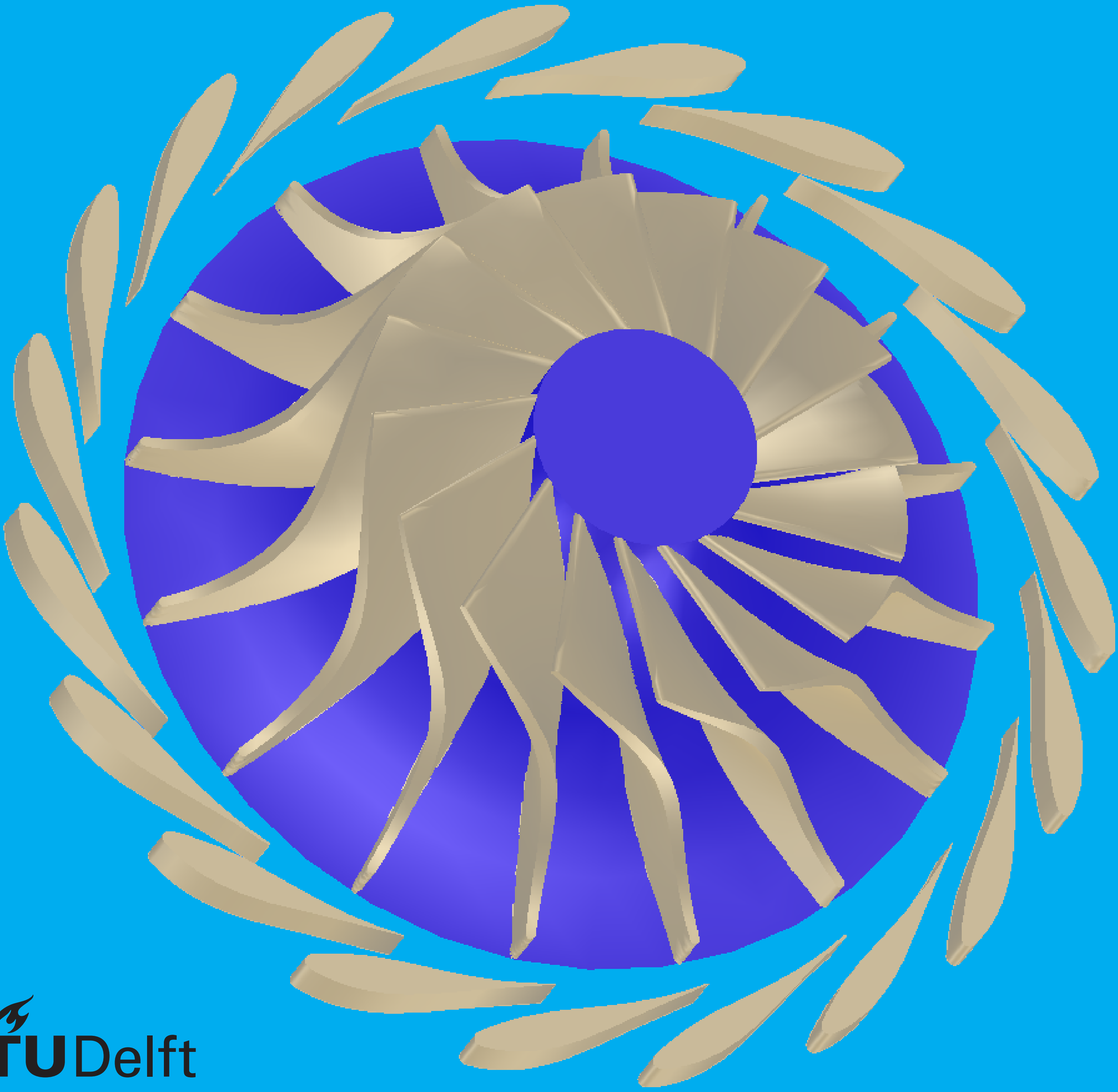
Advanced Parametrisation and Geometrical Analysis of Radial Inflow Turbomachinery

Analysis of geometry construction, geometry discretisation and geometric flow properties of a Radial Inflow Turbomachinery focusing on ORC applications

K. van Giessen

Technische Universiteit Delft



TU Delft

Advanced Parametrisation and Geometrical Analysis of Radial Inflow Turbomachinery

Analysis of geometry construction, geometry discretisation and geometric flow properties of a Radial Inflow Turbomachinery focusing on ORC applications

by

K. van Giessen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday August 25, 2017 at 14:00 PM.

Student number:	4092384	
Thesis Registration number:	143#17#MT#FPP	
Project duration:	December 21, 2016 – August 25, 2017	
Thesis committee:	Prof. dr. ir. P. Colonna,	TU Delft, chairman
	Dr. ir. M. Pini,	TU Delft
	Dr. ir. R. Pecnik,	TU Delft

TU Delft

Preface

This thesis is written as part of the curriculum of the Flight Performance and Propulsion MSc. degree of Aerospace Engineering at the Delft University of Technology. The work done could not have been possible without the help of a number of people.

I would like to thank Salvatore Vitale for his help, his patience, his advice and his endless number of ideas and knowledge of the matter. I would also like to thank him for encouraging me to take the 'Writing English for Technologists' course, as this course has helped tremendously in the process of writing this thesis. Mostly, I want to thank Salvatore for his help, going out of his way to aid me in my research process, his continuous enthusiasm throughout the whole process has gotten me even more excited for the field of turbomachinery.

I would like to thank Matteo Pini for his guidance and advice on producing a high quality thesis report, and now and then steering me in the right direction. I am thankful for the amount of freedom that I was granted throughout my process.

I would like to thank Matteo Burigana for his help in the re-evaluation of the ORCHID blade optimisation cycle. I could not have obtained the data, nor the necessary knowledge of the ORCHID blade design cycle without him. I would like to thank him for his contributions during the brainstorm meetings where we discussed many new and exciting methods and approaches for the optimisation.

Finally, a thank you goes out to my peers, my friends, and my family for their support and their encouraging words, which kept me enthusiastic and motivated even through the rough parts of writing my thesis.

K. van Giessen
Delft, August 2017

Executive Summary

There is an ever-growing need for clean renewable energy in modern day society. Currently, major developments are being made on Decentralised Power conversion systems that use renewable energy sources. The application of ORC radial inflow turbines in Decentralised Power conversion applications seems promising, however, application of this technology is still sparse.

There is a significant technological gap between large and small power plants. Due to this lack of knowledge of the mini-ORC turbine design, the design cycle of ever new mini-ORC turbine design has an extensive duration. The design cycle of ORC (radial inflow) turbines is rather complex, because the organic fluids have a wider range of thermodynamic properties, which not only give more flexibility to the design, but also increase the effective design space that needs to be explored during the design cycle.

To greatly reduce the time required to complete one design cycle, Knowledge-Based Engineering (KBE) could be applied in the (radial inflow) turbomachinery design process. Therefore, the purpose of this thesis, is to reduce the duration of the design cycle, which can lead to a faster transition towards Decentralised Power Generation to help meet the increasing energy demands.

The design cycle of (radial inflow) turbomachinery consists of four main phases, which lead to a final design. The preliminary design phase has been thoroughly researched and will not be discussed in this thesis.

Throughout this thesis the methods used for, the parametrisation phase, where the blade geometry is constructed using a Computer-Aided Design (CAD) tool, and the meshing phase, the phase in which the geometry is meshed are thoroughly discussed. Finally, the thesis has validated the methods by testing the results in a Computational Fluid Dynamics (CFD) solver to analyse the performance.

The investigated methods are combined in a modular and user friendly design tool, which provides a solid and accessible basis for further development of the (radial inflow) turbomachinery design cycle. These methods include: the definition of the meridional channel curves using a set of BSpline curves, the transformation of the 2D channel curves into the 3D camberlines through the blade angle integrator, and the projection of a thickness distribution onto the 3D cambersurface. The added fitting capabilities enable the tool to reconstruct a given geometry from a set of distributions and coordinates. The tool fits 2D meridional curves to provided coordinates, and it fits the blade angle and thickness distribution curves to reconstruct full 3D blade.

The application and automisation of the well-established AutoGrid software, accessible within the turbomachinery design tool, proves to be a solid and valid implementation of mesh generation software in the turbomachinery design cycle. The integrated software package is able to generate high-quality meshes for the (sub-sonic) stator, rotor and diffuser.

The design tool also provides capabilities for mesh generation using ANSYS TurboGrid and Salome. Several 'rules-of-thumb for mesh generation' are implemented in the Salome package. These methods resulted in a Salome mesh package that can produce high-quality hybrid grids.

With the application of the tool on the design process of the RIT ORCHID blade, the design tool has proven to be a valuable tool for the Turbomachinery department at the faculty of Aerospace Engineering. It not only provides the means to easily reconstruct any given rotor blade, it also provides more insight into the geometrical aspects of the flow passage and the blade geometry compared to the well-established ANSYS BladeGen module.

The tool can analyse both the area distribution of the blade passage, as well as the curvature distribution of the blade passage bounds (the blade, hub and shroud). These two parameters are deemed most important for flow characteristics as they can influence flow expansion and flow acceleration.

The meridional channel definition seems to have the greatest influence on the Meridional and the 3D area distribution, while the blade angle distribution primarily influences B2B area distribution. The optimisations, however, have shown that the performance is not only driven by the area distribution.The 3D effects and other flow effects that are not captured by the expansion or contraction of the flow area seem to have significant effect on the turbine performance. The relation between geometry of the flow passage and the flow characteristics should be investigated to a further extend.

List of Figures

List of Tables

Contents

1 Introduction

The dramatic increase in world-wide energy consumption has given rise to environmental problems and a significant shortage in energy resources.[24] Consequently, there is an ever-growing need for clean renewable energy in modern day society. The power plant industry is steadily expanding and evolving to meet those needs.

The branch of 'Decentralised' or 'Embedded' Power Generation (DPG), is gaining more momentum; DPG is likely to become an essential part of the clean energy supply in the near future. Advantages of Decentralised Power Generation over Centralised Power Generation are: the possibility to use more heterogeneous energy sources (among which are renewable energy sources), the possibility of using co-generation, and the lack of distribution losses. One major disadvantage, however, is that it is a much less developed technology compared to Centralised Power Generation. Currently, major developments are being made on Decentralised Power conversion systems that use renewable energy sources. A wider application of this kind of power plants could greatly decrease the environmental impact of energy production, provide a solution for the dwindling energy supply, and simultaneously promote a more competitive energy market. [10]

Decentralised power conversion generally provides a low power output (1kW-1MW). Among the various technologies, Organic Rankine Cycle (ORC) generators seems to be one of the most promising technologies for low power decentralised power conversion. ORC generators use organic fluids rather than steam as a working fluid for their thermodynamic power cycles. This overcomes the challenges of the low mass flow and relatively high expansion ratios (characteristic for a lower power output expander) that would be present if a regular steam expander was to be used. [6] The ORC allows the use of 'relatively low-temperature' renewable energy sources by using the thermal heat from applications such as solar, thermal, geothermal, waste-heat and biomass-combustion energy, which increases the applicability of Decentralised Power Generation to a variety of energy sources. [18]

In 2015, Colonna et al. has highlighted the renewed interest in the concept of low capacity (3 to 50kW) mini-ORC power plants.[6] However, the technological development of high capacity ORC power plants has already reached maturity, but the development of low capacity power plants has not. Therefore, there is a significant technological gap between large and small power plants.[3]

High speed turbines are generally used as the expander for small-scale power generators because of their high power conversion efficiency, their compact size and their low weight. [23] Currently, axial, radial inflow and radial outflow turbines are widely used for ORC power generation, but according to Bahamonde, for low power mini-ORC applications, the radial inflow turbine has a higher turbine efficiency and allows for higher pressure levels, which ensures that a higher level of efficiency of the whole power generation system is achieved. The use of a single radial inflow turbine is much simpler and more cost-effective compared to the use of a corresponding three-stage axial turbine that would be needed for a 10kW ORC turbogenerator.[3]

The application of such ORC radial inflow turbines in small-scale power applications, however, is still sparse, primarily due to the lack of knowledge of the design which results in an extensive duration of the design cycle for this new concept.

The design cycle of ORC (radial inflow) turbines is rather complex, because the organic fluids have a wider range of thermodynamic properties, which not only give more flexibility to the design, but also increase the effective design space that needs to be explored during the design cycle.

The design cycle of (radial inflow) turbomachinery consists of four main phases, which lead to a final design; the cycle is depicted in fig. 1.1. It starts of with a set of initial requirements which leads to a preliminary design. The blade geometry is constructed using a CAD tool, and once the CAD design has been completed the geometry is meshed and tested in a CFD solver to analyse the performance. In general, this design cycle is a lengthy process and requires adequate expertise in the fields of computer-aided design, mesh generation and fluid dynamics.[13]

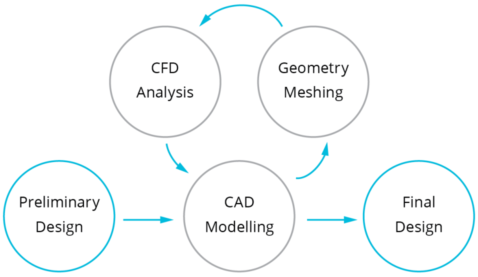


Figure 1.1: Overview of the aspects of design cycle

To greatly reduce the time required to complete one design cycle, KBE could be applied in the (radial inflow) turbomachinery design process.

KBE is a design philosophy in engineering which aims to collect all the necessary engineering knowledge of multiple fields into one system, maximizing the use of design rules and rules of thumb, and minimizing the interdisciplinary delays. Using KBE, the knowledge of different fields would be captured in one integrated tool, to greatly reduce the design time and increase the performance of a turbomachinery blade. An example of KBE can be a design tool with a set of predefined formulae that computes the required mesh resolution for any application, depending only on a small set of flow parameters defined by the user. This circumvents the need of expertise knowledge on mesh generation, and allows the focus to be directed to other parts of the design phase. Ultimately, the coupled CAD tool combined with a mesh generator and designated CFD software can, if used in an iterative manner, become a integrated platform capable of optimizing turbomachinery blades.

Reduce the duration and improving the efficiency of the design cycle, which can lead to a faster and easier transition towards Decentralised Power Generation to help meet the increasing energy demands while simultaneously reducing emissions.

The main focus of this thesis will be to investigate and improve the radial inflow turbomachinery design cycle and reduce the cycle duration where possible, by using KBE.

1.1. Research Questions and Scope

While the main focus of this thesis is on the Design Cycle of Radial Inflow Turbo-machinery, this thesis cannot encompass every aspect of this topic. The research questions and main objectives of this study will be discussed in section 1.1.1, while section 1.1.2 goes into more detail on the scope of the project.

1.1.1. Research Question

Due to the relatively broad topic of this thesis, the thesis will be guided by two research questions. The primary research question, which is mainly focused on the CAD modelling aspect of the design cycle, is as follows:

> ***"What is the most flexible and aerodynamically sensible parametrisation technique to model radial inflow turbomachinery blades in CAD, accounting for subsequent CFD analysis, design and optimisation?"***

The following set of sub-questions will help answer the main research question:

- What parametrization techniques for turbomachinery blades are currently available; how are they defined and what makes them robust and computationally efficient?

- What are the best practises for mesh generation in turbomachinery for subsequent CFD analysis of turbomachinery blades?
- What is the best way to handle and implement geometrical uncertainties?

The second research question is posed to help cover the full scope of the thesis. This question is more focused on the geometry analysis after the CAD modelling phase and is stated as follows:

> *"**How can a parametrised turbine blade be successfully optimised based on geometrical characteristics, before any computationally intensive CFD analysis is conducted?**"*

The following set of sub-questions lead up to answering the secondary research question:

- What geometrical characteristics influence the flow characteristics, and which are the most influential?
- Which design parameters have the biggest influence on these geometrical characteristics?
- What are the desired geometrical characteristics of a radial turbine, within the bounds of the design space?

1.1.2. Scope

The scope of this study entails several different concepts which will be introduced and discussed separately. The topic of the thesis is the improvement of the turbomachinery design cycle. The first phase, the preliminary design, has already been realised by Pini et al. in the recently developed 1D preliminary design tool (zTurbo). To combine the rest of the aspects of the design process of radial turbomachines into one tool, each of the parts of the design process should be researched separately. This regards the three other phases of the design process:

Parameterization of a preliminary design The scope of this study is to construct a set of methods and links to describe a stator or rotor geometry, using only the preliminary design parameters. During th study, the construction methods will be analysed on their accuracy, computational efficiency and their robustness. The parametrisation methods will be integrated in the design tool

Meshing of parametrised geometry The scope of the meshing study is to construct a set of methods that create the fluid domain for the stator or rotor geometry. The methods used in the subsequent meshing of this fluid domain, are thoroughly analysed. The meshing capability methods will also be integrated in the design tool.

CFD coupling and analysis The CFD phase will only be discussed briefly. While the design tool will be coupled in such a way that a full design cycle can be executed, the settings and methods used in the CFD solver will not be discussed. The use of correct flow conditions and turbulence models essential to compare the flow solution with experimental data, this however, requires extensive knowledge and testing of the CFD software, and is beyond the scope of this thesis.

The methods and theories investigated in this manuscript provide a new view on the radial turbomachinery design cycle. Not only does it combine multiple methods from radial turbomachinery design and surface modelling, but it also applies methods used by the neighbouring field of axial turbomachinery design. The investigated methods are combined in a modular and user friendly tool, which ensures adding to the body of knowledge by providing a solid and accessible basis for further development of the (radial) turbomachinery design cycle.

Within the scope of this thesis, the main research objective is to:

> *"**Suggest a possible improvement in performance design of turbo-machinery, by developing an integrated, expandable turbomachinery KBE tool which is able to effectively construct the blade geometries using CAD tools and subsequently analyse and optimise these geometries through CFD**"*

This goal is achieved through the accomplishment of the following deliverables:

1. **Develop** 3D parametric design for radial turbomachinery blades (both rotor and stator) that:

   - constructs any blade based on the preliminary design parameters
   - constructs the corresponding fluid domain for subsequent meshing and CFD analysis

2. **Develop** meshing tools and/or coupling for the fluid domain (Salome, AutoGrid, TurboGrid)

3. **Develop** a coupling between the design tool and the CFD analysis tool (SU2, CFX).

4. **Develop** a geometric analysis method that performs a flow channel analysis in both the 2D and the 3D

5. **Validate** the functionality of the tool by:

   - reconstructing several geometries that are available in the literature
   - performing CFD analysis on the reconstructed geometries

1.2. Report Outline

First, some essential background information on the topics discussed in the thesis is presented in chapter 2. Second, an elaborate decomposition of the software used and of the design tool that has been developed for this thesis is provided in chapter 3. Third, the methodology of: the parametrisation, the geometrical analysis and the meshing that has been applied in the design tool are described in chapters 4 to 6. Fourth, The validation of the developed design tool using geometry and performance data that is available in the literature, is discussed in chapter 7. Fifth, the application of the design tool is presented in chapter 8, by reproducing the design cycle of a design case that is available in the literature. Finally, the thesis is concluded in chapter 9 and section 9.3 by presenting the conclusions and providing a number of recommendations for future research opportunities.

2

Literature Review and Background Information

In the early days radial turbomachinery was often used for aviation propulsion. However, nowadays, radial turbomachinery is mainly applied in power generation solutions and as previously stated, radial turbomachinery is an excellent choice for small-scale power generation applications. In support of this statement, some background information is given in section 2.1 on the fundamentals of turbomachinery power generation, the ORC thermodynamic cycle, and the preliminary design parameters that can be obtained from this. Following the preliminary design, the first two of the three other phases of the design cycle: parametrisation and meshing are thoroughly described in sections 2.2 and 2.3 The advanced settings and model for the theCFD analysis are beyond the scope of this thesis and will therefore not be discussed.

2.1. Preliminary Turbomachinery Design

Compared to an intermediate design, a preliminary design can be computed fairly quickly; the computation is usually based only on fundamental and thermodynamics equations and rules-of-thumb for turbomachinery design. The fundamentals, the thermodynamics and the resulting preliminary design parameters will be described in the next three sections, but none of them can be explained properly until the naming convention used throughout this report has been presented. The sections listed below are illustrated in fig. 2.1 and will be used in subscripts throughout the remainder of the report.

Section 1	Volute ($_1$)
Section 2	Stator Inlet ($_2$)
Section 3	Stator Outlet ($_3$)
Section 4	Rotor Inlet ($_4$)
Section 5	Rotor Outlet ($_5$)

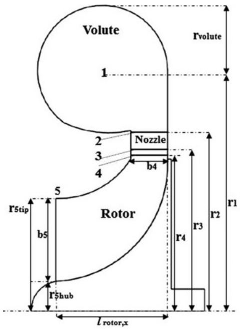


Figure 2.1: Section definition in radial inflow turbine, source:[24]

2.1.1. Fundamentals

The main design purpose of a turbine is to extract mechanical work from the potential and kinetic energy of a flowing fluid, whilst for its counterpart, the radial compressor, the main purpose is to add potential and kinetic energy to the flowing fluid. The work extraction is realised by applying a reduction in the fluid velocity (kinetic energy) - both in terms of the direction and the magnitude - and a reduction in pressure (potential energy) over the turbine stage.

The change in velocity can be further illustrated through the use of velocity triangles. Due to the curvature of the hub and shroud, which curve the flow from radially inwards to axially outwards, the velocity triangles for radial turbomachinery are mapped into two planes: a radial and an axial plane. Both planes and their corresponding velocity triangles are depicted in fig. 2.2. Each velocity triangle shows three main velocity vectors: Absolute Flow Velocity, Relative Flow Velocity and Rotor Blade Velocity, denoted as C, W and U. The magnitude of these velocities can fully determine the power extraction or addition as illustrated by eq. (2.1).

$$\frac{\dot{W}}{\dot{m}} = \frac{1}{2}\Big[\left(U_4^2 - U_5^2\right) + \left(W_4^2 - W_5^2\right) + \left(C_4^2 - C_5^2\right)\Big] \tag{2.1}$$

Note that the equation only requires the flow velocity magnitudes, which are simply a result of the absolute and relative flow angles (α and β respectively) and U. These are the driving parameters in radial turbomachinery design and they will be further discussed in section 2.1.3.

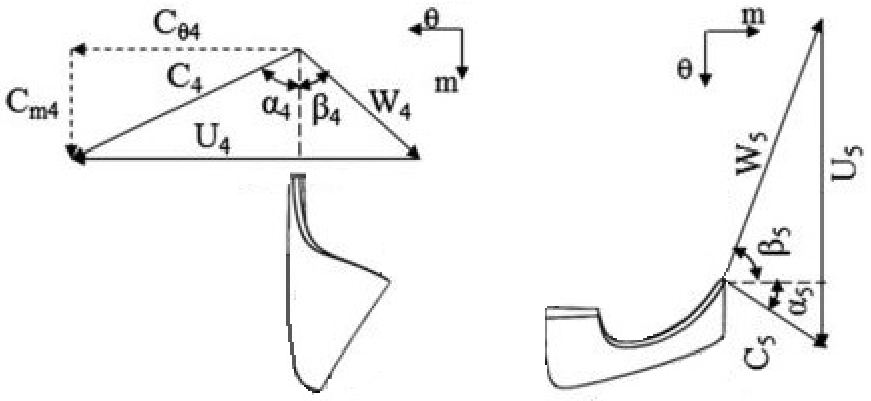


Figure 2.2: Velocity Triangle of radial inflow turbine

Equation 1.1 holds for both axial and radial turbomachinery. In the case of axial turbomachinery, the first term on the right hand-side ($U_4^2 - U_5^2$) is equal to zero, whilst for radial turbomachinery, the same term is greater than zero. The power output of an axial turbomachine is only related to the changes in W and C, while a radial turbomachine can also extract work from the change in U. This makes a single stage radial turbine slightly more effective at extracting power than a single stage axial turbine with equal flow conditions and rotational speeds.

2.1.2. Thermodynamic Cycle

While the thermodynamic cycle traditionally used for radial turbine power generation is the Joule-Brayton Cycle using air, new developments regarding clean renewable energy, have spiked the interest in the use of a new form of the Rankine Cycle. Traditionally, the Rankine Cycle is a relatively simple cycle consisting of an expansion turbine (expander), a condenser, a pump, an evaporator and a super-heater. The cycle converts high-temperature thermal energy into electricity. The newly developed Organic Rankine Cycle allows for the use of organic fluids (such as the R245fa, R236ea or R141b fluids, and even Butane-gas[30]) with a wide variety of molecular weight and no need for a super-heater (see fig. 2.3), which presents several advantages over the traditional stream-driven Rankine Cycle. The increased fluid density allows for many changes in the design of the turbine (expander); it allows for down-scaling; it reduces maintenance costs by avoiding impingement problems; it allows a blade loading greater than one; and it allows a reduction both in pressure and temperature, which in turn reduces the mechanical stresses on the turbine.[23]

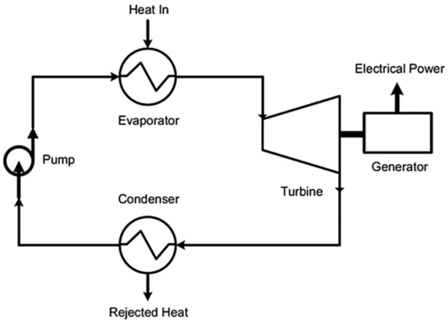
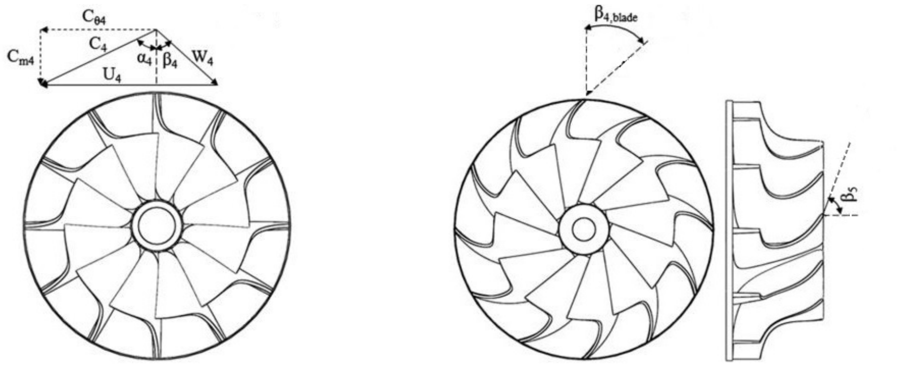


Figure 2.3: Basic Organic Ranking Cycle, source:[27]

According to Bahamonde, the

The down-scaling and maintenance reduction, allow for a reduction in both initial operational costs. The lower mechanical stresses in the turbine allow for non-zero blade angles (with respect to the radial direction). Not only does this non-zero blade angle possibility increase the design flexibility, it also increases the C velocity possible for constant U and W velocities, as can be seen in fig. 2.4. This increase in C will directly increase the theoretical power output of the turbine. An example of such a set-up is the application of a supersonic stator, where an absolute flow velocity (C_4) of above Mach 1.0 is achieved at the rotor inlet, see [1].



(a) Zero blade angle

(b) Non-zero blade angle

Figure 2.4: Regular vs. Non-zero blade angle, source:[23]

Besides increasing the theoretical power output of the turbogenerator, the ORC turbogenerators also provide higher thermal efficiency compared to the traditional steam turbogenerators.[12] They also enable the expander to use a low number of stages, because of the small specific work; in the case of the radial turbine it even allows for a single stage. However, the ORC also poses several design challenges, mainly concerning the expander which, according to Pini et al., 'represents the most critical component in terms of efficiency, compactness and cost.'[22]

Due to the limited number of stages, relatively high expansion ratios per stage are required. This can cause problems when organic fluids are used as the working fluid. The combination of high expansion ratios and the lower speed of sound of organic fluids results in trans-sonic and supersonic flows within the turbines. Alongside complex transonic flow conditions, the thermodynamic state of the fluid also poses a challenge. The fact that the fluid expands under conditions in which the thermodynamics deviate from the ideal gas behaviour, makes the flow analysis more challenging and increases the uncertainty, thereby complicating the design process.[22, 27]

While the use of organic fluids can increase the efficiency and flexibility of the turbine design, it also complicates the thermodynamic calculations. The wide variety of organic fluids that can be used for the different applications increases design space. This increases the computational time of a single design cycle, due to the additional selection process of the organic fluids.

2.1.3. Preliminary Design Parameters

In view of the previously described challenges in ORC turbine design, 'improved designs of ORC turbines are nowadays of significant interest; these improvements require the development of specific design procedures and the evaluation of non-conventional architectures.'[22] Improving the design process and consequently also the optimisation process of the turbine (expander) is therefore essential.

For the optimisation design of an ORC turbogenerator, the two main subjects are the (organic) working fluid and the geometrical design of the turbine. The selection of the optimal working fluid remains beyond the scope of this project, however, the selection procedure has been extensively researched and is available in the literature.[5, 24, 30] Regarding the geometrical design of the turbine, the preliminary design phase has also been thoroughly investigated by Pini et al., however, there is very limited literature available on the detailed design of ORC turbines.

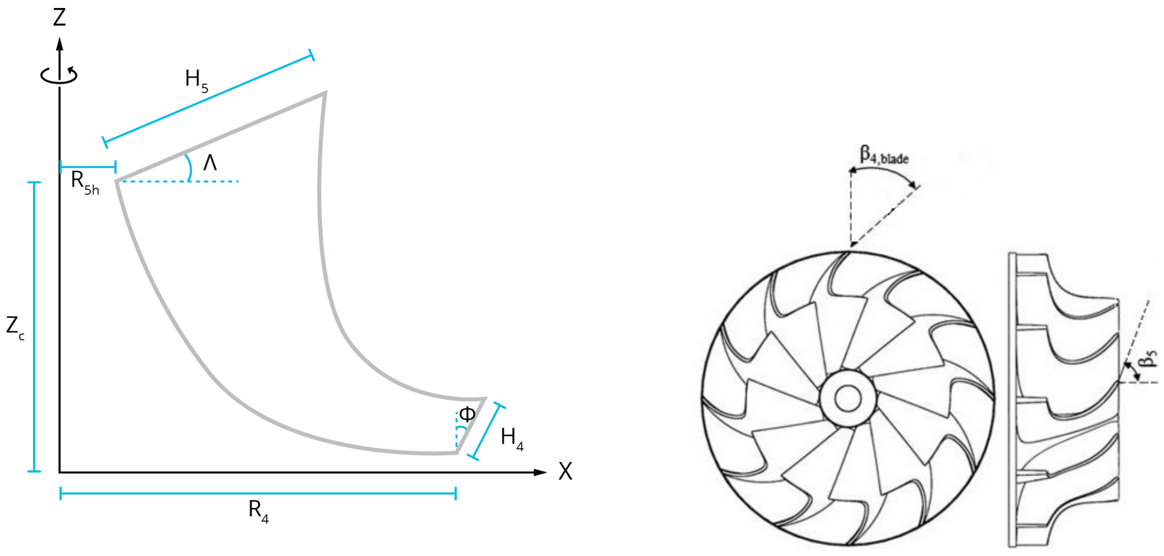
The detailed design requires the preliminary design parameters as input to respect the preliminary design. For the preliminary turbine design, zTurbo, a fully validated 1D design-analysis code, will be used. While zTurbo can evaluate multiple conventional and non-conventional turbine architectures, within the scope of this thesis only the design of a radial inflow turbine (RIT) will be assessed. Besides the wide applicability of the RIT for small-scale power generation, the RIT design is also the most complicated geometry to implement in CAD, it is therefore one of the more desired architectures to parametrise to ensure a faster design cycle.

For the purpose of ORC expander design, zTurbo implements the treatment of real gases and different loss models to generate an efficient and realistic preliminary ORC turbine design, with only a 2% error in efficiency compared to in the literature.[2] The program uses multi-variable optimisation to output the following parameters which serve as the input to the high-fidelity design:

$\boldsymbol{\beta_4}$ Relative inflow angle at rotor inlet

$\boldsymbol{\beta_5}$	Relative outflow angle at rotor outlet
$\mathbf{R_4}$	Radius at Leading Edge / rotor inlet
$\mathbf{H_4}$	Blade Height at Leading Edge / rotor inlet
$\mathbf{R_{5_h}}$	Radius at Trailing Edge / rotor outlet
$\mathbf{H_5}$	Blade Height at Trailing Edge / rotor outlet
$\mathbf{chord}_{Ax}$	Axial chord-length of the rotor
$\boldsymbol{\Delta r}$	interspace distance between stator trailing edge and rotor leading edge
$\mathbf{N_B}$	amount of blades uniformly distributed with constant pitch in the radial plane

Figures 2.5a and 2.5b present the geometrical description behind the coefficients, projected on an arbitrary radial inflow turbine.



(a) Meridional Channel showing radii and heights

(b) Indication of the blade angles of the rotor

Figure 2.5: Geometry Parameters

2.2. Parametrisation

Current ORC Radial-inflow turbines usually feature a supersonic stator and a highly loaded rotor due to the high expansion ratios they both have to handle. This thesis, however, will only focus on the development of the highly loaded rotor, since the parametrisation of the supersonic stator has already been addressed in the thesis of Anand, N. [1].

It is the process in which the preliminary design parameters are used to create an initial Computer-Aided Design (CAD) geometry, which will be useful both for visual inspection, and for the topology data necessary for the subsequent meshing and Computation Fluid Dynamics (CFD) phases of the design cycle. This section will explain which numerical scheme will be used and how a blade geometry can be constructed in CAD in such a way that it is well set up for consecutive analyses.

2.2.1. NURBS

It is the computation and construction of curves and surfaces that forms the bases to all CAD operations. The parametric form of the curves and surfaces (eqs. (2.2a) and (2.2b) respectively) that is generally used, due to the numeric nature of the CAD applications.

$$\mathbf{C}(u) = [x(u), y(u), z(u)] \tag{2.2a}$$

$$\mathbf{S}(u, v) = [x(u, v), y(u, v), z(u, v)] \tag{2.2b}$$

Splines are widely used in CAD applications, since they provide an easy and relatively computationally inexpensive way of defining the referred to curves and surfaces using the parametric form. In particular a B-

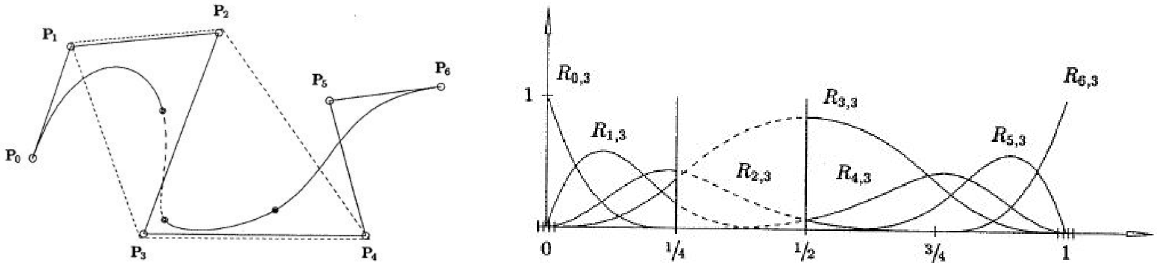
Spline is the prevalent method nowadays; comprising piece-wise polynomials in the parametric form, which allow for good control of the curve. The B-Spline consists of a Basis Function, Control Points and a knot-vector to form the parametric curve. For any B-Spline curve, p is the degree, $n+1$ is the number of control points, and $m+1$ is the number of knots; these parameters are related by: $m = n + p + 1$

While the B-Spline already provides excellent local control, they are still not able to represent every possible shape. Particularly conics and circles cannot be parametrised using simple B-Splines. This lack of control can be overcome by applying weights to the Basis Function. From this, a subset of B-Splines is formed known as the Rational B-Splines, which are able to represent conics and circles. The corresponding weighted Basis Function is defined using eq. (2.3a); the complete parametric representation of a Rational B-Spline is shown in equation eq. (2.3b).

$$R_i^p(u) = \frac{N_i^p(u) w_i}{\sum\limits_{j=0}^{n} N_j^p(u) w_j} \tag{2.3a}$$

$$\mathbf{C}(u) = \sum_{i=0}^{p} R_i^p(u) \mathbf{P}_i \quad a \leq u \leq b \tag{2.3b}$$

In general, the knot-vector that serves as input for the Rational B-Spline is equally spaced, i.e. each knot-span $[u_i : u_i + 1]$ is of constant length. Should one require even more flexibility in parametrisation, a non-uniformly-distributed knot-vector can be used. A Rational B-Spline that uses such a vector is called a Non-Uniform Rational B-Spline (NURBS), an example of which can be seen in fig. 2.5. NURBS curves and surfaces will be predominantly used throughout the thesis project due to their geometric/trigonometric/algebraic properties and their intuitive and excellent control.



(a) NURBS Curve

(b) NURBS Basis Function

Figure 2.6: NURBS curves, source:[21]

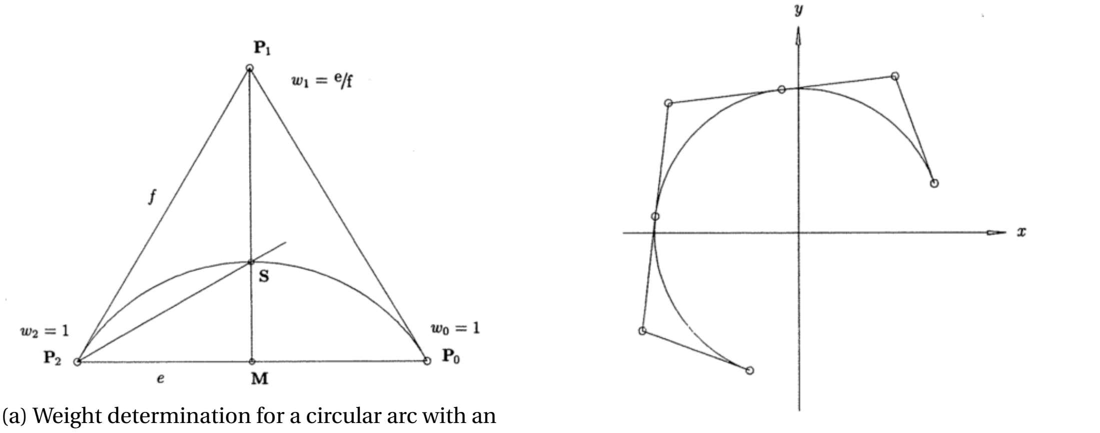
Several of the algebraic properties are particularly useful for fluid-dynamic design in CAD. It is mainly the direction and curvature of the curve (i.e. the derivatives of the curve) that is useful in determining the direction and acceleration of the flow. Due to the numerical definition of a NURBS curve, the derivative can be easily evaluated at any point along the curve. This derivative can, for example, not only be used to define the tangent or the normal vectors along the curve, but it can also be useful for the blending of different splines, both of which are very common procedures in fluid-dynamic design. The first and higher order derivatives of a NURBS curve are defined in eqs. (2.4a) and (2.4b) respectively.

$$C'(u) = \frac{\mathbf{A}'(u) - w'(u)\mathbf{C}(u)}{w(u)} \tag{2.4a}$$

$$C^{(k)}(u) = \frac{\mathbf{A}^{(k)}(u) - \sum_{i=1}^{k} \binom{k}{i} w(u)^{(k)} \mathbf{C}^{(k-i)}(u)}{w(u)} \tag{2.4b}$$

Another excellent property of a NURBS curve, in relation to its derivatives, is the level of continuity of the curve. The continuity and differentiability of a NURBS curve follow from the basis function $N_{i,p}(u)$, since the curve is simply a linear combination of the basis function. Therefore, the curve is infinitely differentiable with the knot interval, while having continuity of $p-k$ at a knot of multiplicity k. This allows for the design of very smooth curves, which directly affect the presence of super-velocities along the blade, effectively reducing the occurrence of shockwaves or separation, both of which are detrimental to the aerodynamic performance of the turbomachine. Section 2.2.3 will go into more detail on these effects.

The ability to create conics and circles, combined with the desired derivatives and continuity properties makes NURBS curves the perfect method for the parametrisation of blade profiles. Nowadays, conical and circular shapes play an intricate part in the parametrisation of leading and trailing edges of blade profiles. It is therefore useful to gain insight into the parametrisation of a circle using a NURBS curve. In figs. 2.7a and 2.7b below. One can see how a NURBS curve can parametrise a partial circle using three or more control points. Figure 2.7a shows how applying a weight of $e/f(=\cos(\theta))$ to $\mathbf{P}_1$ constructs a circular arc with tangency at $\mathbf{P}_0$ and $\mathbf{P}_0$. It is essential that the triangle formed by the control points in bilateral. These circular arcs can be pieced together, as can be seen in fig. 2.7b, to create semi-circles or more.

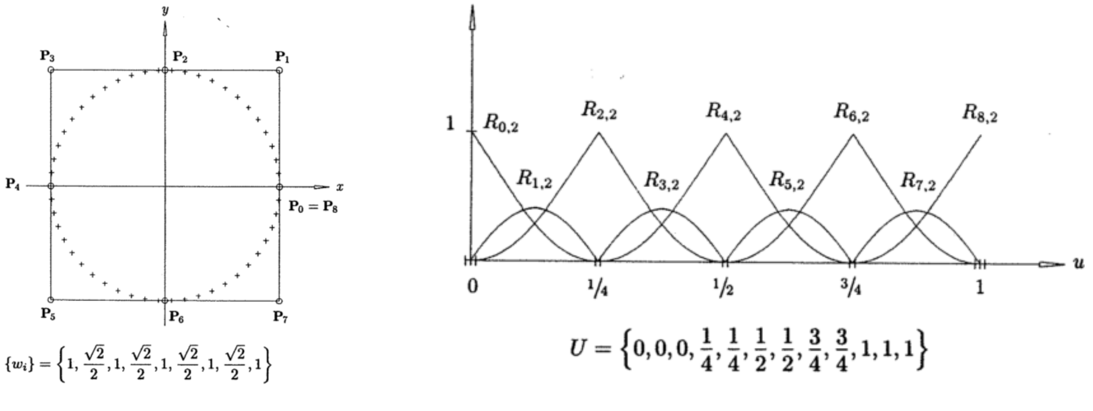


(a) Weight determination for a circular arc with an arbitrary sweep angle

(b) NURBS parametrisation of a partial circle by piecing together circular arcs of equal sweep angle

Figure 2.7: NURBS curves, source:[21]

When piecing together curves, at the location of the joining of the curves, the knot multiplicity should be raised to equal the degree, as can be seen in the knot vector in fig. 2.8b. While this allows the exact intersection of particular points on the NURBS curve, which are necessary to create conics or circles, the Basis Function has only a C_0 continuity as seen in property described above. However, discontinuous functions can sometimes be combined into a continuous curve by the proper placement of the control points. In fig. 2.8a, a NURBS circle with degree 2 ($p = 2$) is shown. This quadratic curve has several knots with a multiplicity of 2 ($k = 2$), hence it should have a C_0 continuity ($p - k = 0$) at the knot (also visible in fig. 2.8b where the basis function is C_0 continuous). However, due to proper control point placement, those knot locations are actually C_3 continuous.


y
P₃
P₂
P₁
P₄
x
P₀ = P₈
P₅
P₆
P₇
$\{w_i\} = \left\{1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1\right\}$
1
$R_{0,2}$
$R_{1,2}$
$R_{2,2}$
$R_{3,2}$
$R_{4,2}$
$R_{5,2}$
$R_{6,2}$
$R_{7,2}$
$R_{8,2}$
0
1/4
1/2
3/4
1
u
$U = \left\{0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1\right\}$

(a) Control Points and Weights

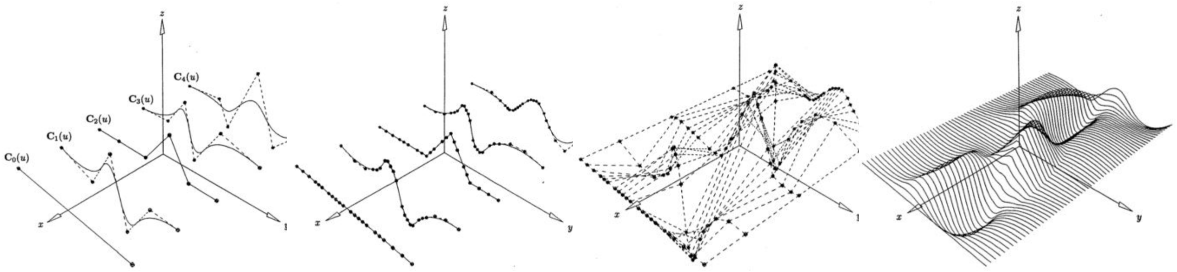
(b) NURBS Basis Function and Knot Vector

Figure 2.8: NURBS parameters for a 9-point circle, source:[21]

Where the curve was a function of one independent variable u, a surface is a function of two independent variables, u and v, which creates a grid of u by v in the $\mathbb{R}^2$ domain that is translated into the (x,y,z) coordinates in the $\mathbb{R}^3$ domain, i.e. 'it is a mapping of a domain $D \simeq \mathbb{R}^2$ to $\mathbb{R}^3$'.[9]

While a B-Spline surface is essentially a stacking of B-Spline curves in the v-direction, a B-Spline surface does not necessarily intersect with all the B-Spline curves that it uses. The lofting operation, commonly used in CAD, uses the set of Control point of the B-Splines as the basis for the Control Net, which, when combined with the corresponding basis functions does not ensure intersection with the aforementioned B-Spline curves. While this lofting operation is computationally efficient and relatively accurate, for high-fidelity Fluid-Dynamic analyses the deviation from the curves might be unacceptable. To ensure exact correspondence of the basis B-Spline curves, two NURBS surface fitting methods can be used: the surface skinning method and the Spine path fitting method.

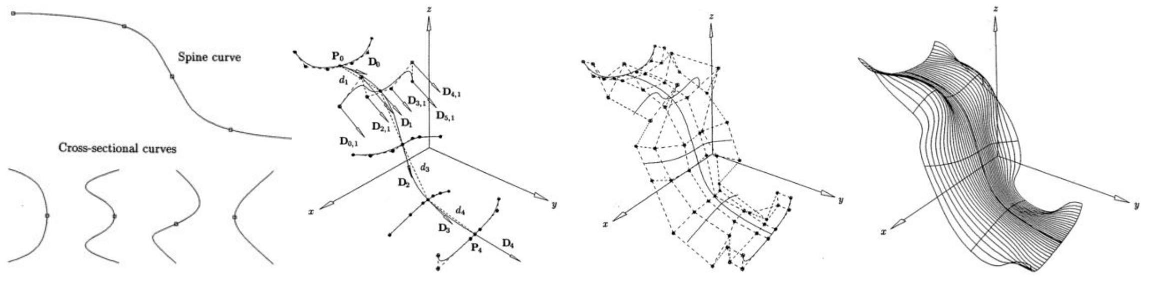
The surface skinning method tries to blend the section curves into one surface, much like the widely-used lofting operation. The blending process first ascertains that the different section curves are compatible, which might require degree raising and knot-insertion. The section curves are then interpolated to raise the amount of CPs per curve, to create enough CPs for the skinned surface. Then, finally, the skinned surface is constructed from the new CPs. The skinning method generally increase the amount of CPs dramatically to ensure proper fitting. The skinning method is used in the construction of the cambersurface from the camberlnine, as will be discussed in chapter 4.



(a) Cross Sections (b) Compatibility (c) Control Points (d) Skinned Surface

Figure 2.9: Skinning Method, source:[21]

Another advanced method for fitting a surface to a set of points (based on the skinned surface fitting method) is the Spine Curve Fitting Method. A similar set of section curves, cross-sections in plane, are given, and a spine or path curve is defined. The section curves are placed along the predefined path, so that the section's planes are perpendicular to the direction (tangent) of the spine curve. The section curves are interpolated, and at each CP the derivative vector is computed, see fig. 2.10b. The surface is skinned using the CPs and it is forced to be parallel to the derivative vectors at the CP locations, see fig. 2.10d.



(a) Spine + Curves (b) Derivatives (c) Control Points (d) Spine Surface

Figure 2.10: Spine Curve Method with Controlled Interpolation, source:[21]

It was found that while this method is very useful in many CAD operation for added control, but for the parametrisation of the rotor in particular this method cannot be used. The complex geometry of the rotor has no planar profiles or curve in any of it's surface parametrisations.

2.2.2. Meridional Channel and Camber-Surface

Ass discussed above, for the parametrisation of the rotor NURBS curves and surfaces are extensively used. First the 2D curves constructed during the meridional channel parametrisation. This parametrisation concerns itself with the shape of the 2D flow passage from the rotor inlet to the rotor outlet. The most commonly used method is the 'mean-line method', which follow the mean streamline throughout the rotor flow passage. This streamline is used to define the hub, tip, inlet and outlet boundaries.

In general, the mean line method starts off by defining the coordinates of the leading edge and trailing edge points at both the hub and tip of the blade in the meridional (2D, radial-axial) plane to define the inlet and outlet boundaries, which requires the preliminary design parameters. Subsequently, the curves, which flow from a radial inwards direction towards the axial outwards direction, of both the hub and tip boundaries are defined. The shape of the curve often represents a circular or elliptical arc, but can also be defined using NURBS curves and a set of Control Points as shown in fig. 2.11.

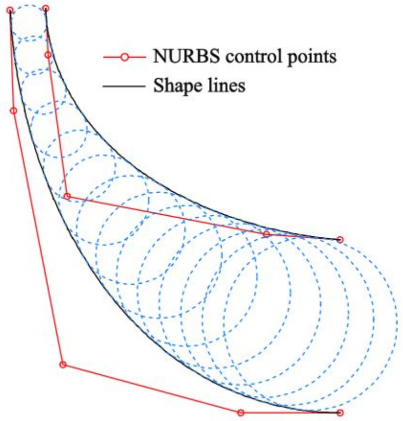


Figure 2.11: Meridional Channel curves, defined using Control Points, source:[16]

The (NURBS) curves defining the 2D meridional channel are then mapped into the 3D plane to form a 3D surface called the camber surface. This surface is particularly complex, because it shows curvature in two directions. Mueller used the geometrical relation between the blade angle (β) and the wrap angle (θ) in eq. (2.5) to convert the β-distribution which, according to Derakshan, is of great influence on the performance of the blade[8], into a θ-distribution which is geometrically much easier to implement into a CAD model.

$$R \cdot \mathrm{d}\theta = \tan(\beta) \cdot \mathrm{d}m \tag{2.5}$$

The β-distribution was parameterised using a B-Spline (see fig. 2.12), to ensure that the integral that follows from eq. (2.5) can be numerically solved using a high discretisation.

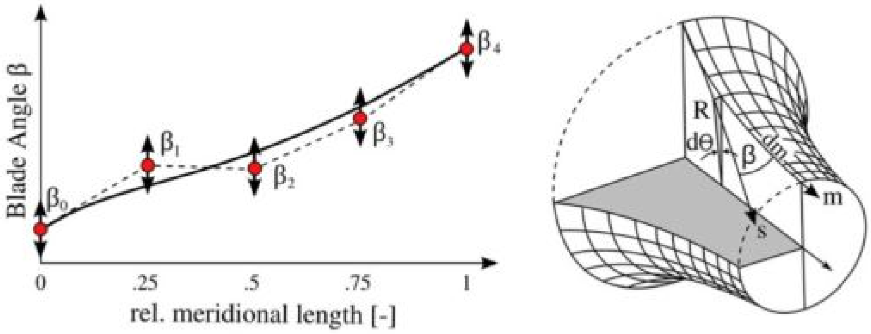
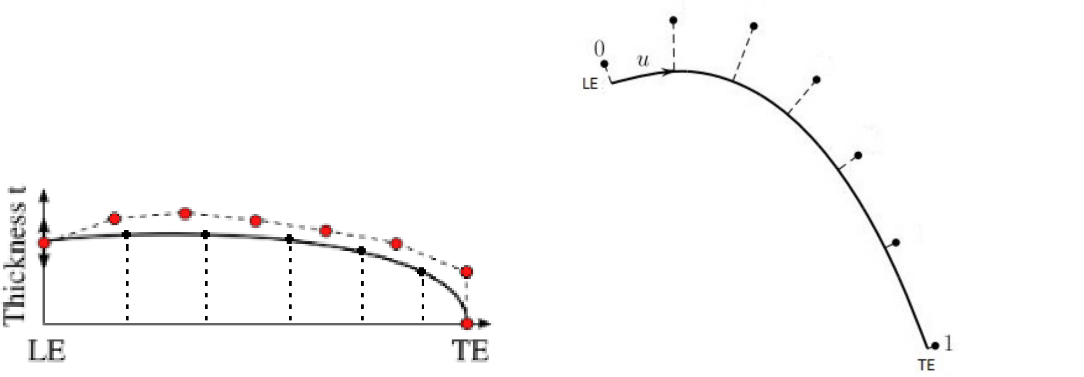


Figure 2.12: β-distribution, and derivation of Wrap Angle, source:[19]

2.2.3. Blade Profile

Considerable research has been conducted into the aerodynamics field of aerofoil and blade profile parametrisation. Particularly the areas of aerofoil and axial blade aerodynamics have been thoroughly investigated, because the performance of axial blades is greatly influenced by changes in the profile shape. In the case of general radial inflow turbine, the profile shape is of secondary importance compared to the beta-distribution as described by Derakshan, however, for turbines with transonic flows, which are more frequently present in ORC applications, the thickness distribution has a significant effect on the performance.[8, 22] Due to the novelty of the ORC radial inflow turbines, limited research regarding the profile parametrisation of radial inflow turbine blades is available. Therefore, the theories and methods discussed below are mainly related to profile parametrisation studies for axial turbine blades, but it si assumed they are also applicable to radial inflow turbines.

A general blade profile is constructed by projecting a thickness distribution perpendicular to the camberline, or, in case of an RIT, a cambersurface. Mueller used an (eight-point) B-Spline curve to define the thickness distribution, which ensures a smooth profile curve along the camberline, see fig. 2.13 for an example of the thickness distribution and the corresponding construction of the profile curve.[14, 19]



(a) Thickness Distribution

(b) Thickness Projection onto Camberline

Figure 2.13: Blade Profile from Thickness Distribution and Camberline, source:[19]

Near the leading edge and trailing edge, a blending operation can be applied, since often specific shapes such as circular or elliptical arcs (which can be described using NURBS) are used for improved aerodynamic performance. Many of the earlier profile parametrisation methods would define the leading edge using a circular arc. An elliptical arc is found to be better suited for the leading edge, as it generally creates less of a pressure spike and separation near leading edge, mainly due to a smoother transition between the leading edge arc and the profile thickness distribution.[15, 29] Research has been conducted into improving the transition of the leading and trailing edge curves into the profile thickness distributions. One particular method has shown promising results regarding the problem at hand.

The (CIRCLE) method devised by Korakianitis[15], ensures that a highly efficient profile design with good aerodynamic performance will result, by defining 2D and 3D blade shapes which are fully continuous along the surface in terms of both their curvature and slope of curvature. The method is based on a published blade-design method, allowing for the use of circular or elliptical shapes on the trailing and leading edges. It provides an innovative and new way of designing both 2D and 3D turbine and compressor blades, by decoupling the thickness and camber definitions which are generally used in conjunction throughout the blade design process.

The curvature and slope of curvature of a blade surface both have an impact on the aerodynamic efficiency performance of the blade, because they directly influence the boundary layer development. To ensure continuity in this slope of curvature, the blade surface is required to be third derivative continuous. However, most parametric splines and surfaces are only first and second derivative continuous, due to third derivative discontinuities at the spline knots.

These third derivative discontinuities can often be found in surface pressure distributions, recognizable by the 'kink' in the distribution, which tends to indicate reduced aerodynamic performance. Such discontinuities frequently occur when a blended circle or ellipse (usually applied in connection with manufacturing considerations) is used for the leading edge, but the CIRCLE method has shown that it can fully remove these discontinuities, thereby ensuring better aerodynamic performance. The CIRCLE method uses the algebraic definition of curvature as seen in eq. (2.6a), and of which the derivative with respect to x is taken, to get the slope of curvature, (eq. (2.6b)).

$$C = \frac{1}{R} = \frac{y''}{[1+(y')^2]^{(3/2)}} \tag{2.6a}$$

$$C' = \frac{dC}{dx} = \frac{y'''[1+(y')^2] - 3y'(y'')^2}{[1+(y')^2]^{(5/2)}} \tag{2.6b}$$

At the trailing edge, a circular arc and a third degree polynomial are defined. The polynomial coefficients are optimized to ensure continuity up to the third derivative. The leading edge, which has the greatest impact on aerodynamic performance, has a circular arc and an additional parabolic construction line, perpendicular to which a thickness distribution is projected. Here, the thickness distribution is optimized to ensure third derivative continuity. The optimisation of coefficients at both the leading and trailing edge, ensure a continuity in terms of curvature and slope of curvature.

While this method is primarily applied for axial turbomachinery and uses a simple y(x) polynomial to describe the curve, the method described can be combined with NURBS curves to be applied for raidal turbomachinery as well. When applying this method to a NURBS curve, one has to ensure not $y'''(x)$ continuity, but $C'''(u)$ continuity.

2.3. Meshing

The second aspect of the design cycle is the meshing of the CAD geometry. Meshing is the process in which the geometry is discretised in preparation for a quantitative numerical analyses. The topology of the CAD file is used to generate a grid or data set of cells or volumes, which can be used to solve Partial Differential Equations (PDEs), both for FEM or CFD analyses. A FEM analysis of the blade is outside the scope of this project, so for this thesis the meshes are generated for CFD analyses only, i.e. for solving the Navier-Stokes PDEs. This section will explain how a grid of cells is generated in such a way that it is well set up for a CFD analyses; this includes a description of the full domain of the mesh, the type of grids used and the quality and quantity of the cells used.

2.3.1. Meshing Domain

The fluid domain of a radial turbine is more complex than most other turbomachinery blade architectures. Not only does it change from a radial direction to an axial direction, it also has a bi-directional curvature in its periodic boundaries, an example of which can be found in fig. 2.14. These periodic boundaries surround the rotor blade geometry, but also extend outwards at both the inlet and outlet of the rotor.

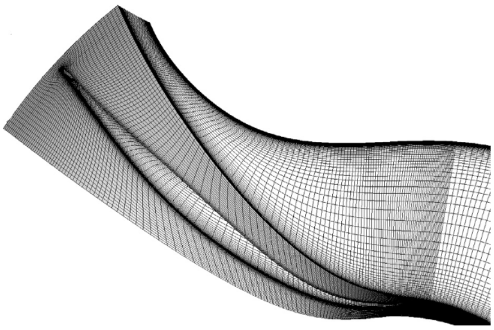


Figure 2.14: Meshing Domain of Turbine Impeller, source:[27]

At the inlet, one has to consider the interspace between the stator and the rotor of the turbine. Oftentimes, a separate stator mesh is added to the rotor mesh, which enables the computation of the interference and mixing effects between the stator and rotor. However, for this computation both meshes have to be combined into a single mesh domain, requiring increased computational time.

At the outlet of the rotor, the domain is extended to simulate the wake of the flow. The domain should directed axially just behind the rotor outlet to ensure a better simulation of the wake and to reduce the volume of the meshing domain. When combining this approach with the periodic boundaries, the wake can be simulated multiple times in one periodic mesh, as indicated by the flow pattern in fig. 7.8b.

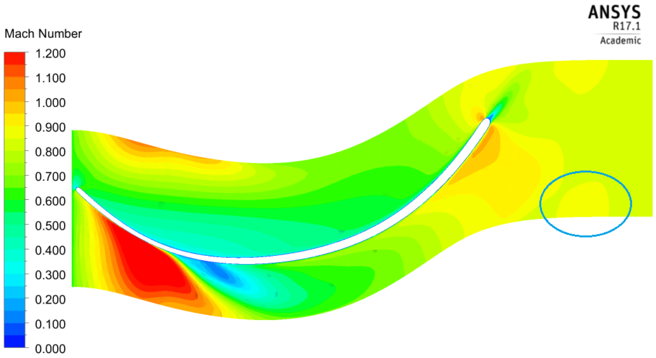


Figure 2.15: Wake in Meshing Domain

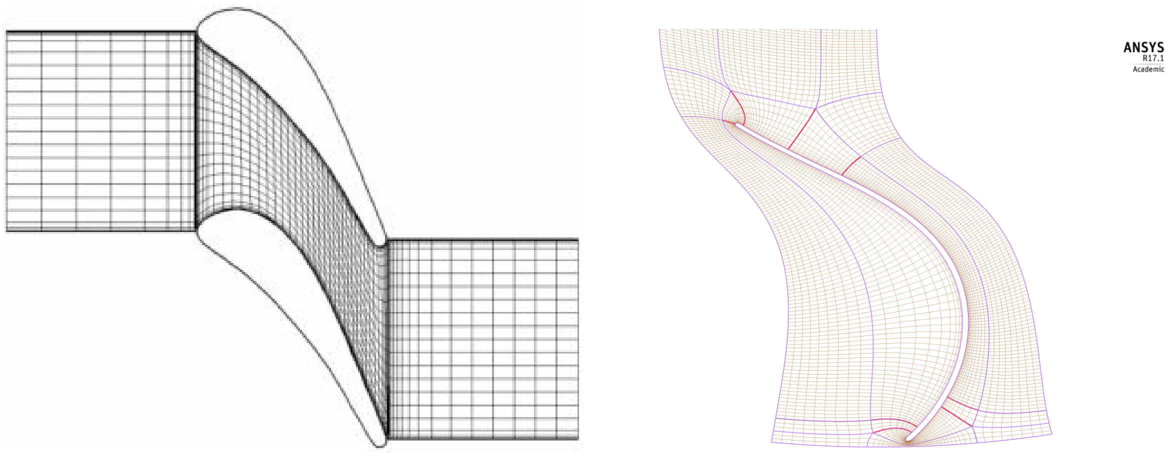
Generally, a centered-blade topology is used instead of a passage-centered topology. This tends to eleminate some solver stability issues, because it avoids the location of the periodic boundary at critical flow locations such as the leading and trailing edges where high gradients in flow conditions are present.

The periodic boundary condition is essential in the meshing of turbomachinery. Therefore great care should be taken in the proper definition and matching of the periodic boundaries. The construction of the periodic boundaries is quite difficult due to the complex geometry of the rotor. Not only does the rotor have curvature in multiple directions, which makes one-to-one matching of nodes very difficult, it also tends to have a high stagger angle at the trailing edge, which could provide difficulties is generating a smooth mesh.

2.3.2. Structured Grids

A structured grid is a 'set of points distributed over a calculation field formed by the intersections of curvilinear coordinate surfaces'.[31] These points then are connected to form quadrilateral cells or hexahedral volumes, see figure fig. 2.16a. Due to the structured nature of this grid, the connectivity does not need to be stored explicitly, since connectivity can be deduced from the locations of the points. This ensures not only reduced data storage, but also provides a less computationally intensive solution and generally increased accuracy compared to an un-structured grid.[28]

While the reduced storage and computation time are always desirable, the structured nature of the grid does pose several challenges. Due to it being limited to quadrilateral and hexahedral shapes, both of which provide limited flexibility, complex geometries are very difficult to mesh. It is also impossible to add extra cells in particular areas of the mesh to increase the mesh resolution, because all grid lines must be conserved. To overcome these limitations, zonal or multi-block grids were developed to enable refined meshing of complex geometries (fig. 2.16b).



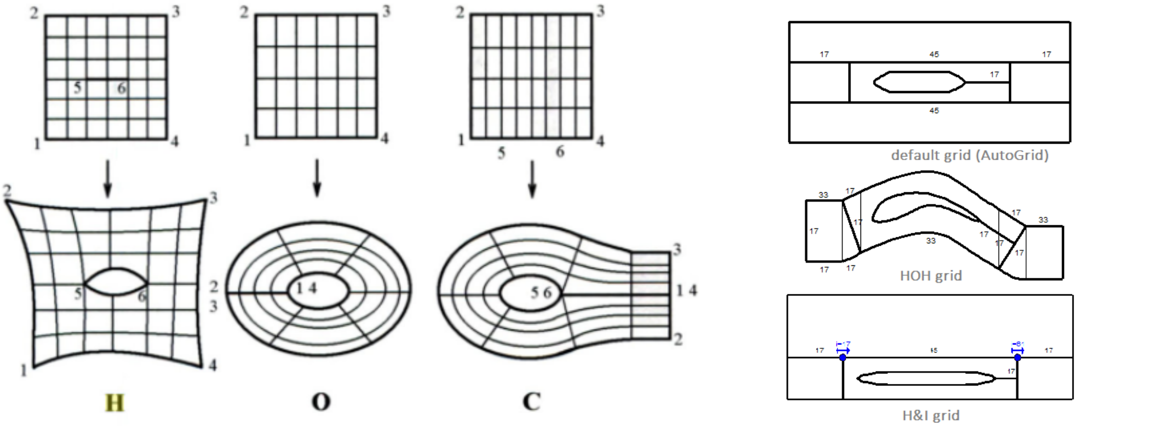
(a) Regular Structured Grid

(b) Multi-block (zonal) Structured Grid

Figure 2.16: Structured Grid examples, source:[32]

The multi-block grid generation method creates several simple structured meshes, one for each (often user-defined) block/partition. After this initial grid generation, the algorithms need to make sure that at the intersections between the blocks the connectivity points are compatible, i.e. number of nodes and their locations must be the same.[7] Unfortunately, this method is considered tedious and requires a certain level of expertise.

Figure 2.17b provides some examples of some widely-used multi-block layouts. These layout generally use a combination of the three main Topology patterns shown in fig. 2.17a: the H-grid, the O-grid, and the C-grid.



(a) Nomenclature for Grid Topology patterns, source: [17]

(b) Multiblock Grid Topologies in AutoGrid

Figure 2.17: Structured Grid Topology Definitions and Examples

2.3.3. Un-Structured Grids

An un-structured grid is ‘a set of points distributed over a calculation field formed with no relation to coordinate directions’. The unstructured mesh generally consists of triangles and tetrahedra in 2D and 3D, respectively, in its most basic form, but may be made of hexahedra or elements of any shape in general.[31] These shapes provide more freedom and are particularly useful in handling shapes with high curvature, allowing for the meshing of rather complex shapes, see fig. 2.18. However, contrary to the structured grids, the connectivity of the points in an un-structured grid do need to be stored explicitly. The un-structured grid generation therefore is more computationally intensive and has more data overhead.

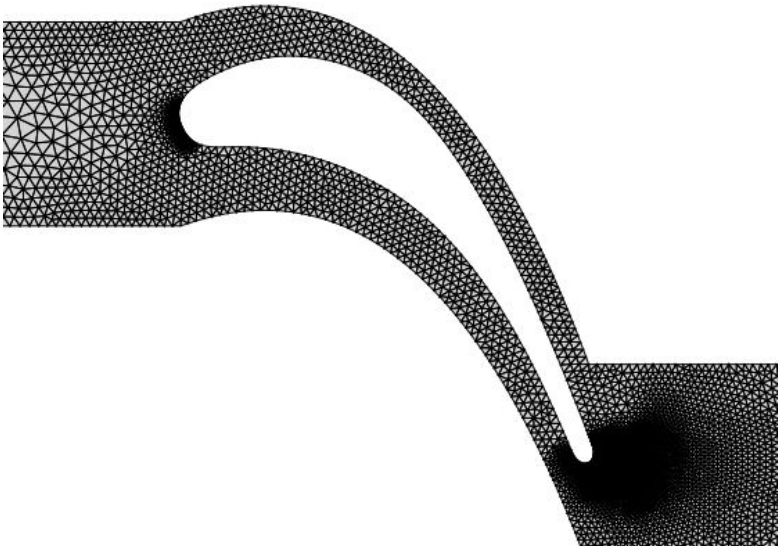


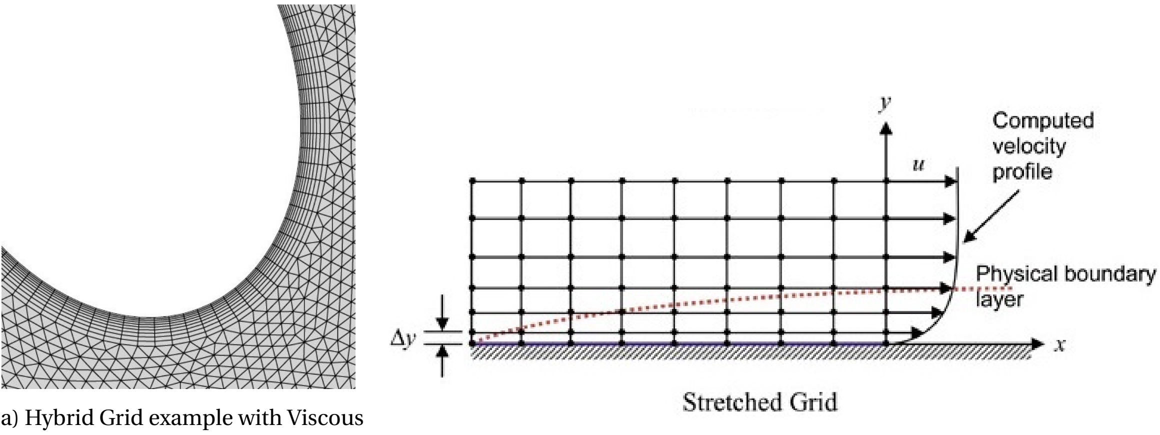
Figure 2.18: Unstructured Grid example, source: [14]

2.3.4. Hybrid Grids

A hybrid grids combines meshing algorithms of both structured and unstructured grids. It allows the meshing of complex geometries, while applying a local refinement using structured grids. In 2D meshes, this is a simple transition from Triangular elements to Quadrangular elements, see fig. 2.19a. In 3D meshes, however, a prism element (or sometimes pyramid element) is used to bridge the structured and unstructured meshes.

A common application of a hybrid grid is a global unstructured grid, with a local refined structured grid at the boundary layer location. A stretching algorithm is applied to the structured mesh to ensure that the velocity profile of the boundary layer can be properly simulated, see fig. 2.19b. This is often called a viscous layer grid, since it is mainly used in viscous computations where boundary layer development is present.

The development of the boundary layer is dependant on flow characteristics and surface conditions such as viscosity and wall shear stress. As the boundary layers develops, it transitions through different layers with different levels of viscous effects and turbulence effects. The y^+ parameter determines in which of these layers the mesh elements are located, and therefore in which layer flow computations are performed. A y^+ parameter of 1 indicates a high mesh-refinement which has elements within the lowest layer, the linear sublayer. A y^+ of 30 or higher has elements located primarily in the log-law layer. Further explanation on how the y^+ parameter determines the Δy, will be discussed in chapter 6.



(a) Hybrid Grid example with Viscous Layer, source: [14]

(b) Boundary Layer, with Viscous Layer (stretched) grid, source: [**?**]

Figure 2.19: Hybrid Grid examples

3 Programme Set-up of Design Tool

The thesis project output, the integrated turbomachinery design tool, is set-up as an extensive Python (2.7) programme, implementing the methods described chapters 4 to 6. The integrated design tool will be able to execute several aspects of the radial inflow turbomachinery design cycle. The design tool uses input from an external preliminary design software package. The integrated turbomachinery design tool THEN takes care of the CAD and Meshing operations. subsequently the the design tool provides input for an external CFD software package. For the execution of a full design cycle, one will make use of the following tools:

zTurbo An external 1D design tool in Fortran that Computes the Preliminary Design

Parapy An integrated turbomachinery design tool in Python consisting of:

- **OCC** A wrapper for the Parametric Design (CAD)
- **Salome** A wrapper for the Mesh Generation (Mesh)
- **SciPy** A Python library for the Design Optimisation (Optimisation)

SU2 An external open-source CFD solver in C++ that performs flow simulations of the Design geometry

zTurbo is a Fortran tool developed at the Delft University of Technology by Pini et al. [22], and will be used to quickly generate a preliminary design for a radial turbomachinery setup.

ParaPy is a software tool originally developed at the Delft University of Technology that provides a set of (Python 2.7) libraries that can apply CAD modelling and meshing in a python based environment. The CAD geometry creation will be realised using Open CASCADE (OCC) version 6.5, which is embedded in the ParaPy: Knowledge-Based Engineering Software Tool (ParaPy) software package. The Salome mesher module, SALOME 6.3, will take care of the meshing operations and is also implemented in the ParaPy software. The SciPy 0.18.1 python library is used for the optimisation and fitting algorithms used in the design tool.

SU2 is the open-source (C++) CFD solver that is used for all fluid flow simulations. For this thesis, SU2 version 4.3.0 is used.

Simultaneous to the development of the integrated design tool, an extensive documentation of the tool will be logged, using the built-in ParaPy documentation system. All the programming and documentation will be done on a Windows 10 operating system.

Since the tool will only be using theoretical calculations and analysis tools, the complete tool and its capabilities will be validated with geometry and perfomance data from the literature and other validated software tools. Examples of these validation cases are: reproducing the geometry and CFD results of the papers by Sauret and Jones. [11, 27]

Another tool that houses the capability of running a full design cycle is ANSYS. ANSYS is heavily used in industry, particularly for the design of turbomachinery blades. Therefore, for the application of the newly developed design tool, a comparison of a testcase, which has been previously conducted in ANSYS, will be done. This will serve both as a validation of the tool, as well as shows how the tool can be applied for performing a design cycle.

3.1. Design Tool Programme Setup

The main focus of the Design Tool, is to provide a flexible and accurate approach in modelling or parametrising all parts of radial inflow turbine. This includes the parametrisation of the stator, the rotor and the diffuser. Figure 3.4 below presents an UML diagram of the parametrisation classes used to create a full radial inflow turbine.

Next to the parametrisation phase, the design tool also provides capabilities for preliminary flow passage and geometry analysis and preliminary optimisation. These are defined in two separate packages. The UML diagrams of both packages are presented in figs. 3.1 and 3.2.

Finally design tool is able to generate a fluid domain and a corresponding mesh for the parametrised blades. The UML diagram of the Meshing package is shown in fig. 3.3.

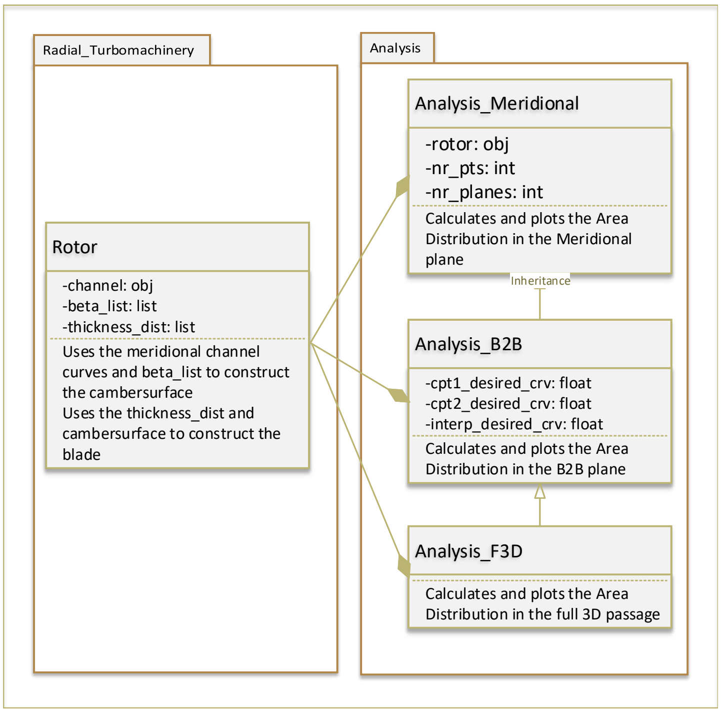

Radial_Turbomachinery
Analysis
Rotor
-channel: obj
-beta_list: list
-thickness_dist: list
Uses the meridional channel curves and beta_list to construct the cambersurface
Uses the thickness_dist and cambersurface to construct the blade
Analysis_Meridional
-rotor: obj
-nr_pts: int
-nr_planes: int
Calculates and plots the Area Distribution in the Meridional plane
Inheritance
Analysis_B2B
-cpt1_desired_crv: float
-cpt2_desired_crv: float
-interp_desired_crv: float
Calculates and plots the Area Distribution in the B2B plane
Analysis_F3D
Calculates and plots the Area Distribution in the full 3D passage

Figure 3.1: Analysis package

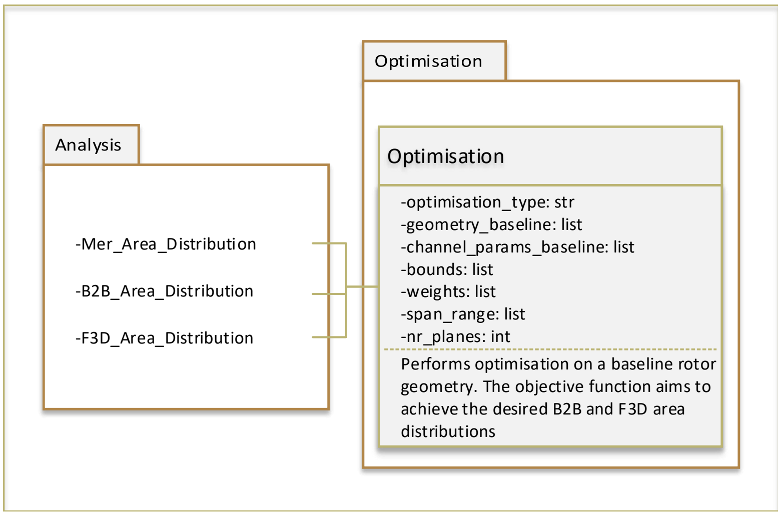


Figure 3.2: Optimisation package

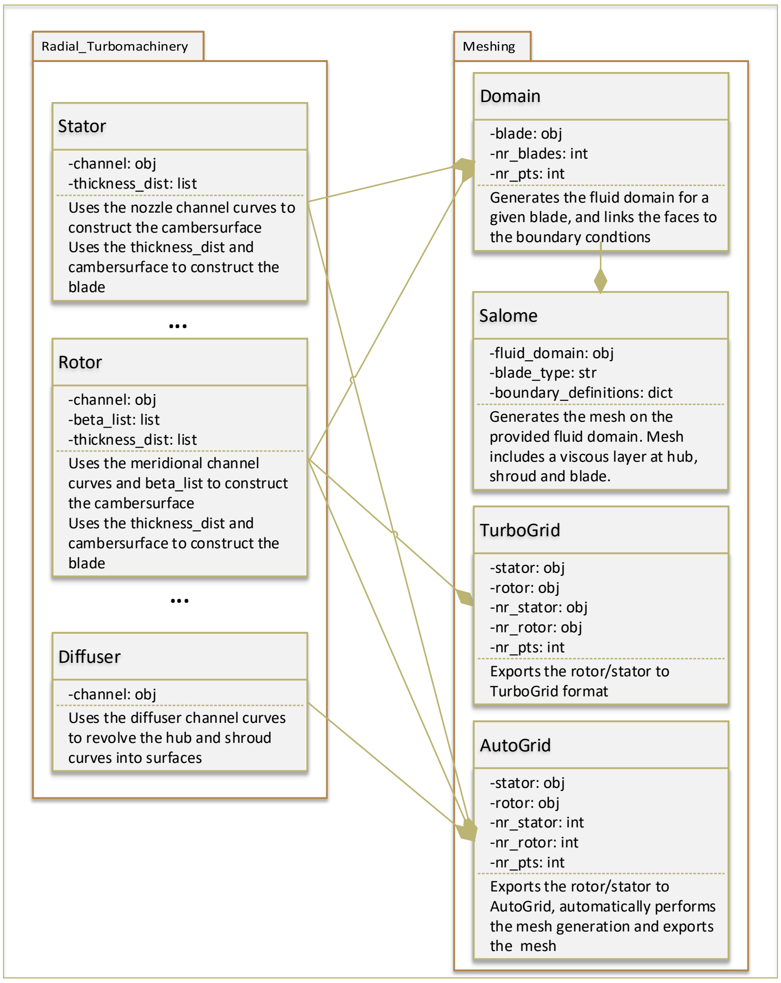

Radial_Turbomachinery
Stator
-channel: obj
-thickness_dist: list
Uses the nozzle channel curves to construct the cambersurface
Uses the thickness_dist and cambersurface to construct the blade
...
Rotor
-channel: obj
-beta_list: list
-thickness_dist: list
Uses the meridional channel curves and beta_list to construct the cambersurface
Uses the thickness_dist and cambersurface to construct the blade
...
Diffuser
-channel: obj
Uses the diffuser channel curves to revolve the hub and shroud curves into surfaces
Meshing
Domain
-blade: obj
-nr_blades: int
-nr_pts: int
Generates the fluid domain for a given blade, and links the faces to the boundary condtions
Salome
-fluid_domain: obj
-blade_type: str
-boundary_definitions: dict
Generates the mesh on the provided fluid domain. Mesh includes a viscous layer at hub, shroud and blade.
TurboGrid
-stator: obj
-rotor: obj
-nr_stator: obj
-nr_rotor: obj
-nr_pts: int
Exports the rotor/stator to TurboGrid format
AutoGrid
-stator: obj
-rotor: obj
-nr_stator: int
-nr_rotor: int
-nr_pts: int
Exports the rotor/stator to AutoGrid, automatically performs the mesh generation and exports the mesh

Figure 3.3: UML Class diagram of Meshing package

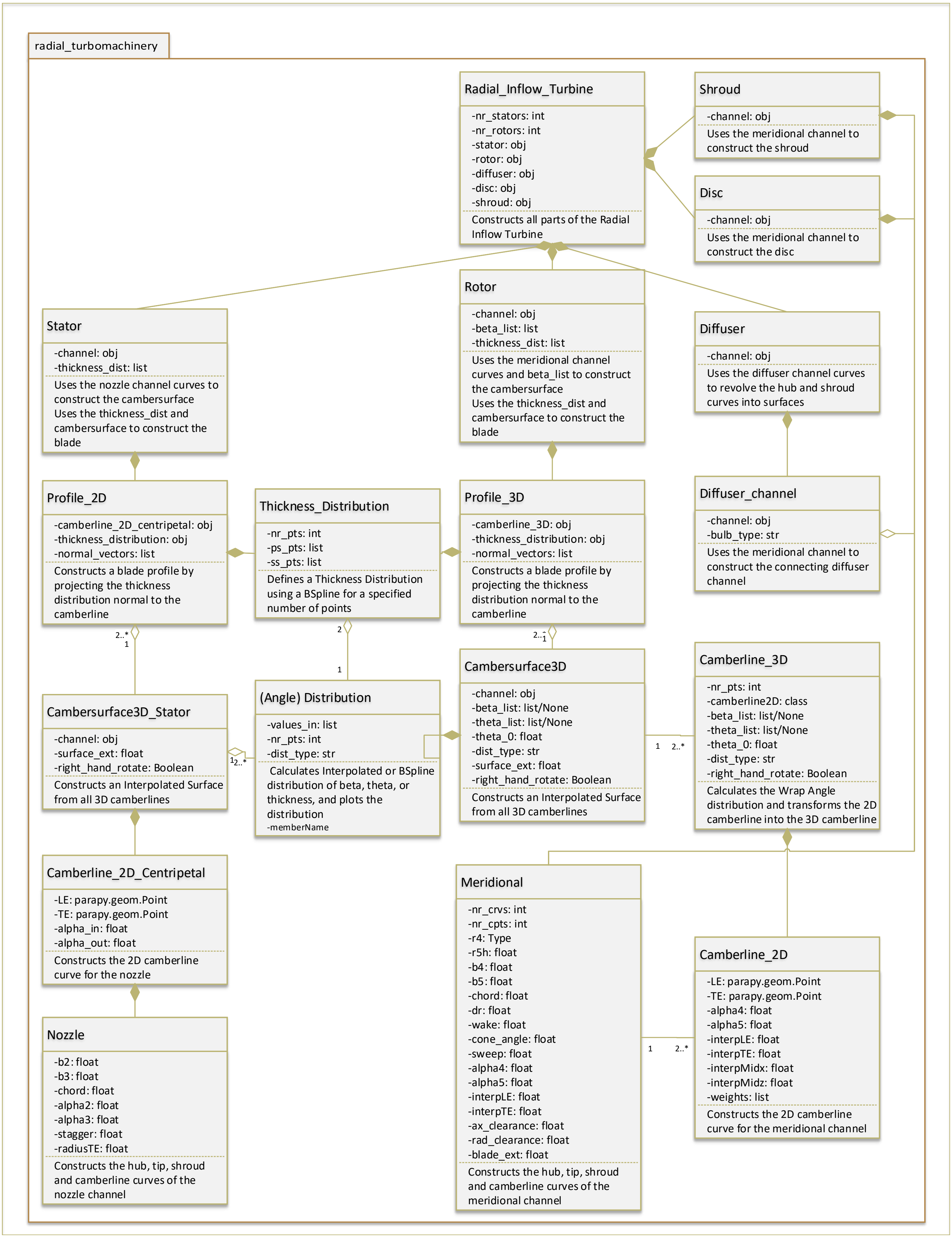

radial_turbomachinery
Radial_Inflow_Turbine
-nr_stators: int
-nr_rotors: int
-stator: obj
-rotor: obj
-diffuser: obj
-disc: obj
-shroud: obj
Constructs all parts of the Radial Inflow Turbine
Shroud
-channel: obj
Uses the meridional channel to construct the shroud
Disc
-channel: obj
Uses the meridional channel to construct the disc
Rotor
-channel: obj
-beta_list: list
-thickness_dist: list
Uses the meridional channel curves and beta_list to construct the cambersurface
Uses the thickness_dist and cambersurface to construct the blade
Stator
-channel: obj
-thickness_dist: list
Uses the nozzle channel curves to construct the cambersurface
Uses the thickness_dist and cambersurface to construct the blade
Diffuser
-channel: obj
Uses the diffuser channel curves to revolve the hub and shroud curves into surfaces
Profile_2D
-camberline_2D_centripetal: obj
-thickness_distribution: obj
-normal_vectors: list
Constructs a blade profile by projecting the thickness distribution normal to the camberline
Thickness_Distribution
-nr_pts: int
-ps_pts: list
-ss_pts: list
Defines a Thickness Distribution using a BSpline for a specified number of points
Profile_3D
-camberline_3D: obj
-thickness_distribution: obj
-normal_vectors: list
Constructs a blade profile by projecting the thickness distribution normal to the camberline
Diffuser_channel
-channel: obj
-bulb_type: str
Uses the meridional channel to construct the connecting diffuser channel
2..*
1
2
1
2..*
1
Cambersurface3D
-channel: obj
-beta_list: list/None
-theta_list: list/None
-theta_0: float
-dist_type: str
-surface_ext: float
-right_hand_rotate: Boolean
Constructs an Interpolated Surface from all 3D camberlines
Camberline_3D
-nr_pts: int
-camberline2D: class
-beta_list: list/None
-theta_list: list/None
-theta_0: float
-dist_type: str
-right_hand_rotate: Boolean
Calculates the Wrap Angle distribution and transforms the 2D camberline into the 3D camberline
1
2..*
Cambersurface3D_Stator
-channel: obj
-surface_ext: float
-right_hand_rotate: Boolean
Constructs an Interpolated Surface from all 3D camberlines
1
2..*
(Angle) Distribution
-values_in: list
-nr_pts: int
-dist_type: str
Calculates Interpolated or BSpline distribution of beta, theta, or thickness, and plots the distribution
-memberName
Camberline_2D_Centripetal
-LE: parapy.geom.Point
-TE: parapy.geom.Point
-alpha_in: float
-alpha_out: float
Constructs the 2D camberline curve for the nozzle
Meridional
-nr_crvs: int
-nr_cpts: int
-r4: Type
-r5h: float
-b4: float
-b5: float
-chord: float
-dr: float
-wake: float
-cone_angle: float
-sweep: float
-alpha4: float
-alpha5: float
-interpLE: float
-interpTE: float
-ax_clearance: float
-rad_clearance: float
-blade_ext: float
Constructs the hub, tip, shroud and camberline curves of the meridional channel
1
2..*
Camberline_2D
-LE: parapy.geom.Point
-TE: parapy.geom.Point
-alpha4: float
-alpha5: float
-interpLE: float
-interpTE: float
-interpMidx: float
-interpMidz: float
-weights: list
Constructs the 2D camberline curve for the meridional channel
Nozzle
-b2: float
-b3: float
-chord: float
-alpha2: float
-alpha3: float
-stagger: float
-radiusTE: float
Constructs the hub, tip, shroud and camberline curves of the nozzle channel

Figure 3.4: UML Class diagram of Radial Turbomachinery Parametrisation package

3.2. Workflow

In the workflow diagram below, an elaborate work flow of the design tool is presented. The workflow is divided into several function lanes, each shows the combined function or goal of the processes in that particular lane.

First the meridional channel or nozzle channel is defined for the rotor or stator respectively. Then this channel is transformed into a 3D Cambersurface using a Blade Angle or Wrap Angle distribution. Subsequently, using a Thickness Distribution, 3D blade profiles are constructed perpendicular to the cambersurface. A blade surface is formed using the 3D blade profiles. At this point, the rotor or stator has been created and the user can opt for creating a mesh of the fluid domain, or perform an analysis on the newly created geometry.

If the blade design is deemed acceptable, a 'Blade-Centered' fluid domain is created from the meridional channel and blade surfaces. This fluid domain is then meshed using a hybrid meshing alogrithm, which applies an ustructured grid, with an added viscous layer.

Finally this mesh will be exported to SU2, which performs the full 3D CFD simulations.

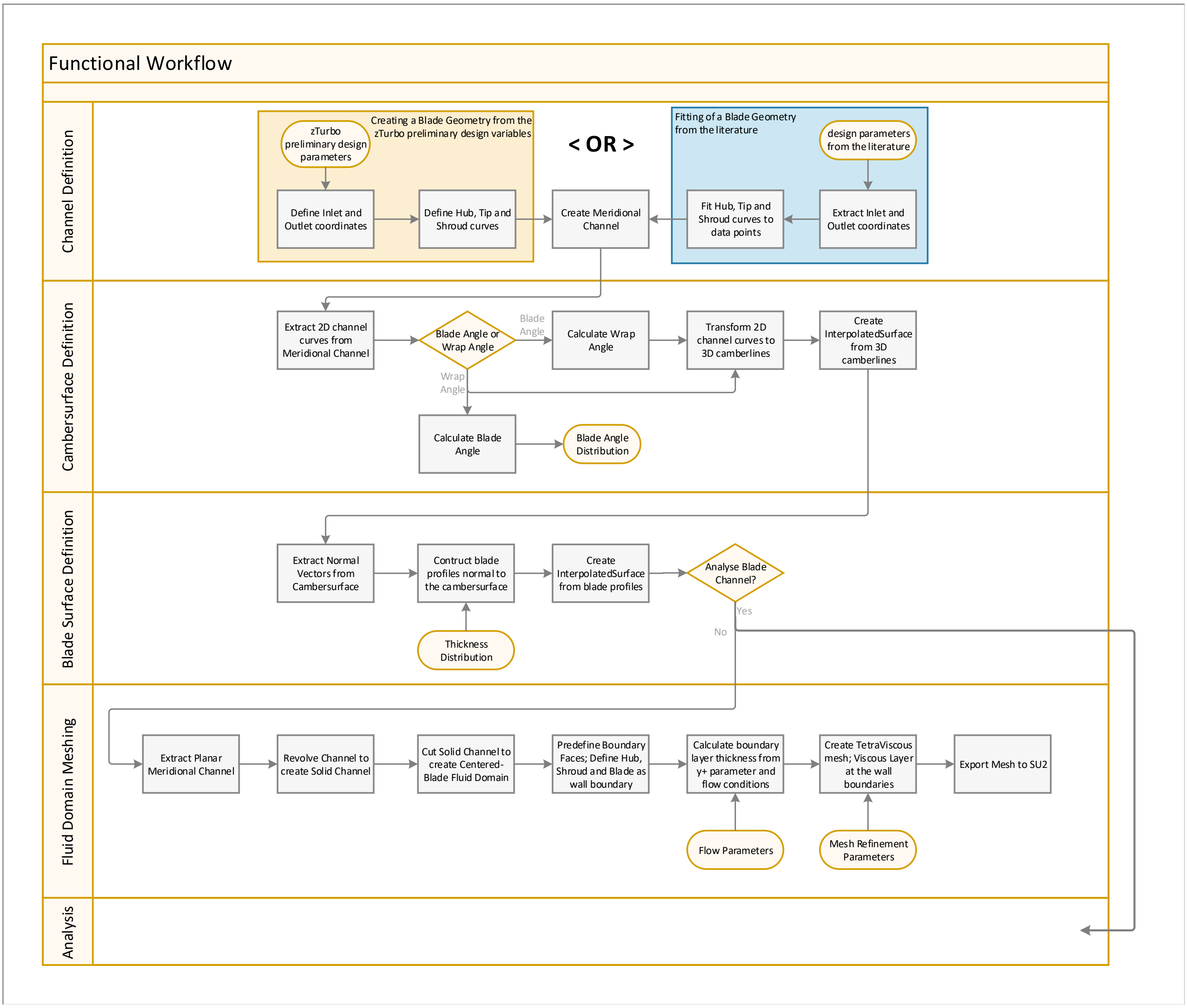

Functional Workflow
Channel Definition
Cambersurface Definition
Blade Surface Definition
Fluid Domain Meshing
Analysis
Creating a Blade Geometry from the zTurbo preliminary design variables
zTurbo preliminary design parameters
Define Inlet and Outlet coordinates
Define Hub, Tip and Shroud curves
< OR >
Create Meridional Channel
Fitting of a Blade Geometry from the literature
design parameters from the literature
Fit Hub, Tip and Shroud curves to data points
Extract Inlet and Outlet coordinates
Extract 2D channel curves from Meridional Channel
Blade Angle or Wrap Angle
Blade Angle
Wrap Angle
Calculate Wrap Angle
Transform 2D channel curves to 3D camberlines
Create InterpolatedSurface from 3D camberlines
Calculate Blade Angle
Blade Angle Distribution
Extract Normal Vectors from Cambersurface
Contruct blade profiles normal to the cambersurface
Create InterpolatedSurface from blade profiles
Analyse Blade Channel?
Yes
No
Thickness Distribution
Extract Planar Meridional Channel
Revolve Channel to create Solid Channel
Cut Solid Channel to create Centered-Blade Fluid Domain
Predefine Boundary Faces; Define Hub, Shroud and Blade as wall boundary
Calculate boundary layer thickness from y+ parameter and flow conditions
Create TetraViscous mesh; Viscous Layer at the wall boundaries
Export Mesh to SU2
Flow Parameters
Mesh Refinement Parameters

Figure 3.5: Flowchart of Workflow of the design tool; divided into function lanes

The geometry can be inspected and analysed first before performing CFD simulations. For the analysis several curves and surfaces are extracted from the blade geometry and the area distribution and curvature distribution of the flow passage is inspected. This analysis workflow can be used in a preliminary optimisation process, where the flow passage is optimised to a prescribed area distribution.

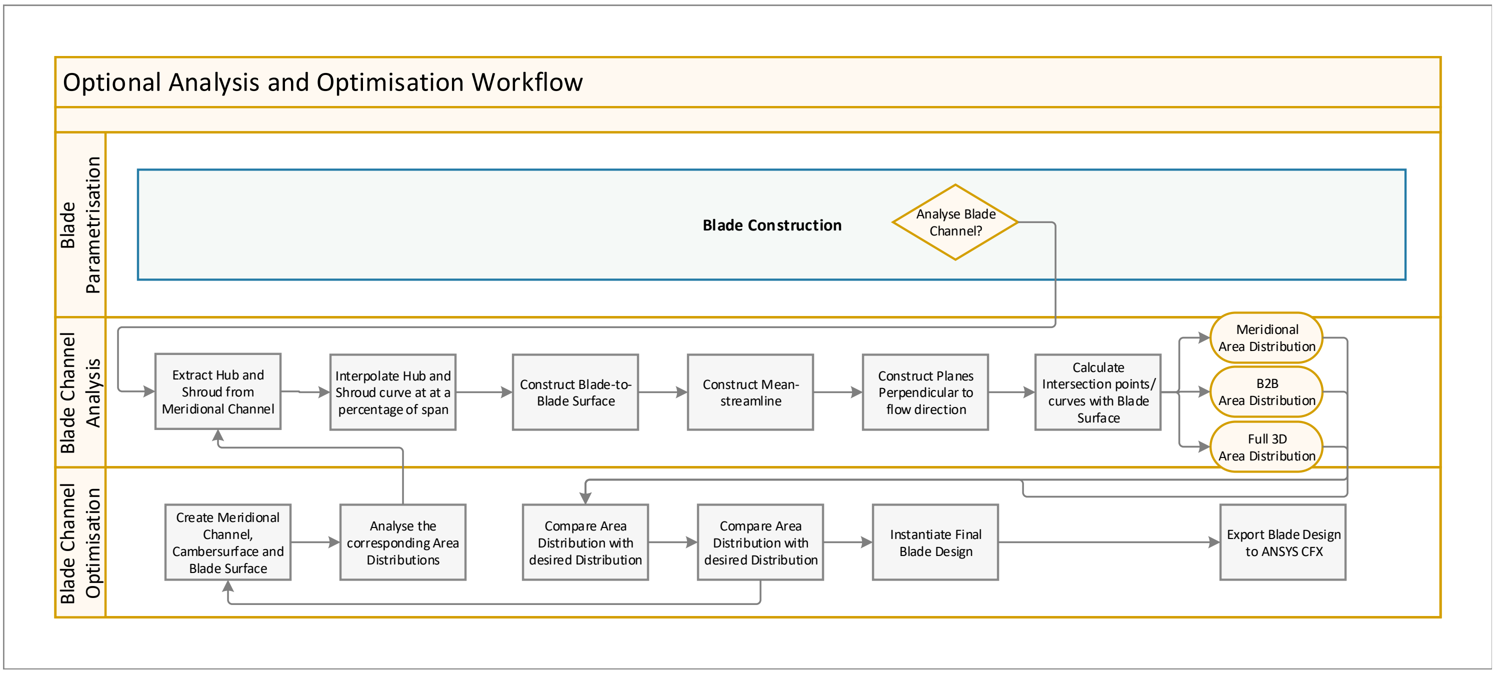

Optional Analysis and Optimisation Workflow
Blade Parametrisation
Blade Construction
Analyse Blade Channel?
Blade Channel Analysis
Extract Hub and Shroud from Meridional Channel
Interpolate Hub and Shroud curve at at a percentage of span
Construct Blade-to-Blade Surface
Construct Mean-streamline
Construct Planes Perpendicular to flow direction
Calculate Intersection points/ curves with Blade Surface
Meridional Area Distribution
B2B Area Distribution
Full 3D Area Distribution
Blade Channel Optimisation
Create Meridional Channel, Cambersurface and Blade Surface
Analyse the corresponding Area Distributions
Compare Area Distribution with desired Distribution
Compare Area Distribution with desired Distribution
Instantiate Final Blade Design
Export Blade Design to ANSYS CFX

Figure 3.6: Flowchart of Analysis and Optimisation Workflow of the design tool

4

Parametrisation of Radial Inflow Turbines

Once the preliminary design is completed, the parametrisation phase start. This is an important part of the turbomachine design cycle, because it not only converts the preliminary design parameters into the geometric CAD model, but it also prepares the geometry for any subsequent meshing and CFD analysis. The use of smart and flexible parameterization techniques ensures a solid foundation for the other design tool capabilities and helps reduce the duration of a full design cycle.

4.1. Cylindrical Coordinate System

The first parameterization decision made, was the conversion from the Carthesian Coordinate system to a Cylindrical (radial) Coordinate system. The Carthesian system is generally used in CAD modelling, because for the general public is the most intuitive coordinate system. For turbomachinery, however, it is not the most applicable coordinate system, due to the fact that turbomachinery blades and the flow around them are generally described in a rotating reference frame. It is therefore more suitable to apply a rotational coordinate system instead of a translational coordinate system. The (rotational) cylindrical coordinate system, as can be seen in fig. 4.1b, describes all Three-dimensional (3D) coordinates through the following three variables: r, θ, and z. Their relation to the Carthesian Coordinate system is described in eqs. (4.1a) to (4.1c).

$$x = r\cos\theta \tag{4.1a}$$

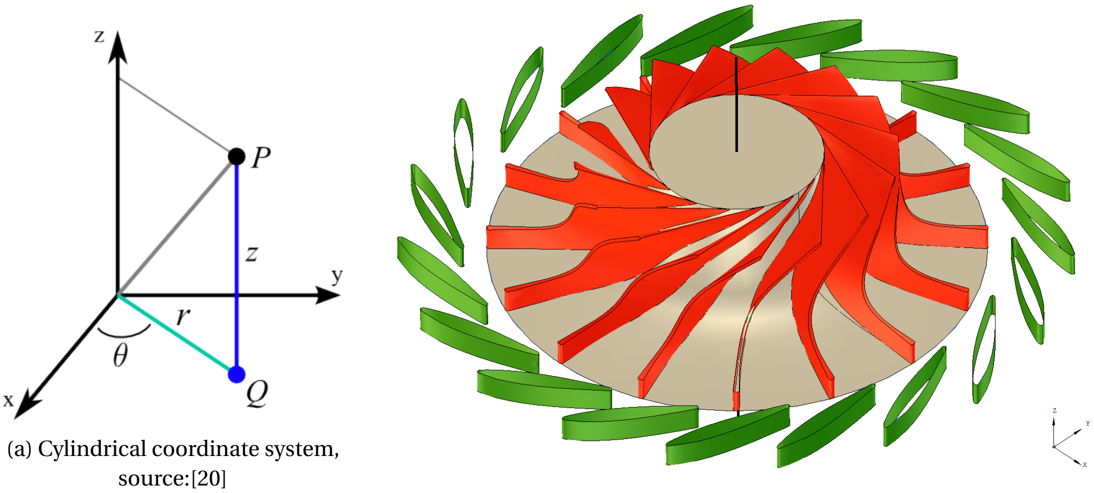
$$y = r\sin\theta \tag{4.1b}$$

$$z = z \tag{4.1c}$$

Therefore, within the design tool, two new class definitions were defined, see eqs. (4.2a) and (4.2b), that use r, θ, and z as an input and convert this to Carthesian coordinates for the ParaPy geometry-kernel to interpret. These definitions will be used throughout the rest of the report.

$$\text{cPoint}(r,\theta,z) \longrightarrow \text{Point}(\text{x},\text{y},\text{z}) \tag{4.2a}$$

$$\text{cVector}(r,\theta,z) \longrightarrow \text{Vector}(\text{x},\text{y},\text{z}) \tag{4.2b}$$



(a) Cylindrical coordinate system, source:[20]

(b) Rotating (around z-axis) system

Figure 4.1: Rotating Coordinate System

As will be evident further on, this coordinate system not only useful in defining the meridional channel and cambersurface of the rotor as a function of r, θ, and z, but it is also particularly useful in the definition and parametrisation of the centripetal stator.

4.2. Meridional Channel Definition

For the parametrisation of the meridional channel, the two main operations are the placement of the inlet and outlet boundares of the rotor and the subsequent placement of the hub, tip and shroud boundaries. The inlet is generally directed into the radial inward direction and the outlet is generally directed into the axial outward direction, but these orientations may be adjusted with the sweep and cone angle parameters respectively. The channel definition is started by defining the hub LE and TE coordinates.

4.2.1. Inlet and Outlet definitions

The leading and trailing edge coordinates of the hub, dependant on the r4, r5h and z-chord, are used as the basis from which the rest of the coordinates is computed. From these coordinate points, a vector with a magnitude of the local blade height and a direction of the local sweep or cone angle is used to define the blade tip coordinates. Similarly, the shroud coordinates are computed by increasing the vector magnitude with the axial or radial clearance, as can be see in fig. 4.2. An example of the computation of the shroud coordinates using the cylindrical reference frame, is given in eqs. (4.3a) and (4.3b) below.

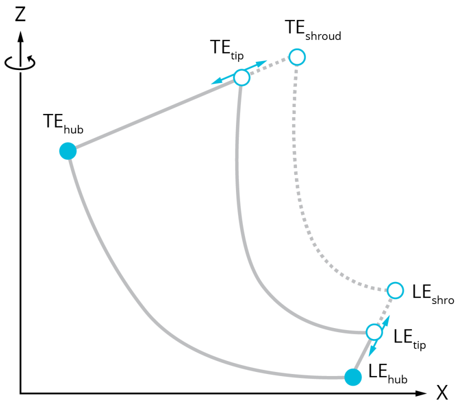


Figure 4.2: Definition of the Inlet, Outlet and tip clearance boundaries

$$LE_{shroud} = \text{cPoint}\,(r4_{hub}, \theta_0, z_0) + \text{cVector}\,(\cos(\Lambda_{cone}), 0, \sin(\Lambda_{cone})) \cdot (H_4 + \varepsilon_{ax}) \tag{4.3a}$$

$$TE_{shroud} = \text{cPoint}\,(r5_{hub}, \theta_0, z_c) + \text{cVector}\,(\sin(\Lambda_{cone}), 0, \cos(\Lambda_{cone})) \cdot (H_5 + \varepsilon_{rad}) \tag{4.3b}$$

4.2.2. Hub and Shroud Curve definition (2D)

While the Leading Edge and Trailing Edge boundary points have already been defined for the meridional channel, the particular channel is shape is defined by the camberline. Using the boundary points and α_{in} and α_{out} parameters, the inlet and outlet vectors were constructed. At the intersection of both vectors an intersection point was defined. A linear interpolation that uses an interpolation parameter between 0.0 and 1.0, was applied between the edge points and intersection point in order to define two intermediate control points. Using the edge points, the control points and a set of four weights, a Rational B-Spline Curve is constructed between the leading and trailing edge.

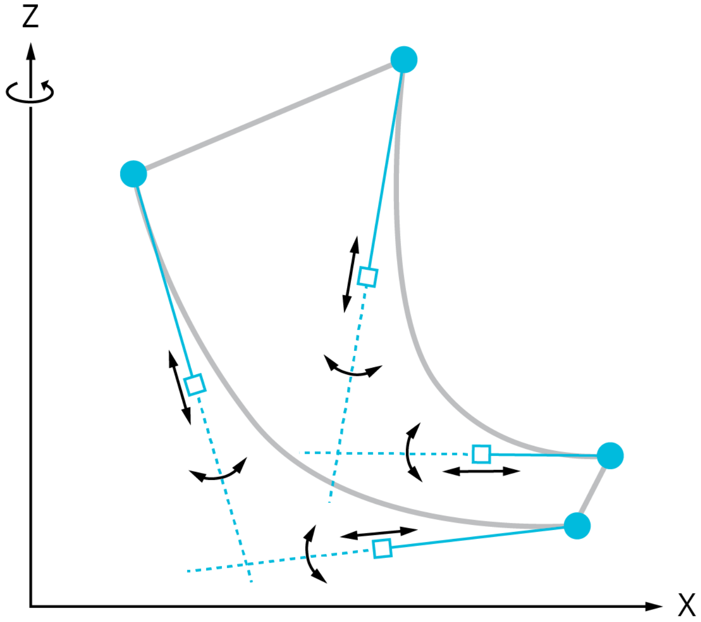
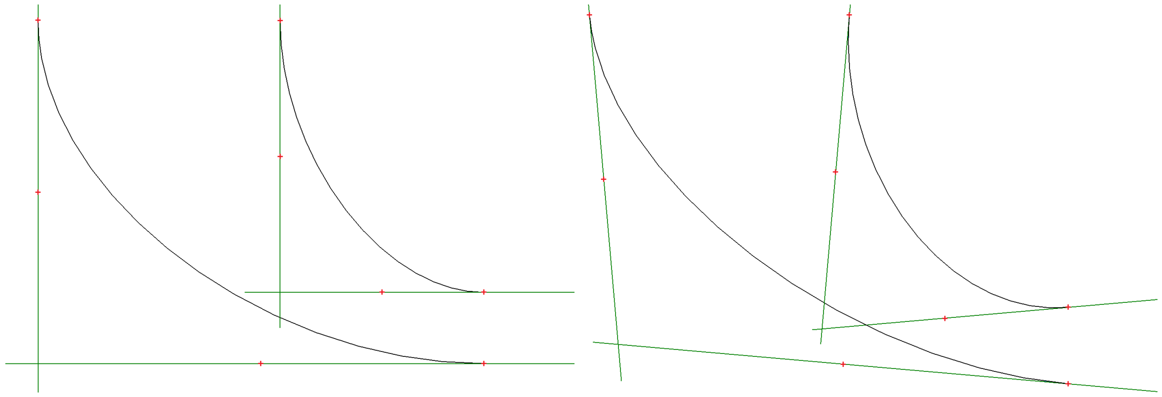


Figure 4.3: Camberline Control Point Construction

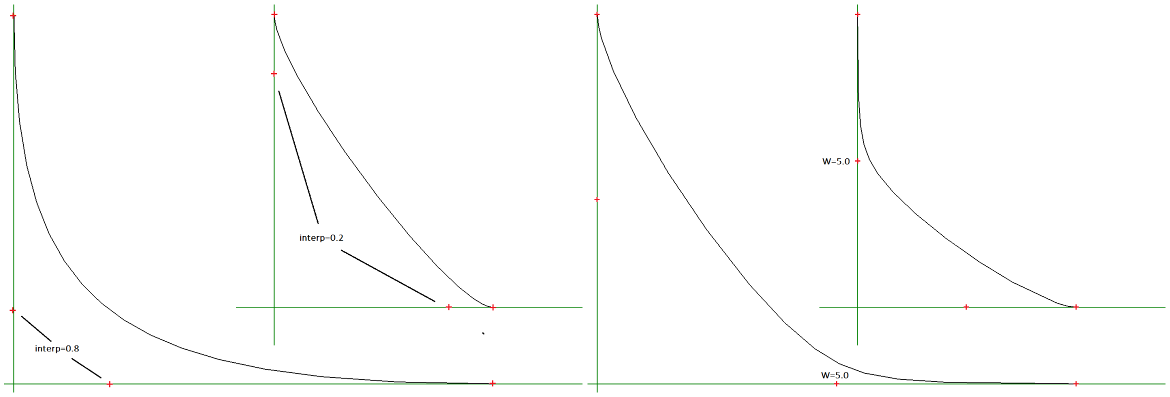
The NURBS (see section 2.2.1) ascertains that the directional vectors at the start and end of the camberlines curve have an angle of α_{in} and α_{out} respectively, by placing the two intermediate control points along the inlet and outlet vectors. The influence of the α_{in},α_{out}, weights and interpolation parameter are shown in figs. 4.4b and 4.5b.



(a) Initial Meridional Channel

(b) Influence of α_{in} and α_{out}

Figure 4.4: Influence of camberline parameters



(a) Influence of interpolation par.

(b) Influence of weights

Figure 4.5: Influence of camberline parameters

Next to these three curves, a minimum of two blade curves are defined in a similar way so that the blade boundaries slightly extend the hub and tip boundaries. If more than two curves are desired, a linear interpolation will be applied along the span-wise-directional inlet and outlet vectors, ensuring a straight (2D) Leading and Trailing Edge at all times.

4.3. Camberline definition (3D)

The numerical conversion of a 2D camberline into a 3D camberline requires a piece-wise rotation of the 2D camberline around z-axis with an angle of θ_i. This Wrap angle (θ) is geometrically the easiest to parametrise in CAD, but for preliminary turbine design, the blade angles give a much better indication of the influence on the fluid flow. The blade angle (or β) distribution is therefore the distribution to be used during parametrisation.

Construction of the 3D camberline is done by applying an axial rotation over an angle of theta at a discrete set of points along the 2D camberline. Note that for this transformation into the 3D camberline, the r and z coordinates remain the same as for the 2D camberline, while the theta changes along the curve to create a new set of control points. As previously stated, this particular transformation is much easier in a radial coordinate system.

$$\mathrm{cPoint}(r_{2D}, \theta, z_{2D})_i \longrightarrow \mathrm{Point}(x_{3D}, y_{3D}, z_{3D})_i \tag{4.4}$$

An example of a conversion from the 2D meridional channel camberline (red) to the 3D cambersurface camberline (yellow) can be seen in fig. 4.6.

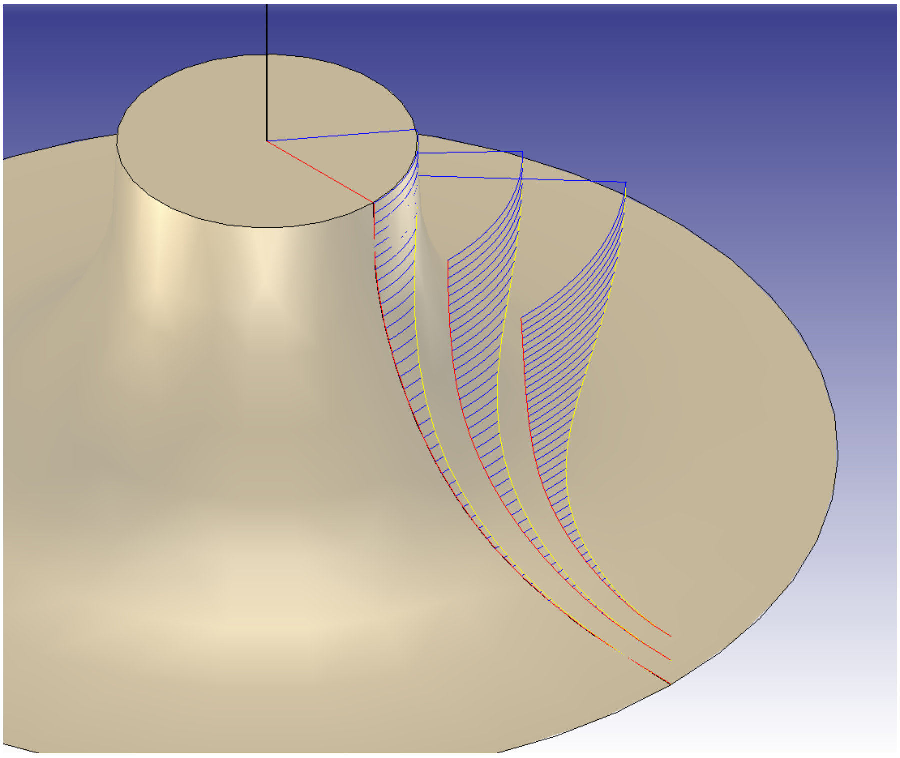
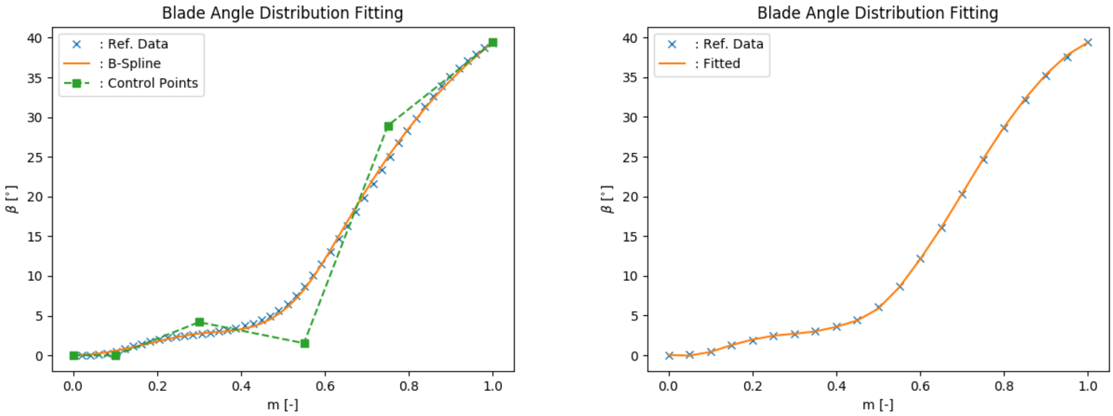


Figure 4.6: Conversion of 2D (red) to 3D (yellow) camberlines

4.3.1. Blade Angle to Wrap Angle

The beta-distribution can be either represented using a simple Bezier curve, for which a set of control points is determined, or by fitting a curve through the provided set of points (Interpolated Curve). An example of both types of fitting are given in figs. 4.7b and 7.4a respectively.



(a) B-Spline fitted beta distribution

(b) Interpolated beta distribution

Figure 4.7: Types of beta-distribution

The beta-distribution can be supplied by zTurbo and is converted into a wrap-angle distribution (θ-distribution), using the method as discussed in section 2.2.2. A numerical integration of eq. (2.5) is applied, resulting in a recursive formula as illustrated in eq. (4.5). The corresponding theta-distribution is shown in fig. 4.8.

$$\theta_i = \begin{cases} 0 & \text{if } i = 0 \\ \theta_{i-1} + \mathrm{d}\theta_i = \theta_{i-1} + \frac{\tan(\beta_{i-1})\cdot \mathrm{d}m}{R_i} & \text{if } i \geq 1 \end{cases} \tag{4.5}$$

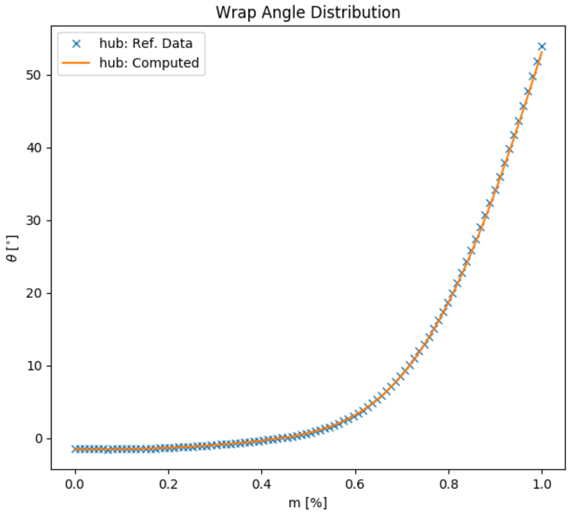
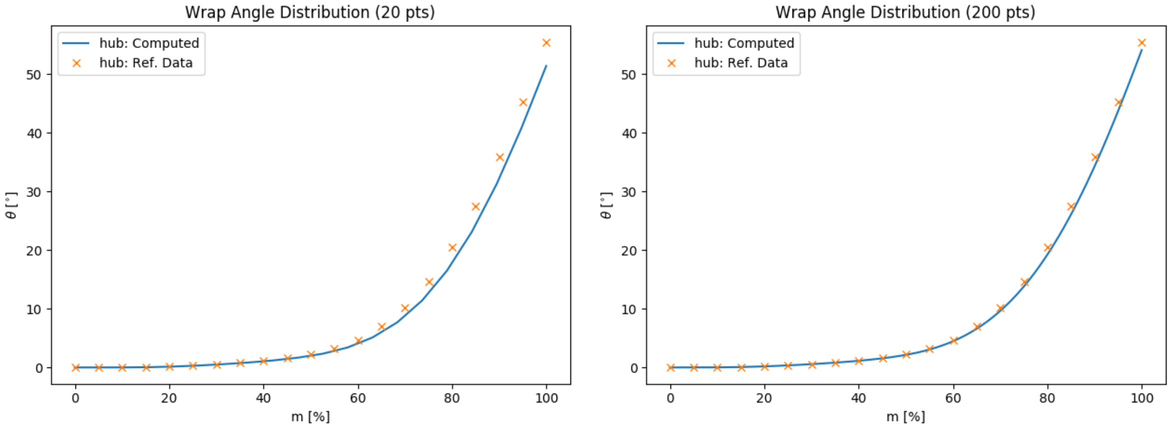


Figure 4.8: Computed theta distribution

By looking at the last few points in fig. 4.8, one can see that a very minor error is present. The theta-distribution method discussed above is prone to propagating errors along the meridional line, hence it affects the last few points in particular. Although the beta-distribution is provided as a continuous function, the numerical integration is performed discretely with a step size of dm. If a low discretisation is used of for example 10 points, the dm is relatively large, and the theta is only evaluated at 10 different beta angles, see fig. 4.9b. Due to the cumulative nature of the integration, the truncation error propagates along the full meridional line, resulting in a relatively high error compared to the analytically deduced answer. Increasing the discretization will reduce the total error significantly, as can be seen in fig. 4.9b, however, at the cost of added computational time. The computational time of subsequent CAD operations, particular the Solid operations, rises dramatically when increasing the discretisation of the wrap angle distribution. This is primarily due to the use of the cambersurface (with a lot of control points) as a cutting tool for solid partitioning operations.



(a) Theta distribution, low discretisation (nr. points=10)

(b) Theta distribution, high discretisation (nr. points=200)

Figure 4.9: Error in theta-distribution for low and high discretisation

Table 4.1 shows the computational time of ParaPy performing a Solid cutting operation for different levels of discretization. Taking both the error and increased computational time into consideration, a discretization in the range of 100 to 200 number of points was deemed optimal for an accurate yet reasonably quick optimisation. One could consider increasing or decreasing the discretisation when a more accurate or a faster optimisation is required.

Nr. of Points [-]	Rel. Error [%]	Comp. Time [ms]
10	14.178	7.00
20	7.350	11.0
50	3.976	33.0
100	2.960	66.0
200	2.471	256
500	2.184	671
1000	2.089	2158
2000	2.042	7963
5000	2.014	48475

Table 4.1: Discretisation level vs. Computational Time

Notice how the relative error converges to roughly 2% even at high discretisation. There seems to be an initial error, which is independent of the discretisation. This could be due to an inaccuracy in the fitted beta distribution (an example of the to-be-expected beta distribution for the reference wrap angle distribution will be shown in the next section.) The error of 2% is not very large and only signficant near the trailing edge, and will geometry-wise not be visible to the naked eye. It can however have an impact on the meshing and CFD analysis. It could particularly influence the outflow angle and with that the throat area of the flow passage. This should be taken account during the CFD analysis.

Generally, the beta distribution is supplied and the wrap angle distribution is computed as a result, however, in literature, sometimes the wrap angle distribution is also presented. The user can then opt for importing the wrap angle directly, avoiding any error that might arise through the conversion of blade angle to wrap angle. A example of a BSpline fitted directly to a wrap angle distribution is given in fig. 7.4b. Note that both angle distributions are defined along the length of meridional line of the 2D camberline.

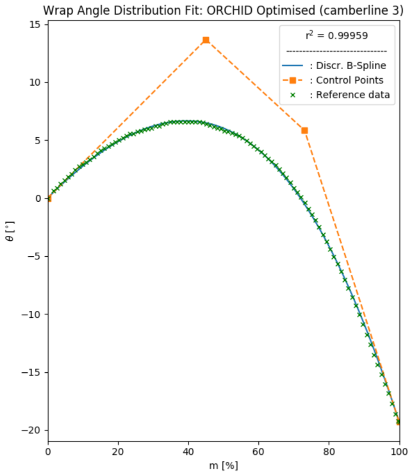


Figure 4.10: BSpline fitted theta distribution

4.3.2. Wrap Angle to Blade Angle

The conversion of the blade angle into the wrap angle is essential for the blade design tool, because the geometry parametrisation is primarily dependant on the wrap angle distribution. For analysis and research purposes, however, it is useful to be able to extract the blade angle distribution from a given wrap angle distribution, e.g. in order to reproduce the input parameters of a specific design. Therefore, when a wrap angle distribution is given, the corresponding blade angles are determined using the inverted formula of eq. (2.5), as illustrated in eq. (4.6). The numerical formula of this is given in eq. (4.7). Similar to the Blade Angle distributions, the provided Wrap Angle Distribution can be either a Bezier Curve or a Fitted Curve. An example of a reproduced Blade Angle Distribution from a given Wrap Angle Distribution can be found in fig. 4.11.

$$\beta = \arctan\left(\frac{R \cdot \mathrm{d}\theta}{\mathrm{d}m}\right) \tag{4.6}$$

$$\beta_i = \arctan\left(\frac{R_{i+1} \cdot \mathrm{d}\theta_{i+1}}{\mathrm{d}m}\right) = \arctan\left(\frac{R_{i+1} \cdot (\theta_{i+1} - \theta_i)}{\mathrm{d}m}\right) \tag{4.7}$$

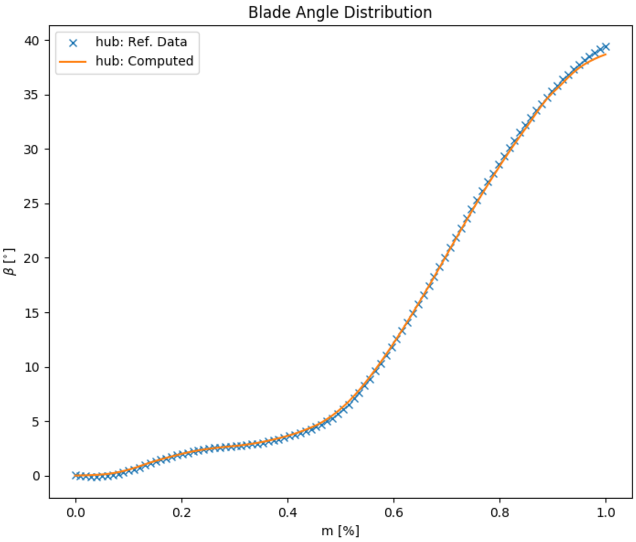


Figure 4.11: Blade Angle Distribution Computed from the provided Wrap Angle distribution

Similar to the computation of the Wrap Angle Distribution, this distribution computation shows some signs of propagation errors in the $m > 0.9$ region. While the curve tends to follow the rather complex shape of the reference data, yet again, an error is present near the trailing edge. This might even show that the reference beta distribution from the literature is incorrect, since it does not seem like a slowly propagating error, but rather an abrupt deviation in the $m > 0.9$ region. While it is useful to perform this back-ward computation as a sanity check, and to possibly find any errors in the provided data, the computed blade angles will not be used in the rest of the design phase. Any error present is therefore much less significant than those in the wrap angle distribution.

4.4. CamberSurface

The cambersurface is constructed from a set of 3D camberlines. As discussed in section 2.2.1, there are several methods of constructing a NURBS surface from a set of NURBS curves. The lofting operation will be avoided, since this operation only uses the control points of each B-Spline Curve and therefore does not ensure that the surface fully intersects with each curve of the set. An Interpolated BSpline Surface will be used ... The resulting cambersurface is not only necessary for the blade parametrisation, but is also used in analysis tools

4.5. Blade Profile

The parameterization of the blade profile requires both a camberline, as described in section 4.3 and a thickness distribution as input. Using both inputs, a thickness is projected onto the camberline, as discussed in chapter 2, to construct a 3D blade profile. The tool has the option to apply circular and elliptical shapes to the leading and trailing edges of the blade profile, however, this does show some curvature problems, as will be discussed below.

4.5.1. Thickness Distribution

Similar to the beta distribution, the thickness distribution can be defined using either a B-Spline constructed from a set of control points, or using a Interpolated Curve through a data set. The distribution is provided for each camberline.

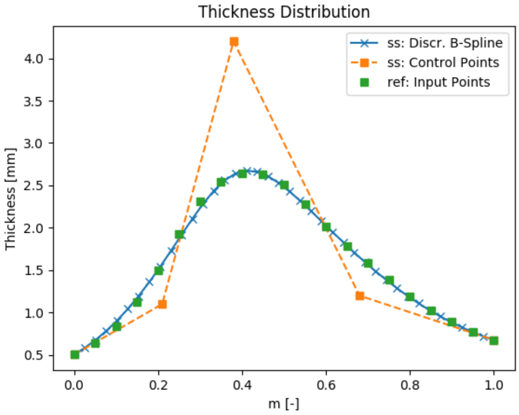


Figure 4.12: B-Spline thickness distribution

The resulting B-Spline thickness distribution will be discretized to a number of points. Along the camberline, an equal amount of equi-spaced points will be projected normal to the camberline, using the local normal vectors of the cambersurface. This ensures that the blade profiles are always perpendicular to the blade span-wise direction.

4.5.2. Leading Edge and Trailing Edge Construction

Blade profiles are often constructed using conical shapes such as circular or elliptical arcs at the blade edges. The design tool makes extensive use of the properties of NURBS curves (as described in chapter 2) for the construction of these conical shapes at the local leading or trailing edge. It was decided to use a circular arc with an angle of 180°, to make the derivative vectors parallel to the camberline at the transition point between the arc and the rest of the profile. The leading edge circular arc is constructed by constructing a 90° arc at both the pressure and the suction side of the profile.

The 90° arc is constructed using a isosceles triangle of control points with a 90° angle an isosceles sides of length equal to the arc radius. The middle control point has a weight of $\frac{\sqrt{2}}{2}$, to ensure the circular shape.

The elliptical arc is constructed using the same method, however, the side of the control point triangle that is parallel to the camberline is extended and its length is equal to the sei-major axis of the ellipse.

In order to connect this arc properly to the rest of the NURBS curve blade profile, the control point triangle is added to the set of control points, and the knot-vector of the curve is adjusted in such a way, that the multiplicity of the second knot (located at the transition point) is elevated to a level equal to that of the degree of the curve.

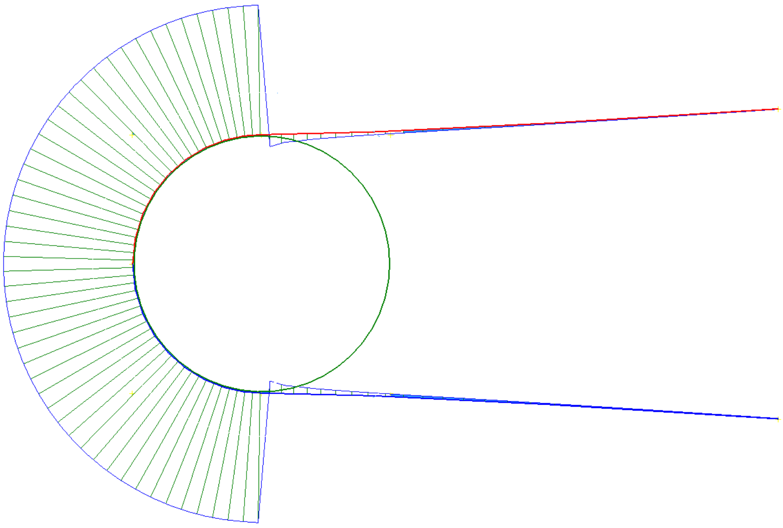


Figure 4.13: Circular Leading Edge

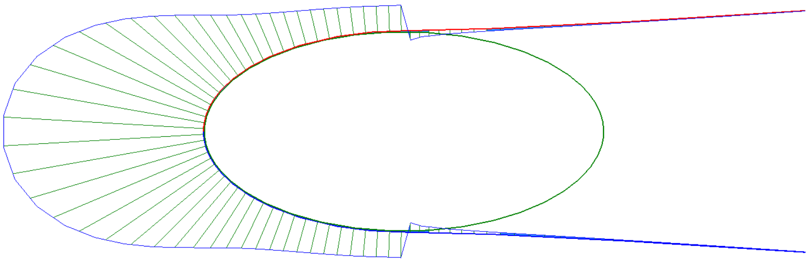


Figure 4.14: Elliptical Leading Edge

Note that, when the mulitplicity of the knot, and the degree of the curve are equal, the basis function is only C0 continuous. Figures 4.13 and 4.14 above show a C1 continuous curve, by using proper control point placement. However, the curvature distribution projected over the profile still shows discontinuities. Ensuring proper control point placement can reduce this even further.

4.5.3. Continuous curvature (CP placement)

According to Korakianitis, by ensuring continuous curvature (C3 continuity) through correct Control Points placement, the several flow characteristics such as separation or high flow acceleration can be avoided along the blade profile. This gives a better blade loading along the rotor by avoiding big pressure spikes. As can be seen in figs. 4.13 and 4.14, the curvature is not continuous at the transition points between the conic shapes and the rest of the profile.

The method used to connect the arc and the profile reduces the continuity at the knot to C0 continuity. This means that the three subsequent control points are constraint in their placement, to ensure a C3 continuity of the curve. At the transition point, the first, second and third derivative of both NURBS curves are defined. These derivatives should be equal for both curves to ensure the continuity up to the third degree. In the equations below, the derivatives are expressed as a function of their control points.

$$C'(0) = \frac{p}{u_{p+1}}(P_1 - P_0) \tag{4.8a}$$

$$C'(1) = \frac{p}{1 - u_{m-p-1}}(P_{n-1} - P_n) \tag{4.8b}$$

$$C''(0) = \frac{P(P-1)}{u_{p+1}} \left(\frac{P_0}{u_{p+1}} - \frac{(u_{p+1}+u_{p+2})}{u_{p+1}} P_1 + \frac{P_2}{u_{p+2}} \right) \tag{4.9a}$$

$$C''(1) = \frac{P(P-1)}{1-u_{m-p-1}} \left(\frac{P_n}{u_{p+1}} - \frac{(u_{p+1}+u_{p+2})}{u_{p+1}} P_{n-1} + \frac{P_{n-2}}{u_{p+2}} \right) \tag{4.9b}$$

$$C'''(0) = \frac{n(n-1)(n-2)}{u_{p+1}} \left(\frac{1}{u_{p+1}} P_3 - 3\frac{(u_{p+1}+u_{p+2})}{u_{p+1}} P_2 + 3\frac{(u_{p+1}+u_{p+2})}{u_{p+1}} P_1 + \frac{1}{u_{p+2}} P_0 \right) \tag{4.10a}$$

$$C'''(1) = \frac{n(n-1)(n-2)}{1-u_{m-p-1}} \left(\frac{1}{u_{p+1}} P_{n-3} - 3\frac{(u_{p+1}+u_{p+2})}{u_{p+1}} P_{n-2} + 3\frac{(u_{p+1}+u_{p+2})}{u_{p+1}} P_{n-1} + \frac{1}{u_{p+2}} P_n \right) \tag{4.10b}$$

By setting $C'(0) = C'(1)$ for all three derivatives, the P_0, P_1, P_2 and P_3 control points after the leading edge transition can be computed, based on the last 4 control points before the transition. fig. 4.15 below shows a joining of a leading edge ellipse with a subsequent mid-section curve. The curvature distribution is projected over this composed curve and seems to be fully continuous. The graph depicted in the figure, shows the curvature and the slope of curvature along the composed curve. This shows that the curve is nearly third-degree continuous, but shows a light kink in slope of curvature. This kink is so small, however, that this discontinuity is most likely due to numerical errors in ParaPy, or perhaps the joining method used to combine the two curves.

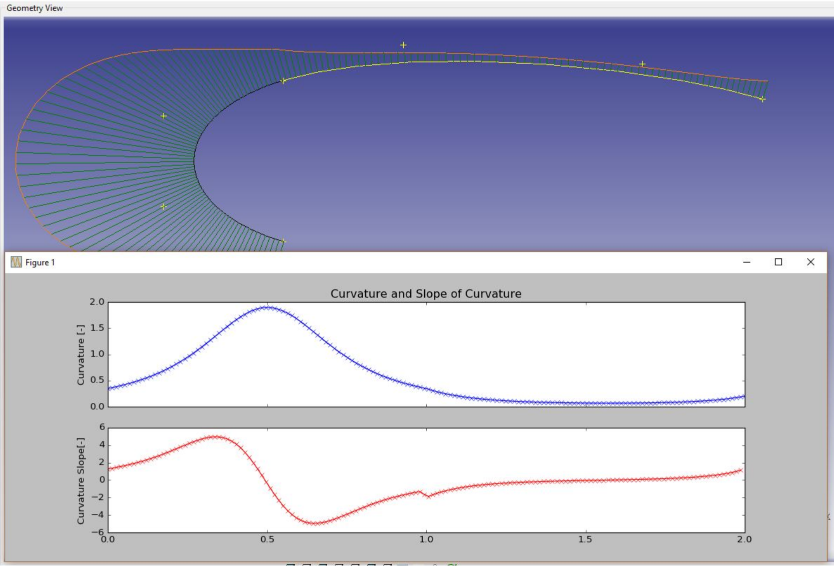


Figure 4.15: Continuous Curvature Piece-wise curves

While this methods seems to help in reduce the discontinuity in the curvature, the constraints applied to the subsequent 4 control points after the transition constrain the mid-section curve in such a way that it becomes very difficult or nearly impossible to apply this to a relatively straight blade with a low stagger angle. Some more investigation should be conducted on the set of constraints imposed on the curve when using this method, to see if the challenges discussed can be overcome.

4.6. 3D Rotor

For the construction of the full 3D rotor, the blade profiles are discretised and an Interpolated Surface is created through all of the discretized points. This ensures a proper fit of the blade surface through each of the profiles.

The profiles have been placed in such a way, that the blade surface extends slightly beyond the hub and shroud curves. These curves are used as a rotating cutting tool to cut away slices of the extended blade surface. This ensures that the resulting blade surface has constant radius at the hub and tip profiles, which ensures that the blade surface can be placed on a disc or inside a shroud. To model the tip-clearance of the rotor, the tip curve will be used as the cutting tool, to cut away the tip of the blade surface.

4.7. 3D Stator

While the rotor is by far the most complex geometry to be parametrized, also for the centripetal stator several smart parametrisation methods were needed. Even though the stator is a non-rotating part, the cylindrical coordinate system is still applied. This is because in the stator nozzle, the fluid is moving in a (centripetal) rotational reference frame, where the inflow and outflow angles are dependent on the radius and radial chord length ($Chord_{rad}$) and stagger angle.

Since the interspace parameter (Δr) has quite some effect on the flow interaction between the rotor and the stator, the Trailing Edge of the stator is defined at a distance of $r_4 + \Delta r$ from the origin. The Leading Edge is then determined at the cross-section of a circle with a radius of $r_{Te} + Chord_{rad}$ and the stagger line directed from the Trailing Edge. In this coordinate system, the stagger angle is defined with respect to the tangent at the Leading Edge.

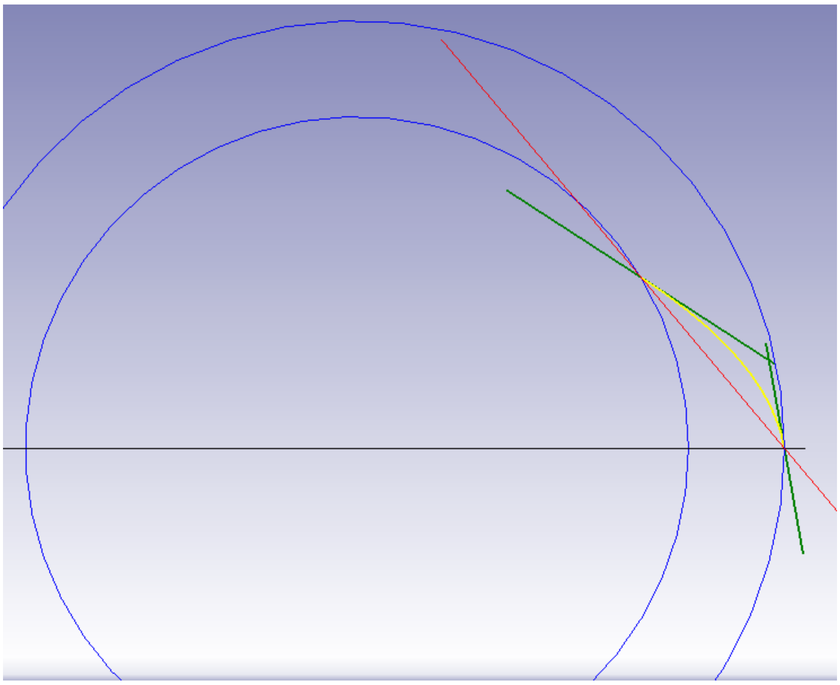


Figure 4.16: Nozzle channel definition of the Stator, in the Centripetal Frame

Camberlines are defined using a BSpline curve through three control points: the Leading Edge point, the intersection of the inflow and outflow lines, and the Trailing Edge point. The use of the intersection point as a control point, ensures that the start and end of the camberline curve are tangential to the inflow and outflow lines.

Using the same method as for the parametrisation of the rotor, the blade profiles of the stator are constructed using the camberlines, the normal vectors of the cambersuface and a thickness distribution. These blade profiles are then used to generate the 3D blade surface of the stator, as can be seen in fig. 4.17

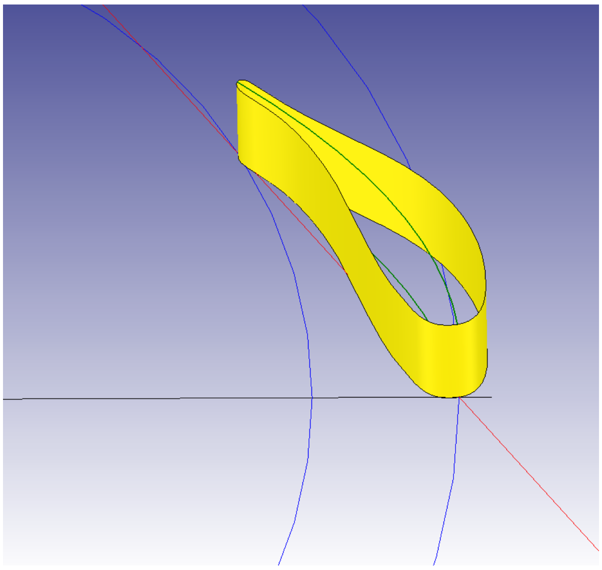


Figure 4.17: Definition of the Stator, in the Centrifugal Frame

4.8. Diffuser

The diffuser is parametrised by extending the wake at the rotor outlet and connecting the hub curve with a bulb curve, and connecting the shroud curve with a diverging diffuser wall curve. The shroud curve of the diffuser is a simple BSpline curve that starts off, tangent to the shroud curve of the rotor and then slowly transitions to a straight line with a specified wedge angle, as can be seen in figure fig. 4.18.

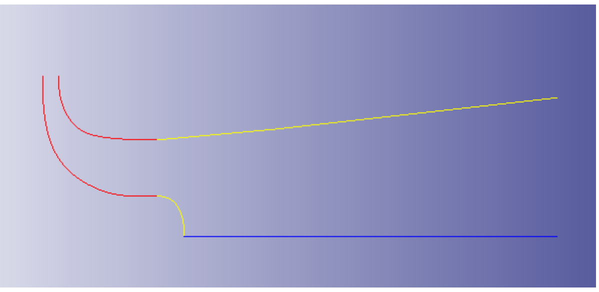


Figure 4.18: Diffuser Channel Definition

The bulb at the hub curve is slightly more complex. Due to the many different types of bulb that are applied in industry, four options were created to parametrise the hub hub at the bulb. Two types of bulb offer a relative smooth transition from the rotor wake into the diffuser area, while the other two are more abrupt. Each type is described and presented in table 4.2 and fig. 4.19.

Bulb Type	Description
Rounded	Single Smooth transition at rotor wake towards zero-radius symmetry point
Smooth	Double smooth transition from rotor wake towards zero-radius symmetry point
Corner	Abrupt transition from rotor wake towards zero-radius symmetry point (90 degree corners)
Sharp	Abrupt transition from rotor wake towards zero-radius symmetry point (60 degree corners)

Table 4.2: Diffuser bulb types

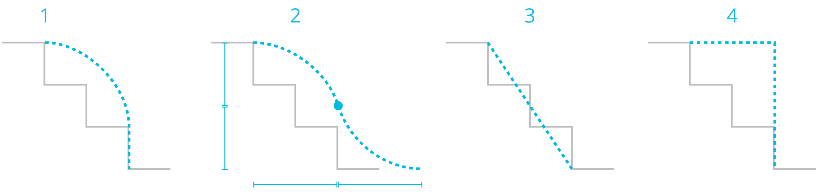


Figure 4.19: Diffuser bulb geometry options

Diffuser: Addition of hub and shroud curve after the rotor wake Option to select different type of bulb (1-4) Running simulations to see if it provides more accurate result Currently only mesh generated using AutoGrid

Conclusion: Diffuser is a simple addition that could greatly improve CFD analyses Need to do some more research into to level of increased accuracy

4.9. Blade Geometry Fitting

The parametrisation methods described above provides all the tool to generate almost any feasible turbine new design, completely from scratch. However, the design tool does not only provide the means to design a turbine from scratch, but it is also able to accurately reproduce current turbine designs given a certain set of input parameters and distributions. The fitting procedures that ensure that a fully reproduced turbine can be generated from a given data set, will be explained below.

Using an initial guess of the parameters, an initial camberline is constructed in ParaPy. Subsequently, the 'minimum_distance()' method of the constructed curve is applied to each point in the respective data set. The distances for each point are summed to give the function value. A minimisation algorithm ('SLSQP') is used change the input parameters in order to minimize the distance between the camberline curve and the data set points. Once the minimization is complete, the current camberline and data set are compared and the 'r^2' parameter is calculated using eq. (4.11)

$$\bar{y} = \frac{1}{n}\sum_{i=0}^{n} y_i \tag{4.11a}$$

$$SS_{tot} = \sum_{i=0}^{n} (y_i - \bar{y})^2 \tag{4.11b}$$

$$SS_{res} = \sum_{i=0}^{n} (y_i - f_i)^2 = \sum_{i=0}^{n} e_i^2 \tag{4.11c}$$

$$r^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{4.11d}$$

4.9.1. Meridional channel fitting

First, the 2D fitting of the meridional channel will be considered. For reproduction purposes, a data set of the hub and shroud curves is often provided in the literature. This data set, generally defined in the R-Z plane is used for the fitting procedure. The inlet and outlet boundaries are parametrised using several geometric parameters such as r4, b4, r5h, b5, etc. These parameters are generally given for a turbine, but can also be extracted from the provided meridional channel data set by looking at the start and end coordinates of the curves. The camberline curves are parameterised with several parameters that cannot be directly extracted from the data set. Therefore, a quick fitting algorithm needs to be applied to fit the: α_{in},α_{out}, weights and interpolation parameters.

4.9.2. Distribution fitting

The fitting of a Wrap Angle, a Blade Angle Distribution or even a Thickness Distribution is very similar to the Meridional Channel fitting. Instead of producing the camberline from the input parameters, a Bezier Curve is generated from an initial guess of control points. Note that, due to the nature of the Bezier Curve, the initial and the final control point should coincide with the initial and final point of the distribution; these points are therefore extracted from the data set up front. The fitting of the distribution is a universal method, but it allows to generate both a proper cambersurface as well as a proper blade profile, depending on which type distribution is fitted.

4.9.3. CAD geometry fitting (check)

If provided, a CAD geometry file (.stl, .stp, .igs) can be imported into ParaPy. A scaling factor is applied for easy conversion of metric units, which sometimes differ between the CAD files. Once properly scaled, the CAD model and the parametrised model can be visually compared. It is a relatively quick final check to verify the fitting processes have been performed correctly, before the parametrised model goes through the time-consuming meshing and CFD phases.

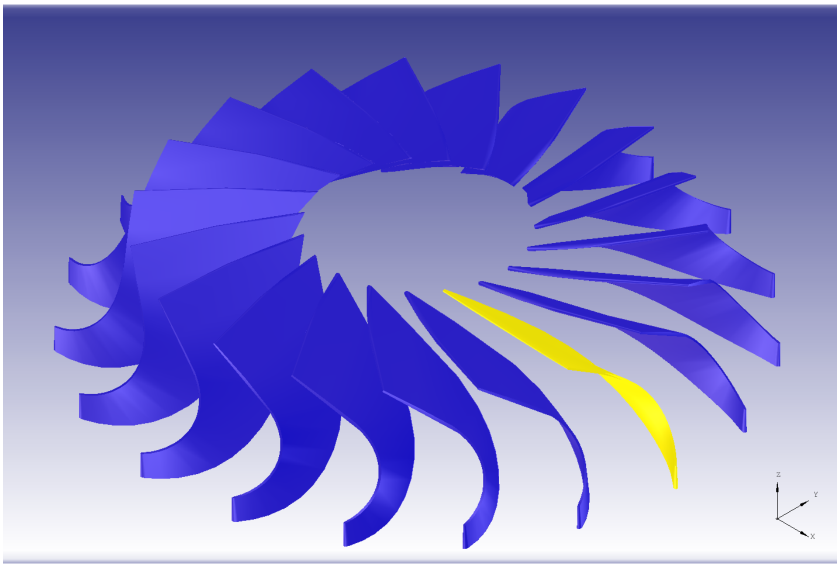


Figure 4.20: CAD Model fitted to parametrised model for visual verification

5

Preliminary Geometrical Analysis of Parametrised Turbine Blade

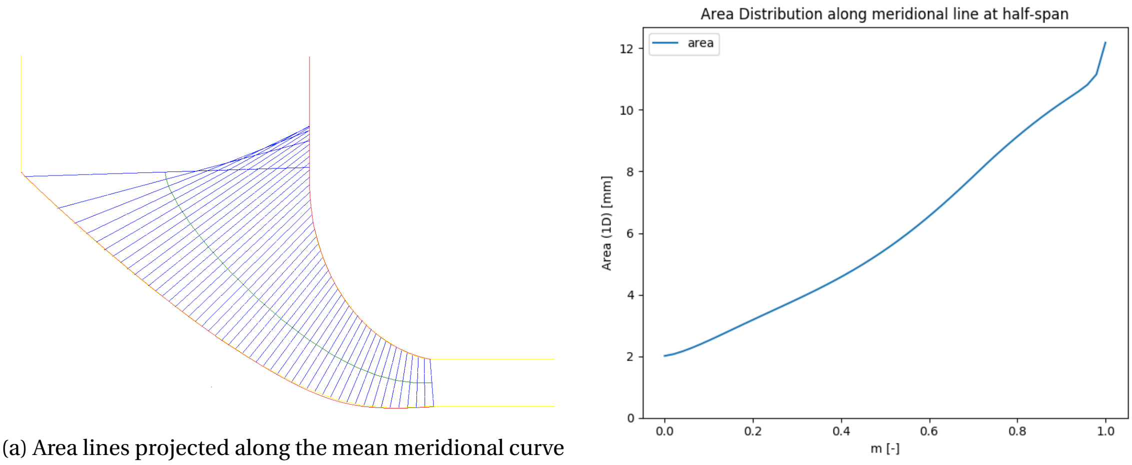
One a geometry is fully constructed, the designers may find it useful to inspect the geometrical properties of the blades, before resorting to a very time-intensive CFD analysis to analyse the full flow conditions. For the purpose of increasing the feedback of the design during the preliminary design phase, several analysis tools were added to the Optiblade design tool. The designers can make use of these analysis tools to further increase the performance of the preliminary design before going to the high fidelity optimisation phases. Alternatively, one could use these tools on a pre-existing design to evaluate the geometrical properties to search for areas of improvement. Analysis tools for the meridional channel, the Blade-to-Blade channel and the full 3D geometry are presented in sections 5.1 to 5.3 below.

5.1. Meridional Analysis

The Meridional channel design has a great impact on the overall performance of the turbine. The channel does not only increases the meridional (1D) flow area as seen by the flow, to accommodate for a proper mass flow from inlet to outlet , it also curves the flow from the radial direction into the axial direction. Therefore, two particular geometric characteristics of the channel are of interest: the meridional area distribution and the curvature distribution.

5.1.1. Area Distribution

In order to calculate the area distribution of the meridional channel, first the mean meridional line is determined using a linear interpolation of the control points and weights of the hub and shroud curves. Along this mean meridional line, a specified number of equally spaced planes are placed; they are oriented perpendicular to the meridional line direction. These planes represent the 1D frontal area as seen by the flow, when moving through the meridional channel. The intersections between the planes and the hub and shroud curves define the area lines within the meridional channel, see fig. 5.1a. The length of these area lines can be plotted against the percentage of the meridional line, as can be seen in fig. 5.1b.



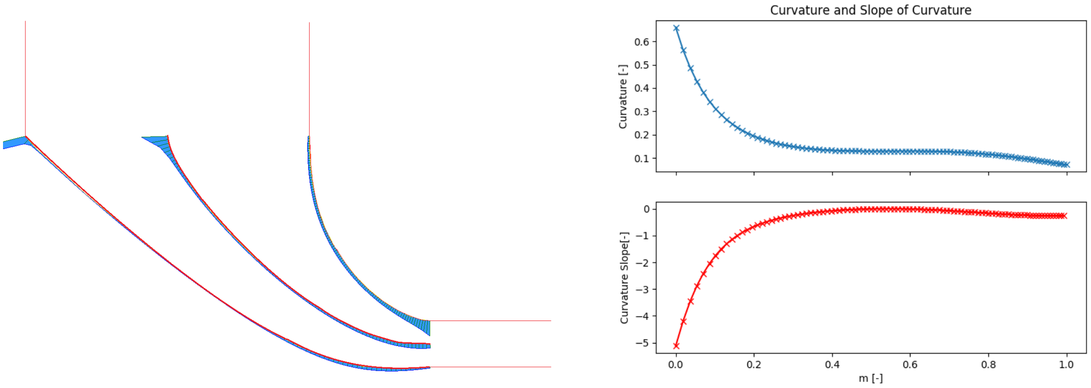
(a) Area lines projected along the mean meridional curve

(b) Area Distribution along the mean meridional curve

Figure 5.1: Meridional Area Distribution Analysis produced in ParaPy

5.1.2. Curvature Distribution

The curvature distribution of the meridional channel presents the level of curvature at the hub curve, the mean meridional curve, and the shroud curve. The curvature at a specified number of points along each curve is determined using a ParaPy-build-in 'curvature_at_point()' method. Then, in order to properly visualize this geometrical characteristic, several curvature lines are projected perpendicular to the curve, the length of which is directly related to the level of curvature at that particular point. The curvature and its derivative can also be plotted against a percentage of the meridional line, to quantitatively inspect the curvature and check for irregularities in the curvature-derivative, which might instantiate local separation in the channel. An example of the projection and plot are given in figs. 5.2a and 5.2b



(a) Curvature projected on hub, mid and shroud curves

(b) Curvature and Slope of Curvature along the shroud curve

Figure 5.2: Meridional Curvature Analysis produced in ParaPy

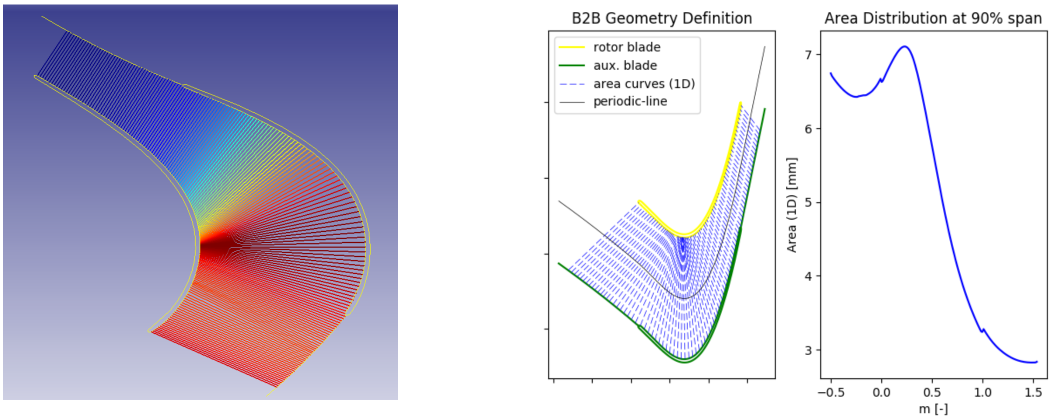
5.2. Blade-to-Blade (B2B) Analysis

When looking at the Blade-to-Blade plane of a turbine, two particular geometry characteristics are interesting for turbomachinery design. The Area Distribution gives an indication of the general expansion or compression of the channel flow, while the Curvature Distribution gives an indication of the acceleration of the flow near the blade surfaces, which can be related to flow characteristics such as separation or shockwave formation.

5.2.1. Area Distribution

The B2B area distribution is found by cutting two blades using a ‘Blade-to-blade’ surface, which is a revolved meridional line. This meridional line line can be defined at any span wise location and will be a linear interpolation of the control points and weights of the hub and tip BSpline curves. The span-wise location is defined as a fraction of the span, ranging from 0.0 at the hub to 1.0 at the tip. A cambersurface is placed in between the two blades; this surface defines the mean streamline at any point along the span. The intersection of the ‘Blade-to-blade’ surface and the center cambersurface determines the mean streamline at the specified span. Along this mean streamline, a specified number of planes are placed equally spaced along the streamline curve and are oriented perpendicular to the streamline direction. These planes represent the frontal area as seen by the flow, when moving through the rotor channel. The intersection of these planes, the ‘Blade-to-Blade’ surface and the bounding rotor blades, define the 1D area curves that the flow sees in that particular ‘Blade-to-Blade’ plane.

The mean-streamline and the suction side blade are extended to construct the ‘virtual’ flow channel that is outside of the blade boundaries, but is still ‘seen’ by the flow. The extension at the leading edge resembles an outward moving spiral, where the curve angle with respect to the origin, remains constant. At the trailing edge, the extension resembles a helix with a ‘helix angle’ equal to the outflow angle.

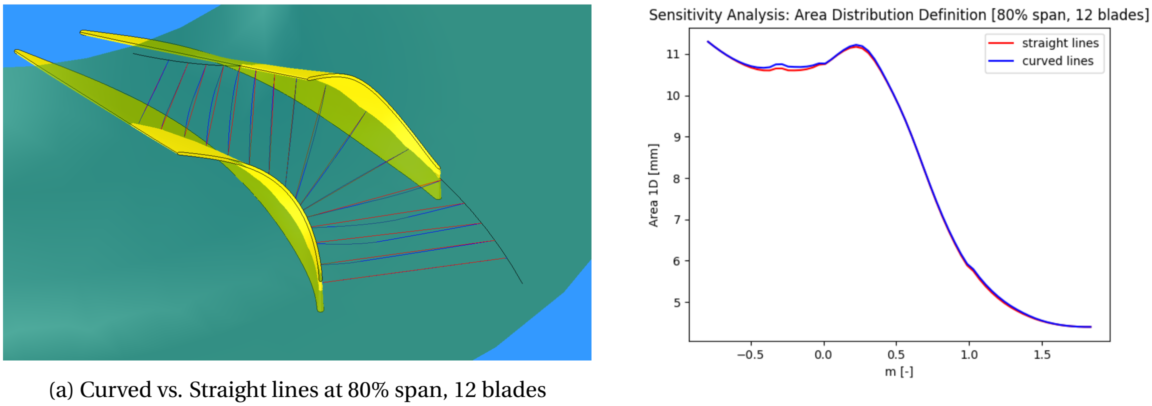


(a) B2B Area lines along the mean streamline

(b) B2B flat projection and area distribution plot at 50% span, 19 blades

Figure 5.3: B2B Area Distribution Analysis produced in ParaPy

The 1D area curves can be exactly constructed using a series of intersection and trimming operations, or the 1D area curves can be approximated using a line segment between the two intersection points at the bounding blades, as illustrated in fig. 5.4a. The line segments are computationally less intensive, but are similarly less accurate in describing the area distribution. A small analysis has been conducted to see which of the two methods is more applicable for the B2B analysis. In this analysis, both methods are compared for different span (i.e. radius) and different pitch, the two factors that directly influence the length of circumferential lines. In indication of the difference between the two methods can be seen in figures 1 and 2. The errors and computational times are tabulated in table 5.1



(a) Curved vs. Straight lines at 80% span, 12 blades

(b) Area Distribution at 80% span, 12 blades

Figure 5.4: B2B Area Distribution Analysis produced in ParaPy

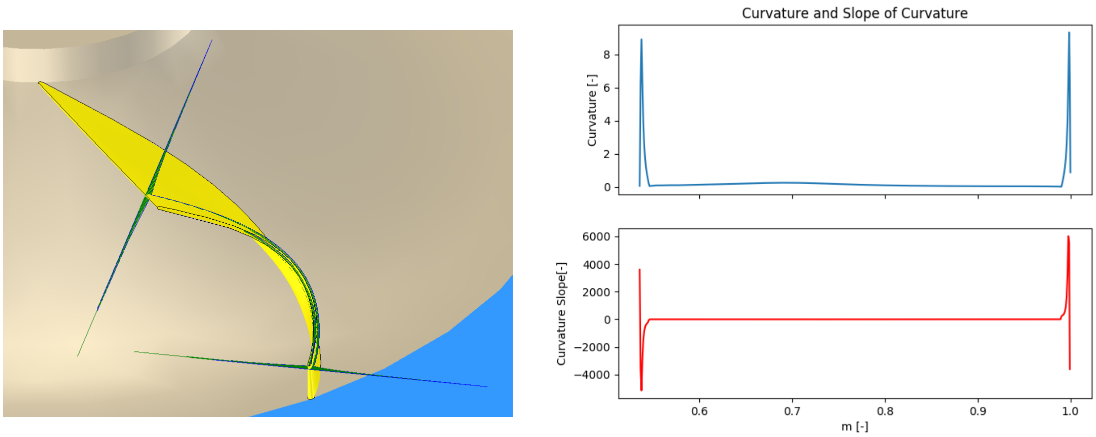
Span [%]	Nr. of blades [-]	Error [mm]	Rel. Error [%]	Δ comp. time [s]	Rel. comp. time [%]
20	11	0.0338	0.301	2.612	19.791
50	11	0.0435	0.379	1.317	12.627
80	11	0.0715	0.603	1.501	14.293
20	15	0.0128	0.159	1.865	14.556
50	15	0.0166	0.200	2.002	15.948
80	15	0.0272	0.319	1.873	13.142
20	19	0.0091	0.144	2.058	17.863
50	19	0.0110	0.170	2.384	16.181
80	19	0.0184	0.278	2.061	19.865

Table 5.1: Errors and Computational time for Exact and Approximated Method

The difference in errors is so marginal (being less than 0.4%), that the reduction in computational time (10% to 20%) is much more essential to the user compared to the slight increase in accuracy. It is therefore decided that the Approximated straight lines method is preferred over the more exact curved lines method. Particularly, when the user want to iteratively evaluate the influence of small adjustments to the geometry on the B2B area distribution, the shorter computational time is preferred over the improved accuracy. The exact curved line method is available as a secondary option.

5.2.2. Curvature Distribution

The curvature distribution is determined using the same method as for the meridional channel analysis. The curvature is evaluated at a given number of points on the cross-sectional blade profile, along the meridional line. Both the pressure and the suction side curvature is projected and plotted. The plots and projections will show a particularly high peak at the leading and the trailing edges.



(a) B2B Curvature lines along the local blade profile

(b) B2B curvature and slope of curvature plot at 50% span, 19 blades

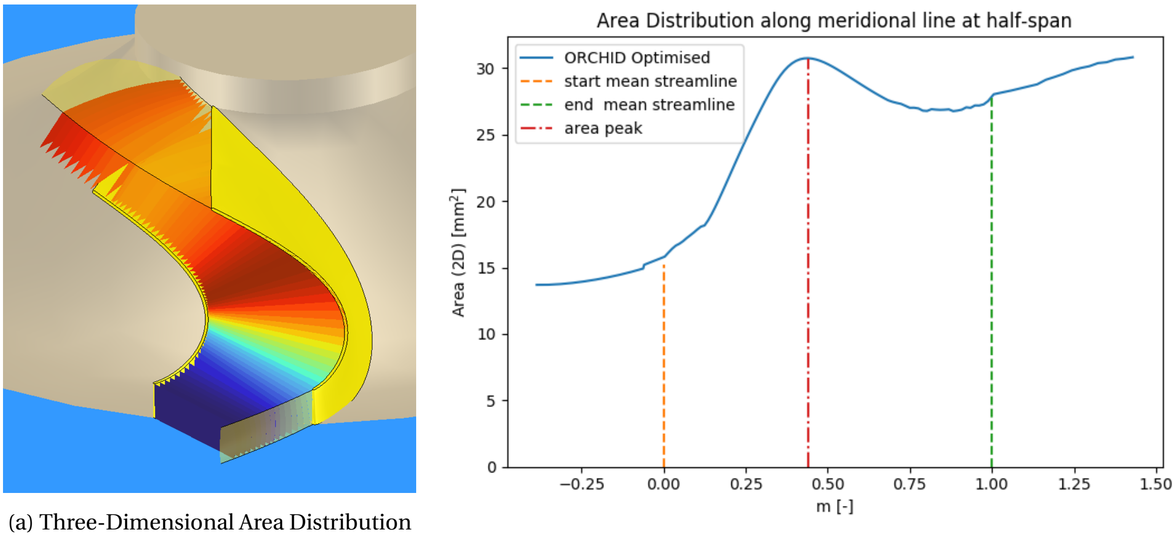
Figure 5.5: B2B Area Distribution Analysis produced in ParaPy

As can be seen in fig. 5.5a, the curvature is defined perpendicular to the cambersurface. The curvature in the direction of the meridional channel curvature is removed from distribution by projecting the profile curvature vectors onto the normal vectors of the cambersurface. The change in magnitude of the projected vector is also reflected in the curvature plots.

5.3. 3D Analysis

A three-dimensional analysis is slightly more computationally intensive than the Blade-to-Blade analysis, but is still much faster than performing a simple CFD analysis on the rotor blade geometry. For the full 3D analysis of the rotor blade, the channel bounded by two blades along the full span is analysed. For the 3D analysis, only the area distribution is calculated. It is very difficult to calculate the curvature of a 3D surface since it has curvature in multiple directions. This also makes it very difficult to visualize over the full 3D blade and makes it also very difficult to interpret. The area distribution provides more significant information about the flow characteristics, particularly about regions of expansion and compression for the flow.

For the 3D analysis, the mean stream line is computed at half-span. Similar to the B2B area distribution, a set of planes is placed perpendicular along this mean streamline. The hub, mid and tip camberlines curves are extended using the same method as for the B2B analysis. These curves are combined to obtain the leading edge and trailing edge extension surfaces. Subsequently, the intersection curves between the planes and the bounding blades and extension surfaces are computed. Several ruled surfaces are constructed using pairs of the intersection curves, as can be seen in fig. 5.6a. The corresponding area of each surface is plotted against the mean streamline in fig. 5.6.



(a) Three-Dimensional Area Distribution of full rotor blade channel

(b) Area Distribution diagram

Figure 5.6: Area Distribution

Note that the color change of the area surfaces in fig. 5.6a match the shape of the Area Distribution curve as presented in fig. 5.6. The blue surface has the lowest area, the dark red surface has the highest area.

5.4. Area Distribution Comparison

As can be clearly seen, the 2D B2B Area Distribution differs significantly from the 3D Area Distribution in its general trend. This is of course due the to fact that the 3D area distribution is intuitively an integration of the meridional area distribution and the Blade-to-Blade area distribution combined. The significant area increase of the meridional combined with the relatively smaller area decrease of the B2B channel, lead to an increasing 3D area distribution. Figure 5.7 shows how each distribution relates to the others.

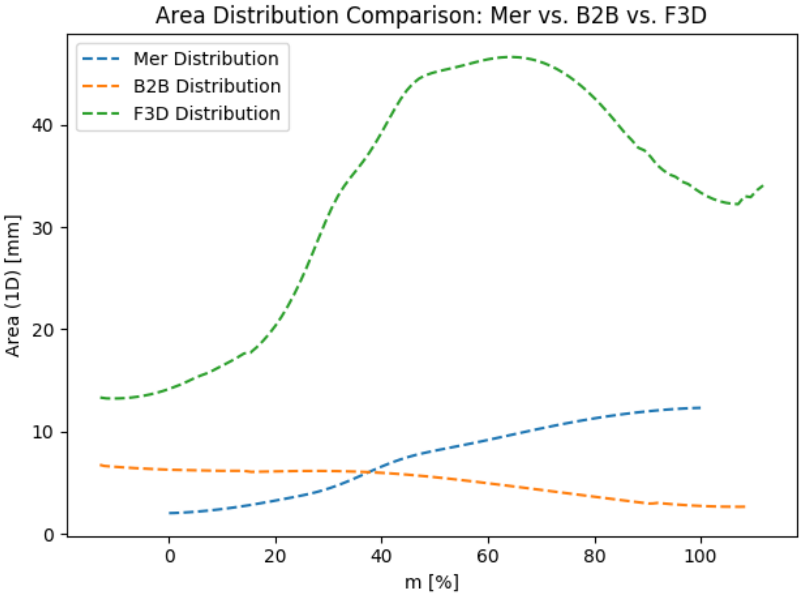


Figure 5.7: Comparison of Meridional, Blade-to-Blade and Full 3D Area Distribution

6 Mesh Generation

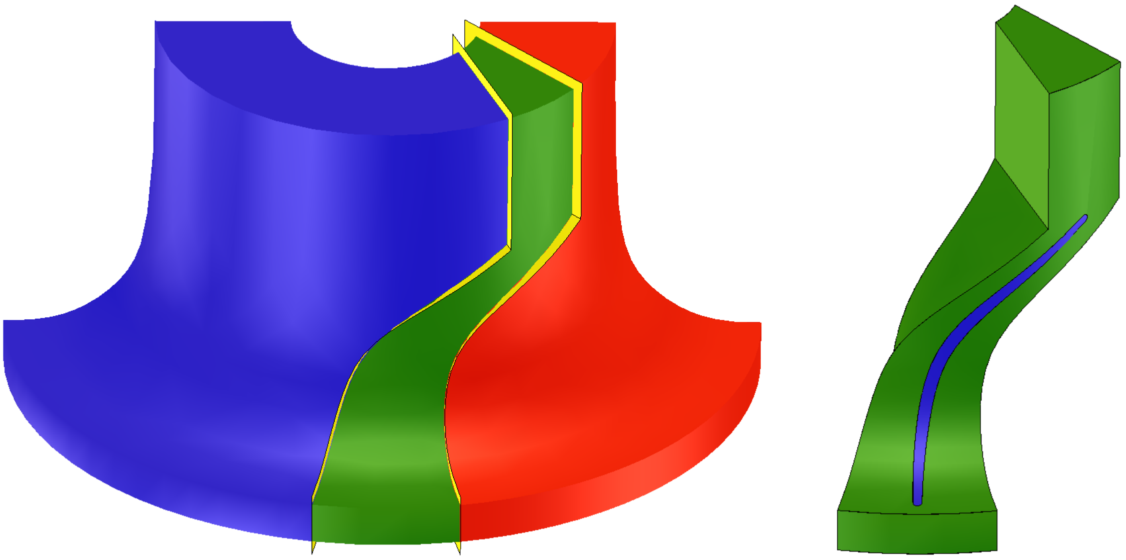
When parametrised geometry is analysed and it geometrical characteristics are deemed acceptable, the next step in the RIT design cycle is the meshing of the parametrised geometry. Throughout the thesis, two different meshing software packages were used: Salome 6.3.0 and AutoGrid5 TM. Salome was extensively used as the main software package responsible for the meshing phase in the turbine design cycle. AutoGrid was mainly used for the validation of the parametrisation methodology and accuracy. The main considerations discussed here will be related to the Salome software package, but the AutoGrid procedure will also briefly be discussed in section 6.2.

6.1. Salome

6.1.1. Domain Definition

A proper definition of the fluid domain is required for generating a high quality mesh with correct boundary conditions. A fluid domain similar to the domain defined in section 2.3, is created in ParaPy.

The bounding surface that is used to partition the circular domain, is composed of the blade's cambersurface, a radially outward extended surface at the leading edge, and an axially outward extended surface at the trailing edge. A 'blade-centered' domain was chosen, hence this bounding surface is rotated to a half pitch in both clockwise and counter-clockwise direction, effectively bounding the blade at both sides at the center of the respective blade channels.



(a) Fluid Domain Definition

(b) Split of Fluid and Blade domain

Figure 6.1: Fluid Domain partitioning

6.1.2. Fully Structured Meshing

Due to the complex shape of the rotor blade, a fully structured mesh was deemed to complex to create, requiring significant solid partitioning operations which are not only very computationally intensive, but are also prone to numerical instabilities in the ParaPy geometry kernel. Often, these solid partitioning operations led to crashes of the ParaPy software, therefore extensive use of these operations is avoided.

When using Salome, structured meshes could be generated, but due to the limited number and poor quality of the multi-block partitions, the design tool could never realise a high quality structured mesh, see fig. 6.2. While currently, the design tool does not have the fully integrated capability of generating high-quality structured meshes, fortunately, current software packages such as AutoGrid or ANSYS TurboGrid do provide a good capability to construct these high quality structured meshes. The coupling between these meshing programs and the design tool has been fully implemented, so even if Salome cannot be used for the structured meshing, the design tool is able to generate a high-quality mesh fully automatically using AutoGrid (as will be discussed in section 6.2).

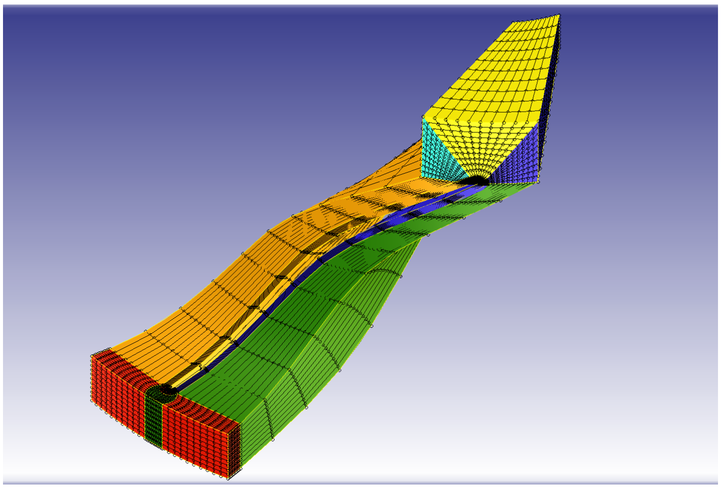


Figure 6.2: Poor Quality Multiblock partitions

6.1.3. Hybrid Meshing

The complex shape led to the usage of a hybrid grid for the integration of Salome into the application. Since a high-quality structured grid is too difficult to generate, the unstructured mesh is chosen. For this meshing procedure, both 2D and 3D meshing algorithms are defined, these include the Tri-hypothesis, which defines the 2D triangular meshing elements, and the Tetra-hypothesis, which defines the 3D tetrahedral meshing elements.

Next to the basic meshing algorithms, several hypotheses are also defined. The hypotheses represent boundary conditions which will be taken into account by meshing algorithms'[25] These hypotheses mainly define the number of nodes per edge boundary or face boundary, or define the maximum element size. One particular hypothesis, however, is what makes the applied mesh a hybrid mesh. This is the ViscousLayer mix-in hypothesis. This hypothesis adds prism elements to the mesh to connect the unstructured and structured parts of the mesh.

Viscous Layer

The application of a viscous layer is particularly essential for the correct modelling of the boundary layer in CFD. The ORC turbine applications can have fluid flows with high levels of turbulence. Since the SU2 solver will use a K-epsilon turbulence model in the calculations, a y+ value of 1.0 is desired.

To generate a viscous layer that satisfies $y^+ = 1.0$, the user puts in several flow characteristics, such as: density (ρ), viscosity (μ) and free-stream velocity (U_∞). Parameters such as the reference length (L) are directly extracted from the geometry. Using eqs. (6.1a) to (6.1e) the Δy corresponding to a y^+ value of 1.0 is computed, and will be used as the thickness for the first layer of the viscous layer mesh. The total viscous layer mesh thickness is determined using the first layer thickness, the user-specified expansion ratio (R_{exp}) and the user-specified number of layers (n) in the viscous layer mesh, using the formula as illustrated in eq. (6.2).

$$Re_x = \frac{\rho U_\infty L}{\mu} \tag{6.1a}$$

$$C_f = \begin{cases} 0.0592 Re_x^{-1/5} & \text{if } 5 \cdot 10^5 < Re_x < 10^7 \\ \left(2 \cdot \log 10(\mathrm{Re}_x) - 0.65\right)^{-2.3} & \text{if } 10^7 < Re_x < 10^9 \\ 0.370 \cdot \log 10(\mathrm{Re}_x)^{-2.584} & \text{if } Re_x > 10^9 \end{cases} \tag{6.1b}$$

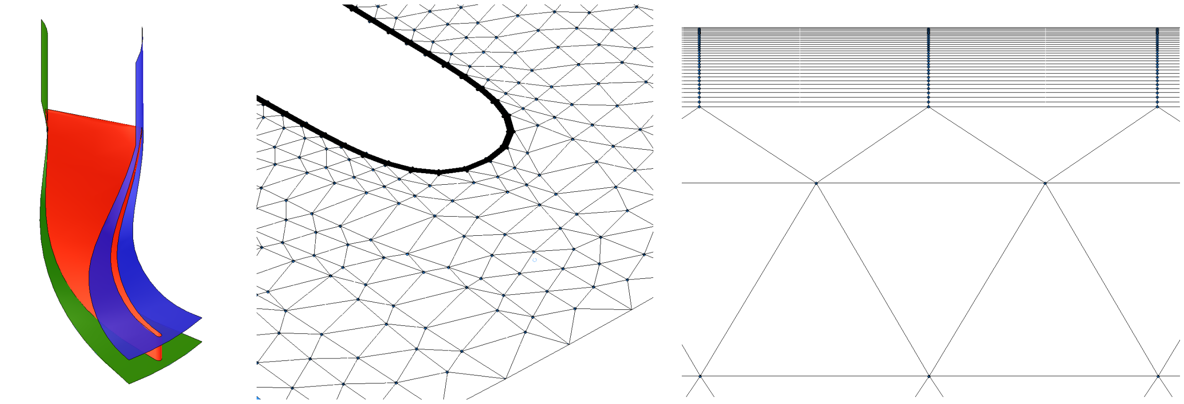
$$\tau_w = C_f \cdot \frac{1}{2} \rho U_\infty \tag{6.1c}$$

$$u^* = \sqrt{\frac{\tau_w}{\rho}} \tag{6.1d}$$

$$\Delta y = \frac{\mu y^+}{\rho u^*} \tag{6.1e}$$

$$t_{viscous} = \Delta y \cdot \sum_{i=0}^{n} R_{exp}^{i} \tag{6.2}$$

The viscous layer hypothesis is an addendum to the previously discussed Tetra-hypothesis. It requires not only the number of layers and layer thickness to be specified, it also requires a set of boundary wall on which to project the viscous layer.



(a) Definition of the Viscous Boundary Walls

(b) Example of the Viscous Layer around the blade only

(c) Example of the Viscous Layer at the shroud of the inlet boundary

Figure 6.3: Viscous Layer examples

Periodicity Matching

As previously described, the periodic boundary condition is essential for CFD simulations of turbomachinery, and cannot be created in SU2 without matching geometrical topology and matching periodicity.

Periodicity matching is more difficult for unstructured grid, because each grid node location is stored separately, therefore each node location cannot be determined with respect to a neighbouring node. Combining this with non-planar boundary faces makes the periodicity matching even more complex.

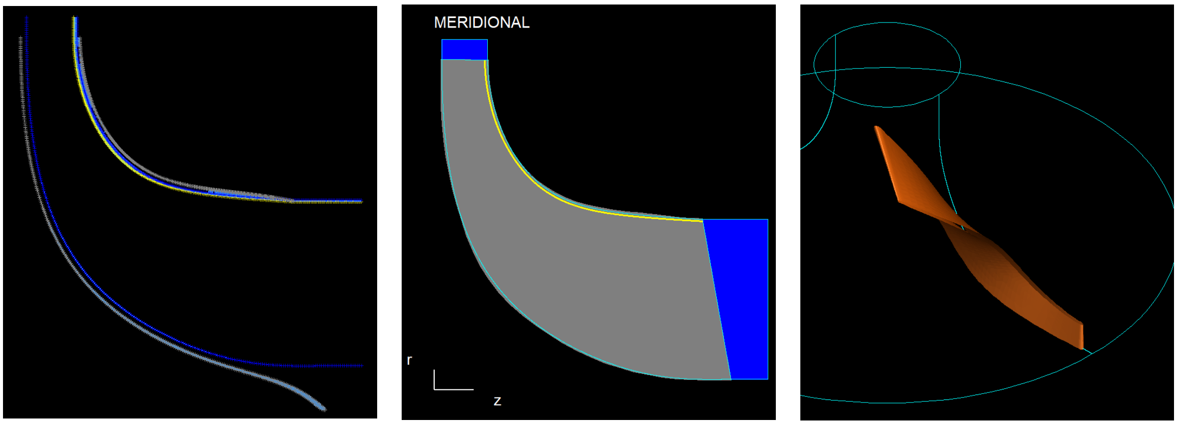
Fortunately, Salome has a function that can project mesh edges and mesh faces. The mesh face of the suction side periodic boundary is projected onto the pressure side periodic boundary. This method is added as an argument to the meshing algorithm and it ensures a matching periodicity during the mesh generation.

6.2. AutoGrid

The main meshing capabilities of the design tool were created in ParaPy Salome. However, for validation purposes, the quality and applicability of the Salome mesh output is currently not sufficient for validation purposes. Therefore, use was made of the AutoGrid5 TM software, to validate various geometry parametrisation aspects of this design tool. The validation process will be discussed further in chapter 7, but the meshing process in AutoGrid will be discussed below.

6.2.1. Blade Definition

The fluid domain is defined by Autogrid itself. The user only needs to input the hub and shroud curves, and the cross-sectional curves along the span. In order to construct a feasible fluid domain, the blade, which is produced from the set of cross-sections, and the hub and shroud curves always need to intersect fully. It is therefore decided to apply a small spanwise extension to the blade at both the hub and shroud. This ensures that the hub and shroud curves remain the same, thereby ensuring that a matching meridional channel is produced. Finally, a tip clearance may be added, which will be off-set inwards from the defined shroud curve. Figure 6.6 shows both the ParaPy defined points and the corresponding AutoGrid Meridional Domain definition. The subsequent definition of the full 3D volumetric fluid domain is handled by the AutoGrid, therefore no volumetric operations are required in ParaPy.



(a) Hub, shroud and profile curves in ParaPy, to be exported to AutoGrid

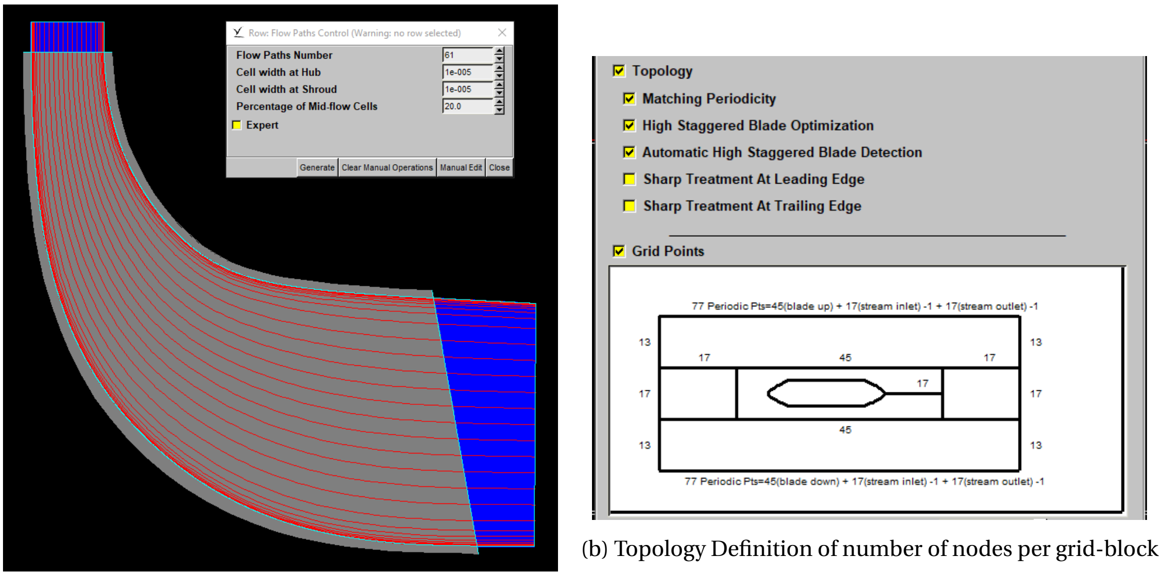
(b) Definition of meridional channel in AutoGrid, including tip clearance

(c) Definition of 3D blade surface in AutoGrid

Figure 6.4: ParaPy to Autogrid Conversion

6.2.2. Structured Meshing

While much of the AutoGrid software is automised, several parameters are selected by the user to ensure that a high quality grid for the particular turbine is generated. Since AutoGrid uses structured grids to generate the mesh, a lot of local control is available to the user. The number of nodes for each mesh block can be fully defined. Through some scripting, this topology definition can already be defined from the ParaPy interface. The tool uses the default AutoGrid topology, which is an HOH-grid surrounded by two additional H-grids at the pressure and suction sides. Note that, in order to create a proper cyclic boundary condition in SU2, the mesh should have a matching periodicity.



Row: Flow Paths Control (Warning: no row selected)

Flow Paths Number 61

Cell width at Hub 1e-005

Cell width at Shroud 1e-005

Percentage of Mid-flow Cells 20.0

□ Expert

☑ Topology

☑ Matching Periodicity

☑ High Staggered Blade Optimization

☑ Automatic High Staggered Blade Detection

□ Sharp Treatment At Leading Edge

□ Sharp Treatment At Trailing Edge

☑ Grid Points

77 Periodic Pts=45(blade up) + 17(stream inlet) -1 + 17(stream outlet) -1

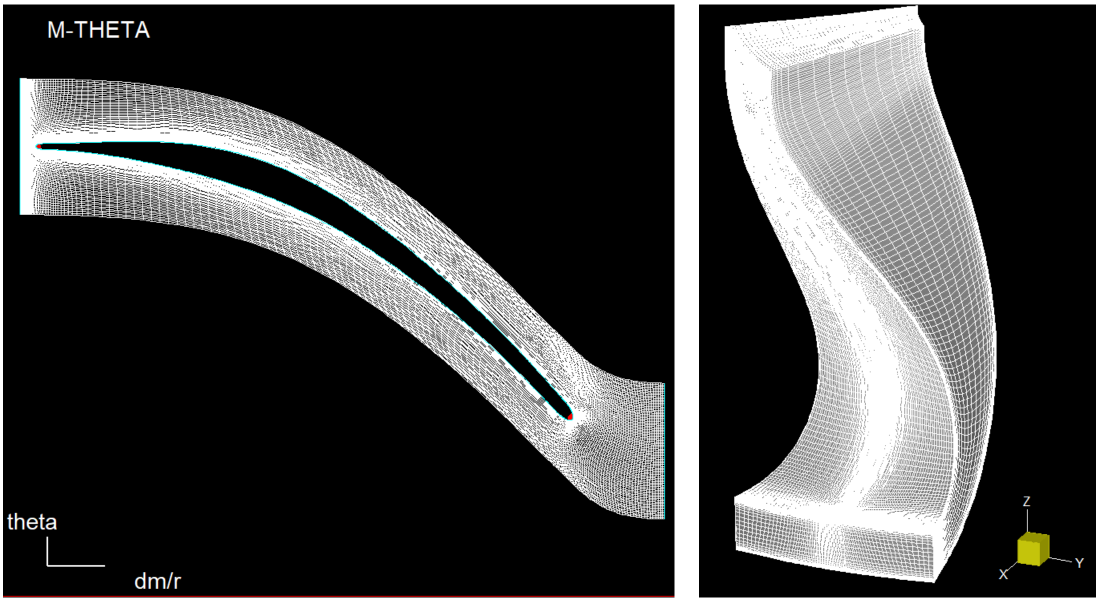
77 Periodic Pts=45(blade down) + 17(stream inlet) -1 + 17(stream outlet) -1

(b) Topology Definition of number of nodes per grid-block

(a) Number of spanwise nodes defining the flow paths

Figure 6.5: Meridional and B2B topology definitions

Once the grid topology and the number of spanwise flowpaths are defined by the user, both the B2B and 3D mesh are generated by AutoGrid. For the rotor blade, the high stagger angle optimisation is selected. Examples of the generated structured meshes are given in figs. 6.6a and 6.6b. The 3D mesh can be exported and can subsequently be used in the SU2 CFD solver.



(a) Autogrid Mesh B2B

(b) Autogrid Mesh 3D

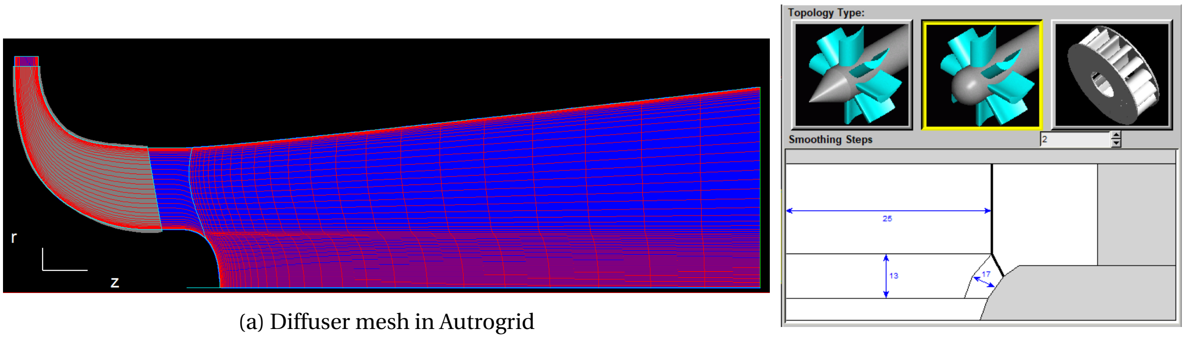
Figure 6.6: Resulting Autogrid Meshes

6.2.3. Additional Diffuser Mesh

AutoGrid has a built-in option that recongnizes when a diffuser shroud and diffuser bulb are added to the fluid domain. It will then make a cut in the wake between the rotor geometry and the diffuser. It can then mesh the diffuser and rotor separately in a multi-block grid. Figure 6.7a shows partitioning of the fluid domain and the division of the flow paths throughout the rotor and diffuser.

The wake discretization can be defined by the user and AutoGrid will ensure that the grid block have a smooth and continuous connection. The wake discretization is dependant of the bulb-type used. The three type are: the sharp, the round, and the radial-straight bulb. See fig. 6.7b for an example of the bulb-types and the wake discretisation options.

Note that SU2 cannot handle zero-radius rotational domains. Therefore, the last row of elements near the zero-radius line need to be removed.



(a) Diffuser mesh in Autrogrid

(b) Diffuser Bulb-type in Autogrid

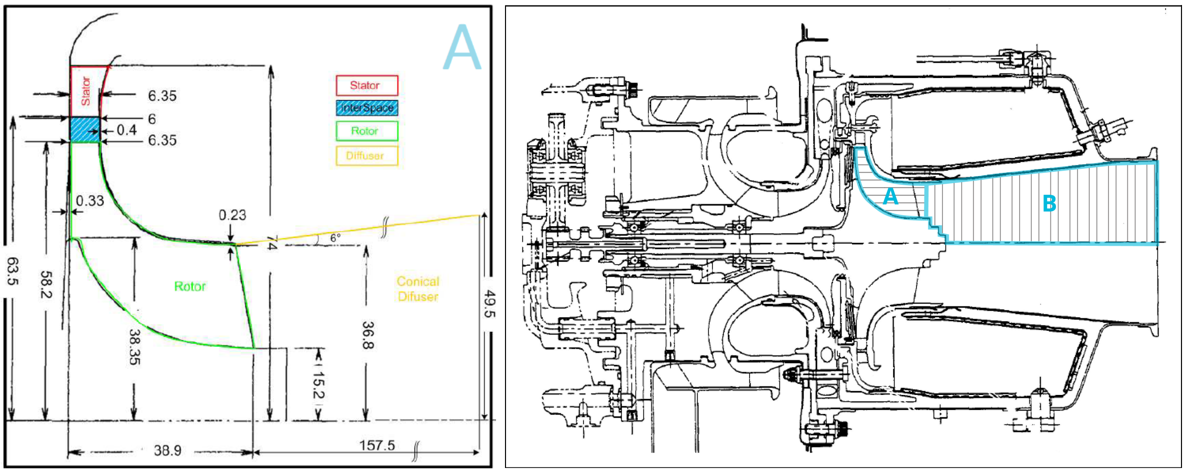
Figure 6.7: AutoGrid Diffuser Options

7 Validation of Turbomachinery Design Tool

The Turbomachinery Design Tool needs to be thoroughly validated. The turbine dimension of a RIT rotor and stator setup from literature are compared with the geometry of the design tool. Also high-fidelity CFD analysis from literature will be used to validate the analysis capabilities of the design tool. The first reference RIT design is described in the paper by Sauret, which provides “a fully open set of data to reproduce the exact geometry of the high pressure ratio single stage radial-inflow turbine used in the Sundstrand Power Systems T-100 Multipurpose Small Power Unit.” [27].

7.1. Geometry Reconstruction of the RIT

For the reconstruction of the exact geometry of the high pressure ratio radial-inflow turbine, several input parameters are fitted for the stator, the rotor and the diffuser geometries.



(a) Turbine sketch (not to scale), source: [26]

(b) Blueprint of the T-100 MPSPU Power Module, source: [11]

Figure 7.1: Sauret Turbine Geometry

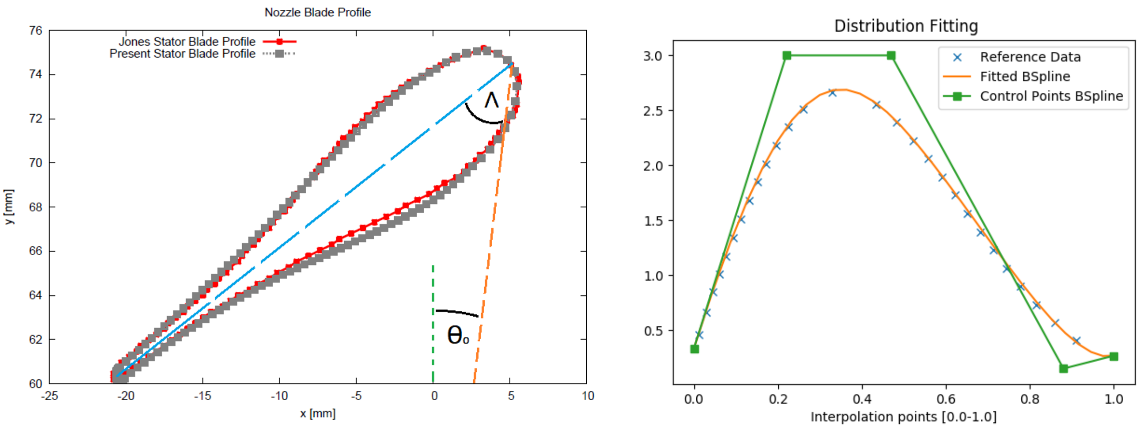
Stator		Rotor		Diffuser	
Input Params	Dimension [mm]	Input Params	Dimension [mm]	Input Params	Dimension
r_2	74	r_4	58.2	r_{5_h}	15.2 [mm]
r_3	63.5	r_{5_h}	15.2	r_{5_t}	36.8 [mm]
b_2	6.35	r_{5_t}	36.8	r_{6_t}	49.5 [mm]
b_3	6.00	b_4	6.35	Chord$_{ax}$	157.5 [mm]
$TE_{thickness}$	0.51	$TE_{thickness}$	0.76	α_{wedge}	12 [°]
		Chord$_{ax}$	38.9		
		Clearances			
		Axial	0.4		
		Radial	0.23		

Table 7.1: Turbine Geometric Dimensions

7.1.1. Stator Reconstruction

For the reconstruction of the stator, the readii and blade height could be found in the literature. The stagger angle, however, had to be extracted from the provide 'Nozzle blade profile' data set (see fig. 7.2a). For this calculation, the conversion from a carthesian to a cylindrical reference frame has to be kept in mind. Note that the leading edge is not located at $x = 0$ mm. An initial rotation of θ_0 also was applied.

The inflow and outflow angle were determined in such a way that the created a nearly straight camerline in the carthesian refernce frame. The profile thickness was projected along the camberline using the Bezier/BSpline fitted Thickness distribution, fig. 7.2b.



(a) Imported Profile Thickness Stator with B-Spline Fit

(b) Corresponding Profile Thickness using a B-Spline Fit

Figure 7.2: Stator Fitted

7.1.2. Rotor Reconstruction

Due to the fairly complex 3D geometry, the reconstruction of the rotor is a tedious process. First the 2D meridional channel curves were fitted with the provided hub and shroud curve coordinates. Then the Blade Angle distributions were fitted, and finally the thickness distribution was fitted.

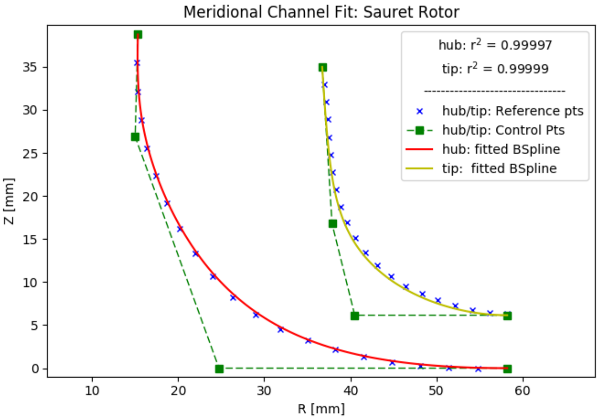
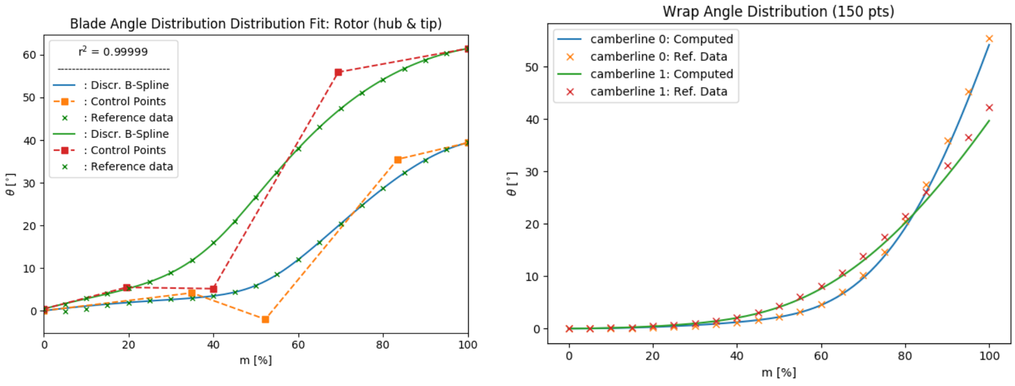


Figure 7.3: B-Spline fitted beta distribution (hub and tip)

Input Params.	Hub	Tip	Unit
Sweep angle	-10.14	N/A	[°]
Cone angle	0.00	N/A	[°]
Inflow angle	-0.50	-0.7	[°]
Outflow angle	0.00	-3.0	[°]
LE interp.	0.62	0.47	[-]
TE interp.	0.51	0.70	[-]
weights			
t1	0.63	0.58	[-]
w1	2.10	1.40	[-]
w2	2.00	2.20	[-]
t2	0.55	0.58	[-]

Table 7.2: Fitted Meridional Channel Parameters

Figure 7.3 above shows a high level fit of the channel curves. Table 7.2 provides the corresponding design parameters. With a r^2 of 0.9997 and higher, this was considered a proper fit.

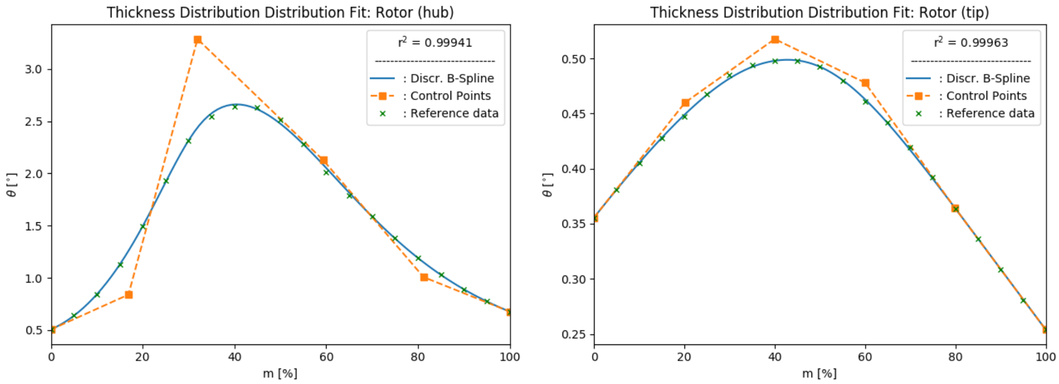


(a) B-Spline fitted beta distribution (hub and tip)

(b) Calculated theta distribution

Figure 7.4: Angle distributions fitting

Figure 7.4a presents the blade angle distribution with a 5 point Bezier curve and the corresponding computed wrap angle distribution. While the blade angle distribution shows a high level of fit, the wrap angle distribution shows a minor deviation from the literature data. Errors of a few degrees are present near the trailing edge. The impact of these errors will become evident in the CFD analysis. The figures below show a fit of the Thickness Distributions of the hub and shroud curve of the rotor. These distributions were fitted using the same method as for the stator.



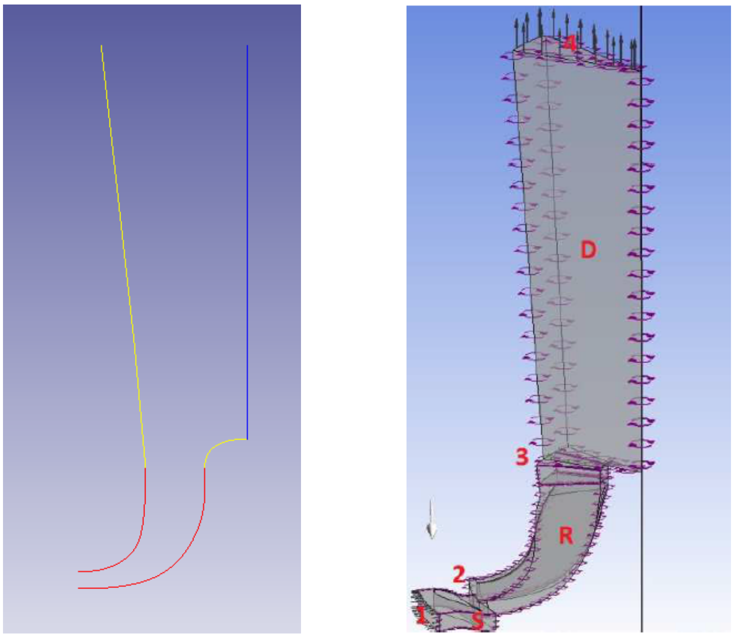
(a) B-Spline fitted thickness distribution (hub) (b) B-Spline fitted thickness distribution (tip)

Figure 7.5: Thickness distributions fitting

7.1.3. Diffuser Reconstruction

While the Sauret did not model the diffuser in the 2012 paper [26], her subsequent 2014 paper shows a full CFD analysis and validation including the diffuser, as shown in fig. 7.6b.

Note how, after the rotor mesh, which has a relatively short wake mesh, the diffuser is modelled as a partial cone ranging along the full radius, i.e. the hub radius is zero. However, looking at fig. 7.1b, there is a more gradual, staircase-like, decrease in hub radius. The 90 degree angle at the transition from the rotor to the diffuser mesh, as modelled by Sauret, was avoided, since such abrupt geometrical changes can cause numerical instabilities in the mesh and CFD solution. Therefore, a BSpline curve was constructed at the hub of the diffuser inlet, to gradually reduce the hub radius to zero and to try to more closely match the actual experimental setup.

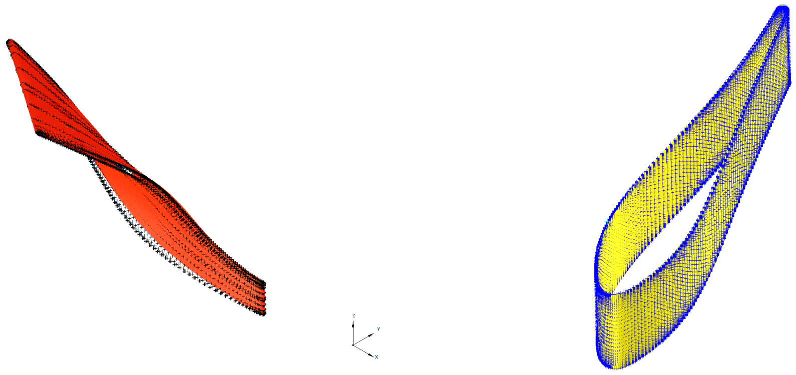


(a) ParaPy Generated Diffuser Model (b) Diffuser Model by Sauret, source: [27]

Figure 7.6: Diffuser Comparison

7.2. Geometry Visual Check

For a visual inspection, the produced geometries are compared with the set of coordinates extracted from the mesh provided in the literature. While this is only a qualitative comparison, it gives a good estimate of the level of fit of the full 3D geometry, instead of looking only at fitted 2D distributions.



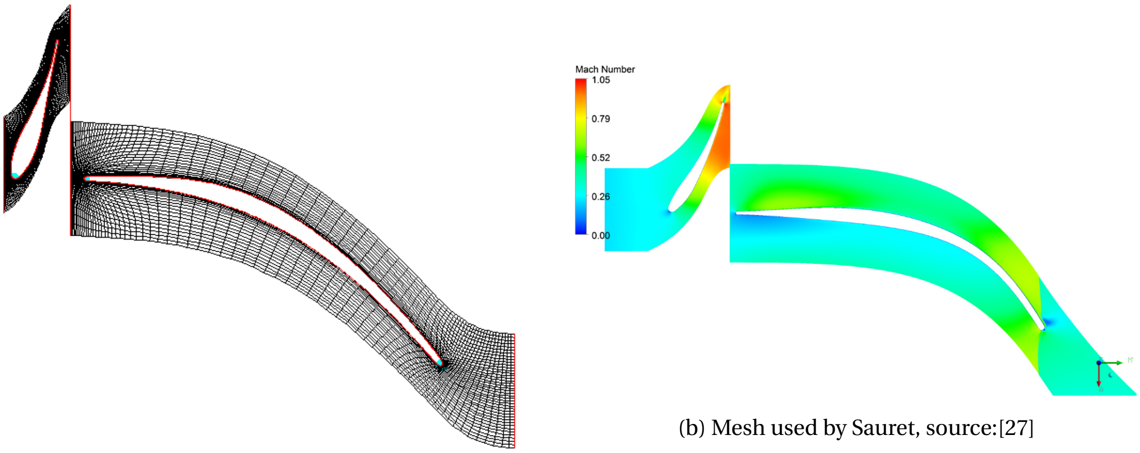
(a) Stator Fitted to the 3D Stator Coordinates

(b) Rotor Fitted to the 3D Rotor Coordinates

Figure 7.7: CAD Comparison

7.3. Mesh Validation

As discussed in chapter 6, for the validation process, the commercial and thoroughly validated AutoGrid5 TM meshing software was used. The AutoGrid meshing process has already been discussed in chapter 6; the resulting meshes are briefly validated using visual inspection. While a visual comparison is not a proper quantitative validation, it provides a quick way to find minor errors or deviations in the mesh construction, misalignment, or any other errors in topology.

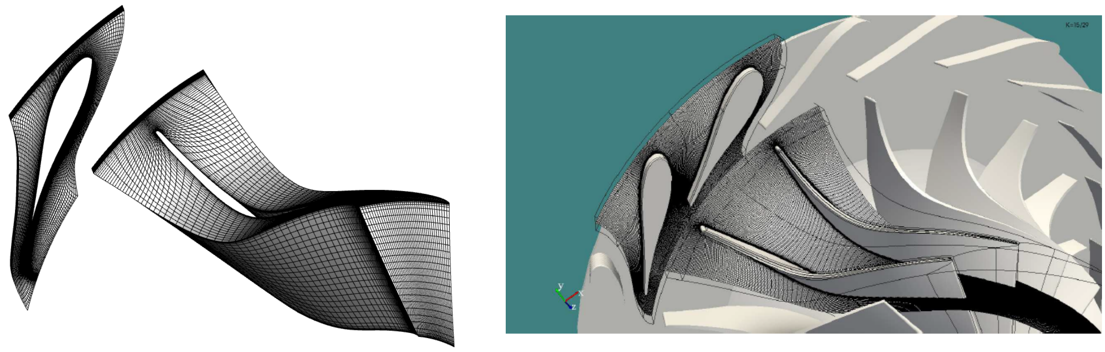


(b) Mesh used by Sauret, source:[27]

(a) B2B Mesh produced using Autogrid

Figure 7.8: B2B meshes produced in AutoGrid

Note that Sauret has chosen to extend the wake mesh in the outflow direction instead of the axial direction. AutoGrid directs the wake mesh into the axial direction to capture multiple wakes in one single fluid domain. This difference in mesh construction might show minor differences in the CFD solution, but both wake meshes are extended by roughly the same length, which will result in similar outflow conditions.



(a) 3D Mesh produced using Autogrid

(b) Mesh used by Sauret, source:[27]

Figure 7.9: Comparison of Mesh Grid

7.4. CFD Validation

With the geometry reconstructed and the mesh created, the validation of the reconstructed turbine performance was performed using the SU2 CFD solver.

For both the stator domain and the rotor domain, the periodic boundaries are coupled with a periodic interface. The two domains are connected through mixing-plane interface. The coupling of interfaces are computed in SU2, by linking the proper boundary faces in the SU2 config file. In table 7.3 below, the nomenclature of all boundary faces are defined, including the interface boundaries are tabulated.

A rotational reference frame is used to resolve the rotor flows. No tip-clearance is applied for the rotor domain. The diffuser domain has not been added for this validation.

	Nomenclature		
Boundary	**Stator**	**Rotor**	**(Diffuser)**
Inlet	inflow	inmix (inmix1)	(inmix2)
Outlet	outmix (outmix1)	outflow (outmix2)	(outflow)
Hub	hub1	hub2	(hub3)
Shroud	shroud1	shroud2	(shroud3)
Pressure Side Periodic	per1	per3	(per5)
Suction Side Periodic	per2	per4	(per6)
Interfaces			
Periodic	per1, per2	per3, per4	per5, per6
Mixing Plane(1)	outmix(1)	inmix(1); (outmix2)	(inmix2)

Table 7.3: Mesh Boundary Nomenclature used in .su2 meshes

The flow solution parameters that are defined in the SU2 config file are presented in table 7.4 below. The SU2 config and mesh files used for the simulations can be found in appendix B.

Flow Parameter	Value	Unit
Turbulence Model	SST	-
Math Problem	Direct	-
Gas Model	Ideal Gas	-
Viscosity Model	Sutherland	-
Mach Number	0.05	-
Pressure	400,000	Pa
Temperature	470.5	K
Density	1.7418	Kg/m3
Turbulence Intensity	0.05	-
Reynolds Number	600,000	-

Table 7.4: Flow Parameters used for SU2 simulations

7.4.1. CFX Comparison

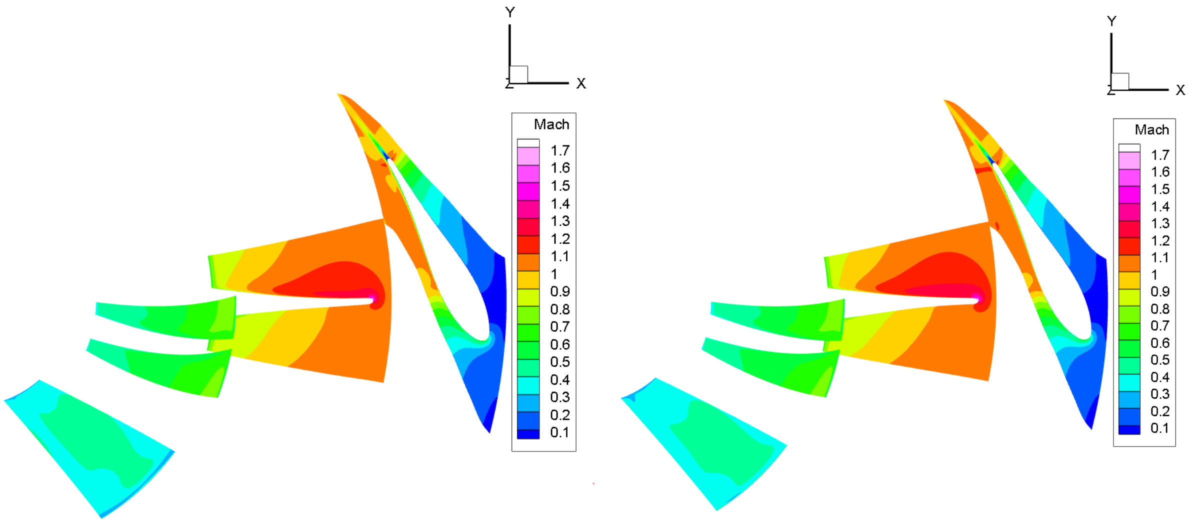
During the reconstruction of the geometry the fitting results were deemed quite successful, and also the visual inspection showed little deviation in geometry. However, there was quite an error in the initial comparison with the CFX data. It was therefore considered likely, that the discrepancy originates from the difference in meshes and not from the geometry itself. This was investigated further, by first extracting the exact blade geometry from the Literature TurboGrid mesh. Then, this geometry was meshed in AutoGrid using the same meshing approach as the ParaPy reproduced geometry.

The resulting two meshes are both analysed in SU2, so that both geometries are analysed using the same meshing software and the same CFD solver. This method allows for a proper validation of the parametrisation aspects of the turbomachinery design tool. This method, however, does not provide the means to validate the CFD results with respect to experimental data, simply because there is too much uncertainty in the CFD solver. A study of the settings and parameters of the CFD solver necessary to match the experimental data, is beyond the scope of this thesis.

7.4.2. SU2 Comparison with identical Mesh Setup

The SU2 comparison with the an identical mesh set-up shows much better results with respect to the CFX solution. Both simulations have performed 10,000 iterations and showed converged flow solutions. For this validation, no grid convergence study has been performed.

First, a qualitative comparison of the flow is made. Figures 7.10a and 7.10b show the mach contours in the Z-plane at stator midspan (3.125 mm), 20 mm and 42 mm in the Z-direction.



(a) Mach Contour Sauret

(b) Mach Contour ParaPy

Figure 7.10: Mach Contours plots of APU turbine at design speed

The Mach contours clearly show that the flow fields are nearly identical. The main difference, which is already difficult to find by visual inspection, is the slightly increased area of the (1.2-1.3) mach contour at the suction side of the rotor, near the leading edge. The ParaPy reproduced geometry shows a slightly increased Mach 1.2 area, but the surrounding mach contours are identical for both simulations.

The efficiency of the turbine is computed for three speeds at or around nominal speed. The efficiencies and the relative errors between the ParaPy reproduced turbine and the Sauret turbine are tabulated in table 7.5 below. The relative errors very small, with the biggest error being 0.2457%. Errors of this order show that the parametrisation methods used are of such accuracy that the difference in performance and flow fields is negligible.

Speed	ParaPy reproduced	Sauret Rotor	Relative Error
0.95 Nominal	89.43	89.34	0.1007%
1.00 Nominal	89.77	89.59	0.2009%
1.05 Nominal	89.76	89.54	0.2457%

Table 7.5: Comparison of Turbine Performance

In short, the reconstruction of the blade and the parametrisation techniques applied have a high level of accuracy and allow for the generation of a high-quality structured mesh.

8 Application of Design Tool

This chapter will discuss the possibilities in application of the Optiblade Design Tool. The recent research into the RIT ORCHID developed at the TU Delft, provided a good set of data useful for ORC RIT design and was therefore used as a reference design.

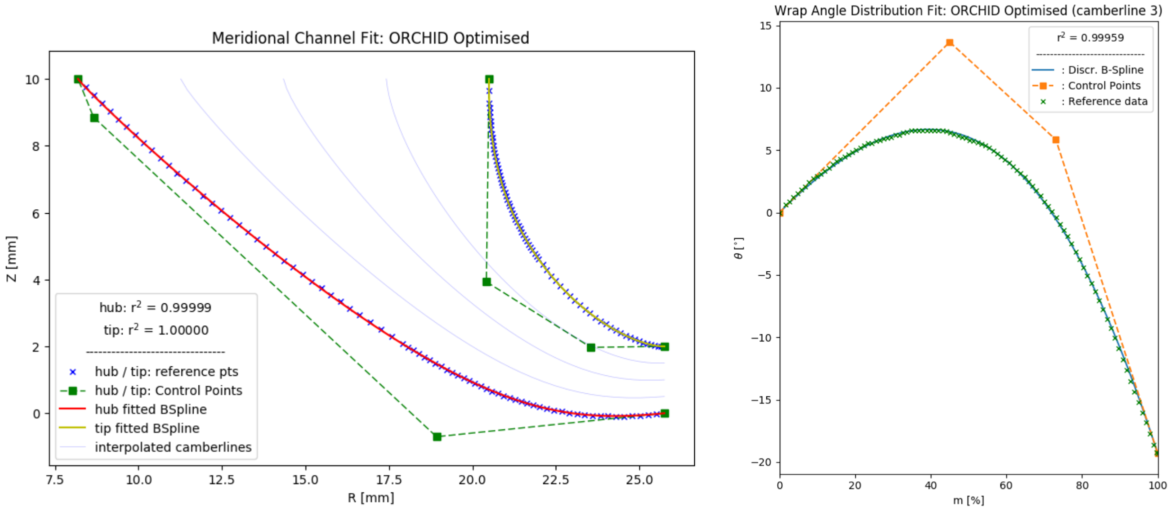
8.1. Reproducing the ORCHID Blade (optimisation) design cycle

The ORCHID project has been set-up by the Propulsion Department of the Aerospace Engineering Faculty at the TU Delft. An initial Baseline ORCHID design was already defined, and M. Burigana has optimised this initial design, by changing the number of blades, the meridional channel curves, and the blade angles. [4] His work focused on improving the efficiency of the rotor, and achieve this improvement CFX and BladeGen were extensively used.

The purpose of applying the turbomachinery design tool to this test-case, is to try to reproduce the design and reduce the duration of the cycle. This will both validate the design tool with respect to ANSYS (a design tool heavily used in industry), and it will show the capabilities of the analysis tool-set.

8.2. Reconstruction of the ORCHID rotor blade

Similar to the validation case, as discussed in chapter 7, both the 2D meridional channel, and the 2D wrap angle distributions are fitted to the supplied reference data, see figs. 8.1a and 8.1b respectively. The resulting 3D fit will be thoroughly analysed in section 8.4.



(a) Fit of Meridional Channel

(b) Fit of Wrap Angle Distribution

Figure 8.1: 2D fitting of Optimised ORCHID blade, using 4 Control Points

The fitting was again performed using the 'SLSQP' method, which provided r^2 values above 0.9999 for each fit. See appendix A for the Wrap Angle Distribution fit of the each of the five camberlines.

8.3. CFD Analysis Comparison

For an in-depth analysis of the accuracy of the blade-modeler functionality of the Optiblade tool, a comparison was made in the commercial ANSYS Workbench environment using two CFX CFD workflows. Workflow 1 consists of the Geometry component which created the RIT blade, the TurboGrid component, the CFX component, and the Results component. Workflow 2 has the same elements, but the Geometry component has been replaced with the Optiblade blade-modeller component, which provides the TurboGrid components a set of files defining the full geometry.

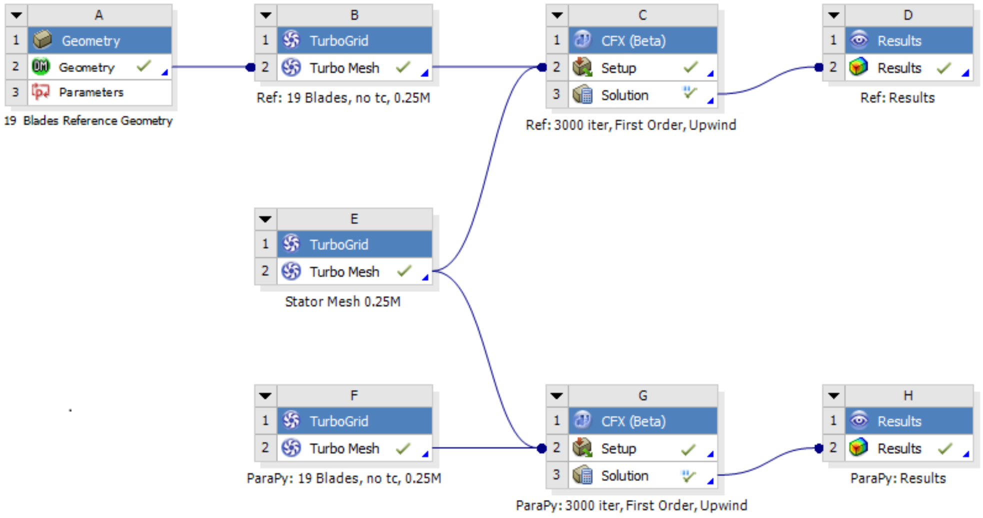


Figure 8.2: ANSYS Workbench, showing both CFX workflows; upper - Workflow 1; lower - Workflow 2

Workflow 1 serves as a benchmark, since it is a workflow that has been used and validated for optimisation design. [4]. Both Workflows have a TurboGrid setup using roughly 250000 nodes for the rotor blade, with a y+ based optimised grid. For the CFX analyses, the following boundary conditions and solver options were used:

Boundary	Definition	Value
Inlet	Subsonic pressure inlet	
	pressure	18.093 bar
	temperature	300°
Outlet	Subsonic pressure outlet	
	Average Static Pressure	0.443 bar
	Pressure Profile Blend	0.05
Stator-Rotor Interface	General Connection	-
	Stage (Mixing-Plane)	-
	Stage Average Velocity	-
Periodic Interface	Rotational Periodicity	-

Table 8.1: Boundary Conditions in CFX

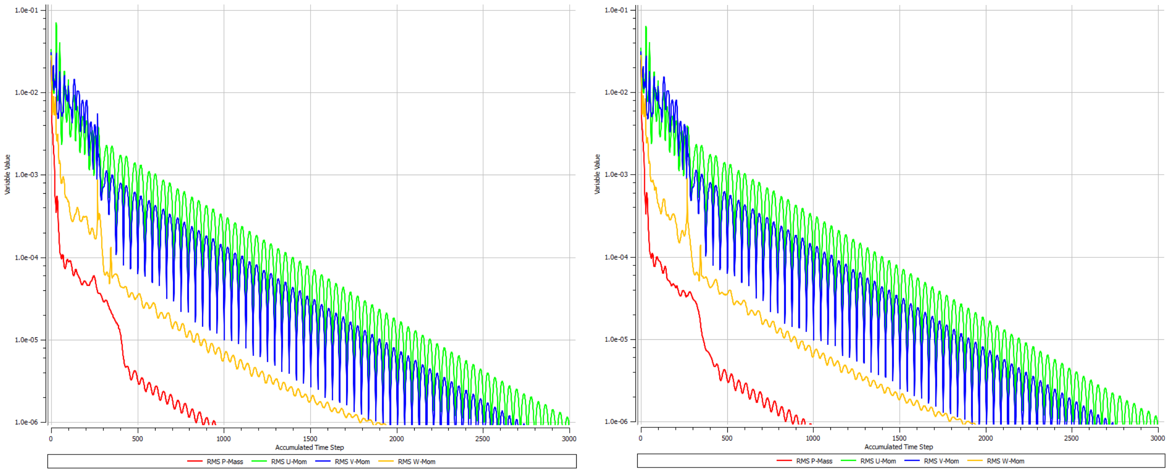
Solver Options	Value
Analysis Type	Steady State
Advection Model	Upwind
Turbulence Numerics	First Order
Number of Iterations	3000
Timescale Factor	0.4

Table 8.2: Solver Options in CFX

8.3.1. Solution Convergence

First, the convergence of both solutions is compared. While the numerical stability of the solution does not provide any information on the performance of the design, a comparison of the convergence is a great first check to see if both designs are in the same numerical region, and it serves as the first check that the data that will be investigated in the next section is a feasible solution. As a rule of thumb, when looking at the general residuals variables for turbomachinery fluid dynamics, a Residual decrease around 4 to 5 orders of magnitude is accepted as a converged solution.

For both workflow solutions, the residuals dropped by more than 4 orders of magnitude, as can be seen in fig. 8.5. The residual monitors for both workflows are nearly identical, showing that both solutions satisfy the rule of thumb, and are therefore considered ’converged’.

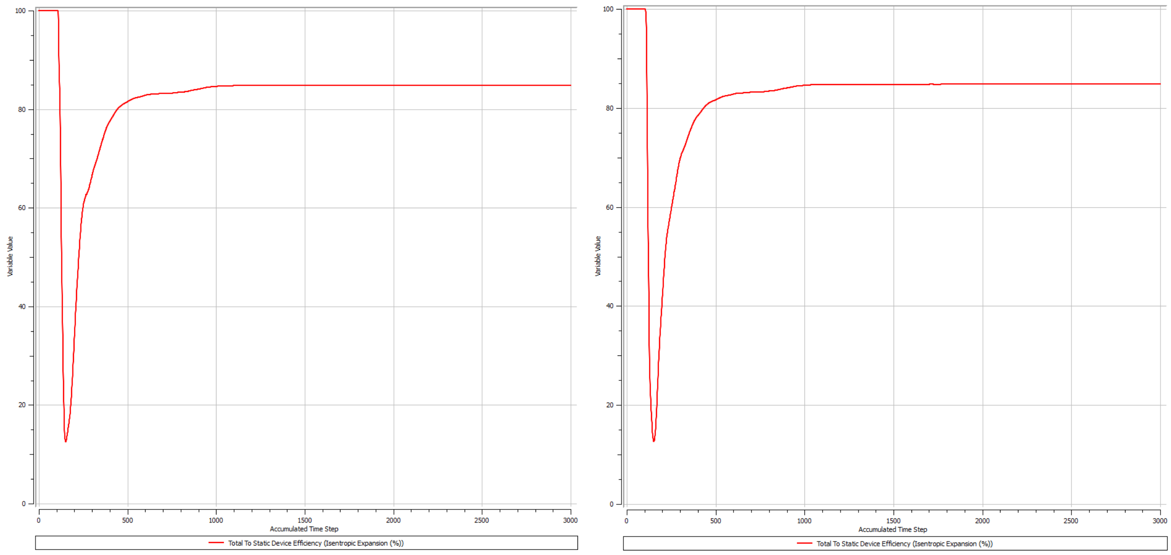


(a) ANSYS Reference blade

(b) ANSYS ParaPy blade

Figure 8.3: Solution Convergence: Residuals

Another qualitative check for the solution convergence, is to check that the efficiency reaches a plateau. Clearly, as can be seen in fig. 8.4, this is the case for both solutions. The initial plateau at 100% is where computations did not yet produce a feasible solution. Then the drop in efficiency is in the region where significant mass-flow difference are still present.

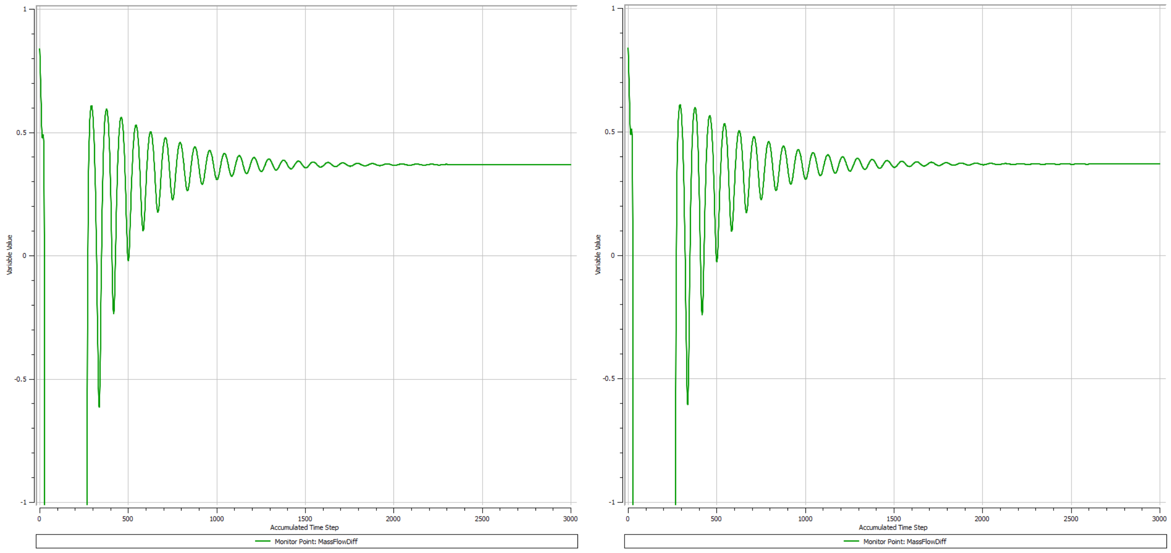


(a) ANSYS Reference blade

(b) ANSYS ParaPy blade

Figure 8.4: Solution Convergence: Efficiency plateau / Steady State

The final convergence check is generally the check of the Mass-flow difference. This Mass-flow difference converges to specific points as can be seen in fig. 8.5. This point is not equal to zero, due to the difference in number of stators and rotors. The Massflow difference is calculated using a single stator and rotor domain and normalised with respect to the stator domain mass flow, thereby leading to a massflow ratio of 12/19 at the rotor domain. Therefore the mass-flow difference converges to (1 - 12/19) = 0.3684, which is visible in both plots.



(a) ANSYS Reference blade

(b) ANSYS ParaPy blade

Figure 8.5: Solution Convergence: Massflow difference plateau

These three sets of plots indicate that both solutions have converged to a reasonable numerical result. Note that the simulations performed are for comparative purposes only. The results from the simulation are not an accurate representation of real life.

8.3.2. Solution Results

The results of both CFD simulations are discussed in this section using several tables and figures. The general performance characteristics are first compared in table 8.11 below, after which the flow characteristics will be discussed.

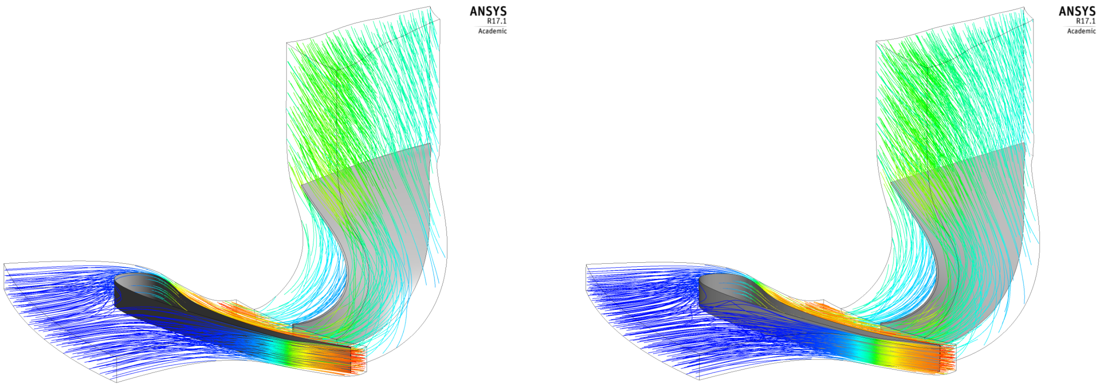
Performance Parameter	ORCHID blade	ParaPy blade	Relative Error
Power	106932 [W]	107042 [W]	0.1029 [%]
Torque	0.0574 [Nm]	0.0573 [Nm]	0.1031 [%]
Isentropic Expansion Efficiency (t-s)	0.848675 [-]	0.848031 [-]	**0.0759 [%]**
Mass flow (total)	0.137112 [kg s^{-1}]	0.137111 [kg s^{-1}]	0.0007 [%]

Table 8.3: Performance Comparison between work-flows

The boundary conditions for both work flows are identical, so several parameters such as the inlet to outlet pressure ratio, which are enforced by the boundary conditions, are identical and are therefore not tabulated.

The relative error in efficiency is less than 0.08%. From this data, it can be concluded that the rotor of the optimised ORCHID design and the replacement rotor that is generated using ParaPy Optiblade are not only geometrically very similar, but they also have similar turbine performance characteristics.

In order to have a closer look at the quantitive flow, several plots of important Turbomachinery flow parameters are presented below. fig. 8.6 plots the flow streamlines along the full 3D domain, fig. 8.36 presents a contour plot of the Mach Number in the Blade-to-Blade plane at mid-span. Finally, fig. 8.37 compares the rotor Blade Loading between the two blades.

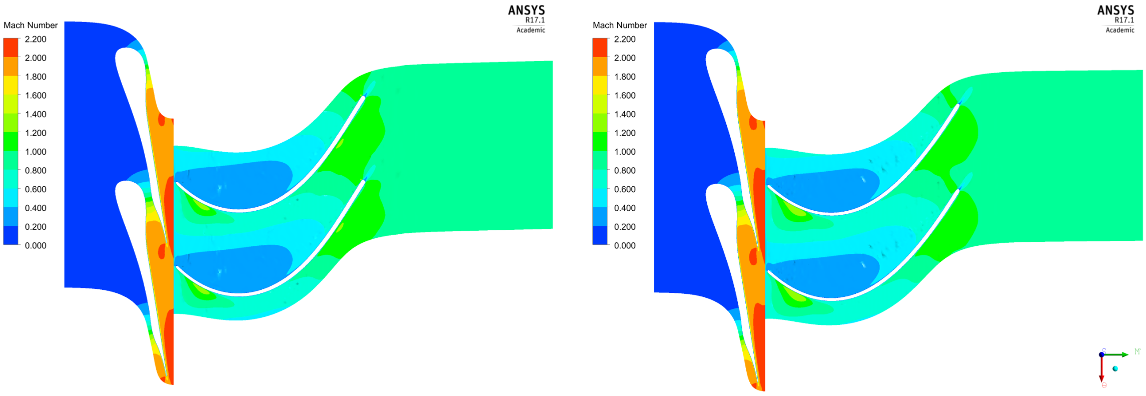


(a) Optimised ORCHID Rotor

(b) ParaPy Reproduced Rotor

Figure 8.6: Comparison of the 3D streamlines using ANSYS CFX Results

Both simulations have an indentical number of streamline seeds. The resulting streamlines are identical, showing only minor visible differences half-way of the suction side of the rotor.



(a) Optimised ORCHID Rotor (b) ParaPy Reproduced Rotor

Figure 8.7: Comparison of Blade-to-Blade Mach Contour using ANSYS CFX Results

The contour plots in figs. 8.36a and 8.36b above clearly indicate that the Mach Numbers of the flow at any point in the flow domain are very similar. The two main difference in Mach contour for the ParaPy blade compared to the ORCHID blade are the slight increase in the red area (Mach Number = 2.0-2.2) near the rotor leading edge and the slight decrease in Mach Number at the pressure side just behind the trailing edge. Fortunately, at both these areas, the contour plot changes by only one contour level, which shows that the differences could be a maximum of 0.2 in Mach Number.

The contour plots indicate very similar flow patterns between the two blades, but a more quantitive comparison of the Mach Numbers have been tabulated in table 8.12 below.

Location	ORCHID Optimised M_{rel} (M_{abs}) [-]	ParaPy Reproduced M_{rel} (M_{abs}) [-]
Stator Inlet	0.0259371	0.0259367
Stator Outlet	1.97464	1.99867
Rotor Inlet	0.542248 (1.98652)	0.569557 (2.0107)
Rotor Outlet	0.953754 (0.473172)	0.907224 (0.486794)

Table 8.4: Local Relative Mach Numbers

The blade loading of the rotor blade often gives a good indication of the flow patterns along and close to the blade surface. Particularly the pressure distribution is often used in the literature for comparing blade performance. In fig. 8.37, the blade loading pressure along the meridional line is plotted for both blades. The static pressure is normalized with respect to the outlet static pressure.

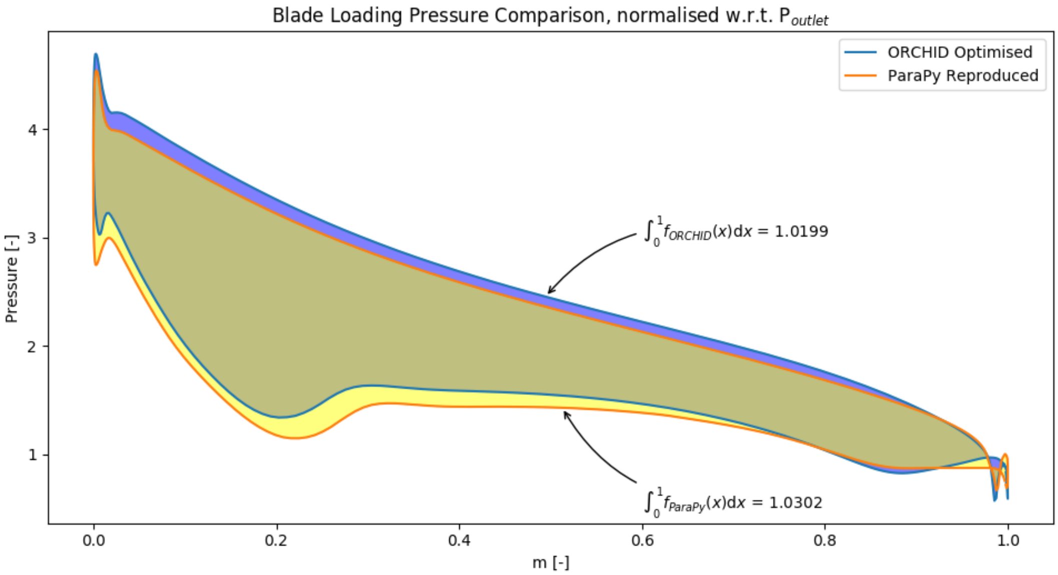


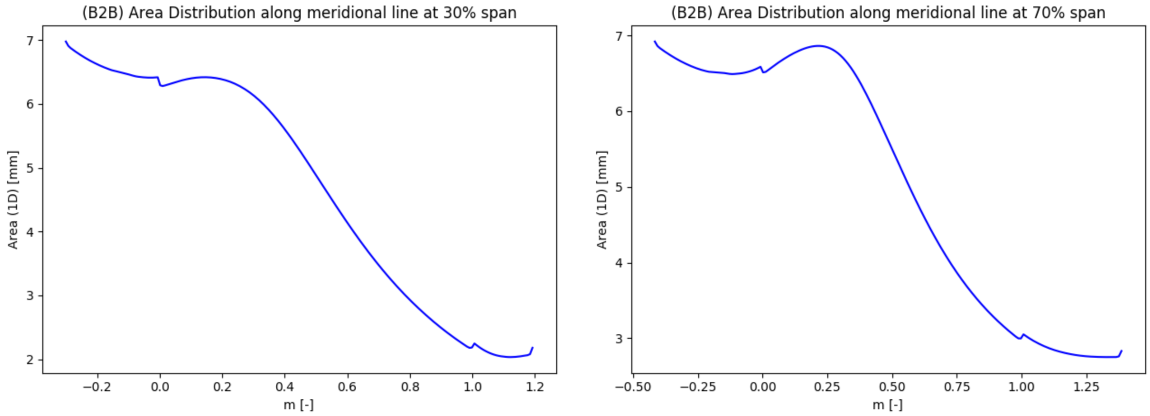
Figure 8.8: Comparison of Blade Loading using ANSYS CFX Results (normalised w.r.t. P_s at the outlet)

Figure 8.37 indicates that the ParaPy blade has a slightly reduced blade loading starting from the leading edge up to 90% of the meridional line, for both the suction and the pressure side. This pressure reduction is in line with the Mach contour plot, which showed a slight increase in Mach number near the leading edge of the rotor. The difference in blade loading is only marginal, with the biggest relative difference of 5.94% at the leading edge suction side, but the difference is more apparent than the differences in overall performance. To further investigate the blade loading, area under both curves has also been determined. The ORCHID Optimised blade loading curve had an area of 1.0199 and the ParaPy reproduced curve had an area of 1.0302. This shows that the blade loading has a relative error of only 1.01%. While being a relatively small error, this error is still significantly bigger than the errors in other performance parameters. Apparently, a small difference in overall turbine performance can cause an error to arise already upstream of the rotor, as can be seen in the contour plots, which then has a more significant effect on the blade loading of the rotor. This shows that any minor changes located anywhere in the fluid domain, can have significant effects on almost any other location in the fluid domain. The exemption here anywhere upstream of where the flow is chocked, e.g. at the supersonic stator throat.

All in all, with the highest error being only a 1.01% difference in blade loading and with the efficiency error at only 0.0759 %, the data shows that the results of the ParaPy reproduced blade are very comparable to the results of the Optimised ORCHID blade.

8.4. Optimised ORCHID Design Cycle Analysis

Now that it is known that the geometry produced using ParaPy is not only very similar to the ORCHID geometry, but also has similar fluid characteristics, the analysis tool of ParaPy can be used to analyse the Optimised ORCHID geometry. The fitted blade was used in the following Blade-to-Blade analyses. The blade was cut along the meridional line, at a fraction of the blade span, and subsequently the resulting blade cross-section was projected to a flat plane. Figures 8.9a and 8.9b present the Blade-to-Blade projection and the corresponding Area Distribution along the meridional line at 30% and 70% span.



(a) Blade-to-Blade Analysis of Optimised ORCHID blade at 30% span

(b) Blade-to-Blade Analysis of Optimised ORCHID blade at 70% span

Figure 8.9: Blade-to-Blade Area Distribution Comparison

As can be clearly seen in the Blade-to-Blade Area Distribution, at the inlet the 1D area first increases slightly (flow compression), after which, at roughly 25% to 35% of the mean line, the channel starts to converge again and the area decreases (flow expansion). Obviously, this compression of the flow should be avoided in a turbine (expander), as this reduces the flow expansion and therefore reduces the work done by the turbine.

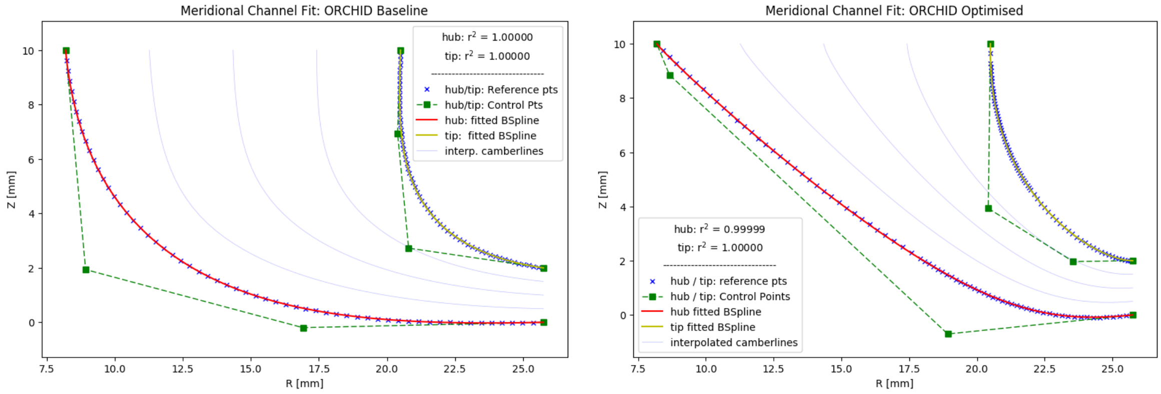
8.4.1. Design and Performance Comparison with the Baseline ORCHID

To see the effects of the B2B Area distribution on the overall performance of the rotor, the Area distribution of the Optimised ORCHID Rotor is compared with the initial Baseline ORCHID Rotor. Of both these rotors, the overall performance has already been computed and is tabulated in table 8.5. The Baseline geometry is reproduced in ParaPy, but now with a total of 19 blades. It was assumed that the reproduction of the Baseline geometry in ParaPy is accurate enough, such that a comparison of the ParaPy generated Baseline and the Optimised geometry is deemed acceptable.

Performance Parameters	**Baseline**	**Optimised**
Power	101594 [W]	106932 [W]
Torque (per blade)	0.0544957 [-]	0.0574 [Nm]
Isentropic Expansion Efficiency (t-s)	0.813014 [-]	0.848675 [-]
Mass flow	0.137112 [kg/s]	0.137112 [kg s^{-1}]
Geometry Parameters	**Baseline**	**Optimised**
Number of Blades	19 [-]	19 [-]
Pitch	0.331 [rad]	0.331 [rad]
Hub curve length	22.863 [mm]	20.865 [mm]
Tip curve length	10.947 [mm]	10.406 [mm]

Table 8.5: Performance Data of Baseline and Optimised

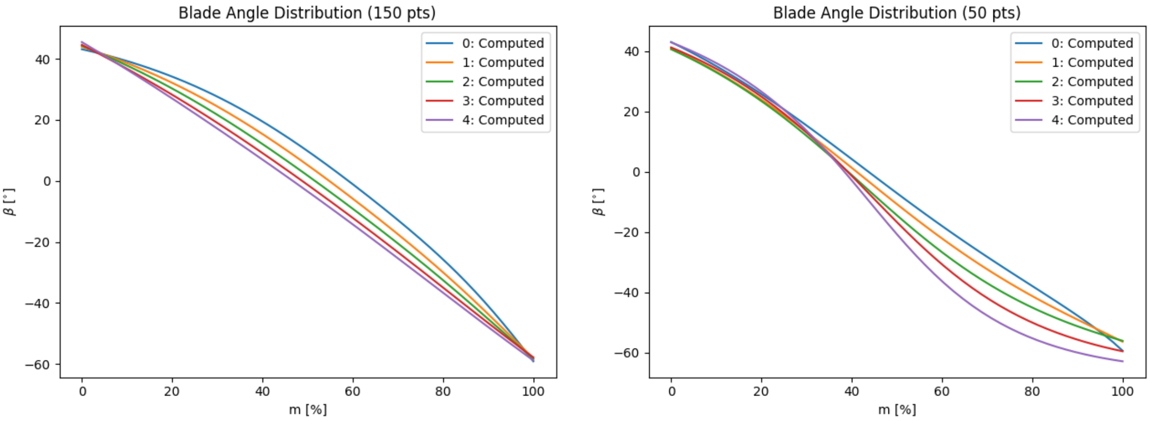
Not only are the number of blades increased for the Optimised geometry, the meridional channel has also been adjust, keeping only the inlet and outlet boundaries constant. Particularly the shape of the hub curve has changed dramatically, as can be seen in fig. 8.10. The different lengths of the hub and shroud curve are given intable 8.5.



(a) Baseline meridional channel

(b) Optimised meridional channel

Figure 8.10: Merdional Channel Comparison



(a) Baseline Blade Angle Distributions

(b) Optimised Blade Angle Distributions

Figure 8.11: Blade Angle Distributions Comparison

8.4.2. Area Distribution Comparison

The corresponding meridional area distributions and B2B area distributions are compared in figs. 8.12 and 8.13. The channel shape mainly influences the meridional area distribution, whilst the blade angles affect the B2B area distribution. The full 3D area distribution is affected by both the channel shape and blade angle distribution, which will be discussed further on.

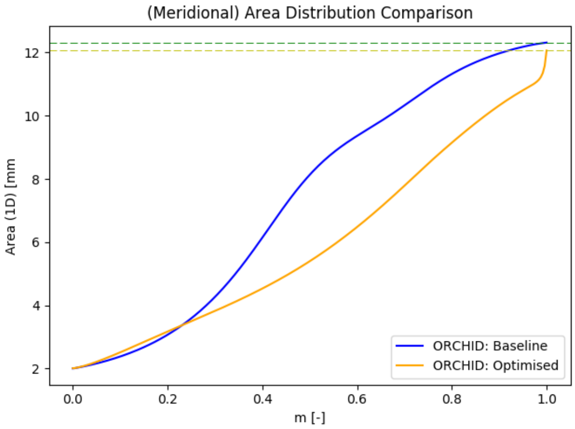
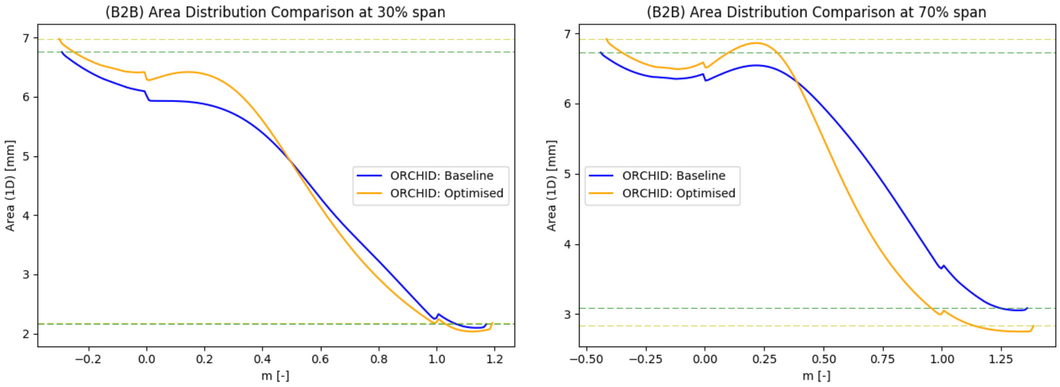


Figure 8.12: Comparison of Meridional Area Distribution



(a) Baseline vs. Optimised, 10% span

(b) Baseline vs. Optimised, 50% span

Figure 8.13: Blade-to-Blade Area Distribution Comparison

It is very interesting to see that the figures indicate that the Optimised ORCHID blade shows more compression along the meridional line compared to the Baseline blade. Particularly at low span, the Baseline compression region is completely absent, whereas the 'Optimised blade' compression region is quite evident in the 10% to 30% meridional length interval. It is, however, also useful to see how these compression regions relate to the overall expansion ratio of the blades, because the plots indicate that the 'Optimised blade' has a slightly higher expansion ratios compared to the 'Baseline blade'.

Span	Area Ratio [-]		Overshoot [mm]		Rel. Overshoot [%]	
	Baseline	Optimised	Baseline	Optimised	Baseline	Optimised
10%	4.2798	4.6685	0.0000	0.0794	0.0000	1.7016
50%	2.6441	2.7870	0.0765	0.2254	2.8918	8.0871
90%	1.8922	2.3165	0.2777	0.4797	14.6783	20.7067

Table 8.6: Area Ratio and Area Peak Overshoot

Finally, the 3D area distribution should be analysed, to see where the performance increase of the optimised geometry comes from.

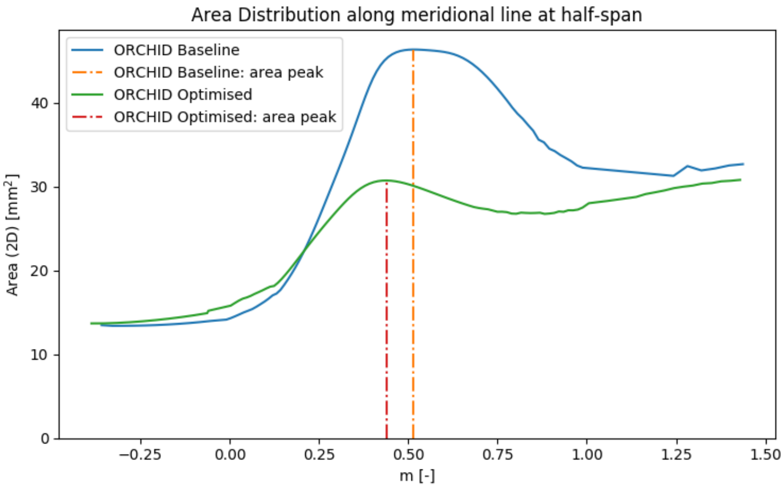
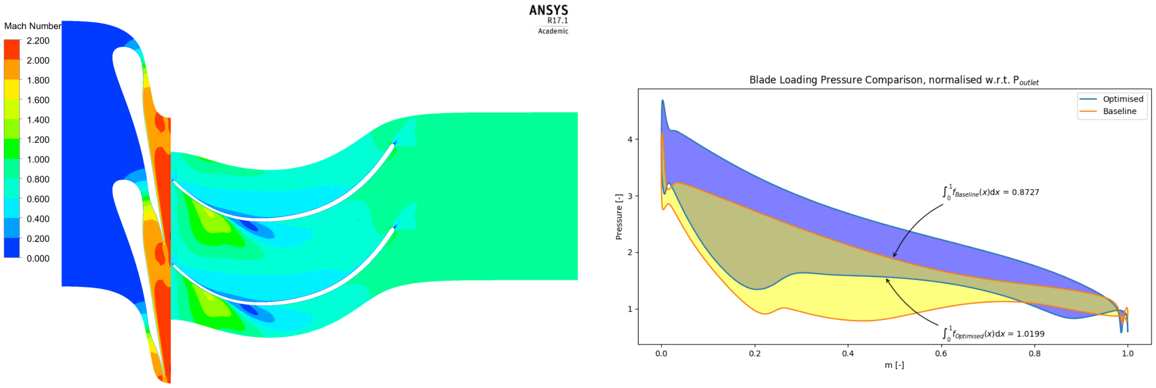


Figure 8.14: Comparison 3D Area Distribution

Here we notice that the area peak is much greater for the Baseline geometry. This area peak was not clearly visible in the B2B area distribution; in fact, the Baseline B2B area distribution looked better better than that of the Optimised ORCHID. However, the Baseline meridional area distribution gave a hint that the full 3D area distribution would show a higher frontal area from m=0.2 onwards. The mach contours of the flow field and the blade loading of the Baseline are investigated to see if this area peak can be seen in the flow data. Figures 8.15a and 8.15b show the CFX anlysed Mach Contour and Blade Loading diagram of the Baseline ORCHID blade, using the same boundary conditions as the Optimised ORCHID analysis.



(a) Blade-to-Blade Mach Contour

(b) Blade Loading Pressure (normalised w.r.t. P_s at the outlet)

Figure 8.15: CFD Results of Baseline ORCIHD using ANSYS CFX

When looking at the blade loading in fig. 8.15b, the slight drop in pressure on suction side of the blade, in the m = 0.225 to 0.6 interval, gives an indication that the flow is forced to expand in such a way that, as in this case is also visible in the B2B mach contour, the flow separates and flow recirculation occurs. It is very likely that the high area peak that is visible in fig. 8.14 is causing this recirculation, simply by expanding the flow too quickly, creating an adverse pressure gradient that causes separation.

8.5. Preliminary Optimisation Cycle

The applicability and usability of the Optiblade design and analysis tool can really be tested when using it in an optimisation design cycle. As a proof of concept, we want to reproduce the optimisation conducted by M. Burigana, who started with the ORCHID Baseline geometry. He first performed an optimisation of the Meridional channel, after which he optimised the blade angles using the new channel as a basis. The number of blades was also increased from 15 to 19, but an optimisation of the number of blades will not be part of this study.

First a low fidelity (or preliminary) design optimisation is conducted by analysing and optimising for the Blade-to-Blade and Full-3D area distributions, after which this design is used in a high-fidelity fluid dynamics computation by CFX.

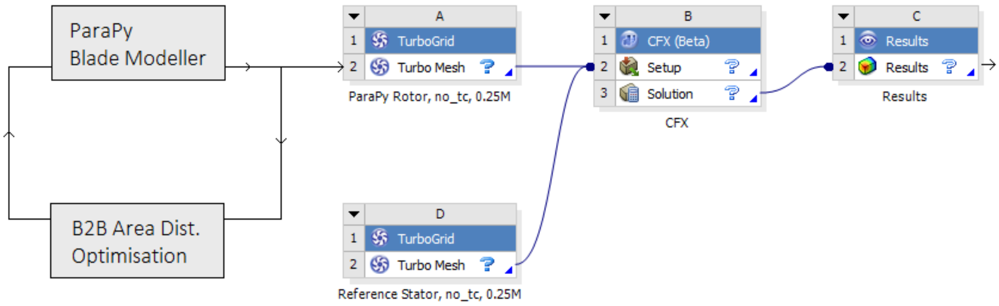


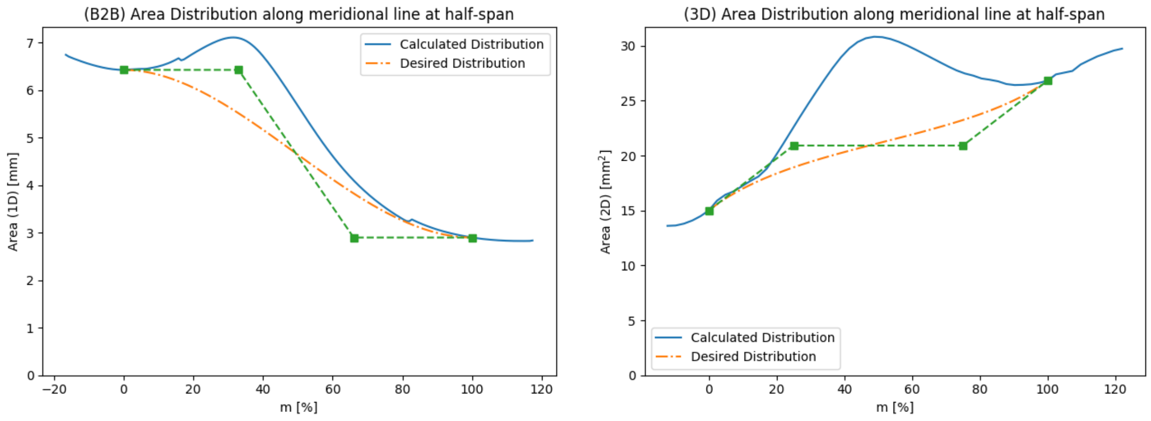
Figure 8.16: B2B optimisation linked to the ANSYS CFX workflow

8.5.1. Minimization Algorithm

For the minimization of the objective function, the SciPy library of python provides several minimization algorithms and functions. Not all of these algorithms are applicable for this particular application. Two algorithms, COBYLA and SLSQP, provide the option of adding constraint functions, which is essential for our current optimisation. For this particular optimisation, the 'SLSQP' algorithm was chosen, since this is the same algorith applied in the fitting methods. This algorithm is even applied in parts of the SU2 CFD computations.

8.5.2. Objective Function

In order to optimise the Blade-to-Blade area distribution of the rotor, one must first determine the desired output. For the expander, a linearly decreasing area distribution would provide a steady expansion along the channel, which reduces not only compression losses, but it also reduces the chance of shock-waves and flow separation. The desired linear area distribution is plotted against the current area distribution, as shown in fig. 8.17a.



(a) Computed Blade-to-Blade Area Distibution vs. Desired B2B Area Distribution

(b) Computed 3D Area Distibution vs. Desired 3D Area Distribution

Figure 8.17: Desired Area Distributions

The to be optimised objective function can be defined as the difference between the linear curve and the computed curve. Note that the computed distribution is calculated from the geometry, which is generated using the design variables and reference values. Similarly, the Desired (Linear) distribution is defined using the inlet and outlet area of the generated geometry. Hence, they are both a function of the design vector $\vec{X}$.

$$f_{obj}(\vec{X}) = \sum_{i=0}^{n} \frac{\left(\text{ComputedDist}(\vec{X})_i - \text{DesiredDist}(\vec{X})_i\right)^2}{n} \tag{8.1}$$

One has to take into consideration the fact that the blade angle at inlet and outlet are optimized considering the flow field (i.e. the rotor inlet flow angle and the rotor flow deflection are optimized, while trying to avoid flow recirculation). However, in this particular optimisation, the blade angle change from inlet to outlet is optimized taking into account the physical consideration of the area distribution. In that sense, only optimising for area distribution could result in a less optimal flow field. It is therefore necessary to not only correctly define which parameters are the design variables, but it is also important to define several constraints and bounds to retain a proper flow field.

8.5.3. Design Variables and Bounds

For the optimisation of the area distributions, it was decided to minimize the number of design variables, and limit the design space to only the meridional channel parameters and the blade angles. In order to investigate the influence of each of the design variables, the optimisation is split into several smaller optimisations, were only a partial design space was explored.

The first optimisation only used the meridional channel parameters for the hub and shroud curves in the optimisation. The inlet and outlet coordinates are fixed, but the parameters that define the shape of the curve are variable, as can be seen in fig. 8.18a

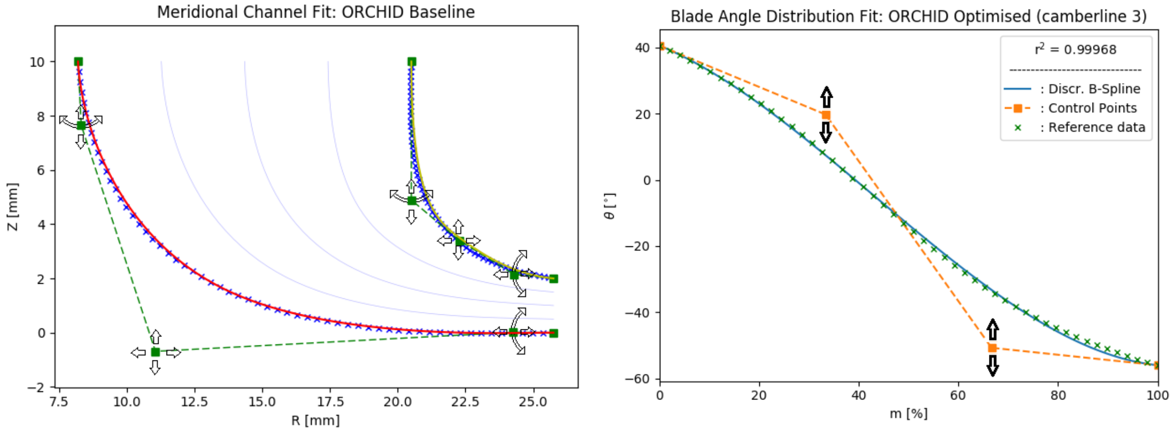
$$\vec{X} = \left[\alpha_{4h}, \alpha_{4s}, \alpha_{5h}, \alpha_{5s}, p_{LE_h}, p_{LE_s}, p_{TE_h}, p_{TE_s}, p_{MIDr_h}, p_{MIDr_s}, p_{MIDz_h}, p_{MIDz_s}\right]^T$$

The second optimisation only used the blade angle distribution of the hub, mid and shroud camberlines in the optimisation. An example of the blade angle distribution that will be explored for camberline 3, is given in fig. 8.18b. Note that the start and end points of the distribution is fixed. This is to ensure that the optimised solution is an improvement of the current design and not a completely new design.

$$\vec{X} = \left[\beta_{1a}, \beta_{1b}, \beta_{2a}, \beta_{2b}, \beta_{3a}, \beta_{3b}\right]^T$$

The third optimisation used the combined design variables of optimisation 1 and 2.

$$\begin{aligned}\vec{X} = \big[&\alpha_{4h}, \alpha_{4s}, \alpha_{5h}, \alpha_{5s}, p_{LE_h}, p_{LE_s}, p_{TE_h}, p_{TE_s}, p_{MIDr_h}, p_{MIDr_s}, p_{MIDz_h}, p_{MIDz_s},\\ &\beta_{1a}, \beta_{1b}, \beta_{2a}, \beta_{2b}, \beta_{3a}, \beta_{3b}\big]^T\end{aligned}$$



(a) Variable Meridional Channel Curves

(b) Variable Blade Angle Distribution

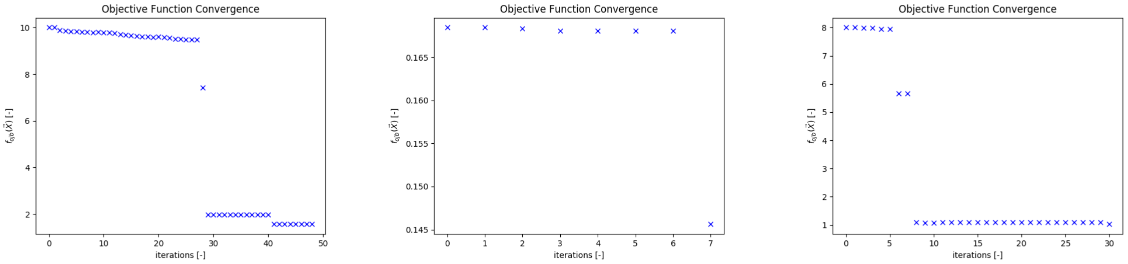
Figure 8.18: Distributions to be explored; Camberline 3

These design variables differ all in type (angles, ratios, and simple floating points) and in bounds. Therefore, each parameter has been normalised with respect to its upper and lower bound (UB and LB) in such a way that the parameter is defined within the interval [0.0, 1.0], as illustrated in eq. (8.2). The using a normalised design vector, the Optimisation function now weights each variable each by apply a relative change with respect to their base value, instead of an absolute change with respect to their base value.

$$x_{norm} = \frac{x - LB}{UB - LB} \quad \longleftrightarrow \quad x = x_{norm} \cdot (UB - LB) + LB \tag{8.2}$$

8.5.4. Optimisation convergence

While the this particular optimisation is far less complex compared to a CFD analysis in ANSYS, the solution convergence can be assessed just as well. Every optimisation returns a scalar function value, which is to be minimized, hence, when the objective function reaches a minimum plateau, the optimisation has converged. The function value is plotted against the iteration count for each optimsation in fig. 8.19 below.



(a) Optimisation 1

(b) Optimisation 2

(c) Optimisation 3

Figure 8.19: Optimisation Function Value Convergence

8.5.5. CFD Analysis of Preliminary Optimised Design

The optimisation module described above has been applied only in the preliminary design phase, before any meshing or CFD operations have been performed. When this preliminary optimisation cycle is completed, the resulting geometry is exported to ANSYS, where it will go through the TurboGrid meshing and CFX analysis modules.

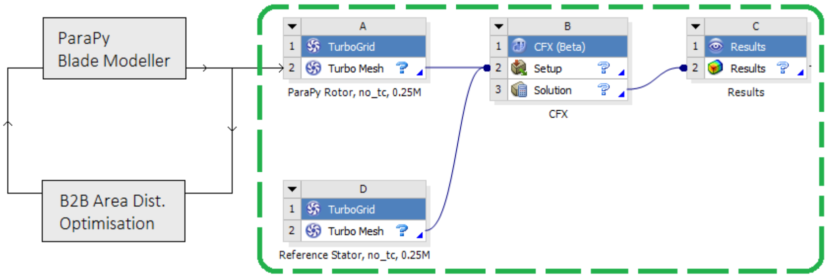


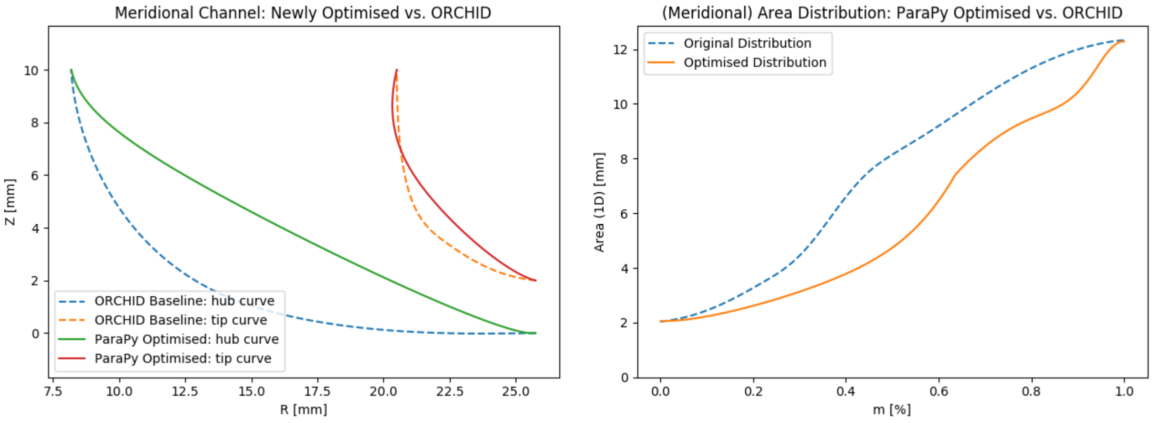
Figure 8.20: The ANSYS CFX workflow

8.6. Optimisation 1: Meridional Channel Optimisation

The first optimisation solution produces a feasible blade design. The resulting geometry and area distributions of the ParaPy optimisation are provided in the next section, and the CFD results of the geometry are provided in the section thereafter.

8.6.1. Area Distribution Optimisation Results

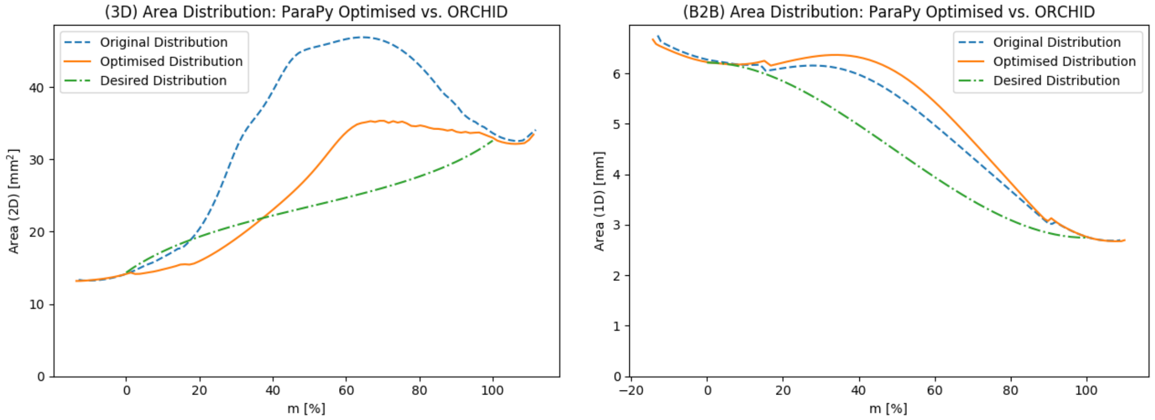
Figures 8.21 and 8.22 below show the resulting meridional channel and corresponding area distributions of the first optimisation.



(a) Meridional Channel Curves

(b) Meridional Area Distribution

Figure 8.21: Optimised Meridional Channel and corresponding Meridional Area Distribution



(a) 3D Area Distribution

(b) B2B Area Distribution

Figure 8.22: Original and Optimised 3D and B2B Area distributions

8.6.2. CFD Results

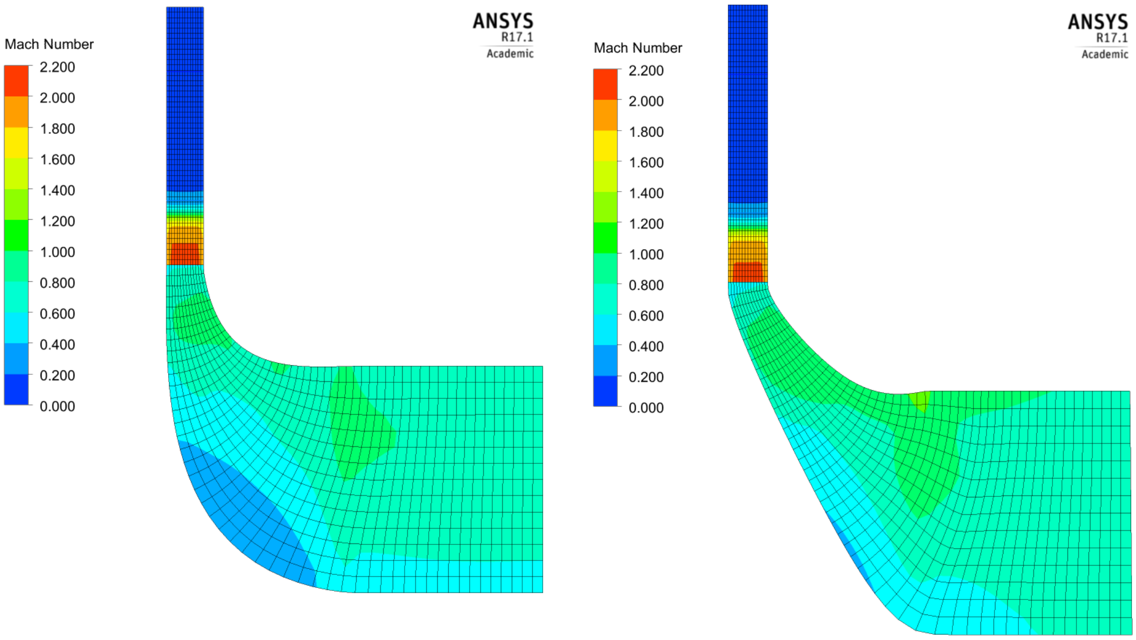
In order to see if the area distribution optimisation has improved the performance of the turbine, the original Baseline blade CFX results are compared with the ParaPy Optimised blade CFX results.

The results of both CFD simulations are discussed in this section using several tables and figures. The general performance characteristics are first compared in table 8.11 below, after which the flow characteristics will be discussed.

Performance Parameter	ORCHID blade	Optimised blade	Relative Improvement
Power	101698 [W]	102108 [W]	+0.40315 [%]
Torque	0.0545516 [Nm]	0.0547711 [Nm]	+0.40237 [%]
Isentropic Expansion Efficiency (t-s)	0.813627 [-]	0.81417 [-]	+0.06673 [%]
Mass flow (total)	0.137112 [kg s^{-1}]	0.137111 [kg s^{-1}]	+0.0 [%]

Table 8.7: Performance Comparison between work-flows

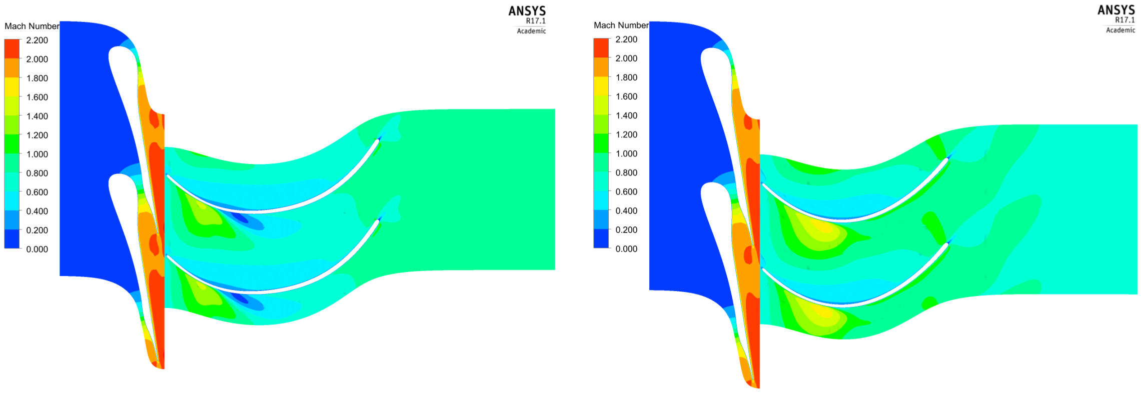
The simulations are very similar in performance, but the optimised blade is slightly better. An improvement of only 0.07% is not a lot, and it might even be contributed to numerical errors. It seems that, quantitatively, the new geometry does not really improve the performance. However, a closer look at the qualitative flow characteristics, improvements can be found. Figures 8.35 and 8.36 present contour plots of the Mach Number in the Meridional plane and in the Blade-to-Blade plane at mid-span. Finally, fig. 8.37 compares the rotor Blade Loading between the two blades.



(a) ORCHID Baseline Rotor

(b) ParaPy Optimised Rotor

Figure 8.23: Comparison of Meridional Mach Contour using ANSYS CFX Results



(a) ORCHID Baseline Rotor

(b) ParaPy Optimised Rotor

Figure 8.24: Comparison of Blade-to-Blade Mach Contour using ANSYS CFX Results

The contour plots above clearly show that the optimised geometry removes most part of the low Mach number contours. This indicates that the separation and flow recirculation is removed in the optimised geometry. This will reduce the entropy creation and should therefore improve the flow conditions and overall turbine performance.

The averaged Mach numbers, at the inlet, outlet and mixing planes are tabulated in table 8.12 below. The data shows that at the stator outlet, the Mach numbers are slightly higher for the optimised blade, which is generally beneficial for the power output.

Location	ORCHID Baseline M_{rel} (M_{abs}) [-]	ParaPy Optimised M_{rel} (M_{abs}) [-]
Stator Inlet	0.0259277	0.0259468
Stator Outlet	2.04249	2.01605
Rotor Inlet	0.605547 (2.04234)	0.575926 (2.01494)
Rotor Outlet	0.78836 (0.576789)	0.769843 (0.575842)

Table 8.8: Local Relative Mach Numbers

The blade loading of the rotor blade is increased dramatically for

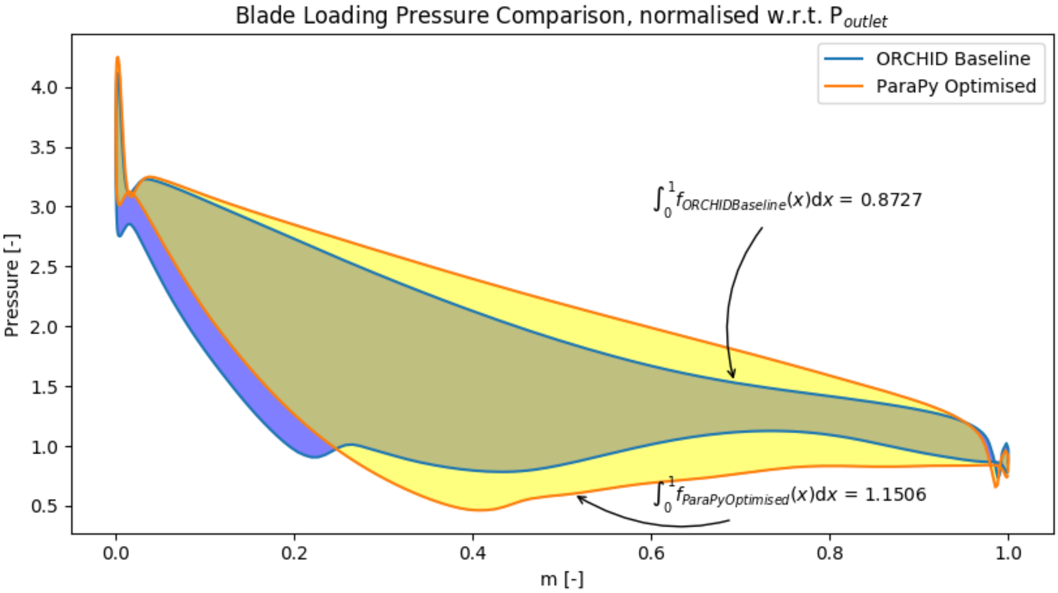
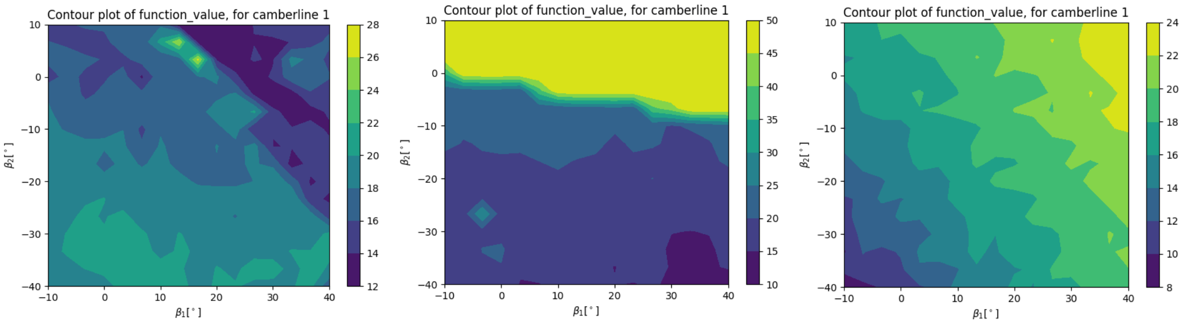


Figure 8.25: Comparison of Blade Loading using ANSYS CFX Results (normalised w.r.t. P_s at the outlet)

8.7. Optimisation 2: Blade-to-Blade Channel Optimisation

Before the blade angle optimisation, an exploration of the design space per camberline was conducted. Each camberline has 2 design variables, hence a simple contour was created per camberline, to inspect the local minima, see fig. 8.26. Note that a function value of 50 indicates an infeasible design.



(a) Hub Camberline parameters (b) Mid Camberline parameters (c) Tip Camberline parameters

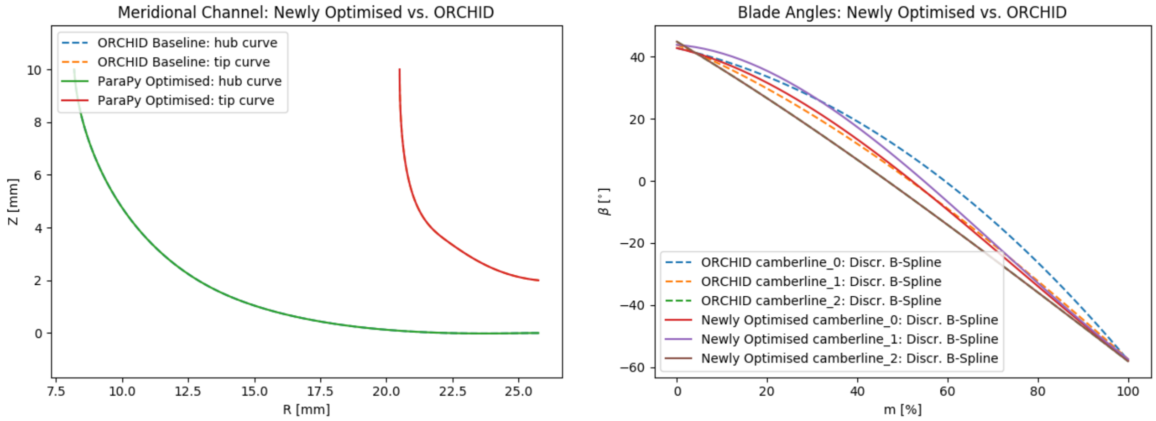
Figure 8.26: Contours of the blade angle design space, per camberline

Only the changes to camberline 2 sometimes produce infeasible designs. Each camberline has one particular local minima and not more than that The big difference in values and contours indicate that it might not be feasible to completely decouple each of the camberlines, and rather optimise all at once, and assess the area distributions at multiple spans.

The next two sections will therefore show the optimisation of all three blade angle distributions together. The meridional channel has been kept the same as the original Baseline blade. Note that, while the original Baseline blade had 5 camberlines, for this optimisation only the hub, mid and tip camberlines are used.

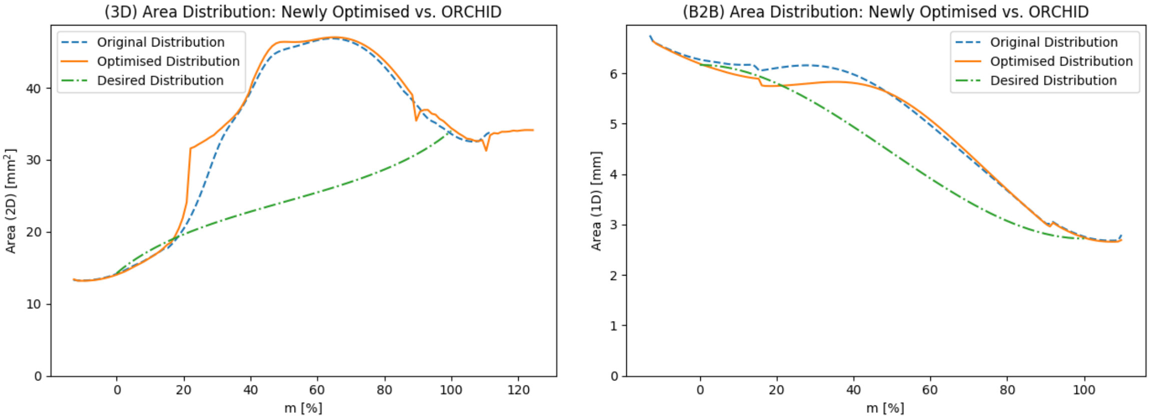
8.7.1. Area Distribution Optimisation Results

Figure 8.27 below shows the (original) meridional channel and optimised blade angle distributions. The corresponding area distributions of the optimisation are shown in fig. 8.28. Note that for this optimisation, the objective function weighted the B2B area distribution higher than the 3D area distribution.



(a) Meridional Channel Curves (b) Meridional Area Distribution

Figure 8.27: Optimised Meridional Channel and Blade Angle Distributions



(a) 3D Area Distribution

(b) B2B Area Distribution

Figure 8.28: Original and Optimised 3D and B2B Area distributions

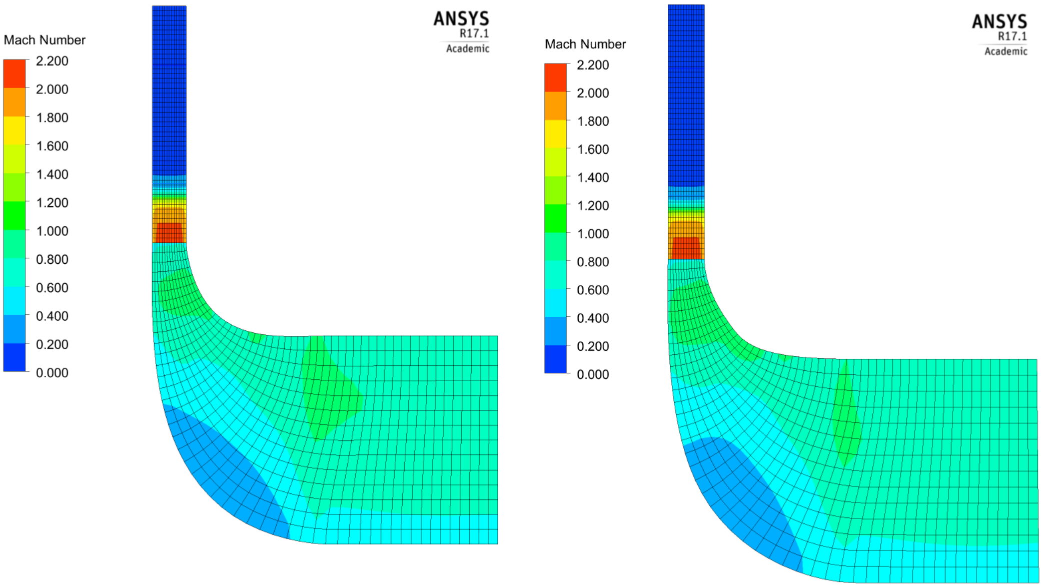
Note that the 3D area distribution shows worse results. The kinks in the curve indicate that during the construction of the flow passage area planes, some local minor numerical instabilities arose. However, these do not disturb the overall trend of the area distribution curve. The B2B area distribution shows a slightly smaller area peak, which should improve the flow in the B2B plane.

8.7.2. CFD Results

The CFD results of the second optimised geometry are shown in the tables and figures below.

Performance Parameter	ORCHID blade	Optimised blade	Relative Improvement
Power	101698 [W]	100670 [W]	-1.0108 [%]
Torque	0.0545516 [Nm]	0.0539997 [Nm]	-1.0117 [%]
Isentropic Expansion Efficiency (t-s)	0.813627 [-]	0.806484 [-]	-0.8779[%]
Mass flow (total)	0.137112 [kg s^{-1}]	0.137111 [kg s^{-1}]	0.0 [%]

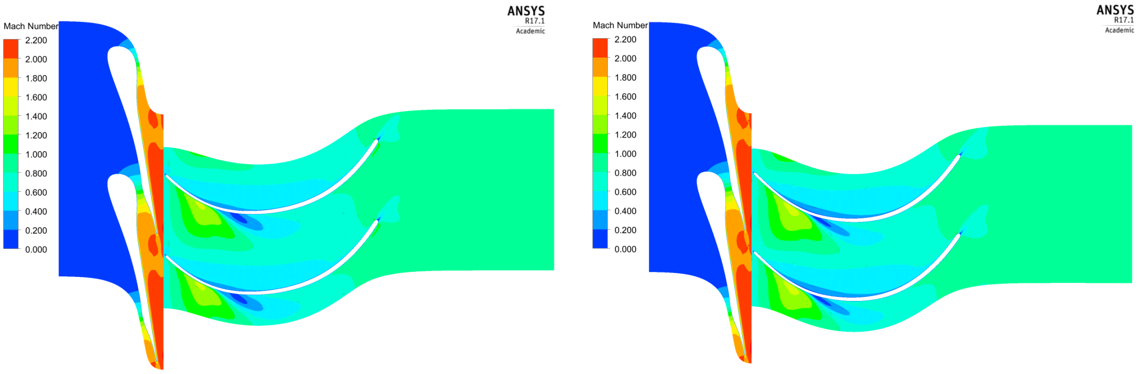
Table 8.9: Performance Comparison between work-flows



(a) Original ORCHID Rotor

(b) ParaPy Optimised Rotor

Figure 8.29: Comparison of Meridional Mach Contour using ANSYS CFX Results



(a) Original ORCHID Rotor

(b) ParaPy Optimised Rotor

Figure 8.30: Comparison of Blade-to-Blade Mach Contour using ANSYS CFX Results

Location	ORCHID Optimised M_{rel} (M_{abs}) [-]	ParaPy Optimised M_{rel} (M_{abs}) [-]
Stator Inlet	0.0259371	0.0259617
Stator Outlet	1.97464	2.04726
Rotor Inlet	0.542248 (1.98652)	0.611639 (2.04652)
Rotor Outlet	0.953754 (0.473172)	0.75609 (0.597496)

Table 8.10: Local Relative Mach Numbers

The blade loading of the rotor blade often gives a good indication of the flow patterns along and close to the blade surface. Particularly the pressure distribution is often used in the literature for comparing blade performance. In fig. 8.37, the blade loading pressure along the meridional line is plotted for both blades. The static pressure is normalized with respect to the outlet static pressure.

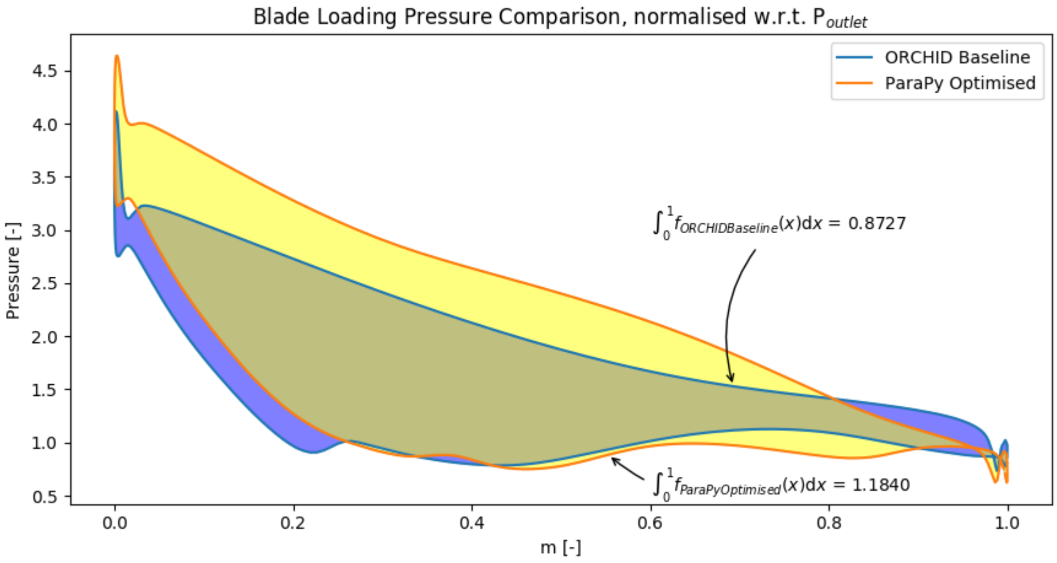


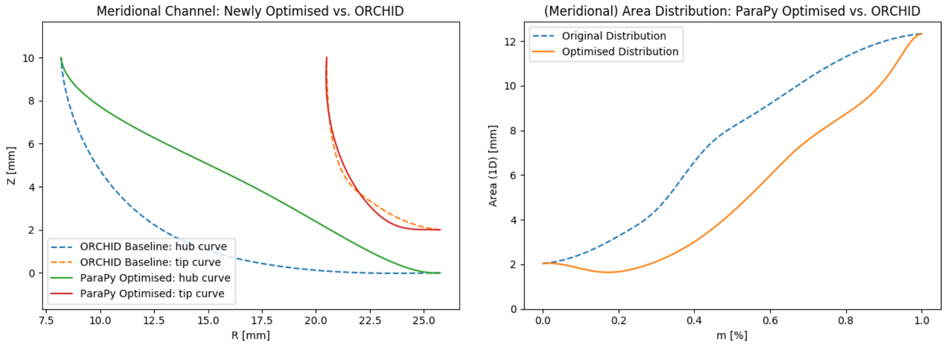
Figure 8.31: Comparison of Blade Loading using ANSYS CFX Results (normalised w.r.t. P_s at the outlet)

8.8. Optimisation 3: Combined Meridional and Blade Angle Optimisation

The sections below describe the results of the optimisation where both the meridional channel and the blade angle distributions were optimised simultaneously.

8.8.1. Area Distribution Optimisation Results

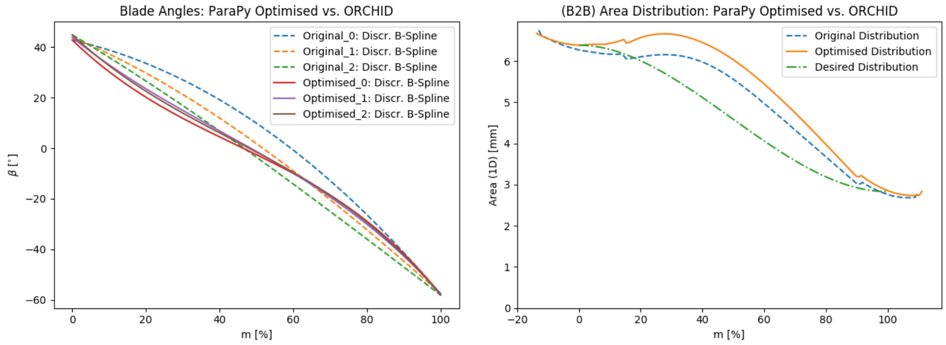
Figures 8.32 to 8.34 below show the resulting meridional channel, the resulting blade angle distributions and the corresponding area distributions of the final optimisation.



(a) Meridional Channel Curves

(b) Meridional Area Distribution

Figure 8.32: Optimised Meridional Channel and corresponding Meridional Area Distribution



(a) 3D Area Distribution

(b) B2B Area Distribution

Figure 8.33: Blade Angle Distribution and corresponding and B2B Area distributions

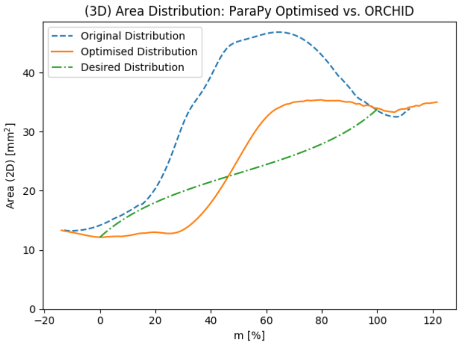


Figure 8.34: 3D Area Distribution

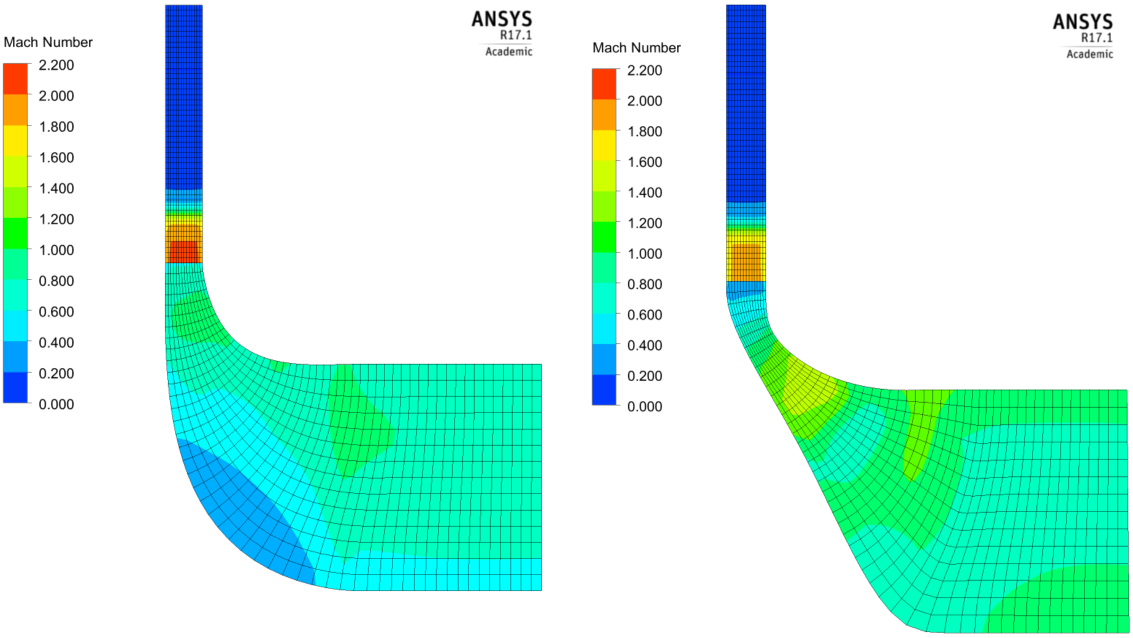
8.8.2. CFD Results

The general performance characteristics are compared in table 8.11 below. The data shows a significant decrease in performance of the turbine. Not only does the efficiency drop by 9%, but also the power is significantly less.

Performance Parameter	**ORCHID blade**	**Optimised blade**	**Relative Improvement**
Power	101698 [W]	92761.9 [W]	-8.7868 [%]
Torque	0.0545516 [Nm]	0.049758 [Nm]	-8.7872 [%]
Isentropic Expansion Efficiency (t-s)	0.813627 [-]	0.740815 [-]	-8.9491 [%]
Mass flow (total)	0.137112 [kg s^{-1}]	0.137111 [kg s^{-1}]	0.0 [%]

Table 8.11: Performance Comparison between work-flows

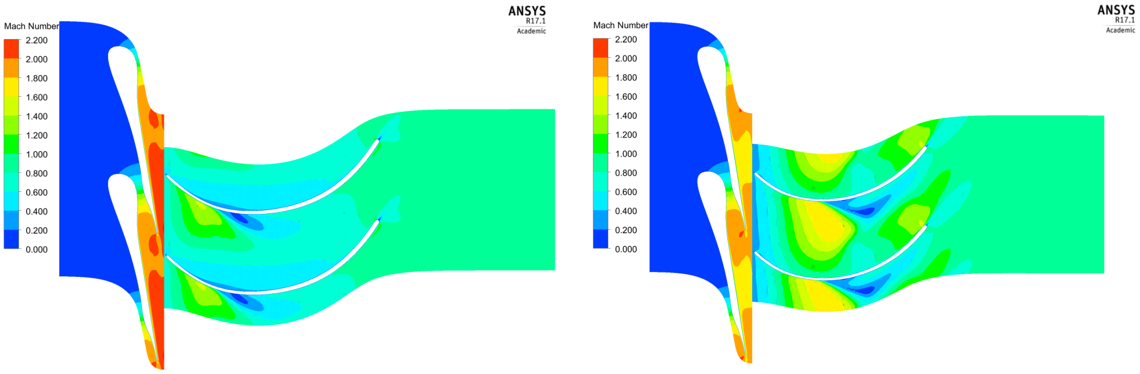
So, according to this data, the performance has decreased significantly. The figures of the mach contours of the meridional and B2B planes below show the origin of this performance drop.



(a) Baseline ORCHID Rotor

(b) ParaPy Optimised Rotor

Figure 8.35: Comparison of Meridional Mach Contour using ANSYS CFX Results



(a) Baseline ORCHID Rotor

(b) ParaPy Optimised Rotor

Figure 8.36: Comparison of Blade-to-Blade Mach Contour using ANSYS CFX Results

The meridional mach contour plot shows a channel with almost fully super-sonic flow speeds. A slight peak in the mach number is present near halfway of the shroud curve. The plot does not show significant flow problems. However, the B2B contour plots above clearly show that the optimised geometry is worse than the original Baseline blade. The separation and recirculation of the flow is more evident in the Optimised plot. This suggests that the efficiency will drop significantly compared to the Baseline, which is indeed the case. The separation is most likely caused by the slightly converging duct near the inlet of the meridional channel. This increased the local mach number, which caused an adverse pressure gradient that instantiated separation.

Location	ORCHID Optimised M$_{rel}$ (M$_{abs}$) [-]	ParaPy Optimised M$_{rel}$ (M$_{abs}$) [-]
Stator Inlet	0.0259371	0.0259495
Stator Outlet	1.97464	1.78998
Rotor Inlet	0.542248 (1.98652)	0.403793 (1.7848)
Rotor Outlet	0.953754 (0.473172)	0.880772 (0.585065)

Table 8.12: Local Relative Mach Numbers

In fig. 8.37, the blade loading pressure shows significantly higher pressure near the leading edge of the Optimised blade. This is also quite evident in the B2B mach contours, since the mach number at that location is much lower for the optimised geometry, hence the pressure is increase significantly. The steep adverse pressure gradient visible in blade loading plot is a good indication that separation of the flow is likely to occurs. The pressure side pressure distribution is affected by the significant suction side pressure drop of the neighbouring blade.

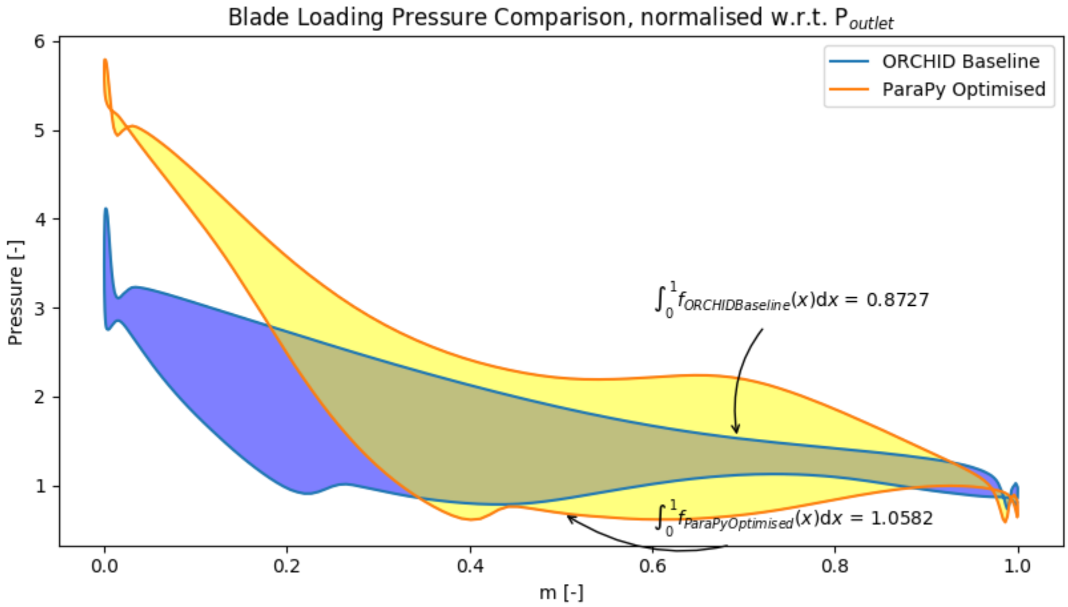


Figure 8.37: Comparison of Blade Loading using ANSYS CFX Results (normalised w.r.t. P_s at the outlet)

It can be concluded, that this particular optimisation was not successful. While the 3D area distribution of the geometry has improved, the changes in shape were detrimental to the overall performance of the turbine.

9
Conclusions & Recommendations

The mini-ORC radial inflow turbine has proven to be one of the most efficient and cost-effective expanders available for low power decentralised power generation. However, this type of turbine is still rarely applied due to the gap in technology of miniature ORC radial inflow turbines. As such the design process of such a blade is also complicated and time consuming. Considering the decreasing time-frame of design projects, the use of prescribed methods and assumptions in an integrated KBE design tool to bride the gap, can provide a better alternative to the current design process. The integrated turbomachinery design tool, produced during this thesis, has done just that.

9.1. Conclusions

This thesis has attempted to present all current state-of-the-art methods and KBE approaches in parametrisation, geometrical analysis and mesh generation of radial inflow turbomachinery blades. The methods and theories investigated provide a new view on the radial inflow turbomachinery design cycle. It combines multiple methods from radial turbomachinery design and advanced curve and surface modelling techniques in a flexible manner to accommodate for a wide range of exotic architecture radial inflow turbomachinery designs.

The investigated methods are combined in a modular and user friendly design tool, which provides a solid and accessible basis for further development of the (radial inflow) turbomachinery design cycle. These methods include: the definition of the meridional channel curves using a set of BSpline curves, the transformation of the 2D channel curves into the 3D camberlines through the blade angle integrator, and the projection of a thickness distribution onto the 3D cambersurface.

The tool has a set of fitting capabilities, that enable the tool to reconstruct a given geometry from a set of distributions and coordinates. It uses a Least-Squares algorithm to fit 2D meridional curves to coordinates from literature, and it fits the blade (or wrap) angle and thickness distribution curves for each camberline, to the provided data sets. The combination of these fitting processes result in a fully reconstructed 3D blade.

The application and automisation of the well-established AutoGrid software, accessible within the turbomachinery design tool, proves to be a solid and valid implementation of mesh generation software in the turbomachinery design cycle. The integrated software package is able to generate high-quality meshes for the (sub-sonic) stator, rotor and diffuser. For these meshes, the Autogrid default grid topology (HOH-grid with additional H-grids at the periodic boundaries) is used, which tends to produce good quality meshes. The skewness and aspect ratio of the elements are also well within the bounds necessary for a quality mesh.

The design tool also provides capabilities for mesh generation using ANSYS TurboGrid and Salome. While ANSYS TurboGrid is a well-established and validated mesh generator, it provides less flexibility in meshing compared to the open-source Salome mesh generator. The design tool has implemented several 'rules-of-thumb for mesh generation' into the Salome package, such as the number of elements per region, or the calculation of the viscous layer thickness for a proper y+ parameter. These methods resulted in a Salome mesh package that can produce high-quality hybrid grids.

With the application of the tool on the design process of the RIT ORCHID blade, the design tool has proven to be a valuable tool for the Turbomachinery department at the faculty of Aerospace Engineering. It not only provides the means to easily reconstruct any given rotor blade, it also provides more insight into the geometrical aspects of the flow passage and the blade geometry compared to the well-established ANSYS BladeGen module.

The tool can analyse both the area distribution of the blade passage, as well as the curvature distribution of the blade passage bounds (the blade, hub and shroud). These two parameters are deemed most important for flow characteristics as they can influence flow expansion and flow acceleration.

The meridional channel definition seems to have the greatest influence on the Meridional and the 3D area distribution, while the blade angle distribution primarily influences B2B area distribution. It has been shown that reducing peaks in the 3D area distribution reduces the occurrence of flow separation at the suction side of the rotor. This is backed-up by the fact that the geometry with the lowest area peak has the highest efficiency.

From this we hope to conclude that it is simply a matter of removing the peaks and steep gradients in the area distribution. The optimisations, however, have shown that the performance is not only driven by the area distribution, in fact, an area distribution with lower peak could still produce geometries with much lower performance. The 3D effects and other flow effects that are not captured by the expansion or contraction of the flow area seem to have significant effect on the turbine performance. The relation between geometry of the flow passage and the flow characteristics should be investigated to a further extend.

9.2. Outcomes

While this thesis has already attempted to cover all research questions posed in the introduction, several outcomes and highlights have not yet been covered properly. The following outcomes should be highlighted:

- the parametrisation methods used provide a means to accurately reconstruct any type of radial inflow turbine architecture (both with zero and non-zero blade inflow angles) with performance errors less than 0.25%.

- the pre-CFD geometrical analysis of the parametrised geometry can provide a means to improve a geometry up-front before moving to the computationally intensive CFD simulations

- the prescribed curvature method can also be applied to radial turbomachinery blades to try to improve performance, but the method severely constrains the shape of the blade profile, it might therefore be better to analyse the curvature after the initial blade profile construction and then try to optimise. This will allow the tool to be more flexible in the blade profile construction.

- the multi-block grid generation method is currently too difficult to implement in the design tool due to the complex shape of the rotor, but the unstructured mesh with additional viscous layer (hybrid mesh) used in the design tool gives a decent result.

In the end, the integrated turbomachinery design tool is user-friendly and modular tool, that is able to generate an optimised or predefined radial inflow turbine design.

9.3. Recommendations

Throughout the course of the thesis, many challenges were encountered, new information was discovered, and new conclusions were drawn, which paved the way for the course of the project. Not every pathway in this thesis has been investigated or attempted, due to lack of time or being outside of the scope of the project. Therefore many paths can still be taken to follow-up on the research of this thesis. To give a brief overview of the possible paths that can be taken, the following recommendations are presented:

Validation of the Salome Hybrid mesh - While the meshes created for validation of the parametrisation were generated using AutoGrid, the tool supports the capability of generating high-quality hybrid meshes using Salome. It is recommended that these hybrid meshes are compared with their structured AutoGrid counterpart, to assess the quality of the mesh. The use of hybrid meshes allows for very unconventional blade architectures to be meshed with relative ease. Therefore, should the assessment of the hybrid meshes capability prove that the meshes produced are of reasonable quality by providing reasonable CFD results, this capability could be utilised for the more exotic turbine architectures.

Development of strategic partitioning methods for Salome Structured mesh - Unfortunately, the use of Structured Grids within the Salome environment did not yet produce meshes of sufficient quality for a successful CFD analysis. The primary reason for the low quality of the structured meshes, was the low-quality definitions of the the multi-block grid. Particularly the partitions of the blocks at the trailing edge were so poorly defined, that the grid in the rotor wake could not be properly discretised. It is very much recommended to start looking into the partitioning of the grid block, particularly for highly staggered configurations. Many methods are available in the literature, so it may be wise to assess several of them and see which block topology can be implemented and ensures a high quality discretisation. Care should be taken with the periodicity; the partitioned multi-blocks should match in periodicity.

Evaluate grid convergence for ORC specific flow simulations - The meshes generated using Salome use KBE by applying several prescribed 'rules-of-thumb' during the meshing process, these rules are defined for general turbomachinery meshing purposes. These rules may be less valid for simulations regarding ORC turbines, as the flow solutions may require different mesh-refinement or mesh-quality levels for an accurate flow simulation. It is therefore recommended to assess the meshes produced for both regular and organic fluids, to see if mesh quality can be increased. This can be assessed by performing a grid convergence study for several different fluids.

Investigate and Validate the CFD configuration in SU2 - Should this research be continued, it is recommended that the focus is shifted towards the CFD solver part of the cycle. The tool currently has only been validated with respect to other computed models, and not with respect to experimental results. The results of the CFX comparison clearly indicate that there is a great difference in calculated efficiency, compared to experimental results. This type of validation however, is beyond the scope of the thesis, as it not only relies on proper parametrisation and meshing of a geometry, it also requires very extensive knowledge of all CFD solver models and settings. This is particularly difficult for ORC turbomachinery flow solutions, for which the flow is highly turbulent, it involves both subsonic and supersonic flow regions, and non-ideal gas thermodynamics is necessary.

Since the design tool can reproduce blade architectures from a given data set, several sets of experimental data can be used to analyse and compare the CFD solutions of SU2 with the experimental results. By investigating how the flow parameters and flow models affect the CFD solutions, an effort can be made to approach the experimental data. The knowledge gained from these investigations can be implemented as a set of methods and design-rules in the design tool. This can be implemented as a separate modular CFD package similar to how the AutoGrid, Salome and TurboGrid packages were implemented.

Implementation of the design tool in the TU Delft computational cluster - The optimisation module currently implemented in the design tool only performs a preliminary optimisation, based on geometrical properties only. While this is a computationally cheap optimisation, it does not directly optimised the overall turbine performance, and is therefore not accurate enough for a high-level optimisation. It has been shown that the design tool can perform one full design cycle, so for the final high-level optimisation of the blade, which iterates over the full design cycle, highly intensive CFD computations will need to be performed to assess the turbine performance directly. Therefore the design tool should be made compatible with the Computational Cluster of the TU Delft. The current optimisation module can be appended to included several advanced optimisation schemes.

While currently, the ParaPy software tool only runs on a Windows operating system, the software package will soon be expanded to both Macintosh OS and Linux. This will allow the ParaPy software to run on the TU Delft computation cluster. This will allow the parametrisation and meshing to be conducted in ParaPy and subsequently, the newly produced mesh can be analysed in the cluster using SU2.

This thesis has shown many promising results, but is has also shown many areas for improvement. The use of the ParaPy software package, which has been licensed by the TU Delft, ensures the possibility of continuation of this research project. The Flight Performance and Propulsion Master Track provides an extensive core course on this software package. This helps towards the continuation of this project, as it removes the initial effort of learning to use the software package. The design tool has been set up and is documented in such a way that the tool can be easily improved and extended.

A

ORCHID fitting

The figures below show the full fitting of the ORCHID Optimised blade.

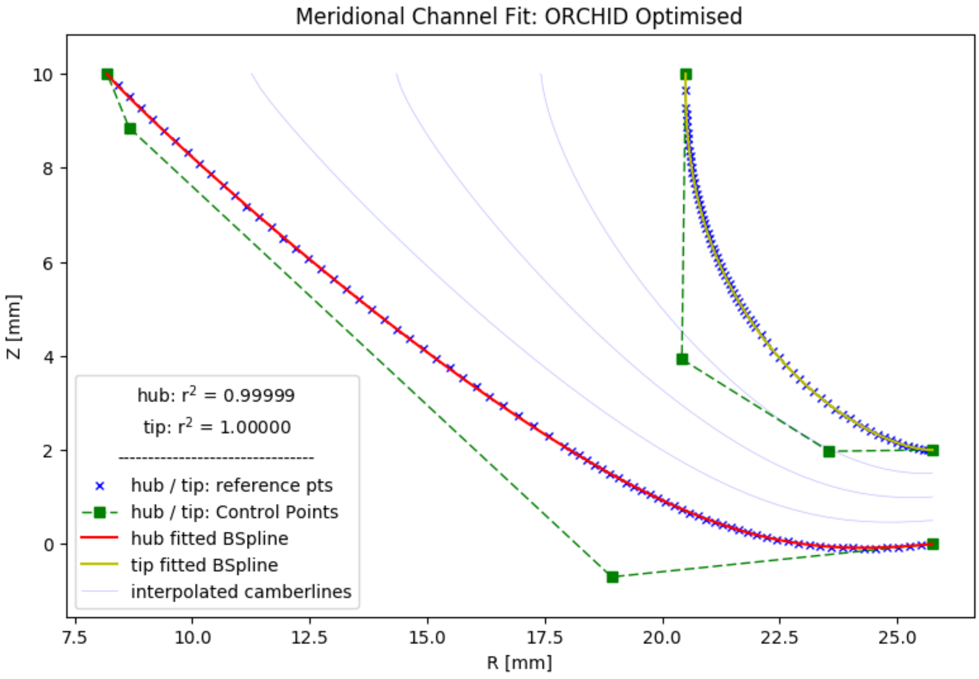
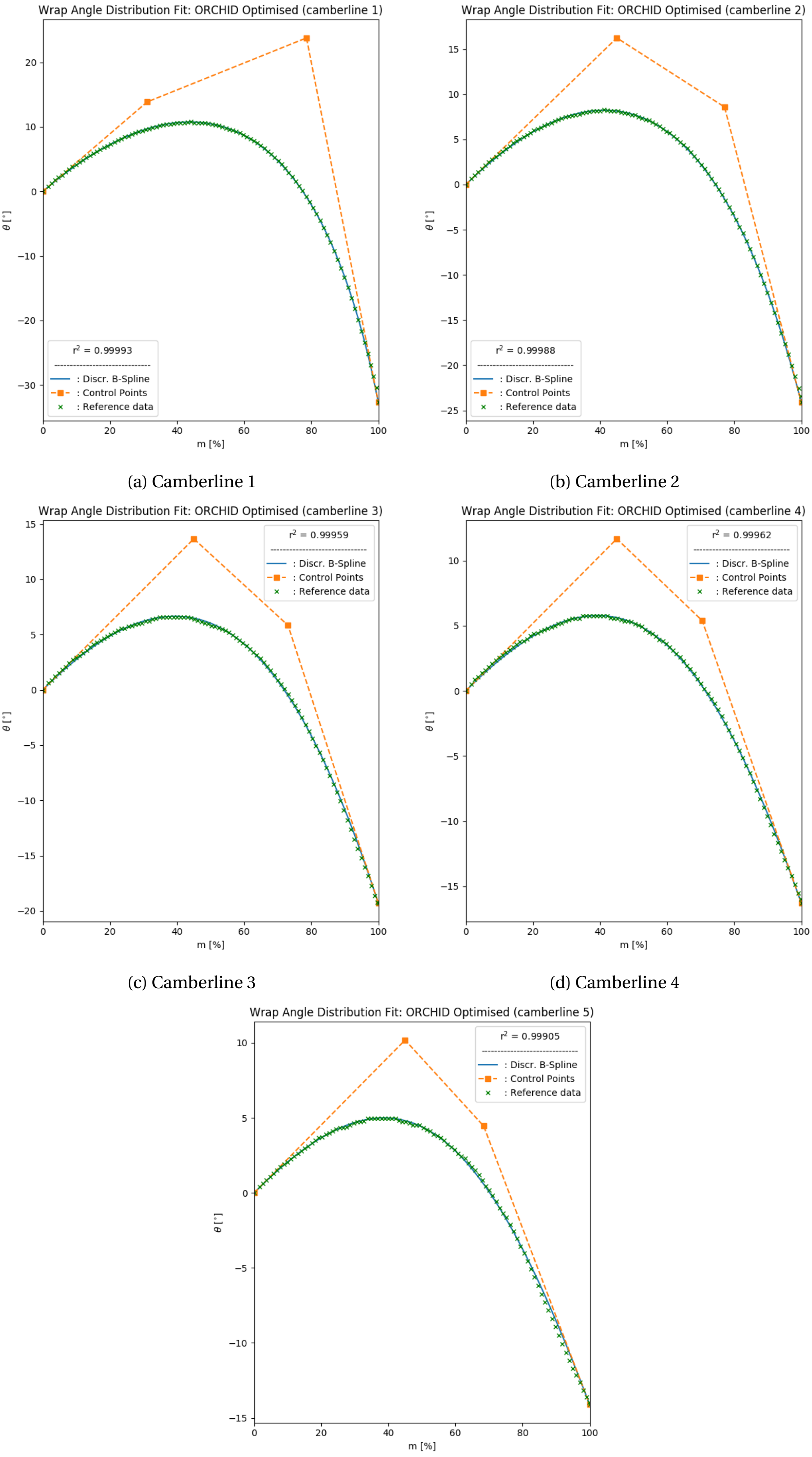


Figure A.1: Fit of Meridional Channel

The fitted BSplines of the Wrap Angle distributions are presented in the figures below.



(a) Camberline 1

(b) Camberline 2

(c) Camberline 3

(d) Camberline 4

(e) Camberline 5

Figure A.2: Fit of Wrap Angle Distribution

B SU2 cfg file

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% SU2 configuration file                                                       %
% Case description: AACHEN turbine 3D 1.5 stage                                 %
% Author: S. Vitale                                                            %
% Institution: Delft University of Technology                                  %
% Date: Feb 28th, 2017                                                         %
% File Version 5.0.0"cardinal"                                                 %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
% ------------- DIRECT, ADJOINT, AND LINEARIZED PROBLEM DEFINITION ------------%
%
% Physical governing equations (EULER, NAVIER_STOKES,
%                               WAVE_EQUATION, HEAT_EQUATION, LINEAR_ELASTICITY,
%                               POISSON_EQUATION)
PHYSICAL_PROBLEM= RANS
%
% Specify turbulent model (NONE, SA, SST)
KIND_TURB_MODEL= SST
%
% Mathematical problem (DIRECT, ADJOINT, LINEARIZED)
MATH_PROBLEM= DIRECT
%
% Restart solution (NO, YES)
RESTART_SOL= YES
%
%
% -------------------- COMPRESSIBLE FREE-STREAM DEFINITION --------------------%
%
% Mach number (non-dimensional, based on the free-stream values)
MACH_NUMBER= 0.05
%
% Angle of attack (degrees, only for compressible flows)
AoA= 0.0
%
% Free-stream pressure (101325.0 N/m^2 by default, only Euler flows)
FREESTREAM_PRESSURE= 400.0E+03
```

```
%
% Free−stream temperature (273.15 K by default)
FREESTREAM_TEMPERATURE= 470.5
%
% Free−stream temperature (1.2886 Kg/m3 by default)
FREESTREAM_DENSITY= 1.7418
%
% Free−stream option to choose if you want to use Density (DENSITY_FS) or Temperature (TEMPERATU
FREESTREAM_OPTION= TEMPERATURE_FS
%
% Free−stream Turbulence Intensity
FREESTREAM_TURBULENCEINTENSITY = 0.05
%
% Free−stream Turbulent to Laminar viscosity ratio
FREESTREAM_TURB2LAMVISCRATIO = 100.0
%
% Reynolds number (non−dimensional, based on the free−stream values)
%REYNOLDS_NUMBER= 6.0E5
%
%Init option to choose between Reynolds (default) or thermodynamics quantities for initializing
INIT_OPTION= TD_CONDITIONS
%
% −−−−−−−−−−−−−−−−−−−−−−−− REFERENCE VALUE DEFINITION −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−%
%
% Reference origin for moment computation
REF_ORIGIN_MOMENT_X = 0.00
REF_ORIGIN_MOMENT_Y = 0.00
REF_ORIGIN_MOMENT_Z = 0.00
%
% Reference length for pitching, rolling, and yawing non−dimensional moment
REF_LENGTH_MOMENT= 1.0
%
% Reference area for force coefficients (0 implies automatic calculation)
REF_AREA= 1.0
%
%
% Flow non−dimensionalization (DIMENSIONAL, FREESTREAM_PRESS_EQ_ONE,
%                              FREESTREAM_VEL_EQ_MACH, FREESTREAM_VEL_EQ_ONE)
REF_DIMENSIONALIZATION= DIMENSIONAL
%
%
% −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− EQUATION OF STATE −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−%
%
% Different gas model (STANDARD_AIR, IDEAL_GAS, VW_GAS, PR_GAS)
FLUID_MODEL= IDEAL_GAS
%
% Ratio of specific heats (1.4 default and the value is hardcoded
%                          for the model STANDARD_AIR)
GAMMA_VALUE= 1.4
%
% Specific gas constant (287.058 J/kg*K default and this value is hardcoded
%                        for the model STANDARD_AIR)
GAS_CONSTANT= 287.06
%
% Critical Temperature (273.15 K by default)
CRITICAL_TEMPERATURE= 514.71
```

```
%
% Critical Pressure (101325.0 N/m^2 by default)
CRITICAL_PRESSURE= 6268000.0
%
% Acentri factor (0.035 (air))
ACENTRIC_FACTOR= 0.646
%
%
% ---------------------------- VISCOSITY MODEL ---------------------------------%
%
% Viscosity model (SUTHERLAND, CONSTANT_VISCOSITY).
VISCOSITY_MODEL= SUTHERLAND
%
% Molecular Viscosity that would be constant (1.716E-5 by default)
MU_CONSTANT= 1.35E-5
%
% Sutherland Viscosity Ref (1.716E-5 default value for AIR SI)
MU_REF= 1.716E-5
%
% Sutherland Temperature Ref (273.15 K default value for AIR SI)
MU_T_REF= 273.15
%
% Sutherland constant (110.4 default value for AIR SI)
SUTHERLAND_CONSTANT= 110.4
%
%
% ---------------------------- THERMAL CONDUCTIVITY MODEL ----------------------%
%
% Conductivity model (CONSTANT_CONDUCTIVITY, CONSTANT_PRANDTL).
CONDUCTIVITY_MODEL= CONSTANT_PRANDTL
%
% Molecular Thermal Conductivity that would be constant (0.0257 by default)
KT_CONSTANT= 0.035
%
%
% --------------------- BOUNDARY CONDITION DEFINITION --------------------------%
%
% Navier-Stokes wall boundary marker(s) (NONE = no marker)
MARKER_HEATFLUX= (blade1, 0.0, shroud1, 0.0,  hub1, 0.0, blade2, 0.0, shroud2, 0.0, hub2, 0.0)
%
%
%MARKER_SHROUD= (shroud2)
% Periodic boundary marker(s) (NONE = no marker)
% Format: ( periodic marker, donor marker, rot_cen_x, rot_cen_y, rot_cen_z, rot_angle_x-axis, rot_angl
MARKER_PERIODIC= (per1, per2, 0.0, 0.0, 0.0, 0.0, 0.0, -18.947368421, 0.0, 0.0, 0.0, per3, per4, 0.0,
%
%
%-------- INFLOW/OUTFLOW BOUNDARY CONDITION SPECIFIC FOR TURBOMACHINERY --------%
%
% Inflow and Outflow markers must be specified, for each blade (zone), following the natural groth of
MARKER_TURBOMACHINERY= (inflow, outmix, inmix, outflow)
%
% Mixing-plane interface markers must be specified to activate the transfer of information between zon
MARKER_MIXINGPLANE_INTERFACE= (outmix, inmix)
%
% Non reflecting boundary condition for inflow, outfolw and mixing-plane
% Format inlet:  ( marker, TOTAL_CONDITIONS_PT, Total Pressure , Total Temperature, Flow dir−nor
% Format outlet: ( marker, STATIC_PRESSURE, Static Pressure value, −, −, −, −, under−relax−avg,
% Format mixing−plane in and out: ( marker, MIXING_IN or MIXING_OUT, −, −, −, −, −, −, under−rel
MARKER_NRBC= (inflow, TOTAL_CONDITIONS_PT, 413.6E+03, 477.6, 1.0, 0.0, 0.0, 1.0, 0.0, outmix, MI
%
%
NRBC_EXTRA_RELAXFACTOR= (0.05, 0.05)
%
SPATIAL_FOURIER= NO
%—————————————————————— TURBOMACHINERY SIMULATION ——————————————————————%
%
% Specify kind of architecture (AXIAL, CENTRIPETAL, CENTRIFUGAL, CENTRIPETAL_AXIAL)
TURBOMACHINERY_KIND= CENTRIPETAL CENTRIPETAL_AXIAL
%
% Specify kind of interpolation for the mixing−plane (LINEAR_INTERPOLATION, NEAREST_SPAN, MATCHI
MIXINGPLANE_INTERFACE_KIND= NEAREST_SPAN
%
% Specify option for turbulent mixing−plane (YES, NO) default NO
TURBULENT_MIXINGPLANE= YES
%
% Specify ramp option for Outlet pressure (YES, NO) default NO
RAMP_OUTLET_PRESSURE= NO
%
% Parameters of the outlet pressure ramp (starting outlet pressure, updating−iteration−frequency
RAMP_OUTLET_PRESSURE_COEFF= (400000.0, 10.0, 1000)
%
% Specify ramp option fr rotating frame (YES, NO) default NO
RAMP_ROTATING_FRAME= NO
%
% Parameters of the rotating frame ramp (starting rotational speed, updating−iteration−frequency
RAMP_ROTATING_FRAME_COEFF= (0.0, 39.0, 500)
%
% Specify Kind of average process for linearizing the Navier−Stokes equation at inflow and outfl
% (ALGEBRAIC, AREA, MASSSFLUX, MIXEDOUT) default AREA
AVERAGE_PROCESS_KIND= MIXEDOUT
%
% Specify Kind of average process for computing turbomachienry performance parameters
% (ALGEBRAIC, AREA, MASSSFLUX, MIXEDOUT) default AREA
PERFORMANCE_AVERAGE_PROCESS_KIND= MIXEDOUT
%
%Parameters of the Newton method for the MIXEDOUT average algorithm (under relaxation factor, to
MIXEDOUT_COEFF= (1.0, 1.0E−05, 15)
%
% Limit of Mach number below which the mixedout algorithm is substituted with a AREA average alg
AVERAGE_MACH_LIMIT= 0.05
%
%
% ——————————————————— SURFACES IDENTIFICATION ———————————————————%
%
% Marker(s) of the surface in the surface flow solution file
MARKER_PLOTTING= (blade1, blade2)
%
%
% ——————————————————— GRID ADAPTATION STRATEGY ———————————————————%
%
% Kind of grid adaptation (NONE, PERIODIC)
```

```
KIND_ADAPT= PERIODIC
%
%
% ———————————————————— DYNAMIC MESH DEFINITION —————————————————————————%
%
% Dynamic mesh simulation (NO, YES)
GRID_MOVEMENT= YES
%
% Type of dynamic mesh (NONE, ROTATING_FRAME)
GRID_MOVEMENT_KIND= ROTATING_FRAME ROTATING_FRAME
%
% Motion mach number (non−dimensional). Used for intitializing a viscous flow
% with the Reynolds number and for computing force coeffs. with dynamic meshes.
MACH_MOTION= 0.35
%
% Angular velocity vector (rad/s) about the motion origi. Example 1250 RPM −> 130.89969389957471 rad/s
ROTATION_RATE_Z  = 0.0   7508.3
%
%
% ——————————— COMMON PARAMETERS DEFINING THE NUMERICAL METHOD —————————————%
%
% Numerical method for spatial gradients (GREEN_GAUSS, WEIGHTED_LEAST_SQUARES)
NUM_METHOD_GRAD= WEIGHTED_LEAST_SQUARES
%
% Courant−Friedrichs−Lewy condition of the finest grid
CFL_NUMBER= 5.0
%
% Adaptive CFL number (NO, YES)
CFL_ADAPT= NO
%
% Parameters of the adaptive CFL number (factor down, factor up, CFL min value, CFL max value )
CFL_ADAPT_PARAM= ( 1.3, 1.2, 5.0, 30.0)
%
%
% ————————————————————— LINEAR SOLVER DEFINITION ————————————————————————%
%
% Linear solver or smoother for implicit formulations (BCGSTAB, FGMRES, SMOOTHER_JACOBI, SMOOTHER_ILU
LINEAR_SOLVER= FGMRES
%
% Preconditioner of the Krylov linear solver (ILU0, LU_SGS, LINELET, JACOBI)
LINEAR_SOLVER_PREC= LU_SGS
%
% Min error of the linear solver for the implicit formulation
LINEAR_SOLVER_ERROR= 1E−4
%
% Max number of iterations of the linear solver for the implicit formulation
LINEAR_SOLVER_ITER= 10
%
%
% —————————————————————— MULTIGRID PARAMETERS ——————————————————————————%
%
% ————————— NOT WORKING WITH PERIODIC BOUNDARY CONDITIONS !!!!! ————————————%
%
%
% ———————————————————— SLOPE LIMITER DEFINITION ————————————————————————%
%
```

```
% Reference element length for computing the slope and sharp edges
%                              limiters (0.1 m, 5.0 in by default)
REF_ELEM_LENGTH= 0.1
%
% Coefficient for the limiter
LIMITER_COEFF= 0.1
%
% Freeze the value of the limiter after a number of iterations
LIMITER_ITER= 999999
%
%
% ------------------------ FLOW NUMERICAL METHOD DEFINITION --------------------------%
%
% Convective numerical method (JST, LAX-FRIEDRICH, CUSP, ROE, AUSM, HLLC,
%                              TURKEL_PREC, MSW)
CONV_NUM_METHOD_FLOW= ROE
%
% Spatial numerical order integration (1ST_ORDER, 2ND_ORDER, 2ND_ORDER_LIMITER)
SPATIAL_ORDER_FLOW= 2ND_ORDER_LIMITER
%
% Slope limiter (VENKATAKRISHNAN, VAN_ALBADA)
SLOPE_LIMITER_FLOW= VENKATAKRISHNAN
%
% 1st, 2nd and 4th order artificial dissipation coefficients
AD_COEFF_FLOW= ( 0.15, 0.5, 0.02 )
%
% Time discretization (RUNGE-KUTTA_EXPLICIT, EULER_IMPLICIT, EULER_EXPLICIT)
TIME_DISCRE_FLOW= EULER_IMPLICIT
%
% Relaxation coefficient
RELAXATION_FACTOR_FLOW= 0.95
%
ENTROPY_FIX_COEFF= 0.001
%
%
% ------------------------ TURBULENT NUMERICAL METHOD DEFINITION --------------------%
%
% Convective numerical method (SCALAR_UPWIND)
CONV_NUM_METHOD_TURB= SCALAR_UPWIND
%
% Spatial numerical order integration (1ST_ORDER, 2ND_ORDER, 2ND_ORDER_LIMITER)
%
SPATIAL_ORDER_TURB= 1ST_ORDER
%
% Slope limiter (VENKATAKRISHNAN, MINMOD)
SLOPE_LIMITER_TURB= VENKATAKRISHNAN
%
% Time discretization (EULER_IMPLICIT)
TIME_DISCRE_TURB= EULER_IMPLICIT
%
% Reduction factor of the CFL coefficient in the turbulence problem
CFL_REDUCTION_TURB= 0.5
%
% Relaxation coefficient
RELAXATION_FACTOR_TURB= 0.95
%
```

```
%
% -------------------------- CONVERGENCE PARAMETERS ---------------------------%
%
% Number of total iterations
EXT_ITER= 10001
% Convergence criteria (CAUCHY, RESIDUAL)
%
CONV_CRITERIA= RESIDUAL
%
RESIDUAL_FUNC_FLOW= RHO_ENERGY
%
% Residual reduction (order of magnitude with respect to the initial value)
RESIDUAL_REDUCTION= 6
%
% Min value of the residual (log10 of the residual)
RESIDUAL_MINVAL= -16
%
% Start convergence criteria at iteration number
STARTCONV_ITER= 10
%
% Number of elements to apply the criteria
CAUCHY_ELEMS= 100
%
% Epsilon to control the series convergence
CAUCHY_EPS= 1E-6
%
% Function to apply the criteria (LIFT, DRAG, NEARFIELD_PRESS, SENS_GEOMETRY,
%                                 SENS_MACH, DELTA_LIFT, DELTA_DRAG)
CAUCHY_FUNC_FLOW= DRAG
%
%
% ------------------------- INPUT/OUTPUT INFORMATION --------------------------%
%
% Mesh input file
MESH_FILENAME= ../jones_turbine_notip3.su2
%
% Mesh input file format (SU2, CGNS, NETCDF_ASCII)
MESH_FORMAT= SU2
%
% Mesh output file
MESH_OUT_FILENAME= meshout.su2
%
% Restart flow input file
SOLUTION_FLOW_FILENAME= restart_flow.dat
%
% Restart adjoint input file
SOLUTION_ADJ_FILENAME= solution_adj.dat
%
% Output file format (PARAVIEW, TECPLOT, STL)
OUTPUT_FORMAT= TECPLOT
%
% Output file convergence history (w/o extension)
CONV_FILENAME= history
%
% Output file restart flow
RESTART_FLOW_FILENAME= restart_flow.dat
```

```
%
% Output file restart adjoint
RESTART_ADJ_FILENAME= restart_adj.dat
%
% Output file flow (w/o extension) variables
VOLUME_FLOW_FILENAME= flow
%
% Output file adjoint (w/o extension) variables
VOLUME_ADJ_FILENAME= adjoint
%
% Output objective function gradient (using continuous adjoint)
GRAD_OBJFUNC_FILENAME= of_grad.dat
%
% Output file surface flow coefficient (w/o extension)
SURFACE_FLOW_FILENAME= surface_flow
%
% Output file surface adjoint coefficient (w/o extension)
SURFACE_ADJ_FILENAME= surface_adjoint
%
% Writing solution file frequency
WRT_SOL_FREQ= 1000
%
% Writing convergence history frequency
WRT_CON_FREQ= 1
WRT_RESIDUALS= NO
```

Bibliography

[1] N. Anand. *Supersonic Turbine Design using Method of Characteristics*. PhD thesis, Delft University of Technology, 2016. URL https://repository.tudelft.nl/islandora/object/uuid{%}3Af6b5bcbb-e894-4361-a1d0-48e6429e3ae4?collection=education.

[2] Juan S. Bahamonde, Matteo Pini, and Piero Colonna. Active Subspaces for the Preliminary Fluid Dynamic Design of Unconventional Turbomachinery. *Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016)*, (June):8572–8586, 2016. doi: 10.7712/100016.2433.7806. URL http://www.eccomasproceedia.org/conferences/eccomas-congresses/eccomas-congress-2016/2433.

[3] Sebastian Bahamonde, Matteo Pini, Carlo De Servi, Antonio Rubino, and Piero Colonna. Method for the Preliminary Fluid Dynamic Design of High-Temperature Mini-Organic Rankine Cycle Turbines. *Journal of Engineering for Gas Turbines and Power*, 139(8):082606, mar 2017. ISSN 0742-4795. doi: 10.1115/1.4035841. URL http://gasturbinespower.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4035841.

[4] M. Burigana. Radial Turbines. 2016.

[5] Huijuan Chen, D. Yogi Goswami, and Elias K. Stefanakos. A review of thermodynamic cycles and working fluids for the conversion of low-grade heat. *Renewable and Sustainable Energy Reviews*, 14(9):3059–3067, 2010. ISSN 13640321. doi: 10.1016/j.rser.2010.07.006. URL http://www.sciencedirect.com/science/article/pii/S1364032110001863.

[6] Piero Colonna, Emiliano Casati, Carsten Trapp, Tiemo Mathijssen, Jaakko Larjola, Teemu Turunen-Saaresti, and Antti Uusitalo. Organic Rankine Cycle Power Systems: From the Concept to Current Technology, Applications, and an Outlook to the Future. *Journal of Engineering for Gas Turbines and Power*, 137(10):100801, oct 2015. ISSN 0742-4795. doi: 10.1115/1.4029884. URL http://gasturbinespower.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4029884.

[7] W N Dawes. THE PRACTICAL APPLICATION OF SOLUTION-ADAPTION TO THE NUMERICAL SIMULATION OF COMPLEX TURBOMACHINERY PROBLEMS. *Proo. Aerospace Sci*, 29:221–269, 1992.

[8] Shahram Derakhshan, Bijan Mohammadi, and Ahmad Nourbakhsh. The comparison of incomplete sensitivities and Genetic algorithms applications in 3D radial turbomachinery blade optimization. *Computers & Fluids*, 39(10):2022–2029, 2010. ISSN 00457930. doi: 10.1016/j.compfluid.2010.07.003.

[9] E. Dimas and D. Briassoulis. 3D geometric modelling based on NURBS: a review. *Advances in Engineering Software*, 30(9):741–751, 1999. ISSN 09659978. doi: 10.1016/S0965-9978(98)00110-0.

[10] Dries Haeseldonckx and William D 'haeseleer. The environmental impact of decentralised generation in an overall system context. *Renewable and Sustainable Energy Reviews*, 12:437–454, 2008. doi: 10.1016/j.rser.2006.07.004. URL http://ac.els-cdn.com/S1364032106001006/1-s2.0-S1364032106001006-main.pdf?{_}tid=887b47f8-747c-11e7-ac29-00000aacb360{&}acdnat=1501346567{_}0502119701c56be457d662786e846c76.

[11] A. C. Jones. Design and Test of a Small, High Pressure Ratio Radial Turbine. *Journal of Turbomachinery*, 118(2):362, 1996. ISSN 0889504X. doi: 10.1115/1.2836651. URL http://turbomachinery.asmedigitalcollection.asme.org/article.aspx?articleid=1464968.

[12] Seok Hun Kang. Design and experimental study of ORC (organic Rankine cycle) and radial turbine using R245fa working fluid. *Energy*, 41(1):514–524, 2012. ISSN 03605442. doi: 10.1016/j.energy.2012.02.035. URL http://www.sciencedirect.com/science/article/pii/S0360544212001302.

[13] Georgia N. Koini, Sotirios S. Sarakinos, and Ioannis K. Nikolos. A software tool for parametric design of turbomachinery blades. *Advances in Engineering Software*, 40(1):41–51, 2009. ISSN 09659978. doi: 10.1016/j.advengsoft.2008.03.008.

[14] R C W De Koning. *Development of a Parametric 3D Turbomachinery Blade Modeler*. PhD thesis, Delft University of Technology, 2015. URL `http://repository.tudelft.nl/islandora/object/uuid{%}3A9bbcf030-af4b-42c8-a7e5-2157bde13706?collection=education`.

[15] T. Korakianitis, M. A. Rezaienia, I. A. Hamakhan, and A. P. S. Wheeler. Two- and Three-Dimensional Prescribed Surface Curvature Distribution Blade Design (CIRCLE) Method for the Design of High Efficiency Turbines, Compressors, and Isolated Airfoils. *Journal of Turbomachinery*, 135(4):041002, jun 2013. ISSN 0889-504X. doi: 10.1115/1.4007443. URL `http://turbomachinery.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4007443`.

[16] Yan Li and Xiao-dong Ren. Investigation of the organic Rankine cycle (ORC) system and the radial-inflow turbine design. *Applied Thermal Engineering*, 96:547–554, 2016. ISSN 13594311. doi: 10.1016/j.applthermaleng.2015.12.009.

[17] V. D. Liseikin. *Grid generation methods*. ISBN 3319578464. URL `https://books.google.nl/books?id=R9MnDwAAQBAJ{&}dq=h-o+grid+topology`.

[18] Reuben Moore. Thermodynamic cycle integrated ORC turbine design , optimization and.

[19] Lasse Mueller, Zuheyr Alsalihi, and Tom Verstraete. Multidisciplinary Optimization of a Turbocharger Radial Turbine. *Journal of Turbomachinery*, 135(2):021022, nov 2012. ISSN 0889-504X. doi: 10.1115/1.4007507. URL `http://turbomachinery.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4007507`.

[20] Nykamp DQ. Cylindrical coordinates - Math Insight. URL `http://mathinsight.org/cylindrical{_}coordinates`.

[21] Les Piegl and Wayne Tiller. The NURBS Book, 1996. ISSN 00104485.

[22] Matteo Pini, Giacomo Persico, Emiliano Casati, and Vincenzo Dossena. Preliminary Design of a Centrifugal Turbine for Organic Rankine Cycle Applications. *Journal of Engineering for Gas Turbines and Power*, 135(4):042312, mar 2013. ISSN 0742-4795. doi: 10.1115/1.4023122. URL `http://gasturbinespower.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4023122`.

[23] Kiyarash Rahbar, Saad Mahmoud, and Raya K Al-Dadah. Mean-line modeling and CFD analysis of a miniature radial turbine for distributed power generation systems. doi: 10.1093/ijlct/ctu028.

[24] Kiyarash Rahbar, Saad Mahmoud, Raya K. Al-Dadah, and Nima Moazami. Modelling and optimization of organic Rankine cycle based on a small-scale radial inflow turbine. *Energy Conversion and Management*, 91:186–198, 2015. ISSN 01968904. doi: 10.1016/j.enconman.2014.12.003.

[25] Salome. SALOME Mesh User's Guide: About Hypotheses. URL `http://docs.salome-platform.org/7/gui/SMESH/about{_}hypo{_}page.html`.

[26] Emilie Sauret. Open Design of High Pressure Ratio Radial-Inflow Turbine for Academic Validation. In *Volume 7: Fluids and Heat Transfer, Parts A, B, C, and D*, page 3183. ASME, nov 2012. ISBN 978-0-7918-4523-3. doi: 10.1115/IMECE2012-88315. URL `http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?doi=10.1115/IMECE2012-88315`.

[27] Emilie Sauret and Yuantong Gu. Three-dimensional off-design numerical analysis of an organic Rankine cycle radial-inflow turbine. *Applied Energy*, 135:202–211, 2014. ISSN 03062619. doi: 10.1016/j.apenergy.2014.08.076. URL `http://www.sciencedirect.com/science/article/pii/S0306261914008873`.

[28] Chris Sideroff. Multi-Block Structured Meshing and Pre-Processing for OpenFOAM Turbomachinery Analysis. URL `http://www.personal.psu.edu/dab143/OFW6/Training/sideroff{_}slides.pdf`.

[29] R. E. Walraevens and N. A. Cumpsty. Leading Edge Separation Bubbles on Turbomachine Blades. *Journal of Turbomachinery*, 117(1):115, 1995. ISSN 0889504X. doi: 10.1115/1.2835626. URL `http://turbomachinery.asmedigitalcollection.asme.org/article.aspx?articleid=1464671`.

[30] E.H. Wang, H.G. Zhang, B.Y. Fan, M.G. Ouyang, Y. Zhao, and Q.H. Mu. Study of working fluid selection of organic Rankine cycle (ORC) for engine waste heat recovery. *Energy*, 36(5):3406–3418, 2011. ISSN 03605442. doi: 10.1016/j.energy.2011.03.041. URL `http://www.sciencedirect.com/science/article/pii/S036054421100209X`.

[31] Nigel Weatherill, Bharat Soni, and Joe Thompson, editors. *Handbook of Grid Generation*. CRC Press, dec 1998. ISBN 978-0-8493-2687-5. doi: 10.1201/9781420050349. URL `http://www.crcnetbase.com/doi/book/10.1201/9781420050349`.

[32] Sergiy Yershov and Viktor Yakovlev. MESH RESOLUTION EFFECT ON 3D RANS TURBOMACHINERY FLOW SIMULATIONS.