

# Towards Computationally Efficient Coupled Propeller- Wing Optimisation

Master of Science Thesis  
Aerospace Engineering - Power and Propulsion

João Quinten Benjamin Exalto  
January 2023



# Towards Computationally Efficient Coupled Propeller-Wing Optimisation

Master of Science Thesis  
Aerospace Engineering - Power and Propulsion

by

João Quinten Benjamin Exalto

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Friday January 13, 2023 at 2:00 PM.



Student number: 4647882

Project duration: September 1, 2021 - January 13, 2023

Thesis committee: Dr. ir. G. La Rocca,  
Dr. ir. M. F. M. Hoogreef,  
Prof. Dr. J. R. R. A. Martins<sup>1</sup>,  
M. Eng. B. Pacini,  
Dr. ir. D. M. J. Peeters,

TU Delft, FPP, chair

TU Delft, FPP, supervisor

University of Michigan, MDOLab, supervisor

University of Michigan, MDOLab, supervisor

TU Delft, ASM, ASCM

---

<sup>1</sup>Pauline M. Sherman Collegiate Professor

# Preface

My time at the University of Michigan has unequivocally been the most enjoyable year out of my life. Both in terms of the friends I made and the stellar supervision from the MDO Lab, with special thanks to Bernardo. I also want to thank Maurice for being so incredibly understanding and always having useful feedback. I honestly could not have asked for a better thesis supervisor. Lastly, I obviously could not have done any of this without the help and support of my friends and family. Even the shortest talks or signs of support meant a huge deal to me. The list is too long to include here but anybody in my direct environment will know who I am talking about, ranging from colleagues, roommates and fellow rowers in Michigan and Delft to high school friends to other people I had the privilege of meeting during my 64 months as a student.

I have genuinely enjoyed diving into the world of Multidisciplinary Design Optimisation, and facing the adversities that I came across. As Bernardo once told me: *Us integrators and framework developers are in the uncomfortable position of working with the models given to us and deal with their flaws.*

*João Quinten Benjamin Exalto  
Delft, January 2023*

# Summary

Much like the oil crisis in the 1980s, the aviation industry is once again considering more environmentally sustainable propulsion options due to concerns around climate change. More fuel efficient propulsion methods form the pillar on which environmentally sustainable aviation can be built, since it is the propulsion system that emits harmful greenhouse gasses. The propulsion system consists of an energy carrier, currently kerosene, and an energy to thrust converter, often a jet engine. Although efficiency gains are being made, using kerosene and jet engines will prove infeasible in the long run due to global warming concerns. Replacing the jet engine with a propeller offers an attractive solution since propeller shaft power can be supplied by batteries or fuel cells. Furthermore, propellers are relatively efficient due to their infinite bypass ratio. Additionally, a surge in demand for Urban Air Mobility [3] incentives propeller optimisation studies.

Aircraft are a combination of complex and interacting systems. For this reason it is important to consider interactions between wings and propellers when designing either. Propeller-wing optimisation is therefore an increasingly important topic. Propeller-wing optimisation literature is scarce, likely due to the complexity of a coupled propeller-wing system. Optimising a propeller-wing system is possible with high-fidelity simulations but often takes a substantial amount of time. The aim of this research is to address the lack of coupled propeller-wing aerostructural optimisation. The knowledge gap is addressed by designing a novel coupled propeller wing framework that is suited for computationally efficient optimisation studies. Furthermore, the optimisation framework will be modular such that it can be easily extended. The ability to expand the framework increases the scope and impact of this research.

Computational efficiency for gradient-based optimisation is determined by the model choice, the gradient assessment method, and the optimisation architecture. This research project encompasses the former two and leaves the optimisation algorithm out of its scope: The optimisation algorithm used is SNOPT [13], a Sequential Quadratic Programming algorithm. Modelling choices play an important role since computational efficiency is often inversely related to accuracy. Therefore, the models should be carefully chosen such that all are of the desired level of fidelity and computational expense. Additionally, the gradient assessment method is arguably the most important choice to determine the accuracy and computational efficiency of an optimisation architecture. The cost of gradient-based optimisation scales with either the number of design variables or objectives and constraints. Whether this scaling is linear (thus poor) can heavily impact the computational efficiency of the optimisation framework. For instance, a non-intrusive method such as finite differences scales linearly with the number of design variables. For each new design variable the function has to be perturbed and compared to the non-perturbed function. Furthermore, finite difference schemes can return inaccurate derivatives if the design space has a noisy output function. Intrusive methods such as the direct and adjoint method scale with the number of inputs or outputs, respectively. Thus, the number of inputs or outputs can be increased without any additional computational cost. However, the implementation of these intrusive gradient assessment methods is non-trivial. The adjoint and direct method require derivative verification procedures. Derivative verification can be performed by comparing the model's derivatives to the derivatives returned by the complex-step method. The complex-step is accurate to machine precision in terms of accuracy. However, it is slow and scale linearly with the number of design variables. For these reasons, the complex-step method should only be used for derivative verification.

The design methodology consists of three chapters that each contain *model theory*, *derivative verification* and *model validation*. The three chapters describe the propeller, wing and slipstream model.

The propeller model uses a hybrid blade element momentum (HBEM) model called HELIX. HBEM is an adaptation of blade element momentum (BEM) theory that improves the classical BEM's versatility by including analyses for hover and forward flight (in a helicopter configuration) configurations. BEM calculates propeller performance by relying on blade element theory and momentum theory to converge to the same thrust coefficient values. HELIX uses a stall model to account for airfoil stall and transition. The input variables for the sectional airfoil lift are lift slope, zero lift angle of attack, and the stall angle. The stall

model significantly improves the fidelity of the propeller model since it prevents the optimiser from exploiting non-physical angles of attack. The only modification that was made to the propeller was including a subsystem that returns the propeller's velocity distribution as this was required for the slipstream model. Although HELIX has been configured to optimise blade chord and sectional airfoil characteristics, it should be noted that including some of these variables in the design variable vector will return non-feasible results. For instance, it was seen that including propeller chord as a design variables would return a propeller with a maximised chord value near the root, whereas the remainder of the propeller chord would be at the lower bound. The model validation showed that the propeller code overestimates lift at higher advance ratios. The thrust over-estimation at higher advance ratios should be considered when interpreting the optimisation results and coupled model validation.

The wing's aerostructural properties are modelled using the open source code OpenAeroStruct [16] (OAS). OAS uses a Vortex Lattice Method (VLM) to model the wing's performance and either a wingbox or tube model to predict structural performance. OAS also considers the wing deformation due to aerodynamic loads, and its effect on wing aerodynamics. After the wing is deformed, the updated wing aerodynamic performance is assessed. OAS iterates on this deformation until the system converges, after which the wing characteristics are returned. OAS required a number of modifications. The first being a remeshing function. The remeshing function is necessary since the propeller must align with the VLM panels. The remeshing function guarantees continuous propeller location and radius design variables. However, it is expected that the remeshing function introduces an error that is accumulating over the optimisation iterations. The second modification is important for coupling the wing to the slipstream model. The correction factor matrix returned by the slipstream model (discussed in the next paragraph) is added to the Aerodynamic Influence Coefficient (AIC) matrix. Lastly, the velocity vector, similarly returned by the slipstream model, has to be incorporated in the VLM system. It is important to note that any model modifications are accompanied by model derivative modifications as well.

The propeller and wing model are coupled with a slipstream model. The slipstream model is based on the Rethorst correction factor [31] and returns a correction factor matrix. The correction factor guarantees that the slipstream boundary conditions are satisfied. Without the correction factor the VLM would consider the slipstream to be of infinite height. In reality, the slipstream has a circular shape. The correction factor guarantees that the slipstream has this circular shape. The Rethorst correction factor was used in several wing-propeller analysis studies. Optimisation however introduces a number of challenges that do not have to be solved for analysis purposes. One of the requirements for optimisation is to have a system that is fully continuous. Additionally, the Rethorst correction factor requires the VLM panel to be in the centre of the propeller slipstream. Furthermore, the slipstream edges have to be aligned with the slipstream boundaries. For analysis, one could simply create a VLM mesh that satisfies both of these conditions. This is not possible for optimisation since it would require remeshing each iteration if the wing span, propeller radius or propeller location change. Remeshing introduces a highly discontinuous system that will cause the optimiser to diverge. To solve this issue, an overset mesh was configured for the correction factor. The overset mesh scales with the propeller radius and translates to the left and right with the propeller location. The overset mesh always covers the entire wing. The correction factor of the overset mesh is interpolated to the VLM mesh and neglects the values that are outside the wing span. This approach is similar to how wing-tip propellers are simulated in previous studies using the Rethorst correction factor. Together with the remeshing function in OAS this approach returns a correction factor and a velocity factor that are provided to OAS. Another important note about the Rethorst correction factor is that it consists of an even and odd solution. The odd solution is however computationally expensive due to the nested integrals with Bessel functions. Research by Nederlof [25] shows that the even solution alone returns an acceptable solution. For these reasons the odd solution is neglected in the current slipstream model. The slipstream model does contain the code to include the odd solution. In future iterations of the framework a more efficient algorithm for Bessel function assessment [1] should be used to increase the fidelity of the model. The model derivatives were verified by comparing to a finite differences scheme.

Model coupling was done using OpenMDAO [14], an open-source optimisation framework well suited for gradient based optimisation. An eXtended Design Structure Matrix (XDASM) is given in Figure 1. The propeller

model passes the propeller velocity distribution to the slipstream model, that will return a correction matrix and velocity vector. The correction factor and velocity vector are included in the VLM system that assesses the aerodynamic performance of the wing. Finally, using the returned lift, drag, weight, and propeller power the constraints and objective functions can be calculated.

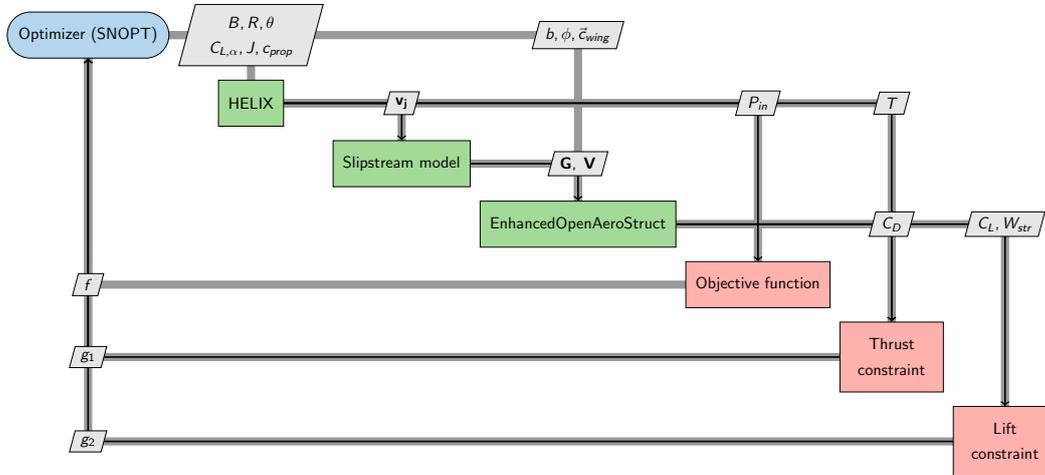


Figure 1: The eXtended Design Structure Matrix for the wing-propeller system

The coupled wing-propeller optimisation returned a slower spinning propeller and the twist was adjusted such that the thrust constraint was satisfied, the isolated wing optimisation resulted in a twist distribution that was lowest at the tips and highest at the root. The twist did however slightly reduce behind the propeller. This is likely because the propellers have a very significant impact on the lift distribution. The propeller augmented wing section produces substantially more lift than the rest of the wing, even with smaller twist values. The optimisation results could be affected by modelling or numerical issues. However, the aim of this research was to show that coupled aerostructural wing-propeller optimisation is possible, which was achieved. Furthermore, it was noted that the optimiser tended to move the propellers inboard. The optimiser moved the propellers inboard due to the coupled model's inability to model tangential velocity components. Wing-tip propellers that rotate inboard-up could reduce the induced drag by increasing the effective aspect ratio [34]. The effective aspect ratio increases by wing tip vortex dispersion, which can only be accounted for if swirl is included in the slipstream model.

The title of this dissertation is *Towards Computationally Efficient Aerostructural Coupled Wing-Propeller Optimisation*. Therefore, the methodological results and recommendations are almost as important as the optimisation results. It was discovered that the remeshing function is prone to introduce asymmetries in the system. Asymmetries in the remeshing function are negligible for analysis purposes. However, the adjoint method uses partial derivatives. If the remeshing function is asymmetric it also introduces asymmetries in the differentiated code and could lead the optimiser to non-physical results (for instance moving one propeller inboard and the other outboard). The asymmetries in the model were resolved but any future modifications should take asymmetries between the left and right side of the wing into account. Future iterations of the framework could also include slipstream contraction and deflection, acoustic analysis, a model for tangential flow components and the odd Rethorst correction. It should also be noted that the current wing structural tube model should be replaced with a wingbox model.

The research project resulted a novel propeller wing optimisation framework. Future iterations of the framework can elaborate the framework and include other relevant aerodynamic, structural or acoustic models. The framework has its flaws and serves as a preliminary version that shows the feasibility of computationally efficient coupled mid-fidelity aerostructural wing-propeller optimisation.

# Contents

Preface	i
Summary	ii
Nomenclature	vii
List of Figures	x
List of Tables	xii
1 Introduction	1
1.1 Research Objective	2
1.2 Potential Solution	2
1.3 Scope and Limitations	3
1.4 Research Outline	4
2 Propeller Model	6
2.1 The Propeller Model	6
2.1.1 Blade Element Momentum Theory	6
2.1.2 HELIX	8
2.1.3 Velocity Distribution	9
2.1.4 Limitations of a BEM model	9
2.2 Model Validation	10
2.3 Derivative Verification	10
3 Wing Model	12
3.1 AeroStructural Model	12
3.1.1 Aerodynamic Wing Model	12
3.1.2 Structural Wing Model	13
3.1.3 AeroStructural Converge	15
3.2 OAS Modifications	15
3.2.1 Remeshing Function	15
3.2.2 Propeller Weight Sizing	16
3.2.3 Wing to Propeller Feedback	17
3.2.4 Wing Model Limitations	18
3.3 Derivative Verification	18
4 Slipstream Model	19
4.1 Wing Crossing a Finite Slipstream	19
4.1.1 The Rethorst Model	19
4.1.2 Limitations of the Rethorst Model	22
4.2 Slipstream Model Implementation	22
4.2.1 The Overset Mesh	23
4.2.2 Wing-Tip Propellers	27
4.2.3 Performance Enhancements	28
4.2.4 Overset Accuracy Assessment	28
4.2.5 Slipstream Velocity Profile	29
4.2.6 Limitations of an Overset Mesh	31

4.3	Derivative Verification . . . . .	32
4.3.1	Finite Differences . . . . .	32
4.3.2	Complex Step Differences . . . . .	33
4.3.3	Forward AD Verification . . . . .	34
4.3.4	Reverse AD Verification . . . . .	35
4.3.5	Derivative Verification Results . . . . .	36
4.4	Slipstream Model Limitations . . . . .	36
5	Model Coupling . . . . .	38
5.1	Model Coupling . . . . .	38
5.1.1	Linear Solver Converge . . . . .	39
5.2	Coupled Model Validation . . . . .	40
5.2.1	Inboard Propeller Configuration Validation . . . . .	40
5.2.2	Wingtip Propeller Validation . . . . .	42
5.3	Derivative Verification . . . . .	43
5.4	The Adjoint and Direct Method . . . . .	43
6	Optimisation Results . . . . .	47
6.1	Isolated Propeller Optimisation . . . . .	47
6.2	Isolated Wing Optimisation . . . . .	48
6.3	Coupled Wing-Propeller Optimisation . . . . .	49
6.4	Efficiency and Accuracy Assessment . . . . .	52
7	Methodological Requirement . . . . .	53
7.1	Optimisation Specific Requirements . . . . .	53
7.1.1	Remeshing Function . . . . .	53
7.1.2	Model Symmetry . . . . .	53
7.2	Recommendations . . . . .	54
7.2.1	Odd Rethorst Correction . . . . .	55
7.2.2	Nonuniform Propeller Inflow Field . . . . .	55
7.2.3	Slipstream Contraction . . . . .	56
7.2.4	Slipstream Deflection . . . . .	57
7.2.5	Tangential Velocity . . . . .	57
7.2.6	AeroAcoustics . . . . .	58
7.2.7	Mesh Warping . . . . .	58
7.2.8	Structural Model . . . . .	58
7.2.9	Computational Efficiency Correction Matrix . . . . .	58
7.2.10	Wing to Propeller Feedback . . . . .	59
8	Conclusion . . . . .	60
	References . . . . .	63

# Nomenclature

## Abbreviations

Abbreviation	Definition
AD	Algorithmic Differentiation
AIC	Aerodynamic Influence Coefficients
BEM	Blade Element Momentum
CS	Complex Step
DEP	Distributed Electric Propulsion
EM	Electrical Motor
FD	Finite Differences
ESP	Equivalent Specific Power
FEM	Finite Element Method
HBEM	Hybrid Blade Element Momentum
MDO	Multidisciplinary Design Optimisation
OAS	OpenAeroStruct
PMAD	Power Management and Distribution
SNOPT	Sparse Nonlinear Optimizer
UAV	Urban Air Vehicles
VLM	Vortex Lattice Method

## Symbols

Symbol	Definition	Unit
$a$	Axial Induction Factor	[-]
$a'$	Tangential Induction Factor	[-]
$\mathbf{A}$	Aerodynamic Influence Coefficient matrix	[1/m]
$A_d$	Annulus Area	[m <sup>2</sup> ]
$b$	Span	[m]
$b_{\text{overset}}$	Overset Span	[m]
$B$	Number of Blades	[-]
$c$	Chord length	[m]
$C_T$	Thrust Coefficient	[-]
$C_l$	Lift Coefficient	[-]
$C_{l,\alpha}$	Lift Coefficient Gradient	[1/rad]
$C_d$	Drag Coefficient	[-]
$C_n$	Normal Coefficient	[-]
$C_T$	Thrust Coefficient	[-]
$C_{T,BET}$	Thrust Coefficient, Blade Element Theory	[-]
$C_{T,MT}$	Thrust Coefficient, Momentum Theory	[-]
$d$	Sectional Drag	[-]
$D$	Drag	[-]
$e$	Vortex Edge Location	[m]
$E_{0,tot}$	Total Energy	[-]
$f$	Vortex Edge Location	[m]

Symbol	Definition	Unit
$f_{total}$	Prandtl Tip Correction	[-]
$g$	Gravitational Constant	[kg m/s <sup>2</sup> ]
$G$	Rethorst Correction Factor	[kg m/s <sup>2</sup> ]
$\mathbf{G}$	Rethorst Correction Factor Matrix	[kg m/s <sup>2</sup> ]
$G_{even_{oo}}$	Even Rethorst Correction Factor, panel and control point inside slipstream	[kg m/s <sup>2</sup> ]
$G_{even_{oj}}$	Even Rethorst Correction Factor, panel inside and control point outside slipstream	[kg m/s <sup>2</sup> ]
$G_{even_{jo}}$	Even Rethorst Correction Factor, panel outside and control point inside slipstream	[kg m/s <sup>2</sup> ]
$G_{even_{jj}}$	Even Rethorst Correction Factor, panel and control point outside slipstream	[kg m/s <sup>2</sup> ]
$G_{odd_{oo}}$	Odd Rethorst Correction Factor, panel and control point inside slipstream	[kg m/s <sup>2</sup> ]
$G_{odd_{oj}}$	Odd Rethorst Correction Factor, panel inside and control point outside slipstream	[kg m/s <sup>2</sup> ]
$G_{odd_{jo}}$	Odd Rethorst Correction Factor, panel outside and control point inside slipstream	[kg m/s <sup>2</sup> ]
$G_{odd_{jj}}$	Odd Rethorst Correction Factor, panel and control point outside slipstream	[kg m/s <sup>2</sup> ]
$h$	Increment	[-]
$i$	Imaginary Number	[-]
$I$	Bessel Function of first kind	[-]
$I'$	Derivative of Bessel Function of first kind	[-]
$J$	Propeller Advance Ratio	[-]
$K$	Bessel Function of second kind	[-]
$K'$	Derivative of Bessel Function of second kind	[-]
$l$	Sectional Lift	[-]
$L$	Lift	[-]
$p_s$	Static Pressure	[-]
$p_t$	Total Pressure	[-]
$Q$	Torque	[N · m]
$r_{prop}$	Propeller Radius	[m]
$r_{min}$	Propeller Minimum Radius	[m]
$R$	Range	[-]
$\mathcal{R}$	Residual	[-]
$T$	Thrust	[N]
$V$	Velocity	[m/s]
$\vec{V}$	Velocity Vector	[m/s]
$V_a$	Axial Velocity	[m/s]
$V_r$	Rotational Velocity	[m/s]
$V_t$	Tangential Velocity	[m/s]
$V_\infty$	Freestream Velocity	[m/s]
$V_0$	Freestream Velocity	[m/s]
$V_{jet}$	Slipstream Velocity	[m/s]
$V_j$	Slipstream Velocity	[m/s]
$W_{OE}$	Operational Empty Weight	[N]
$W_{PI}$	Payload Weight	[N]
$W_{fuel}$	Fuel Weight	[N]
$W_{MTOW}$	Maximum Take-Off Weight	[N]
$\partial$	Partial Derivative Operator	[-]
$\alpha$	Angle of Attack	[rad]

Symbol	Definition	Unit
$\eta$	Normalised Wing Spanwise Location	[-]
$\eta_{\text{propulsive}}$	Propulsive Efficiency	[-]
$\epsilon$	Relative Error	[-]
$\Gamma$	Circulation	[m]
$\xi$	Normalised Wing Chordwise Location	[-]
$\lambda$	Dummy Running Variable	[-]
$\mu$	Normalised Propeller Spanwise Location	[-]
$\mu_v$	Slipstream-Freestream Velocity Ratio	[-]
$\omega$	Rotational Velocity	[rad/s]
$\vec{\phi}$	Wing Twist	[deg]
$\Pi_{\text{panel}}$	Panel Ratio	[-]
$\rho$	Density	[kg/m <sup>3</sup> ]
$\vec{\theta}_{\text{prop}}$	Propeller Twist Vector	[deg]

# List of Figures

1	The eXtended Design Structure Matrix for the wing-propeller system . . . . .	iv
1.1	Structure of the thesis report . . . . .	4
2.1	Schematic of Blade Element Theory, where each blade section generates lift [20] . . . . .	6
2.2	Schematic of momentum theory, where the full annulus produces a momentum increase [26] . . . . .	6
2.3	Chord distribution returned by an optimisation process with sectional chord design variables . . . . .	10
2.4	PROWIM propeller <sup>2</sup> . . . . .	10
2.5	PROWIM twist distribution [34] . . . . .	10
2.6	Validation results for HELIX compared to experimental results from the PROWIM propeller . . . . .	11
3.1	A single horseshoe vortex on a lifting surface with a control point . . . . .	13
3.2	The 6 DOF are shown at each node for a beam element [27] . . . . .	14
3.3	OAS iterative solving scheme for aerostructural iteration . . . . .	15
3.4	Visualisation of the moving nodes method in which the panels closest to the jet are altered . . . . .	16
3.5	The used fully electric powertrain configuration, as given by de Vries[11] . . . . .	17
3.6	Upwash induced from wing on propeller [2] . . . . .	17
4.1	Schematic of slipstream crossing wing without Rethorst correction . . . . .	20
4.2	Schematic of slipstream crossing wing with Rethorst correction . . . . .	20
4.3	Geometry used by Rethorst [31], picture taken from Willemsen [39] . . . . .	20
4.4	The full (left), even (middle) and odd (right) mirror vortices used by Rethorst [31], picture taken from Willemsen [39] . . . . .	21
4.5	A comparison of a VLM with an without a correction factor for the slipstream, picture taken from work by van der Leer [20] . . . . .	22
4.6	Comparison between no solution, even solution and full solution, CFD data taken from Nederlof's work [25] . . . . .	23
4.7	The 'dynamic' mesh before and after moving the propeller used for wing-propeller analyses by van der Leer [20], Nederlof [25] and Willemsen [39] . . . . .	24
4.8	Visualisation of the overset mesh moving with respect to the VLM mesh by a change in propeller location . . . . .	25
4.9	Visualisation of the overset mesh scaling with respect to the VLM mesh by a change in propeller radius or wing span . . . . .	26
4.10	A schematic of how the correction factor is interpolated to the VLM mesh for an arbitrary panel . . . . .	26
4.11	Flow diagram of the Rethorst model. The parameters are passed to the multiprop function, which generates a copy of the VLM mesh by using the VLM mesh function. Afterwards, the design variables are passed to the velocity distribution function which discretises the propeller and passes it to the correction matrix function, that calculates the correction for the overset mesh. Eventually this solution is interpolated to the VLM mesh. . . . .	27
4.12	The symmetry of the Rethorst correction stems from each panel's control point having the same distance to the propeller centre. The figure shows that the calculations for the right and left side of the wing use the same input variable $\delta x$ . . . . .	28
4.13	Arbitrary Rethorst correction for a given panel with a peak (l) and no peak (r) at its centre . . . . .	29
4.14	A schematic of how the propeller velocity distribution is translated to the VLM . . . . .	30
4.15	A comparison of two slipstream velocity distributions, for which either only addition (green) or addition plus subtraction of correction factor values (red) is necessary . . . . .	30

4.16	The schematic shows how the ‘double bubble’ velocity profile can be obtained using addition and subtraction of velocity increments (green) and decrements (red) . . . . .	31
4.17	The discontinuities occur in between the black dots because the Rethorst correction switches between equations. . . . .	32
4.18	Schematic comparing the computational expense and accuracy of four different methods with which model derivatives can be obtained . . . . .	33
4.19	Forward difference scheme [23] . . . . .	33
4.20	Central difference scheme [23] . . . . .	33
4.21	Optimal step-size $h$ shown graphically [23] . . . . .	34
4.22	Example of the significance of step-size in a noisy function [23] . . . . .	34
4.23	FD versus CS accuracy comparison [23] . . . . .	35
4.24	Schematic of forward AD of an arbitrary wing model, with the red arrow representing the program direction in which the derivatives are calculated . . . . .	35
4.25	Schematic of reverse AD of an arbitrary wing model, with the red arrow representing the program direction in which the derivatives are calculated . . . . .	36
5.1	eXtended Design Structure Matrix for the coupled wing-propeller aerostructural optimization. . . . .	38
5.2	N2 diagram of the coupled wing-propeller aerostructural model . . . . .	39
5.3	N2 diagram of fully coupled system with wing to propeller feedback . . . . .	40
5.4	Schematic of the working of a linear and nonlinear solver . . . . .	40
5.5	Lift distribution for a propeller with a velocity distribution that has a double parabola shape . . . . .	41
5.6	PROWIM configured in conventional tractor configuration [34] . . . . .	41
5.7	Coupled wing-propeller validation results for inboard propeller . . . . .	42
5.8	PROWIM configured in wingtip tractor configuration [34] . . . . .	42
5.9	Coupled wing-propeller validation results for wingtip propeller . . . . .	43
5.10	Constraint imposed on $x$ and $u$ by residual equation [23] . . . . .	45
5.11	An overview of the different available linear solvers in OpenMDAO [23] . . . . .	45
5.12	Comparison of computation cost for direct-and adjoint-method [23] . . . . .	46
6.1	Slipstream velocity profile and twist and radius distribution of original and optimised propeller . . . . .	48
6.2	Chord and twist distributions of an isolated optimised wing (top) and lift distribution (bottom) . . . . .	49
6.3	Optimised lift distribution . . . . .	50
6.4	The coupled propeller’s velocity distribution (top) and twist distribution (bottom) . . . . .	50
6.5	Optimised wing twist distribution compared to baseline design, baseline in blue and optimised in orange . . . . .	51
6.6	Optimised propeller twist distribution compared to baseline design, baseline in blue and optimised in orange . . . . .	51
6.7	Optimisation results for the first 40 iterations . . . . .	51
7.1	Schematic of how an early version of the Rethorst correction was defined, this configuration resulted in asymmetric derivatives for the AD’d code . . . . .	54
7.2	Schematic of the version of the Rethorst model that has symmetric derivatives . . . . .	55
7.3	Full versus even correction factor solution, CFD data taken from Nederlof [25] . . . . .	56
7.4	Van Arnhem et al. engineering method for assessing propeller performance in a non-uniform inflow field [4] . . . . .	56
7.5	Slipstream contraction for different radial stations as compared to Veldhuis [10, 36, 39] . . . . .	57
7.6	Schematic of slipstream deflection [39] . . . . .	58

# List of Tables

2.1	Derivative verification results of HELIX . . . . .	11
3.1	Relevant system ESP and efficiency values, taken from de Vries' work[11] . . . . .	17
3.2	Derivative verification results for remeshing function . . . . .	18
4.1	The limitations of the Rethorst (l) with their respective solution (r) . . . . .	22
4.2	Forward AD derivative verification results . . . . .	36
4.3	Reverse AD derivative verification results . . . . .	37
5.1	Total derivative verification results . . . . .	44
6.1	Elements of optimisation design vector, $\vec{x}$ . . . . .	47
6.2	Propeller optimisation results . . . . .	48
6.3	Wing optimisation results . . . . .	49
6.4	Elements of optimisation design vector, $\vec{x}$ . . . . .	49
6.5	Results of the coupled wing-propeller optimisation . . . . .	50
6.6	Efficiency and accuracy assessment of model with direct method versus finite differences. The time was averaged over 4 runs on an Intel i7 12700k . . . . .	52

# Introduction

The rapidly increasing socioeconomic pressure for the aviation industry to decrease its emissions requires alternative propulsion methods. Propellers offer an attractive solution. Propellers are also used on Urban Air Vehicles (UAVs), often in Distributed Electric Propulsion (DEP) configurations. DEP configurations have been speculated to provide more efficient aircraft-propulsor combination [11]. DEP could be advantageous as it could increase generated lift per unit span of wing. Decreasing the wing area and weight. A weight reduction can cause a so called 'snowball effect'; the lift requirement decreases, thus the drag decreases and consequently the propulsor's thrust requirement decreases. The need for more efficient aircraft and large demand for UAVs [3] incentivises research into propeller-wing design optimisation.

Aircraft are, above anything else, a combination of complex and interacting systems. For this reason the aviation industry has been at the frontier of Multidisciplinary Design Optimisation (MDO) technology since MDO's birth in the 1970s. Many aerospace systems have experienced extensive optimisation procedures to perform at optimal conditions. However, literature on the optimisation of coupled propeller-wing configurations is limited, with only a few articles discussing its impact and possible benefits [8, 9, 19]. A study by van der Leer [20] discussed the potential benefit of including wing-tip propellers (in tractor configuration). Van der Leer showed that the potential benefit of wing-tip mounted propellers is dependent on the design variables of both the wing and propellers. With this conclusion, van der Leer demonstrates the importance of a wing-propeller optimisation study, since optimisation is the exploration of a design space. Van der Leer also showed the significance of including aerodynamics and structural models in the wing-propeller design process. Van der Leer's model does however contain discontinuities and can therefore not be used for optimisation purposes. Another study by Clarke [8] assessed multipoint wing-propeller optimisation using the multiple-fidelity framework SUAVE [21]. Clarke's method assessed the slipstream using a frozen vortex wake model, that inherently does not satisfy the slipstream boundary conditions. Moreover, the propeller blade twist is not considered in the optimisation process. Furthermore, Clarke did not focus on generating an efficient wing-propeller optimisation framework, and does not comment on the computational expense of his optimisation architecture. SUAVE is a gradient-free optimisation framework. Gradient free optimisation is often not feasible for larger problems and also does not guarantee an efficient optimisation procedure. Therefore, a computationally efficient propeller-wing optimisation framework would be valuable.

A plausible reason for the scarcity of wing-propeller optimisation studies is the lack of computationally efficient wing-propeller models that can readily be used for optimisation. Computationally efficient optimisation models not only consider the efficiency of aerostructural (or other relevant disciplines) models but also the derivative calculation methods and the optimisation algorithm. A frequently used optimisation algorithm, due to its outstanding performance for larger constrained systems, is SNOPT [13]. The derivative calculation method is arguably the most important factor in determining whether a model is computationally efficient. For gradient assessment, either a non-intrusive or intrusive method can be used. Non-intrusive methods are finite differences or the complex step method [22]. Intrusive methods include algorithmic differentiation, the direct method, and the adjoint method [23]. All methods vary in terms of accuracy and computational efficiency. In general, it can be said that either the direct or adjoint method yield the best results in terms of accuracy and computational cost. Using the direct or adjoint method is dependent on whether the number

of inputs is greater than the number outputs.

Optimisation, or MDO, is in itself a multi-faceted subject. Building an optimisation model requires satisfying several requirements. These requirements range from how derivatives are calculated to whether the model contains discontinuities. Satisfying these requirements is essential and must be considered when addressing the knowledge gap:

*To date, an extensive propeller-wing optimisation procedure is missing due to the lack of a computationally efficient aerostructural coupled propeller-wing model.*

## 1.1. Research Objective

To further address this issue, this thesis aims to present and discuss a computationally efficient framework for coupled propeller-wing optimisation. Furthermore, a modular architecture is considered to be a requirement, since a modular framework can easily be improved or extended in future iterations. Therefore, the research objective is stated as:

***To configure a modular wing-propeller framework suitable for computationally efficient aerostructural optimisation,***  
 which will be achieved by  
***Configuring a mid-fidelity propeller-wing model suitable for optimisation that uses efficient gradient calculation methods to yield faster and more accurate optimisation performance.***

This research objective can be achieved by answering the following research questions:

1. How accurate is the propeller-wing model?
  - (a) How accurately can the aerostructural solver predict wing aerodynamic performance?
  - (b) How accurately can the propeller model HELIX predict propeller performance?
  - (c) How accurately can the coupled wing, propeller and slipstream model predict coupled model performance?
2. What is the computational cost of simultaneous propeller-wing optimisation and how can this be reduced?
  - (a) How will the adjoint method change the optimisation procedure?
    - i. What is the factor by which optimisation speed can be increased by using an adjoint versus finite differences?
    - ii. How are the results affected by the use of an adjoint versus finite differences/complex step?
  - (b) How will the direct method change the optimisation procedure?
    - i. What is the factor by which optimisation speed can be increased by using the direct method versus finite differences?
    - ii. How are the results affected by the use of an adjoint versus finite differences/complex step?
3. How does simultaneous wing-propeller optimisation affect wing and propeller designs?
  - (a) How is the optimised wing design affected by the slipstream in the mid-fidelity model, compared to isolated optimisation?
  - (b) How is the optimised propeller design affected by the presence of a wing in the mid-fidelity model, compared to isolated optimisation?
  - (c) What wing design variable influences propeller design and performance most?
  - (d) What propeller design variable influences wing design and performance most?

## 1.2. Potential Solution

These research questions can be answered with a computationally efficient propeller-wing optimisation framework. Van der Leer [20] showed that aerodynamic and structural effects are important to consider when assessing the coupled wing-propeller system. Therefore, the model used to achieve the research objective consists of a wing aerostructural solver, using a VLM to model aerodynamics for the wing, and a BEM model

to evaluate propeller performance. The propeller and wing models will be coupled using a slipstream model, based on Rethorst's correction factor [31].

The propeller is modelled by a Hybrid BEM code called HELIX [26]. HELIX extends the applicability of classical BEM models by including analysis for off-axial propeller configurations. Furthermore, the HELIX code has a working adjoint, and can therefore be used for computationally efficient optimisation. HELIX requires a function that outputs the slipstream velocity distribution. Additionally, this velocity distribution function requires a derivative code.

The wing aerostructural model is OpenAeroStruct (OAS) [16]. OAS is a readily available aerostructural optimisation/analysis software that will be used to represent the wing aerodynamic and structural performance. OAS requires several modifications before it can be coupled to the slipstream model. The Rethorst method requires the slipstream to be aligned with the VLM panels. In other words, the propeller edges must align with the VLM panels. A remeshing function was therefore included in OAS. The remeshing function assesses what panels are closest to the propeller edges and adjusts their locations such that these align with the propeller. Furthermore, the Rethorst correction has to be coupled to the VLM calculations. Lastly, the velocity distribution must be communicated to the VLM system.

The slipstream model is based on the Rethorst correction factor. The Rethorst correction model assesses a wing crossing a slipstream of finite height. In other words, the correction model satisfies the slipstream boundary conditions. The correction factor has several requirements; the propeller must be in the centre of the wing, the propeller edges must align with the VLM mesh, the centre of the propeller and the centre of a VLM panel must align. The second requirement is satisfied by the first OAS modification discussed in the previous paragraph. The first and third requirements are satisfied by implementing the correction factor on an overset mesh. The overset mesh overlays the VLM wing, after which the correction factor is interpolated from the overset mesh to the VLM mesh. Other models, used for propeller-wing analysis, that implement the Rethorst method remesh the wing depending on the wing span, propeller location and radius. Such remeshing procedures can not be used for optimisation since these introduce discontinuities. The overset mesh provides a fully continuous system, and is thus suitable for optimisation.

In order to verify that the coupled model encompasses all relevant phenomena, its accuracy has to be assessed by comparing it to experimental data. Furthermore, the propeller, wing, and slipstream models contain derivative codes, with forward and reverse Algorithmic Differentiation (AD). Verifying these forward and reverse AD codes is essential. If the derivative codes are wrong, it could lead the optimiser into the wrong direction. Both the forward and reverse AD codes are verified for all models. With those, both the direct and adjoint method can be used. The adjoint and direct method are important since these guarantee the computational efficiency of the framework.

After the model has been configured and validated, the second step of the research project is to evaluate computational efficiency. The computational performance can be evaluated by comparing the time needed to carry out an optimisation using adjoints and other derivative finding methods. The adjoint method will be used to evaluate the system's derivatives. Comparing the coupled model using an adjoint to several other derivative finding methods will give insights into the accuracy and computational efficiency of the coupled model.

After the model has been configured and its computational efficiency evaluated, a comparison can be made by comparing the results of coupled optimisation for a test case. The comparison to isolated wing- and propeller-optimisation will give an answer to the question whether a coupled wing-propeller optimisation process yields different results and is or is not worth the added complexity.

### 1.3. Scope and Limitations

The research proposal's scope includes the necessary elements to be able to optimise a wing-propeller system, whilst remaining computationally efficient. The research scope inherently introduces several limitations. The scope, and therefore limitations, are discussed in this section.

The proposed propeller model is limited due to its inability to predict spanwise flow components. Therefore, sectional blade properties can not be optimised, since the propeller model will favour generating all lift on the most inboard section, which is not physical. Furthermore, the propeller model does not guarantee the structural integrity of the propeller. Lastly, the propeller validation showed an over-prediction of thrust at higher advance ratios, that should be considered when analysing optimisation results.

The wing aerodynamic model performs worse for higher angles of attack due to its inability to assess flow separation. Furthermore, wing thickness is not considered in the aerodynamic model. The wing structural model is a tubular spar model. The tubular spar model does not have the level of fidelity of a wingbox model. Furthermore, the structural model lacks a validation procedure.

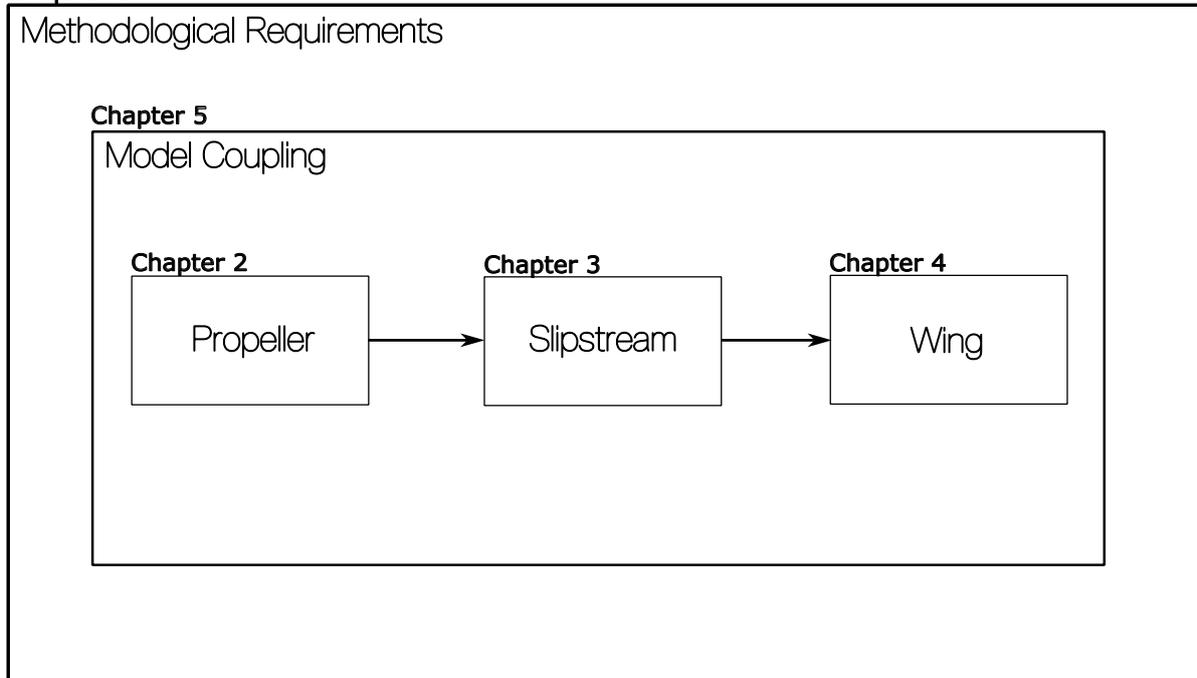
The slipstream model has several limitations. Firstly, the slipstream model can not model tangential flow components or their impact only accounting for axial effects on the wing. Although the slipstream model can satisfy the slipstream's boundary conditions, its current implementation neglect part of the solution. The slipstream model uses mirror vortices, from which half, the odd solution, are expensive to calculate. Calculating the odd solution substantially increases the computational cost. Since the research objective is to produce a computationally efficient wing-propeller optimisation framework, it was chosen to neglect the odd solution.

The scope of this framework will encompass building a computationally efficient framework that can optimise a propeller-wing system. Future iterations of the framework can easily increase the scope of this work as the proposed framework will be modular.

## 1.4. Research Outline

An overview of the thesis structure is given in [Figure 1.1](#). The remainder of this section elaborates on the structure of this report.

### Chapter 6



**Figure 1.1: Structure of the thesis report**

[Chapter 2](#) describes the propeller model. The optimisation architecture uses the propeller model HELIX [26]. HELIX uses a Hybrid Blade Element Method (HBEM) that is especially well-suited for UAV rotorcraft as it can analyse propellers in hover and forward flight and off-axial configurations. The code required little modifications and includes a working adjoint, thus can be readily used in the propeller-wing framework.

[Chapter 3](#) details the aerostructural wing model. The wing's aerostructural properties are assessed using OpenAeroStruct (OAS) [16]. OpenAeroStruct combines an aerodynamic model, a VLM, with a Finite Element Method (FEM) model to evaluate structural properties. OAS required a number of modifications to make it suitable for wing-propeller optimisation. The first modification to OAS was including an induced velocity profile on the wing to model the slipstream's flow field since the wing is not experiencing a homogeneous velocity field. The second adaptation encompassed adding the Rethorst correction factor to the

Aerodynamic Influence Coefficient (AIC) matrix. Other subsystems, such as a wing to propeller interaction code, are also discussed in [Chapter 3](#).

As recognised by several authors [31, 37], a Vortex Lattice Method (VLM) submerged in a slipstream over-predicts lift if only the freestream velocity vector is changed. This over-prediction can be attributed to the lack of a correction factor that satisfies the slipstream boundary conditions. Rethorst [31] addressed this issue by deriving a correction factor that is added to the AIC. The correction factor modifies the circulation of each VLM panel and satisfies the slipstream's boundary conditions. The slipstream model is implemented with an overset mesh. The overset mesh was required for optimisation since the system has to be fully continuous. A discontinuous system introduces discontinuous derivatives, which is undesirable for optimisation. The overset mesh introduces several challenges that are discussed in [Chapter 4](#).

The propeller, wing, and slipstream code were all equipped with a forward and reverse algorithmic differentiated (AD) code, which are necessary for the direct and adjoint method, respectively. The direct and adjoint method are intrusive methods used to efficiently obtain the system's derivatives. Albeit efficient and accurate, intrusive gradient calculation methods require thorough verification. Additionally, intrusive gradient assessment methods could require modifications to prevent discontinuous derivatives, as was necessary for the wing and slipstream models.

[Chapter 5](#) discusses model coupling. The optimisation framework OpenMDAO [14] was used to couple the propeller, slipstream, and wing models. OpenMDAO is an open-source optimisation framework that is particularly well-suited for gradient based optimisation. The OpenMDAO architecture provides the opportunity to analytically or numerically calculate derivatives, which are essential for efficient and accurate optimisation.

The wing-propeller system can be optimised after the models have been coupled. [Chapter 6](#) describes isolated wing and propeller optimisation as well as coupled optimisation. The coupled system's design variables are mirrored from the left to the right side of the wing. [Chapter 6](#) also assesses the accuracy and computational efficiency of the framework.

[Chapter 7](#) discusses methodological considerations and results as well as recommendations before a conclusion is given in [Chapter 8](#). The methodological results are important since in the design of the wing-propeller framework several requirements were discovered that affected system design choices. Furthermore, optimisation inherently introduces challenges that are not relevant for analysis purposes. These findings are summarised in [Chapter 7](#) as these will be relevant in the future development of this framework.

# 2

## Propeller Model

The propeller model is expected to accurately and efficiently assess propeller performance. The MDOLab has developed a mid-fidelity Blade Element Momentum (BEM) model suitable for efficient aeroacoustic optimisation by using the adjoint method to assess the system's derivatives. This BEM model, called HELIX [26], was written in Fortran90 and was differentiated using the open-source differentiating tool Tapenade.

This chapter aims to provide a description of BEM, followed by description of the propeller code HELIX. Both model validation and derivative verification are detailed after the model theory. The derivatives are verified by comparing the derivatives returned by the algorithmic differentiated code to the ones obtained with complex step differences, that approach machine accuracy [22].

### 2.1. The Propeller Model

The propeller model is one of the three (wing, propeller and slipstream) models used in this research project. The propeller model used is a Hybrid BEM (HBEM) method called HELIX. This section elaborates on BEM and HELIX. Afterwards a validation process is shown in which experimental TU Delft PROWIM [34] is compared to numerical data returned by HELIX.

#### 2.1.1. Blade Element Momentum Theory

Blade Element Momentum theory (BEM) combines momentum theory with blade element theory to determine axial and tangential induced velocities. BEM assumes that each blade element is responsible for the change of momentum of air that passes through its respective annulus. In other words, there is no radial interaction between blade elements. BEM integrates forces over the propeller radius to predict thrust, torque and power. A visual representation of BEM is given by Figure 2.1. Figure 2.2 gives a schematic of momentum theory. The division of the propeller disk into annuli is shown on the left of Figure 2.1, where the right side of the figure shows a blade section with angle of attack,  $\alpha$ , pitch  $\beta$ , inflow angle  $\phi$ , and the rotational- and axial-velocity.

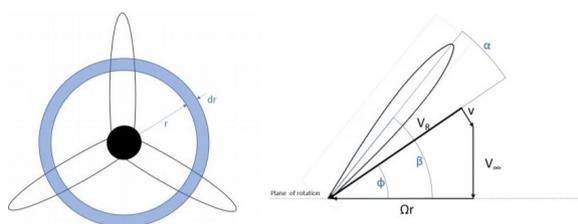


Figure 2.1: Schematic of Blade Element Theory, where each blade section generates lift [20]

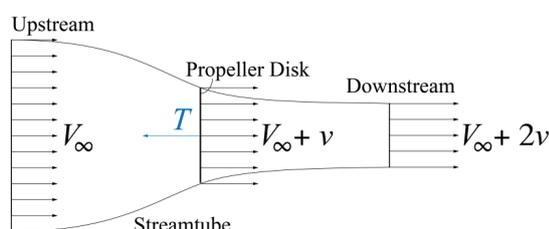


Figure 2.2: Schematic of momentum theory, where the full annulus produces a momentum increase [26]

The axial- and tangential velocities are given by Eq. (2.1) and Eq. (2.2). The induction factors  $a$  and  $a'$  represent axial and tangential induction, respectively.

$$\begin{aligned} V_a &= V_\infty(1 + a) \\ &= V_\infty + V_a \end{aligned} \quad (2.1) \quad \begin{aligned} V_t &= \Omega r(1 - a') \\ &= \Omega r - v_t \end{aligned} \quad (2.2)$$

The thrust and torque are derived from the lift and drag forces using Eq. (2.3) and Eq. (2.4), using the nomenclature defined in Figure 2.1. The airfoil lift and drag are defined by Eq. (2.5) and Eq. (2.6), with  $V_r$  - the radial velocity - defined as Eq. (2.7).

$$\delta T = \delta L \cos \phi - \delta D \sin \phi \quad (2.3) \quad \frac{\delta Q}{r} = \delta L \sin \phi + \delta D \cos \phi \quad (2.4)$$

$$l = 0.5C_l\rho V_r^2 c \quad (2.5) \quad d = 0.5C_d\rho V_r^2 c \quad (2.6) \quad V_r = \frac{V_\infty(1 + a)}{\sin \phi} \quad (2.7)$$

Combining Eq. (2.3) - Eq. (2.4) with Equations Eq. (2.5) - Eq. (2.6) gives rise to the Equations for partial thrust and torque (per radii): Equations Eq. (2.8) and Eq. (2.9), with  $B$  as the number of blades.

$$\frac{\delta T}{\delta r} = 0.5\rho V_r^2(1 + a)^2 Bc \frac{C_l \cos \phi - C_d \sin \phi}{\sin \phi^2} \quad (2.8) \quad \frac{\delta Q}{\delta r} = 0.5\rho V_r^2(1 + a)^2 Brc \frac{C_l \cos \phi + C_d \sin \phi}{\sin \phi^2} \quad (2.9)$$

A BEM model is an iterative solver for the axial and azimuthal induction factors. Therefore, a second set of Equations is required that describe the azimuthal and axial induction factors. This second set of Equations forms the bridge between *Blade Element Theory* and *Momentum Theory: Blade Element Momentum Theory*.

Momentum theory gives Eq. (2.10) and Eq. (2.11). By equating Eq. (2.8) to Eq. (2.10) and Eq. (2.9) to Eq. (2.11), a third set of Equations is obtained, given by Eq. (2.12) and Eq. (2.13), further described by Eq. (2.14) through Eq. (2.16).

$$\delta T = 2\rho V_\infty^2 a(1 - a)\delta A_d \quad (2.10) \quad \delta Q = \rho\delta A_d V_\infty(1 - a)2\Omega a' r^2 \quad (2.11)$$

$$a = \frac{1}{\frac{4 \sin \phi^2}{\sigma C_n} + 1} \quad (2.12) \quad a' = \frac{1}{\frac{4 \sin \phi \cos \phi}{\sigma C_t} - 1} \quad (2.13)$$

$$\sigma = \frac{Bc}{2\pi r} \quad (2.14) \quad C_n = C_l \cos \phi + C_d \sin \phi \quad (2.15) \quad C_t = C_l \sin \phi - C_d \cos \phi \quad (2.16)$$

So far, the BEM theory modelled the propeller as an actuator disk with an infinite number of blades. However, due to the number of blades being finite, a wing tip correction must be applied. For that purpose, the Prandtl tip loss factor can be implemented. The Prandtl tip loss factor is mathematically represented by Eq. (2.17) - Eq. (2.19) and should be applied to the tip and hub of the propeller blade.

$$f_{hub}(\mu) = \frac{2}{\pi} \arccos \exp \left( \frac{B}{2} \frac{\mu_{root} - \mu}{\mu} \sqrt{1 + \frac{\Lambda^2 \mu^2}{(1 - a)^2}} \right) \quad (2.17)$$

$$f_{tip}(\mu) = \frac{2}{\pi} \arccos \exp \left( -\frac{B}{2} \frac{\mu_{tip} - \mu}{\mu} \sqrt{1 + \frac{\Lambda^2 \mu^2}{(1 - a)^2}} \right) \quad (2.18)$$

$$f_{total} = f_{tip} \cdot f_{hub} \quad (2.19)$$

### 2.1.2. HELIX

HELIX is a hybrid BEM (HBEM) model that will be used due to its outstanding optimisation performance [26]. HELIX includes forward and reverse algorithmic differentiated code, and can thus optimise a propeller with the adjoint and direct method. Furthermore, future iterations of the wing-propeller framework will include acoustic optimisation. This acoustic model can easily be implemented since it has been designed for HELIX.

HBEM converges when the thrust coefficient returned by momentum theory and blade element theory is equal. Mathematically it means that the model converges when Eq. (2.20) equals zero, for a given thrust  $T$ .

$$\mathcal{R}(T) = C_{T,BET}(T) - C_{T,MT}(T) \quad (2.20)$$

The question remains; how is  $C_T$  calculated with momentum or blade element theory? Momentum theory calculates the thrust coefficient by momentum theory calculates the thrust coefficient with Eq. (2.21), with the density  $\rho$ , inflow velocity  $V$  and radius  $R$ .

$$C_{T,MT} = \frac{T}{0.5\rho V^2 \cdot \pi R^2} \quad (2.21)$$

Blade element momentum theory includes blade geometry and orientation in the thrust coefficient assessment. Davoudi's HBEM model, used in HELIX, uses a linear inflow model,

$$\lambda = \lambda_0(1 + k_x r \cos \psi + k_y r \sin \psi), \quad (2.22)$$

with  $\lambda_0$  as the uniform inflow ratio (can be obtained through momentum theory),  $k_x$ ,  $k_y$  as model specific parameters and  $\psi$  as the azimuthal angle of the rotor. The radial ( $u_r$ ), tangential ( $u_t$ ) and perpendicular ( $u_p$ ) velocity components are calculated using the inflow ratio  $\psi$  and Eq. (2.23) through Eq. (2.25), in which  $\mu$  is the normalised radial velocity.

$$u_r = \mu \cos(\psi); \quad (2.23) \quad u_t = r + \mu \sin(\psi); \quad (2.24) \quad u_p = \lambda \quad (2.25)$$

An initial lift estimation of each blade section can be calculated by using the effective angle of attack of each blade section,  $\alpha_{\text{eff}}$ , and the lift-curve-slope  $C_{l_\alpha}$ . The equation for the lift coefficient is given below in Eq. (2.26).

$$C_l = C_{l_\alpha} \alpha_{\text{eff}}, \quad (2.26)$$

with  $\alpha_{\text{eff}} = \theta - \alpha_{l=0} - \phi$  with  $\theta$  as the blade pitch,  $\alpha_{l=0}$  the zero-lift angle of attack and  $\phi$  the angle of attack of the incoming velocity ( $\phi = \tan(u_p/u_t)$ ). HELIX uses an inflow model that accounts for airfoil stall and transition. The input variables for the sectional airfoils are lift slope, stall angle, zero angle of attack lift coefficient and transition parameter  $M$  that determines how fast the lift curve decreases after stall occurs. By using Beard's stall model [5], the lift coefficient can be computed at every angle of attack. The sectional airfoil characteristics can also be included as design variables and be optimised varying for each propeller blade section. The stall model significantly improves the fidelity of the propeller since it prevents the optimizer from exploiting unphysical angles of attack [5]. The lift coefficient equation including a stall model is given below:

$$C_l = (1 - \sigma)C_{l_\alpha} \alpha_{\text{eff}} + \sigma[2 \sin(\alpha_{\text{eff}} - \alpha_{L=0}) \sin^2(\alpha_{\text{eff}} - \alpha_{L=0}) \cos(\alpha_{\text{eff}} - \alpha_{L=0})], \quad (2.27)$$

with

$$\sigma = \frac{1 + e^{-M(\alpha_{\text{eff}} - \alpha_{L=0} - \alpha_0)} + e^{M(\alpha_{\text{eff}} - \alpha_{L=0} + \alpha_0)}}{[1 + e^{-M(\alpha_{\text{eff}} - \alpha_{L=0} - \alpha_0)}][1 + e^{-M(\alpha_{\text{eff}} - \alpha_{L=0} + \alpha_0)}]}. \quad (2.28)$$

The coefficient of thrust for blade element theory can be computed as:

$$C_{T,BET} = \frac{N_b}{4R^2\pi^2} \int_0^R \int_0^{2\pi} C_l u^2 c(r) \cos(\phi) d\psi dr, \quad (2.29)$$

where  $N_b$  is the number of blades on the rotor,  $c(r)$  the chord length of the blade as a function of the rotor radius and  $u$  is the magnitude of the velocity at the blade section. This application of the blade element

theory calculates the thrust at each position, and thus requires a full rotor evolution to assess the propeller thrust. The system converges once the residual, given by Eq. (2.20), equals zero. Robust convergence is guaranteed by using a line search method to identify a region of the system that contains a solution. Afterwards Brent's method is used to solve the root finding problem that the residual introduces. HELIX is written for adjoint-based optimization and coupling with aeroacoustic analysis tools. The HELIX code contains a forward and reverse algorithmic differentiated code, necessary for the direct or adjoint method [23]. The possibility of extending the model with an aeroacoustic analysis code is of significance since propeller acoustic optimization is becoming increasingly important [3].

### 2.1.3. Velocity Distribution

HELIX outputs a velocity distribution to the slipstream model (discussed in Chapter 4. The Bernoulli equation, given in Eq. (2.30), is used to convert propeller thrust to velocity

$$p_s + \frac{\rho V^2}{2} = p_t, \quad (2.30)$$

with  $p_t$  and  $p_s$  as static and total pressure, respectively. It was assumed that the thrust per blade element was distributed over the total annulus. It was also assumed that this distributed thrust was fully converted into dynamic pressure. These assumptions, together with Bernoulli's equation result in Equation Eq. (2.31):

$$\frac{T}{A_{\text{annulus}}} = \frac{\rho V^2}{2}, \quad (2.31)$$

where  $T$  is the thrust produced by the annulus and  $A_{\text{annulus}}$  the area of the blade annulus. This equation is applied over the blade radius for each annulus and returns an array with velocity distribution. HELIX also contains the forward and reverse AD'd code for this procedure. Forward and reverse AD is elaborated on in Section 4.3.

### 2.1.4. Limitations of a BEM model

The disadvantages of a BEM model are:

- Ineptitude to predict wake expansion
- Each annuli is independent of the other, i.e. no interaction in spanwise direction, thus neglecting effects such as radial flow
- Does not include tip losses
- Does not account for yaw angles
- Does not account for unsteady effects
- Does not include a correction for heavily loaded rotors
- Isn't guaranteed to converge

Some of these limitations encountered in BEM models can be overcome by using additional models. For instance, the inability to calculate tip losses can be solved by using the Prandtl Tip Loss correction model, yaw can be included by introducing a third velocity component and wake expansion can be modelled using a separate slipstream model.

Arguably the largest disadvantageous of BEM is its inability to consider spanwise flow velocity. Spanwise flow causes the Himmelskamp effect. A rotating aerodynamic body tends to stabilise the boundary layer as the centrifugal force pushes the flow in the spanwise direction. The spanwise flow increases the achievable angles of attack by delaying stall. Consequently, higher lift coefficient values are feasible along the blade. Furthermore, the blades also experience a Coriolis force [12]. The Coriolis effect pushes separated flow towards the trailing edge and, similar to the centrifugal, promoting attached flow.

BEM limitations that are important for optimisation purposes are mostly related to how the propeller is discretised. HELIX includes radius and chord variables for each spanwise section. However, the optimiser chooses to maximise the chord and radius variables near the root since no spanwise flow interaction is included in the BEM model. This causes the optimiser to move to non-feasible designs since the structural integrity of the blade can not be guaranteed this way. An example of a chord distribution returned by an optimisation process that includes sectional chord values as design variables is given in Figure 2.3.

Figure 2.3 shows an unfeasible propeller chord distribution. Moving the full thrust load towards the hub of the propeller has consequences that are not properly reflected by the BEM model. The only way to account

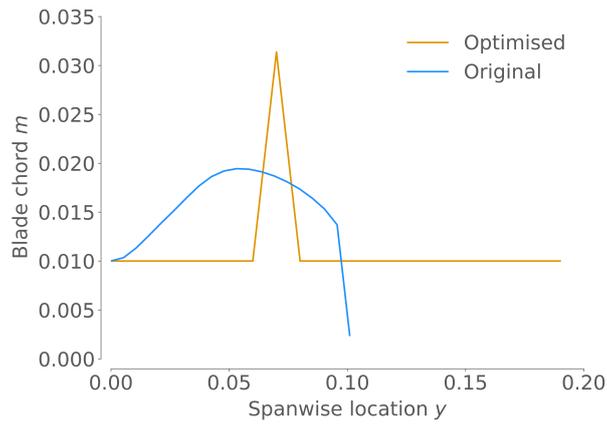


Figure 2.3: Chord distribution returned by an optimisation process with sectional chord design variables

for this, without imposing additional constraints, is by using a propeller model that includes spanwise flow effects.

## 2.2. Model Validation

For the propeller code validation, experimental data from the TU Delft PROWIM propeller, shown in Figure 2.4, was used. Figure 2.5 shows the PROWIM chord and twist distributions. By varying the advance ratio in the numerical model a range of thrust coefficients was returned, shown in Figure 2.6. The propeller model does a relatively good job at predicting the low advance ratio thrust coefficient but shows more significant disparity at the high advance ratio regime. The discrepancy at higher advance ratios is likely related to the airfoil stall models that were used since the angle of attack on the propeller blade sections increases at higher advance ratios.



Figure 2.4: PROWIM propeller<sup>1</sup>

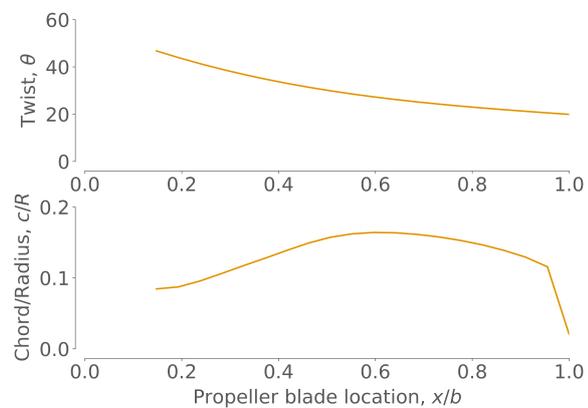


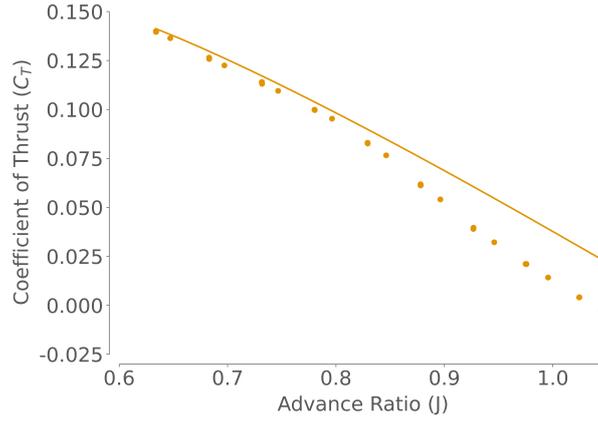
Figure 2.5: PROWIM twist distribution [34]

The propeller model can not be considered to be highly accurate at higher advance ratios. If the purpose of this study was to perform an in depth aerodynamic analysis of a wing-propeller system then the propeller model wouldn't have been sufficient. However, this study focuses on building an architecture for an aerostructural wing-propeller system. Therefore, the disparity at higher advance ratios should be considered when analysing results but can be assumed to not drastically impact the optimisation process.

## 2.3. Derivative Verification

Using the adjoint method requires the model to provide model partial derivatives. These model derivatives can efficiently and accurately be obtained using a derived version of the code. However, deriving the model

<sup>1</sup>[bit.ly/3WjSnFz](https://bit.ly/3WjSnFz), Accessed on the 22<sup>nd</sup> of December 2022



**Figure 2.6: Validation results for HELIX compared to experimental results from the PROWIM propeller**

is a tedious process that could introduce errors. It is for this reason that derivative verification is a crucial step. If the derived model contains errors it could lead the optimiser into the wrong direction, since the optimiser uses gradients to determine what direction to move in.

Derivative verification is performed by comparing finite differences to complex step differences, to eventually evaluate the correctness of forward and reverse Algorithmic Differentiation (AD). [Table 2.1](#) gives a comparison between finite and complex step differences and forward AD. Reverse AD is verified by assessing whether the dot products of the forward and reverse input variables is the same as the dot product of the forward and reverse output variables. Reverse AD is not included in the table since the residual is zero for all in- and output combinations. The derivative verification procedure is elaborated on in [Section 4.3](#).

**Table 2.1: Derivative verification results of HELIX**

Design Variable	Finite Differences	Complex Step	Direct/Forward AD
Twist <sub>root</sub>	+0005.25592 <b>017173767</b>	+0005.25592188734690	+0005.255921887346 <b>34</b>
Twist <sub>mid</sub>	+0035.6431 <b>0574531555</b>	+0035.64311787701828	+0035.64311787701 <b>954</b>
Twist <sub>tip</sub>	+0052.4891 <b>8604850769</b>	+0052.48919162513608	+0052.48919162513 <b>997</b>
Chord <sub>root</sub>	+1089.456 <b>42447471618</b>	+1089.45672395754854	+1089.4567239575 <b>8970</b>
Chord <sub>mid</sub>	+5586.0 <b>3968286514282</b>	+5586.04638024675023	+5586.046380246 <b>94668</b>
Chord <sub>tip</sub>	+3627.14 <b>0921831130981</b>	+3627.14509970930521	+3627.145099709 <b>40662</b>
$\omega$	+0037.5209391 <b>100547790</b>	+0037.52093083191834	+0037.52093083191 <b>935</b>

# 3

## Wing Model

The aerostructural properties of the wing are assessed using the aerostructural solver OpenAeroStruct (OAS) [16]. OAS was developed at the University of Michigan's MDO Lab to provide a lightweight, low-fidelity aerostructural solver that makes use of the OpenMDAO [14] framework and symbolic derivatives and adjoint and direct method. Symbolically differentiating the aerostructural solver provides the opportunity to accurately and efficiently optimise the aerostructural characteristics of the wing.

Coupling the slipstream model and wing model requires some modifications to the existing OAS code, as described in Section 3.2. The modifications introduce new derivatives that require a verification procedure, which is described in Section 3.3. The verification requires verifying the partial derivatives returned by the aerostructural solver. Wing validation is included in Chapter 5.2 as the 'prop-off' case. The wing structural validation is missing as it was also not included in the original OpenAeroStruct paper. OpenAeroStruct does have the option to use a wingbox structural model which is more realistic [7]. Future iterations of the framework should include the wingbox model as is discussed in Section 7.2.8.

### 3.1. AeroStructural Model

This section describes the aerostructural solver OpenAeroStruct [16]. OAS encompasses a Vortex Lattice Method (VLM) to assess the wing's aerodynamic performance, whilst a Finite Element Method (FEM) guarantees structural integrity. However, before the aerostructural can be used, a number of modifications are necessary, that are discussed in Section 3.2.

#### 3.1.1. Aerodynamic Wing Model

The wing's aerodynamics are represented by a VLM [2] model. A VLM divides the surface into quadrilateral panels. Afterwards, each panel is assigned a horseshoe vortex running over its boundaries. Each vortex consists of a bound vortex in spanwise direction and two trailing vortices. The trailing vortices extend into the freestream direction. Each panel gets assigned a control point, located at 75% panel chord. A single horseshoe vortex, with control point, is visualised in Figure 3.1.

This control point becomes important when examining the influence of the horseshoe vortex. The horseshoe vortex will induce a velocity on the surrounding space. The Biot-Savart law relates the velocity at an arbitrary point  $P$  to the vortex filament. The Biot-Savart law is given in Eq. (3.1) where  $\Gamma$  represents the vortex strength,  $d\vec{l}$  as the vortex filament segment and  $\vec{r}$  as the distance from the vortex to the point of interest. Equation Eq. (3.2) is obtained after integrating the Biot-Savart law over a semi-infinite straight line, with  $h$  as the distance from point  $P$  to the start point of the vortex filament.

$$d\vec{V}_\Gamma = \frac{\Gamma}{4\pi} \frac{d\vec{l} \times \vec{r}}{|\vec{r}|^3} \quad (3.1) \quad V = \frac{\Gamma}{4\pi h} \quad (3.2)$$

Helmholtz's theorem [2] states that:

1. The strength of a vortex filament is constant along its length.

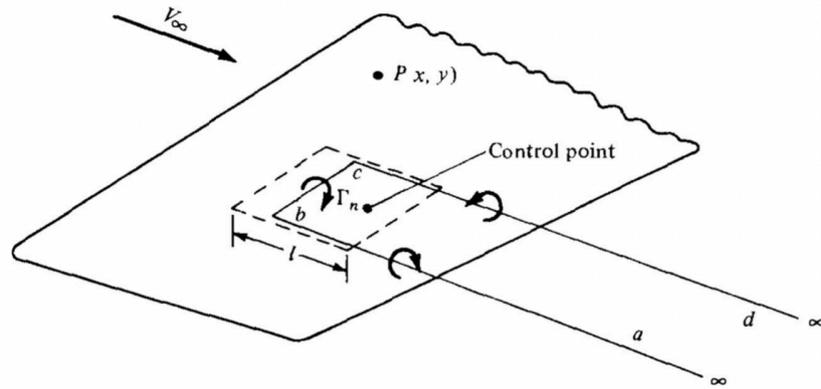


Figure 3.1: A single horseshoe vortex on a lifting surface with a control point

2. A vortex filament cannot end in a fluid, thus it must extend to the boundaries of the fluid or form a closed path.

Following Helmholtz's theorem, the horseshoe vortex's strength is constant over its finite and semi-infinite parts. A lifting surface is modelled by superimposing multiple horseshoe vortices in spanwise direction. The Weissinger lifting line theory model [38] has one horseshoe vortex in chordwise direction, and multiple in spanwise direction. The VLM model is an extension of Weissinger's lifting line model by including multiple horseshoe vortices in chord- and spanwise direction.

In Figure 3.1, the panel borders are sketched with the dashed line, and the horseshoe vortex with solid lines. The imposed boundary condition (BC) guarantees flow tangency to the wing at the control point. In other words, the normal velocity at the control point has to be zero. This BC has to be satisfied at every control point. Combining the circulation induced velocity, from all panels in the model space, and the freestream velocity at the control point should result in a normal velocity equal to zero. Mathematically, this is represented by Eq. (3.3):

$$\mathbf{A}\Gamma = -\mathbf{V}_\infty \cdot \mathbf{n}, \quad (3.3)$$

where  $\mathbf{A}$  is the Aerodynamic Influence Coefficient (AIC) matrix,  $\mathbf{V}_\infty$  the freestream velocity and  $\mathbf{n}$  the normal vector of the panel. Do note that the normal vector  $\mathbf{n}$  is dependent on the camber line. Solving the linear system gives the circulation strengths of the horseshoe vortices. The Kutta–Joukowski theorem gives circulation per unit span that can be translated to lift using the Kutta Joukowski theorem, given in Eq. (3.4).

$$L' = \rho_\infty V_\infty \Gamma. \quad (3.4)$$

The sectional panel forces can be translated into the wing's lift and drag forces. The lift and drag forces correspond to the force vector in the upward and freestream directions, respectively.

#### Viscous Drag Model

OpenAeroStruct uses Raymer's [30] skin friction estimates for a flat plate. Skin friction is estimated based on the airfoil thickness-to-chord ratio, the Reynolds number and other flow properties such as density. A flow factor accounts for pressure drag due to flow separation. Raymer's method is a semi-empirical model and is considered to be valid up to drag-divergence Mach numbers.

### 3.1.2. Structural Wing Model

The structural model in OAS uses a finite element method (FEM). OAS uses a FEM approach that uses spatial beam elements, thus having six degrees of freedom (DOF) per node. The spatial beam element is a combination of truss, beam and torsion elements. Therefore the spatial beam element carries axial, bending and torsional loads all at the same time. Since each beam consists of two nodes, each beam has 12 DOFs in

total: each node can translate and rotate in the  $x$ -,  $y$ -, and  $z$ -direction. These translations and rotations are shown in Figure 3.2. The axial DOFs are  $u_1$  and  $u_2$ , the  $z$ -plane bending are  $v_1$ ,  $v_2$ ,  $\alpha_{z1}$  and  $\alpha_{z2}$ , the  $y$ -plane bending are  $w_1$ ,  $w_2$ ,  $\alpha_{y1}$  and  $\alpha_{y2}$ , and the torsion DOFs are  $\alpha_{x1}$  and  $\alpha_{x2}$ .

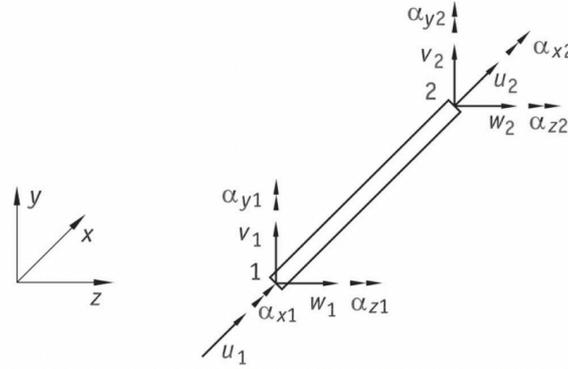


Figure 3.2: The 6 DOF are shown at each node for a beam element [27]

The stiffness matrix for a single element is given by Eq. (3.5)

$$[k]_e = \begin{pmatrix} k_1 & 0 & 0 & 0 & 0 & 0 & -k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12k_2^z & 0 & 0 & 0 & 6k_2^z l & 0 & -12k_2^z & 0 & 0 & 0 & 6k_2^z l \\ 0 & 0 & 12k_2^z & 0 & -6k_2^z l & 0 & 0 & -12k_2^z & 0 & 6k_2^z l & 0 & 0 \\ 0 & 0 & 0 & k_3 & 0 & 0 & 0 & 0 & 0 & -k_3 & 0 & 0 \\ 0 & 0 & -6k_2^y l & 0 & 4k_2^y l^2 & 0 & 0 & 0 & 6k_2^y l & 0 & 2k_2^y l^2 & 0 \\ 0 & 6k_2^z l & 0 & 0 & 0 & 4k_2^z l^2 & 0 & -6k_2^z & 0 & 0 & 0 & 2k_2^z l^2 \\ -k_1 & 0 & 0 & 0 & 0 & 0 & k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12k_2^z & 0 & 0 & 0 & -6k_2^z l & 0 & 12k_2^z & 0 & 0 & 0 & -6k_2^z l \\ 0 & 0 & -12k_2^z & 0 & 6k_2^z l & 0 & 0 & -12k_2^z & 0 & -6k_2^z l & 0 & 0 \\ 0 & 0 & 0 & -k_3 & 0 & 0 & 0 & 0 & 0 & k_3 & 0 & 0 \\ 0 & 0 & 6k_2^y l & 0 & -4k_2^y l^2 & 0 & 0 & 0 & -6k_2^y l & 0 & -2k_2^y l^2 & 0 \\ 0 & -6k_2^z l & 0 & 0 & 0 & -4k_2^z l^2 & 0 & 6k_2^z & 0 & 0 & 0 & -2k_2^z l^2 \end{pmatrix}, \quad (3.5)$$

with

$$k_1 = \frac{EA}{L} \quad (3.6) \quad k_2^z = \frac{EI_z}{L^3} \quad (3.7) \quad k_2^y = \frac{EI_y}{L^3} \quad (3.8) \quad k_3 = \frac{GJ}{L} \quad (3.9)$$

and  $E$  is the Young's modulus,  $A$  is the beam's cross-sectional area,  $L$  is the beam length,  $G$  the shear modulus,  $J$  the polar moment of inertia and the  $I$ s are the second moments of area about the three local coordinate directions. With the stiffness matrix, the linear system  $\mathbf{Ku} = \mathbf{f}$  can be solved, where  $\mathbf{K}$  is the global stiffness matrix as given in Equation Eq. (3.5),  $\mathbf{u}$  as the vector of displacements and rotations at the nodes, and  $\mathbf{f}$  as the forces and moments acting at the nodes.

The tube model is used because the wingbox model in OAS only works for a wing modelled with the symmetry condition, which is not a requirement but a modelling choice [6]. This symmetry condition can not be used however since the correction values imposed on the left half of the wing differ from those on the right side of the wing. Therefore, a tube model is used since this does work with a non-symmetry assumption for the wing. The wingbox can be included by redefining the OAS wingbox function.

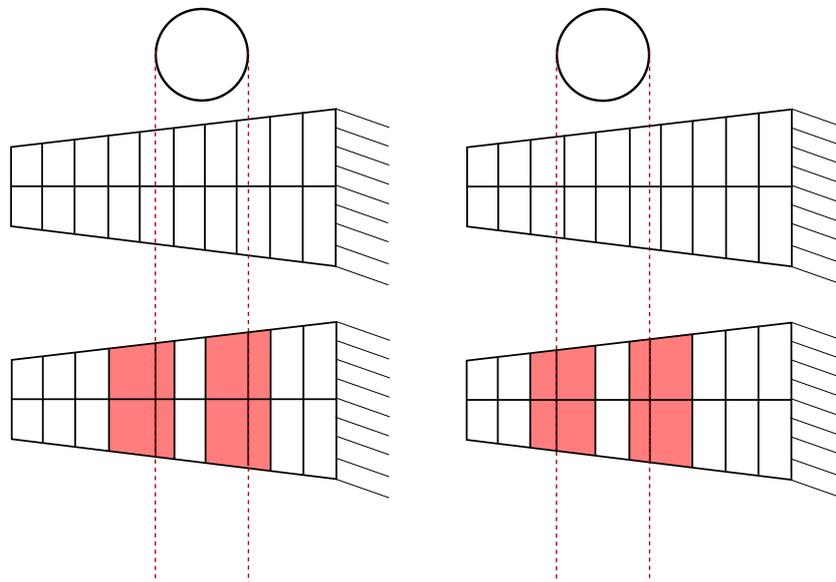
#### Failure Constraint in OAS

Although the isolated principal stresses might not exceed a material's yield stress, the combination of different stress modes might still cause yielding. Richard von Mises proposed a formula for combining the the principal stresses into an equivalent stress value. This equivalent 'Von Mises' stress is compared to the yield stress of the material. If the equivalent stress is larger than the failure condition of the material, the material will fail. This equivalent stress is otherwise known as the Von Mises stress.



To meet the prerequisites posed by the Rethorst model and maintain a fully continuous design space, the VLM mesh requires alterations. However, it is important to realise that one can not simply move, add or remove discretisation points during an optimisation process since this leads to discontinuous derivatives and will thus cause the optimiser to fail or return erroneous results. The goal in the OAS-modifications therefore is to return an aerodynamically sound model with the smallest number of changes. An idea was to, if the slipstream does not align with the VLM mesh, calculate the velocity of the panel based on how far the slipstream is in the VLM panel. However, using such an averaging function would essentially increase the propeller span, thus making it a discrete variable. Therefore, it was chosen to include a remeshing function discussed in this section.

The mesh in OAS was changed by an additional subroutine that checks whether the mesh is aligned, and if it is not it will move a panel's edge. This process is repeated for the other edge of the slipstream, and eventually changes the size of 4 VLM panels if 1 propeller is configured on the wing, this method is dubbed the 'moving nodes' method. The moving nodes method scales linearly—and thus poorly—with an increase in propellers. The designer should take the poor scaling into account when considering the mesh refinement. If more propellers are added to a coarse mesh, the remeshing function will have a more severe impact. A visualisation of the 'moving nodes' method is given in [Figure 3.4](#), in which two wing-propeller configurations are shown and the affected panels.



**Figure 3.4: Visualisation of the moving nodes method in which the panels closest to the jet are altered**

The remeshing function requires a model derivative code as well. The differentiated code has to be verified. If the derivatives of a single subsystem are incorrect it could lead the optimizer to wrong results. Important to notice is that the derivatives are discontinuous. Discontinuities occur when a slipstream edge passes over a control point, thus halfway over the panel in spanwise direction. A discontinuity occurs when the propeller moves as shown in [Figure 3.4](#). The highlighted panels in [Figure 3.4](#) are the affected panels. As the propeller moves to the left, the affected panels suddenly change, thus introducing discontinuities in the system.

Wingtip propellers, or when a slipstream edge is outside of the wingspan experience a similar remeshing procedure but only for a single panel. Bounds should be set such that the propeller can not move past the wingtip. Furthermore, an important safety feature was included such that when the slipstream edge is 'in' the last panel, it never moves that last node. This is important since the propeller could otherwise change the wingspan.

### 3.2.2. Propeller Weight Sizing

Propeller weight sizing includes the sizing of the powertrain—i.e. gears, shafts and engine—and fuel carrier weight, with the latter being batteries for this specific optimisation study. Fortunately, an electrical power

system can perform reasonably well without gears and thus solely the batteries, Power Management and Distribution (PMAD), electrical motor (EM) and propeller efficiency must be accounted for. The fully electrical powertrain architecture is visualised in Figure 3.5, as taken from the work of de Vries[11]

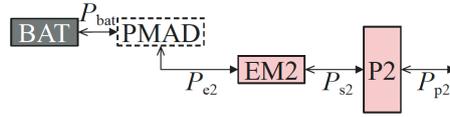


Figure 3.5: The used fully electric powertrain configuration, as given by de Vries[11]

For a first order battery mass estimation can be based on a variation of the Breguet range equation, given in Eq. (3.10), in which  $e_{bat}$  is the battery specific energy,  $W_{OE}$  is the operating empty weight,  $W_{PL}$  is the payload weight and  $E_{0,tot}$  is the required electrical energy.

The battery will be assumed to be 100% accurate and internal resistance to be zero. To preserve battery life the maximum state-of-charge is set to 20%. These assumption result into an equivalent specific power (ESP) of  $1500 \text{ Wh/kg}$  —equalling an ESP of  $6 \text{ kW/kg}$ —which is based on attainable future state of the art technology, as stated by de Vries [11].

$$R = \mu_{EM}\mu_p \left( \frac{L}{D} \right) \frac{E_{0,tot}}{W_{OE} + W_{PL} + \frac{g}{e_{bat}} E_{0,tot}} \quad (3.10)$$

The EM and power converter ESP are estimated to be  $5.92 \text{ kW/kg}$ , as given by de Vries [11]. The ESP in de Vries' work accounts for thermal management and PMAD systems and the lack of knowledge about coolings system and cables. An overview of the ESP values and efficiencies of the specific systems is given in Table 3.1.

Table 3.1: Relevant system ESP and efficiency values, taken from de Vries' work[11]

Component	Value
EM equivalent specific power	$5.92 \text{ kW/kg}$
EM efficiency	0.95 –
Battery specific power	$6.0 \text{ kW/kg}$
Battery Efficiency	0.95 –
Battery state-of-charge	20 %

### 3.2.3. Wing to Propeller Feedback

The wing interacts with the propeller by inducing upwash on the propeller system. A schematic of the wing's upwash is shown in Figure 3.6 in which the red vertical arrow represents the induced upwash on the propeller. The wing's upwash induces an angle of attack on the propeller, subsequently changing the propeller performance. The wing's upwash is dependent on the propeller location in chordwise and spanwise direction and the wing's sectional lift. The induced angle of attack code outputs a three-dimensional velocity component for each propeller in the system.

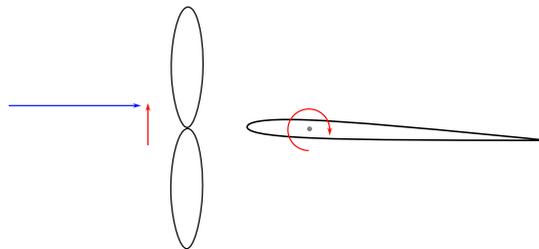


Figure 3.6: Upwash induced from wing on propeller [2]

For now, it is assumed that the panel closest to the propeller hub, in spanwise direction, induces the upwash angle of attack on the propeller. This is a simplified representation of the actual physical phenomenon, since the blade hub and tip are assumed to have wing-induced velocity. The wing affects the propeller over its entire span, as shown by van Arnhem et al. [4]. Van Arnhem's method is elaborated on in [Chapter 7](#).

### 3.2.4. Wing Model Limitations

A number of model limitations have been discussed. This section aims to summarise those limitations and to serve as a basis for [Chapter 7](#), in which model recommendations are discussed.

#### VLM limitations

VLM has a number of limitations, from which the biggest are its inability to model wing thickness, wings at higher angle of attack or viscous forces. The latter can be included by adding a viscous drag model, as was done for OAS. VLM models do not model separation and are thus generally bad at modelling wings at higher angles of attack [2]. It is important to keep this limitation in mind when an optimiser exploits the higher angles of attack design space. To model thickness it would be best to use a higher fidelity wing panel method.

#### Tubular Model Limitations

As will be seen in [Chapter 6](#), the tube structural model did introduce some difficulties since the structural weight would be negative at some iterations. The tube model inherently misses the fidelity of a wingbox. Using a wingbox model would improve the model's accuracy. The possibility of using a wingbox model is discussed in [Section 3.1.2](#).

#### Remeshing Limitations

It was found that the remeshing function can introduce an accumulating error. The error was accumulated because the mesh is not reset at every iteration, which was assumed at first. The remeshing error can cause the aerostructural model to diverge. The diverging of the aerostructural model could become an issue during optimisation. Future iterations of the framework could benefit from a different solution than a remeshing function.

## 3.3. Derivative Verification

Derivative verification for the wing component was limited to only checking the subsystems' codes that had been modified or added. Three partial derivatives were checked for the remeshing function: the derivative of the mesh with respect to the in mesh, the mesh after it has been modified by the other meshing functions, the propeller location and the propeller radius. As can be seen in [Table 3.2](#), the derivatives match up well with a maximum relative error in the order of magnitude  $1e-2$ .

Important to keep in mind is that the remeshing function is discontinuous. This discontinuity was visualised in [Figure 3.4](#). The relative error of  $1e-2$  is acceptable and can likely be attributed to the finite differences scheme that is used as a verification method. Finite differences are notoriously susceptible for errors due to discontinuities, thus there is a realistic possibility that the finite difference scheme is (partly) responsible for the larger relative error.

**Table 3.2: Derivative verification results for remeshing function**

Output, $df$	Input, $dx$	Model Deriv.	Check Deriv.	Relative Error
Mesh	In mesh	1.1336e-04	1.1328e-04	7.0147e-04
Mesh	Propeller location	3.3389e-03	3.3481e-03	1.6237e-02
Mesh	Propeller radius	1.4834e-01	1.4834e-01	4.6858e-05

# 4

## Slipstream Model

The wing and propeller models communicate with each other through a slipstream model. The slipstream model works directly on the aerodynamic influence coefficient matrix by applying a correction factor to it. This correction factor satisfies the boundary conditions of the slipstream that crosses the wing. This correction factor is based on the work of Prabhu [29] and Rethorst [31]. Furthermore, the slipstream model was written in Fortran90 and algorithmically differentiated.

The chapter starts by detailing the slipstream model in Section 4.1. Afterwards, the slipstream model implementation is discussed in Section 4.2, followed by derivative verification in Section 4.3. Model validation is part of Chapter 5, in which the coupled model validation is discussed. Model validation is included in Chapter 5, in which the coupled wing-propeller model is validated against experimental results. The correction factor itself does not have a physical property since it is a mathematical tool to satisfy slipstream boundary conditions. Therefore, the model itself does not have a validation procedure.

### 4.1. Wing Crossing a Finite Slipstream

Koning [18] found that a VLM representation of a wing crossing a finite circular slipstream requires a correction to satisfy the slipstream's boundary conditions. Without a correction factor, and by only changing the slipstream velocity vector on the right-hand-side of the VLM, the slipstream would be of infinite height. However, since the jet has a finite height, and circular shape, the VLM system needs additional terms that modify the Aerodynamic Influence Coefficient (AIC) matrix.

Rethorst [31] extended Koning's [18] work by adapting the correction model to accurately represent wing-slipstream models with higher slipstream to freestream velocity ratios. Rethorst's model has been previously applied to a VLM system to assess propeller-wing models that include either one wing-tip propeller, by R. Willemsen [39], or a dual propeller system with an inboard and wing-tip propeller - by van der Leer [20]. Additionally, Rethorst's model can represent a wing that is submerged in a slipstream with a velocity profile, as shown by Prabhu [29].

A schematic of a wing with and without the Rethorst correction is given in Figure 4.1 and Figure 4.2. The former clearly shows how the slipstream would be of infinite height without a correction factor. Figure 4.2 shows a wing with the Rethorst correction and thus a circular slipstream.

This section will describe the Rethorst correction method. A more elaborate description of Rethorst's method can be found in the work of van der Leer [20], Nederlof [25], Prabhu [29], Rethorst [31] or Willemsen [39].

#### 4.1.1. The Rethorst Model

Rethorst's method is based on the geometry shown in Figure 4.3. The model assumes that there are two velocities; inside the slipstream,  $V_j$ , and outside the slipstream,  $V_\infty$  or the freestream velocity. VLM usually solves for one boundary condition: the normal velocity at the panels should be zero. The strengths of the horseshoe vortices should guarantee that the normal velocity equals zero at the control points. However, when a slipstream of finite height is introduced, two additional boundary conditions are relevant:

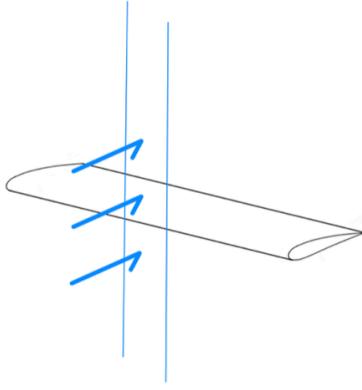


Figure 4.1: Schematic of slipstream crossing wing without Rethorst correction

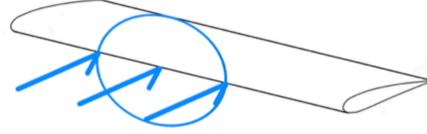


Figure 4.2: Schematic of slipstream crossing wing with Rethorst correction

- The pressure on each side of the jet must be zero (mathematically represented by Eq. (4.1), with  $u_{a,\infty}$  and  $u_{a,j}$  as the disturbance velocity inside and outside the jet in the axial direction, respectively)
- Slipstream continuity needs to be satisfied, i.e. the slipstream velocity needs to be in the same direction in- and outside the jet (mathematically represented by Eq. (4.2), with  $u_{r,j}$  and  $u_{r,\infty}$  as the disturbance velocity inside and outside the jet in radial direction, respectively)

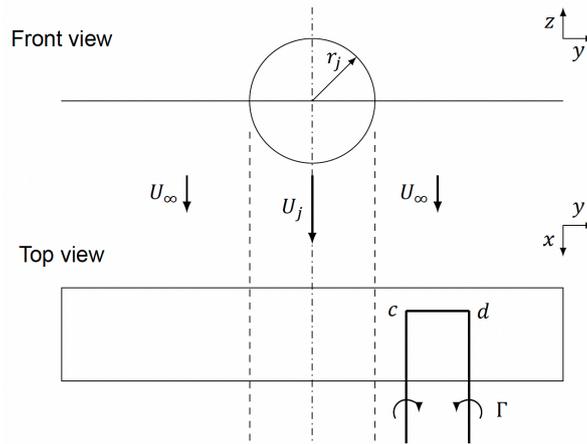


Figure 4.3: Geometry used by Rethorst [31], picture taken from Willemsen [39]

$$V_j u_{a,j} = V_\infty u_{a,\infty} \quad (4.1)$$

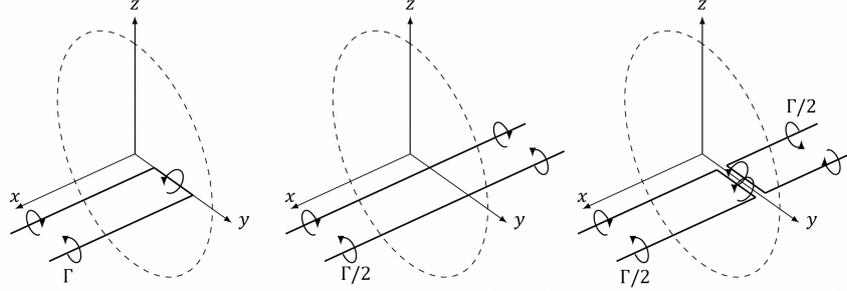
$$V_j u_{r,\infty} = V_\infty u_{r,j} \quad (4.2)$$

To satisfy these boundary conditions, a correction factor matrix is added to the VLM's Aerodynamic Influence Coefficient (AIC) matrix. The AIC values are calculated such that the VLM's boundary condition, normal velocity at control points equals zero, is satisfied. The correction matrix satisfies the second and third boundary condition (given by Eq. (4.1) and Eq. (4.2)), introduced by the slipstream. A VLM without correction factor is given in Eq. (4.3), while Eq. (4.4) shows the VLM formulation with correction matrix.

$$\begin{bmatrix} AIC_{11} & \dots & AIC_{1n} \\ \vdots & \ddots & \vdots \\ AIC_{n1} & \dots & AIC_{nn} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_n \end{bmatrix} = \begin{bmatrix} V_{\perp,1} \\ \vdots \\ V_{\perp,n} \end{bmatrix} \quad (4.3)$$

$$\left( \begin{bmatrix} AIC_{11} & \dots & AIC_{1n} \\ \vdots & \ddots & \vdots \\ AIC_{n1} & \dots & AIC_{nn} \end{bmatrix} + \begin{bmatrix} G_{11} & \dots & G_{1n} \\ \vdots & \ddots & \vdots \\ G_{n1} & \dots & G_{nn} \end{bmatrix} \right) \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_n \end{bmatrix} = \begin{bmatrix} V_{\perp,1} \\ \vdots \\ V_{\perp,n} \end{bmatrix} \quad (4.4)$$

Including the Rethorst correction factor is equivalent to including mirror vortices that satisfy the boundary conditions at the slipstream. The method has a number of similarities with the method one would employ when a wing is near a solid surface: when a wing is near a solid surface, mirror vortices ‘in’ the solid surface guarantee the normal velocity at the solid surface to be zero. For Rethorst’s method, the mirror vortices are split into an even and odd solution, as shown in [Figure 4.4](#).



**Figure 4.4: The full (left), even (middle) and odd (right) mirror vortices used by Rethorst [31], picture taken from Willemssen [39]**

The equations for the even and odd vortices are split up into four distinct parts. Whether the control points and vortices are inside or outside the jet determines which equation is relevant. The distinction between equations is indicated in the subscript of the equations. The first letter of the subscript indicates whether the control point is inside (*j*) or outside (*o*) the jet, and the second letter indicates whether the vortex is inside or outside the jet. The second level subscript —*e* or *o*—indicates whether the equations represents the even or odd solution, respectively. The full set of eight equations is given in [Eq. \(4.5\)–Eq. \(4.12\)](#).

In [Eq. \(4.5\)–Eq. \(4.12\)](#),  $r_j$  represents the jet radius,  $\eta = y/r_j$  the spanwise location,  $\chi = x/r_j$  the chordwise location (both normalised by the jet radius), and  $\mu = V_j/V_\infty$  the ratio of slipstream versus freestream velocity. The variables *e* and *f* are the locations of the edges of the vortices. The odd solution uses modified Bessel function of the first and second kind; *I* and *K*, respectively and their derivatives; *I*′, *K*′. The Bessel functions are all of order  $2\nu + 1$  and the integrals use the dummy running variable  $\lambda$ . A convergence study by Nederlof [25] showed that having  $\lambda$  run from 0 to 20 and using  $\nu = 5$  as an upper boundary is sufficient to converge the odd equations.

$$G_{jj\text{even}}(\chi) = \frac{1 - \mu^2}{5 \cdot (1 + \mu^2)} \left[ \frac{1}{1/f - \chi} - \frac{1}{1/e - \chi} + \frac{1}{1/f + \chi} - \frac{1}{1/e + \chi} \right] \quad (4.5)$$

$$G_{oj\text{even}}(\chi) = \frac{(1 - \mu)^2}{5 \cdot (1 + \mu^2)} \left[ \frac{1}{\chi - e} - \frac{1}{\chi - f} + \frac{1}{f + \chi} - \frac{1}{e + \chi} \right] \quad (4.6)$$

$$G_{jo\text{even}}(\chi) = -G_{oj\text{even}}(\chi) \quad (4.7)$$

$$G_{oo\text{even}}(\chi) = -G_{jj\text{even}}(\chi) \quad (4.8)$$

$$G_{jj\text{odd}}(\chi, \xi) = \frac{8}{5\pi\chi} \sum_{\nu} (2\nu + 1)^2 \int_0^{\infty} \frac{K_\nu K_\nu I_\nu(\xi\chi) \sin(\xi\chi)}{[1/(1/\mu^2) - 1] - \lambda I_\nu K_\nu} \int_{e\lambda}^{f\lambda} \frac{I_\nu \lambda_\beta}{\lambda_\beta} d\lambda_\beta d\lambda \quad (4.9)$$

$$G_{oj\text{odd}}(\chi, \xi) = \frac{8}{5\pi\chi} \sum_{\nu} (2\nu + 1)^2 \int_0^{\infty} \left[ \frac{1}{\mu - \lambda(1/\mu - \mu)I_\nu K_\nu} \right] \frac{K_\nu(\chi\lambda) \sin(\xi\lambda)}{\lambda} \int_{e\lambda}^{f\lambda} \frac{I_\nu \lambda_\beta}{\lambda_\beta} d\lambda_\beta d\lambda \quad (4.10)$$

$$G_{jo\text{odd}}(\chi, \xi) = \frac{8}{5\pi\chi} \sum_{\nu} (2\nu + 1)^2 \int_0^{\infty} \left[ \frac{1}{\mu - \lambda(1/\mu - \mu)I_\nu K_\nu} \right] \frac{K_\nu(\chi\lambda) \sin(\xi\lambda)}{\lambda} \int_{e\lambda}^{f\lambda} \frac{I_\nu \lambda_\beta}{\lambda_\beta} d\lambda_\beta d\lambda \quad (4.11)$$

$$G_{oo\text{odd}}(\chi, \xi) = \frac{8}{5\pi\chi} \sum_{\nu} (2\nu + 1)^2 \int_0^{\infty} \frac{K_\nu K_\nu I_\nu(\xi\chi) \sin(\xi\chi)}{[1/(1/\mu^2) - 1] - \lambda I_\nu K_\nu} \int_{e\lambda}^{f\lambda} \frac{I_\nu \lambda_\beta}{\lambda_\beta} d\lambda_\beta d\lambda \quad (4.12)$$

A comparison between a VLM with and without correction by van der Leer [20] is shown in Figure 4.5. Van der Leer her work clearly indicates the need of a correction factor to prevent a significant lift overestimation. The lift distribution without correction can clearly be seen to predict higher lift values than the distribution with a correction in Figure 4.5.

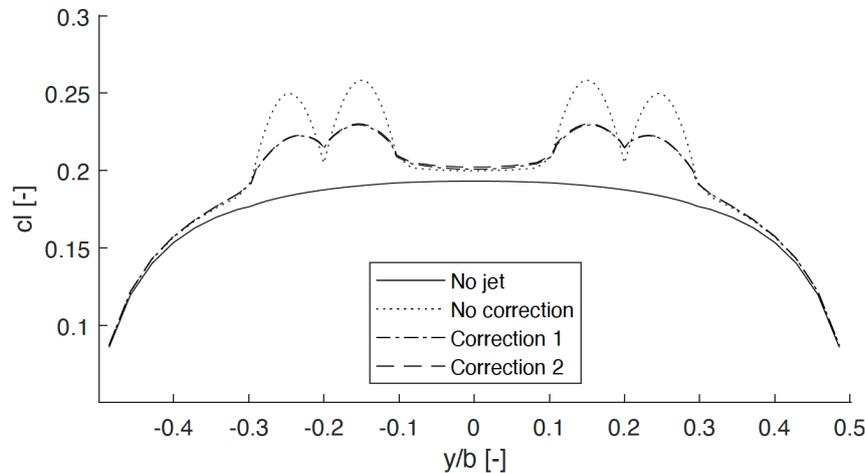


Figure 4.5: A comparison of a VLM with an without a correction factor for the slipstream, picture taken from work by van der Leer [20]

#### 4.1.2. Limitations of the Rethorst Model

To prevent potential issues, it is important to identify a model's limitations and possible disadvantages before determining the implementation strategy. The Rethorst method is based on a number of assumptions that introduce challenges when using the model for optimisation. Table 4.1 gives the limitations associated with the Rethorst model and possible solutions.

Table 4.1: The limitations of the Rethorst (l) with their respective solution (r)

Limitation	Solution
The odd solution is expensive to calculate	Neglect odd solution, discussed in Section 4.2
Slipstream has to be centred on the wing	Extend wing with imaginary part that is thereafter neglected [20, 39], discussed in Section 4.2
Slipstream edges have to be aligned with VLM panels	Slightly modify VLM mesh by moving 4 nodes, discussed in Section 4.2
The centre of the slipstream has to be aligned with a panel centre	Use overset mesh, discussed in Section 4.2.1

## 4.2. Slipstream Model Implementation

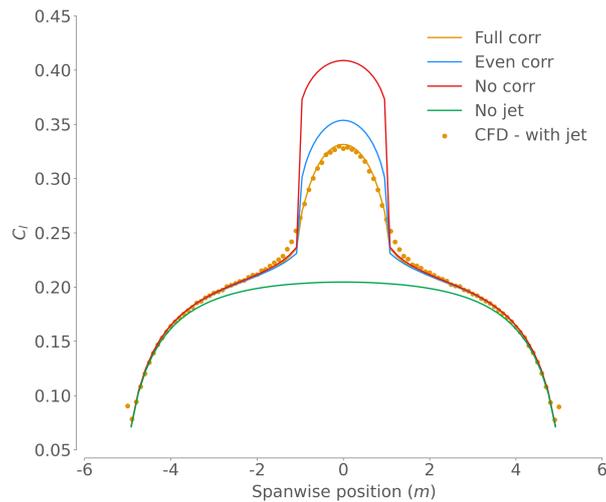
When optimising a system, additional difficulties are introduced as compared to single analyses. The challenges and limitations have been summarised in Section 4.1.2 and serve as a basis for the implementation of the Rethorst method.

The assessment of the odd solution is computationally expensive and therefore forms a problem for 'computationally efficient' optimisation. The largest portion of the computational overhead is caused by the evaluation of Bessel functions. The Python package Numpy offers an algorithm to compute the modified Bessel functions of the first and second kind that is deemed sufficiently fast for optimisation. However, one of the requirements for the optimisation process is to efficiently and accurately assess derivatives. Differentiating Numpy's Bessel function code is not trivial and is left out of the scope of this research project. Therefore, the slipstream model only considers the even solution. Solely including the even solution shows substantially better computational performance as compared to the 'full correction', in which the odd and even solution

corrections are calculated. The 'no correction' performs significantly worse than the even correction case in terms of accuracy. Therefore, neglecting the odd solution gives a decent degree of accuracy, whilst being computationally efficient. The comparison between full correction, even correction and no correction is visualised in Figure 4.6, using CFD data used from Nederlof [25]. Figure 4.6 clearly shows a loss in accuracy when the odd solution is neglected. However, since the calculation procedure's computational time decreases orders of magnitude it is deemed reasonable to neglect the odd solution.

The paper including the Bessel function algorithm that Numpy employs, Algorithm 644 by D.E. Amos [1], is publicly available and could be used in future work to increase the accuracy of the optimisation model.

Important to note is that the odd solution varies with chordwise position with respect to the propeller, due to the  $\mu$  variable in Eq. (4.9)-Eq. (4.12). Therefore, neglecting the odd solution increases the error when the wing is discretised in chordwise direction.



**Figure 4.6: Comparison between no solution, even solution and full solution, CFD data taken from Nederlof's work [25]**

#### 4.2.1. The Overset Mesh

Optimisation introduces additional constraints and challenges, as compared to singular analyses, due to the necessity for the system to be differentiable, flexible and robust. In other words, the correction factor should be able to deform without introducing discontinuities. For a VLM and FEM system this means the size of the mesh panels, with respect to each other, should remain (roughly) the same during the optimisation process, and no nodes should (dis)appear.

Analysis models that include the Rethorst correction use a 'dynamic' mesh, which freely deforms and reallocates panels where necessary. A schematic of the meshing for an analysis method, used by van der Leer [20], Nederlof [25], and Willemsen [39], is given in Figure 4.7, which shows the mesh before and after moving the propeller. This approach works for analyses, but when optimisation is performed it will result in discontinuous derivatives. The system is discontinuous since a node 'jumps' from the left to the right side of the wing, with respect to the propeller. The derivative calculation methods used in the wing-propeller architecture would fail since nodes (dis)appear during the optimisation when either the wingspan, propeller size or propeller location are changed.

Furthermore, using the method shown in Figure 4.7, the optimiser is likely to optimise the mesh instead of the physical system. Optimising the mesh could occur due to the local refinement shown in Figure 4.7 at the propeller location. This local refinement could yield more or less advantageous results. Changing the location of this local refinement might corrupt the optimiser to move to a point where, according to the mesh, an optimum occurs. In other words: the model does not represent the physical system and significant disparities between the mathematical model and physics could occur. The severity of this issue can be assessed by a mesh refinement study. A mesh refinement study is subject for future iterations of the framework as it did not fit in the current scope.

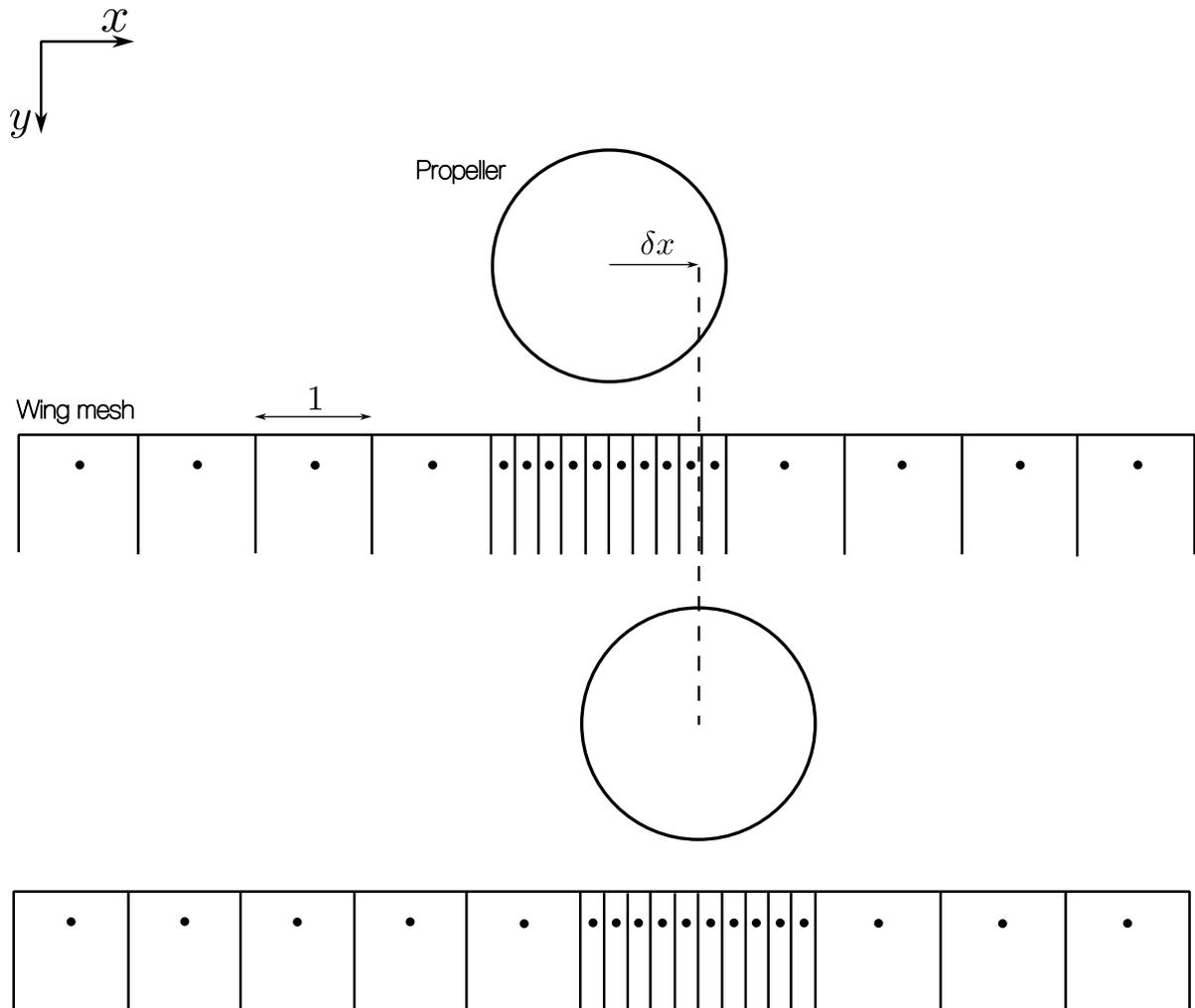


Figure 4.7: The 'dynamic' mesh before and after moving the propeller used for wing-propeller analyses by van der Leer [20], Nederlof [25] and Willemsen [39]

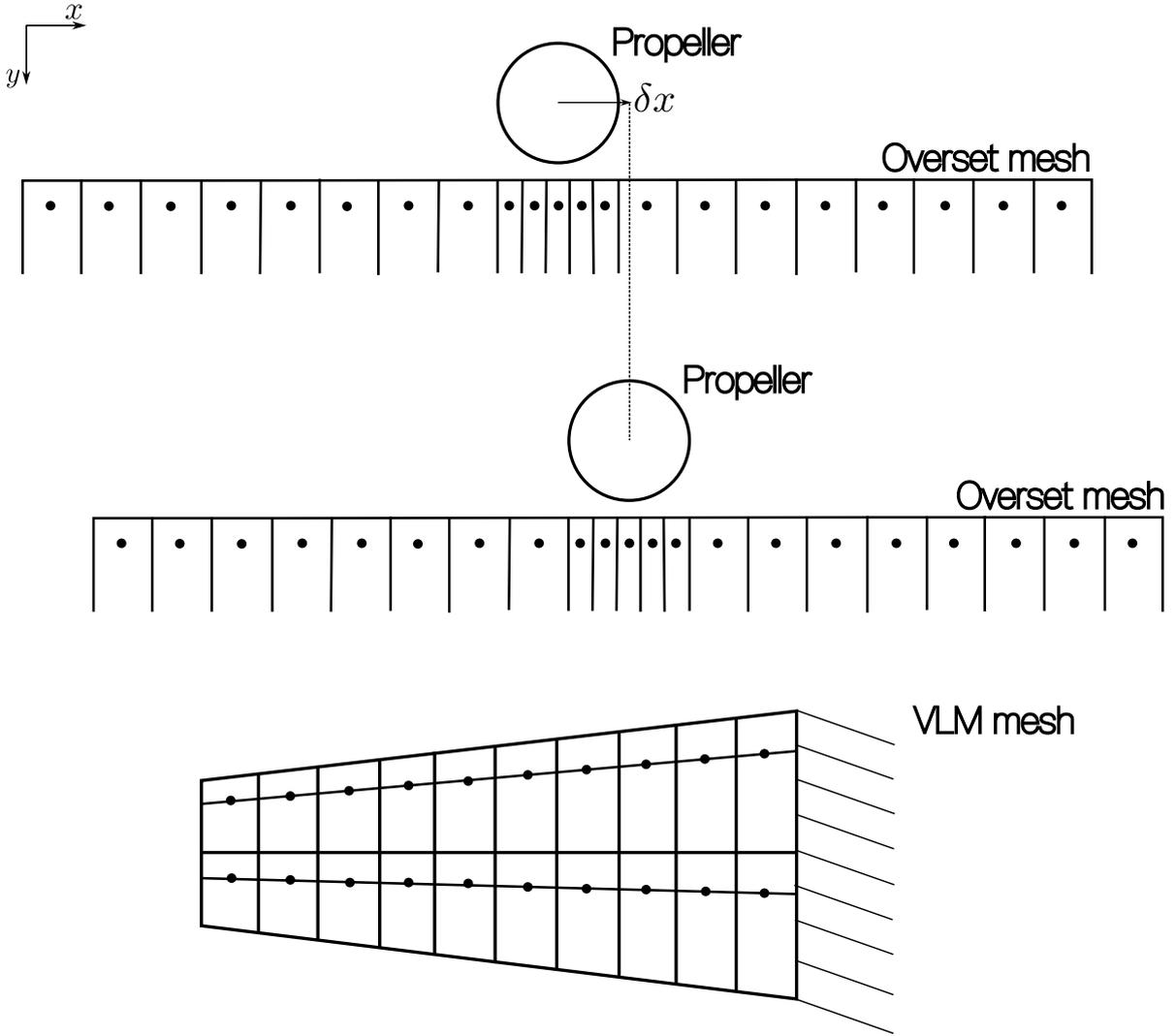
Since the 'dynamic' mesh is not suitable for this framework, a different meshing approach is considered: the overset mesh. Before discussing the overset mesh it is important to understand the limitations of the Rethorst method to guarantee the validity of the overset mesh. As discussed in Section 4.1.2, Rethorst's model is limited by the following requirements:

- The slipstream should be in the centre of the wing,
- The centre of the slipstream should be aligned with the centre of a VLM panel,
- The slipstream edges have to be aligned with the VLM panels.

Without additional effort, these three requirements would restrict the optimisation design space since it would require the propeller location, propeller radius and wing span to be discrete design variables: if the propeller moves, scales or the wing scales it would change the propeller alignment with respect to the VLM mesh.

These requirements are important for calculating the correction matrix but not for the VLM system. Therefore, an overset mesh can be used to extrapolate the correction matrix's values to the VLM mesh. The overset mesh would scale linearly with the propeller radius and move left or right depending on the propeller location. The moving and scaling of the overset mesh, with respect to the VLM mesh, with one chordwise panel for simplicity, is visualised in Figure 4.8 and Figure 4.9, respectively.

The overset mesh is required to span the entire wing at all times. To guarantee the overset mesh spanning the entire wing, the most extreme configurations should be considered. Since wingtip propellers are included



**Figure 4.8: Visualisation of the overset mesh moving with respect to the VLM mesh by a change in propeller location**

as design possibilities, the most extreme configuration will occur when the propeller radius is minimised, the propeller is located at the wingtip, and the wing span is maximised. Eq. (4.13) gives the ratio between overset mesh panels inside and outside the slipstream. Eq. (4.14) gives the equation for the overset mesh span.

$$\Pi_{\text{panel ratio}} = \frac{b_{\text{max}} - 2r_{\text{min}}}{n_{\text{panels, wing}}} / \frac{2r_{\text{min}}}{n_{\text{panels, prop}}} \quad (4.13) \quad b_{\text{overset}} = \frac{2r_{\text{prop}}}{n_{\text{panels, prop}}} \Pi_{\text{panel ratio}} n_{\text{panels, wing}} \quad (4.14)$$

The overset mesh will have a constant number of panels inside the jet, and outside the jet. In other words, a non-uniform spacing is used. This non-uniform spacing guarantees that the propeller is represented by an odd number of panels. Having an odd number of, equally spaced, panels inside the propeller slipstream guarantees that the propeller centre coincides with a panel centre. A schematic of the overset mesh and how the interpolation, without scaling, works is given in Figure 4.10

A flow diagram of the Rethorst model is given in Figure 4.11. The `mod_multiprop.f90` function is the main function that passes the design variables for each propeller in the wing-propeller system. Afterwards, the propeller is discretised by `mod_velocity_distribution.f90` which calculates the correction matrix for each overset mesh. The solution for each overset mesh is interpolated to the VLM mesh using the `mod_overset_interpolation.f90` function. The meshing function outputs the exact same mesh as the

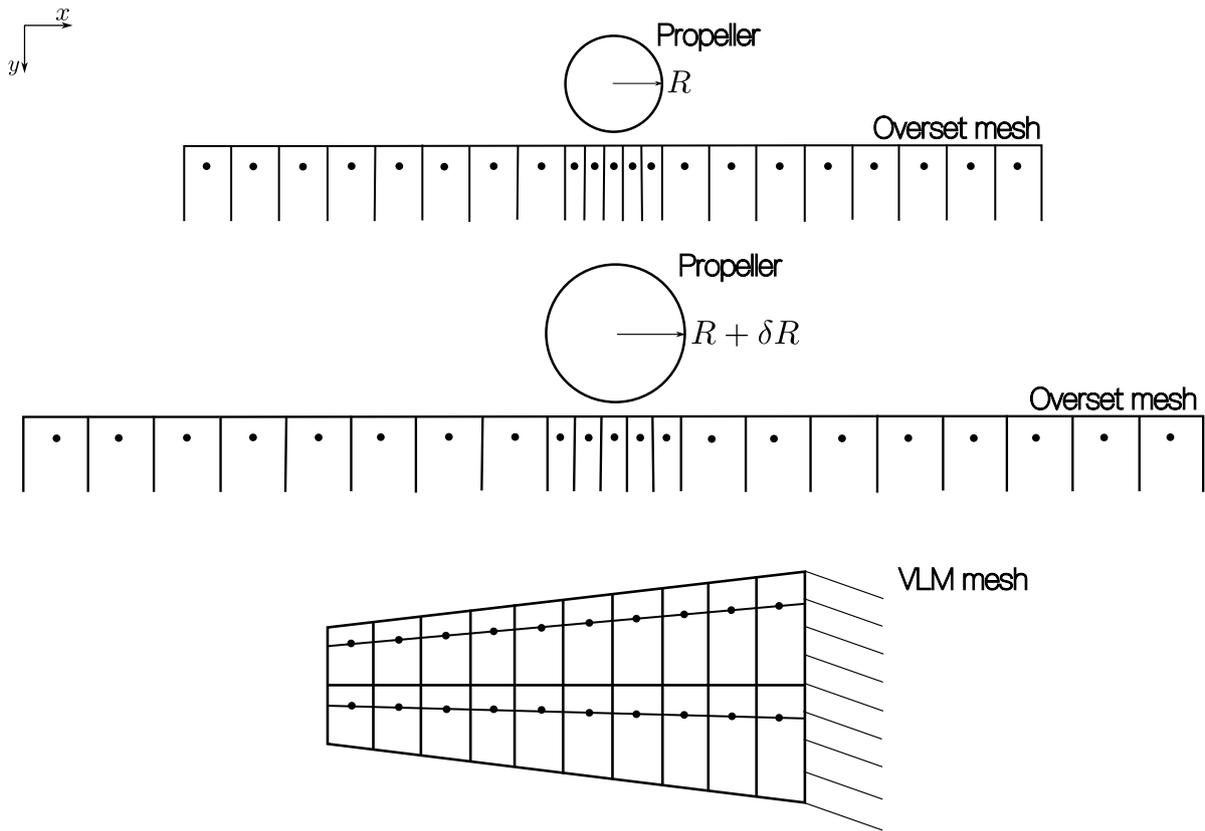


Figure 4.9: Visualisation of the overset mesh scaling with respect to the VLM mesh by a change in propeller radius or wing span

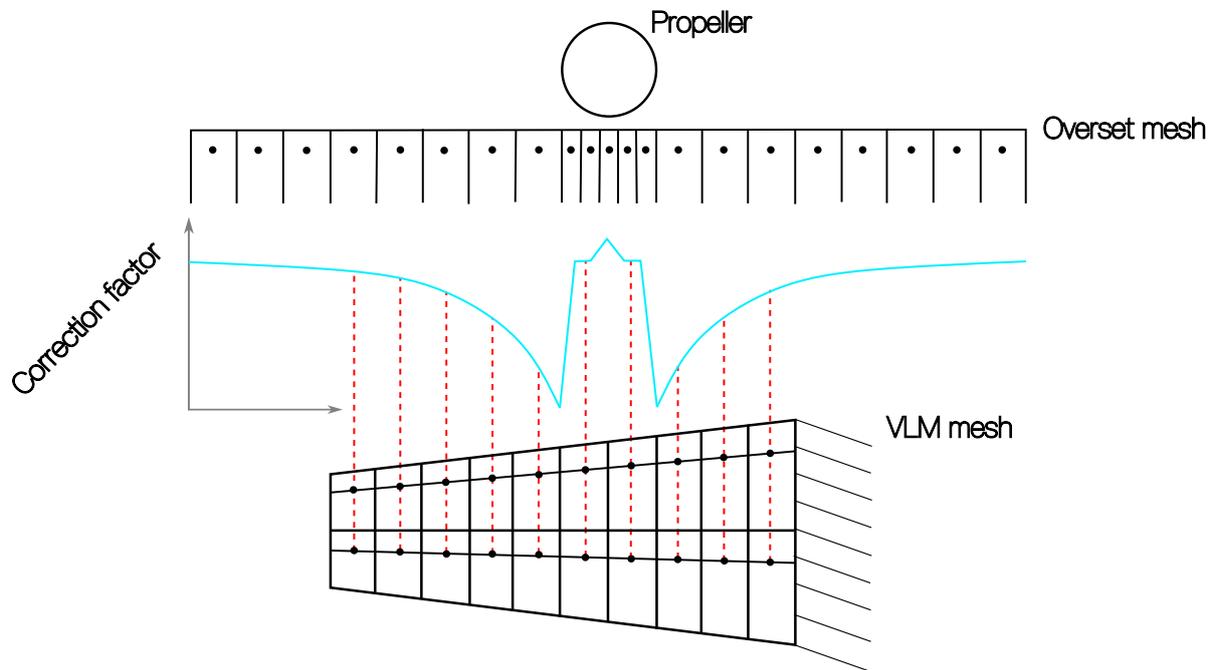
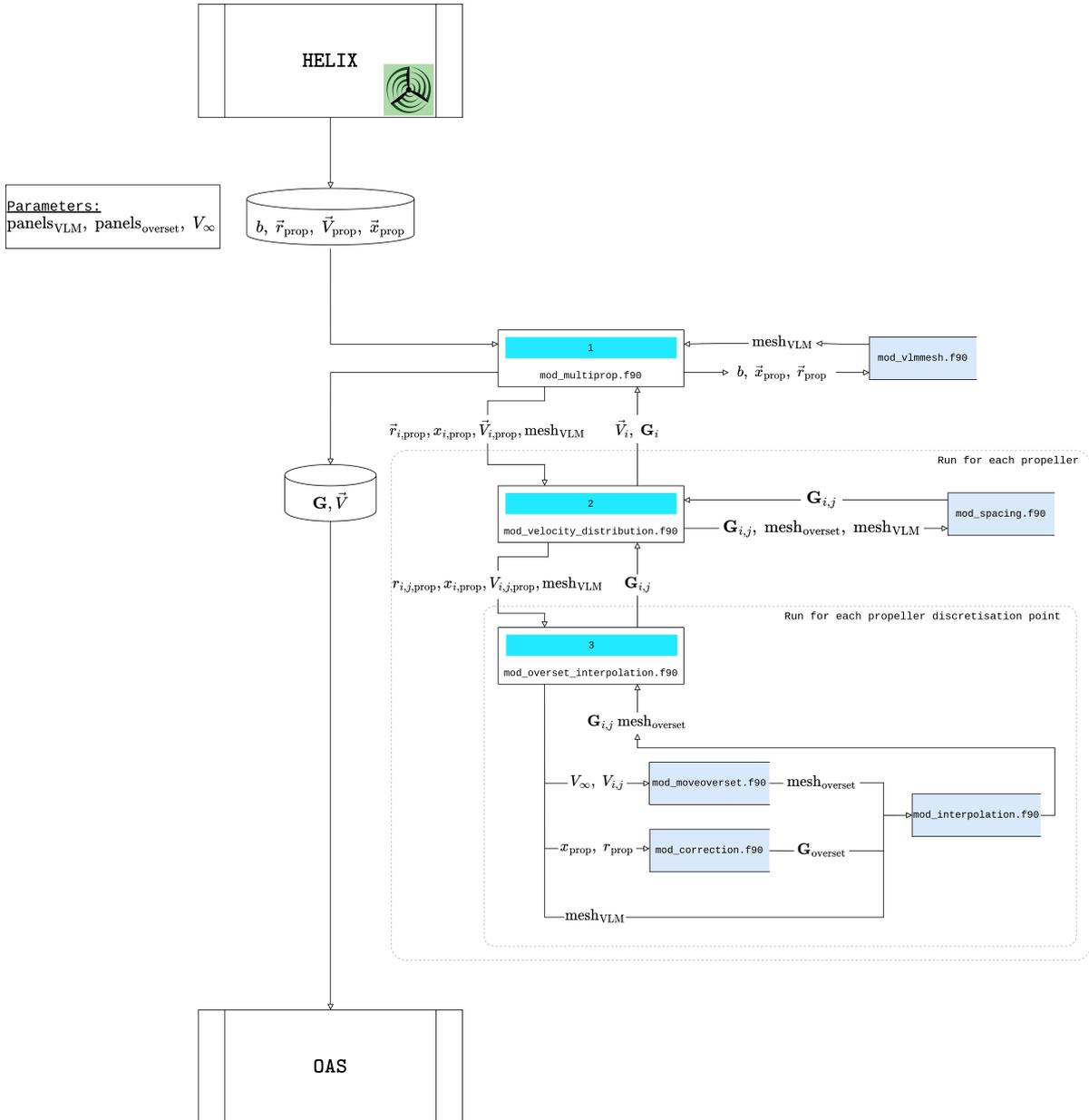


Figure 4.10: A schematic of how the correction factor is interpolated to the VLM mesh for an arbitrary panel

wing model is using. This is necessary to guarantee the correction values are interpolated from the overset mesh to the VLM mesh. A scaling function is used after the oversetmesh values are interpolated to the VLM

mesh. The scaling function is important since the interpolated correction values originate from panels that are smaller than the VLM mesh. The Rethorst method eventually outputs a correction matrix that is unique for the propeller-wing system.



**Figure 4.11: Flow diagram of the Rethorst model. The parameters are passed to the multiprop function, which generates a copy of the VLM mesh by using the VLM mesh function. Afterwards, the design variables are passed to the velocity distribution function which discretises the propeller and passes it to the correction matrix function, that calculates the correction for the overset mesh. Eventually this solution is interpolated to the VLM mesh.**

#### 4.2.2. Wing-Tip Propellers

The wing-propeller system could benefit from wing-tip propeller configurations due to wing-tip vortex dispersion [34]. Wing-tip propeller configurations were included in the research objective for this reason. Important for wing-tip propellers is that only one panel is modified by the remeshing function. Wing-tip propellers introduce a discontinuity in the system, since the remeshing function changes when the slipstream edge passes

the wing span. It has to be considered that this discontinuity could introduce errors and difficulties for the optimisation procedure. Difficulties could occur since, perhaps, modifying 2 panels instead of 4 per propeller is more beneficial from a mathematical point of view. If modifying fewer panels is more beneficial, the optimiser might prefer wing-tip propellers although these are not more beneficial in reality. Furthermore, wing-tip vortex dispersion can not be modelled with the current framework since spanwise flow components are not considered.

### 4.2.3. Performance Enhancements

Another benefit of using an overset mesh is that the correction matrix is symmetric around all 4 axes, since the propeller is always located on the overset mesh centre. The symmetry can be verified by looking at the Rethorst correction equations given by Eq. (4.5) through Eq. (4.12). The equations depend on the distance to the slipstream centre, thus it does not matter whether the calculations occur on the left or right side of the propeller. Using the symmetry condition speeds up the correction factor and derivative calculations by a factor of 4. To use this symmetry condition the upper left quarter of the total correction matrix is assessed, after which this quarter is inverted around the vertical axis (for the upper right quarter), the horizontal (for the lower left quarter) and the diagonal axis (for the lower right quarter). Eq. (4.15) shows a correction matrix without an imposed symmetry condition on the left, where the matrix on the right does include the symmetry condition. The matrices are equivalent in terms of values, but the matrix on the right takes 4 times as little computational overhead to compute.

$$G = \begin{bmatrix} G_{1,1} & G_{1,2} & G_{1,3} & G_{1,4} & G_{1,5} \\ G_{2,1} & G_{2,2} & G_{2,3} & G_{2,4} & G_{2,5} \\ G_{3,1} & G_{3,2} & G_{3,3} & G_{3,4} & G_{3,5} \\ G_{4,1} & G_{4,2} & G_{4,3} & G_{4,4} & G_{4,5} \\ G_{5,1} & G_{5,2} & G_{5,3} & G_{5,4} & G_{5,5} \end{bmatrix} = \begin{bmatrix} G_{1,1} & G_{1,2} & G_{1,3} & G_{1,2} & G_{1,1} \\ G_{2,1} & G_{2,2} & G_{2,3} & G_{2,2} & G_{2,1} \\ G_{3,1} & G_{3,2} & G_{3,3} & G_{3,2} & G_{3,1} \\ G_{2,1} & G_{2,2} & G_{2,3} & G_{2,2} & G_{2,1} \\ G_{1,1} & G_{1,2} & G_{1,3} & G_{1,2} & G_{1,1} \end{bmatrix} \quad (4.15)$$

The symmetry is visually shown in Figure 4.12. The correction factors for the left and the right side of the wing, with respect to the propeller are the same. It should be noted that when the odd correction is used and the number of chordwise panels is larger than 1, the symmetry condition does not apply. The odd correction is dependent on the normalised chordwise distance to the propeller as well.

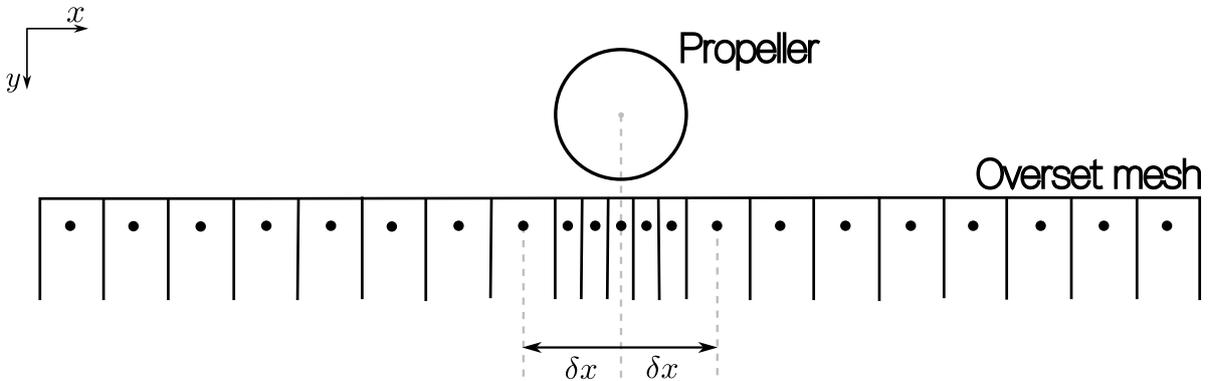
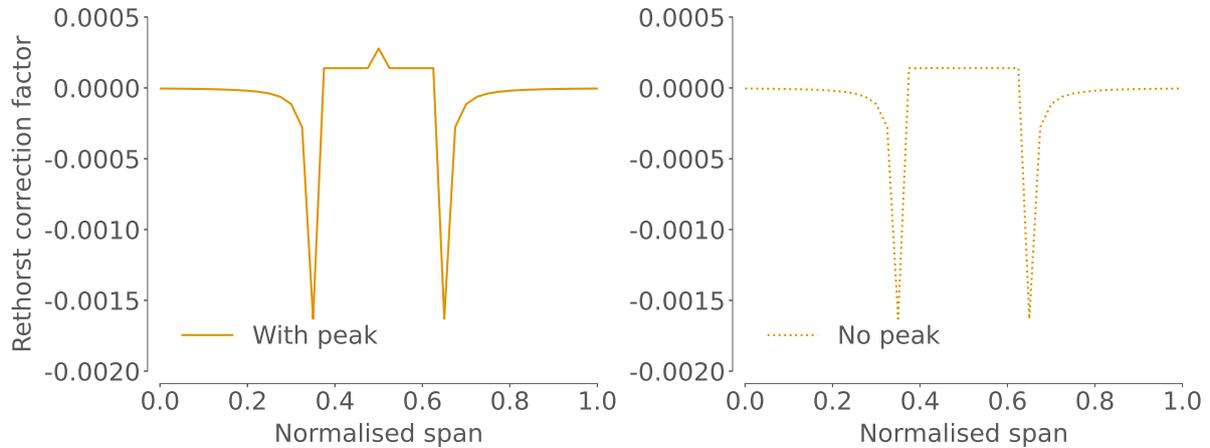


Figure 4.12: The symmetry of the Rethorst correction stems from each panel's control point having the same distance to the propeller centre. The figure shows that the calculations for the right and left side of the wing use the same input variable  $\delta x$

### 4.2.4. Overset Accuracy Assessment

The Rethorst correction factor has a 'peak' at the centre of the propeller for each correction matrix row. The peak is a mathematical artefact of the correction factor for the centre panel. The peak can be seen at half span in the left graph of Figure 4.13, where the graph on the right of Figure 4.13 gives the Rethorst correction without a peak. A potential issue with the overset mesh is that it does not guarantee translating this peak to the VLM at every iteration. A system in which the peak is not interpolated to the VLM mesh is shown in Figure 4.10. A concern was that neglecting this peak could lead to model errors. A model with

this peak, and a model without this peak were compared to assess whether it has a significant effect on the model output. It was seen that with or without the peak almost identical results were retrieved. Therefore, the peak can be neglected.



**Figure 4.13: Arbitrary Rethorst correction for a given panel with a peak (l) and no peak (r) at its centre**

Figure 4.10 shows the significance of using a highly refined VLM mesh. If the VLM mesh is too coarse, the interpolated correction matrix might neglect an important part of the correction factor. Moreover, the propeller velocity distribution is discretised based on the VLM mesh.

After the remeshing procedure, the induced propeller velocity distribution is calculated, using the VLM mesh and the slipstream velocity distribution. A schematic of how the induced propeller velocity is calculated from the propeller velocity distribution given by the propeller model is shown in Figure 4.14. The location of the VLM panels determines what velocity is induced on the wing for that panel. The number of discretisation points is constant, as a varying number of discretisation points would return a highly discontinuous system. The discontinuities could occur due to the propeller being discretised in, for instance, four steps the first iteration and five steps the next iteration.

It might be speculated that a large number of propeller discretisation points is favourable because it would represent the induced velocity more accurately. Although this statement is factually correct, it should be considered that the VLM mesh has to accommodate these discretisation points. Taking Figure 4.14 as an example: if the user defines 10 slipstream velocity discretisation points, the simulation would crash, since only three VLM nodes are inside the propeller radius. The number of VLM nodes inside the propeller radius is affected by the propeller radius, wing span, and the spanwise VLM refinement. Therefore, the spanwise VLM discretisation and number of propeller velocity discretisation points should be set such that the system can accommodate the smallest propeller radius with the largest wing span.

#### 4.2.5. Slipstream Velocity Profile

Prabhu [29] showed that Rethorst's correction factor can be super positioned to represent a velocity distribution. Prabhu's work on slipstream velocity distribution was later applied by van der Leer [20] and Willemssen [39] in their aircraft analysis studies, which showed that addition *and* subtraction are possible to create a slipstream velocity profile. Subtraction is necessary since sole addition of velocity increments creates a single velocity 'bubble', where a propeller slipstream consists of a double bubble shape. The single versus 'double bubble' velocity profile is visualised in Figure 4.15.

The fidelity of the velocity profile depends on the number of discretisation points, the more discretisation points, the more the velocity profile will resemble a smooth velocity profile. Much like the number of VLM panels, the number of velocity profile discretisation points must be determined before the optimisation is performed. Changing the number of discretisation points during the optimisation could result in discontinuous derivatives, and thus possibly making the optimiser diverge. Figure 4.16 shows how an addition of velocity increments and decrements results in a velocity profile. The number of discretisation points is equal to the number of velocity profiles shown in Figure 4.16.

The location of the discretisation points is dependent on the wing VLM mesh. The discretisation points

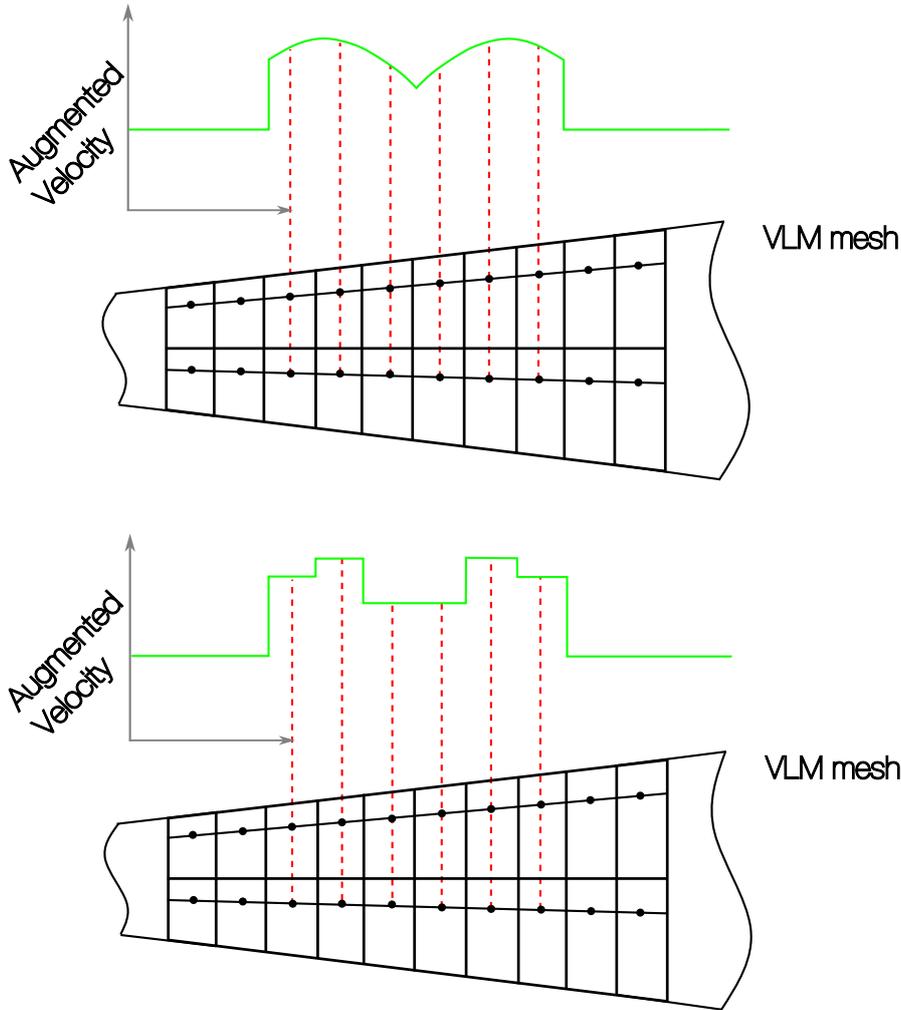


Figure 4.14: A schematic of how the propeller velocity distribution is translated to the VLM

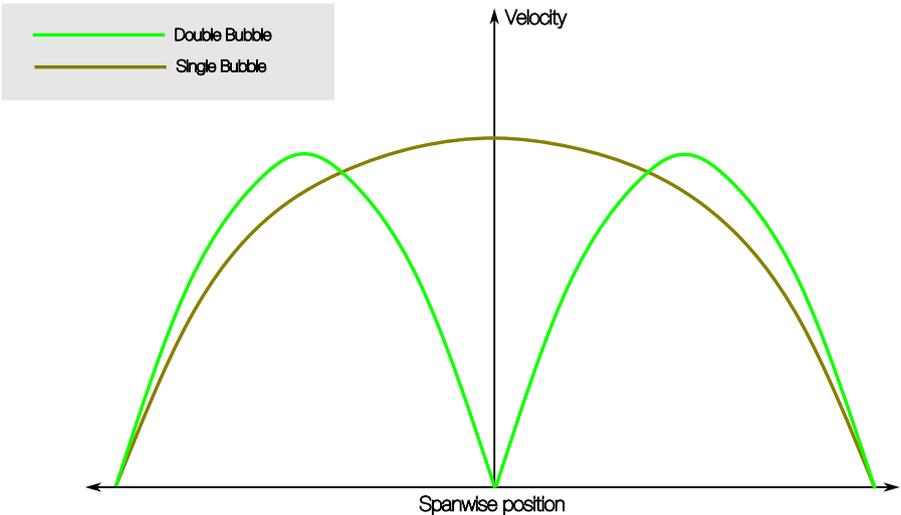
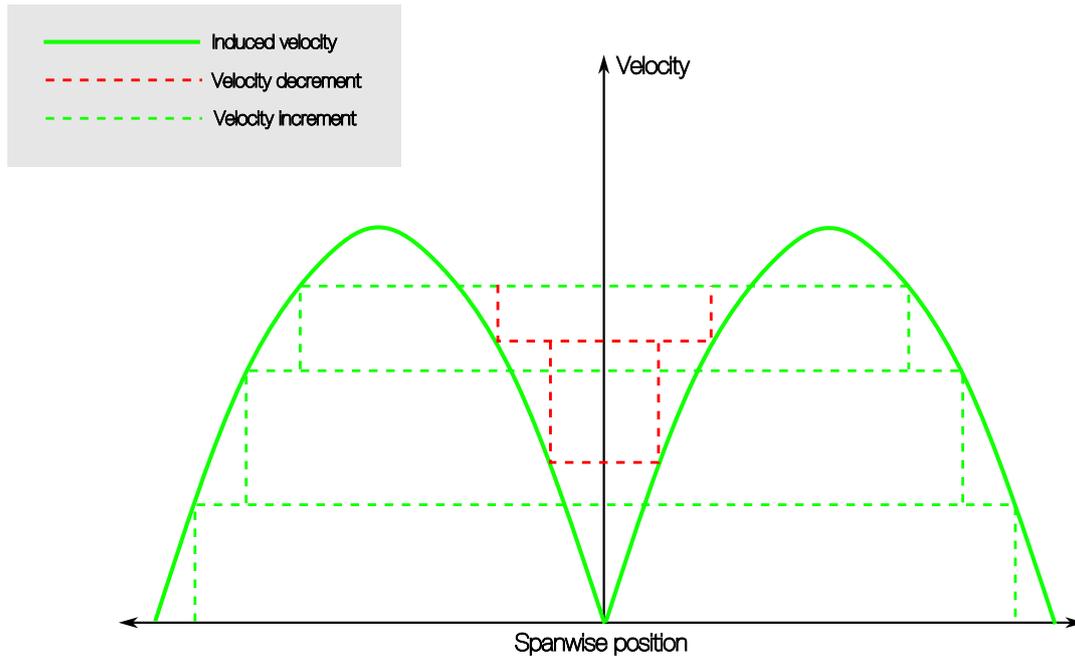


Figure 4.15: A comparison of two slipstream velocity distributions, for which either only addition (green) or addition plus subtraction of correction factor values (red) is necessary

have to respect the VLM panel locations since each VLM panel needs to be assigned a discrete velocity value. Decreasing the propeller radius or decreasing the wing span might therefore result in a deficit of VLM panels (in spanwise direction) and result in an error. Therefore, the propeller radius, number of VLM panels and wing span need to be carefully bounded before the optimisation to make sure the optimisation does not return an error. The optimisation architecture has a built-in feature that checks whether the propeller velocity profile discretisation points can ever be larger than the number of VLM panels inside the propeller radius.

to conclude, the overset mesh implementation considers all aforementioned elements and can therefore effectively represent a velocity profile. It is important to guarantee that a sufficient number of VLM panels are inside the slipstream such that the VLM mesh can accommodate the number of slipstream discretisation points.



**Figure 4.16:** The schematic shows how the ‘double bubble’ velocity profile can be obtained using addition and subtraction of velocity increments (green) and decrements (red)

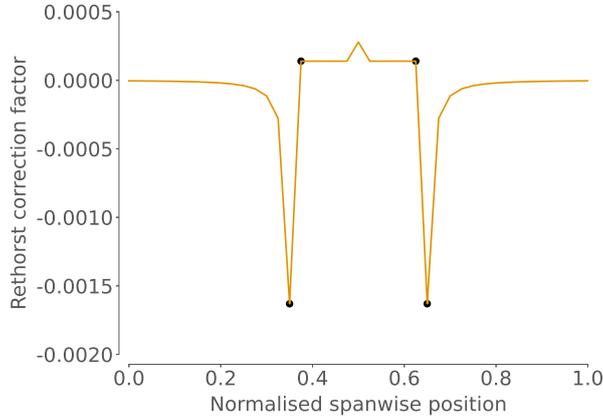
#### 4.2.6. Limitations of an Overset Mesh

The limitations that are inherently associated with an overset mesh are related to a loss of computational efficiency and accuracy. Part of the calculated solution is neglected as the propeller radius increases, since the overset mesh scales with the propeller radius as shown in Figure 4.9. The solutions that are neglected are those that are not interpolated to the VLM mesh. Computing values that are thereafter neglected are a direct loss of computational efficiency. This loss of computational efficiency can be resolved by only calculating the relevant part of the overset mesh, which could be implemented in future iterations of the framework. A second source of computational efficiency loss is the additional interpolation step to extract the correction values for the VLM mesh from the overset mesh values. This is a step that is unfortunately required for an overset mesh and can not be prevented.

Interpolating values from the overset mesh to the VLM mesh results in a loss of accuracy. Loss of accuracy is significant when a volatile function is being interpolated to a coarse mesh. The Rethorst correction model returns a substantially volatile solution, increasing the interpolation accuracy loss. Therefore, it is important to use a fine VLM mesh to properly capture the correction factor’s behaviour. A wing mesh with 100 spanwise panels consistently converged for several design variable values, and was thus considered sufficiently accurate.

### 4.3. Derivative Verification

The symbolically (forward and reverse) derived code adheres to the code structure shown in [Figure 4.11](#). Derivative verification was particularly important for the slipstream since this model was built from scratch, thus none of the derivatives had been defined yet. Furthermore, the model inherently has some discontinuities. The discontinuities occur either due to the four distinct equations used for the correction factor or the mesh function. The discontinuities due to the four correction factor functions occur at the dots in [Figure 4.17](#). The discontinuities are expected to be small and not cause any significant issues in the optimisation process.



**Figure 4.17: The discontinuities occur in between the black dots because the Rethorst correction switches between equations.**

Derivatives can be verified in 4 steps, as summarised below:

1. Obtain Finite (FD) Derivatives
2. Obtain Complex Step (CS) Derivatives
3. Obtain Forward Algorithmic (AD) Derivatives
4. Obtain Reverse Algorithmic (AD) Derivatives

As stated in [Chapter 1](#), a fast and accurate way to calculate derivatives is crucial to the research objective. The fastest and most accurate way to obtain derivatives is to use the adjoint method, which is much more accurate than finite differencing the entire system but less computationally expensive than CS or AD. This section will focus on how the partial derivatives, necessary for the adjoint method elaborated on in [Section 5.4](#), are obtained. A schematic of the various methods with which one can obtain derivatives and their respective accuracy and computational expense is given in [Figure 4.18](#). All AD procedures were automated by using the open-source software Tapenade [\[15\]](#).

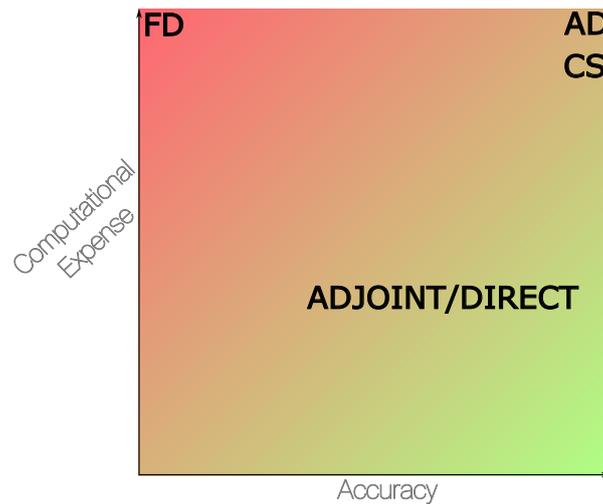
#### 4.3.1. Finite Differences

A first derivative estimate can be obtained using finite differences. Finite differences are based on perturbing the model with an infinitesimal step and assessing the change in output function. One can choose to use a *forward*-, *backward*-, or *central*-difference scheme. The equations for all three are given by [Equations \(4.16\)](#) through [\(4.18\)](#), respectively. In these equations,  $\hat{e}_j$  represent the unit vector in the  $j$ th direction of the system and  $h$  is the step-size.

$$\frac{\delta f}{\delta x_j} = \lim_{h \rightarrow 0} \frac{f(x + h\hat{e}_j) - f(x)}{h} \approx \frac{f(x + h\hat{e}_j) - f(x)}{h} + \mathcal{O}(h) \quad (4.16)$$

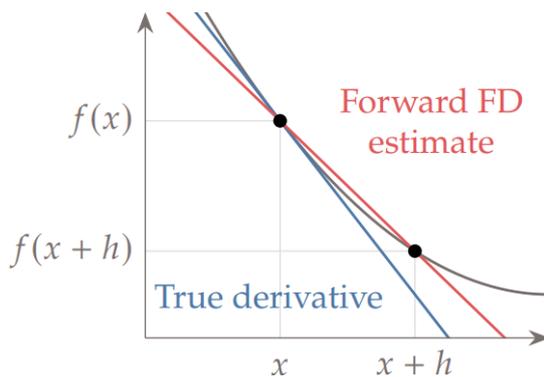
$$\frac{\delta f}{\delta x_j} = \lim_{h \rightarrow 0} \frac{f(x) - f(x - h\hat{e}_j)}{h} \approx \frac{f(x) - f(x - h\hat{e}_j)}{h} + \mathcal{O}(h) \quad (4.17)$$

$$\frac{\delta f}{\delta x_j} = \lim_{h \rightarrow 0} \frac{f(x + h\hat{e}_j) - f(x - h\hat{e}_j)}{2h} \approx \frac{f(x + h\hat{e}_j) - f(x - h\hat{e}_j)}{h} + \mathcal{O}(h^2) \quad (4.18)$$

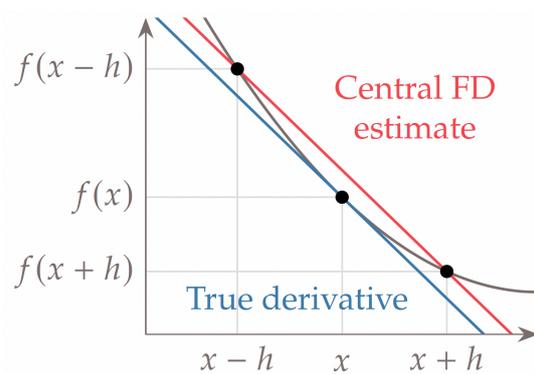


**Figure 4.18:** Schematic comparing the computational expense and accuracy of four different methods with which model derivatives can be obtained

Evident from Eq. (4.16) through Eq. (4.18) is that the truncation error is smallest for the central difference scheme, which therefore is the most accurate method of computing derivatives with a finite difference scheme. The truncation error stems from the Taylor series expansion. Graphically, the increase in accuracy for a central difference scheme is apparent by comparing Figure 4.19 and Figure 4.20: the derivative returned by the latter is more accurate than the derivative of the former.



**Figure 4.19:** Forward difference scheme [23]



**Figure 4.20:** Central difference scheme [23]

Unfortunately, finite difference schemes suffer from the *step-size dilemma*. The error of finite difference methods scales with either  $h$  or  $h^2$ , as given by  $\mathcal{O}(h)$  and  $\mathcal{O}(h^2)$ . One would like to choose the step-size  $h$  as small as possible to reduce the error. However, as  $h$  get smaller, a round-off error, called *subtractive cancellation*, becomes dominant. Given this second error, one can reason there must be an optimum step-size. This optimum is graphically shown in Figure 4.21. Finding this optimum step-size is non-trivial, thus making finite differences an unattractive option.

Lastly, one needs to be aware of noisy solutions in finite-difference approximations. When choosing the wrong step-size in a noisy solution a finite difference schema could even return the wrong sign for the derivative, as shown in Figure 4.22.

### 4.3.2. Complex Step Differences

The complex step derivative approximation uses complex variables to compute the derivatives of real functions [22]. The complex-step method is accurate but computationally expensive. Furthermore, complex step implementation is hard since the code must be complexified, i.e. rewritten such that complex numbers can be used. The complex-step method is similar to finite differences, derived by using a Taylor series expansion.

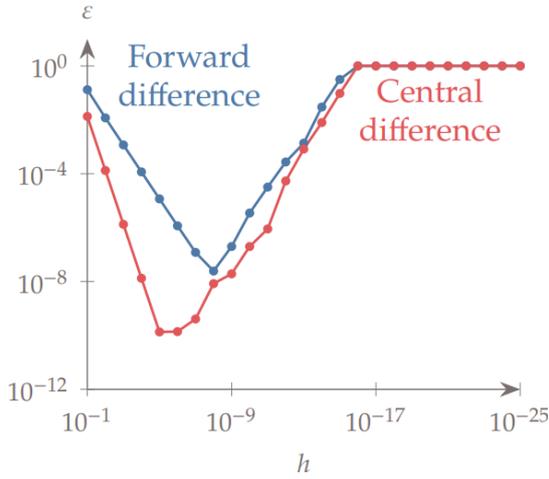


Figure 4.21: Optimal step-size  $h$  shown graphically [23]

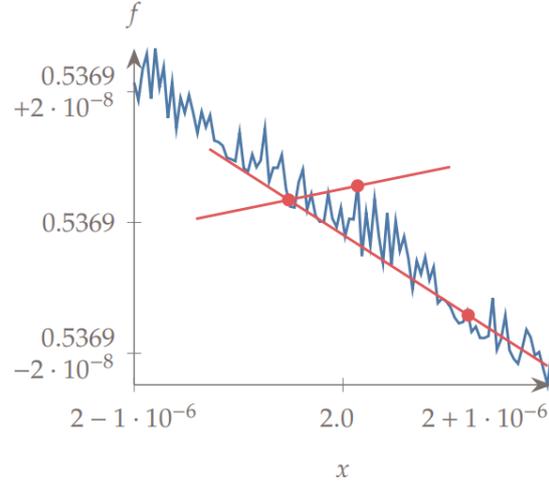


Figure 4.22: Example of the significance of step-size in a noisy function [23]

However, instead of using a real step  $h$ , one now treads into the Imaginary domain by using step-size  $ih$ . Given that  $f$  is a real *analytic* function the Taylor series expansion would become (the  $i$  disappears in front of the third term on the right hand-side since  $i^2 = -1$ ):

$$f(x + ih\hat{e}_j) = f(x) + ih \frac{\delta f}{\delta x_j} - \frac{h^2}{2} \frac{\delta^2 f}{\delta^2 x_j^2} - i \frac{h^3}{6} \frac{\delta^3 f}{\delta x_j^3} + \dots \quad (4.19)$$

The imaginary part of both sides in Equation (4.19) gives:

$$\text{Im}(f(x + ih\hat{e}_j)) = h \frac{\delta f}{\delta x_j} - i \frac{h^3}{6} \frac{\delta^3 f}{\delta x_j^3} + \dots \quad (4.20)$$

Eventually, Eq. (4.20) provides the basis for the complex-step derivative approximation given by Eq. (4.21):

$$\frac{\delta f}{\delta x_j} = \frac{\text{Im}(f(x + ih\hat{e}_j))}{h} + \mathcal{O}(h^2) \quad (4.21)$$

One must provide a complex number with perturbation to this approximation and take the imaginary part of the output to obtain the derivative. Computer programmes must be equipped to deal with imaginary functions, which is straightforward for languages like Python and Matlab. In addition to the derivative, the value of the function is given by the real part of the Taylor expansion in Eq. (4.19):

$$f(x) = \text{Re}(f(x + ih\hat{e}_j)) + \mathcal{O}(h^2) \quad (4.22)$$

The main benefit of the complex-step is the fact that the subtraction error is cancelled since no subtraction is necessary. Therefore, the only source of numerical error is the truncation error, until the step-size gets in the order of  $10^{-200}$  and the numerical precision error of  $f$  becomes dominant.

To answer the question ‘How small should  $h$  be?’, it is important to realise that the smallest representable number, when working with double precision, is approximately  $10^{-324}$ . Any number smaller than this will become zero. Therefore, it is important to choose a certain  $h$  that causes the imaginary part of the function to be non-zero. The error versus step-size for forward-difference, central-difference, and complex-step are visualised in Figure 4.23.

### 4.3.3. Forward AD Verification

For forward AD the model undergoes symbolic differentiation after which a code including all the partial derivatives is returned. The total derivatives are eventually obtained by multiplying the partial derivatives according to the chain rule. A schematic of forward AD for an arbitrary wing model is given in Figure 4.24. The forward AD code sets one of the system inputs seeds to one, whilst the other seeds are set to zero.

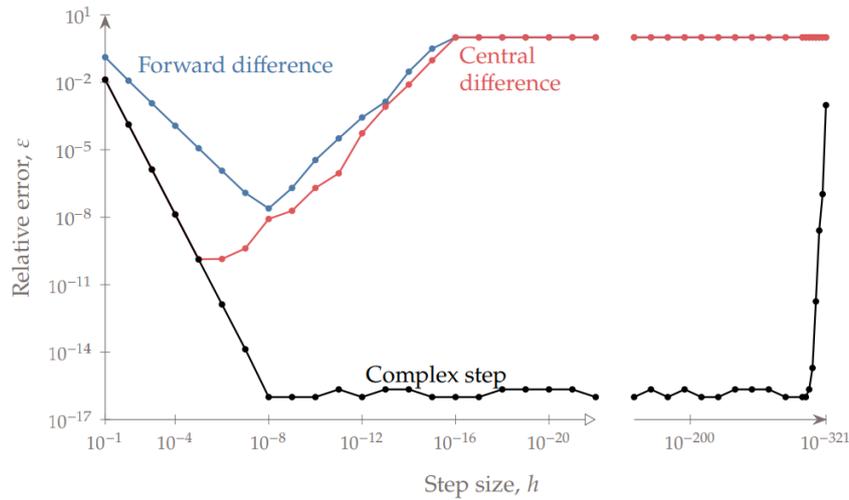


Figure 4.23: FD versus CS accuracy comparison [23]

By setting the seeds the code will only calculate the output derivatives with respect to the design variable with a non-zero seed value. Figure 4.24 gives an arbitrary example in which the twist seed is set to one. The output seeds in Figure 4.24 that are red are the affected outputs whilst the grey output seeds remain unaltered. The grey seeds therefore indicate their derivatives, with respect to twist, to be zero. This process is repeated for each design variable and therefore *forward AD scales linearly with the number of inputs*.

### Forward AD

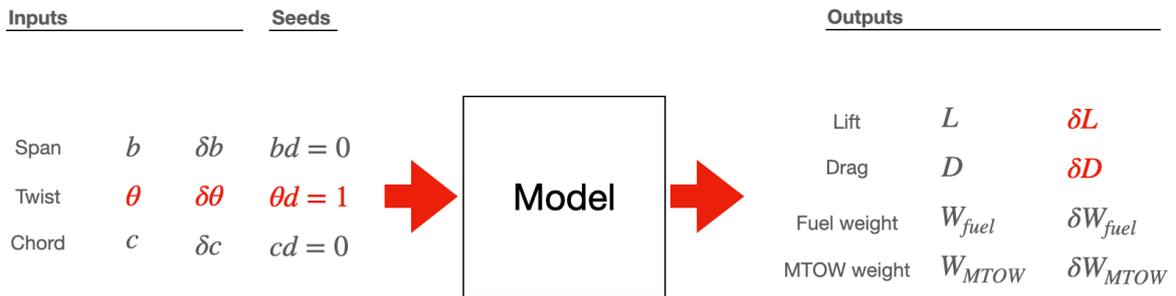


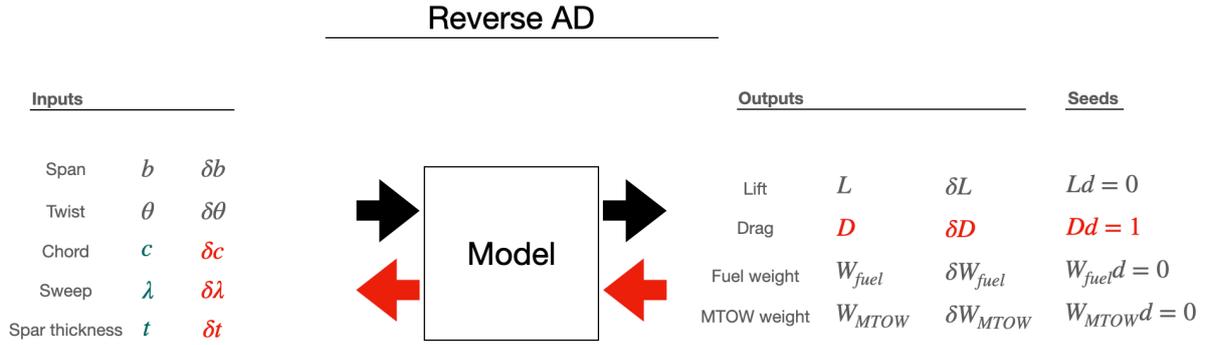
Figure 4.24: Schematic of forward AD of an arbitrary wing model, with the red arrow representing the program direction in which the derivatives are calculated

Verifying the forward AD derivatives can be performed by comparing them to the complex step differences obtained in the previous step of the verification procedure.

#### 4.3.4. Reverse AD Verification

As mentioned in Section 4.3.3, forward AD scales linearly with the number of inputs. This could be a problem when there is, for example 100 inputs and only one output, since that means the code has to be run 100 times, which is inefficient. Reverse AD solves this problem since it does not set seeds for the inputs but for the outputs, as can be seen in Figure 4.25. The reverse differentiated code is first run in forward mode, during which all intermediate values are stored. Afterwards, the output seeds are considered and the code is run in reverse, eventually returning the derivatives of the inputs with respect to the outputs (which can then be inverted to obtain the derivatives of outputs versus inputs). This process is repeated for each output and therefore *reverse AD scales linearly with the number of outputs*.

Verifying the reverse AD derivatives is slightly more complex than verifying any of the previous derivatives. The reverse AD code return the inverse of the derivatives returned by the forward AD model. Assuming the



**Figure 4.25: Schematic of reverse AD of an arbitrary wing model, with the red arrow representing the program direction in which the derivatives are calculated**

linear system can be inverted, the verification procedure needs to satisfy Eq. (4.23) (otherwise called the ‘dot-product test’), with subscripts  $d$  and  $b$  indicating forward and reverse seeds, respectively, and  $x$  and  $y$  as in- and output, respectively. Important in the reverse AD verification procedure is to use the output seeds that were returned by the forward AD code.

$$\vec{x}_d \cdot \vec{x}_b = \vec{y}_d \cdot \vec{y}_b \quad (4.23)$$

Each model will be configured with both reverse and forward mode AD. Both methods have to be configured since the entire system will either use forward or reverse AD (depending on whether the adjoint or direct method is used, as elaborated on in Section 5.4). Important to realise is that a single reverse AD analysis will inherently take at least twice as long as a single forward AD analysis. Therefore, a careful study should be made of what mode is desirable.

#### 4.3.5. Derivative Verification Results

The derivative (forward and reverse) verification returned the forward and reverse AD derivatives given in Table 4.2 and Table 4.3, respectively. In Table 4.2 and Table 4.3,  $\varepsilon$  is the relative error between the AD derivatives and control derivatives,  $\mathbf{G}$  is the correction matrix and  $\vec{V}$  is the velocity induced on the wing. With relative errors in the order of magnitude  $1e-5$  to  $1e-7$  it is more likely that the numerical precision of the finite differences causes the error than the AD code. Therefore it can be stated with high confidence that the Rethorst derivatives are verified and the slipstream model can be used to run the optimisation with the direct and adjoint method.

**Table 4.2: Forward AD derivative verification results**

Output, $df$	Input, $dx$	Forward AD	Check Deriv.	$\varepsilon$
<b>G</b>	Propeller location	1.0554e-03	1.0554e-03	3.8162e-05
<b>G</b>	Propeller radius	1.1271e-03	1.1271e-03	3.0164e-05
<b>G</b>	Propeller velocity output	3.3661e-04	3.3661e-04	1.1726e-04
<b>G</b>	Wing span	3.3661e-04	3.3661e-04	2.3568e-06
$\vec{V}$	Propeller location	8.4362e+01	8.4362e+01	4.5852e-08
$\vec{V}$	Propeller radius	7.7010e+01	7.7010e+01	5.8645e-08
$\vec{V}$	Propeller velocity distribution	9.8029e+00	9.8029e+00	1.1110e-07
$\vec{V}$	Wing span	2.5783e+01	2.5783e+01	4.9110e-07
$\vec{V}$	$V_\infty$	1.2166e+01	1.2166e+01	8.2740e-08

## 4.4. Slipstream Model Limitations

Several model limitations have already been mentioned in previous sections of this chapter. Understanding the slipstream model limitations is essential for deciding on an implementation strategy. This section discusses

**Table 4.3: Reverse AD derivative verification results**

<b>Output, <math>df</math></b>	<b>Input, <math>dx</math></b>	<b>Reverse AD</b>	<b>Check Deriv.</b>	$\epsilon$
<b>G</b>	Propeller location	1.0554e-03	1.0554e-03	3.8162e-05
<b>G</b>	Propeller radius	1.1271e-03	1.1271e-03	3.0164e-05
<b>G</b>	Propeller velocity distribution	3.3661e-04	3.3661e-04	1.1726e-04
<b>G</b>	Wing span	3.3661e-04	3.3661e-04	2.3568e-06
$\vec{V}$	Propeller location	8.4362e+01	8.4362e+01	4.5852e-08
$\vec{V}$	Propeller radius	7.7010e+01	7.7010e+01	5.8645e-08
$\vec{V}$	Propeller velocity distribution	9.8029e+00	9.8029e+00	1.1110e-07
$\vec{V}$	Wing span	2.5783e+01	2.5783e+01	4.9110e-07
$\vec{V}$	$V_\infty$	1.2166e+01	1.2166e+01	8.2740e-08

the limitations of the current implementation of the Rethorst correction factor.

Accuracy limitations can be attributed to the model's inability to capture slipstream contraction and deflection, and neglecting the odd solution. The slipstream is assumed to have a constant radius over the wing. Including a slipstream contraction model would improve the model's fidelity. It should then also be noted that the even correction factor is not uniform in the chordwise direction. Furthermore, the slipstream is assumed to have no deflection. In reality, the slipstream will deflect due to the angle of attack of the wing. The current model assumes the slipstream to always be perfectly tangential with the wing. Lastly, neglecting the odd correction will introduce errors. Ignoring the odd solution means that the slipstream boundary conditions are not satisfied. It should also be considered that the odd solution is non-uniform in the chordwise direction, whilst the even solution is uniform in the chordwise direction. It can therefore be speculated that adding more panels in the chordwise direction might reduce the model's accuracy if only the even solution is considered.

# 5

## Model Coupling

Model coupling is an essential step in optimisation framework development. Model coupling is achieved using the open-source Python package OpenMDAO [14]. Models can be added in OpenMDAO as subsystems, after which the in- and outputs are automatically coupled by virtue of the variable name matching. OpenMDAO exploits derivative sparsity, thus providing a highly efficient optimisation framework. After model coupling, the coupled model's derivatives can be verified. Verifying the coupled model's derivatives should be quick, assuming that the individual model's derivatives have been verified. Model coupling can however introduce errors, thus derivative verification is once again crucial. The eXtended Design Structure Matrix (XDSM) of the optimization framework is given in Figure 5.1.

This chapter discusses model coupling and how the scope of the current framework had been decreased due to issue with the linear solver. Afterwards the coupled model's derivatives are verified, followed by an explanation of the adjoint and direct method.

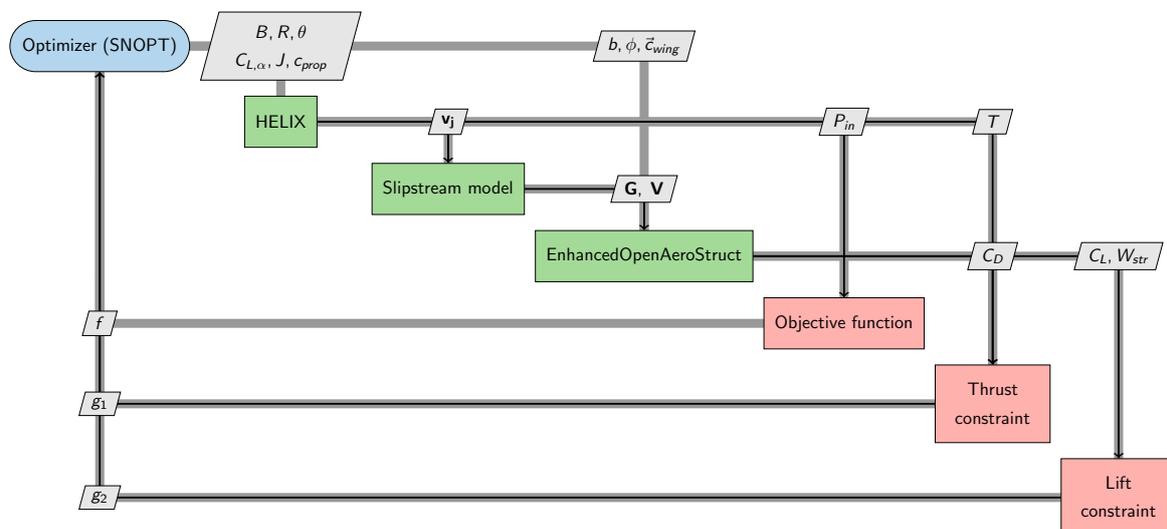


Figure 5.1: eXtended Design Structure Matrix for the coupled wing-propeller aerostructural optimization.

### 5.1. Model Coupling

The open-source Python package OpenMDAO [14] will be used to perform model coupling. OpenMDAO has various convenient built-in features to aid during the coupling procedure, especially with regard to the model's partial derivatives. The partial derivatives can either be declared explicitly, which is one of the research objectives, or by use of finite differences or complex step differences. The latter two would significantly

decrease the computational efficiency and accuracy of the model derivatives. OpenMDAO also includes an option to use finite differences to assess total derivatives, which will significantly increase computational cost and decrease derivative accuracy.

The N2 diagram of the coupled model is given in Figure 5.2. The figure below shows that there is no feedback from the wing to the propeller. Issues with the wing to propeller feedback are elaborated on in Section 5.1.1.

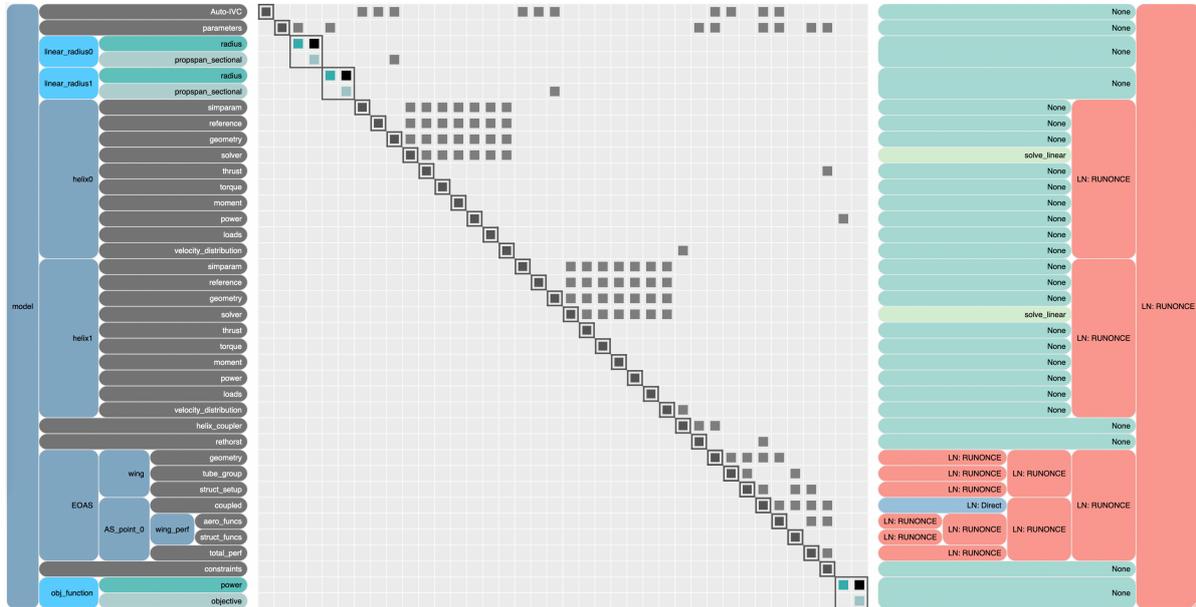


Figure 5.2: N2 diagram of the coupled wing-propeller aerostructural model

### 5.1.1. Linear Solver Converge

A significant amount of time was spent on including propeller to wing feedback by including a slipstream model, as described in Chapter 4. To include a two-way coupling, which is desirable because in subsonic aerodynamics upstream effects occur, a model should be included that calculates the induced angle of attack of the wing on the propeller. This induced angle of attack can be straightforwardly calculated since the wing VLM directly outputs the circulation of each wing panel. This circulation, calculated in OpenAeroStruct, can be coupled to a subsystem that calculates the induced angle of attack of the wing on the propeller using the propeller location and wing circulation.

Including wing to propeller feedback would result in a system represented by the N2 diagram in Figure 5.3. However, when the fully coupled system was configured it became apparent that the linear solver had serious trouble converging.

The *linear* solver in this architecture is responsible for calculating the derivatives of the system, where the *nonlinear* solver is responsible for solving the system. Whenever a cyclic dependence is present in a system a nonlinear solver is required to converge the system. The difference between a nonlinear and linear solver is graphically shown in Figure 5.4

To solve this issue various nonlinear solvers have been configured: Block Gauss-Seidel, Block Jacobi and a Krylov subspace solver plus preconditioner. The latter can provide a convenient and effective solution for large systems but choosing a preconditioner is not trivial. Without a correct preconditioner the Krylov subspace method will unfortunately also not yield any better results. A preconditioner was found that yielded better results but would still diverge in later design iterations. Due to time limitations this problem was unfortunately not solved.

It is likely that due to a large Jacobian, introduced by the fine VLM mesh which has numerous nodes, the nonlinear solver had issues converging. This problem can likely be solved by performing a more extensive research into preconditioners – which can be an entire thesis on its own [23] – or by changing the way OpenAeroStruct defines its derivatives for the mesh transformations: including additional sparsity into the

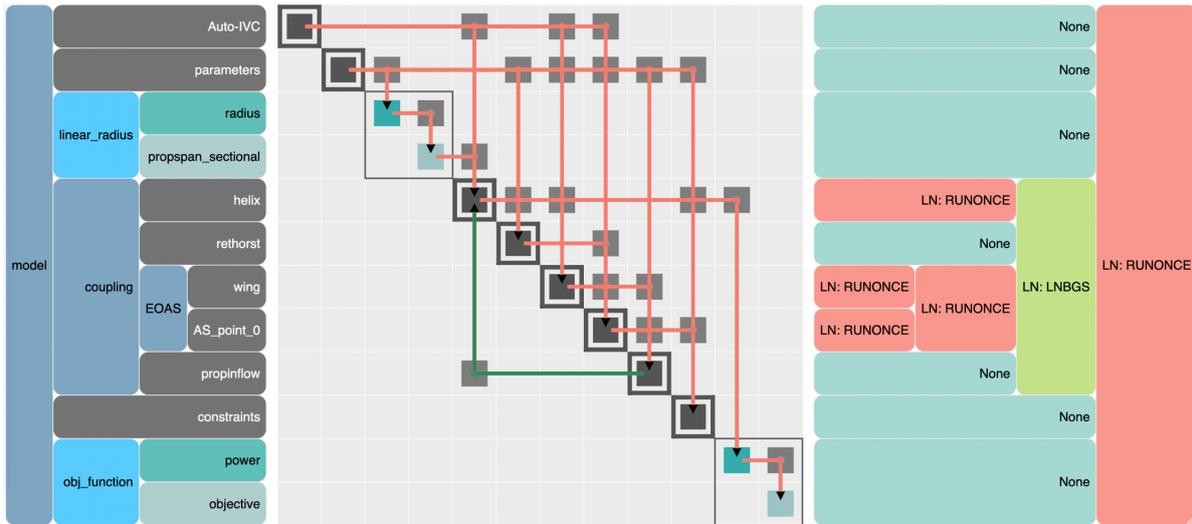


Figure 5.3: N2 diagram of fully coupled system with wing to propeller feedback

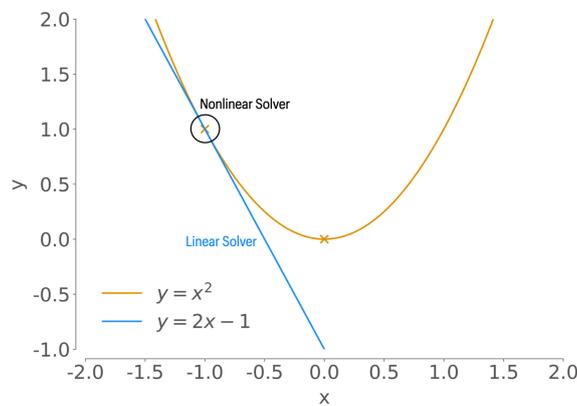


Figure 5.4: Schematic of the working of a linear and nonlinear solver

mesh sub-Jacobian might solve the convergence issue.

Figure 5.5 shows a lift distribution for a propeller with a velocity distribution that has the shape of a double parabola. The Rethorst correction is superimposed to create the velocity distribution shown, as was discussed in Chapter 4.

## 5.2. Coupled Model Validation

Model validation was carried out using the TU Delft PROWIM propeller used in studies by Sinnige [34]. The TU Delft PROWIM propeller was configured in tractor configuration in previous work by Sinnige, who collected the experimental data. A second validation procedure was performed to validate the model's ability to model wing tip propellers. The wing-propeller setup is shown in Figure 5.6.

### 5.2.1. Inboard Propeller Configuration Validation

The propeller code HELIX can be verified by comparing numerical and empirical data for the thrust coefficient versus advance ratio, shown in Figure 2.6. The validation results show a disparity between numerical and experimental values at higher advance ratios. The relatively large error at high advance ratios is likely a result of how the propeller airfoil characteristics are embedded in the BEM code. The airfoil properties are incorporated by supplying the  $C_{L,\alpha}$  slope, the zero lift angle and the angle at which stall occurs. When the airfoil enters the stall region Beard's stall model [5] is used, which is not a perfect representation of the airfoils' post-stall lift curves. The overprediction of lift in the higher advance ratio region is unfortunate, but

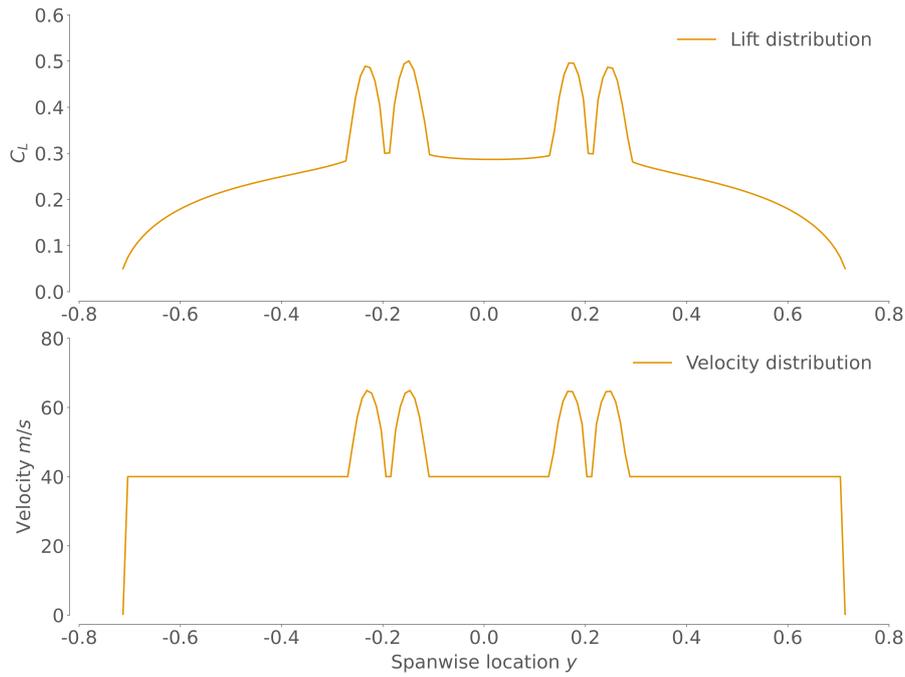


Figure 5.5: Lift distribution for a propeller with a velocity distribution that has a double parabola shape

for DEP as often seen in UAVs, the lower advance ratio is more important. Albeit that larger propellers, with lower rotational speeds, are more efficient, propeller radius is often bounded by either span or other nearby propellers. To satisfy thrust requirements, with a limited span, the propeller rotational speed must be increased, thus entering the lower advance ratio domain. This validation disparity should be considered for each optimisation process however.

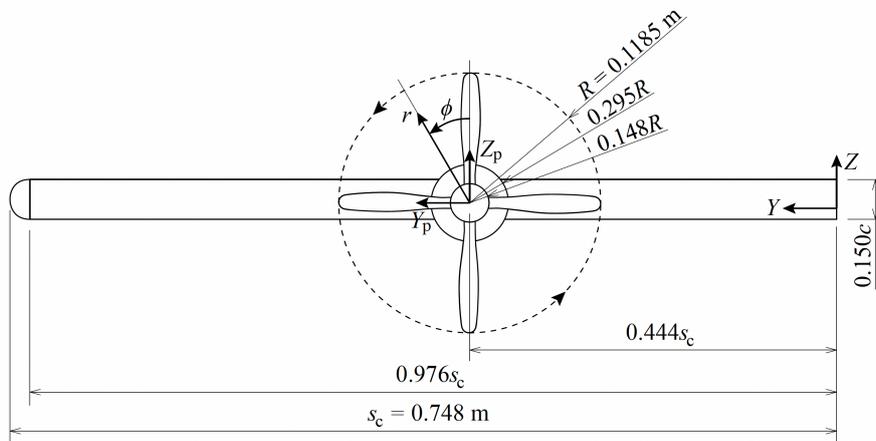


Figure 5.6: PROWIM configured in conventional tractor configuration [34]

The coupled wing-propeller data can be validated by comparing numerical and experimental data gathered by Sinnige [34], shown in Figure 5.6. The validation of the wing-propeller system is shown in Figure 5.7. The numerical and experimental data show considerable similarities. However, there seems to be a relation between an increase in advance ratio and disparity in the numerical versus experimental data. As is in line with Figure 2.6, the propeller model overpredicts thrust as the advance ratio increases. This effect is communicated to the wing model, and its disparity is visible in the middle graph in Figure 5.7. Figure 5.7 shows a comparison between the numerical solution at three different advance ratios versus the prop-off case. The prop-off wing validation data is also shown in Figure 5.7.

To conclude, the model is assumed to be reasonably accurate for optimisation, albeit that a lift over-prediction is likely to occur. An overall lift over-prediction will occur due to the limited Rethorst model. Additionally, the lift over-prediction is larger at higher advance ratios.

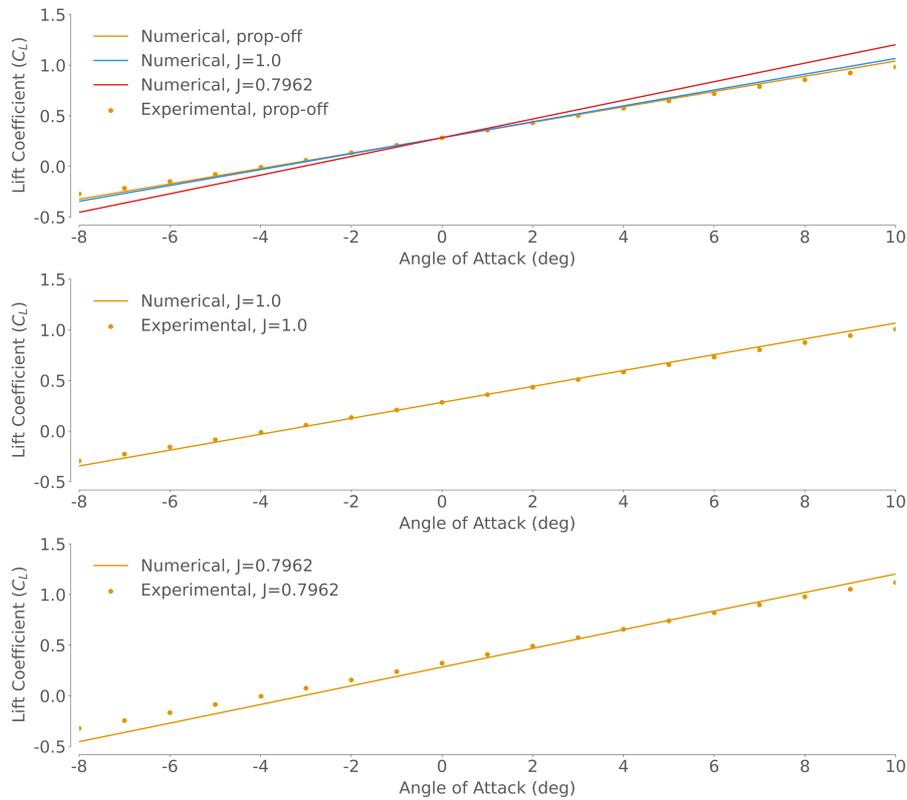


Figure 5.7: Coupled wing-propeller validation results for inboard propeller

### 5.2.2. Wingtip Propeller Validation

The wing-tip propeller configuration is shown in Figure 5.8.

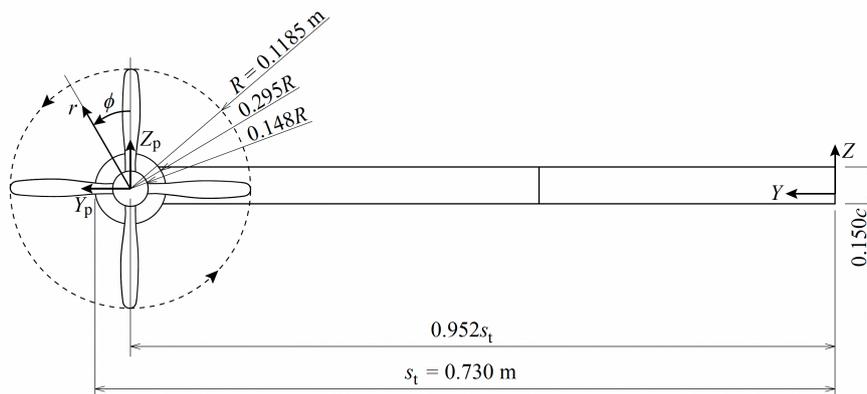


Figure 5.8: PROWIM configured in wingtip tractor configuration [34]

The wingtip propeller validation has a similar trend compared to the experimental data the inboard propeller validation. The numerical model generally over predicts lift at positive angles of attack, and under predicts at negative angles of attack. One of the limitations of the current slipstream model is its inability to account for spanwise flow components. Wingtip propellers have a substantial impact on the wing's effective

aspect ratio by wingtip vortex dispersion [34]. The wingtip vortices are pushed away from the wingtip, thus increasing the effective aspect ratio and decreasing the induced drag. It is likely that the validation for lower advance ratios is further off due to the model's inability to model spanwise flow interaction.

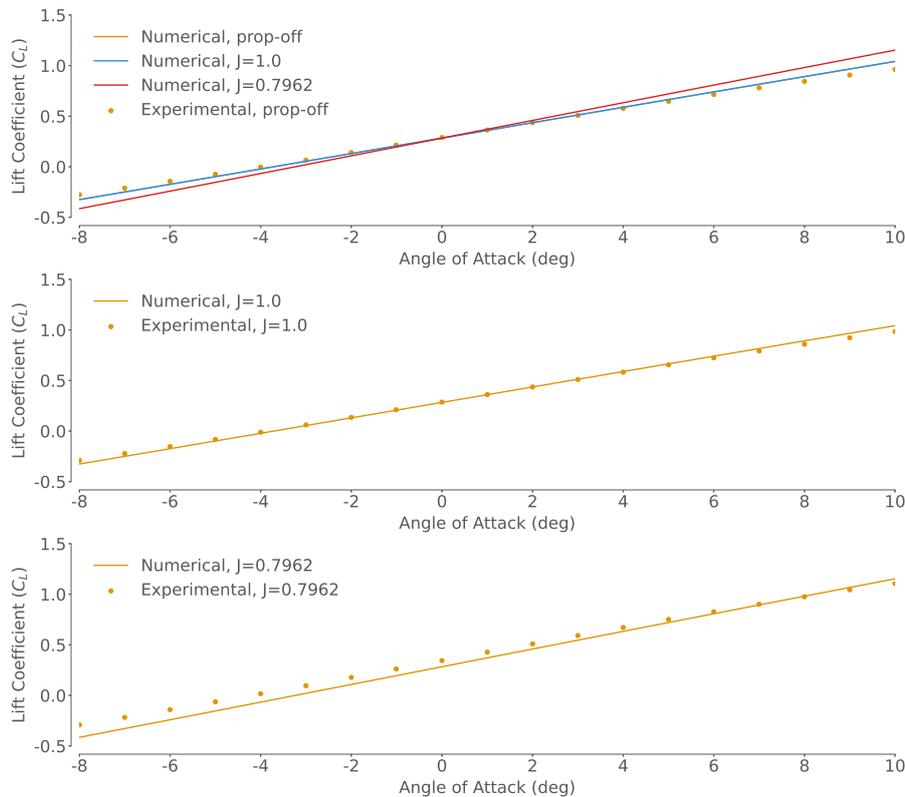


Figure 5.9: Coupled wing-propeller validation results for wingtip propeller

### 5.3. Derivative Verification

Although all model derivatives have been verified, it's important to similarly consider the *total* derivatives. Model coupling introduces room for error, and thus to guarantee that the system is working with correct derivative information, it's important to verify the total derivatives as well. OpenMDAO's `check_totals(method='central')` is used to check the total derivatives, where the central differences scheme is used to check the model derivatives.

The total derivatives that have to be checked are the output function, namely; wing failure, lift equal to weight, thrust equal to drag and the objective function propeller power with respect to the inputs propeller rotational speeds, propeller locations, propeller radii, wing chord, wing twist. As can be seen in Table 5.1 - where  $df$  is an output and  $dx$  an input - the largest relative error for the total derivatives is in the order of  $1e-4$ . The forward AD derivatives are checked with respect to central finite differences. Relative errors of this magnitude could be the result of finite computer precision and limited finite difference precision. Therefore, it can be stated that all model outputs, constraints, and objective functions derivatives with respect to the inputs are verified, meaning the architecture can be assumed mathematically correct.

### 5.4. The Adjoint and Direct Method

Implicit analytic methods can be considered the midway between finite-difference and Algorithmic Differentiation (AD) methods. Implicit analytic methods require knowledge about the governing equations and relevant state variables. One can derive a system through a continuous or discrete method. This section will solely discuss the discrete method since its often preferred over the continuous method. The preference for the discrete method stems from the fact that the resulting derivatives through the discrete method are consistent with the function values since they use the same discretisation. The continuous method could

Table 5.1: Total derivative verification results

Output, $df$	Input, $dx$	Model Deriv.	Check Deriv.	$\epsilon$
Wing structural failure	Rotational rate propeller	1.1336e-04	1.1328e-04	7.0147e-04
Wing structural failure	Propeller Twist	3.3389e-03	3.3481e-03	1.6237e-02
Wing structural failure	Wing chord	1.4834e-01	1.4834e-01	4.6858e-05
Wing structural failure	Wing twist	4.9010e-02	4.9010e-02	6.7031e-08
$L - W = 0$ constraint	Rotational rate propeller	1.6976e-03	1.6976e-03	4.9075e-06
$L - W = 0$ constraint	Propeller Twist	0.0000+00	0.0000+00	nan
$L - W = 0$ constraint	Wing chord	1.1093e-01	1.1093e-01	3.0508e-08
$L - W = 0$ constraint	Wing twist	2.7303e-01	2.7303e-01	1.5538e-09
$T - D = 0$ constraint	Rotational rate propeller	1.3703e-01	1.3703e-01	2.3585e-09
$T - D = 0$ constraint	Propeller Twist	3.2349e-02	3.2349e-02	3.2197e-07
$T - D = 0$ constraint	Wing chord	4.4190e-01	4.4190e-01	1.5092e-08
$T - D = 0$ constraint	Wing twist	1.1588e-02	1.1588e-02	3.2137e-08
Objective function	Rotational rate propeller	5.5198e+03	5.5198e+03	4.9119e-09
Objective function	Propeller Twist	5.6750e+01	5.6750e+01	5.6368e-08
Objective function	Wing chord	0.0000e+00	0.0000e+00	nan
Objective function	Wing twist	0.0000e+00	0.0000e+00	nan

lead to significant errors if the discretisation is too coarse [28].

A discretised numerical model can be written as a collection of residuals:

$$r(u; x) = 0 \quad (5.1)$$

The semicolon depicts that the design variables  $x$  are fixed whilst these equations are solved to find state variables  $u$ . The functions of interest,  $f(x, u)$ , are often explicit. However, since  $u$  is an implicit function of  $x$ ,  $f$  is an implicit function of  $x$  as well. Finding  $u$  corresponding to  $r(u; x) = 0$  requires a solver. Once  $u$  is known, the system can be solved for  $f$ . Usually, one would use either finite-differences or AD to find  $df/dx$ . Implicit analytic methods avoid calculating costly nonlinear solutions or inefficient finite-difference schemes. Ultimately, one wants to obtain the derivatives in the Jacobian given by Eq. (5.2).

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial u} \frac{du}{dx} \quad (5.2)$$

This derivative describes how state variables change due to a certain perturbation in design variables since one aims to maintain the residual equal to zero. This approach is favourable since calculating partial derivatives,  $\frac{\partial f}{\partial x}$  and  $\frac{\partial f}{\partial u}$ , is cheap and can often be done symbolically. Additionally,  $\frac{du}{dx}$  is solved by equating the residual to zero after an  $x$ -perturbation. Following this rationale, the differential form of the residual can be written as Eq. (5.3).

Eq. (5.3) can also be seen as a constraint since it forms a hypersurface, or manifold, to which the residual equation must abide. This constraint is visualised by Figure 5.10. Rewriting Eq. (5.3) gives Eq. (5.4). The latter equation is more useful since it shows that solely the partial derivatives,  $\frac{\partial r}{\partial u}$  and  $\frac{\partial r}{\partial x}$ , are required to solve the total derivative  $\frac{du}{dx}$ , instead of having to use the computationally expensive solver. The next step is to rewrite Eq. (5.2) using Eq. (5.4), giving (5.5). The partial derivatives in Eq. (5.5) can be solved using any of the previously discussed methods: symbolic differentiation, finite differences, complex step or AD.

$$\frac{\partial \mathcal{R}}{\partial u} \frac{du}{dx} = -\frac{\partial \mathcal{R}}{\partial x} \rightarrow \frac{du}{dx} = -\frac{\partial \mathcal{R}}{\partial u}^{-1} \frac{\partial \mathcal{R}}{\partial x} \quad (5.4)$$

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \frac{\partial f}{\partial u} \frac{\partial \mathcal{R}}{\partial u}^{-1} \frac{\partial \mathcal{R}}{\partial x} \quad (5.5)$$

At this point in the method, the distinction between *direct* and *adjoint* is made. The direct method will solve for the two fractions on the far right in Eq. (5.5). The direct method solves for  $\phi$  in Eq. (5.6). The adjoint method solves for  $\psi^T$  as given in Eq. (5.7). A linear solver solves the system given in Eq. (5.6) or Eq. (5.7) and returns either  $\phi$  or  $\theta^T$ , which can then be substituted into Eq. (5.5) with which the derivatives are then obtained.

$$\frac{dr}{dx} = \frac{\partial r}{\partial x} + \frac{\partial r}{\partial u} \frac{du}{dx} = 0 \quad (5.3)$$

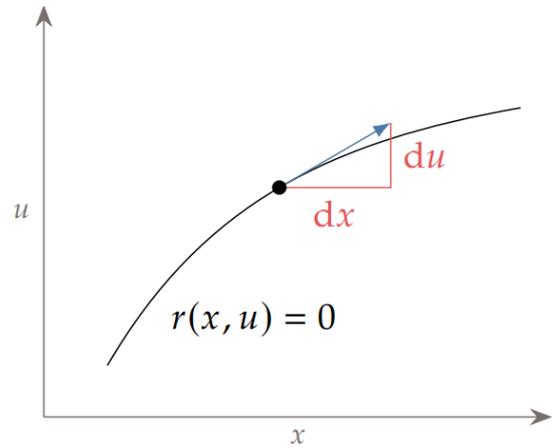


Figure 5.10: Constraint imposed on  $x$  and  $u$  by residual equation [23]

$$\frac{\partial \mathcal{R}}{\partial u} \phi = \frac{\partial \mathcal{R}}{\partial x} \quad (5.6)$$

$$\frac{\partial \mathcal{R}^T}{\partial u} \psi = \frac{\partial f}{\partial u} \quad (5.7)$$

The reason the adjoint method is worth the effort is because trying to solve Eq. (5.5) requires calculating the inverse of  $\partial r/\partial u$ . Generally, obtaining a matrix's inverse is a computationally expensive procedure. The computational overhead of the  $\partial r/\partial u$  matrix can quickly become excessive as for instance a mesh is refined or design variables are added. The direct or adjoint method do not require the inverse of the  $\partial r/\partial u$  matrix, and are thus significantly less expensive. OpenMDAO uses the Modular Analysis and Unified Derivatives (MAUD) [14]. MAUD redefines every problem in terms of a residual equation, and thus the direct or adjoint method can be used.

At this point the linear solver, discussed in Section 5.1.1, is introduced. Following the adjoint or direct method derivation, a linear system must be solved to obtain  $\psi$  or  $\phi$ . For simple systems, direct solvers can be used. Direct solvers are extremely robust and accurate. However, as often is the case in coupled systems, iterative solvers have to be used. An iterative solver introduces an error, which explains why the adjoint and direct method are less accurate than the complex step or algorithmic differentiation methods, as shown in Figure 4.18. An overview of the available linear solvers (in OpenMDAO) is given in Figure 5.11.

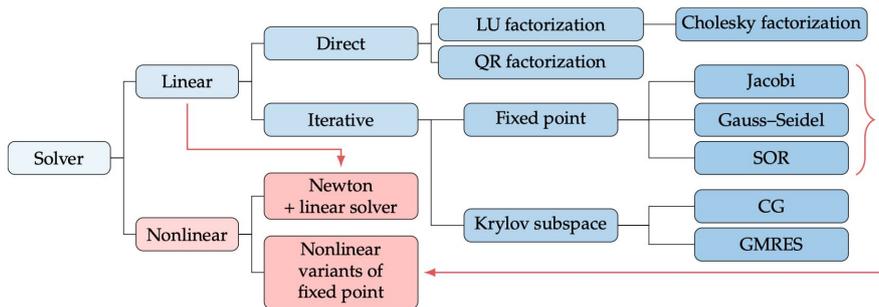


Figure 5.11: An overview of the different available linear solvers in OpenMDAO [23]

The direct- and adjoint-method are analogues to the forward- and backward-AD method. Similar to the trade-off made in AD, one needs to determine what is less costly: the direct- or adjoint-method. The difference is in the number of in- and outputs, this is graphically displayed in Figure 5.12. If the number of outputs is great than the number of inputs, the direct method is preferred. If the number of inputs is greater than the number of outputs the adjoint method is preferred.

Both methods share significant similarities, the adjoint-method is implemented much more often than the direct-method. This is due to the fact that the direct-method competes with easily implementable method

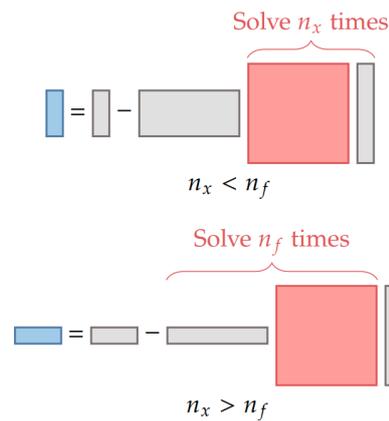


Figure 5.12: Comparison of computation cost for direct and adjoint method [23]

such as finite differences, complex step and forward-mode AD. The adjoint method only competes with backward-AD which is often plagued by a large memory requirement. To conclude, implementing implicit analytic methods is time costly and labour intensive. However, when the system depends on multiple iterative solvers, such as the propeller-wing model, implementing an implicit analytic method might be worthwhile to invest time in.

The propeller-wing optimisation architecture has fewer outputs than inputs, as can be seen in Figure 5.1, thus the adjoint method is more efficient. *Computationally efficient optimisation* also encompasses limiting the necessary amount of computer memory, which is where reverse AD could decrease the computational efficiency. However, the current model does not store significant amounts of data (in the order of Megabytes) and is thus not a limiting factor, where the optimisation speed is. The framework will be configured with both the adjoint and direct method, such that the framework can efficiently run with more outputs than inputs and the other way around. The adjoint and direct method are both expected to be much more accurate and faster than using a finite differences scheme. Furthermore, as the framework is expanded, the efficiency gains from the adjoint or direct method will increase: If the model's calculation time increases, the iteration time, from a model with finite differences, increases by the number of design variables times the time increase. For all these reasons, the adjoint and direct method are the preferred choice of derivative assessment method.

# 6

## Optimisation Results

The coupled system can, in combination with the verified forward and reverse AD codes, be used for optimisation. Currently, the model is configured to mirror the propeller and wing characteristics from the left to the right side of the wing. This was imposed since symmetry is always a requirement in aircraft design and it speeds up the optimisation process. Furthermore, the wing to propeller feedback and the propeller weight models are not included in the optimisation, as these introduced difficulties discussed in [Section 5.1.1](#).

This chapter first details isolated propeller and wing optimisation. Afterwards, the coupled system's optimisation results are discussed. The chapter concludes with an efficiency and accuracy comparison between the adjoint and direct method and an optimisation that approximates derivatives with a finite difference scheme. The optimisation parameters are given in [Table 6.1](#).

**Table 6.1: Elements of optimisation design vector,  $\vec{x}$**

Parameter	Value	Unit
Angle of Attack, $\alpha$	2	deg
Freestream velocity, $V_\infty$	40	m/s
Air density, $\rho$	1.225	kg/m <sup>3</sup>
Reynolds number, $Re$	1.0e5	-
Shear modulus, $G$	30.0e9	Pa
Young's modulus, $E$	70.0e9	Pa
Material density, $\rho_{\text{material}}$	3.0e3	kg/m <sup>3</sup>

### 6.1. Isolated Propeller Optimisation

The isolated propeller model only accounts for aerodynamic effects. The structural feasibility of the design is not considered during the optimisation. Structural integrity of a propeller blade can be roughly achieved by bounding the propeller radius, chord and airfoil shape. Initially, the chord and radius were given a lower limit that would guarantee the structural integrity of the propeller. Furthermore, the chord and radius were discretised over  $n$  spanwise elements. Interestingly enough, running an optimisation with spanwise independent sections resulted in an infeasible propeller design shown in [Figure 2.3](#). The optimiser chose to generate all (required) thrust over one section. This was done to minimise the power required. From a mathematical point this design makes sense since all thrust is created close to the hub so the torque and thus power (force  $\times$  distance) is minimised. BEM methods do not take spanwise flow components into account, which further explains why concentrating all thrust in one section is not penalised by the model since distributing the thrust, and thus pressure, loading will decrease the amount of spanwise flow.

The isolated propeller optimisation minimised the propeller power for a given thrust constraint. The optimisation formulation and results are given in [Eq. \(6.1\)](#) and [Table 6.2](#), respectively. The optimisation terminated with a feasibility in the order of 1e-15 and optimality of 1e-12, indicating tight convergence.

$$\begin{aligned} \min \quad & P_{\text{prop}} \\ \text{s.t.} \quad & T(\vec{x}) = T_0 \end{aligned} \quad (6.1)$$

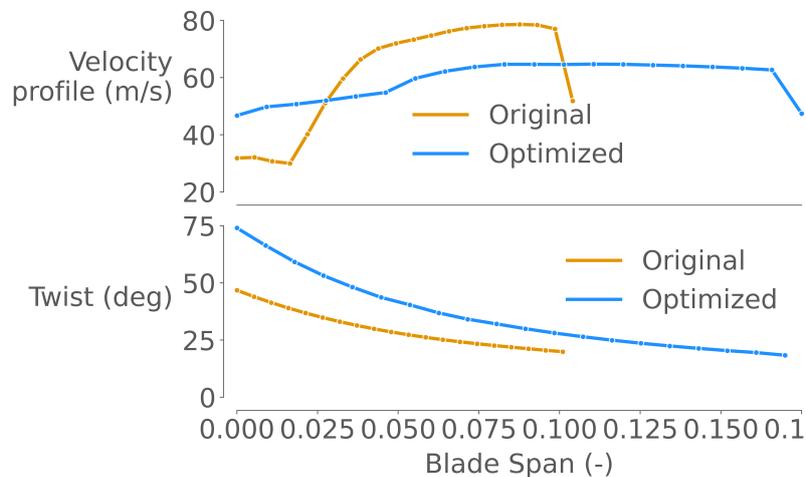
DV (units)	LB	Initial	UB	Optimised
Blade Twist (deg)	15	Figure 6.1	80	Figure 6.1
Blade radius (m)	8e-2	0.1185	2e-1	1.73e-1
Rotational speed (rad/s)	800	1320	1700	800
Objective	Initial value		Optimised value	
Power (W)	-	1534	-	1247

**Table 6.2: Propeller optimisation results**

The propeller twist optimisation returned a similar twist-curve as the original propeller, increased in value and distributed over a larger radius, which can be seen in Figure 6.1. The similarity between the original and optimised twist is expected since the PROWIM propeller design is probably the result of an optimisation process itself. The top half of Figure 6.1 shows the velocity profile of the propeller before and after optimisation. The optimiser aimed for a more evenly distributed thrust, and therefore velocity profile, which is to be expected since part of the original propeller was generating negative thrust. Do note that the twist is plotted at the start of each panel and velocity output at the centre of each panel.

The propeller optimisation created a larger, slower spinning propeller. Larger, slower spinning propellers are generally more efficient because the jet velocity approaches the freestream velocity. Mathematically, the benefit of having the jet velocity close to the freestream velocity is represented by Eq. (6.2), the equation for propulsive efficiency, since this equation approaches 1 as  $V_{jet}$  approaches  $V_0$ .

$$\eta_{propulsive} = \frac{2}{1 + \frac{V_j}{V_0}} \quad (6.2)$$



**Figure 6.1: Slipstream velocity profile and twist and radius distribution of original and optimised propeller**

To summarise, the propeller optimisation shows that larger slower spinning propellers are more efficient, as can be deduced from the propulsive efficiency equation. Furthermore, the twist distribution resembled the original twist distribution, but increased, to account for the radius increase and speed decrease.

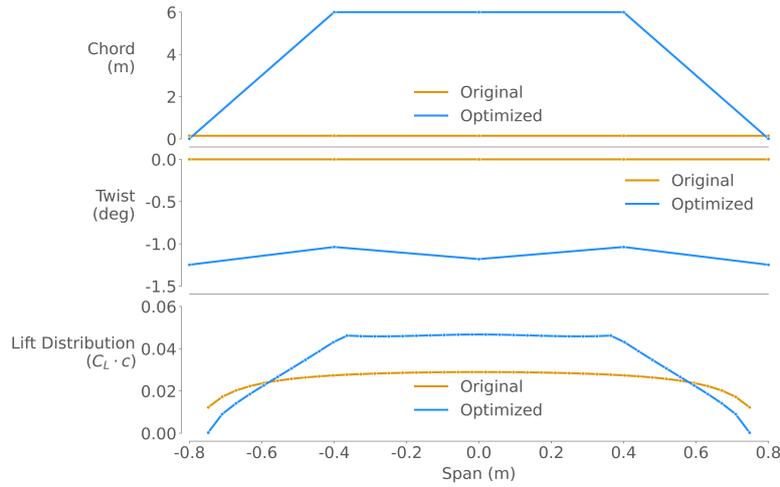
## 6.2. Isolated Wing Optimisation

Isolated wing optimisation, minimising drag for a given lift constraint, is expected to resemble an elliptical lift curve [2]. The top graph in Figure 6.2 gives the chord distribution for the optimised versus original wing, where the bottom graph shows the lift distribution. The optimiser attempts to approach an elliptical lift

distribution, therefore minimising the induced drag. The chord distribution is somewhat like one might expect from previous wing optimisation studies [16]. However, the twist distribution shows a peculiar decrease in twist at quarter span. This local twist decrease persisted in aerodynamics only optimisation. Future iterations of the framework should attempt to solve or explain this. For now it is assumed to be a mathematical artefact that has to be accounted for in the wing-propeller optimisation. The initial and optimised lift and drag are given in Table 6.3.

**Table 6.3: Wing optimisation results**

Outputs	Initial	Optimised
Lift (N)	54.8	54.8
Drag (N)	4.08e-1	3.60e-1

**Figure 6.2: Chord and twist distributions of an isolated optimised wing (top) and lift distribution (bottom)**

### 6.3. Coupled Wing-Propeller Optimisation

The coupled wing-propeller system optimisation procedure minimised the propeller power, whilst constraining thrust to be equal to drag, lift equal to weight and the structural integrity had to be guaranteed. The optimisation parameters, such as angle of attack,  $W_0$ , and freestream velocity are given in Table 6.1.

The optimisation statement is given in Eq. (6.3). The optimisation results are tabulated in Table 6.5 and both propellers have identical designs, due to mirroring one propeller to the other half of the wing. Furthermore, the wing chord was not considered as a design variable due to issues with the structural failure constraint. It was found that the tubular structural model in combination with the failure constraint did not return reliable results, and could thus not be considered for the coupled wing-propeller optimisation.

**Table 6.4: Elements of optimisation design vector,  $\vec{x}$** 

Design variable	Symbol	Unit	Size
Propeller twist	$\vec{\theta}_{\text{prop}}$	deg	$[N_{\text{prop}}, H_{\text{prop}}]$
Propeller advance ratio	$\vec{J}_{\text{prop}}$	-	$[N_{\text{prop}}]$
Wing twist	$\vec{\phi}$	deg	$[H_{\text{wing}}]$

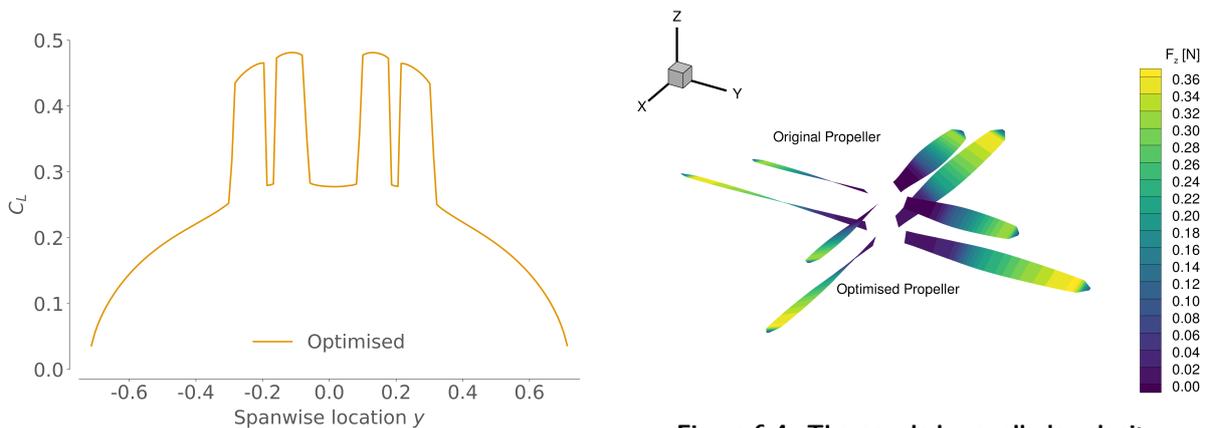
$$\begin{aligned}
 &\min P_{\text{prop}} \\
 &\text{s.t. } T(\vec{x}) - D(\vec{x}) = 0 \\
 &\quad L(\vec{x}) - W(\vec{x}) = 0 \\
 &\quad \text{Failure}(\vec{x}) < 0
 \end{aligned} \tag{6.3}$$

The failure constraint in Table 6.5 is a function integrated in OpenAeroStruct [16] that assesses whether structural integrity can be guaranteed. The failure constraint is satisfied if it is smaller than zero. The other two constraints, lift equals weight and thrust equals drag, are inherently included since the cruise phase of an aircraft is optimised and are both satisfied throughout the optimisation.

**Table 6.5: Results of the coupled wing-propeller optimisation**

DV (unit)	LB	Initial	UB	Optimised
Blade Twist (deg)	15	Figure 6.1	80	Figure 6.1
Rotational speed (rad/s)	800	1320	1700	800, 800
Wing twist (m)	-5	0	5	Figure 6.2
Results		Initial		Optimised
Failure	-	-9.43e-1	-	-9.47-1
1-Lift/Weight=0	-	5.1e-1	-	9.88e-12
1-Thrust/Drag=0	-	3.338e-1	-	-4.02e-13
Power (W)	-	2.641e3	-	2.044e3

The coupled wing propeller optimisation moved the propellers inboard whilst increasing their size and decreasing the rotational speed at which the propellers operate. One might expect the propellers to move outboard, as numerous studies suggest wing-tip propellers to be more efficient [34] [35] [39]. However, the current slipstream code accounts for axial velocity and neglects tangential velocity components: swirl. The lift increases if the propeller is moved inboard as can be seen in Figure 4.5. Including tangentially induced velocities could possibly change the outcome of the optimisation. The optimised propeller design in the coupled system is similar to the one in the individually optimised propeller. Figure 6.4 gives a comparison between the original (top) and optimised (lower) propeller thrust distribution. The coupled system's propeller turned out to be smaller than the isolated optimised propeller. This is likely due to the varying thrust constraint, since thrust is dependent on drag. The lift distribution shown in Figure 6.3 is strongly affected by the propeller influence. This is likely due to the fact that two relatively small propellers have to generate the required thrust. Furthermore, the  $W_0$  parameter was set to 20 kg, thus a significant amount of lift was required. The optimiser chose to generate the majority of the lift behind the propellers. Generating most lift behind the propeller makes sense since this section of the lift experiences the strongest lift augmentation due to propeller influence.



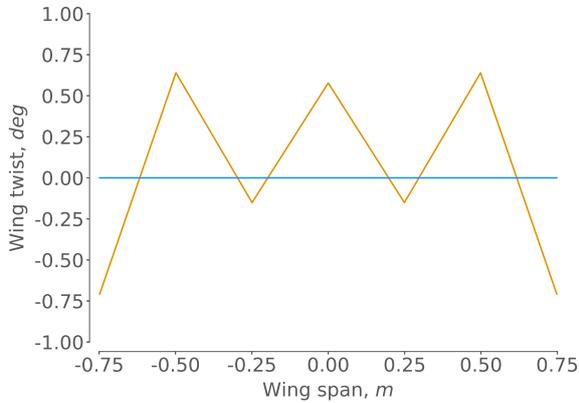
**Figure 6.3: Optimised lift distribution**

**Figure 6.4: The coupled propeller's velocity distribution (top) and twist distribution (bottom)**

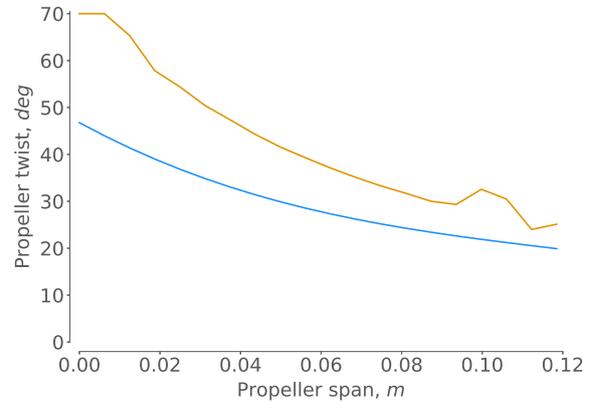
The twist distribution of the optimised wing is shown in Figure 6.5, where the propeller's optimised and original twist distributions are shown in Figure 6.6. The wing twist distribution shows some interesting behaviour since the twist values behind the propellers is smaller than the ones next to it. This is likely to limit the lift augmentation caused by the propeller. The propeller twist distribution looks less smooth as compared to the isolated propeller twist optimisation shown in Figure 6.1. The propeller interacts with

the wing however and it is suspected that the smoother twist distribution returned a less beneficial velocity distribution output. Setting a tighter convergence tolerance or changing the objective function sensitivity could affect the optimisation outcomes as well.

These optimisation results could also indicate some kind of numerical error or inaccuracy in one of the models. Furthermore, the structural tube model showed to be unreliable at times. All in all, the optimisation process shows the feasibility of coupled wing-propeller optimisation, it should however be noted that the system could benefit from additional aerodynamic and structural model improvements.

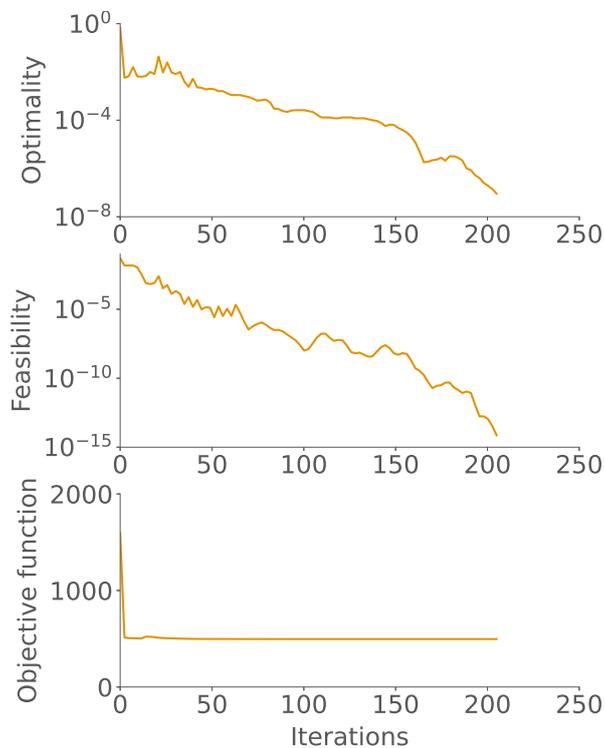


**Figure 6.5: Optimised wing twist distribution compared to baseline design, baseline in blue and optimised in orange**



**Figure 6.6: Optimised propeller twist distribution compared to baseline design, baseline in blue and optimised in orange**

The optimisation converged in 205 iterations for 25 design variables. The first order optimality, feasibility and objective function plots are given in [Figure 6.7](#).



**Figure 6.7: Optimisation results for the first 40 iterations**

## 6.4. Efficiency and Accuracy Assessment

The adjoint and direct methods are compared to an optimisation that approximates the total derivatives with a finite difference scheme. Table 6.6 summarises the efficiency by comparing the time required and the accuracy by comparing the returned feasibility and first order optimality. Table 6.6 shows how the adjoint and direct method both require less time per iteration and fewer iterations as compared to optimisation using a finite difference scheme. Furthermore, the FD optimisation was not able to converge to the same first order optimality and was therefore terminated at the 95<sup>th</sup> iteration. Furthermore, it should be noted that with the adjoint method, design variables can be added without any additional computational cost.

The adjoint method clearly shows to be a powerful derivative assessment tool as it is clearly the fastest derivative assessment method. The direct method would however be faster if the number of outputs is greater than the number of inputs. Finite differences derivatives show poor performance since the optimiser could not converge. The framework can run with both the adjoint and direct method, thus designers can add inputs or outputs without impairing the computational efficiency of the optimisation.

**Table 6.6: Efficiency and accuracy assessment of model with direct method versus finite differences. The time was averaged over 4 runs on an Intel i7 12700k**

	Adjoint Method	Direct Method	Finite Differences
First order optimality	2.0e-8	2.0e-8	2.4e-7 <b>Did not converge</b>
Feasibility	5.2E-13	5.2E-13	4.4e-12
Iterations	205	205	95
Time (s)	1872	2622	1873
Time/iteration (s)	9.13	12.8	19.7



# Methodological Requirement

Future work is needed before the framework can be deployed. The goal of this thesis was to explore the feasibility of coupled wing-propeller optimisation and propose a framework. Inherent to a novel and preliminary framework is a collection of methodological findings. This section aims to summarise information that was gathered and elaborate on recommendations and future work that can be performed to improve the accuracy and fidelity of the framework. A goal of this research project was to build a highly modular framework. Modularity was a priority such that the model is easily extendable in future stages of the project.

## 7.1. Optimisation Specific Requirements

Optimisation introduces a plethora of challenges as compared to analysis. Optimisation requires fully continuous and dynamic functions. Optimisation also requires models to be flexible and be able to change without changing the number of discretisation points. These requirements make defining a model for optimisation harder than for analysis. During this research, a number of different avenues were explored in the process of implementing the aerostructural models. Exploring model implementation methods led to the discovery of optimisation specific requirements. These requirements are summarised as methodological results in this section. A number of these requirements, such as the overset mesh, have already been discussed in the previous sections.

### 7.1.1. Remeshing Function

An important step to couple a propeller to a VLM is to develop an overset mesh. Substantial remeshing procedures are necessary without an overset mesh, or the propeller location, span, and wing-span would have to be discrete variables. The overset mesh is required to be symmetrical and should not return a noisy solution. The implementation and differentiation of the overset mesh is challenging, and has limitations that should be considered when coupling a propeller to a VLM.

It is important to consider that the remeshing procedure modifies two panels per propeller. It was experienced that this remeshing function caused an error that would accumulate throughout the optimization. Without an effective way to remesh, or prevent remeshing, the propeller location and radius can not be included as design variables. It has to be assessed whether modifying two panels, or 'distributing' the change over all VLM panels is favorable. Additionally, a sensitivity study of the mesh refinement could help determine suitable framework parameters that help the wing-to-propeller feedback loop converge.

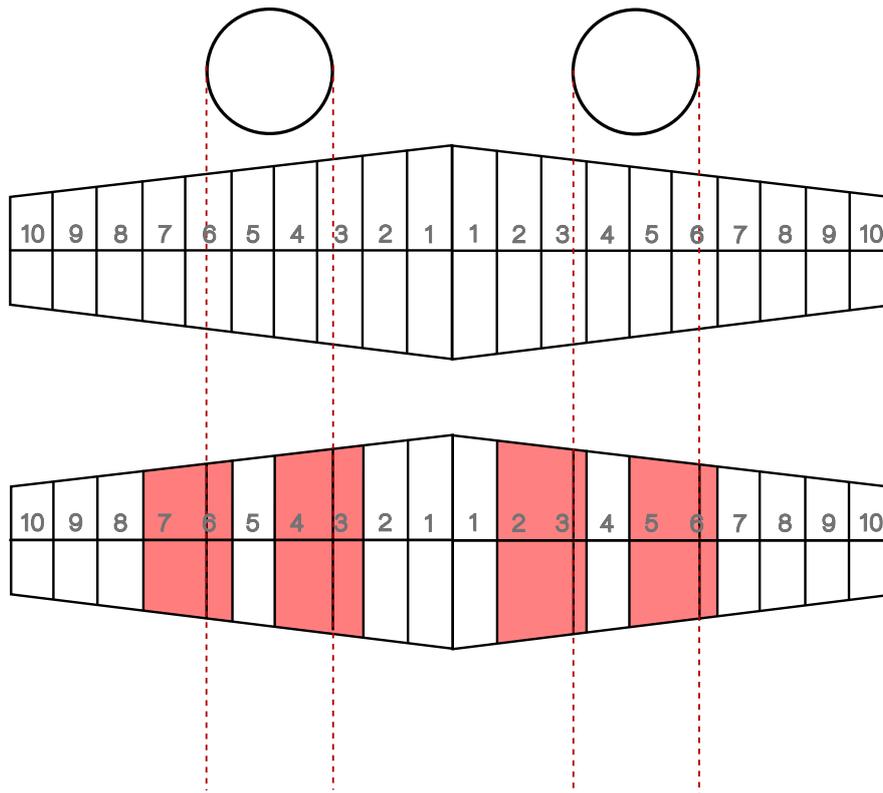
### 7.1.2. Model Symmetry

A challenge that is introduced by optimisation, as compared to analysis, is the requirement for a fully symmetric model. The symmetry of the model does not only encompass a symmetric AIC or correction matrix, as was mentioned in Chapters 4 and 3. The symmetry of the correction matrix was guaranteed by mirroring the upper left quarter solution to the right top, the bottom left and the bottom right quarters of the correction matrix.

The remeshing function, included in OAS and Rethorst, has an important role in the model symmetry

requirement. If a model is not symmetrical from the left to the right side of the wing, analysis will return correct results but the AD'd code will have asymmetrical derivatives. An earlier version of the remeshing function returned asymmetrical derivatives between the left and right wing. Do note that the derivatives were verified, as the AD'd code was correct. However, the way the model was defined was not suitable for optimisation.

The asymmetry occurred because the remeshing function favoured moving panels to the left on both sides of the wing. The derivatives were corrupted by the way the model was defined. The asymmetry resulted in the optimiser favouring moving the propellers to the left. Figure 7.1 shows how a different panel is affected by consistently moving VLM to the left. Figure 7.1 exaggerates this effect, if the panels are small the effect will be less severe and the model is symmetric, for analyses purposes.

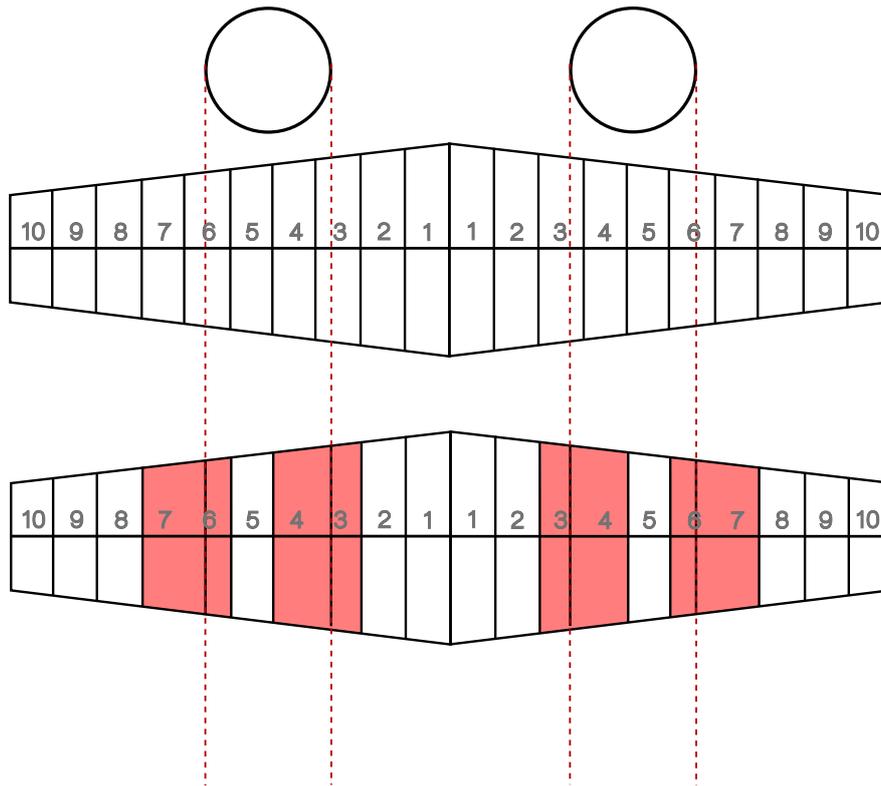


**Figure 7.1: Schematic of how an early version of the Rethorst correction was defined, this configuration resulted in asymmetric derivatives for the AD'd code**

Figure 7.2 shows how the current remeshing function works. The current remeshing function returns fully symmetric derivatives for propellers on either the left or right side of the wing. Furthermore, the correction factor and the velocity vector that were returned for propellers on both sides of the wing share this symmetric property. The spacing function, that scales the correction factor to accommodate for mesh refinement differences between the VLM and overset mesh also accounts for this change in modelling.

## 7.2. Recommendations

Inaccuracies in the models, and simplifying assumptions, occur due to neglecting the odd solution of the slipstream model and the propeller's validation results at higher advance ratio. These two issues should be addressed if one wishes to improve the accuracy of the models. The full slipstream correction factor could possibly be implemented by using D.E. Amos 644 algorithm for Bessel functions [1]. Furthermore, MPI parallelism, as currently configured for HELIX and OAS, could be applied to the slipstream code to leverage parallel computing, and increase the system's efficiency. Lastly, a number of aerodynamic effects, such as tangential flow and slipstream contraction are missing. Including the missing aerodynamic phenomena may improve the system's accuracy and possibly yield different optimization results.



**Figure 7.2: Schematic of the version of the Rethorst model that has symmetric derivatives**

### 7.2.1. Odd Rethorst Correction

The fidelity of the simulation can be improved by including the odd solution of the Rethorst correction factor. As was mentioned in [Chapter 4](#), the odd solution is a computationally expensive function. The odd solution is expensive due to the double nested integral containing Bessel functions. D.E. Amos [1] describes a computationally efficient algorithm to calculate Bessel function of the first and second order. It should be noticed that the Bessel functions' derivatives are also required for the Rethorst correction. The forward AD procedure can be used to obtain these. A comparison between the full and only even correction factor is shown in [Figure 7.3](#).

All in all, this thesis does produce the Fortran code with which the odd correction can be calculated. The algorithmic derived code (forward and reverse) have yet to be generated and verified for the odd solution.

### 7.2.2. Nonuniform Propeller Inflow Field

Although it could not be included in the coupled optimisation procedure, the current wing to propeller feedback function assumes the entire propeller experiences the same upwash. In reality, the upwash angle changes over the propeller radius because of the varying distance to the wing. Van Arnhem et al. [4] developed a mid-fidelity engineering method to assess propeller performance in a non-uniform inflow field. Van Arnhem's model is a straightforward way to increase the fidelity of the propeller-wing model, by including more accurate feedback from the wing to the propeller.

Van Arnhem's method starts by assuming a uniform inflow field, and thus advance ratio  $J_\infty$  and calculates the loading across the blade. Van Arnhem's method approaches the deviation in local advance caused by the wing induced velocities. The method attempts to estimate the local change in thrust and torque,  $dT$  and  $dQ$ , respectively. The method calculates the local torque and thrust deviation by assuming the local advance ratio, for one specific blade element, is experienced over the entire blade. After converging the iterative BEM solver the thrust at that blade section is compared to the propeller's thrust that was returned for  $J_\infty$ . [Figure 7.4](#) represents van Arnhem's method. The difference in torque and thrust loading can clearly be seen as a result of the local change in advance ratio. The optimisation process would benefit from this model since it increases the amount of wing to propeller feedback. Furthermore, the model requires the propeller

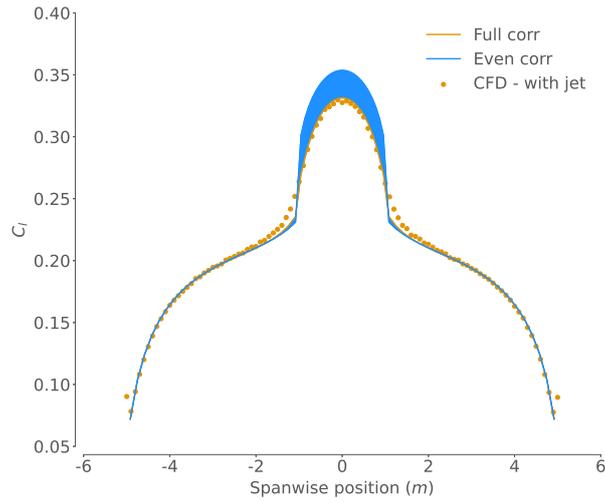


Figure 7.3: Full versus even correction factor solution, CFD data taken from Nederlof [25]

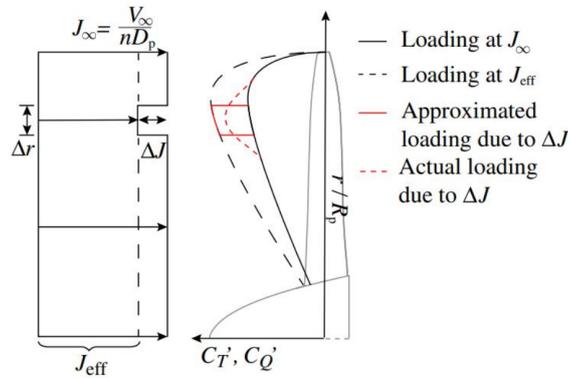


Figure 7.4: Van Arnhem et al. engineering method for assessing propeller performance in a non-uniform inflow field [4]

model to converge several times. Therefore the adjoint method will prove extra efficient, as compared to non-intrusive methods or direct AD, since an iterative system has to be converged.

### 7.2.3. Slipstream Contraction

Willemsen [39] uses a slipstream model that is based on the slipstream tube model [24, 36], in combination with the Rethorst correction factor. The slipstream is discretised in all three dimensions. The induced velocity at point P is affected by three sources of vorticity: axial, tangential and propeller disk vorticity. Willemsen expanded the tube model by including deflection and contraction models. Assuming that a propeller model is used that can account for non-uniform inflow: the propeller model provides the azimuthal circulation distribution.

Slipstream contraction occurs due to the increase in dynamic pressure, as a way to conserve the continuity of mass flow. Assuming the flow to be incompressible, Equation (7.1) can be used to calculate the slipstream radius at any axial position. It is computationally expensive to calculate this for every axial position since  $u_a$  has to be known.

$$\frac{r}{r_0} = \sqrt{\frac{U_\infty + u_{a,0}}{U_\infty + u_a}} \quad (7.1)$$

Conway's [10] analytical solution is less computationally expensive and can therefore be used instead of Eq. (7.1). Conway's analytical solution can be superimposed to form a radial velocity distribution, much like the super imposing of the Rethorst model. Eq. (7.2) gives the mathematical formulation of Conway's

solution. Important to note is that the axial velocity distribution at the propeller has to be represented by using eight polynomials, as determined by Willemssen. The coefficients,  $U_{a0,\mu}$  can be determined by using a least squares method.

$$U_a(r, 0) = \sum_{\mu=1}^N U_{a0,\mu} \left(1 - \frac{r^2}{R^2}\right)^\mu \quad (7.2)$$

Figure 7.5 shows a comparison between the Conway model and slipstream tube model. Figure 7.5 shows good agreement between Conway's model and the slipstream tube model, implemented by Veldhuis [36]. The mean slipstream contraction, at  $r/R = 0.57$ , follows the tube contraction model well. Conway's analytical solution also shows good agreement with a frozen vortex model [39]. This model is therefore suitable for future implementation into the optimisation architecture. Furthermore, the model is easily differentiable, and does not contain discontinuities, and thus the symbolic derivatives can be obtained.

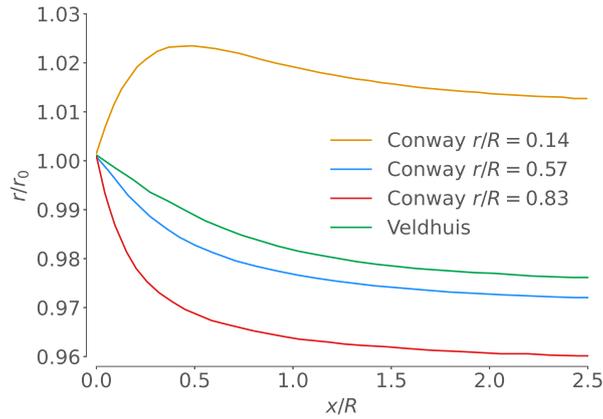


Figure 7.5: Slipstream contraction for different radial stations as compared to Veldhuis [10, 36, 39]

Including this model will reduce the effect of the slipstream on the wing. The slipstream's effect will decrease since the slipstream radius will decrease. Therefore, adding panels in the chordwise direction will have a larger effect on the model's output.

#### 7.2.4. Slipstream Deflection

The slipstream deflects due to externally induced vertical velocities. The current slipstream model does not take this deflection into account. The deflection angle changes in the axial direction since the flow velocity will change as well. With a constant axial velocity the deflection would be overestimated, since the deflection angle decreases with decreasing axial velocity. Including a deflection model will emphasise the importance of the propeller distance in axial direction to the wing. A schematic of slipstream deflection is given in Figure 7.6.

#### 7.2.5. Tangential Velocity

Wing tip propellers could increase the aerodynamic efficiency of a wing due to wing tip vortex dispersion [34, 36]. Wing tip vortex dispersion is an effect of propeller swirl interacting with the wing tip vortices. The wing tip swirl is 'reduced' by the propeller swirl, assuming the propeller is rotating inboard up. Currently, no mid-fidelity models have been identified that can accurately and computationally efficiently model swirl or wing tip vortex dispersion. It should be considered that the model might have to affect the wing's aspect ratio, since pushing the wing tip vortex outboard increases the effective wing aspect ratio.

It is expected that including a swirl model will favour moving the propeller outboard, where currently the optimiser chooses to move the propeller inboard, due to the lift increase caused by moving the propeller inboard.

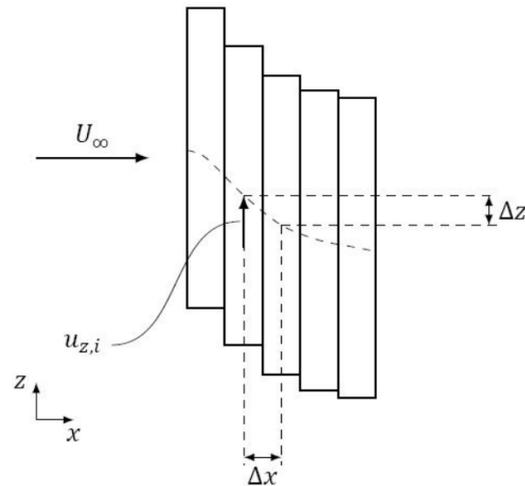


Figure 7.6: Schematic of slipstream deflection [39]

### 7.2.6. AeroAcoustics

An advantage of the current framework is that it is highly modular and easily extendable. Acoustics have become an increasingly important topic due to possible community annoyance and noise pollution due to UA [3, 17]. NASA considers acoustics to be one of the ten fields that require technological development before UA can be developed on a large scale [17]. Furthermore, it is no secret that noise pollution has been one of the main worries around expanding airports in terms of size and number of flights. For all these reasons, acoustic optimisation has substantial importance.

Pulse [26] is a mid-fidelity aeroacoustics code that can easily be implemented in the current framework. Rotating blade noise can be divided into tonal and broadband noise. Tonal noise is periodic where broadband noise is more consistent. Acoustics can be modelled by using direct numerical simulation, which accurate but expensive. The Ffowcs Williams and Hawkins Model (FWH) [40] model is based on the Navier Stokes equations with Lighthill's analogy [32] with surface integrals over monopole and dipole noise sources. The FWH model is extensively used and validated for rotorcraft applications and used by the Pulse acoustics code.

### 7.2.7. Mesh Warping

It was found that due to the remeshing function in OAS, the mesh would slowly deform in its entirety. In other words, more than the 4 panels per propeller were affected by the propeller moving in spanwise direction and change in radius. The propeller radius and location were removed as design variables for this reason, as these variables would deform the mesh. The increasingly deformed mesh was effectively accumulating an error, that caused the optimiser to diverge around the 20<sup>th</sup> iteration.

### 7.2.8. Structural Model

The current structural model uses a tubular spar model. The tubular spar model can only adjust its thickness and misses the fidelity of a wingbox model. OAS has a wingbox implementation [6]. However, the wingbox requires the symmetry condition within OAS to be set to true, which currently can not be done due to the Rethorst correction output. Rewriting the correction matrix for AIC symmetry conditions is discussed in Section 7.2.9

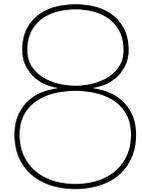
### 7.2.9. Computational Efficiency Correction Matrix

Currently the Rethorst correction is calculated for the entire wing. Calculating the correction for the entire wing is necessary due to the correction factor varying between the left and right half of the wing, for a given propeller on the left or right side. The Rethorst correction can be rewritten to assume a symmetric wing and return the correction matrix for half the wing. It should be noted that the correction matrix will still be

calculated but only half the correction matrix is given to OAS. This half correction matrix is then added to half the AIC matrix, thus using the AIC symmetry condition and increasing computational efficiency.

#### 7.2.10. Wing to Propeller Feedback

Wing to propeller feedback can be included by solving the issues around the linear solver. The linear solver, as discussed in Section 5.1.1, solves the partial derivatives matrix of the residual with respect to the  $u$  parameters. The  $\mathcal{R}/u$  matrix is necessary for the adjoint method. The subsystem, including derivatives, for the wing to propeller feedback is included in the current framework. Therefore, understanding and solving the linear solver issues will open the door for wing to propeller feedback.



## Conclusion

This masters dissertation presents a novel framework with which computationally efficient aerostructural coupled wing-propeller optimisation can be performed. The framework couples a BEM code named HELIX to a slipstream code, that eventually feeds into a wing model. The outputs of the wing and propeller (BEM) model eventually return the objective and constraint functions. The framework has a working and verified adjoint. With the adjoint, design variables can be added without any additional computational cost.

The propeller and wing model development required little modifications as most of the code was readily available, including forward and reverse algorithmically differentiated code. The propeller model required the addition of a velocity distribution code. The main limitation of the propeller model is its inability to model spanwise flow components. It was discovered that the model's inability to predict spanwise flow prevented the chord from being included as a design variable. Therefore, the propeller design variables that can be optimised are the radius, rotational velocity and twist.

The wing model required three modifications. A remeshing function was included to guarantee propeller slipstream and VLM panel alignment. Furthermore, the correction factor and velocity vector, both returned by the slipstream model, had to be incorporated into the VLM model. However, it was found that the remeshing function in the wing aerostructural solver, OAS, performed worse than expected. Its poor performance was due to mesh errors accumulating, as it was initially assumed that the mesh would be reset at every optimisation iteration. However, since the mesh was not reset each iterations the VLM mesh would slowly distort. At a certain point in the optimisation process, often around 20 iterations, the system would diverge and crash. Therefore it can be concluded that the remeshing function should be revisited. Another important point of attention is the model symmetry requirement.

Optimisation introduces an additional set of challenges as compared to analysis, and thus changes the slipstream model's implementation method. Previous implementations of the Rethorst correction factor on a VLM mesh [20, 25, 39] contained discontinuities and could therefore not be used for optimisation. Additionally, the slipstream model has several requirements that the method implementation has to satisfy: The Rethorst slipstream model must align a VLM panel control point with the propeller centre, the propeller must be in the centre of the wing and the slipstream boundaries should be aligned with other VLM panels as well. The latter requirement was addressed by the remeshing function included in the wing model. It was found that an overset mesh offers a solution for the remaining two requirements. The correction factor was calculated for the overset mesh and interpolated from the overset mesh to the VLM mesh. Furthermore, a first iteration of the slipstream model contained asymmetries. For analysis, subtle asymmetries are negligible, especially when the mesh gets refined. However, for optimisation purposes asymmetries could be fatal. An asymmetry in the partial derivative code could lead to asymmetric total derivatives. Asymmetric derivatives could lead the optimiser to wrong results, by for instance moving one propeller to the left and the other to the right. To conclude, the current slipstream model is continuous and does not have any asymmetries and can therefore be used for optimisation.

The accuracy of the propeller, wing and slipstream models is somewhat limited for this first version of the framework. The propeller validation showed that the model was good at predicting propeller performance

at lower advance ratios, where there was some disparity at higher advance ratios. The coupled propeller-wing system similarly showed some disparities between the experimental and numerical data, especially at the higher angles of attack and advance ratios, i.e. above and below 5 and -5 degrees angle of attack, respectively. However, since this is an optimisation study the models were considered reasonably accurate. Future iterations of the framework can benefit from more accurate propeller, slipstream and wing models.

The isolated propeller optimisation, optimising for minimum shaft power, results show a slower rotating propeller with a twist distribution that adjusts for this lower rotational velocity. The propeller twist distribution adapted in such a way that the thrust over drag performance was optimised for each blade section. The isolated wing optimisation, optimising for minimum drag of a given lift constraint, returned a wing with a somewhat unexpected twist distribution. However, the optimised wing lift distribution did resemble an elliptical distribution more than the baseline lift distribution, which is as expected. The coupled system decreased the wing twist behind the propeller, likely due to slightly reduce the effect of the local lift augmentation, which is considerably strong. The propeller model had trouble approaching the isolated propeller optimised design. The efficiency and accuracy assessment did show that the adjoint method is substantially faster than the direct method and an optimisation process that approximates the total derivatives with finite differences. The adjoint and direct method did score the same in terms of accuracy, which is as expected. All in all, the coupled optimisation process showed the feasibility of coupled wing-propeller optimisation. The optimisation results also show that the framework could benefit from current model improvement and additional mid-fidelity aerostructural models.

The framework shows the power of intrusive gradient assessment methods and proves the feasibility of wing-propeller optimisation. The current coupled framework does not return significantly different propeller or wing design as compared to isolated propeller or wing optimisation. The propeller model decreased the rotational speed as much as possible and adjusted the twist to account for the decrease in rotational velocity. The wing model had the peculiar tendency to lower the twist on the second section, as counted from the wing twist. This was found in the isolated and coupled optimisation. It has yet to be assessed whether this was due to the aerodynamic or structural model, or their combination. It is hypothesised that including additional models that take into slipstream contraction, deflection or spanwise flow components could yield different propeller and wing designs by modelling more comprehensive interaction between the two systems. Although the implementation of the adjoint and direct method are non-trivial, it is clear that these methods provide substantial efficiency and accuracy gains, since the system using finite differences could not converge. Furthermore, the adjoint method took less than half the time for one iteration than the system using finite differences. The direct and adjoint method both converge to the exact same results, proving the consistency of the provided forward and reverse AD code.

To conclude, the study provides a robust wing-propeller framework that converges consistently and well. The slipstream model communicates axial effects from the propeller to the wing model. Furthermore, adding aerodynamic models could substantially improve the framework's accuracy. As of now the framework performs well in terms of computational efficiency. To sustain this computational efficiency it is essential to provide forward and reverse AD code with any model modifications or additions. The framework is an important first step towards computationally efficient wing-propeller optimisations that can easily be extended.

# References

- [1] D. E. Amos. "Algorithm 644: A Portable Package for Bessel Functions of a Complex Argument and Nonnegative Order". In: 12.3 (1986). ISSN: 0098-3500. DOI: [10.1145/7921.214331](https://doi-org.tudelft.idm.oclc.org/10.1145/7921.214331). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/7921.214331>.
- [2] John D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, 1991.
- [3] K R Antcliff et al. "Baseline Assumptions and Future Research Areas for Urban Air Mobility Vehicles". In: *AIAA Scitech 2019 Forum*. San Diego, California, 2019, pp. 1–18. DOI: [10.2514/6.2019-0528](https://doi-org.tudelft.idm.oclc.org/10.2514/6.2019-0528).
- [4] Nando van Arnhem et al. "Engineering method to estimate the blade loading of propellers in nonuniform flow". In: *AIAA Journal* 58.12 (2020), pp. 5332–5346.
- [5] R. W. Beard and T. W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.
- [6] Shamsheer S Chauhan and Joaquim RRA Martins. "Low-fidelity aerostructural optimization of aircraft wings with a simplified wingbox model using OpenAeroStruct". In: *International Conference on Engineering Optimization*. Springer, 2018, pp. 418–431.
- [7] Shamsheer S. Chauhan and Joaquim R. R. A. Martins. "Low-Fidelity Aerostructural Optimization of Aircraft Wings with a Simplified Wingbox Model Using OpenAeroStruct". In: *Proceedings of the 6th International Conference on Engineering Optimization, EngOpt 2018*. Lisbon, Portugal: Springer, 2018, pp. 418–431. DOI: [10.1007/978-3-319-97773-7\\_38](https://doi-org.tudelft.idm.oclc.org/10.1007/978-3-319-97773-7_38).
- [8] Matthew A Clarke et al. "Aerodynamic Optimization of Wing-Mounted Propeller Configurations for Distributed Electric Propulsion Architectures". In: *AIAA AVIATION 2021 FORUM*. 2021, p. 2471.
- [9] Julia A Cole et al. "On the Integrated Aerodynamic Design of a Propeller-Wing System". In: *AIAA Scitech 2019 Forum*. 2019, p. 2300.
- [10] John T Conway. "Analytical solutions for the actuator disk with variable radial distribution of load". In: *Journal of Fluid Mechanics* 297 (1995), pp. 327–355.
- [11] R. de Vries. "Hybrid-Electric Aircraft with Over-the-Wing Distributed Propulsion: Aerodynamic Performance and Conceptual Design". English. PhD thesis. Delft University of Technology, 2022. DOI: [10.4233/uuid:ef87dc11-e7b2-4726-a41f-28588d64c58d](https://doi-org.tudelft.idm.oclc.org/10.4233/uuid:ef87dc11-e7b2-4726-a41f-28588d64c58d).
- [12] Horia Dumitrescu and Vladimir Cardos. "Inboard stall delay due to rotation". In: *Journal of aircraft* 49.1 (2012), pp. 101–107.
- [13] Philip E Gill, Walter Murray, and Michael A Saunders. "SNOPT: An SQP algorithm for large-scale constrained optimization". In: *SIAM review* 47.1 (2005), pp. 99–131.
- [14] Justin S. Gray et al. "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization". In: *Structural and Multidisciplinary Optimization* 59.4 (Apr. 2019), pp. 1075–1104. DOI: [10.1007/s00158-019-02211-z](https://doi-org.tudelft.idm.oclc.org/10.1007/s00158-019-02211-z).
- [15] Laurent Hascoet and Valérie Pascual. "The Tapenade Automatic Differentiation Tool: Principles, Model, and Specification". In: 39.3 (2013). ISSN: 0098-3500. DOI: [10.1145/2450153.2450158](https://doi-org.tudelft.idm.oclc.org/10.1145/2450153.2450158). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/2450153.2450158>.
- [16] John P. Jasa, John T. Hwang, and Joaquim R. R. A. Martins. "Open-source coupled aerostructural optimization using Python". In: *Structural and Multidisciplinary Optimization* 57.4 (Apr. 2018), pp. 1815–1827. DOI: [10.1007/s00158-018-1912-8](https://doi-org.tudelft.idm.oclc.org/10.1007/s00158-018-1912-8).
- [17] Wayne Johnson, Christopher Silva, and Eduardo Solis. "Concept vehicles for VTOL air taxi operations". In: *AHS Specialists' Conference on Aeromechanics Design for Transformative Vertical Flight*. ARC-E-DAA-TN50731. 2018.

- [18] C Koning. "Influence of the propeller on other parts of the airplane structure". In: *Aerodynamic Theory*. Springer, 1935, pp. 361–430.
- [19] Heyecan Koyuncuoglu and Ping He. "Coupled Wing-propeller Aerodynamic Optimization Using the Adjoint Method". In: *AIAA SCITECH 2022 Forum*. 2022, p. 0015.
- [20] Quinty van der Leer. "A review of the wingtip-mounted propeller configuration". MA thesis. Delft University of Technology, 2020.
- [21] Trent W Lukaczyk et al. "SUAVE: an open-source environment for multi-fidelity conceptual vehicle design". In: *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2015, p. 3087.
- [22] Joaquim R. R. A. Martins, Ilan M. Kroo, and Juan J. Alonso. "An Automated Method for Sensitivity Analysis Using Complex Variables". In: *Proceedings of the 38th AIAA Aerospace Sciences Meeting*. AIAA 2000-0689. Reno, NV, Jan. 2000.
- [23] JRRR Martins and Andrew Ning. *Engineering design optimization*. Cambridge University Press, 2021.
- [24] Luis Miranda and J Brennan. "Aerodynamic effects of wingtip-mounted propellers and turbines". In: *4th Applied Aerodynamics Conference*. 1986, p. 1802.
- [25] Robert Nederlof. "Improved modeling of propeller-wing interactions with a lifting-line approach: Investigation of a suitable correction method to account for the finite slipstream height". In: (2020).
- [26] Bernardo Pacini et al. "Towards Efficient Aerodynamic and Aeroacoustic Optimization for Urban Air Mobility Vehicle Design". In: *AIAA AVIATION 2021 FORUM*. 2021, p. 3026.
- [27] G. Pesare. "Modeling of unsteady vortex lattice method for wing flutter control in OpenMDAO". MA thesis. Sapienza - Universita di Roma, 2016.
- [28] Jacques EV Peter and Richard P Dwight. "Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches". In: *Computers & Fluids* 39.3 (2010), pp. 373–391.
- [29] Ramadas K Prabhu. "Studies on the interference of wings and propeller slipstreams". In: (1984).
- [30] Daniel Raymer. *Aircraft design: a conceptual approach*. American Institute of Aeronautics and Astronautics, Inc., 2012.
- [31] Scott Rethorst. "Aerodynamic of Nonuniform Flows as Related to an Airfoil Extending Through a Circular Jet". In: *Journal of the Aerospace Sciences* 25.1 (1958), pp. 11–28.
- [32] RH Self. "Jet noise prediction using the Lighthill acoustic analogy". In: *Journal of sound and vibration* 275.3-5 (2004), pp. 757–768.
- [33] S Sethi et al. "On using the Kreisselmeier-Steinhauser function in simultaneous analysis and design". In: *38th Structures, Structural Dynamics, and Materials Conference*. 1997, p. 1289.
- [34] Tomas Sinnige. "Aerodynamic and Aeroacoustic Interaction Effects for Tip-Mounted Propellers: An Experimental Study". PhD thesis. Delft University of Technology, 2018.
- [35] Quinty Van der Leer and Maurice Hoogreef. "Aero-Propulsive and Aero-Structural Design Integration of Turboprop Aircraft with Electric Wingtip-Mounted Propellers". In: *AIAA SCITECH 2022 Forum* (2022), p. 0167.
- [36] Leonardus Louis Maria Veldhuis. "Propeller wing aerodynamic interference". In: (2005).
- [37] LLM Veldhuis and PM Heyma. "Aerodynamic optimisation of wings in multi-engined tractor propeller arrangements". In: *Aircraft design* 3.3 (2000), pp. 129–149.
- [38] J Weissinger. *The lift distribution of swept-back wings*. Tech. rep. 1947.
- [39] Roger Willemsen. "A Sensitivity Study on the Aerodynamic Performance of a Wingtip-Mounted Tractor Propeller-Wing System". MA thesis. Delft University of Technology, 2020.
- [40] JE Ffowcs Williams. "Sound sources in aerodynamics-Fact and fiction". In: *AIAA Journal* 20.3 (1982), pp. 307–315.