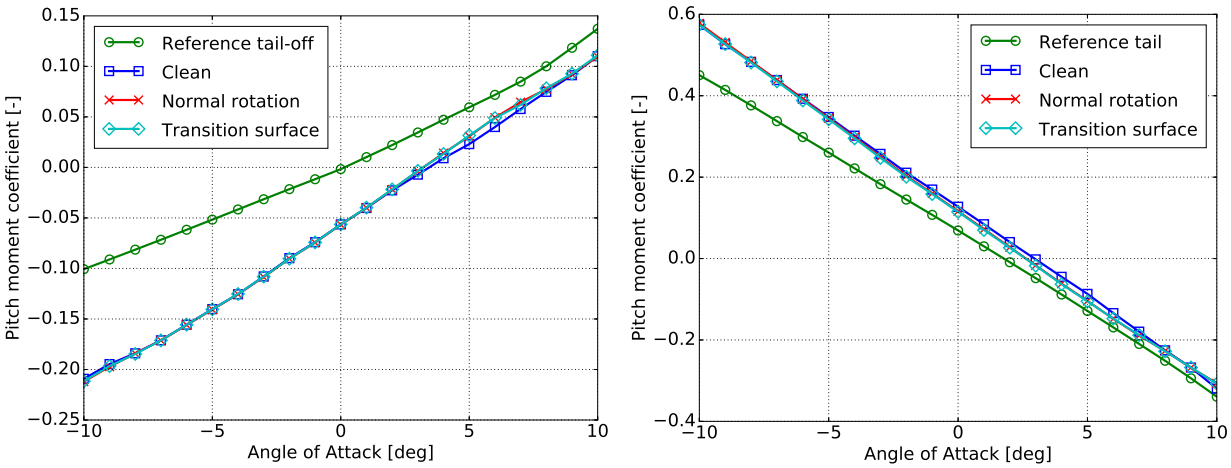
(a) No wake sheets covering the gap (b) Wake sheets covering the gap

Figure 6.23: Pressure distribution for wing with "gap" option

Figure 6.22a shows that the zero lift angles of the three remaining models are approximately 0.7 degrees lower than the zero lift angle of the reference data. Table 6.7 shows that the lift curve slopes of the three remaining models are predicted lower than the lift curve slope of the reference data. Table 6.7 also shows that the minimum drag coefficients are underestimated by a value of approximately 0.005. The underestimation of the minimum drag coefficients are attributed to the lack of the twin turbofan engines, engine pylons and the dorsal fin of the Fokker100 model.

The moment curves of figure 6.22d suggest that the moment curves of the Fokker100 models are very accurate but keep in mind that this is the total moment curve of the whole aircraft. The individual components can have moment curves that are not so accurate as the total moment curve. Figure 6.24 shows the tail-off moment curve and the moment curve of the T-tail of the Fokker100. These curves clearly show that individual components deviate quite clearly from the reference moment curve.

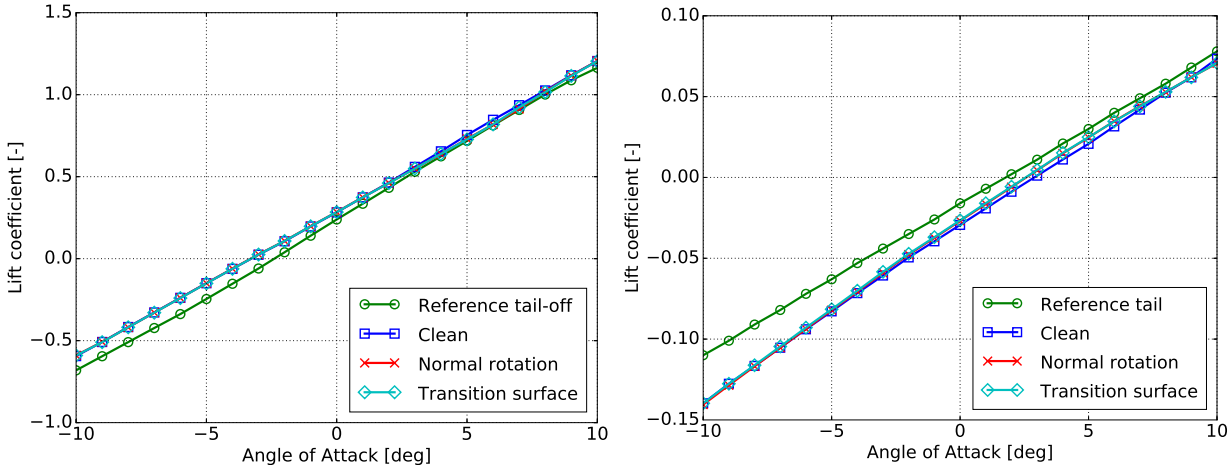
The deviation of the moment curve is caused by the overestimation of the lift curves of the main wing and the horizontal tail as shown in figure 6.25. The probable cause of the overestimation of the lift curves may be attributed to the underestimation of the boundary layer effects over the aircraft. Since the trend of the drag curve is almost the same as the reference, while the drag is too low. A stronger boundary layer effects increase the boundary layer thickness, which in turn decreases the lift by decambering the airfoil. A thicker boundary layer means more skin friction which increases the drag and decreases the difference between the drag coefficients of the models and the reference data.



(a) Tail-off moment curve

(b) Tail moment curve

Figure 6.24: Moment curve of tail-off and tail configuration



(a) Tail-off lift curve

(b) Tail lift curve

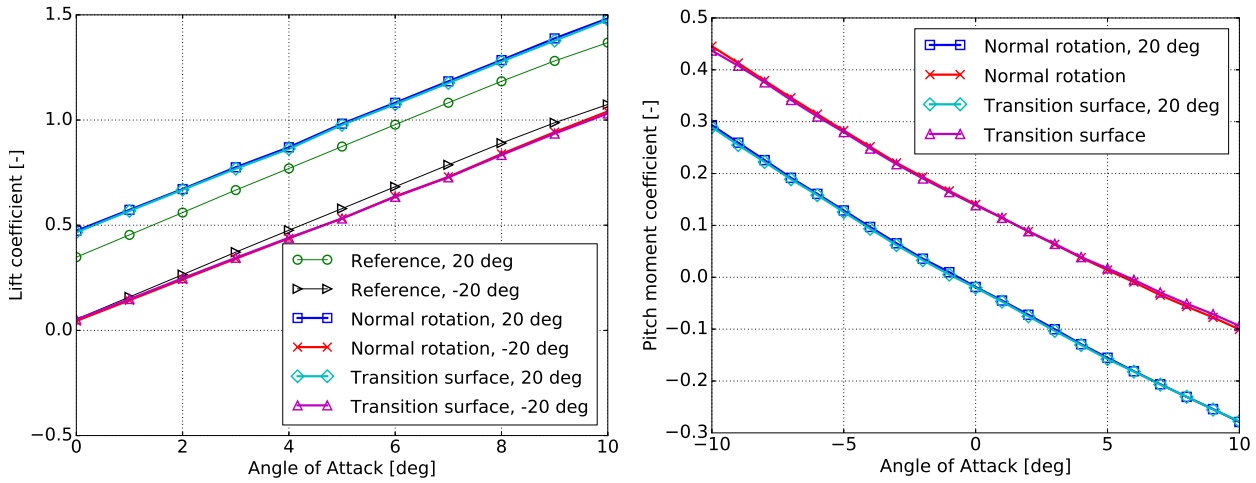
Figure 6.25: Lift curve of tail-off and tail configuration

6.4.2. Moveable Deflection

The next step in the validation process is to examine the effect of the control surfaces on the performance of the different models. The control surfaces that are examined in this section are the ailerons. The reason for choosing the ailerons is that is located on the main wing and the main wing is the main contributor to the lift and drag of the aircraft. The change in the lift and drag is thus most visible when the ailerons are deflected. Note that the ailerons in this section have the same deflection angle for the left and the right wings.

Figure 6.26 shows the impact of the aileron deflections on the lift and moment curve of the aircraft. The lift and moment curve for the two different moveable models are almost identical to each other, but the lift curve of 'transition surface' moveable model is slightly lower than the lift curve of the 'normal rotation' moveable model. Besides that, no other differences between the two models can be identified when observing the lift and moment curves.

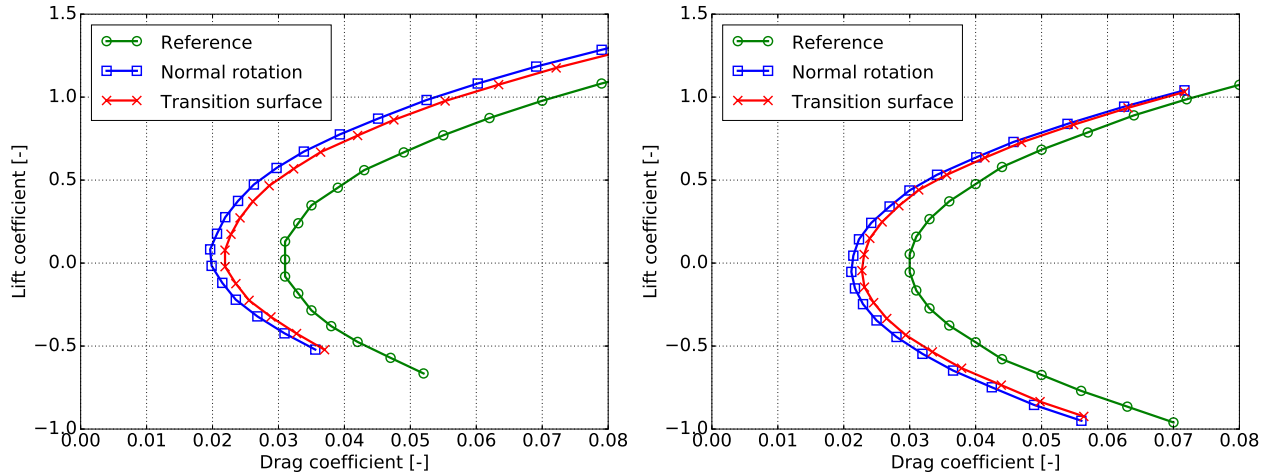
Figure 6.26a also shows two reference curve with an aileron deflection of 20 and -20 degrees. Comparing these reference curves to the lift curves of the models and it tells us that the positive aileron deflection has greater error margin than the negative aileron deflection. The bigger error margin is attributed to the fact that lift of the wing is overestimated as shown in section 6.4.1. This means that lift of the deflected ailerons is also overestimated. The difference between the positive deflection and the negative deflection is that the lift of the positive aileron deflection increases the total lift of the aircraft,



(a) Lift curve (b) Moment curve

Figure 6.26: Lift and moment curve for deflected ailerons

while the negative aileron deflection decreases the lift. So the overestimation of the lift is thus enhanced when the aileron is deflected positively and the overestimation of the lift is suppressed when the aileron is deflected negatively.

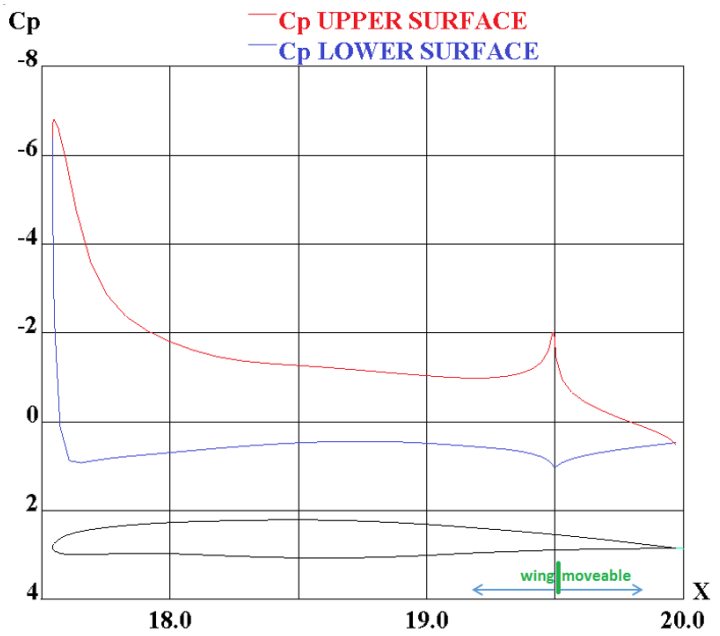


(a) 20 degree of aileron deflection (b) -20 degree of aileron deflection

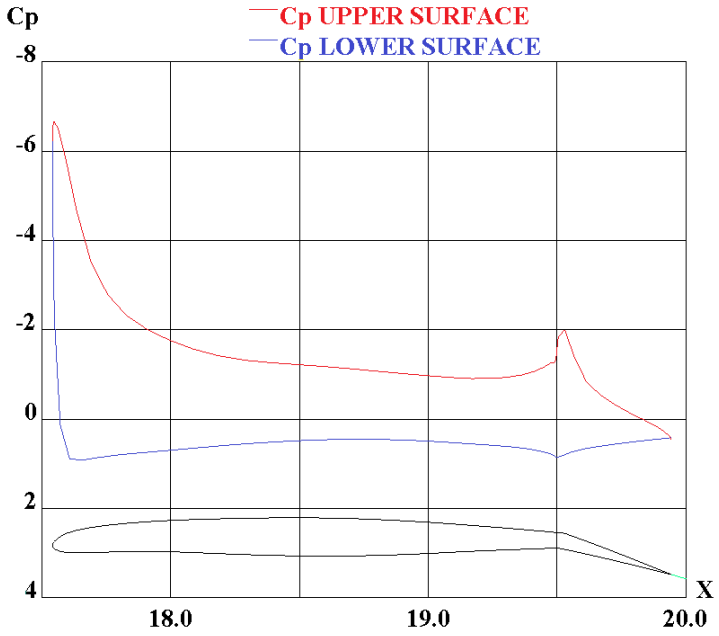
Figure 6.27: Drag polars for deflected ailerons

The only difference between the two moveable models can be observed in the drag polars of figure 6.27. The figures show that the drag of the 'transition surface' moveable model is higher for both the deflection angle of 20 and -20 degrees and this higher value decreases the error margin between reference data and drag polar of 'transition surface' model. The error margin in the minimum drag coefficients of the models with no deflected moveables was 0.005. The error margin of 'normal rotation' and the 'transition surface' models with 20 degrees of aileron deflection are 0.011 and 0.007 respectively. This means that error margins for 'normal rotation' and 'transition surface' models are increased by 0.006 and 0.003 respectively. The overall conclusion from figure 6.27 is that drag model of the 'transition surface' moveable model is more accurate than the drag model of the 'normal rotation' moveable model.

The difference in the drag between the moveable models can be explained by observing the chord-wise pressure distribution of the two models at span position of 10 meters, see figure 6.28. The pressure peak of 'transition surface' moveable seems to occur more to the aft of the wing when compared to the pressure peak of the 'normal rotation' moveable. The more aft position of the pressure peak generates a higher adverse pressure gradient at the trailing edge for the 'transition surface' model. A higher adverse pressure leads to earlier transition or separation of the boundary layer and thus an increase in skin friction drag.



(a) Normal rotation moveable



(b) transition surface moveable

Figure 6.28: Pressure distribution at span = 10m with aileron deflection of 20 degrees

The increase in skin friction drag could explain the higher drag value for the 'transition surface' model, but table 6.8 suggest that the skin friction only accounts for a small portion of the drag increase. The remaining drag increase must thus be attributed to a higher induced drag for the 'transition surface' moveable model.

This higher induced drag is caused the different locations of the suction peaks of the moveables. The suction peak of the 'normal rotation' model is located on the wing and not on the moveable itself, while the suction peak of 'transition surface' model is located on the moveable itself. This means that lift generated by the suction peak of 'normal rotation' model is pointed less to the aft and hence the lower induced drag.

Table 6.8: Drag and skin friction coefficients for two moveable options

	α=-10		α=0		α=10	
	CD	CDf	CD	CDf	CD	CDf
Normal rotation	0.036	0.011	0.026	0.011	0.101	0.011
Transition surface	0.037	0.011	0.029	0.012	0.105	0.012
Delta	0.001	0.000	0.003	0.001	0.004	0.001

6.4.3. Conclusion

The wing with moveable model with a gap between the sides of the moveable and the wing is unstable. The instability is caused by the presence of the gap. The presence of the gap creates a very low pressure area, which it accelerates the surrounding airflow. The surrounding airflow is accelerated to a ridiculous amount due to the fact VSAERO uses first order potential method for modelling the airflow. A first order panel method does not take into account the viscous effects. VSAERO does employ a boundary layer integral method to compute the viscous effects, but this method does not prevent the creation of the very low pressure area around the gap between the wing and the moveable.

The zero lift angles, the lift curve slopes and drag coefficients of the 'normal rotation' and the 'transition surface' models with zero moveable deflections are under estimated, while the moment curves were predicted quite accurately. The underestimation of the drag coefficient is attributed to the underestimation of the zero lift drag. The zero lift drag is underestimated because the models did not have the engines, engine pylons and the dorsal fin.

The differences in the lift, drag and moment curves between the 'transition surface' model and the 'normal rotation' model are very low and they are almost identical. A difference between the two models can be observed when both the left and the right ailerons are deflected to 20 degrees. The 'normal rotation' moveable models generate a slightly higher lift, but a much lower drag than the 'transition surface' moveable model. The difference between the 'normal rotation' and the 'transition surface' model is caused by the location of the suction peak of the moveable.

The suction peak of the 'normal rotation' model is located on the wing in front of the start of the moveable. The suction peak of the 'transition surface' on the other hand is located at the start or slightly behind the start of the moveable. The lift generated by the suction peak of 'normal rotation' moveable is pointed less to the aft when compared to the suction peak of 'transition surface' model. The results is a lower induced drag and a slightly higher lift for the 'normal rotation' model. It is therefore recommended to use the 'transition surface' model when only considering the longitudinal characteristics of the Fokker100 aircraft.

6.5. Lateral-Directional Characteristics

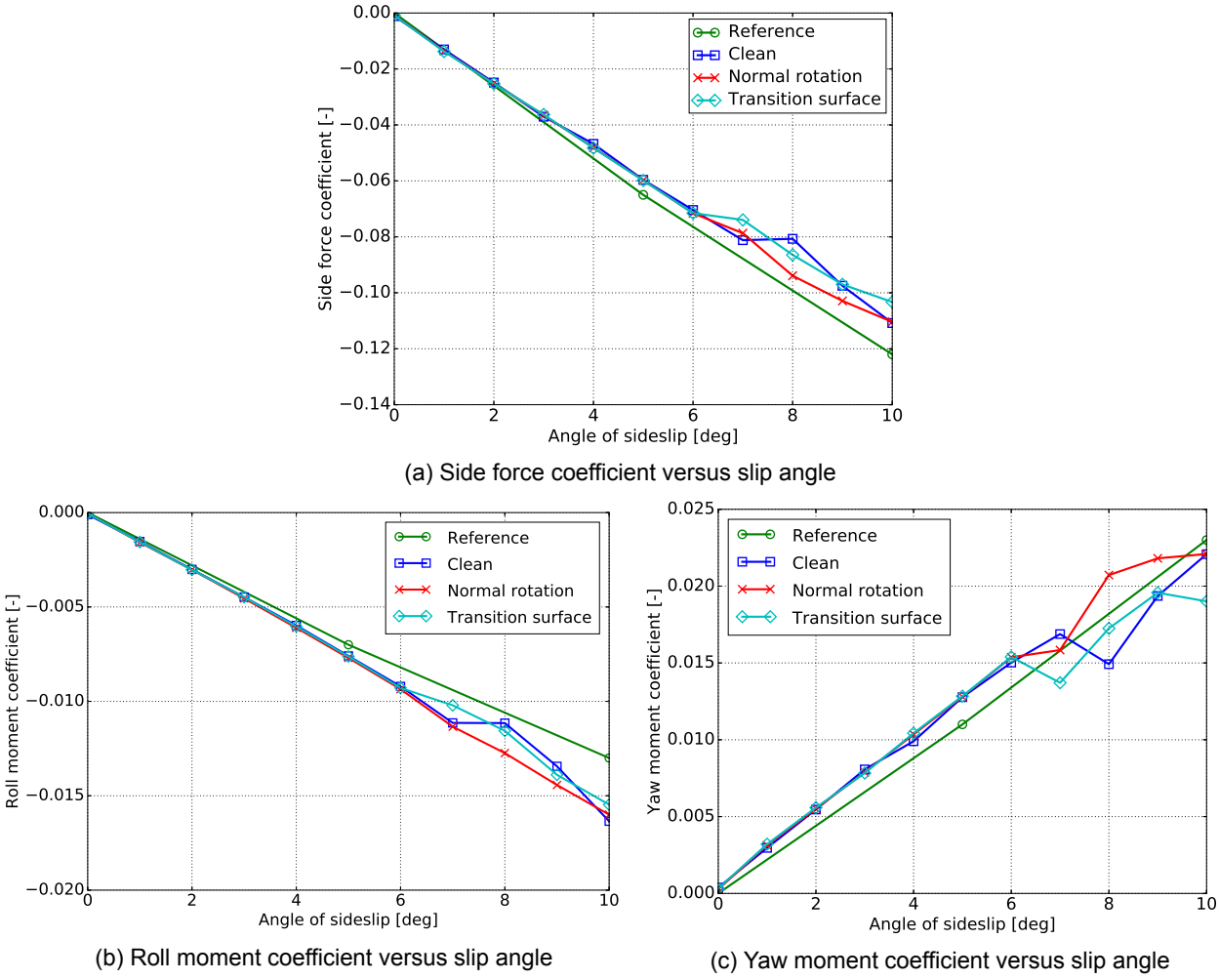
This section validates the lateral-directional characteristics of the Fokker100 models and tries to explain any unusual behaviour in the results. The validation process is performed for an aircraft model with a clean wing, a model with moveables modelled according to the 'normal rotation' option and a model with transition surfaces between the wing and the moveables. Figure 6.21 shows the location and the size of the moveable while table 6.6 shows the settings for the mesh of the models. The results for the lateral characteristics are obtained under the conditions specified by table 6.1. The reference point for the roll and yaw moment coefficients are listed in table 6.2.

6.5.1. Zero Deflection

The effects of the different models without any deflection for the control surfaces are analysed in this sub-section. The parameters that are important for analysing the lateral characteristics of the models are the side force, roll moment and the yaw moment coefficients. These parameters are plotted in figure 6.29 against the slip angle, while the angle of attack of the models is zero degrees. The first order derivatives of the trend lines for the curves of figure 6.29 are listed in table 6.9.

Figure 6.29 and table 6.9 show that there are no distinct differences between the various curves of the three Fokker100 models. Figure 6.29a shows that the side force coefficient of the models are underestimated, while figures 6.29b and 6.29c show that the roll moment and the yaw moment coefficient

of the models are overestimated.



(a) Side force coefficient versus slip angle

(b) Roll moment coefficient versus slip angle

(c) Yaw moment coefficient versus slip angle

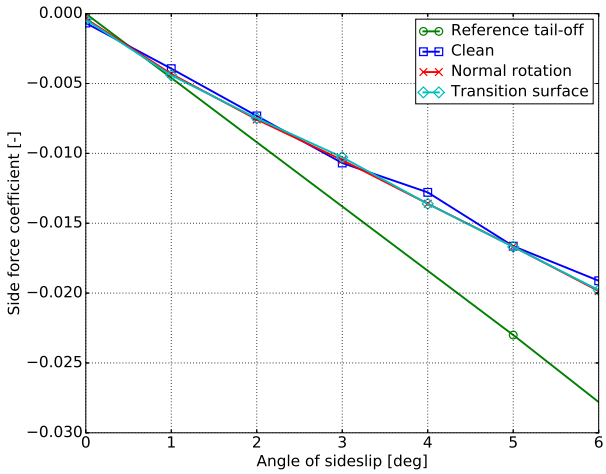
Figure 6.29: Lateral characteristics of Fokker100 model with zero deflections

Table 6.9: C_{Y_β}, C_{l_β} and C_{n_β} for curves of figure 6.29, derivatives in [1/rad]

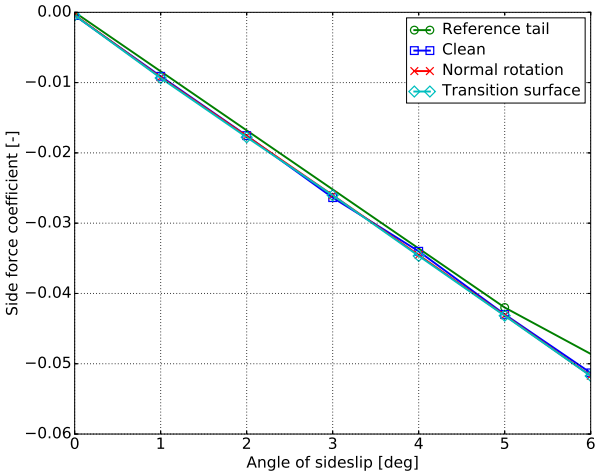
	Clean	Normal rotation	Transition surface	Reference
C_{Y_β}	-0.659	-0.670	-0.670	-0.699
C_{l_β}	-0.0859	-0.0859	-0.0859	-0.0745
C_{n_β}	0.138	0.143	0.143	0.132

Figure 6.30 separates the side force, roll moment and the yaw moment curves of figure 6.29 into curves for a tail-off configuration and into curves for the tail of the Fokker100. A tail-off configuration consist of the data of the fuselage of the main wing, while the tail configuration consists of the vertical tail and the horizontal tail planes. Doing this will give us more insight in which components are responsible for the deviation of the simulation data from the reference data. According to figures 6.30a and 6.30b, the wing-fuselage combination is the source of the deviation in the side force coefficient, but note that this is not entirely accurate since the dorsal fin of the Fokker100 is not modelled. The tail of the Fokker100 is the source of the deviation of the roll moment according to figures 6.30c and 6.30d.

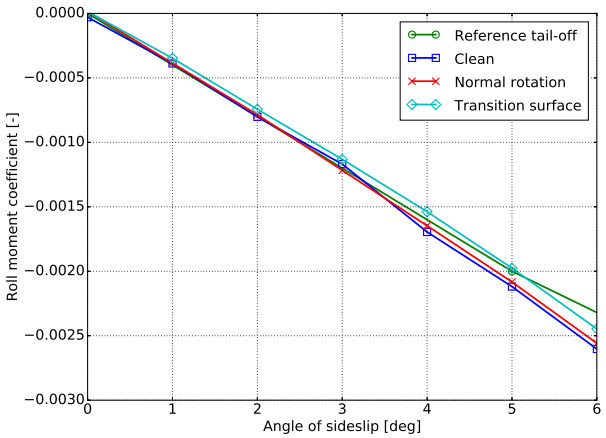
Examining figure 6.30e shows us that the trend of the yaw moment curve is almost the same as the trend of the reference data. Also, the deviation between the yaw moment curves and the reference data are very small. This means that the side force of the fuselage is quite well predicted by VSAERO since the main contributor to the yaw moment for conventional aircraft is most of the time the fuselage.



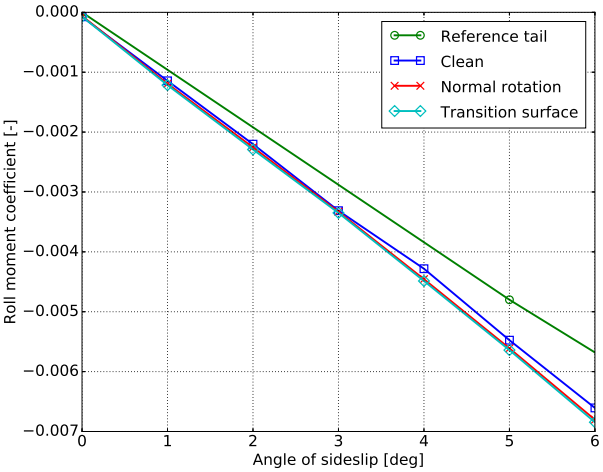
(a) Tail-off side force coefficient versus slip angle



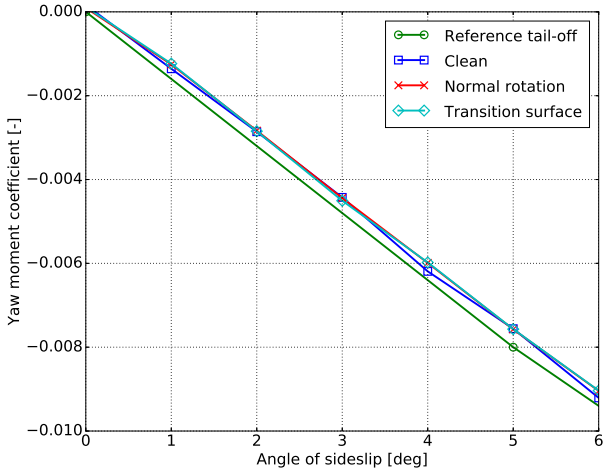
(b) Tail side force coefficient versus slip angle



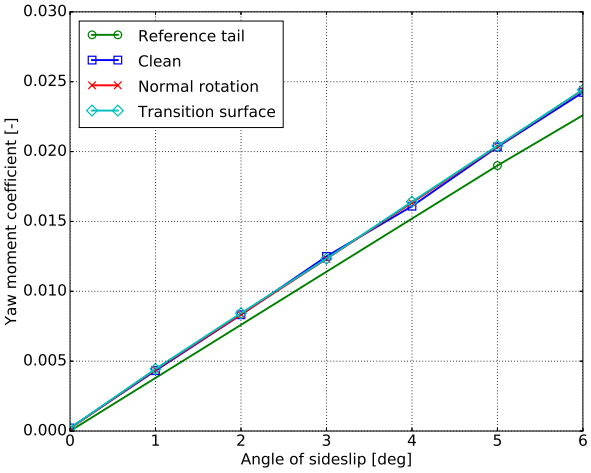
(c) Tail-off roll moment coefficient versus slip angle



(d) Tail roll moment coefficient versus slip angle



(e) Tail-off yaw moment coefficient versus slip angle



(f) Tail yaw moment coefficient versus slip angle

Figure 6.30: Lateral characteristics of Fokker100 model with tail-off and tail configuration

Another interesting fact that can be derived from figures 6.29a and 6.29c is that the side force of the whole aircraft is not exactly zero at zero sideslip angle. The sideforce coefficient seems to deviate with 0.001 with approximately 0.00045 for tail-off configuration and 0.00055 for the tail configuration. The source of this deviation is unknown, it may be noise.

Figure 6.29 also shows that there are instabilities in the curves for a sideslip angle of 7 or higher. These instabilities are caused by an improper stitching of the root wake line of the wing to the fuselage. The root wake line is currently stitched as the blue line of figure 6.31 to the fuselage. Using this stitched wake line for simulation will result in a wake that partially intersects the aft part of the fuselage, see figure 6.32. This wake-fuselage intersection is the source of the instabilities that has been observed in figure 6.29. Since stitching the root wake line as the red line of figure 6.31 reduces these instabilities, see figure 6.33.

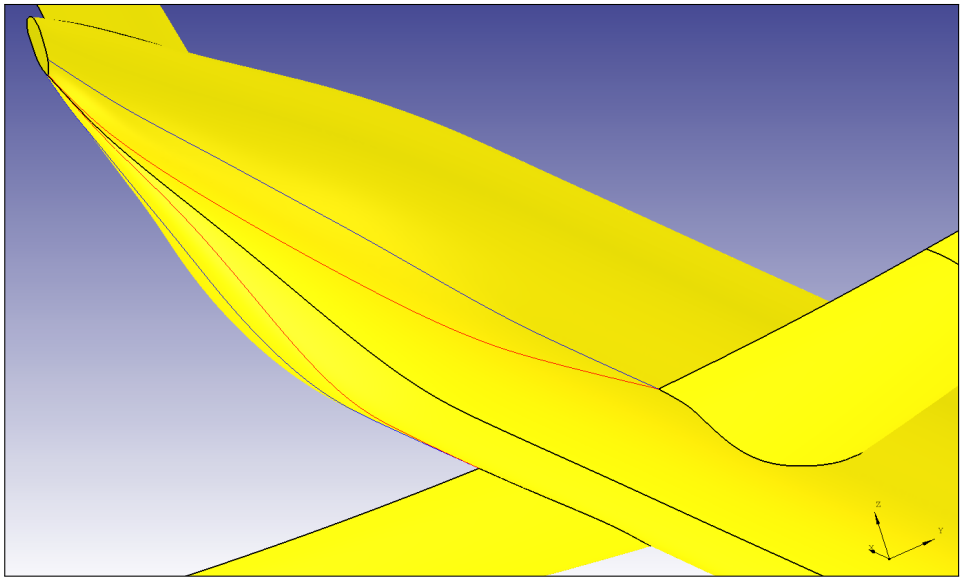


Figure 6.31: Blue line is standard stitched wakeline, red line is adapted stitched wakeline

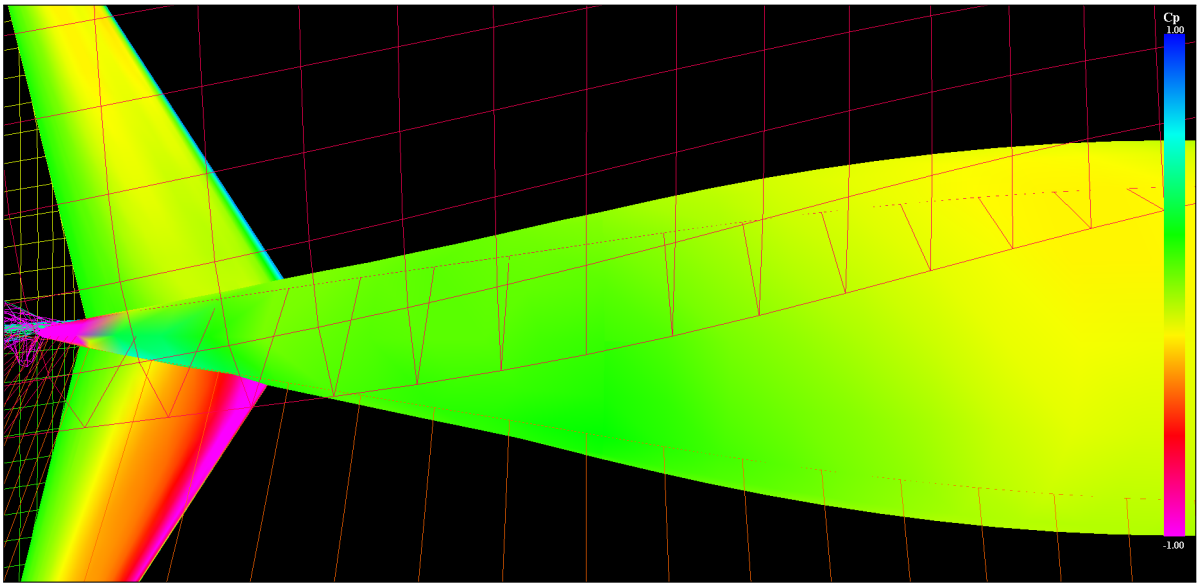


Figure 6.32: Wake of the wing intersects the aft part of the fuselage at sideslip of 10 degrees, bottom view

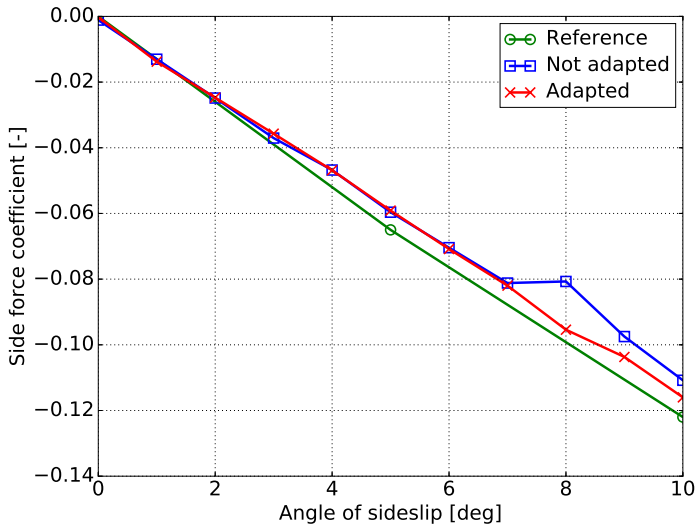


Figure 6.33: Side force coefficients versus sideslip angle of adapted stitched wakeline

6.5.2. Moveable Deflection

The ailerons and the rudder are the control surfaces of an aircraft to control the yawing and rolling moment of the aircraft. This section will thus analyse the effects of the ailerons and rudder deflections on the lateral-directional characteristics of the aircraft.

The lateral performance of the Fokker100 models with maximum aileron deflection are shown in figures 6.34, 6.35a and 6.35b. These figures show the side force, roll moment and yaw moment coefficients against the side slip angles with a deflection of 20 degrees for the right aileron and a deflection of -20 degrees for the left aileron. The lateral performance of the Fokker100 models with maximum rudder deflection are shown in figures 6.36a, 6.36b and 6.36c. These figures show the side force, roll moment and yaw moment coefficients against the side slip angles with a deflection of 33 degrees for the rudder to the port side of the aircraft.

Aileron

Prediction of the lateral-directional characteristics of the aircraft with the deflection of the ailerons seems to be less accurate than the predictions without the aileron deflection. The highest deviation from the reference data can be observed in the roll moment coefficient with an average deviation of 45 percent

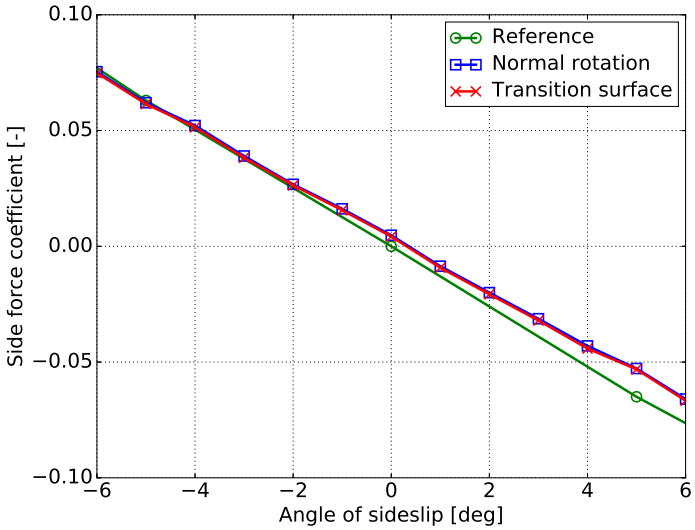
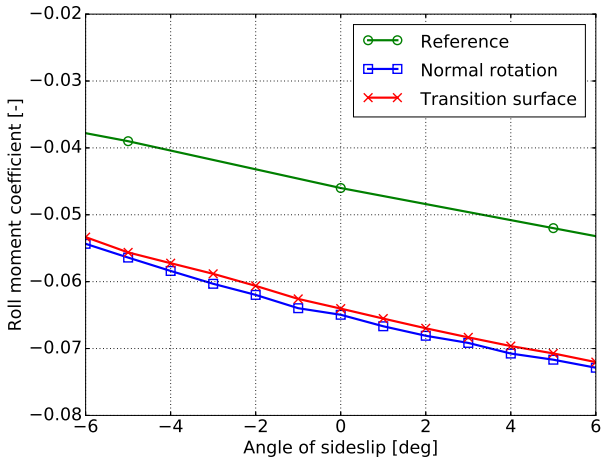
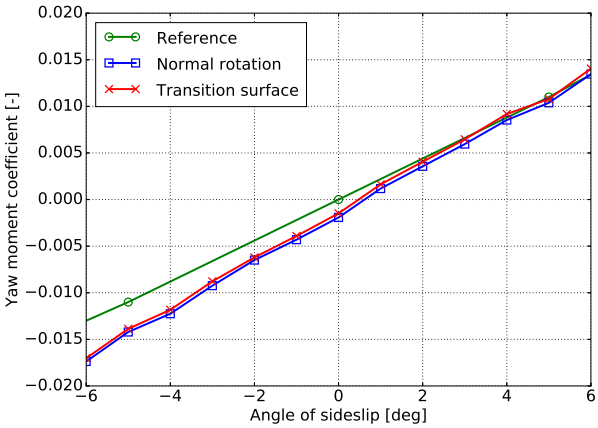


Figure 6.34: Side force coefficient versus slip angle, right aileron = 20 deg and left aileron = -20 deg

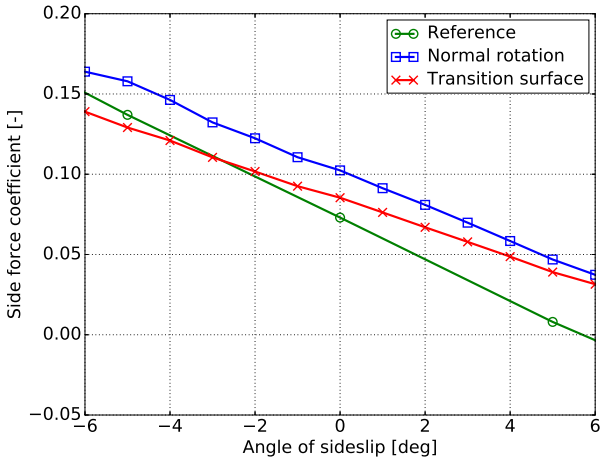


(a) Roll moment coefficient versus slip angle, right aileron = 20 deg and left aileron = -20 deg

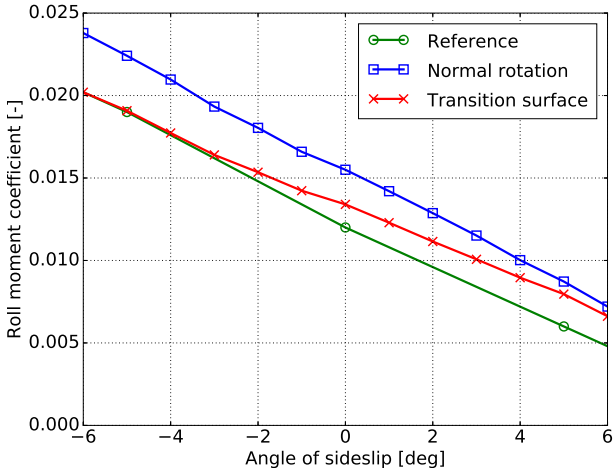


(b) Yaw moment coefficient versus slip angle, right aileron = 20 deg and left aileron = -20 deg

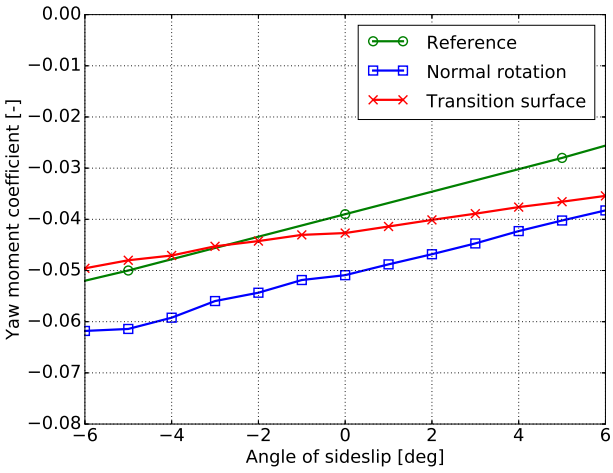
Figure 6.35: Lateral characteristics of Fokker100 model with aileron deflection



(a) Side force coefficient versus slip angle, rudder = 33 deg



(b) Roll moment coefficient versus slip angle, rudder = 33 deg



(c) Yaw moment coefficient versus slip angle, rudder = 33 deg

Figure 6.36: Lateral characteristics of Fokker100 model with rudder deflection

from the reference values. This large deviation is mainly caused by the over prediction of the lift of the wing with an aileron deflection of 20 degrees, see figure 6.26a.

The performance of the 'transition surface' moveable model performs overall better than the 'normal rotation' moveable model. The 'transition surface' model is slightly better that the 'normal rotation' model in the roll and yaw moment of the models. The difference in the roll moment coefficient is caused by the slight difference in the lift prediction of the ailerons, see figure 6.26a. This figure shows that the lift curve of the 'transition surface' is slightly lower. This lower lift causes the slight lower roll moment curve of the 'transition surface' model.

Rudder

Deflecting the rudder of the models produces a lot of differences between the two moveable modelling options. Both modelling options have its advantages and disadvantages when predicting the lateral-directional characteristics of the Fokker100.

The advantage of 'normal rotation' is that it predicts the trend of the reference curve quite well, while this is obviously not the case for the 'transition surface' model. A good prediction of the trend line will allow the user compute a more accurate control derivatives for the 'normal rotation' model.

The advantage of the 'transition surface' model is that its deviation from the reference curve is smaller than the deviation of the 'normal rotation' model. A smaller deviation means that it is more accurate in predicting the lateral-directional performance of the aircraft.

The difference in the performance of the two different rudder models are shown in the pressure distribution of the vertical tail, see figures 6.37 and 6.38. The pressure distributions show that the pressure peak of the 'transition surface' moveable starts more to the aft of the airfoil than the 'normal rotation' moveable. Also, the pressure peak of the 'normal rotation' moveable is starting one panel row ahead of the moveable of the wing, which is very odd. This forward placement of the suction peak is increasing the pressure distribution in front of the moveable and hence resulting in a higher side force coefficient for the 'normal rotation' model. This higher side force coefficient, in turn, increases the roll and yaw moment coefficients of the 'normal rotation' model.

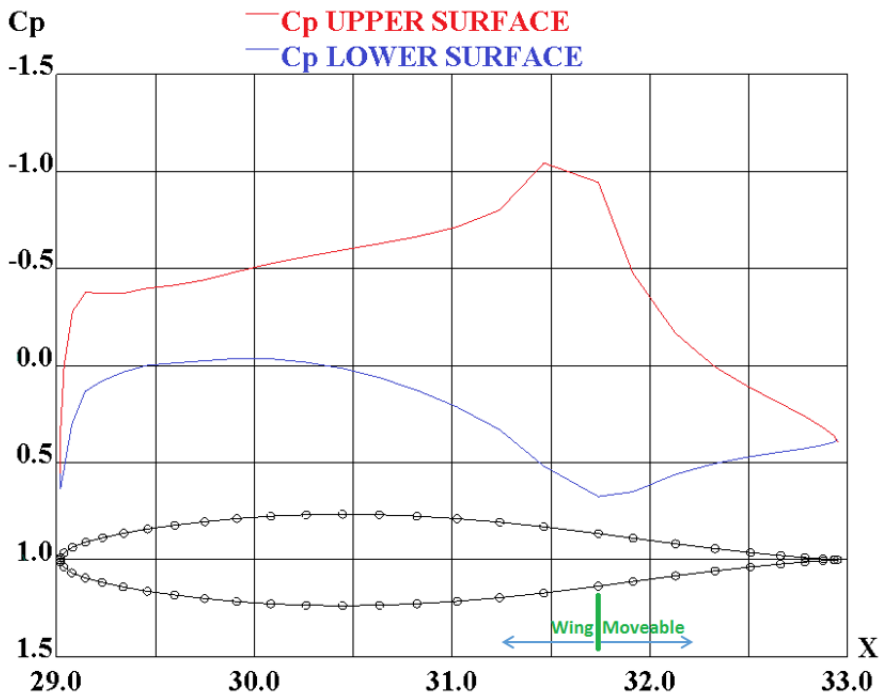


Figure 6.37: Pressure distribution of the vertical tail at 3m from the fuselage center line with deflection 33 degrees for the rudder, 'normal rotation' option

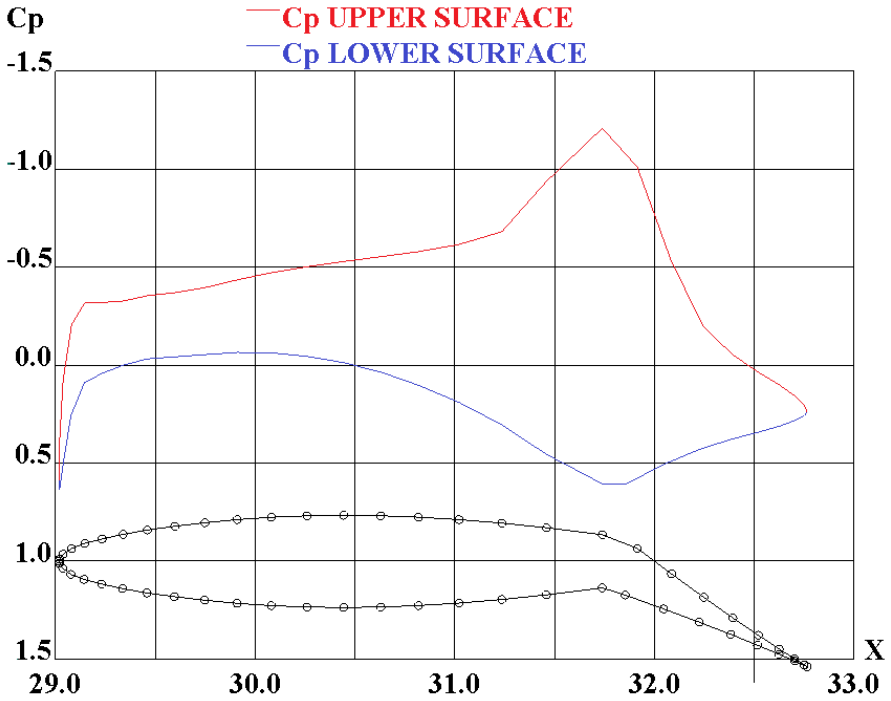


Figure 6.38: Pressure distribution of the vertical tail at 3m from the fuselage center line with deflection 33 degrees for the rudder, 'transition surface' option

6.5.3. Conclusion

The aerodynamic simulations in VSAERO are unstable when the Fokker100 models are simulated under high slip angles. These instabilities are caused by the improper stitching of the root wake lines of the main wing, where the wake of the main wing is intersecting the aft part of the fuselage during the wake relaxation process in VSAERO.

The lateral characteristics of both the 'normal rotation' model and the 'transition surface' are almost identical to each other when the deflection angles of the control surfaces are zero. The side force curve of the models was underestimated, which is caused by the underestimation of the side force of the main wing. The yaw moment and the roll moment curves of the models were overestimated.

Deflecting the aileron primarily affects the roll moment coefficients of both models since the primary function of the ailerons is to change the lift distribution over the wing to control the roll rate of the aircraft. The roll moment curve is also the curve with the largest deviation from the reference curve and this deviation is the smallest for the 'transition surface' model.

Deflecting a rudder has a huge impact on the prediction capabilities of the two models. The 'normal rotation' moveable model predicts the trend line of the reference curve quite well, while the error between the model's data and the reference data are bigger than the error for the 'transition surface' moveable model. The difference between the two models is caused by the different location of the suction peak of the moveables. The suction peak of the moveable for the 'normal rotation' moveable is located more to the front and this increases the side force of the vertical tail.

6.6. Conclusion

VSAERO settings

The runtime and accuracy of the aerodynamic solver VSAERO can be tuned by changing the type of solver or the number of wake and viscous iterations.

Two solvers were tested and they were the direct solver LAPACK and the indirect solver LAPACK BGS. The results show that the indirect solver is the better solver since its runtime is half of the runtime of the direct solver, while the difference in the results was negligible.

Wake iterations are more resource intensive than viscous iterations and this is reflected in the runtime. The runtime of one wake iteration is approximately equal to the runtime of 7 viscous iterations.

Wake iterations converge very quickly and 3 to 5 iterations are usually needed for convergence. The convergence of the viscous iterations, on the other hand, is very dependent on the testing conditions. For low angle attack, the Fokker100 model requires 15 viscous iterations, while for high angle of attack 5 iterations were needed for convergence.

Model settings

The accuracy of the results generated by VSAERO is very dependent on how the models are constructed. This thesis has investigated the impact of the panel density, the shape of the wake lines and congruent versus incongruent mesh models on the accuracy of the results.

The panel density can be varied in the chordwise and in the spanwise direction. Varying the chordwise density mainly influences the suction peak of the wing. More chordwise panels mean a more accurate prediction of the suction peak. The spanwise density, on the other hand, affects the root and tip performance of the wing. The impact of the spanwise panel density on the root lift distribution is only relevant at low angle of attack or at side slip angles, while the tip performance is only relevant at high angle of attack. A higher spanwise density at the root and at the tip produces a smoother spanwise lift distribution with higher lift at the root and lower lift at the tip

The impact of the start shape of the wake line is very dependent on the aircraft model and its testing conditions. The shape of the wake lines is irrelevant if none of the aircraft wakes is intersecting a component of the aircraft model. If a wake is intersecting another component than the wake line shape is very important. The wake line has to be modelled such that the wake does not intersect any other aircraft components during the wake relaxation.

Two meshing methods were investigated and they were the incongruent and congruent meshing method. The incongruent meshing method meshed all the aircraft components separately. This will result in a panel model, where the mesh grids of two neighbouring components are not aligned with each other at the intersection between the components of the aircraft. For the congruent method the aircraft is modelled as one panel model and the mesh grids of two neighbouring components are aligned at the intersection between the components. The results show that the congruent meshing method is the better method for discretizing an aircraft model to a panel model since the incongruent meshing method is incapable in predicting the interaction of the two neighbouring components accurately.

Longitudinal and Lateral-Directional Characteristics

The wing with moveable model with a gap between the sides of the moveable and the wing is unstable. The instability is caused by the presence of the gap. The presence of the gap creates a low pressure area, which accelerates the surrounding airflow. The surrounding airflow is accelerated to a ridiculous amount due to the fact VSAERO uses first order potential method for modelling the airflow. A first order panel method does not take into account the viscous effects. VSAERO does employ a boundary layer integral method to compute the viscous effects but this method does not prevent the creation of the very low pressure area around the gap between the wing and the moveable. This flaw makes the moveable modelling method with a gap between the wing and moveable useless.

The longitudinal and the lateral-directional characteristics of both the 'normal rotation' and the 'transition surface' moveable models are almost identical to each other. But this is only true if the deflection angles of all the moveables/control surfaces are zero. Also, both models underestimate the drag of the aircraft, which is caused by the lack of the engines, engine pylons and dorsal fin for the aircraft models. The trend of the drag curve, on the other hand, is predicted quite good.

Some differences between the 'normal rotation' and the 'transition surface' model can be observed when the moveables are deflected. Deflecting a moveable creates a suction peak at the aft part of the wing. The location of this suction peak is different for the two moveable models. The suction peak of the 'normal rotation' model is located on the wing in front of the start of the moveable. The suction peak of the 'transition surface', on the other hand, is located at the start or slightly behind the start of the moveable. The effect of the different locations of the suction peak is that a suction peak on the wing will generate less induced drag than a suction peak located on the deflected moveable. The lift generated by the suction peak of 'normal rotation' moveable is pointed less to the aft when compared to the suction peak of 'transition surface' model. The results is a lower induced drag and a slightly higher lift for the 'normal rotation' model.

The wrong placement of the suction peak of the moveables makes the prediction capability of the 'normal rotation' model inferior to the prediction capability of the 'transition surface' model when the

moveables are deflected. The 'normal rotation' model is also the model that has the highest deviation in the lift, drag and side force of the aircraft when compared to the reference data. So it is recommended to use the 'transition surface' modelling option for the moveable.

7

Validation of Stability & Control Derivatives

One of the research goals of this thesis is to compute the stability & control derivatives using the parametric aircraft modelling tool MMG and the aerodynamic solver VSAERO. These derivatives can be used to analyse the handling qualities and the stability of the aircraft model.

Section 7.1 will show the method that has been used to derive the stability & control derivatives for the Fokker100 model. The simulated derivatives are then validated in section 7.2 by comparing it to the reference data, while section 7.3 investigates the impact of these derivatives on the stability modes.

7.1. Derivation Method

Stability & control derivatives are parameters that measure the rate of how fast the forces and moments of an aircraft changes by varying the flight conditions such as flight speed, angle of attack, elevator deflection etc. These derivatives are used to determine the stability of the aircraft during a disturbed motion around an equilibrium condition. The equilibrium condition assumes that the aircraft is flying steadily with no manoeuvres that drastically changes the flight conditions.

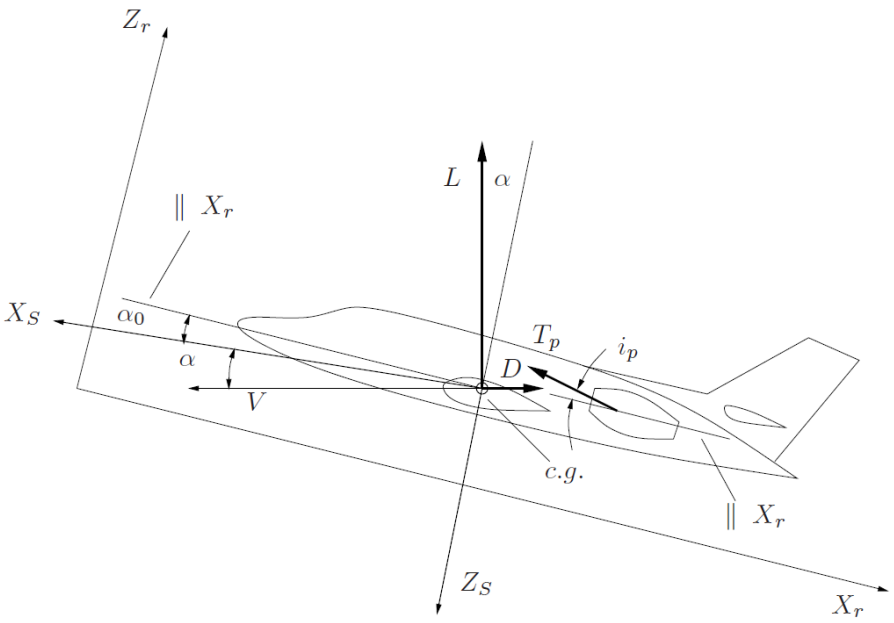


Figure 7.1: The attitude of the stability reference frame relative to the Xr- and Zr-axis, after a disturbance from the equilibrium flight condition [21]

Figure 7.1 shows the attitude of the stability reference frame with all the relevant parameters of the aircraft for deriving the stability & control derivatives. α_0 is the angle of attack of the initial equilibrium condition with no other disturbances. The axis denoted with a S is the stability reference frame with the X-axis pointing in the flight direction of the aircraft at the initial equilibrium condition.

The stability & control derivatives that is derived using the stability reference frame are given in figures 7.2 and 7.3. These figures show the derivatives that are relevant for performing an analysis on the stability and the handling qualities of an aircraft.

$$C_{X_u} = \frac{1}{\frac{1}{2}\rho V S} \cdot \frac{\partial X}{\partial u} \neq \frac{\partial C_X}{\partial \hat{u}}$$
$$C_{Z_u} = \frac{1}{\frac{1}{2}\rho V S} \cdot \frac{\partial Z}{\partial u} \neq \frac{\partial C_Z}{\partial \hat{u}}$$
$$C_{m_u} = \frac{1}{\frac{1}{2}\rho V S\bar{c}} \cdot \frac{\partial M}{\partial u} \neq \frac{\partial C_m}{\partial \hat{u}}$$
$$C_{X_\alpha} = \frac{1}{\frac{1}{2}\rho V S} \cdot \frac{\partial X}{\partial w} = \frac{\partial C_X}{\partial \alpha}$$
$$C_{Z_\alpha} = \frac{1}{\frac{1}{2}\rho V S} \cdot \frac{\partial Z}{\partial w} = \frac{\partial C_Z}{\partial \alpha}$$
$$C_{m_\alpha} = \frac{1}{\frac{1}{2}\rho V S\bar{c}} \cdot \frac{\partial M}{\partial w} = \frac{\partial C_m}{\partial \alpha}$$
$$C_{Z_{\dot{\alpha}}} = \frac{1}{\frac{1}{2}\rho S\bar{c}} \cdot \frac{\partial Z}{\partial \dot{w}} = \frac{\partial C_Z}{\partial \frac{\dot{\alpha}\bar{c}}{V}}$$
$$C_{m_{\dot{\alpha}}} = \frac{1}{\frac{1}{2}\rho S\bar{c}^2} \cdot \frac{\partial M}{\partial \dot{w}} = \frac{\partial C_m}{\partial \frac{\dot{\alpha}\bar{c}}{V}}$$
$$C_{X_q} = \frac{1}{\frac{1}{2}\rho V S\bar{c}} \cdot \frac{\partial X}{\partial q} = \frac{\partial C_X}{\partial \frac{q\bar{c}}{V}}$$
$$C_{Z_q} = \frac{1}{\frac{1}{2}\rho V S\bar{c}} \cdot \frac{\partial Z}{\partial q} = \frac{\partial C_Z}{\partial \frac{q\bar{c}}{V}}$$
$$C_{m_q} = \frac{1}{\frac{1}{2}\rho V S\bar{c}^2} \cdot \frac{\partial M}{\partial q} = \frac{\partial C_m}{\partial \frac{q\bar{c}}{V}}$$

$$C_{Y_\beta} = \frac{1}{\frac{1}{2}\rho V S} \cdot \frac{\partial Y}{\partial v} = \frac{\partial C_Y}{\partial \beta}$$
$$C_{\ell_\beta} = \frac{1}{\frac{1}{2}\rho V Sb} \cdot \frac{\partial L}{\partial v} = \frac{\partial C_\ell}{\partial \beta}$$
$$C_{n_\beta} = \frac{1}{\frac{1}{2}\rho V Sb} \cdot \frac{\partial N}{\partial v} = \frac{\partial C_n}{\partial \beta}$$
$$C_{Y_{\dot{\beta}}} = \frac{1}{\frac{1}{2}\rho V Sb} \cdot \frac{\partial Y}{\partial \dot{v}} = \frac{\partial C_Y}{\partial \frac{\dot{\beta}b}{V}}$$
$$C_{n_{\dot{\beta}}} = \frac{1}{\frac{1}{2}\rho Sb^2} \cdot \frac{\partial N}{\partial \dot{v}} = \frac{\partial C_n}{\partial \frac{\dot{\beta}b}{V}}$$
$$C_{Y_p} = \frac{2}{\frac{1}{2}\rho V Sb} \cdot \frac{\partial Y}{\partial p} = \frac{\partial C_Y}{\partial \frac{pb}{2V}}$$
$$C_{\ell_p} = \frac{2}{\frac{1}{2}\rho V Sb^2} \cdot \frac{\partial L}{\partial p} = \frac{\partial C_\ell}{\partial \frac{pb}{2V}}$$
$$C_{n_p} = \frac{2}{\frac{1}{2}\rho V Sb^2} \cdot \frac{\partial N}{\partial p} = \frac{\partial C_n}{\partial \frac{pb}{2V}}$$
$$C_{Y_r} = \frac{2}{\frac{1}{2}\rho V Sb} \cdot \frac{\partial Y}{\partial r} = \frac{\partial C_Y}{\partial \frac{rb}{2V}}$$
$$C_{\ell_r} = \frac{2}{\frac{1}{2}\rho V Sb^2} \cdot \frac{\partial L}{\partial r} = \frac{\partial C_\ell}{\partial \frac{rb}{2V}}$$
$$C_{n_r} = \frac{2}{\frac{1}{2}\rho V Sb^2} \cdot \frac{\partial N}{\partial r} = \frac{\partial C_n}{\partial \frac{rb}{2V}}$$

Figure 7.2: Definitions of the stability derivatives. Left: symmetric motions - Right: asymmetric motions. [21]

$$C_{X_{\delta_e}} = \frac{1}{\frac{1}{2}\rho V^2 S} \cdot \frac{\partial X}{\partial \delta_e} = \frac{\partial C_X}{\partial \delta_e}$$
$$C_{Z_{\delta_e}} = \frac{1}{\frac{1}{2}\rho V^2 S} \cdot \frac{\partial Z}{\partial \delta_e} = \frac{\partial C_Z}{\partial \delta_e}$$
$$C_{m_{\delta_e}} = \frac{1}{\frac{1}{2}\rho V^2 S\bar{c}} \cdot \frac{\partial M}{\partial \delta_e} = \frac{\partial C_m}{\partial \delta_e}$$
$$C_{X_{\delta_t}} = \frac{1}{\frac{1}{2}\rho V^2 S} \cdot \frac{\partial X}{\partial \delta_t} = \frac{\partial C_X}{\partial \delta_t}$$
$$C_{Z_{\delta_t}} = \frac{1}{\frac{1}{2}\rho V^2 S} \cdot \frac{\partial Z}{\partial \delta_t} = \frac{\partial C_Z}{\partial \delta_t}$$
$$C_{m_{\delta_t}} = \frac{1}{\frac{1}{2}\rho V^2 S\bar{c}} \cdot \frac{\partial M}{\partial \delta_t} = \frac{\partial C_m}{\partial \delta_t}$$

$$C_{Y_{\delta_a}} = \frac{1}{\frac{1}{2}\rho V^2 S} \cdot \frac{\partial Y}{\partial \delta_a} = \frac{\partial C_Y}{\partial \delta_a}$$
$$C_{\ell_{\delta_a}} = \frac{1}{\frac{1}{2}\rho V^2 Sb} \cdot \frac{\partial L}{\partial \delta_a} = \frac{\partial C_\ell}{\partial \delta_a}$$
$$C_{n_{\delta_a}} = \frac{1}{\frac{1}{2}\rho V^2 Sb} \cdot \frac{\partial N}{\partial \delta_a} = \frac{\partial C_n}{\partial \delta_a}$$
$$C_{Y_{\delta_r}} = \frac{1}{\frac{1}{2}\rho V^2 S} \cdot \frac{\partial Y}{\partial \delta_r} = \frac{\partial C_Y}{\partial \delta_r}$$
$$C_{\ell_{\delta_r}} = \frac{1}{\frac{1}{2}\rho V^2 Sb} \cdot \frac{\partial L}{\partial \delta_r} = \frac{\partial C_\ell}{\partial \delta_r}$$
$$C_{n_{\delta_r}} = \frac{1}{\frac{1}{2}\rho V^2 Sb} \cdot \frac{\partial N}{\partial \delta_r} = \frac{\partial C_n}{\partial \delta_r}$$

Figure 7.3: Definitions of the control derivatives. Left: symmetric motions - Right: asymmetric motions. [21]

Not all the stability & control derivatives shown in figure 7.2 and 7.3 can be derived using the aerodynamic solver VSAERO. These derivatives are the derivatives with respect to $\dot{\alpha}$, derivatives with respect to $\dot{\beta}$ and derivatives with respect δ_t. The remaining derivatives can be derived and these are the derivatives with respect to airspeed u, angle of attack α, slip angle β, pitch rate q, roll rate p, yaw rate r, elevator angle δ_e, aileron angle δ_a and rudder angle δ_r. The **derivatives with respect to an angle** are computed **in radians** and not in degrees.

10 simulations of the aircraft model are performed in VSAERO to derive the stability & control derivatives. One simulation is performed at the initial equilibrium condition of the aircraft, while the other nine simulations represent the different disturbed conditions of the model. These disturbed conditions is almost similar to the equilibrium condition, but with a slight variation in u, α, β, q, p, r, δ_e, δ_a or δ_r. Each simulation of the nine simulations is then used together with the equilibrium data to derive a set of derivatives.

As an example, one simulation has increased the angle of attack of the equilibrium condition to an angle which is denoted as α. The stability derivatives C_{X_u}, C_{Z_u} and C_{m_u} can then be derived by determining the slope of the curve between the two corresponding data points. Expressed in a formula form, C_{X_u}, C_{Z_u} and C_{m_u} will become equation 7.1, 7.2 and 7.3 respectively. Parameters denoted with 0 are the parameters from the equilibrium condition, while the other parameters are the parameters from the current simulation.

$$C_{X_\alpha} = \frac{C_X - C_{X_0}}{\alpha - \alpha_0} \tag{7.1}$$

$$C_{Z_\alpha} = \frac{C_Z - C_{Z_0}}{\alpha - \alpha_0} \tag{7.2}$$

$$C_{m_\alpha} = \frac{C_m - C_{m_0}}{\alpha - \alpha_0} \tag{7.3}$$

All the other derivatives can be derived in a similar fashion as shown in the example above except for the derivatives with respect to the airspeed u. Figure 7.2 shows that C_{X_u}, C_{Z_u} and C_{m_u} are not equal $\frac{\partial C_X}{\partial u}$, $\frac{\partial C_Z}{\partial u}$ and $\frac{\partial C_m}{\partial u}$ respectively. The derivatives are determined using equations 7.4, 7.5 and 7.6, which are given by the reader of the flight dynamics course [21].

$$C_{X_u} = 2C_{X_0} + \left(\frac{\partial T_c}{\partial V} - \frac{\partial C_D}{\partial V}\right)V \tag{7.4}$$

$$C_{Z_u} = 2C_{Z_0} + \left(-\frac{\partial C_L}{\partial V} - \frac{\partial T_c}{\partial V}\left(\alpha_0 + i_p\right)\right)V \tag{7.5}$$

$$C_{m_u} = 2C_{m_0} + \frac{\partial C_m}{\partial V}V \tag{7.6}$$

7.2. Results

This section determines the accuracy of the stability & control derivatives of the Fokker100 models by comparing it with the reference data of the Fokker100 [20]. The stability & control derivatives of the different Fokker100 models are determined around the initial equilibrium condition given in table 7.1 and around the reference point given in table 7.2. The stability & control derivatives of the different Fokker100 models are given in table 7.3 and 7.4. Table 7.3 shows the longitudinal derivatives, while table 7.4 shows the lateral derivatives. The derivatives given in the tables are divided into static, dynamic and control derivatives. Most of the reference derivatives of table 7.3 and 7.4 are from the Fokker100 report [20] except for the derivatives with respect to the pitching velocity. The derivatives C_{Z_q} and C_{m_q}

Table 7.1: Fokker100 initial equilibrium condition

Variable	Value	Unit
Altitude	2000	[m]
Mach number	0.3	[-]
Reynolds number	$20.3 \cdot 10^6$	[-]
Angle of attack α_0	5	[deg]
Slip angle β_0	0	[deg]

Table 7.2: Reference parameters for the stability and control derivatives

Variable	Value	Unit
Chord	3.83	[m]
Area	93.5	[m^2]
Span	28.08	[m]
Center of gravity, X-coordinate	16.95	[m]
Center of gravity, Y-coordinate	0.00	[m]
Center of gravity, Z-coordinate	-0.25	[m]

of the reference data are not available and these derivatives are estimated using using equation 7.7 and 7.8 from the flight dynamics reader [21]. This means that C_{Z_q} and C_{m_q} are just rough estimates and does not represents the actual value of the Fokker100.

Table 7.3: Longitudinal stability & control derivatives with α = 5 degree and β = 0 degree

	Derivatives	Reference	Clean		Normal rotation		Transition surface	
			Value	Difference**	Value	Difference**	Value	Difference**
Static	C_{X_u}	-0,006	-0,004	24,8	-0,010	71,8	-0,017	195,3
	C_{Z_u}	-1,582	-1,616	2,2	-1,641	3,7	-1,657	4,8
	C_{m_u}	-0,138	-0,133	3,5	-0,157	13,9	-0,166	20,2
	C_{X_α}	0,445	0,477	7,1	0,457	2,6	0,448	0,7
	C_{Z_α}	-5,998	-5,972	0,4	-6,075	1,3	-6,172	2,9
	C_{m_α}	-1,617	-1,735	7,3	-1,508	6,8	-1,463	9,5
Dynamic	C_{X_q}	0,000	0,267		0,313		0,267	
	C_{Z_q}	-4,804*	-6,900	43,6	-6,833	42,2	-7,010	45,9
	C_{m_q}	-11,947*	-15,170	27,0	-15,700	31,4	-15,297	28,0
Control	$C_{X_{\delta_e}}$	-0,029			-0,040	36,9	-0,043	49,0
	$C_{Z_{\delta_e}}$	-0,346			-0,543	57,0	-0,506	46,5
	$C_{m_{\delta_e}}$	-1,466			-2,272	55,0	-2,204	50,3

* Rough estimation based on equation 7.7 and 7.8.
** Difference in percentage and is computed as the absolute value of $\frac{Reference\ value - actual\ value}{Reference\ value} * 100$.

$$C_{Z_q} = -2C_{N_{h_\alpha}} \left(\frac{V_h}{V}\right)^2 \frac{S_h l_h}{S \cdot MAC} \tag{7.7}$$

$$C_{m_q} = -1.1C_{N_{h_\alpha}} \left(\frac{V_h}{V}\right)^2 \frac{S_h l_h^2}{S \cdot MAC^2} \tag{7.8}$$

Examining table 7.3 shows us that the static derivatives are predicted quite well, where the most derivatives have error percentages of below ten percent except for C_{X_u}. The error percentage of C_{X_u} is irrelevant since the reference value is so small that it is negligible. The dynamic derivatives seem to be overestimated by approximately thirty-six percent, but this error percentage does not say a lot since the reference values are rough estimates. The control derivatives, on the other hand, are the derivatives which have the highest error percentage when compared to the static and dynamic derivatives. The average error for the control derivatives is approximately fifty percent.

Table 7.4 shows us that lateral static derivatives are less accurate than the longitudinal static derivatives with an average error percentage of fourteen percent. The dynamic derivatives with respect to the roll rate are the most accurate when compared to the derivatives with respect to yaw rate. The average error for the roll rate derivatives is seventeen percent while the average error for the yaw rate is twenty-eight percent. The lateral control derivatives are predicted more accurately than the longitudinal control derivatives with an average error of 30 percent excluding $C_{n_{\delta_a}}$ since it is negligibly small.

Table 7.3 and 7.4 shows that neither the 'normal rotation' or the 'transition surface' modelling options is outperforming the other. The 'normal rotation' option seems to be more accurate in the longitudinal

stability derivatives and lateral control derivatives, but the 'transition surface' option is more accurate in the lateral stability derivatives and the longitudinal control derivatives.

The conclusion is that the derivatives of the different models are quite acceptable with the static derivatives are predicted as the most accurate with an average error of seven percent. The control derivatives, on the other hand, are the least accurate with an average error of 40 percent. Also, the overall performance of two moveable modelling options is that they are performing equally well. The 'normal rotation' option is more accurate in the lateral control derivatives while the 'transition surface' option is more accurate in the longitudinal control derivatives.

Table 7.4: Lateral stability & control derivatives with α = 5 degree and β = 0 degree

	Derivatives	Reference	Clean		Normal rotation		Transition surface	
			Value	Difference*	Value	Difference*	Value	Difference*
Static	C_{Y_β}	-0,707	-0,588	16,9	-0,594	15,9	-0,593	16,0
	C_{l_β}	-0,089	-0,107	19,9	-0,100	12,6	-0,098	9,7
	C_{n_β}	0,107	0,119	10,3	0,122	13,7	0,121	12,2
Dynamic	C_{Y_p}	0,345	0,280	18,7	0,198	42,5	0,281	18,6
	C_{l_p}	-0,467	-0,519	11,1	-0,451	3,5	-0,537	14,9
	C_{n_p}	-0,098	-0,089	9,4	-0,069	30,3	-0,087	11,8
	C_{Y_r}	0,835	0,415	50,3	0,461	44,8	0,471	43,6
	C_{l_r}	0,187	0,249	33,4	0,226	21,2	0,227	21,5
	C_{n_r}	-0,199	-0,172	13,2	-0,165	17,0	-0,218	9,7
Control	$C_{Y_{\delta_a}}$	0,000			0,024		0,023	
	$C_{l_{\delta_a}}$	-0,152			-0,196	29,3	-0,198	30,1
	$C_{n_{\delta_a}}$	0,001			0,004	542,2	0,004	680,9
	$C_{Y_{\delta_r}}$	0,127			0,168	32,3	0,181	42,7
	$C_{l_{\delta_r}}$	0,011			0,019	72,8	0,020	77,7
	$C_{n_{\delta_r}}$	-0,066			-0,085	29,0	-0,091	38,1

* Difference in percentage and is computed as the absolute value of $\frac{Reference\ value - actual\ value}{Reference\ value} * 100$.

7.3. Stability Motions

This section investigates the effects of the stability derivatives of the different models on the stability modes of the aircraft. The longitudinal stability of the aircraft is the main focus of section 7.3.1, while section 7.3.2 investigates the lateral stability of the aircraft.

The flight dynamics for the analysis of the longitudinal and the lateral motions are modelled as a second order differential equations. The aircraft system for deriving the differential equation is a simple system with a mass m, a spring k and a damper c. The differential equation for this aircraft system is written as

$$F(t) = m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx \tag{7.9}$$

The solution for deriving the oscillating motions can be found by assuming the following

$$x = Ae^{\lambda t} \tag{7.10}$$

The characteristic equation of the second order differential equation can be derived by substituting equation 7.10 into equation 7.9, which is given by equation 7.11. Equation 7.11 can be rewritten to the form given by equation 7.12, where ζ is the damping ratio and ω_n is the natural frequency of the system.

$$\lambda^2 + \frac{c}{m}\lambda + \frac{k}{m} = 0 \tag{7.11}$$

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0 \tag{7.12}$$

The eigenvalues of the characteristic equation can be defined as equation 7.13 or equation 7.14, where η is the damping response of the system and ω is damped natural frequency.

$$\lambda_{1,2} = -\zeta\omega_n \pm i\omega_n\sqrt{1-\zeta^2} \tag{7.13}$$

$$\lambda_{1,2} = \eta \pm i\omega \tag{7.14}$$

The damped natural frequency ω can be used to derived the period of the oscillation, which is governed by equation 7.15. The damping response η, on the other hand, is used to determine the time for halving or doubling the amplitude of the oscillating motion.

$$P = \frac{2\pi}{\omega} \tag{7.15}$$

$$t_{\frac{1}{2}} \; or \; t_2 = \frac{ln2}{|\eta|} \tag{7.16}$$

7.3.1. Longitudinal Motion

For the longitudinal stability, the phugoid and the short period modes are the oscillating motions that is investigated in this subsection.

The phugoid mode is very slow occurring oscillation, where the energy of the aircraft is slowly interchanged between the kinetic and the potential energy. This motion is characterised by changing pitch angle, airspeed and altitude. The natural frequency and the damping ratio of the phugoid mode can be approximated by the following equations [22].

$$\omega_{n_p} = \sqrt{\frac{-Z_u g}{u_0}} \tag{7.17}$$

$$\zeta_p = \frac{-X_u}{2\omega_{n_p}} \tag{7.18}$$

The short period mode is a fast oscillating motion in order of a few seconds. This oscillating motion primarily affects the pitch angle of the aircraft, where the pitch angle changes very rapidly around its centre of gravity. The natural frequency and the damping ratio of the short period mode are approximated using the equations below [22].

$$\omega_{n_{sp}} = \sqrt{\frac{Z_\alpha M_q}{u_0} - M_\alpha} \tag{7.19}$$

$$\zeta_{sp} = -\frac{M_q + M_{\dot{\alpha}} + \frac{Z_\alpha}{u_0}}{2\omega_{n_{sp}}} \tag{7.20}$$

The natural frequency ω_n, the damping ratio ζ, the period P and the time to halving the amplitude $t_{\frac{1}{2}}$ of the phugoid and short period modes are shown in tables 7.5 and 7.6 respectively. An eigenvalue

plot of the phugoid mode and the short period mode is shown in figure 7.4. The eigenvalues for the phugoid mode are located on the right side of the figure, while the short-period mode is located on the left side of the figure.

The natural frequency and the period of the simulation data for the phugoid mode are approximated quite well with a difference of approximately of 2 percent. The damping ratio and the time to halving the amplitude of the motion, on the other hand, is predicted very bad with a difference ranging from 30 percent to 200 percent. This huge difference is caused by the differences in the C_{X_u} of the reference data and the simulation data since the damping ratio is dependent on the force derivative X_u. Also note that the damping ratio of the phugoid mode is very small, which exaggerate the error percentages. The small differences between the damping ratios of the reference data and of the simulated data cannot be observed in the eigenvalue plot of figure 7.4.

The short period mode for all the models are predicted quite well since the differences in the period and the time to halving the period are below ten percent.

Overall the 'normal rotation' modelling option performance slightly better than the 'transition surface' option. Except for the damping ratio of the phugoid mode, the damping ratio of the 'normal rotation' option performance almost three times better than the 'transition surface' option.

Table 7.5: Comparison of the phugoid mode for the reference data and simulation data

Phugoid mode							
Parameter	Reference	Clean		Normal rotation		Transition surface	
		Value	Difference*	Value	Difference*	Value	Difference*
ω_n $[1/s]$	0,145	0,147	1,1	0,148	1,9	0,149	2,3
ζ $[-]$	0,0026	0,0018	29,2	0,0045	75,7	0,0077	197,2
$t_{\frac{1}{2}}$ $[s]$	1854,7	2591,1	39,7	1036,4	44,1	609,7	67,1
P $[s]$	43,3	42,9	1,0	42,5	1,8	42,5	1,9

* Difference in percentage and is computed as the absolute value of $\frac{Reference\,value - actual\,value}{Reference\,value} * 100$.

Table 7.6: Comparison of the short period mode for the reference data and simulation data

Short period mode							
Parameter	Reference	Clean		Normal rotation		Transition surface	
		Value	Difference*	Value	Difference*	Value	Difference*
ω_n $[1/s]$	1,93	2,09	8,4	1,89	2,0	1,85	4,3
ζ $[-]$	0,264	0,257	2,8	0,290	9,9	0,299	13,1
$t_{\frac{1}{2}}$ $[s]$	1,4	1,3	5,1	1,3	7,1	1,3	7,6
P $[s]$	3,4	3,1	7,9	3,5	2,8	3,6	5,6

* Difference in percentage and is computed as the absolute value of $\frac{Reference\,value - actual\,value}{Reference\,value} * 100$.

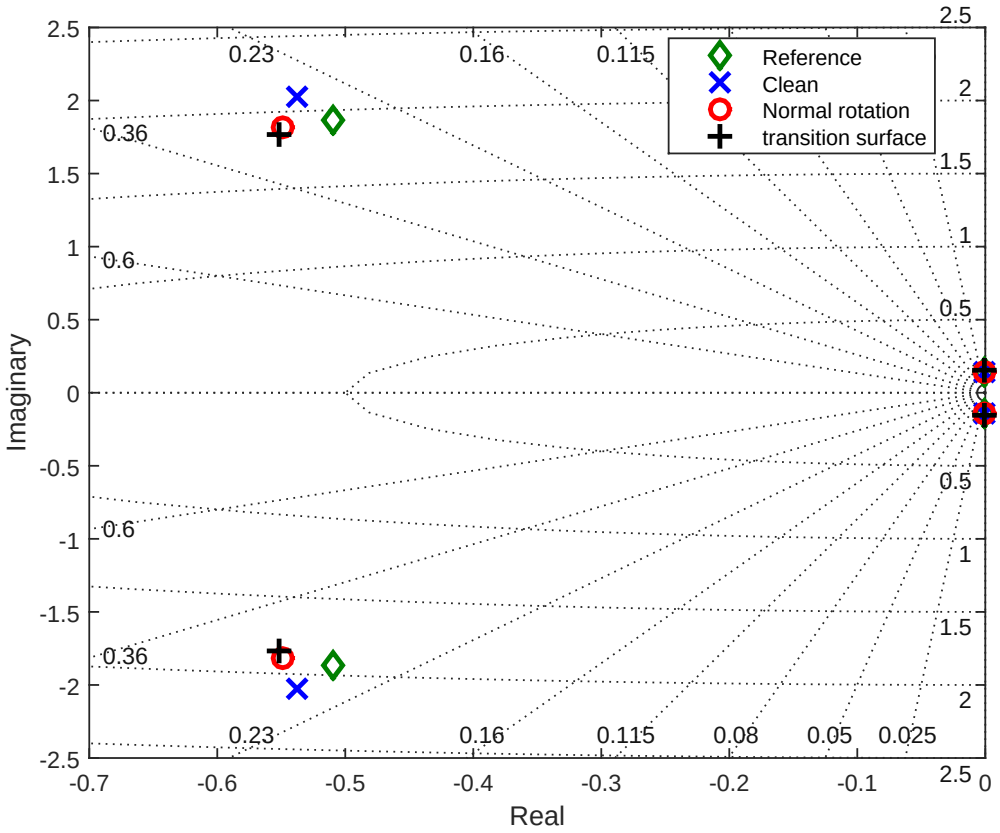


Figure 7.4: Eigenvalue plot of the phugoid mode and the short period mode

7.3.2. Lateral Motion

The spiral mode, the roll mode and the Dutch roll mode are the oscillation modes of the lateral motion of the aircraft that is inspected in this subsection.

The spiral mode is inherent to all aircraft and this mode is often slow and unstable. This motion is usually initiated by a small roll disturbance. This small rolling motion causes a small non-negligible yawing motion, which in turn can increase the rolling motion if the aircraft is unstable. The eigenvalues of the spiral mode is approximated using equation 7.21 [22].

$$\lambda_{spiral} = \frac{L_\beta N_r - L_r N_\beta}{L_\beta} \tag{7.21}$$

The roll mode is a fast motion related to the rolling motion of the aircraft itself. This motion describes if the aircraft continues to roll when the aircraft is experiencing a rolling moment or it stops and reverts to the equilibrium condition. The eigenvalue of this roll mode is approximated by the following equation [22].

$$\lambda_{roll} = L_p \tag{7.22}$$

The Dutch roll mode is an oscillatory motion that consists of rolling motion and damped yawing motion. The natural frequency and the damping ratio can be approximated using equations 7.23 and 7.24 respectively [22].

$$\omega_{n_{dr}} = \sqrt{\frac{Y_\beta N_r - N_\beta Y_r + u_0 N_\beta}{u_0}} \tag{7.23}$$

$$\zeta_{dr} = -\frac{1}{2\omega_{n_{dr}}} \frac{Y_\beta + u_0 N_r}{u_0} \tag{7.24}$$

Equations 7.21 and 7.22 shows that the solution for the eigenvalue does not consist of a complex value, which means that ω is equal to 0 and ζ must either be 1 or -1. The standard form of the eigenvalue can be simplified to equation 7.25. This equation tells us that ζ must be -1 if λ is bigger than 0 or ζ must be 1 if λ is lower than 0 since the natural frequency can never be negative.

$$\lambda = -\zeta\omega_n \tag{7.25}$$

Table 7.7 shows the results for the spiral mode and the eigenvalues are shown in the eigenvalue plot of figure 7.5. The eigenvalues of the spiral mode are located on the positive real axis of figure 7.5. The eigenvalue plot shows that the spiral mode is unstable for all the models and the reference data since the damping response η is positive. Table 7.7 shows that the accuracy of the simulations is very bad. The differences in the natural frequency range from 144 percent for the 'transition surface' option to 335 percent for the 'normal rotation' option. The bad accuracy is attributed to the very small value of the natural frequency ω_n. The eigenvalue plot shows that the spiral modes are predicted quite well since they are clustered closely together.

Table 7.7: Comparison of the spiral mode for the reference data and simulation data

Spiral mode							
Parameter	Reference	Clean		Normal rotation		Transition surface	
		Value	Difference*	Value	Difference*	Value	Difference*
$\omega_n\ [1/s]$	0,0221	0,0909	312,4	0,0960	335,1	0,0540	144,7
$t_2\ [s]$	31,4	7,6	75,7	7,2	77,0	12,8	59,1

* Difference in percentage and is computed as the absolute value of $\frac{Reference\ value - actual\ value}{Reference\ value} * 100$.

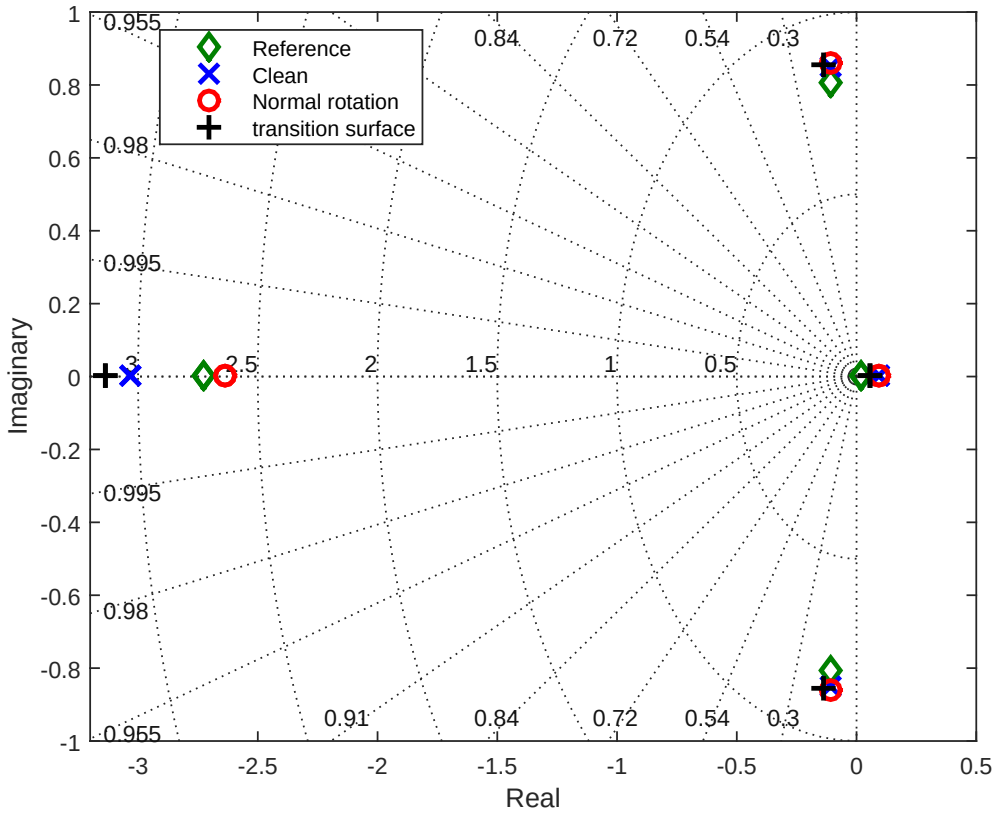


Figure 7.5: Eigenvalue plot of the spiral mode, roll mode and the dutch roll mode

The results of the roll mode is displayed in table 7.8 and the eigenvalues of the roll mode are located on the left side of figure 7.5. The eigenvalue plot of figure 7.5 shows that the roll mode is stable for all the models and reference aircraft. The difference in the natural frequency is very acceptable with the lowest value 3.4 percent for the 'normal rotation' option and the maximum value of 15 percent for the 'transition surface' model.

The characteristics of the Dutch roll mode for the different data sets are given in table 7.9. The eigenvalues of the Dutch roll mode are also plotted in the eigenvalue plot of figure 7.5. The eigenvalues are located on the top right side and the bottom right side of figure 7.5. The natural frequency of the different models is predicted quite well with an average difference of approximately 6 percent. The accuracy of the damping ratio is predicted quite well for the 'transition surface' option with a difference of 5.2 percent while the difference for the clean wing model and the 'normal rotation' model is around twenty percent.

The 'normal rotation' option is better in predicting roll mode of the models, while the 'transition surface' option is the better in predicting the spiral and Dutch roll modes.

Table 7.8: Comparison of the roll mode for the reference data and simulation data

Roll mode							
Parameter	Reference	Clean		Normal rotation		Transition surface	
		Value	Difference*	Value	Difference*	Value	Difference*
ω_n $[1/s]$	2,73	3,03	11,1	2,64	3,4	3,14	15,0
$t_{\frac{1}{2}}$ $[s]$	0,25	0,23	10,0	0,26	3,5	0,22	13,0

* Difference in percentage and is computed as the absolute value of $\frac{Reference\ value - actual\ value}{Reference\ value} * 100.$

Table 7.9: Comparison of the Dutch roll mode for the reference data and simulation data

Dutch roll mode							
Parameter	Reference	Clean		Normal rotation		Transition surface	
		Value	Difference*	Value	Difference*	Value	Difference*
ω_n $[1/s]$	0,814	0,856	5,2	0,866	6,4	0,864	6,2
ζ $[-]$	0,164	0,133	18,8	0,128	21,5	0,155	5,2
$t_{\frac{1}{2}}$ $[s]$	5,2	6,1	17,0	6,2	19,8	5,2	0,7
P $[s]$	7,8	7,4	5,4	7,3	6,5	7,4	6,0

* Difference in percentage and is computed as the absolute value of $\frac{Reference\ value - actual\ value}{Reference\ value} * 100.$

7.3.3. Conclusion

The phugoid, short period and Dutch roll modes are the modes that are exhibiting oscillating motions. The natural frequencies of these modes are predicted quite well, which has an average difference of five percent. The accuracy of the damping ratio, on the other hand, is very dependent on the type of mode. The mode that has the worst damping ratio is the phugoid mode, which has differences ranging from 29 percent for 'normal rotation' option to 197 percent for the 'transition surface' option. The most accurate damping ratio is the damping ratios of the short period mode, which has an average difference of nine percent. But note that the damping ratio of the models for the phugoid mode is very small, which

exaggerate the error percentage of the damping ratio. The phugoid mode is predicted quite good when eigenvalues of the phugoid mode are plotted in an eigenvalue plot. The eigenvalues of the phugoid mode are clustered very closely together.

The spiral and the roll mode are the non-oscillating modes. The accuracy of the natural frequency for the roll mode is acceptable with an average error of ten percent. The accuracy of the spiral modes is predicted quite good when the eigenvalues of the spiral mode are plotted in an eigenvalue plot. The eigenvalues of the spiral mode are located closely together.

The 'normal rotation' option is the slightly better option for predicting the phugoid and the short period mode. Except for the damping ratio of the phugoid mode, the damping ratio of this mode is predicted three times better by the 'normal rotation' option. The 'transition surface' option, on the other hand, is better in predicting the spiral and Dutch roll modes, while the 'normal rotation' option is better in predicting the roll mode.

8

Conclusion & Recommendations

8.1. Conclusion

This thesis is part of a project that focuses on developing an optimisation framework for dynamically scaled flight testing. The optimisation framework must design the scale model and the flight test such that the performance of the scale model is representative to the full-scale aircraft. This thesis contributes to the optimisation module that optimises the aerodynamics of the scale model. This thesis contributes to this framework by focusing on the development of a parametric model that is capable of deriving the stability and control derivatives.

The objective of this thesis is to create a generic parametric model that can be coupled with a first order panel method tool with an integral boundary layer method to derive the stability & control derivatives of the model. The goals of the thesis are mostly related to the construction of the parametric model with moveables for the VSAERO tool such that it produces acceptable aerodynamic results. The most important goal is how to model the trailing edge moveables in VSAERO. Three modelling options were investigated and they are the 'normal rotation', 'transition surface' and 'gap' moveable modelling options. The 'normal rotation' option is a mathematical operation in VSAERO that rotates the normal vectors of the body panels that represents the moveable. The 'transition surface' option is a model with moveables that are not-slotted and has transition surfaces in the spanwise direction. The 'gap' option is also a wing with not-slotted moveable but without transition surfaces in the spanwise direction. The research goals of this thesis were:

- Determine the best option out of the three proposed moveable options to model the moveables of a wing for an aerodynamic solver that is based on the first order panel method.
- Compute the stability & control derivatives of the parametric model and investigate the accuracy of these derivatives.

Out of the three modelling options, the not-slotted moveable model with a gap between the sides of the moveable and the wing was useless in VSAERO. The model was unstable due to the presence of the gap. The gap creates a very low pressure locally, which accelerates the surrounding airflow to a ridiculously high value. This flaw makes the moveable modelling method with a gap between the wing and moveable infeasible for a first order panel method like VSAERO.

No differences between the 'normal rotation' and the 'transition surface' options were observed when the moveable deflection angle was zero. Some differences between the two remaining modelling options were observed when the moveables were deflected.

Some differences between the 'normal rotation' and the 'transition surface' model can be observed when the moveables are deflected. Deflecting a moveable creates a suction peak at the aft part of the wing. The location of this suction peak is different for the two moveable models. The suction peak of the 'normal rotation' model is located on the wing in front of the start of the moveable. The suction peak of the 'transition surface' on the other hand is located at the start or slightly behind the start of the moveable. The effect of the different locations of the suction peak is that a suction peak on the wing will generate less induced drag than a suction peak located on the deflected moveable. The lift

generated by the suction peak of 'normal rotation' moveable is pointed less to the aft when compared to the suction peak of 'transition surface' model. The results is a lower induced drag and a slightly higher lift for the 'normal rotation' model. The error margin between the minimum drag coefficient of a model and the reference data is larger for the 'normal rotation' model than for the 'transition surface' model.

The results of stability & control derivatives for 'normal rotation' and the 'transition surface' were quite acceptable with the static derivatives as the most accurate derivatives with an average error of seven percent. The control derivatives, on the other hand, was the least accurate with an average error of 40 percent. The overall performance of the two moveable modelling options is that they were performing equally well when only the accuracy of the derivatives was observed. The 'transition surface' modelling option was more accurate in the prediction of lateral stability derivatives and the longitudinal control derivatives, while the 'normal rotation' option was more accurate in predicting the longitudinal stability derivatives and the lateral control derivatives and. But the accuracy of the side force, yaw moment and roll moment due to the deflecting control surface were very bad for the 'normal rotation' option when it is compared to the accuracy of the 'transition surface' option.

The effects of simulated derivatives on the stability motions were investigated using equations that approximated the eigenmotions of the aircraft. The predictions of the stability motions were acceptable. The 'normal rotation' option was slightly better in predicting the phugoid mode and the short period mode than the 'transition surface' option. The 'normal rotation' was better in predicting the roll mode of the aircraft, while the 'transition surface' option was better in predicting the spiral and the Dutch roll mode.

The validation of the aerodynamic characteristics of the Fokker100 aircraft has shown that the 'transition surface' modelling option is the better option because it produced more accurate aerodynamic results than the 'normal rotation' modelling option. The validation of the stability and control derivatives, on the other hand, has shown that neither of the options is better than the other one. Both moveable modelling options have its strengths and weakness when concerning the prediction of the stability and control derivatives. Thus based on the accuracy of the aerodynamic characteristics, the 'transition surface' modelling option is the best option for modelling the moveables of a first order panel method solvers like VSAERO.

8.2. Recommendations

This section lists some recommendations about some future work that can be performed on the parametric aircraft tool called Multi-Model Generator. Also, some other general recommendations are given that are related to the thesis and the aerodynamic solver VSAERO.

General

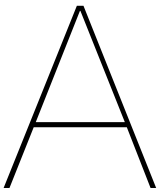
- The validation of the different moveable options were only performed on the Fokker100 aircraft. Different validations should be carried with different aircraft models.
- Investigate the effect of the boundary layer integral method. Since this investigation was not conducted in this thesis and the standard VSAERO values were used to for the boundary layer calculations. The results have shown that the drag was underestimated while using viscous iterations to produce the results.
- The effect of the shape of the stitched wake lines of the wings was investigated a little bit in this thesis. Future work can investigate this further in more details with a lot more different shapes for the stitched wake lines.
- The coupling of the Multi-Model Generator and the aerodynamic solver VSAERO can be automated, such that it automatically starts running VSAERO with the output file created by the mesher tool.
- A coupling should be made between Multi-Model Generator, VSAERO and the flight mechanics toolbox of Mark Voskuijl to analyse the effect of stability & control derivatives on the stability and the handling qualities of the aircraft model.

Dynamically scaled flight testing

- Create a scale model and a full-scale model and compare the results with each other and validate it against reference data.
- Investigate the accuracy of the scale models at high Mach number, where there are compressibility effects.
- Create an optimisation framework that optimises the shape of the aircraft to reduce the aerodynamic differences between the scale model and the full-scale model.
- Couple the aerodynamic optimisation process with the optimisation process that optimises the mass distribution of the scale model.

Multi-Model Generator

- The moveables are currently working for the trailing edges moveables. This tool can be expanded to include the leading moveables.
- The current moveable modelling code does not allow the moveable to cross the kink of the wing. Expand this code such that is possible to model a moveable that goes beyond the kink of the wing.
- Expand the code for generating different possible stitching edges for the wing wake. The current version can only produce one stitching edge that does not change when flight conditions are changed.
- The current aircraft mesher can only mesh the wing, where there is an intersection on both sides of the wing like a T-tail. The mesher will not work if there is only one intersection at one side of the wing like a vertical fin on the blended wing body aircraft. So future work can expand this mesher or create a new mesher that can cope with this flaw.
- Expand or create a new meshing tool such that it produce models for CFD simulations.
- Expand or create a new moveable modelling code to include flaps and slats with more than one element.



Scale factors for the scaling principles

Table A.1: Scale factors for Froude, Mach and Reynolds scaling principles related to atmospheric test conditions [3].

Parameter	Ratio	Scaling factor		
		Froude scaling	Mach scaling	Reynolds scaling
Length	l_m/l_a	n	n	n
Vehicle attitude	α'_m/α'_a	1	1*	1*
Control surface deflection	δ_m/δ_a	1	1	1
Froude number	$\left(\frac{V^2}{lg}\right)_m/\left(\frac{V^2}{lg}\right)_a$	1	$\frac{g_a}{g_m}\left(\frac{V_{S,m}}{V_{S,a}}\right)^2\frac{1}{n}=1$**	$\frac{g_a}{g_m}\left(\frac{\nu_m}{\nu_a}\right)^2\frac{1}{n^3}=1$**
Mach number	$\left(\frac{V}{V_S}\right)_m/\left(\frac{V}{V_S}\right)_a$	$n^{1/2}\frac{V_{S,a}}{V_{S,m}}$	1	$\frac{1}{n}\frac{\nu_m}{\nu_a}\frac{V_{S,a}}{V_{S,m}}$
Reynolds number	$\left(\frac{Vl}{\nu}\right)_m/\left(\frac{Vl}{\nu}\right)_a$	$\frac{\nu_a}{\nu_m}n^{3/2}$	$\frac{\nu_a}{\nu_m}\frac{V_{S,m}}{V_{S,a}}n$	1
Gravity field	g_m/g_a	1	$\left(\frac{V_{S,m}}{V_{S,a}}\right)^2\frac{1}{n}$	$\left(\frac{V_m}{V_a}\right)^2\frac{1}{n}$
Velocity	V_m/V_a	$n^{1/2}$	$\frac{V_{S,m}}{V_{S,a}}$	$\frac{1}{n}\frac{\nu_m}{\nu_a}$
Fluid density	ρ_m/ρ_a	$\frac{\rho_m}{\rho_a}$	$\frac{\rho_m}{\rho_a}$	$\frac{\rho_m}{\rho_a}$
Dynamic pressure	$\overline{q}_m/\overline{q}_a$	$\frac{\rho_m}{\rho_a}n$	$\frac{\rho_m}{\rho_a}\left(\frac{V_{S,m}}{V_{S,a}}\right)^2$	$\frac{\rho_m}{\rho_a}\left(\frac{1}{n}\frac{V_m}{V_a}\right)^2$
Reduced angular rate	$\left(\frac{\Omega l}{V}\right)_m/\left(\frac{\Omega l}{V}\right)_a$	1	1	1
Angular rate	Ω_m/Ω_a	$\frac{1}{n^{1/2}}$	$\frac{1}{n^{1/2}}\frac{V_{S,m}}{V_{S,a}}$	$\frac{1}{n^2}\frac{\nu_m}{\nu_a}$
Reduced time	$\left(\frac{tV}{l}\right)_m/\left(\frac{tV}{l}\right)_a$	1	1	1
Time	t_m/t_a	$n^{1/2}$	$n\frac{V_{S,a}}{V_{S,m}}$	$n\frac{V_a}{V_m}$
Angular displacement	φ'_m/φ'_a	1	1	1
Angular acceleration	$\dot{\Omega}_m/\dot{\Omega}_a$	$\frac{1}{n}$	$\left(\frac{V_{S,m}}{V_{S,a}}\right)^2\frac{1}{n^2}$	$\left(\frac{\nu_m}{\nu_a}\right)^2\frac{1}{n^4}$
Linear displacement	s_m/s_a	n	n	n
Linear acceleration	a_m/a_a	1	$\left(\frac{V_{S,m}}{V_{S,a}}\right)^2\frac{1}{n}$	$\left(\frac{\nu_m}{\nu_a}\right)^2\frac{1}{n^3}$
Strouhal number	$\left(\frac{\omega l}{V}\right)_m/\left(\frac{\omega l}{V}\right)_a$	1	1	1
Oscillatory frequency	ω_m/ω_a	$\frac{1}{n^{1/2}}$	$\frac{1}{n^{1/2}}\frac{V_{S,m}}{V_{S,a}}$	$\frac{1}{n^2}\frac{\nu_m}{\nu_a}$

* Ability to satisfy attitude scaling is dependent on satisfying Froude number similitude.
** Satisfying Froude number similitude is not possible due to $g_m = g_a$. Exception is that $n = V_{S,m}/V_{S,a}$ for Mach scaling and $n = V_m/V_a$ for Reynolds scaling.

Table A.2: Scale factors for Froude, Mach and Reynolds scaling principles related to model properties [3].

Parameter	Ratio	Scaling factor			
		Rigid model, Froude	Flexible model		
		Mach and Reynolds scaling	Froude scaling	Mach scaling	Reynolds scaling
Length	l_m/l_a	n	n	n	n
Relative density factor	$\left(\frac{m}{\rho l^3}\right)_m / \left(\frac{m}{\rho l^3}\right)_a$	1	1	1	1
Mass	m_m/m_a	$\left(\frac{\rho_m}{\rho_a}\right) n^3$	$\left(\frac{\rho_m}{\rho_a}\right) n^3$	$\left(\frac{\rho_m}{\rho_a}\right) n^3$	$\left(\frac{\rho_m}{\rho_a}\right) n^3$
Relative mass moment of inertia	$\left(\frac{I}{\rho l^5}\right)_m / \left(\frac{I}{\rho l^5}\right)_a$	1	1	1	1
Mass moment of inertia	I_m/I_a	$\left(\frac{\rho_m}{\rho_a}\right) n^5$	$\left(\frac{\rho_m}{\rho_a}\right) n^5$	$\left(\frac{\rho_m}{\rho_a}\right) n^5$	$\left(\frac{\rho_m}{\rho_a}\right) n^5$
Radius of gyration	k_m/k_a	n	n	n	n
Aeroelastic bending	$\left(\frac{EI'}{\rho V^2 l^4}\right)_m / \left(\frac{EI'}{\rho V^2 l^4}\right)_a$	- - - -	1	1	1
Bending stiffness	$(EI')_m / (EI')_a$	- - - -	$\left(\frac{\rho_m}{\rho_a}\right) n^5$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{V_{S,m}}{V_{S,a}}\right)^2 n^4$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{V_m}{V_a}\right)^2 n^2$
Area moment of inertia	I'_m/I'_a	- - - -	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{E_a}{E_m}\right) n^5$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{E_a}{E_m}\right)\left(\frac{V_{S,m}}{V_{S,a}}\right)^2 n^4$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{E_a}{E_m}\right)\left(\frac{V_m}{V_a}\right)^2 n^2$
Aeroelastic torsion	$\left(\frac{GJ'}{\rho V^2 l^4}\right)_m / \left(\frac{GJ'}{\rho V^2 l^4}\right)_a$	- - - -	1	1	1
Torsional stiffness	$(GJ')_m / (GJ')_a$	- - - -	$\left(\frac{\rho_m}{\rho_a}\right) n^5$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{V_{S,m}}{V_{S,a}}\right)^2 n^4$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{V_m}{V_a}\right)^2 n^2$
Area torsional moment of inertia	J'_m/J'_a	- - - -	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{G_a}{G_m}\right) n^5$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{G_a}{G_m}\right)\left(\frac{V_{S,m}}{V_{S,a}}\right)^2 n^4$	$\left(\frac{\rho_m}{\rho_a}\right)\left(\frac{G_a}{G_m}\right)\left(\frac{V_m}{V_a}\right)^2 n^2$

Bibliography

[1] J. Chambers, *Modeling Flight NASA Latest Version: The role of dynamically scale Free Flight Models in support of NASA aerospace programs*, Vol. 3 (Joseph Chambers, 2015).

[2] D. B. Owens, J. M. Brandon, M. A. Croom, C. M. Fremaux, E. H. Heim, and D. D. Vicroy, *Overview of dynamic test techniques for flight dynamics research at nasa larc,* AIAA Paper **3146**, 5 (2006).

[3] C. H. Wolowicz, J. Bowman Jr, and W. P. Gilbert, *Similitude requirements and scaling relationships as applied to model testing*, Tech. Rep. (NASA, 1979).

[4] B. Zohuri, *Dimensional Analysis and Self-Similarity Methods for Engineers and Scientists*, Vol. 3 (Springer International Publishing, 2015).

[5] D. van Ginneken, *Automatic Control Surface Design & Sizing for the Prandtl Plane*, Msc. thesis, Delft University of Technology (2009).

[6] B. Senden, *The Aerodynamic Performance of a High-Lift System with an Elliptic Flap*, Msc. thesis, Delft University of Technology (2015).

[7] J. Grotenhuis, *Development of a Conceptual Controllability Analysis Tool*, Msc. thesis, Delft University of Technology (2007).

[8] M. Arntzen, *Elastic aircraft aerodynamics: Improving DARloads aerodynamics to enhance load calculations*, Msc. thesis, Delft University of Technology (2010).

[9] G. La Rocca, L. Krakers, and M. J. van Tooren, *Development of an icad generative model for blended wing body aircraft design,* in *AIAA 2002 conference, USA, Sept 2002 AIAA 2002 5447* (2002).

[10] G. La Rocca and M. J. van Tooren, *A modular reconfigurable software modelling tool to support distributed multidisciplinary design and optimisation of complex products,* in *16th CIRP International Design Seminar* (2006).

[11] T. van den Berg, *Parametric modeling and aerodynamic analysis of multi-element wing configurations*, Msc. thesis, Delft University of Technology (2009).

[12] J. H. Koning, *Development of a KBE application to support aerodynamic design and analysis. TOWARDS A NEXT-GENERATION MULTI-MODEL GENERATOR.*, Msc. thesis, Delft University of Technology (2010).

[13] M. van Hoek, *Structural Design, Analysis and Optimization of a Lifting Surface in a Knowledge Based Engineering Environment*, Msc. thesis, Delft University of Technology (2009).

[14] M. Abedian Kalkhoran, *Development of a new Capability Module to automate the aerodynamic analysis of complete aircraft configurations by means of a commercial panel code*, Msc. thesis, Delft University of Technology (2014).

[15] ParaPy B.V., *ParaPy website,* (September 2016), https://www.parapy.nl.

[16] Y. Brouwers, *Development of KBE applications to support the conceptual design of passengers aircraft fuselages. Further steps towards the development of a Design Engineering Engine for aircraft MDO*, Msc. thesis, Delft University of Technology (2011).

[17] B. M. Kulfan, *Universal parametric geometry representation method,* Journal of Aircraft **45**, 142 (2008).

[18] J. Nathman, *VSAERO: A Code for Calculating the Nonlinear Aerodynamic Characteristics of Arbitrary Configuration*, ANALYTICAL METHODS, INC., version 7.7 ed. (2014), users' Manual.

[19] J. Katz and A. Plotkin, *Low-Speed Aerodynamics*, 2nd ed. (Cambridge University Press, 2001).

[20] Anonymous, *Fokker 100 aerodynamic characteristics data, internal report,* Delft University of Technology.

[21] J. Mulder, W. van Staveren, J. van der Vaart, E. de Weerdt, C. de Visser, A. in 't Veld, and E. Mooij, *AE3202 Flight Dynamics Reader*, Tech. Rep. (Delft University of Technology, March 2013).

[22] R. C. Nelson, *Flight stability and automatic control*, Vol. 2 (WCB/McGraw Hill, 1998).