3D-simulation of multi-stage turbomachinery by means of a non-reflecting mixing plane interface

V. Francés Mollá

Technische Universiteit Delft

TU Delft

3D-simulation of multi-stage turbomachinery by means of a non-reflecting mixing plane interface

by

V. Francés Mollá

to obtain the degree of Master of Science in Aerospace Engineering

at the Delft University of Technology,

to be defended publicly on Friday June 8, 2017 at 15:00.

Thesis registration number:	134#17#MT#FPP	
Student number:	4412095	
Project duration:	December 1, 2015 – June 8, 2017	
Supervised by:	Dr. Ir. M. Pini,	TU Delft
	Ir. S. Vitale,	TU Delft
Thesis committee:	Dr. Ir. M. Pini,	TU Delft, supervisor
	Dr. Ir. P. Colona,	TU Delft
	Dr. Ir. A.C. Viré,	TU Delft

TU Delft

Summary

Turbomachines have experienced a fast development over the last years, and are present in a wide range of devices involved in the generation of electricity and the transportation sector. Current challenges in both sectors have driven quite a number of developments regarding the increase of efficiency and use of new types of devices.

Those changes have also affected turbomachinery design, and new innovative approaches are being used in order to solve current problems. As the design of those machines is sometimes done without lot of prior knowledge, new design techniques are required.

Design optimization is usually involving a wide range of disciplines. When it comes to turbomachinery applications, this becomes more challenging as there are lot of parameters involved in the design. The heterogeneous range of applications imposes challenges from the designer point of view as different architectures may need to be considered, different working fluids, different number of stages, etc.

In this Master Thesis, the problem of multi-stage turbomachinery analysis will be undertaken. Simulation of multi-stage turbomachinery presents a number of limitations when it comes both to modelling of the problem and the use of CFD codes, among which the main one is the modelling of the stator-rotor interface.

To overcome this problem, a number of solutions are proposed, of which some are presented in the present report. The Mixing-Plane interface is one of the most used, especially when design optimization is considered. The Mixing-Plane interface creates an artificial interface between turbomachinery blade rows which forces the match of the average of physical quantities at both upstream and downstream sides of the interface.

However, since in general specialised turbomachinery solvers are multi-block structured, current approaches based on the Mixing-Plane interface are not adapted to unstructured edge based solvers. This imposes limitations in the mesh types that can be used for simulating turbomachinery, and add some design constrains. In this Master Thesis the existing methods are analysed and extended to the edge based unstructured solver SU2.

After implementing the proposed approach, a multi-stage turbine for which experimental data is available is simulated. By comparing the results with the experiments and simulations with another CFD solver, the implemented method is validated.

Finally, the main conclusions are presented and some recommendations for future work and possible next steps are drawn.

Acknowledgments

This thesis project has been an enriching experience for me, and I feel very grateful for all the people that have made it possible. During this time, a lot of people have contributed to helping me in the finalisation of this Master Thesis.

First of all, I would like to thank Matteo Pini for giving me the opportunity to work on this interesting project and letting me know more about this exciting topic. And of course, I could not forget to mention Salvo Vitale. Your constant help and dedication have surely been of great help to the finalisation of this project, but more importantly, have helped me to learn a lot and develop in areas I would have not expect.

I would also like to thank my family for all their support and good energy, especially to my parents and my sister for helping me getting through the day even though they were a couple of kilometres away. Thank you to my grandmas Lola and Isabel, for all the love they sent any opportunity they had. And special mention to my grandpa Vicente, your strong support until the day you left has been very important to me.

And finally I would like to thank all the friends I have made in Delft and outside that have contributed to make this an unforgettable experience: thank you Julia and Sofia, for all those endless working days in different projects; thank you Blue Stripes for all the great times we had together; and thank you besties for all the memories.

V. Francés Mollá
Delft, June 2017

Contents

List of Tables

List of Figures

Nomenclature

Acronyms

BC	Boundary Conditions
CFD	Computational Fluid Dynamics
SU2	Stanford University Unstructured
NICFD	Non-Ideal Compressible Fluid Dynamics
ORC	Organic Rankine Cycle
CO2	Carbon dioxide

Symbols

A	Area
a	Speed of sound
c	Characteristic variable
F	Flux
h	Enthalpy
$\mathbf{J_f}$	Jacobian matrix
$\dot{m}_f$	Mass flow
p	Pressure
P	Pitch
R	Radius
s	Entropy
v	Velocity
x, y, z	Cartesian coordinates
α	Tangential flow angle $v_\mathrm{t}/v_\mathrm{n}$
γ	Radial flow angle $v_\mathrm{r}/v_\mathrm{n}$
θ	Circumferential coordinate
ζ	Span-wise coordinate
ϕ	Generic variable
π	Primitive variable
ρ	Density

Subindices

n	Relative to normal direction
t	Relative to tangential direction
ζ	Relative to span-wise direction
tot	Total conditions
in,BC	Relative to inlet Boundary Conditions

1 Introduction

1.1. Background

Turbomachinery have nowadays a wide range of applications, extensively used in many industries including aerospace, automotive, ventilation, power generation, chemical manufacturing or petroleum exploration [1].

Due to their wide range of applications, a lot of efforts are made to improve their design. Improving the designs by increasing aerodynamic efficiency would have a rather large effect in turbomachine performance, and therefore having a positive impact by reducing energy consumption, and both economic and environmental cost.

Turbomachinery can differ both in their geometry features and layout, or the use of different working fluids. It is common to classify turbomachinery based in their flow direction into axial, radial and mixed flow turbomachines. This differentiation will be relevant on the present work, as the work needs to be generalised for the different turbines.

For some of these applications multi-stage configurations are used. The main advantage of such configuration is that efficiency that can be reached is much higher than what we could reach with a single stage at the expense of increasing design complexity.

Design of multi-stage turbomachinery requires accurate prediction of the fluid flow around rotors and stators, as well as at the interfaces. This imposes a need of accurate and sophisticated numerical models that can capture real flow physics. Computational Fluid Dynamics (CFD) solvers based on RANS equations give a reliable solution for flow simulations [2]. However, applying such techniques to turbomachinery problems usually presents a set of challenges that need specific solutions.

The focus of this work will be on solving one of the limitations of these simulations when they are applied to multi-stage turbomachinery problems. If a full multi-stage turbomachinery problem needs to be tackled, an interface between the different elements will be needed. In this project, various solutions to this problem are analysed.

Full unsteady CFD simulations are currently very expensive computationally and become impractical specially when a design optimization problem is set up. Steady-state simulations become a good compromise between computational cost and accuracy; hence, the so called mixing plane approach was selected.

The work presented in this report is based on the approach proposed by Saxer and Giles [3]. However, the approach has some limitations allowing only for the simulation of structured meshes. In this Master Thesis the approach is extended to the unstructured edge solver SU2, allowing for the simulation of different geometries

and turbine architectures, increasing the freedom while preparing the mesh and analysing the geometry.

In the spirit of making this approach usable for any general turbomachine problem, such as Organic Rankine Cycle (ORC) turbomachines [4], [5] or those based on the use of super critical CO2 as working fluid [6], the equations are generalised so that Non-Ideal Compressible Fluid Dynamics (NICFD) simulations can be performed.

1.2. Project goal and motivation

Current developments in the turbomachinery industry are imposing challenges in the design process. Some of the current design problems are very complex and advanced design methods that involve optimization are required. Many variables are involved in the turbomachine design problem, among the presence of more than a blade row can be highlighted. Therefore an interface that solves the problem of rotor-stator interaction is required.

SU2 has demonstrated already its power for fluid shape optimization, mainly in external aerodynamics problems. Turbomachinery simulations have some additional difficulties that make this not yet possible. The implementation of the three dimensional Mixing-Plane interface in SU2 will allow us to perform multi-stage simulations in SU2, and finally couple it with the built-in adjoint solver, which will allow for shape optimization of full turbomachines geometry.

SU2 is an open-source unstructured edge flow-solver, written in C++, under development at various universities, including Stanford University and TUDelft. Since it is an open-source code, the reader of this report can find the code of the implementation available on Github.

The Mixing-Plane interface has been proven to be a reliable mechanism to allow the simulation of multi-stage turbomachinery. However it has some limitations that reduce quite a lot the possibilities regarding the modelling and simulation of this turbomachinery. With the current implementations of the Mixing-Plane, unstructured grids cannot be simulated. Current methods also have some limitations when non uniformly spaced meshes are considered at the interfacing boundaries.

From the ideas presented, the main research question is

Is it possible to effectively extend the Mixing-Plane approach to an edge base unstructured solver such as SU2?

This main research question, lead us to establish some sub-questions that will need to be answered before the main questions can be solved.

The Mixing-Plane approach that will be used in this project, will need the implementation of Non Reflective Boundary Conditions [3]. *Can we extend the current approach to allow for non-ideal fluid dynamics simulation?*

Can we adapt the Mixing-Plane theory so that computations can be done in parallel increasing the computation performance and reducing the required time?

Main objective to be accomplished by the end of this Master Thesis can be summarised in the following sentence:

Implementation and validation of a three dimensional Mixing-Plane interface that will allow for turbomachine simulation and design optimization in a open-source CFD solver.

From the theory point of view, the main novelty of the project would be the extension of the three dimension Mixing-Plane approach to allow the use of unstructured mesh (such as those used by SU2). By using the real gas flows capabilities already implemented in SU2, the extension of the Mixing-Plane formulation to allow the use of any kind of fluid. It will also be able to handle any kind of existing turbomachinery typology.

From the practical point of view, the successful implementation and validation of the Mixing-Plane interface will allow for better simulations of turbomachinery and the development of more sophisticated approaches in multi-stage turbomachinery design.

Due to time and resources restrictions, the following research project will focus mainly on the implementation and validation of the three dimensional Mixing-Plane interface. The shape optimization problem can be performed as a research problem itself since it involves further work and needs the implementation to be validated in order for the shape optimization to make sense. The need of a validated implementation to further continue research in this area encourages the proper realisation of this project.

1.3. Report outline

This Thesis is structured as follows:

In **Chapter 2** a literature review about the state of the art of turbomachines simulations is presented. The review concentrates on the study of steady-state simulation of multi-stage turbomachinery and its limitations.

Chapter 3 shows the methodology used in the implementation of the Non Reflective Boundary Conditions (NRBC) and the Mixing-Plane (MP) interface. Current method is extended for its implementation to a vertex centred unstructured solver.

Chapter 4 presents the results of the validation of a test case using the method described before by comparing the results with experimental data and other CFD simulations published in literature. The preparation of that test case and the conditions used for such a simulation are also explained.

Finally, in **Chapter 5** relevant conclusions are obtained and some recommendations regarding future work are presented.

2

Literature Review

The field of turbomachinery simulations and applications is one that is very wide and covers a lot of different disciplines, from aerodynamics to heat transfer and material design problems. This literature review will focus on the aerodynamic design challenges. Since CFD is nowadays used as the main design tool for turbomachinery applications, its limitations will be studied and a justification of SU2 as selected solver.

As the main focus of this project is to analyse the flow behaviour in a multi-stage turbomachine, the choices done regarding the treatment of the rotor-stator interface are very important.

Different methods to solve the rotor-stator interface are investigated in order to find the more appropriate for its implementation in SU2. Due to the focus on design optimization, the steady-state Mixing-Plane interface is chosen. Various implementation approaches are analysed, and the chosen method is then explained in detail.

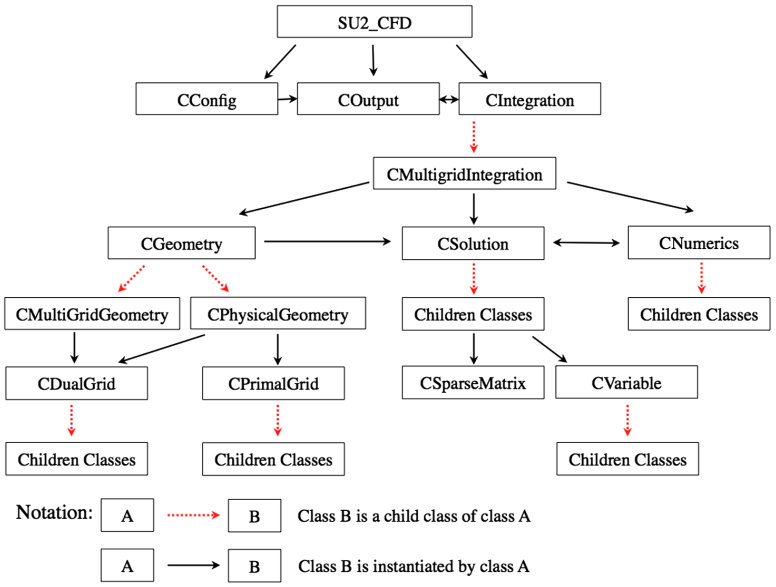
2.1. Relevance of SU2 as CFD solver

SU2 is a computational analysis and design suite that has been developed to solve multi physics analysis and perform optimization tasks using unstructured mesh topologies. Moreover, SU2 is being developed using an open-source strategy and it uses a structure that makes it perfectly suitable for extension of its functionality. SU2 code structure is presented in Figure 2.1. Without going into much details, SU2 code is divided into different classes that are interconnected to each other and its built in a way that is relatively easy to add and recycle new capabilities with a high level of modularity.

Since SU2 is built with this philosophy in mind, problems involving different physic phenomena can be easily modelled and solved. SU2 has demonstrated its capabilities by solving both the flow and adjoint systems of equations to provide a high- fidelity predictive capability and sensitivity information that can be used for optimal shape design using a gradient- based framework, goal-oriented adaptive mesh refinement, or uncertainty quantification [7].

What makes SU2 a powerful suite is its capacity of efficiently solve flow and adjoint equations combined with its surrounding infrastructure for automatic shape design. Its open-source conception has created a global community of research and development teams that continue adding value and finding solutions to new engineering applications, especially in the field of shape design, new numerical methods and CFD simulations.

The convenience of use of SU2 is being already proven by the various members of the research community. As an example, Economon *et al.* [8] performed single- and multi-node optimizations, as well as the use of a scalable linear multi grid solver. Optimizations of a full CFD application for scalable RANS calculations on unstructured meshes have been performed.



[htpb]

Figure 2.1: SU2 code structure [7]

An study showing the potential of adjoint optimization is presented by Albring *et al.* [9]. By using Algorithmic Differentiation the Jacobian Matrix does not need to be calculated, and therefore possible errors and time consuming operations are avoided. This approach allows automatic adaptation to changes and modifications in the flow solver. An application of this optimization has been presented by Zhou *et al.* [10].

If these studied solutions can be applied into turbomachinery design problems, SU2 can become a very powerful tool in the field. Rigorous and cost effective simulation of rotating effects is one of the major problems widely studied. Ramezani *et al.* [11] shows how the implemented multiple reference frames is implemented into SU2 increasing its capabilities regarding the simulation of SU2.

Propulsion and power group of TU Delft is currently working on the turbomachinery module within SU2, which is capable of working with Non-Ideal Compressible Fluid-Dynamics (NICFD), which includes dense vapours, supercritical flows and compressible two-phase flows. These fluids have a very different thermodynamic response compared to a perfect gas, being really sensitive to changes near the critical point [6].

Some works already showed the potential of SU2 as a turbomachinery flow simulation tool. For example, mini-ORC (Organic Rankine Cycle) turbines. This kind of turbines have a large interest for renewable power applications. However, the appearance of NICFD greatly complicates turbomachinery design because of the complex fluid dynamics involved in the process. Therefore there is room for increasing efficiency and improving design. However, this cannot be done by other methods and design optimization is of great help in such cases [5].

As part of the development of the turbomachinery module, a rotor-stator interface needs to be implemented to allow for multi-stage turbine simulation. The Mixing-Plane interface is selected, and is required to consider several turbine architectures and using the SU2 capabilities of solving non-ideal fluids by making profit of the modularity of SU2. The inclusion of the Mixing-Plane interface into the SU2 suite has also promising use of the adjoint solver in SU2, which will allow for shape optimization by modification of the simulated geometry.

2.2. Limitations of turbomachinery CFD

Even though CFD plays a big role in design of turbomachinery, limitations of current methods need to be taken into account and care needs to be taken when making assumptions, especially when engineers are relying always more and more on CFD and less on experimental data. As CFD is not a exact science and errors

are involved, Denton [12] presented a detailed study where he highlights the biggest limitations regarding simulations of turbomachinery. Most of the sources of errors in turbomachinery can be found in:

- Numerical errors due to discretization approximations.
- Modelling errors, where the true physics are unknown or are too complex to model, for example, capturing the turbulence characteristics.
- Uncertain boundary conditions, such as inlet total conditions such as pressure or enthalpy profiles.
- Geometry modelling uncertainties in certain regions, such as tip clearances or leading edge curvature.
- Assumption of steady-state flow.

Despite these limitations CFD remains an extremely valuable tool for turbomachinery design but it should be used on a comparative basis and not as only reference when evaluating the performance of a turbine.

One of the main sources of errors in turbomachinery is that coming from the estimation of the loss generated by flow mixing in compressible fluids. In a turbomachine, one can clearly identify different contributors to the mixing losses by differentiating simultaneous linear disturbances, i.e., from unsteady entropy, vorticity, and pressure waves. The results of the analysis have important effects for numerical simulations of turbomachinery flows, especially the mixing loss at the stator/rotor interface in steady simulations are to be taken into account [13] and will be relevant to the presented work.

A multitude of sources contribute to the unsteadiness. The relative motion of neighbouring blade rows, in conjunction with the spatially non-uniform pressure fields locked to loaded blades, leads to an unsteady pressure distribution in both through the so called potential interaction. When highly loaded blades are simulated, is of special importance that this is taken into account. Highly loaded blades influence substantially the flow downstream that cross the rotor-stator interface. When such non uniformities appear, methods that ensure non reflectivity at both inflow and outflow boundaries at the interface need to be used.

When performing steady calculations in turbomachinery, errors will appear as the nature of the flow is inherently unsteady. However, they are much easier to implement and require much less computational power. A common practice in steady simulations is to perform a mixed out average at the outflow boundaries. To provide uniform and steady boundary conditions at the inflow of the next blade row. Nonetheless, it needs to be considered that this approximation has an associated mixing loss that needs to be considered. Other methods are possible, but are not conservative, and hence should not be used in the final solutions.

Fritsch also points out that the mixing loss is due to different factors:

- The circumferential translation of the control volume.
- Increase in entropy due to the sliding interface.
- Placing the boundary further downstream do not affect the mixing loss itself as the contribution to the mixing loss would not make any effect, since the entropy waves do not travel with the sound waves.
- Lastly, the presence of shock waves crossing the interface also has an effect.

Another source of losses as it was pointed out by Denton [12] are those related to secondary flows. Some of the sources identified by Pullan [14] and need to be taken into consideration when setting up simulations. One can highlight:

- The exit flow field of the stator can cause vortices downstream of the rotor.
- Regions of the stator wake with steep gradients of absolute total pressure normal to the axisymmetric stream-surface of the primary flow have been shown to form the vortices at rotor exit.

- In unsteady numerical simulations, the formation of these vortices was found to be extremely sensitive to the predicted total pressure gradients at stator exit.

- A steady-state rotor calculation using mixing out the stator exit flow to pitch-wise uniformity, as is done in mixing-plane calculations can generate 10 percent less loss than the unsteady simulation. Therefore one should be careful when performing steady-state calculations in turbomachines.

2.3. Rotor–stator steady–state interface methods

When analysing the problem of rotor-stator interface, different approaches may be followed. In a first approximation, we could divide the treatment of this interfaces in steady or unsteady. In this section a description of both methods is presented. Table 2.1 summarises the main alternatives studied here.

Table 2.1: Rotor–stator interface methods comparison

	Mixing-Plane	Frozen Rotor	Unsteady simulations
Computational cost	Low	Low	Very high
Overall performance prediction	Good	Position dependent	Very good
Local flow changes	No	Yes	Yes
Rotor stator position dependent	No	Yes	No
Adjusted geometry	No	No	Yes

2.3.1. Unsteady simulation

The flow in a turbomachine is naturally unsteady, hence when performing a CFD simulation is natural to think that a unsteady simulation might be required. Tucker [15] makes a good analysis of the different possible approaches that one can chose when performing such a simulation. Figure 2.2 shows a summary of the different techniques ha can be used to perform a turbomachinery simulation by using different turbulent models and comparing unsteady and steady methods. These methods present differences in the accuracy level, especially regarding the modelling of the turbulence and the treatment of unsteady and steady simulations.

As compared with steady methods, unsteady simulations provide a better prediction of the flow behaviour and are able to predict some phenomena that cannot be recognised in steady simulations.

Unsteady sliding mesh stator rotor simulations are the more complex type of calculation. However, these type of simulations require the pitch of the consecutive blades to match. Since that is usually not the case to avoid resonance problems, the phase lag method firstly introduced by Giles [16] should be used.

Other kind of simulations include hybrid steady-unsteady stator-rotor simulations, as the one proposed by Montomoli *et al.* [17]. Such simulations give the designer the option to simulate an unsteady stage embedded in a steady-state simulation. This way mesh size can be reduced and computational cost as well while still capturing unsteady features of the natural flow. Other advanced methods for simulating turbomachinery include time-inclinded simulations [16] or Adamszyk stresses [18].

Given the scope of this project, computational cost is highly limited. Therefore, performing such calculations would become impractical. Hence, steady-state methods need to be used.

2.3.2. Steady state methods

In steady state simulations, two main methods are used to study the rotor-stator interface:

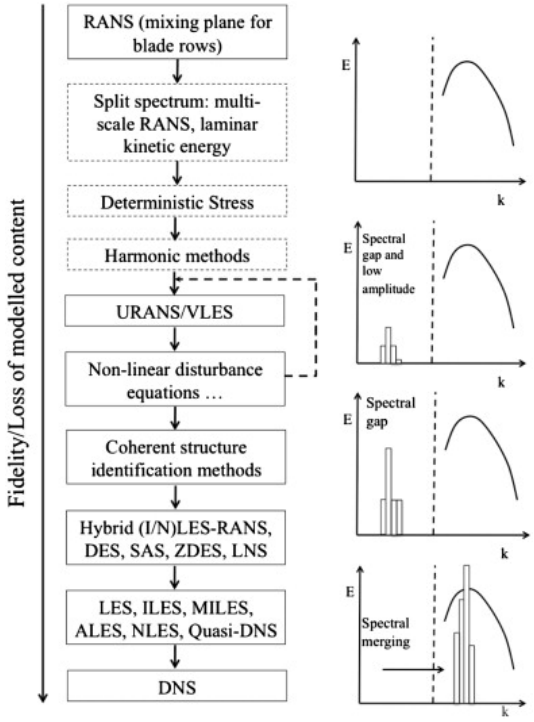


Figure 2.2: Mixing–Plane fidelity [15].

Frozen rotor A frozen rotor simulation consist on considering the position of the rotor with respect to the stator fixed but introducing the rotating effects in those mesh elements that are rotating. This results in a steady flow solution without transient effects included. Rotating waves, secondary flows, leading edge pressure increases and other effects naturally transitory will be predicted for a given position and will stay static. This makes frozen rotor powerful to understand the effects of some phenomena for a given position, but make the results very dependent on the chosen relative location of the stator and rotor blade rows.

Nonetheless, for certain situations a frozen rotor simulation might be good enough for obtaining a good approximation of the overall performance of the turbine [19]. Although transient inertia or acceleration effects cannot be predicted and the dependence on the position is high.

Main advantage of the frozen rotor approach are:

- Accurate simulation of the flow through the turbomachine for a given relative position.
- Wake effects can be predicted
- Influence of upstream flow in blades downstream can be properly predicted.

Mixing-Plane The first method using what we refer to as the Mixing-Plane interface was firstly introduced by Denton [20].

The Mixing-Plane interface is a solution that allow us to perform steady state calculations for turbomachinery that remove the dependency of the results on the relative position between the rotor and the stator, providing a good result for the performance parameters of the turbomachine.

By using the Mixing-Plane, a good estimation of the overall performance of the turbomachine can be obtained. However, due to the averaging at the interface, some transient phenomena cannot be predicted and especial attention needs to be put, especially in off-design conditions.

There are several implementations of the Mixing-Plane interface, and they will be presented in the next section.

2.4. Mixing-Plane interface

The Mixing-Plane in its more general formulation (Figure 2.3) can be described as follows:

1. The quantities at the outflow boundary of the upstream blade row are transferred into a structured interface.
2. The quantities are averaged in the interface in the pitch-wise direction.
3. Once the quantities are uniform in the pitch-wise direction, the average quantities are transferred to the downstream blade row inflow.
4. According to the boundary conditions settings, the average quantities are imposed as boundary conditions.

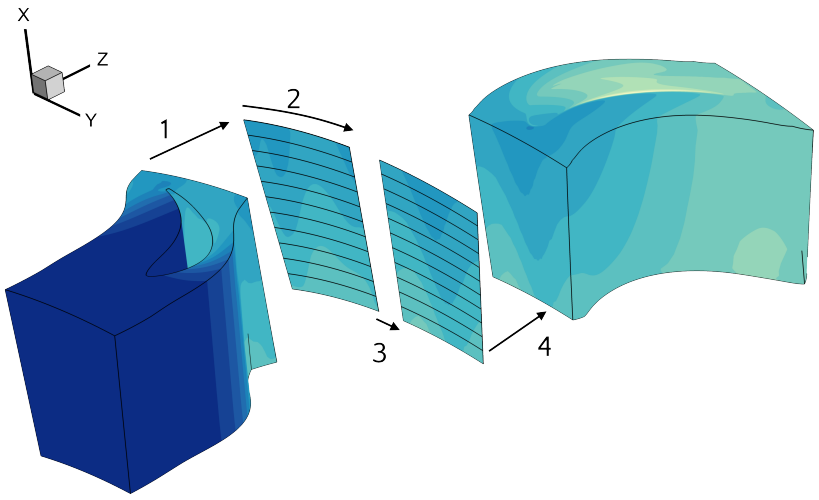


Figure 2.3: Mixing-Plane interface formulation.

This general formulation presents differences depending on the averaging method that is finally chosen (Section 2.4.1), the variables that are transferred or the chosen method to impose the boundary conditions.

In the original method proposed by Denton [20], flow quantities are averaged by performing a mixing process between the exit flow upstream of the interface and the flow crossing the Mixing-Plane. This method allows for the conservation of mass, energy and momentum.

The main drawback of this method is that it imposes circumferentially uniform flow at the downstream inflow boundary. This means that the predicted flow would not represent accurately the characteristics of the real flow, especially when the inflow boundary condition is too close to the leading edge of the blade as it does not allow the flow to adapt circumferentially.

Some years later, Denton [21] proposed a modification of his original method that allows for non uniform flow at both sides of the interface. The boundary condition imposed in this case is less restrictive, imposing only the values of the average of the variables at the inflow boundary to match those at the Mixing-Plane, allowing for non-uniformities at the inflow of the blade downstream. By relaxing this boundary condition, even though there might still be some problems, improves the prediction of the flow around the leading and trailing edges of the different blades.

However, when blade loading is too high or if there are wakes crossing the boundary faces at the Mixing-Plane, the proposed solution by Denton is not enough and the predicted flow is not representative of reality. To solve this problem, Saxer and Giles [3] implemented the Non Reflective Boundary Conditions adapted to turbomachinery applications and included in this implementation the Mixing-Plane interface. This approach is based on the characteristic analysis of the Navier-Stokes equations and is presented in a later section.

Since the Mixing-Plane is a solution that is used widely in the industry to perform steady state calculations, several solutions have been proposed in order to solve some of the problems the original solution have. Table 2.2 shows a summary of the studied methods.

Table 2.2: Mixing-Plane methods comparison

	Pros	Cons
Denton [21]	Simple implementation Conservative Widely used and validated	Reverse flow problems Bad prediction of local changes near inflow boundary
Saxer and Giles [3]	Implement NRBC Improves flow prediction near boundaries	
Wang [22]	Solves reverse flow problem Introduces virtual control volume concept	Adds complexity
Gisbert and Corral [23]	Solves reverse flow problem Only difference in formulation of equations	Not enough validation data available
Du and Ning [24]	Better prediction of shock waves Flow prediction near boundaries highly improved	Very complex implementation Need of creation of virtual structured mesh inside the solver
Hanimann *et al.* [25]	Implicit method Good behaviour near design conditions	Problems with reverse or chocked flow

A new form of the Giles equations is proposed by Gisbert and Corral [23] based on the two-dimensional non-reflecting boundary conditions. As opposed to the commonly used approach, the proposed method determines the differences that result in the conservation of mass, momentum and energy after the boundary condition is enforced, ensuring conservation at any instant during the iterative process. The reverse flow within the Mixing-Plane boundary is naturally treated, but both inlet and outlet boundary conditions fail when the Mixing-Plane normal velocity tends to zero, giving rise to sharp variations of the fluid variables that must be properly limited to prevent convergence problems. This method add complexity to the currently used methods, but solves the problem of reverse flow by still conserving the properties of non-reflectiveness and conservation.

A novel implicit implementation of the Mixing-Plane is proposed by Hanimann *et al.* [25]. The implicit implementation of the Mixing-Plane imposes changes in the flow variables to be equal. It can improve the performance of the code especially during the start up, reducing the problems caused by back flow or choked conditions. In this method the non-reflectiveness is removed, as by making the method implicit, most of the non uniformities are already dumped by using the implicit Mixing-Plane. The implicit Mixing-Plane proved the expected superior behaviour compared to the explicit characteristics-based approach. The implicit approach leads to the better prediction of flow structures such as, e.g., the point of separation on a downstream located blade.

Wang [22] includes the virtual control volume concept into the mixing plane interface. This method creates a virtual volume of null physical size at the interface where the flow is mixed out. The created virtual control volume (Figure 2.4) is assumed to present no flux in the radial or circumferential direction, ensuring conservation of mass, momentum and energy across the interface. This is achieved by using the semi discrete flow equation updating scheme to convert flux differences to conservative flow variable differences then to characteristic variable differences. The main advantage of this method is that it reduces problems related to

reverse flow calculations, by performing the mixing in the virtual control volume.

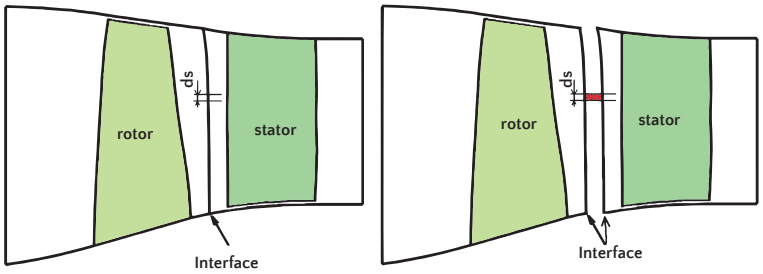


Figure 2.4: Schematic of an artificial interface between a rotor and a stator (left) and the virtual control volume (right) [22].

One more method based on the Mixing-Plane is proposed by Du and Ning [24]. For steady simulation of flow in multi-stage turbomachines, this mixing-plane method solves pitch-wise non-uniformities by performing the mixing out of the flow in an hypothetical extended mixing region. By doing this, artificial reflections can be eliminated, helping in the simulation of highly loaded machines. This method however has the limitation that the extended region needs to be meshed, adding more complexity to the solver, as the interface needs to be created and computational grid is then extended. In our case, since SU2 is an unstructured vertex solver, such a solution would be impractical.

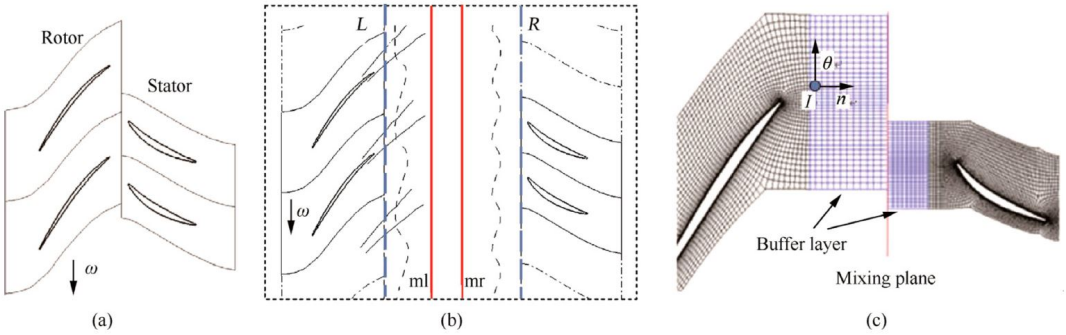


Figure 2.5: Schematic view of pitch-wise mixing model using the extended structured mesh [24].

After analysing some of the methods proposed in literature for solving the rotor-stator interface problems, no clear advantage from any of the presented methods can be observed with respect to the classic and well known approach proposed by Saxer and Giles. Therefore, this is taken as starting point and the proposed solution is extended to the unstructured edge solver SU2.

2.4.1. Averaging process

The Mixing-Plane interface bases it working principle on the averaging process. At the Mixing-Plane, as well as anywhere else in the flow, the laws of conservation must be satisfied. In order to do so, the mixed-out average is used. However, the mixed-out average may have some convergence problems when the results simulated are far from the actual solution. Therefore, the area and mass flow average are defined as they may help to initialise the solution.

Area average

The area average method will be based on the area assigned to each vertex. This is the default method to be used due to its robustness. However, pressure, temperature and other physical variables obtained using this method may not be predicting the momentum of the flow since no law of conservation is involved in the calculation of this average.

$$\bar{\phi} = \frac{\sum_{i=1}^{N_v} \phi_i A_i}{\sum_{i=1}^{N_v} A_i} \tag{2.1}$$

Mass flow average

The mass flow average method uses the mass flow as weighing variable. This method provides a more representative solution than area average, although it may present more convergence problems in case of reverse flow appears during the calculations. It is recommended to initialize the solver with the area averaging, and switch to mass flow average once the phenomena disappears.

$$\bar{\phi} = \frac{\sum_{i=1}^{N_v} \phi_i \dot{m}_{f_i}}{\sum_{i=1}^{N_v} \dot{m}_{f_i}} \tag{2.2}$$

Mixed out average

Because it is based on the flow conservation laws, the mixed-out average is considered a better representation of the flow since it reflects losses associated with non-uniformities in the flow profiles. This averaging method, such as the mass-flow based, presents problems with convergence when intense reverse flow is crossing the Mixing-Plane.

Therefore, best practice is to initiate the solution with area averaging, and switch to mixed-out average after reverse flow problems are exterminated.

The mixed-out procedure assumes that sufficiently far from the blade the flow is uniform, so that the fluxes based upon the average quantities must be equal to the specific integral fluxes at the boundary of interest. Thus, by solving the following non linear system of equations

$$\bar{F}_\rho = \frac{1}{A}\int_A \rho v_\mathrm{n}\,\mathrm{d}A = \bar{\rho}\bar{v}_\mathrm{n} \tag{2.3}$$

$$\bar{F}_\mathrm{n} = \frac{1}{A}\int_A (\rho v_\mathrm{n} + p)\,\mathrm{d}A = \bar{\rho}\bar{v}_\mathrm{n}\bar{v}_\mathrm{n} + \bar{p} \tag{2.4}$$

$$\bar{F}_\mathrm{t} = \frac{1}{A}\int_A \rho v_\mathrm{n} v_\mathrm{t}\,\mathrm{d}A = \bar{\rho}\bar{v}_\mathrm{n}\bar{v}_\mathrm{t} \tag{2.5}$$

$$\bar{F}_\zeta = \frac{1}{A}\int_A \rho v_\mathrm{n} v_\zeta\,\mathrm{d}A = \bar{\rho}\bar{v}_\mathrm{n}\bar{v}_\zeta \tag{2.6}$$

$$\bar{F}_e = \frac{1}{A}\int_A \rho h_\mathrm{tot}\,\mathrm{d}A = \bar{\rho}\bar{v}_\mathrm{n}\bar{h}_\mathrm{tot} \tag{2.7}$$

the mixed-out-averaged primitive vector $(\bar{\rho}, \bar{v}_\mathrm{n}, \bar{v}_\mathrm{t}, \bar{z}_\mathrm{t}, \bar{p})$ can be computed.

By expressing the enthalpy in Equation (2.7) as a function of the average density and pressure

$$\bar{h}_\mathrm{tot} = h(\bar{\rho}, \bar{p}) + \frac{\bar{v}_\mathrm{n}^2 + \bar{v}_\mathrm{t}^2 + \bar{v}_\zeta^2}{2}, \tag{2.8}$$

the energy flux equation can be reformulated as

$$f^i = \bar{F}_e - \bar{\rho}^i \bar{v}_{\mathrm{n}}^i \left(h(\bar{\rho}^i, \bar{p}^i) + \frac{\left(\bar{v}_{\mathrm{n}}^i\right)^2 + \left(\bar{v}_{\mathrm{t}}^i\right)^2 + \left(\bar{v}_{\zeta}^i\right)^2}{2} \right) = 0 \tag{2.9}$$

Equation (2.9) is solved iteratively with a Newton-Raphson method to obtain the average pressure $\bar{p}$ and eventually the average thermodynamic state.

2.5. Quasi-3D Non Reflective Boundary Conditions.

As the main method used in this project is based on the implementation proposed by Saxer and Giles [3] regarding the quasi-3D Non-Reflective Boundary Conditions, in this section they will be explained in the detail, and in Chapter 3, the main changes are pointed out.

In this section the boundary conditions that will be used in the inflow and outflow, as well as the conditions that are imposed in the Mixing-Plane are going to be presented. In turbomachinery applications is very common that the both the outflow and inflow boundary conditions are set really close to the blades, especially when applying this to multi-stage simulations, due to the usually small space existing between rotor and stator rows. As a consequence, far field boundary conditions cannot be used as is not accurate to assume that the flow is undisturbed and uniform. Imposing such boundary condition lead to the appearance of fictitious reflections which compromises the accuracy of the solution. Furthermore, smaller computational domains allow lowering the computational cost, and by imposing non reflecting boundary conditions, mesh size can be significantly reduced.

The first step of the implementation of the NRBC is to decompose the flow into its average component, where the user specified quantities or the Mixing-Plane are imposed, and the fluctuating component, which is obtained by means of a Fourier decomposition.

2.5.1. Characteristic variables

The NRBC analysis is based on the use of the characteristic variables. Those variables are defined from the physical quantities. As shown in the work by Giles [26], the relation between the primitive and characteristic variables is given by:

$$\mathbf{c} = \mathbf{Q} \cdot \boldsymbol{\pi} \tag{2.10}$$

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} -a^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \rho a & 0 & 0 \\ 0 & 0 & 0 & \rho a & 0 \\ 0 & \rho a & 0 & 0 & 1 \\ 0 & -\rho a & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \rho - \bar{\rho} \\ v_{\mathrm{n}} - \bar{v}_{\mathrm{n}} \\ v_{\mathrm{t}} - \bar{v}_{\mathrm{t}} \\ v_{\zeta} - \bar{v}_{\zeta} \\ p - \bar{p} \end{pmatrix} \tag{2.11}$$

By analysing the physical meaning for every characteristic variable it can be concluded that:

- c_1 is related to the linearised perturbation in entropy
- c_2 is the vorticity in the tangential direction

- c_3 is the vorticity in the span wise direction
- c_4 downstream running pressure wave
- c_5 upstream running pressure wave

When calculating the change in the boundaries in every time step, the characteristic variables are defined in terms of perturbation to the average flow at that time step. Is important to note that the boundary conditions are implemented at a point of the calculation where changes at the variables have been distributed already by the solver, but the values at the nodes have not been updated yet. The changes produced at the boundary can be used to define changes at the primitive variables:

$$\boldsymbol{\delta c} = \mathbf{Q} \cdot \boldsymbol{\delta \pi} \tag{2.12}$$

Outgoing characteristics variables should be able to update and correctly calculate by the algorithm already. The changes in the incoming variables however have to be updated using the theory for the non reflecting boundary theory and therefore changes produced by the solver algorithm are discarded in the incoming characteristic variables.

Once the changes in characteristic variables have been calculated, changes in primitive variables π can be computed by using the relation:

$$\boldsymbol{\delta \pi} = \mathbf{Q}^{-1} \cdot \boldsymbol{\delta c} \tag{2.13}$$

$$\begin{pmatrix} \rho - \bar{\rho} \\ v_{\mathrm{n}} - \bar{v}_{\mathrm{n}} \\ v_{\mathrm{t}} - \bar{v}_{\mathrm{t}} \\ v_{\zeta} - \bar{v}_{\zeta} \\ p - \bar{p} \end{pmatrix} = \begin{pmatrix} -\frac{1}{a^2} & 0 & 0 & \frac{1}{2a^2} & \frac{1}{2a^2} \\ 0 & 0 & 0 & \frac{1}{2\rho a} & -\frac{1}{2\rho a} \\ 0 & \frac{1}{\rho a} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\rho a} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} \tag{2.14}$$

From the changes in the primitive variables, the changes in the conservation variables can also be obtained and then the full flow field will be updated. Following sections will describe the different situations where the variables are updated.

As defined by the general non-reflecting boundary conditions theory, at each inflow and outflow boundary it will be a number of incoming and outgoing modes as a result of the Fourier decomposition. While the changes in the average of the outgoing characteristics will be calculated directly by the algorithm, the changes in the average of the incoming characteristics need to be determined to satisfy the physical conditions imposed at the boundaries which are:

- At the inflow, the stagnation enthalpy (or temperature), stagnation pressure and the flow direction.
- At the outflow, typically the static pressure is imposed.

These different conditions depending on the situation analysed are treated in the following sections considering not only both inflow and outflow conditions, but also subsonic and supersonic flow conditions.

2.5.2. Subsonic inflow

Subsonic inflow boundary conditions require a non straightforward implementation of the non reflectivity to avoid neglecting second order effects in the variations of enthalpy and entropy. Therefore, we enforce that entropy and stagnation enthalpy are uniform across the inflow. To analyse properly changes in the incoming variables, the flow variables are decomposed in two elements. On one side, the average values across the boundaries and on the other the harmonic variations.

To accomplish the requirement that entropy, flow angle and stagnation enthalpy have certain values at the inlet, the following residuals will be driven to zero.

$$R_1 = \bar{s} - s_{\text{BC}} \tag{2.15}$$
$$R_2 = \bar{v}_{\text{t}} - \bar{v}_{\text{n}} \tan(\alpha_{\text{in,BC}}) \tag{2.16}$$
$$R_3 = \bar{v}_{\zeta} - \bar{v}_{\text{n}} \tan(\gamma_{\text{in,BC}}) \tag{2.17}$$
$$R_4 = \bar{h} - h_{\text{BC}} \tag{2.18}$$

Entropy and enthalpy will be defined according to its fluid model and therefore we will treat them as a function of other variables and derivatives will not be simplified. $\alpha_{\text{in,BC}}$ and $\gamma_{\text{in,BC}}$ are the angles imposed at the boundary that define the direction of the flow with respect to the tangential and span wise coordinate respectively. The average changes in the incoming characteristic variables are obtained by one step of a Newton-Raphson procedure:

$$\mathbf{R} = -\mathbf{J_c} \cdot \boldsymbol{\delta}\bar{\mathbf{c}} \tag{2.19}$$
$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{pmatrix} = -\frac{\partial(R_1, R_2, R_3, R_4)}{\partial(c_1, c_2, c_3)} \begin{pmatrix} \delta\bar{c}_1 \\ \delta\bar{c}_2 \\ \delta\bar{c}_3 \\ \delta\bar{c}_4 \end{pmatrix} \tag{2.20}$$
$$= -\frac{\partial(R_1, R_2, R_3, R_4)}{\partial(\rho, v_{\text{n}}, v_{\text{t}}, v_{\zeta}, p)} \cdot \frac{\partial(\rho, v_{\text{n}}, v_{\text{t}}, v_{\zeta}, p)}{\partial(c_1, c_2, c_3)} \begin{pmatrix} \delta\bar{c}_1 \\ \delta\bar{c}_2 \\ \delta\bar{c}_3 \\ \delta\bar{c}_4 \end{pmatrix} \tag{2.21}$$

where $\mathbf{J_c}$ can be expressed as

$$\mathbf{J_c} = \begin{pmatrix} \left(\frac{\partial s}{\partial \rho}\right)_p & 0 & 0 & 0 & \left(\frac{\partial s}{\partial p}\right)_\rho \\ 0 & -\tan(\alpha_{\text{in,BC}}) & 1 & 0 & 0 \\ 0 & -\tan(\gamma_{\text{in,BC}}) & 0 & 1 & 0 \\ \left(\frac{\partial h}{\partial \rho}\right)_p & v_{\text{n}} & v_{\text{t}} & v_\zeta & \left(\frac{\partial h}{\partial p}\right)_\rho \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{a^2} & 0 & 0 & \frac{1}{2a^2} \\ 0 & 0 & 0 & \frac{1}{2\rho a} \\ 0 & \frac{1}{\rho a} & 0 & 0 \\ 0 & 0 & \frac{1}{\rho a} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \tag{2.22}$$

$$= \begin{pmatrix} -\frac{1}{a^2}\left(\frac{\partial s}{\partial \rho}\right)_p & 0 & 0 & \frac{1}{2}\left(\left(\frac{\partial s}{\partial p}\right)_\rho + \frac{1}{a^2}\left(\frac{\partial s}{\partial \rho}\right)_p\right) \\ 0 & \frac{1}{\rho a} & 0 & -\frac{1}{2\rho a}\tan(\alpha_{\text{in,BC}}) \\ 0 & 0 & \frac{1}{\rho a} & -\frac{1}{2\rho a}\tan(\gamma_{\text{in,BC}}) \\ -\frac{1}{a^2}\left(\frac{\partial h}{\partial \rho}\right)_p & \frac{1}{\rho a}v_{\text{t}} & \frac{1}{\rho a}v_\zeta & \frac{1}{2}\left(\left(\frac{\partial h}{\partial p}\right)_\rho + \frac{1}{a^2}\left(\frac{\partial h}{\partial \rho}\right)_p + \frac{1}{\rho a}v_{\text{n}}\right) \end{pmatrix} \tag{2.23}$$

Inverting the Jacobian, we can easily obtain the changes in the average characteristic variables. The inversion of the Jacobian matrix is done numerically by using the general inversion formula for 4x4 matrices implemented in the code.

$$\boldsymbol{\delta \bar{c}} = -\mathbf{J_c}^{-1} \cdot \mathbf{R} \tag{2.24}$$

$$\begin{pmatrix} \delta\bar{c}_1 \\ \delta\bar{c}_2 \\ \delta\bar{c}_3 \\ \delta\bar{c}_4 \end{pmatrix} = -\mathbf{J_c}^{-1} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{pmatrix} \tag{2.25}$$

After obtaining the influence in the average component, the changes due to variations across the boundary are analysed . Firstly, the outgoing fifth characteristic is evaluated at each point. To do that the Fourier transformation need to be calculated.

$$\hat{c_{5k}} = \frac{1}{P}\int_0^P c_5 \exp\left(-\frac{i2\pi k y}{P}\right) \mathrm{d}y \tag{2.26}$$

$$\beta = i\,\mathrm{sgn}(k)\sqrt{1-M^2} \tag{2.27}$$

The two characteristic that depend on the outgoing one can be then obtained in the Fourier space:

$$\hat{c}_{2ks} = -\frac{(\beta + M_{\text{t}})}{M_{\text{n}} + 1}\hat{c}_{5k} \tag{2.28}$$

$$\hat{c}_{4ks} = \frac{(\beta + M_{\text{t}})^2}{(M_{\text{n}} + 1)^2}\hat{c}_{5k} \tag{2.29}$$

Performing the inverse Fourier transform, the information in spatial coordinates is extracted:

$$c_{2js} = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} \hat{c_{2ks}} \exp\left(\frac{i2\pi k y_j}{P}\right) \tag{2.30}$$

$$c_{4js} = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} \hat{c_{4ks}} \exp\left(\frac{i2\pi k y_j}{P}\right) \tag{2.31}$$

The change in those two characteristics at every node j can be then obtained as:

$$\delta c_{2js} = c_{2js} - c_{2j} \tag{2.32}$$
$$\delta c_{4js} = c_{4js} - c_{4j} \tag{2.33}$$

For the remaining characteristic variables, the value is just set to zero.

$$\delta c_{1js} = 0 \tag{2.34}$$
$$\delta c_{3js} = 0 \tag{2.35}$$

Once that the local corrections for the non-reflectivity have been applied, those are added to the average global changes, and multiplied by an under relaxation factor. The imposition of an under relaxation factor is required in order to

$$\delta c_{1j} = \sigma(\delta \bar{c}_1 + \delta c_{1js}) \tag{2.36}$$
$$\delta c_{2j} = \sigma(\delta \bar{c}_2 + \delta c_{2js}) \tag{2.37}$$
$$\delta c_{3j} = \sigma(\delta \bar{c}_3 + \delta c_{3js}) \tag{2.38}$$
$$\delta c_{4j} = \sigma(\delta \bar{c}_4 + \delta c_{4js}) \tag{2.39}$$

To complete the changes in the characteristic variables, the changes in the outgoing characteristic are taken from the changes already calculated by the CFD solver algorithm by using the relation between characteristics and primitive variables (Equation (2.11)).

$$\delta c_{5j} = \delta c_{5j,\mathrm{CFD}} \tag{2.40}$$

2.5.3. Supersonic inflow

The treatment of the inflow boundary condition is reasonably equivalent in the case of supersonic flow (yet axially subsonic), only difference being in the definition of β.

$$\beta_{\text{supersonic}} = -\operatorname{sgn}(v)\sqrt{1 - M^2} \tag{2.41}$$

Since β is no longer depending on the Fourier modes, there is no need to perform the Fourier transform. Hence, the ideal steady state values for the second and fourth incoming characteristics are given by

$$c_{2js} = -\frac{(\beta + M_{\mathrm{t}})}{M_{\mathrm{n}} + 1} c_{5j} \tag{2.42}$$
$$c_{4js} = \frac{(\beta + M_{\mathrm{t}})^2}{(M_{\mathrm{n}} + 1)^2} c_{5j} \tag{2.43}$$

The rest of the implementation of the boundary condition is totally equivalent as that of subsonic flow. Only possible change might be to impose the Riemann invariant as a BC instead of that of the flow angle, since the first is uniform along the inflow boundary, as is done in [27].

2.5.4. Subsonic outflow

The subsonic outflow boundary conditions are obtained directly from the NRBC theory. Since in this case the first four characteristics are outgoing, they do not need to be set. Hence, just the fifth characteristic will need to be set.

Change in pressure with respect of variations on the fifth characteristic are given by

$$\frac{\mathrm{d}p}{\mathrm{d}c_5} = \frac{1}{2} \tag{2.44}$$

and so the change in the average fifth characteristic variable is

$$\delta\bar{c_5} = -2(\bar{p} - p_{BC}) \tag{2.45}$$

To obtain the local changes in the fifth characteristic, the Fourier transform is used again to obtain the Fourier modes of the second and fourth characteristic:

$$\hat{c}_{5ks} = \frac{\hat{c}_{2k}(2M_\mathrm{n})}{\beta - M_\mathrm{t}} - \frac{\beta + M_\mathrm{t}}{\beta - M_\mathrm{t}}\hat{c}_{4k} \tag{2.46}$$

where $\hat{c}_{2k}$ and $\hat{c}_{4k}$ are given by

$$\hat{c}_{2k} = \frac{\sum_{j=1}^{N} c_{2j} \exp\left(-\frac{i2\pi k y_j}{P}\right)}{N} \tag{2.47}$$

$$\hat{c}_{4k} = \frac{\sum_{j=1}^{N} c_{2j} \exp\left(-\frac{i2\pi k y_j}{P}\right)}{N} \tag{2.48}$$

Given that complex conjugate pairs will occur,

$$c_{5js} = 2\,\mathrm{Re}\left(\sum_{k=1}^{\frac{N}{2}-1} \hat{c_{5ks}} \exp\left(\frac{i2\pi k y_j}{P}\right)\right) \tag{2.49}$$

The variation on the fifth characteristic for every node is described as

$$\delta c_{5js} = c_{5js} - c_{5j}. \tag{2.50}$$

In order to boost convergence, an under relaxation factor σ is imposed.

$$\delta c_{5j} = \sigma(\delta\bar{c}_5 + \delta c_{5js}). \tag{2.51}$$

The rest of the variables are then obtained directly from the information already provided by the solver algorithm and the relation between characteristics and primitive variables (Equation (2.11)).

$$\delta c_{1j} = \delta c_{1j,\mathrm{CFD}} \tag{2.52}$$

$$\delta c_{2j} = \delta c_{2j,\mathrm{CFD}} \tag{2.53}$$

$$\delta c_{3j} = \delta c_{3j,\mathrm{CFD}} \tag{2.54}$$

$$\delta c_{4j} = \delta c_{4j,\mathrm{CFD}} \tag{2.55}$$

2.5.5. Supersonic outflow

As with the inflow boundary condition, the implementation is fairly equivalent to the case of subsonic outflow (still axially subsonic). The main difference resides in the definition of beta which is again:

$$\beta_{\text{supersonic}} = -\operatorname{sgn}(v)\sqrt{1 - M^2} \tag{2.56}$$

Therefore since there is no dependence on k Fourier mode anymore (as in the supersonic outflow case), the calculation can be simplified and we will need to simply impose:

$$c_{5js} = \frac{c_{2j}(2M_{\text{n}})}{\beta - M_{\text{t}}} - \frac{\beta + M_{\text{t}}}{\beta - M_{\text{t}}} c_{4j} \tag{2.57}$$

The rest of the implementation is equivalent to that of the subsonic outflow case.

3
Method

This chapter presents the implementation of the Non-Reflecting Boundary Conditions (NRBCs) and of the Mixing-Plane interface within the SU2 CFD solver. Since, SU2 is vertex-based unstructured solver, the implementation of the NRBCs and of the Mixing-Plane is generalized for an unstructured solver.

The chosen implementation of the Mixing-Plane interface divides the boundary in bands where the relevant physical quantities are averaged. Depending on the turbine type that is being simulated, the coordinate chosen for performing the span wise division, ζ, will differ. For axial turbomachinery, the radius will be the ordering coordinate, $\zeta = R$, hence equal radius bands will be defined. For radial flow turbomachinery (centripetal and centrifugal), the span coordinate ζ will correspond to the axis of rotation, z (Figure 3.1).

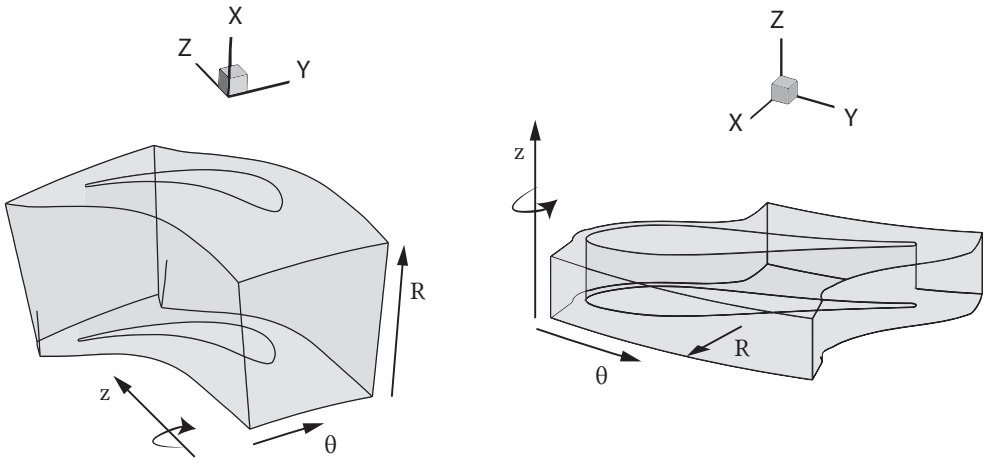


Figure 3.1: Axial vs. Radial Turbomachine Pitch

The original implementation of the Mixing-Plane is built with structured meshes in mind. Therefore all the vertex can be easily assigned to a band if the mesh meets the requirements. This condition may be too restrictive, especially when it comes to new types of turbomachines which use complex geometries or especially close to blades boundary interfaces. To resolve this problem, the vertex ordering algorithm is implemented.

3.1. Vertex ordering algorithm

In order to impose the Quasi-3D NRBCs, the inflow and outflow boundaries of each blade row are divided into span-wise bands. For each band, the vertex are ordered in the pitch-wise directions to compute the Fourier decomposition of the outgoing characteristic (Section 2.5).

The ordering of the vertex is performed in a preprocessing step. A new boundary data structure, called *TurboVertex* is the outcome of this ordering. The new *TurboVertex* data structure, differently from the standard *Vertex* structure of SU2 [28], allow to access the information at the inflow and outflow in a span-wise and pitch-wise ordered manner. The procedure to create the *TurboVertex* structure is divided in the following steps:

1. computation of the span-wise bands;
2. allocation of the vertexes to the bands in a pitch-wise disordered manner;
3. pitch-wise reordering of the vertexes for each band.

3.1.1. Computation of the Span-Wise bands

The number of span-wise bands in which the inflow and outflow boundaries are sub-divided can be computed in three different ways: *Equispaced, User-Specified*, and *Automatic*. With the *Equispaced* option the inflow and outflow are subdivided in a equispaced manner in the number of bands specified by the user. With the *User-Specified* option, the boundaries are discretized into arbitrary bands provided by the user via an input file (Table 3.1). With the *Automatic* option the number of bands are automatic selected using the vertexes on the edge that is the intersection between the inflow (or outflow) surface and one of the periodic faces (Figure 3.2).

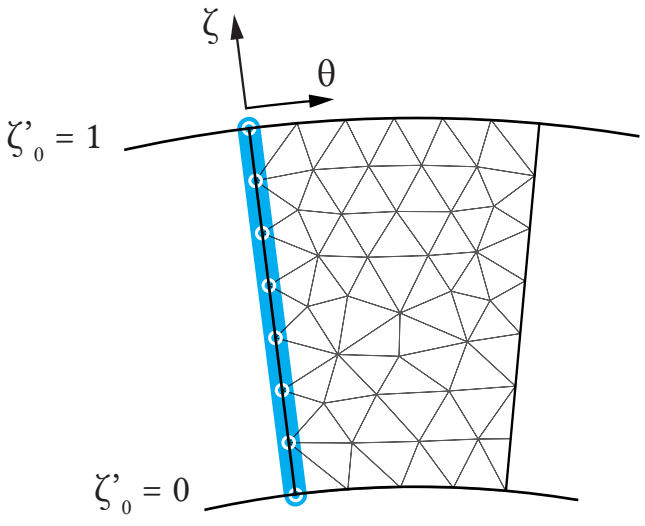


Figure 3.2: Computation of the bands with the automatic .

3.1.2. Allocation of the vertexes to the bands

Once the bands have been computed, each vertex of the inflow and outflow boundary is assigned to the closest band (Figure 3.3).

Table 3.1: Computation of the Span-Wise bands. Example.

	Automatic	Equispaced	User-specified
User input	`AUTOMATIC`	`EQUISPACED`	`USER`
Span-wise division.	Identified directly from periodic boundary	$\zeta_i = i\,\frac{\zeta_{\max} - \zeta_{\min}}{N} + \zeta_{\min}$	List of points provided by user, e.g., $\zeta = (0, 0.3, 0.55, 0.85, 1)$

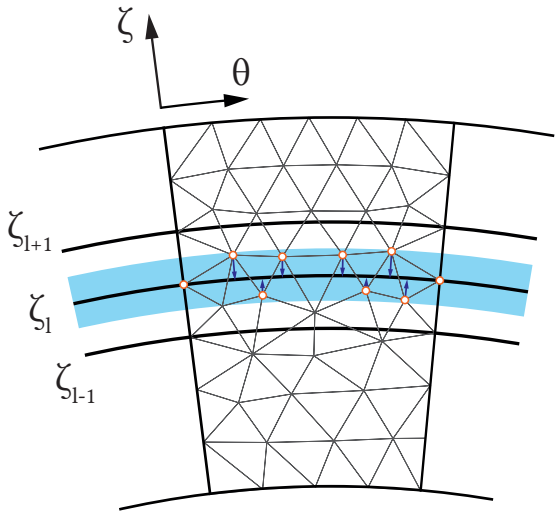


Figure 3.3: Allocation of the vertex to the closest span-wise band.

3.1.3. Pitch-wise reordering of the vertex

After step 2 the vertexes are ordered in the span-wise bands. However, to compute the Spatial Fourier decomposition along the pitch, the vertexes are ordered in the normalized cylindrical coordinate

$$\bar{\theta} = \theta - \theta_{\min} \tag{3.1}$$

where θ is the cylindrical coordinate and $\theta_{\min}$ is the minimum value of θ for the specified band, which coincides with the value at the periodic face.

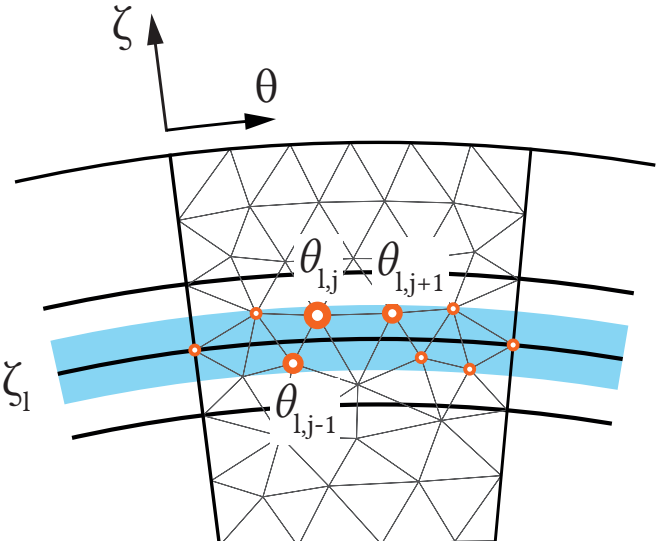


Figure 3.4: Reordering of the span-wise vertexes in the pitch-wise direction.

3.2. Generalization of the NRBCs for unstructured solver

The procedure to compute the Quasi-3D NRBCs follows the same approach explained in Section 2.5. However, a modification of the Fourier Spatial Decomposition is needed in order to take into account of a correct pitch-wise coordinate.

The integral form of the spatial Fourier transform for a generic outgoing characteristic, c_i, is

$$\hat{c}_i(k) = \frac{1}{P}\int_0^P c_i e^{-\frac{2\pi i k}{P}y}\,\mathrm{d}y, \quad \text{with} \quad (-\infty < k < +\infty) \tag{3.2}$$

where P is the pitch length, y is a pitch-wise coordinate and k is a generic frequency.

After doing some algebraic simplification, the discretized version of Equation (3.2) is

$$\hat{c}_i(k) = \frac{1}{P}\sum_{j=0}^{n-1} c_{i,j} e^{-\frac{2\pi i k}{P}y_j}\Delta y_j, \quad \text{with} \quad \left(-\frac{n}{2}+1 < k < \frac{n}{2}-1\right), \tag{3.3}$$

where

$$y_j = \sum_{m=1}^{j}\Delta y_m \quad \text{with} \quad (0 < y_j < P) \tag{3.4}$$

is the cumulative distance among the ordered bands vertexes, and

$$\Delta y_m = y_m - y_{m-1} \tag{3.5}$$

is the distance among two adjacent vertexes. While (Equation (3.3)) can be easily computed on a structure grids (Figure 3.5), it can not be used on a general unstructured grid. For unstructured grid, as can be seen in Figure 3.6, it is not guaranteed that the pitch wise coordinate, Equation (3.4), is $(0 < y_j < P)$.

To overcome this issue, the normalised cylindrical coordinate is used and the vertexes projection on the band is considered for the computation (Figure 3.6).

By using this projection over the band l, the requirement of a structured mesh is no longer needed since it allows the radius to be different to that of the band. Considering that

$$y = R\,\bar{\theta} \tag{3.6}$$

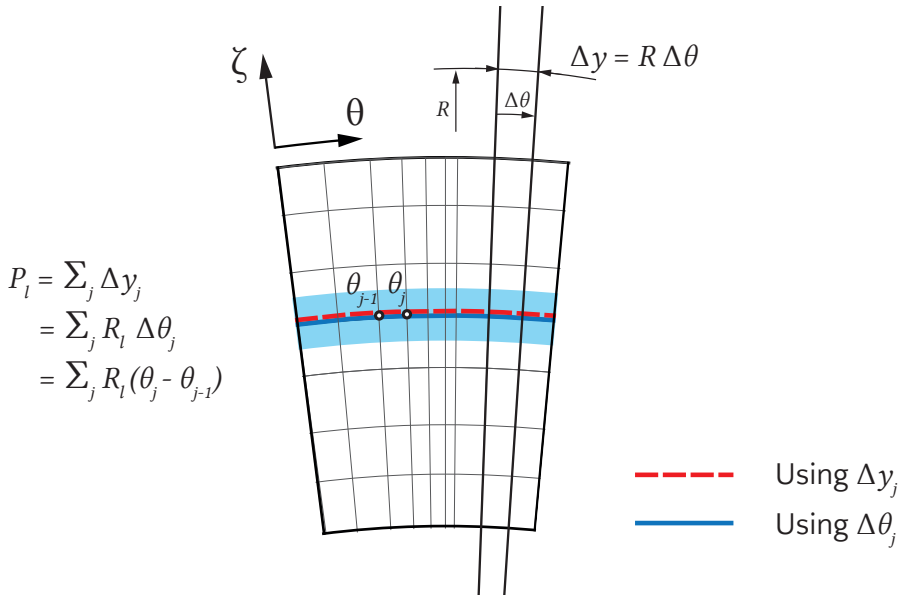


Figure 3.5: Example of quasi 3D NRBC on structured grid.

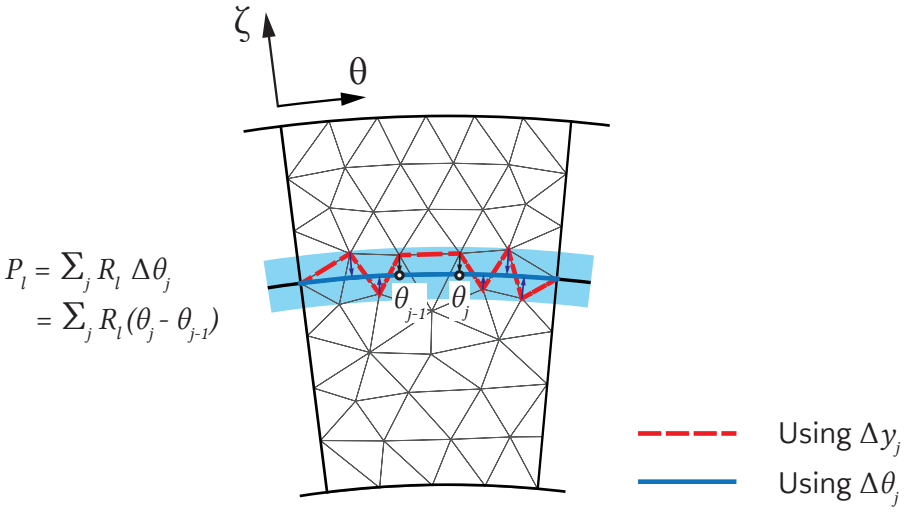


Figure 3.6: Example of Quasi-3D NRBC on unstructured grid.

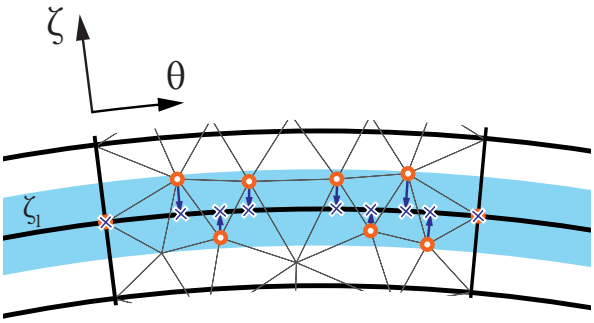


Figure 3.7: Projection over band l of vertex in an unstructured grid.

using this approximation, the equations are now transformed to account for this projection method. The integral form of the spatial Fourier transform for a generic outgoing characteristic, c_i, is

$$\hat{c}_i(k) = \frac{1}{P}\int_0^P c_i e^{-\frac{2\pi i k}{P}R\bar{\theta}}\, R\, \mathrm{d}\bar{\theta}, \quad \text{with} \quad (-\infty < k < +\infty) \tag{3.7}$$

where P is the pitch length, R is the radius, $\bar{\theta}$ is the normalised circumferential coordinate, and k is a generic frequency. With this new formulation the pitch P may be calculated as:

$$P_l = R_l\, \bar{\theta}_{\text{pitch}} \tag{3.8}$$

To impose the Quasi-3D NRBCs, this transformation has to be computed for each band of the inflow and outflow boundaries. The discretized version of (Equation (3.7)) is

$$\hat{c}_i(k) = \frac{1}{\bar{\theta}_{\text{pitch}}}\sum_{j=0}^{n-1} c_{i,j} e^{-\frac{2\pi i k}{\bar{\theta}_{\text{pitch}}}\bar{\theta}_j}\Delta\bar{\theta}_j, \quad \text{with} \quad \left(-\frac{n}{2}+1 < k < \frac{n}{2}-1\right), \tag{3.9}$$

where R_l is the radius at the considered band.

$$\bar{\theta}_j = \sum_{m=1}^{j}\Delta\bar{\theta}_m, \quad \text{with} \quad \left(0 < \bar{\theta}_m < \bar{\theta}_{\text{pitch}}\right) \tag{3.10}$$

is the cumulative distance among the ordered bands vertexes, and

$$\Delta\bar{\theta}_m = \left(\bar{\theta}_m - \bar{\theta}_{m-1}\right) \tag{3.11}$$

is the projected distance among two adjacent vertexes.

This generalised method when applied to structured grids would be equivalent to existing method. When applied to unstructured grids, the proposed method imposes an approximation, but the integration still happens within the blade pitch.

With the proposed method, a Mixing-Plane interface can be applied regardless the mesh type used, allowing for quite some flexibility in the process of creating the mesh.

3.3. Implementation of Mixing-Plane interface

In general, it can be considered that the turbomachine component q is located upstream, and the component r to be downstream. Inflow of q and outflow of r is treated as usual, and here we will be interested in the outflow of q and the inflow of r as the relevant boundaries at the mixing interface, since these are the two to be matched together. Here the only change to what has been presented already will be the difference in the calculation of average changes in the four incoming characteristic at inflow q and the incoming characteristic at the outflow r.

To ensure conservation of mass, momentum and energy, the flux coming out of q should be equal to the flux of those variables passing through r. As flux calculation is used, the average flow quantities must be matching.

$$\rho_q = \rho_r \tag{3.12}$$

$$v_{\mathrm{n}_q} = v_{\mathrm{n}_r} \tag{3.13}$$

$$v_{\mathrm{t}_q} = v_{\mathrm{t}_r} \tag{3.14}$$

$$v_{\zeta_q} = v_{\zeta_r} \tag{3.15}$$

$$p_q = p_r \tag{3.16}$$

Note that the match of this velocities will need to be done in the absolute frame of reference, which is taken into account already in SU2 via implementation of a multiple rotating frame [11].

This condition is to be satisfied at the converged solution, and therefore the differences will need to be driven to zero. As this is done by means of the Non-Reflective Boundary Conditions, non matching conditions in the current run, will be interpreted as a jump in characteristic variables.

$$\begin{pmatrix} \Delta c_1 \\ \Delta c_2 \\ \Delta c_3 \\ \Delta c_4 \\ \Delta c_5 \end{pmatrix} = \begin{pmatrix} -a^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \rho a & 0 & 0 \\ 0 & 0 & 0 & \rho a & 0 \\ 0 & \rho a & 0 & 0 & 1 \\ 0 & -\rho a & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \rho_q - \rho_r \\ v_{\mathrm{n}_q} - v_{\mathrm{n}_r} \\ v_{\mathrm{t}_q} - v_{\mathrm{t}_r} \\ v_{\zeta_q} - v_{\zeta_r} \\ p_q - p_r \end{pmatrix} \tag{3.17}$$

Taking into account the propagation direction of the characteristics for each of the components, the difference between characteristics will need to be driven to zero.

At the outflow q:

$$\delta \bar{c}_5 = \sigma \Delta c_5 \tag{3.18}$$

At the inflow r:

$$\delta \bar{c}_1 = \sigma \Delta c_1 \tag{3.19}$$
$$\delta \bar{c}_2 = \sigma \Delta c_2 \tag{3.20}$$
$$\delta \bar{c}_3 = \sigma \Delta c_3 \tag{3.21}$$
$$\delta \bar{c}_4 = \sigma \Delta c_4 \tag{3.22}$$

To ensure well possessedness and convergence, the under-relaxation factor is used again. After obtaining the changes in the average characteristics, the calculation of the boundary condition is equivalent as for a standard inflow and outflow boundary condition.

3.3.1. Interpolation at the Mixing-Plane interface

In multi-stage turbomachinery simulations there are many cases when the meshes at both sides of the Mixing-Plane interface are non-matching, for example, when simulating cases where tip clearance is present, or in the case that unstructured meshes are used. Moreover, depending on the method selected for creating the bands at the interface, those bands might be non matching at each side of the interface. Hence in order to transfer information from one side of the interface to the other, an interpolation algorithm is required.

In favour of simplicity, a linear interpolation algorithm is implemented to transfer the information that travels from the donor side q to the target side r.

Firstly, the closer span to the donor k for each target is obtained by comparing the span values ζ between the donor and the target side. Once the accordance between both sides is defined, the average values are interpolated. In order to increase efficiency and modularity of the algorithm an interpolation coefficient ξ is defined.

$$\xi_r = \frac{\zeta_r - \zeta_q}{\zeta_{q+1} - \zeta_q} \tag{3.23}$$

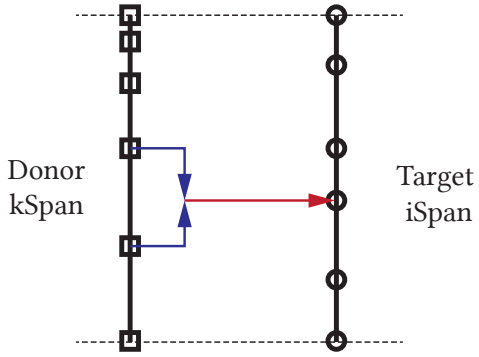


Figure 3.8: Target (qSpan) – donor (rSpan) interface

For each of the averaged variables, $\bar{\phi}$, at the target span ζ_r the information from the donor is obtained from

$$\phi_r = \xi_r \cdot \left(\phi_{q+1} - \phi_q\right) + \phi_q \tag{3.24}$$

The inclusion of this linear interpolation allows the Mixing-Plane interface to match the values of the relevant variables when the interfaces are not matching. This improves the flexibility in the mesh creation process, letting the user to consider tip clearance cases or more adequate meshes for any of the turbomachine elements individually.

4

Results and validation

The Mixing-Plane interface, a stage and a half axial turbine is considered. This axial turbine has been built and tested at RWTH Aachen [29].

In this chapter the results of the simulation of this test cases together with a comparison between the experimental results and numerical simulations available in literature are presented.

4.1. Aachen turbine. 1-1/2 stage axial turbine.

4.1.1. Test case definition

The geometry of the ERCOFTAC Test case 6 [30], commonly referred as the Aachen 1-1/2 stage turbine, or simply as the Aachen turbine, can be easily found in literature and is described by [30].

The Aachen turbine test case layout consist of 2 stators and 1 rotor (Figure 4.1). The casing construction is made of a fixed mounted outer ring and 3 inner rings. The two stators use the Traupel profile. Both stators are identical in profile geometry, number of blades and stagger. For the rotor, a modified VKI-profile has been used. More details in chapter A. Both stators and rotor are made up of untwisted prismatic blades.

Other geometry and details are summarised in Table 4.1

4.1.2. Domain discretization

In order to simulate the test cases presented in this section a multi-block structured mesh will be used for each blade. The meshes are generated with NUMECA Autogrid, and an HO-grid is used. HO grids are constructed starting from the blade geometry. An O-grid is defined surrounding the blade to allow for the H-grid to adapt better. A set of H-grids are defined from the O-grid. H-grids consist of deformed square prisms where the same number of nodes are defined at the parallel sides (Figure 4.2).

The number of points at both the pressure and the suction side need to be matching in order to get a periodic mesh. This is also a requirement if tip clearance wants to be considered in SU2. Since SU2 works with unstructured grids, the multi block grid is then transformed into a single block grid.

Unstructured grids are not used as obtaining high quality unstructured mesh for turbomachine applications is currently quite complicated and mesh quality cannot be guaranteed. Hence, structured meshes are used instead.

Table 4.1: Aachen turbine data

Section	Stator 1	Rotor 1	Stator 2
Aspect ratio	0.887	0.917	0.887
Pitch [mm]	47.6	41.8	47.6
Number of blades	36	41	36
Reynolds number, based on chord and design exit velocity	6.80E+05	4.90E+05	6.80E+05
Rotational speed [rpm]	–	3500	–
Tip diameter [mm]	600	600	600
Chord	62	60	62
Profile	Traupel	VKI-modified	Traupel

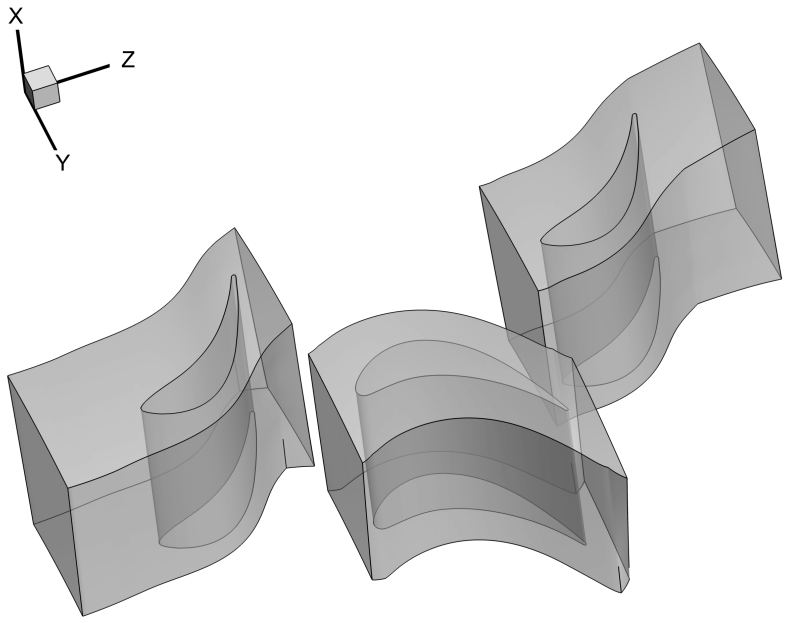
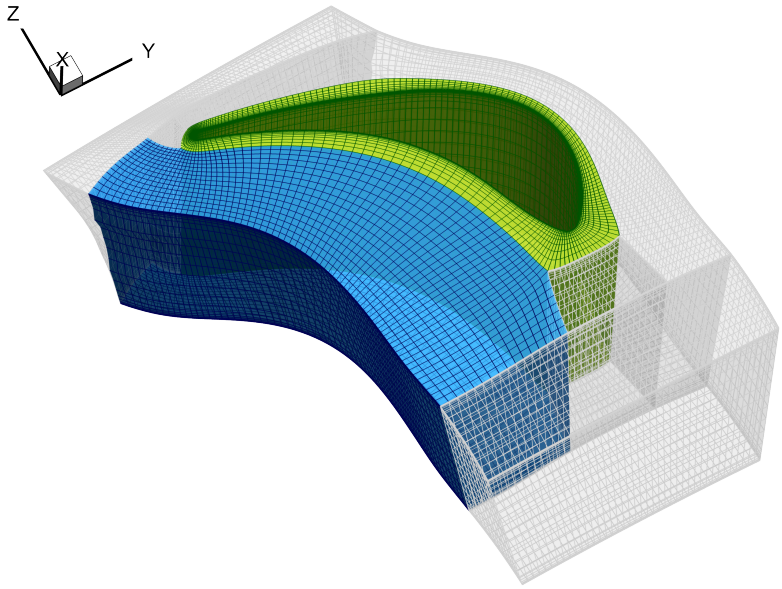


Figure 4.1: Aachen turbine geometry

The generation of a structured grid with tip clearance can generate problems at the area just on top of the blade. The mesh at the tip will be generated from the mesh surrounding the blade profile. Therefore the number of points at both the suction and pressure side of the blade will need to be equal in order to have a matching mesh. The generation of a non matching mesh will require to define an internal artificial interface in the solver that interpolates the solution in that area. Such an interface is not currently defined in SU2 for this applications and therefore this needs to be avoided. In any case, such a mesh is not recommended as it may lead to problems with flow discontinuities or singularities (Figure 4.3).

This requirement will be imposed into the mesh generated without tip clearance so that both solutions can be comparable.

The mesh covers one blade passage per row. Two meshes are considered in order to evaluate the impact of the number of grid points in SU2. The finer mesh will account up to approximately 5 million nodes, while a coarser mesh consists of around 1 million nodes.



[ht!]

Figure 4.2: Block mesh types. O-mesh around the blade, highlighted in green. H-mesh block highlighted in blue.

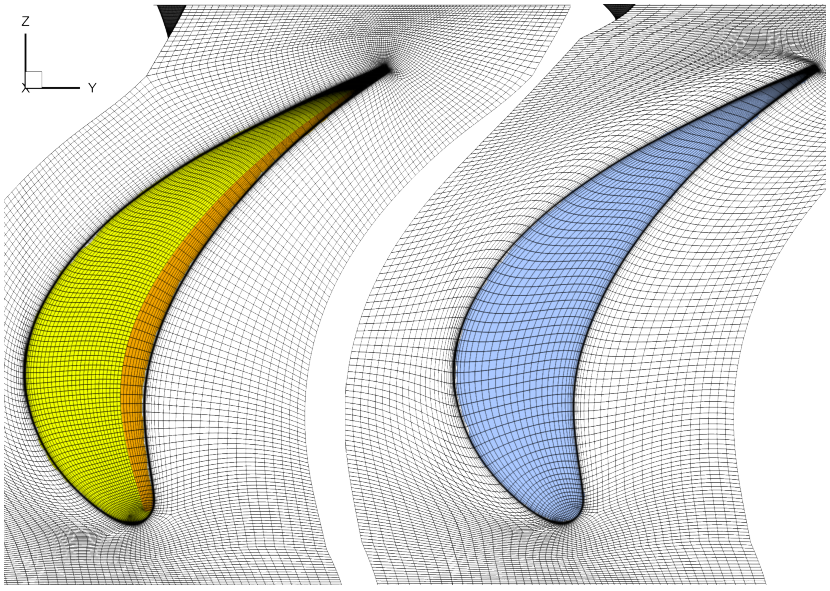


Figure 4.3: Matching and non-matching tip clearance mesh

4.1.3. Boundary conditions

From the reference experimental data, the boundary conditions (BC) are obtained. As explained in the method, the Non-Reflective Boundary Conditions are used both at the inlet, outlet and mixing interfaces. Table 4.2 summarises the imposed boundary conditions when taking the values directly from the experimental results.

The inlet boundary conditions are obtained from the measurements at the inlet plane, and the 1D average values of total pressure and total temperature are imposed at the boundary. The outlet static pressure is selected such that the average pressure at the plane matches the average of the experimental distribution. As in the experiments, only part of the hub wall is moving, while the shroud is remains static. Periodic conditions are enforced along the boundaries upstream and downstream of each blade. Due to the way the mesh is constructed the mesh is matching at the periodic boundaries.

Table 4.2: Boundary conditions. High mass flow studied case

Inlet BC	Total Pressure [Pa]	169458.4
	Total Temperature [K]	305.75
Outlet BC	Average Static Pressure [Pa]	99741.5

Table 4.3: Boundary conditions. Low mass flow studied case

Inlet BC	Total Pressure [Pa]	158497.2
	Total Temperature [K]	309.06
Outlet BC	Average Static Pressure [Pa]	99614.87

In order to validate the results obtained with the implemented method in SU2, those results are compared both with available experimental data and numerical simulations available in literature.

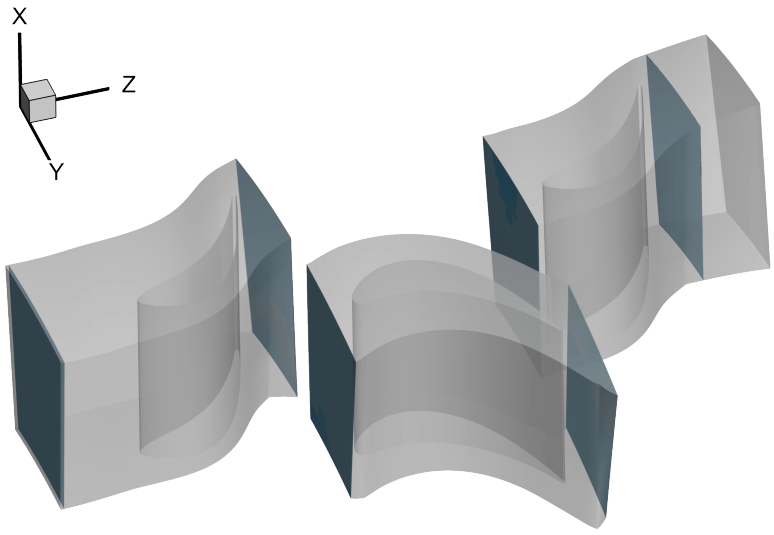


Figure 4.4: Location of measurement planes. 3D view

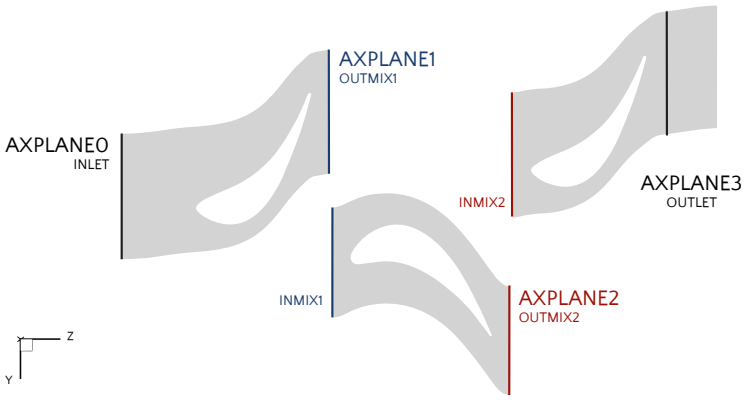


Figure 4.5: Location of measurement planes. 2D view

4.2. Validation

In this section, the results at the boundary conditions are compared, with a predominant focus on the results at the Mixing-Plane interface.

The results are presented at 4 different stations (Figure 4.5) where the experimental results are obtained. First plane will be located at the inflow boundary condition while the other three are located 8.8 mm behind the trailing edge of each of the blades. In order to validate a good performance of the Mixing-Plane, the results are analysed at both the stator and rotor side where a Mixing-Plane boundary condition is imposed.

Results for all comparison are presented as the pitch-wise average of the data for each span position considered. This way of presenting the data was selected for 2 main reason:

- the experimental results are available in the same format,
- and the span-wise mixing-plane behaviour can be better investigated.

As it is common practise in literature, the analysis will focus on two significant variables for the turbine performance. Total pressure, as this is related directly to the overall performance of the turbine; and flow angle, since it helps understanding the flow behaviour inside the turbine and point out the differences between results. Flow angle is measured as shown in Figure 4.6.

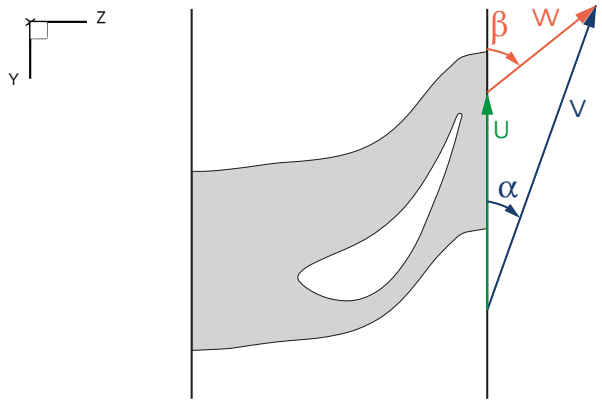


Figure 4.6: Flow angle definition

A detailed analysis of the pitch-wise behaviour is done when analysing the impact of the new method implemented for the non-reflective equations.

4.2.1. Mixing-Plane interface validation

To verify the good agreement at both sides of the Mixing-Plane interface, the 5 million mesh of the Aachen turbine is used, and the five primitive variables used for the Mixing-Plane definition are compared at both sides of the interface.

As it has been explained already, the average flow variables are matched at the interface, while perturbations from the average at each band are allowed by imposing the NRBC.

In this section, both the flow upstream and downstream are analysed and compared by their pitch-wise average.

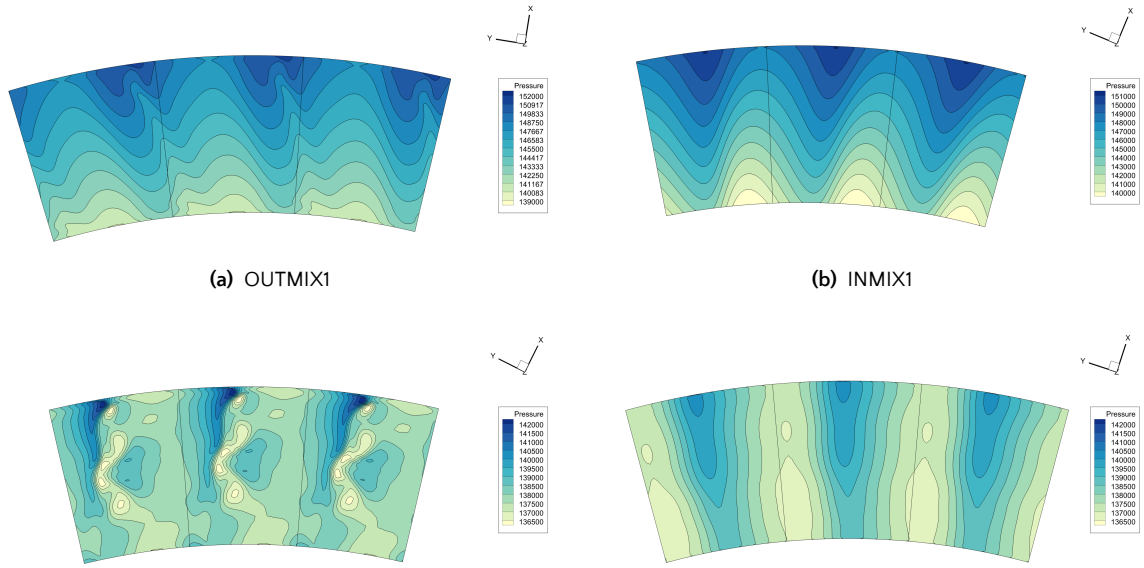
Regarding Static Pressure (Figures 4.7 and 4.8), comparing the pitch-wise averaged values a quite good agreement can be found for both the mixing interfaces. By looking at the differences in the x-y plane, expected

differences are observed since the fluctuations over the average locally at each side of the interface are not required to match.

Checking the behaviour of Static Density (Figures 4.9 and 4.10), similar conclusions are reached.

For the 3 velocity components studied in general a good agreement is found at both sides of the interface. For axial velocity, v_n (Figures 4.11 and 4.12), matching in the average is found as well, and small variations in the plane are found.

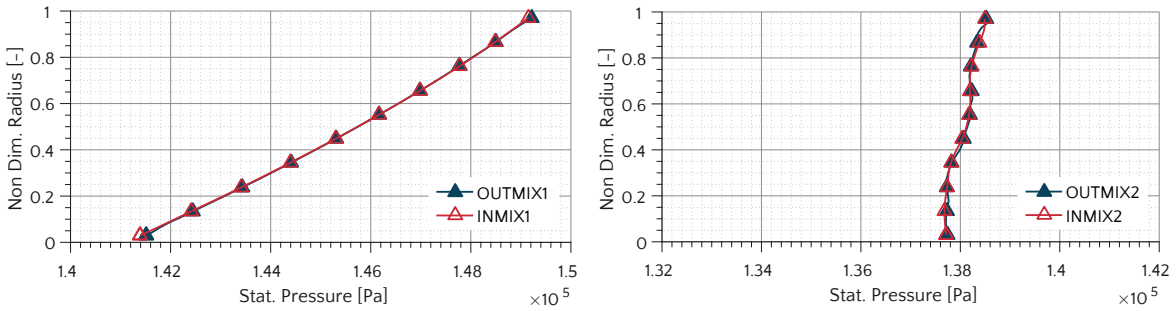
For circumferential velocity, v_t, (Figures 4.13 and 4.14) and for radial velocity, v_ζ, (Figures 4.15 and 4.16) the bigger differences at both sides of the interface are observed since the variables at the outlet presents quite some fluctuations, while at the inflow, for every band the flow remains quite uniform. This is the case since at the outflow boundaries the flow is influenced by the trailing edge wake, while the flow at the inflow do not need big adaptations to predict the steady state solution.



(a) OUTMIX1 **(b)** INMIX1

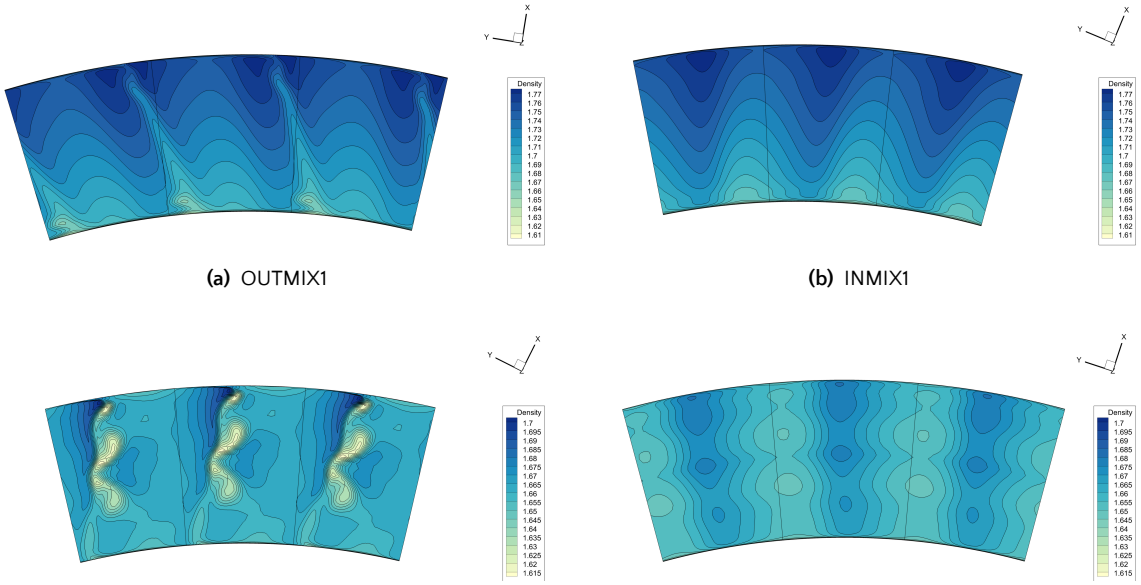
(c) OUTMIX2 **(d)** INMIX2

Figure 4.7: Static Pressure at both sides of the mixing plane. Results from SU2, 5M mesh.



(a) AXPLANE1 **(b)** AXPLANE2

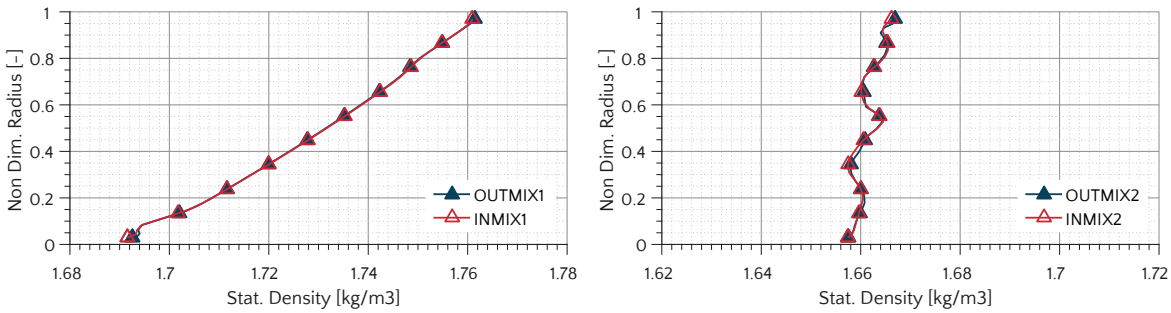
Figure 4.8: Pitch-wise averaged Static Pressure at both sides of the Mixing-Plane. Results from SU2, 5M mesh.



(a) OUTMIX1 **(b)** INMIX1

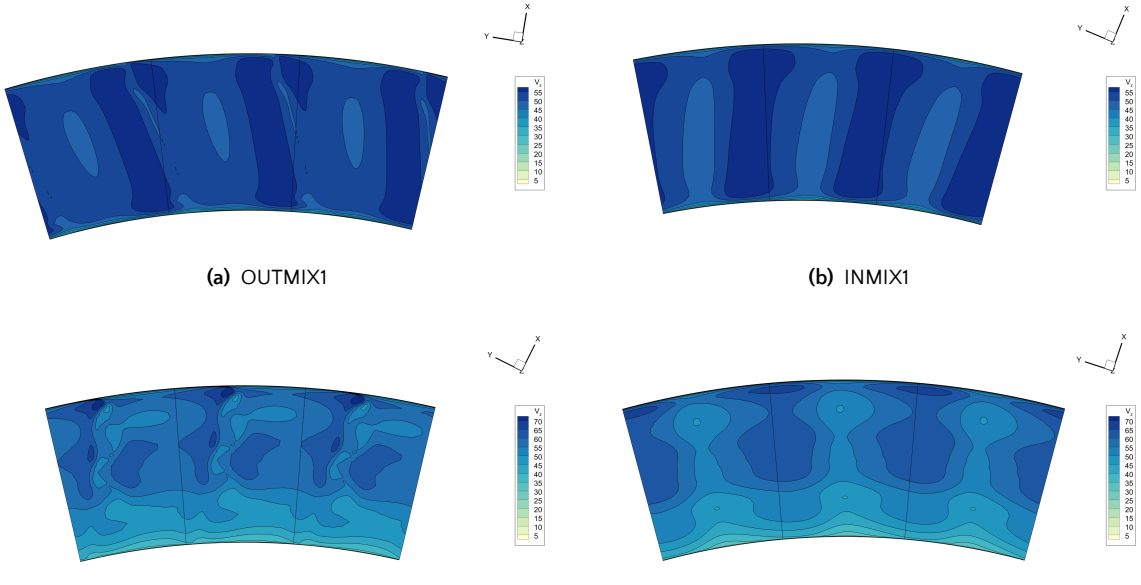
(c) OUTMIX2 **(d)** INMIX2

Figure 4.9: Static Density at both sides of the mixing plane. Results from SU2, 5M mesh.



(a) AXPLANE1 **(b)** AXPLANE2

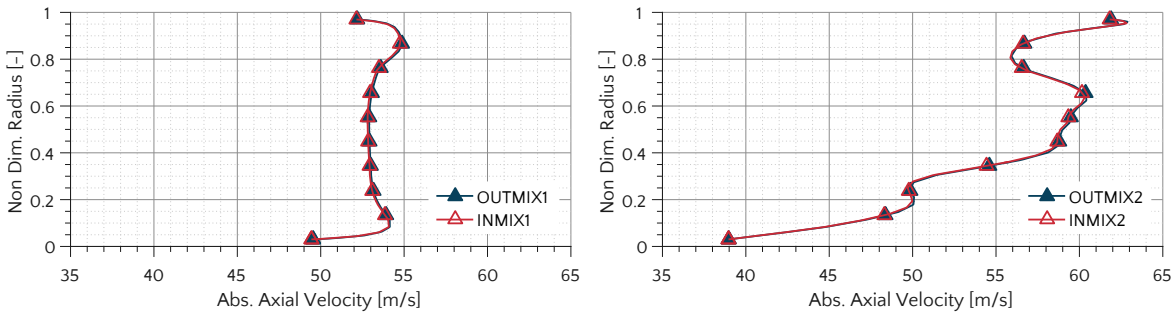
Figure 4.10: Pitch-wise averaged Static Density at both sides of the Mixing-Plane. Results from SU2, 5M mesh.



(a) OUTMIX1 **(b)** INMIX1

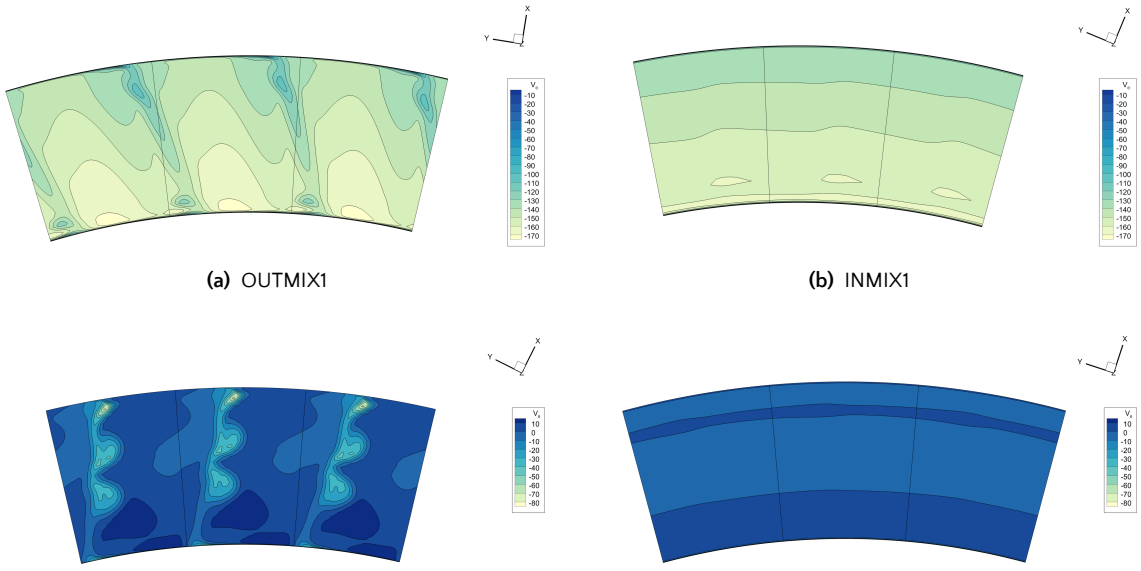
(c) OUTMIX2 **(d)** INMIX2

Figure 4.11: Axial Velocity at both sides of the mixing plane. Results from SU2, 5M mesh.



(a) AXPLANE1 **(b)** AXPLANE2

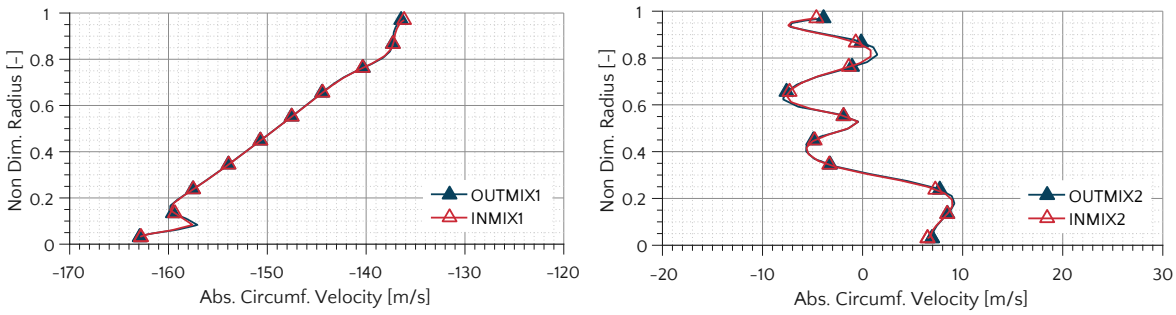
Figure 4.12: Pitch-wise averaged Axial Velocity at both sides of the Mixing-Plane. Results from SU2, 5M mesh.



(a) OUTMIX1 **(b)** INMIX1

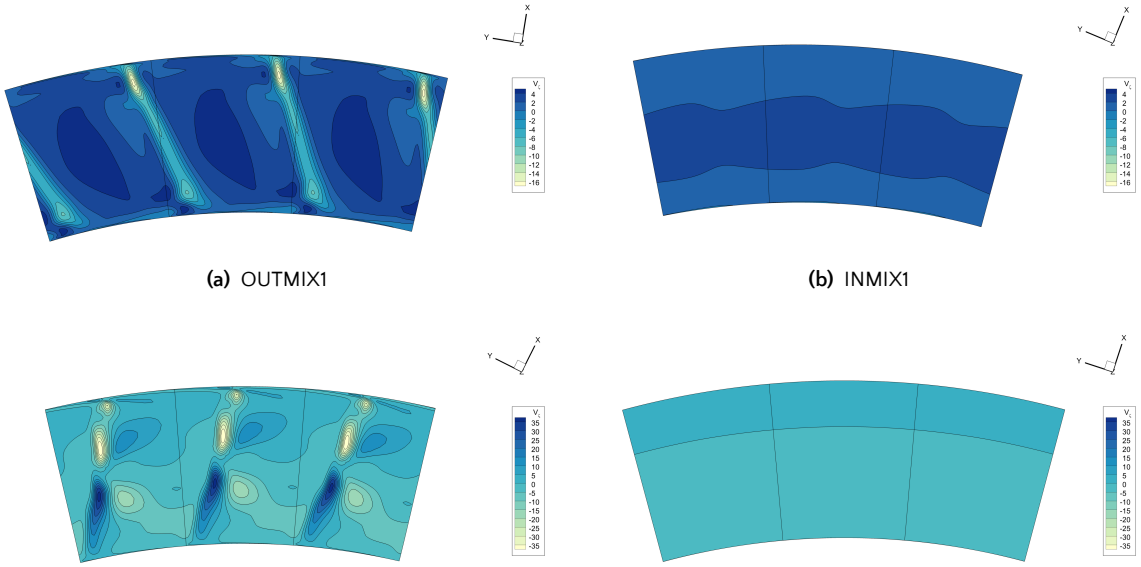
(c) OUTMIX2 **(d)** INMIX2

Figure 4.13: Circumferential Velocity at both sides of the mixing plane. Results from SU2, 5M mesh.



(a) AXPLANE1 **(b)** AXPLANE2

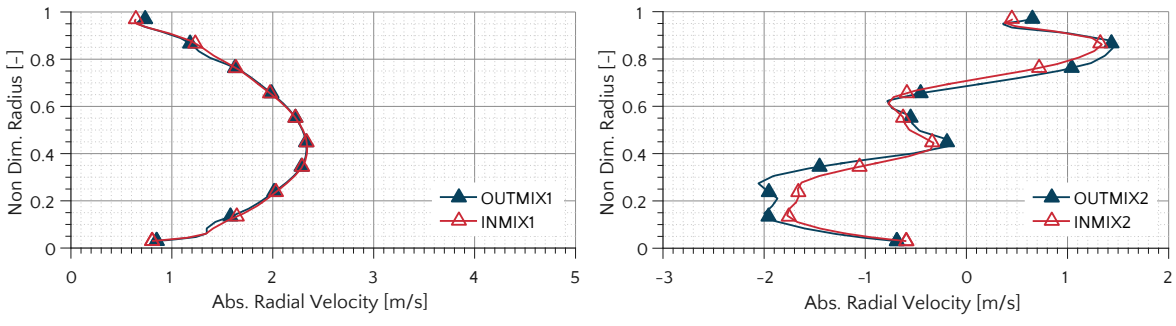
Figure 4.14: Pitch-wise averaged Circumferential Velocity at both sides of the Mixing-Plane. Results from SU2, 5M mesh.



(a) OUTMIX1 **(b)** INMIX1

(c) OUTMIX2 **(d)** INMIX2

Figure 4.15: Radial Velocity at both sides of the mixing plane. Results from SU2, 5M mesh.



(a) AXPLANE1 **(b)** AXPLANE2

Figure 4.16: Pitch-wise averaged Radial Velocity at both sides of the Mixing-Plane. Results from SU2, 5M mesh.

4.2.2. Mesh sensitivity analysis in SU2. 1M elements mesh vs 5M

First of all, a mesh sensitivity analysis is performed for simulations in SU2 with 2 different mesh sizes. Both meshes use the same topology but a different number of elements. The bigger one uses 5 million elements, while the smaller one uses 1 million. A blade to blade comparison is shown in fig. 4.17.

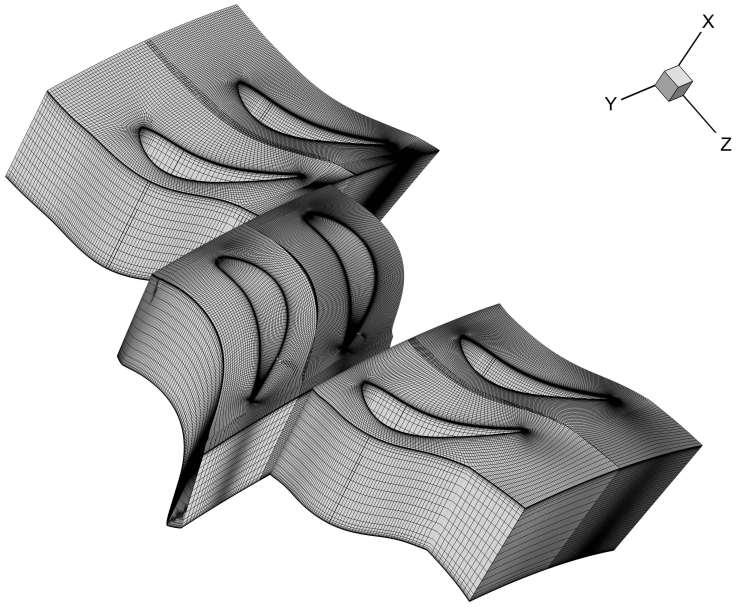
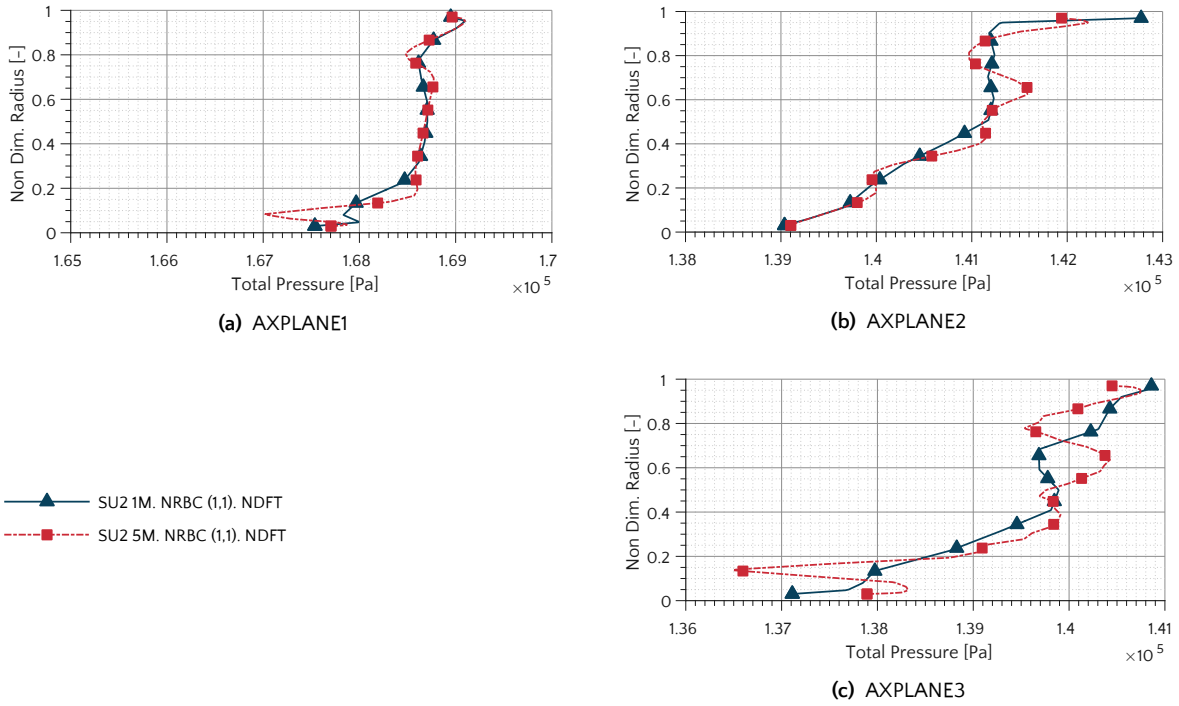


Figure 4.17: Mesh comparison of 1M and 5M mesh.

When one compares the total pressure at the different stations (Figure 4.18), a quite good agreement between the solution for both meshes is found. This good agreement will allow us to make further analysis concentrating only on the 1 million mesh results. Main differences are observed at the outlet plane (Figure 4.18c), where a less uniform distribution of the total pressure is observed when the resolution is higher. In the 1 million mesh this is not observed, as this effects are neglected due to the lower resolution. However, the main trend is followed in both cases.

If attention is put in the flow angle Figure 4.19, a similar conclusion is reached. Main discrepancies here are found after the rotor, as the main flow deflection is shifted from 40% span to 50% span. However, the magnitude of the change remains the same.

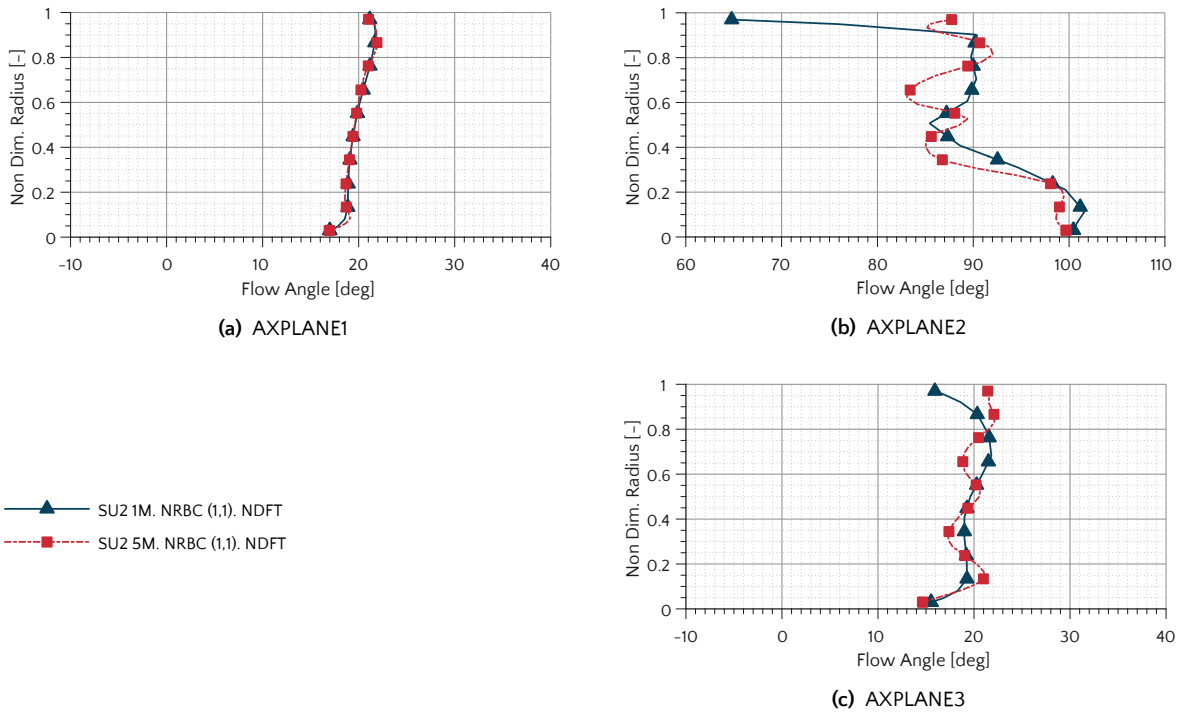


(a) AXPLANE1

(b) AXPLANE2

(c) AXPLANE3

Figure 4.18: Pitch-wise averaged Total Pressure. Results from SU2, 5M mesh and 1M mesh comparison.



(a) AXPLANE1

(b) AXPLANE2

(c) AXPLANE3

Figure 4.19: Pitch-wise averaged Flow Angle. Results from SU2, 5M mesh and 1M mesh comparison.

4.2.3. Non-Reflective Boundary Conditions. Non uniform Fourier decomposition.

As mentioned in section 2.5, if a characteristic analysis of the flow conditions at the boundary is performed, two main components can be differentiated. On one side, the average component and on the other, the fluctuating component. On the simulations analysed so far, only the average component have been analysed.

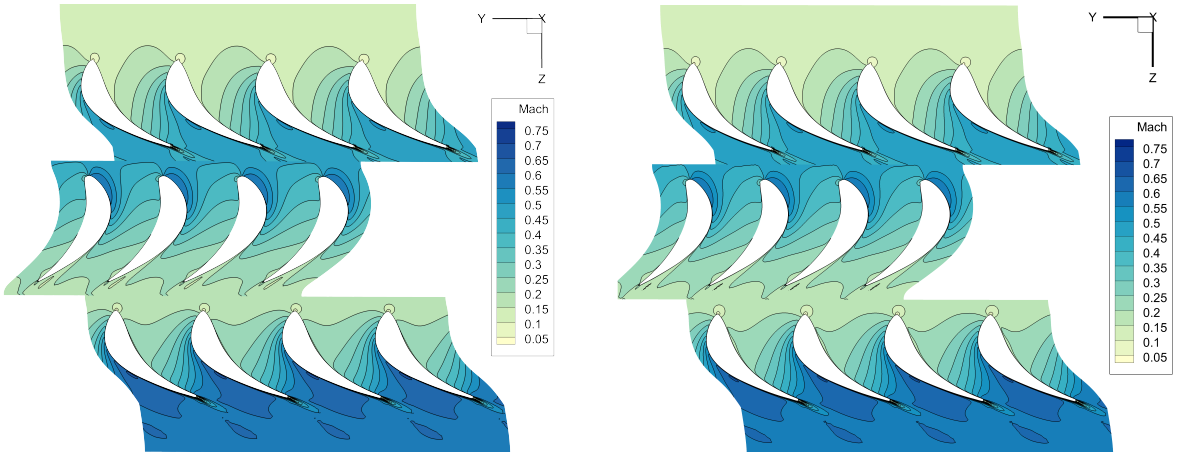
In this section, the results of adding the results from the Fourier component, are analysed. As it can be

seen in Figure 4.21 and 4.21, a good agreement is found between both simulations, observing just a small offset in the flow angle after the rotor and stator 2. The differences after the rotor near the shroud, may be cause by the influence of the tip clearance that becomes relevant when the non-reflectivity is imposed. In the outlet boundary condition, some differences were expected as well, as the flow behaviour is slightly different. However, since the outlet boundary is not too close to the blade exit, the discrepancy is rather small.

Since in this test case, the influence of the Fourier component, has already be shown to be small, the differences due to the treatment given to this Fourier component is expected to be small, and, indeed, if one looks at Figure 4.21 and Figure 4.22, a quite small difference is observed.

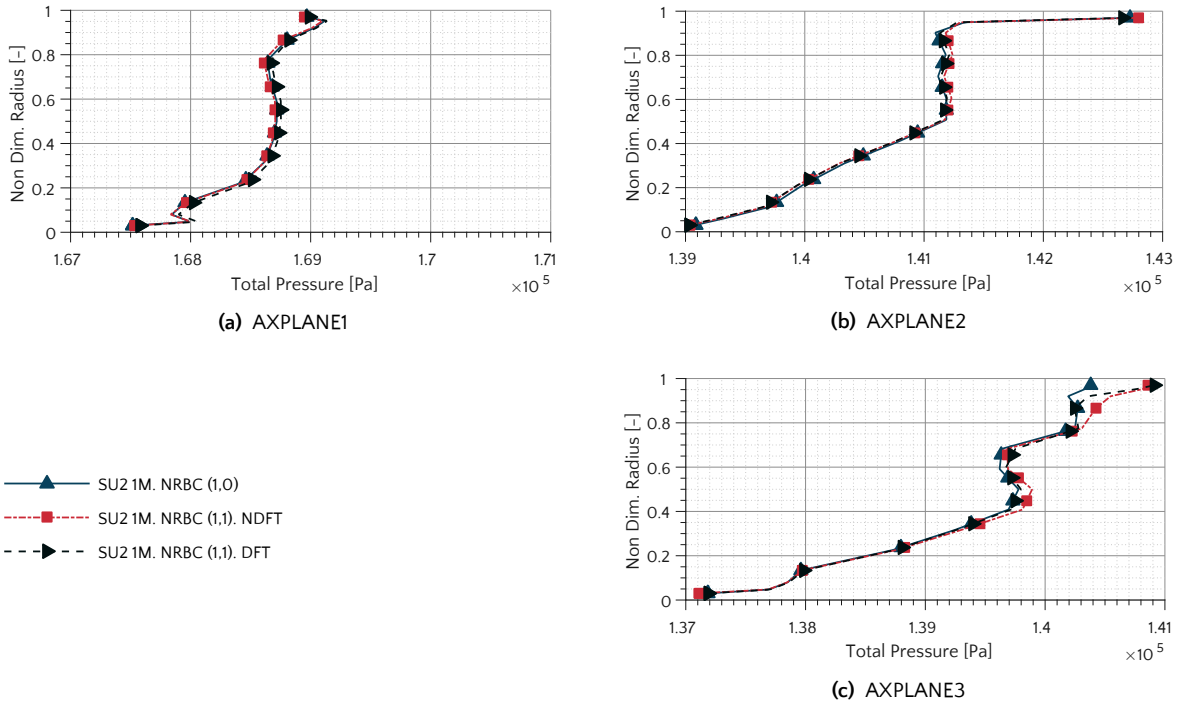
However, since the purpose of this change is to allow for more complicated test cases to use more complex and less restricted meshes, the fact that it presents a good agreement for such a case provides a good foundation to say that it will work well in such situations.

To observe the differences in the flow more clearly a blade to blade view of the Mach field is shown for both conditions (Figure 4.20). In this case, as in the average behaviour, small changes can be observed, however the effect of applying the non-reflectivity is not too relevant for this case.



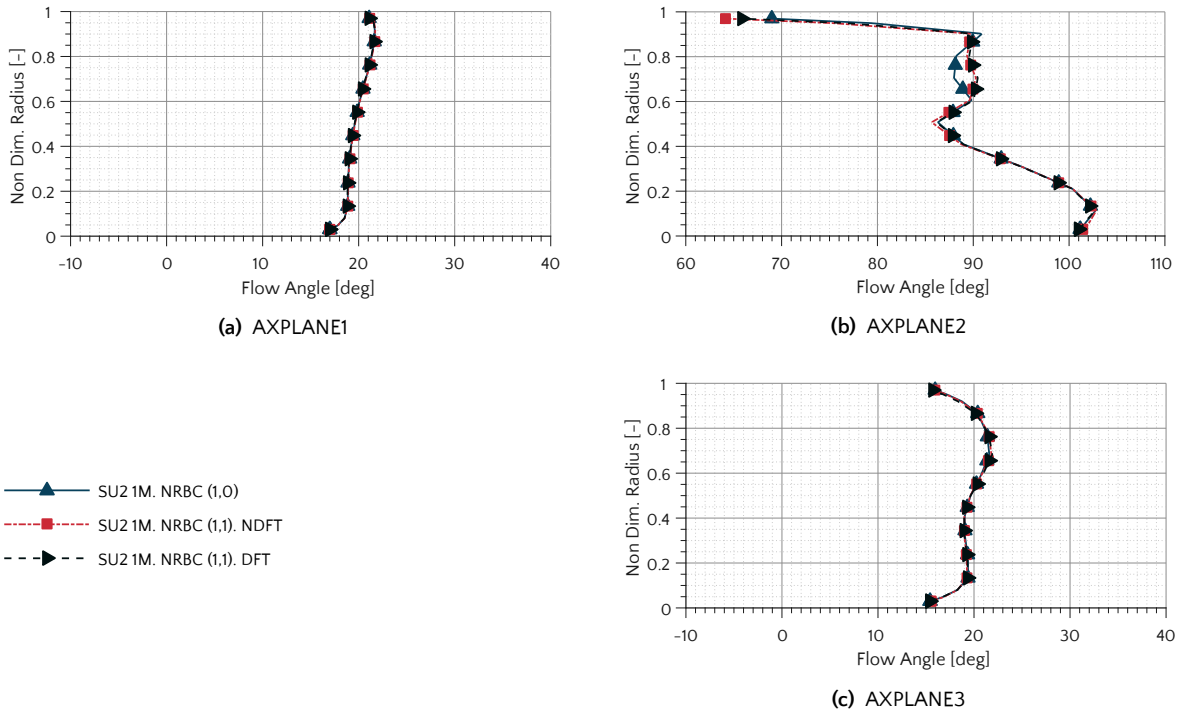
(a) NRBC (1,0) **(b)** NRBC (1,1)

Figure 4.20: Mach profiles at mid-radius. Results from SU2, NRBC comparison.



(a) AXPLANE1 **(b)** AXPLANE2 **(c)** AXPLANE3

Figure 4.21: Pitch-wise averaged Total Pressure. Results from SU2, Fourier Transform method comparison.



(a) AXPLANE1 **(b)** AXPLANE2 **(c)** AXPLANE3

Figure 4.22: Pitch-wise averaged Flow Angle. Results from SU2, Fourier Transform method comparison.

4.2.4. Comparison with experimental results and literature data. High mass flow

The results obtained with SU2 are now compared with the experimental results presented by Walraevens and Callus [31]. As pointed out already by Aubé and Hirsch [32], the experimental data has been collected at different times. This means that the data obtained for the different measurement planes is obtained with different inflow and outflow conditions, which limits the capabilities of comparison with the CFD solver results.

If one looks at the big numbers, mass flow in the measurements presents differences up to 0.3 kg/s, which explains most of the differences found in the data, since the boundary conditions are imposed using the data from *AXPLANE0* and *AXPLANE3*. At those measurement planes, a good agreement is found between experimental data and computed results for the boundary conditions imposed. For the total pressure, an increase is observed after the first stator, which is due to the lower pressure imposed compared to the one actually at the exit in the moment of measuring the data at that point. However, even though an offset is observed, the general trend is followed.

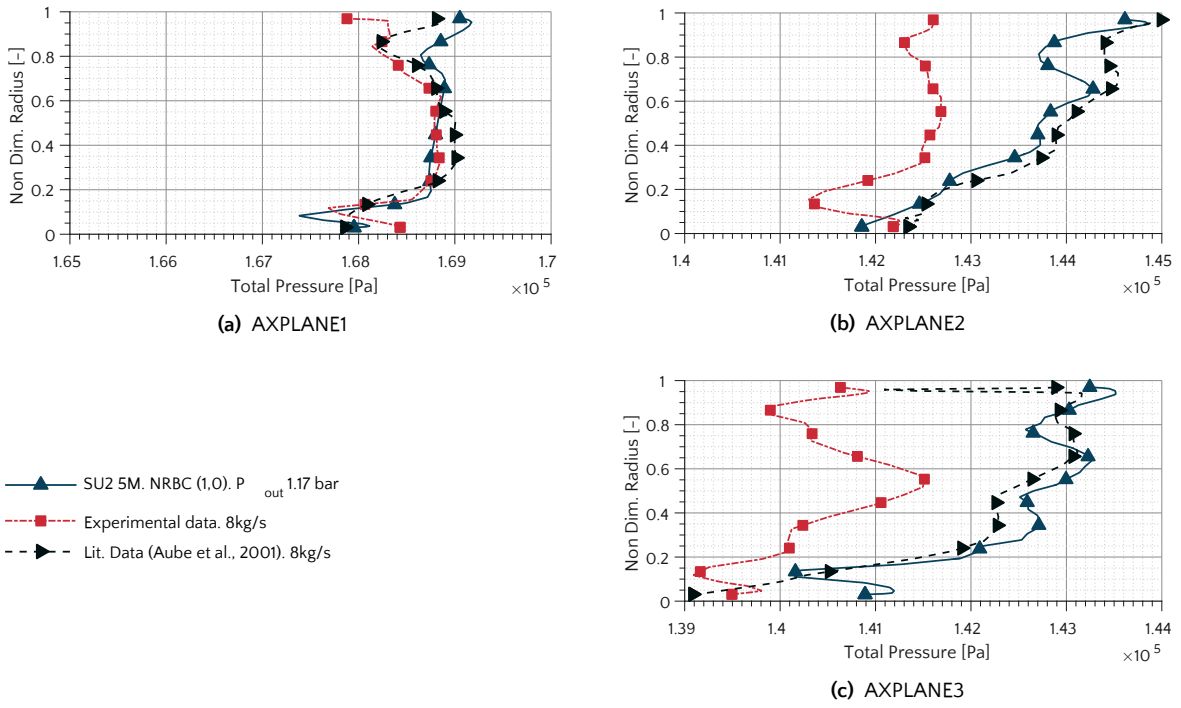
For the flow angle, biggest differences are found by comparing the results after the rotor. However, once again, the trend is being followed closely and bigger differences only near the tip.

By looking at other references that use the Aachen test case for validation, quite big differences are observed due to the aforementioned reason. Different techniques are used in order to compare the results from the experimental data that have different compromises. For example, in the work presented by Aubé and Hirsch [32], the outlet pressure at the boundary is such that the static pressure along the outlet boundary is imposed to yield the experimental mass flow at measuring station 2. This causes a better agreement in station 2, and differences in other stations as it can be seen in the paper.

Imposing boundary conditions such as those proposed by Aubé and Hirsch, a good agreement between SU2 and the computations presented in the paper are found. The boundary conditions that are imposed in this case are

Table 4.4: Boundary conditions. High mass flow studied case. Modified with data from literature.

Inlet BC	Total Pressure [Pa]	158497.2
	Total Temperature [K]	309.06
Outlet BC	Average Static Pressure [Pa]	117000.0

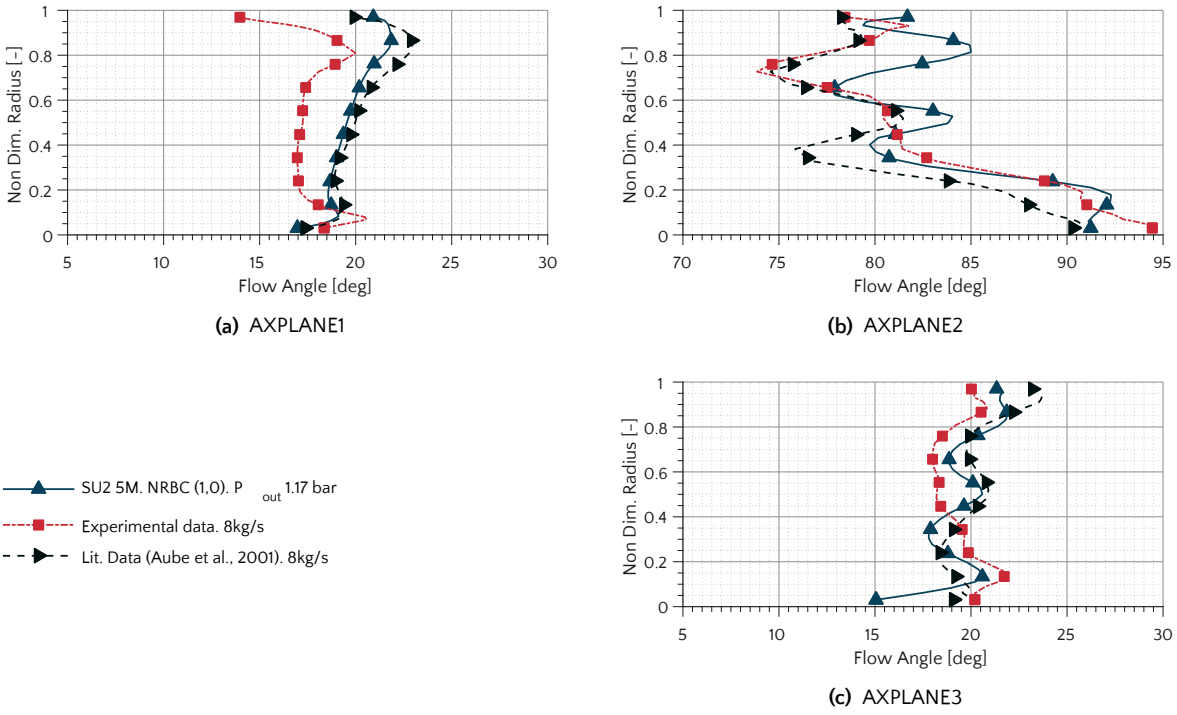


(a) AXPLANE1

(b) AXPLANE2

(c) AXPLANE3

Figure 4.23: Pitch-wise averaged Total Pressure. High mass flow case. Comparison between SU2 and simulations by Aubé and Hirsch [32]



(a) AXPLANE1

(b) AXPLANE2

(c) AXPLANE3

Figure 4.24: Pitch-wise averaged Flow Angle. High mass flow case. Comparison between SU2 and simulations by Aubé and Hirsch [32]

4.2.5. Comparison with experimental results and literature data. Low mass flow

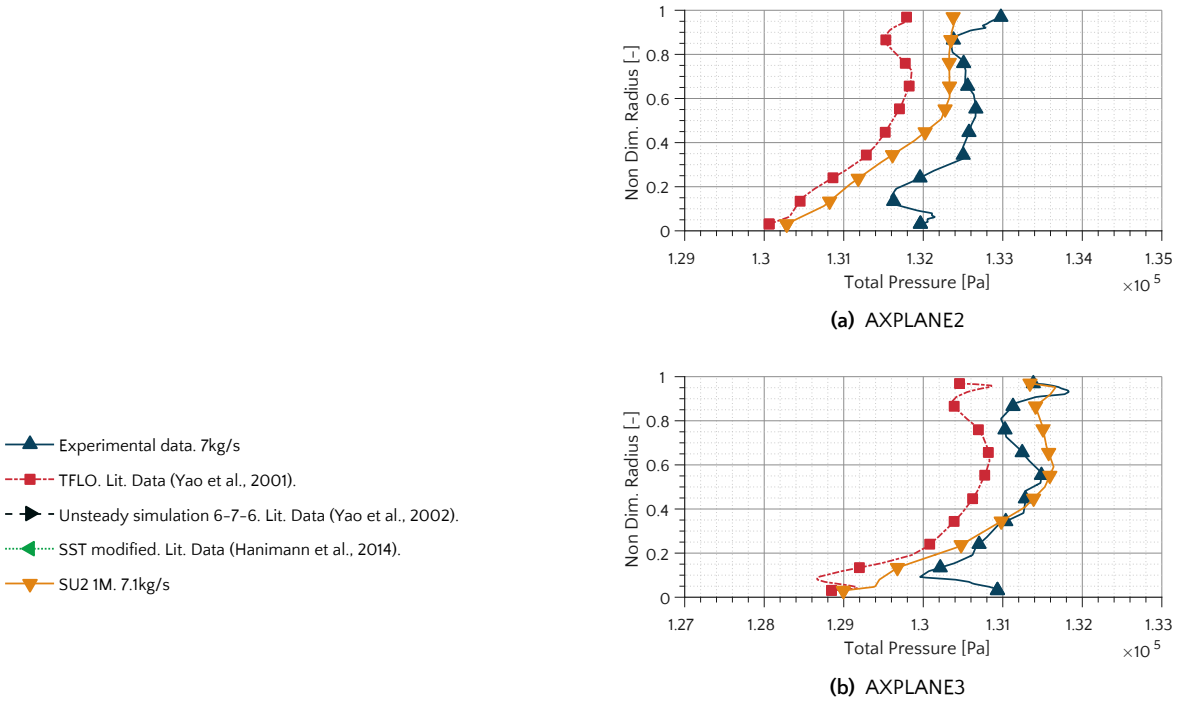
More data available in literature uses the test case with adapted conditions from the low mass flow case. Table 4.5 summarises the conditions that are imposed in SU2 in order to match the pressure distribution at the mixing interfaces and the mass flow through the turbine to a value of 7.1 kg/s.

Table 4.5: Boundary conditions. High mass flow studied case. Modified with data from literature.

Inlet BC	Total Pressure [Pa]	153500.0
	Total Temperature [K]	309.00
Outlet BC	Average Static Pressure [Pa]	111000.0

When analysing the results for the total pressure (Figure 4.25) , comparing with the experimental data [31] and Yao *et al.* [33] a good agreement with experimental data is found.

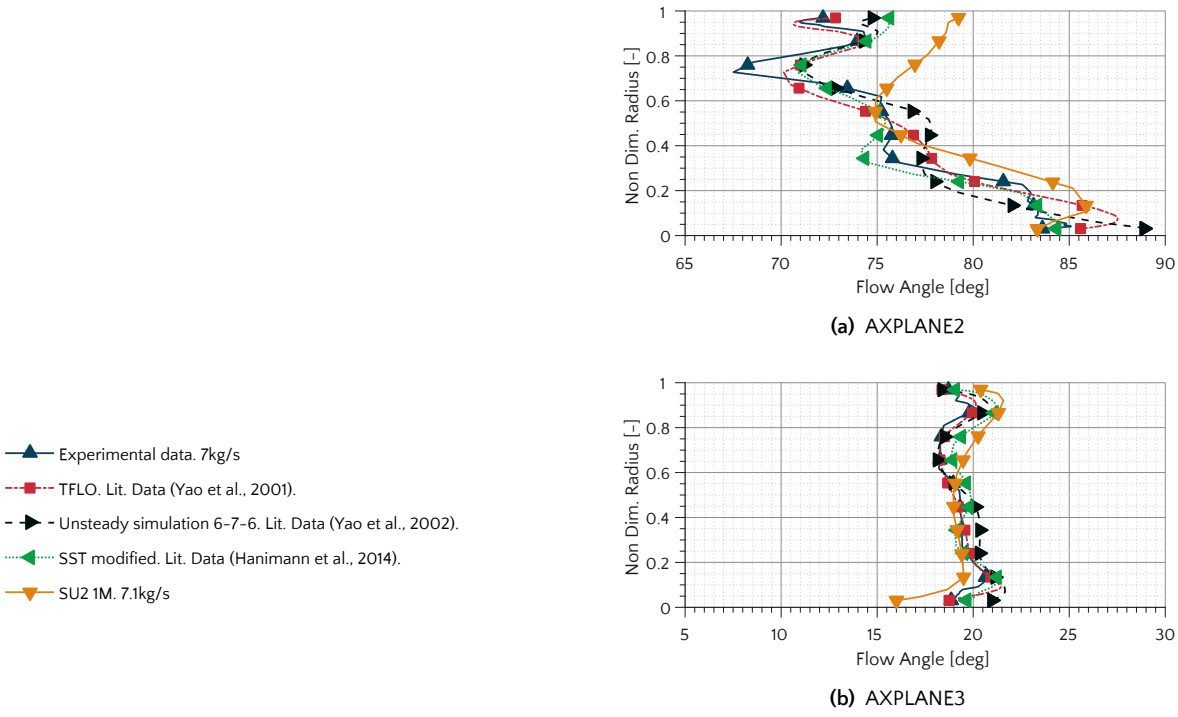
By comparing the flow angle (Figure 4.26), where also the data from Yao *et al.* [34] and Hanimann *et al.* [25] is available, the flow angle below the mid radius region is predicted accurately, however differences in flow angle behaviour are found near the tip region, due to a low resolution in the span-wise direction of the mesh used for this simulation. A simulation with an improved mesh as that shown in Section 4.2.2 would drive the results of the simulation closer to what is predicted in literature.



(a) AXPLANE2

(b) AXPLANE3

Figure 4.25: Pitch-wise averaged Total Pressure. Low mass flow case. Comparison between SU2 and literature data.



(a) AXPLANE2

(b) AXPLANE3

Figure 4.26: Pitch-wise averaged Flow Angle. Low mass flow case. Comparison between SU2 and literature data.

4.3. Summary

To summarise, in this section a detailed analysis of the results obtained with SU2 has been made and in general, quite a good agreement with the data provided in literature is found.

For the scope of this project, that is, the validation of the Mixing-Plane interface, it can be observed that

for the average variables at both sides of the interface (*INMIX* and *OUTMIX* planes) were matching between themselves quite well which means the implementation of the Mixing-Plane interface is done correctly.

Moreover, the proposed modification of the Non-Reflective Boundary Conditions formulation has proven to work quite well under the tested simulations. However, in order to obtain full conclusions regarding this, a full analysis with more different test cases should be performed.

A comparison with literature data has been performed in order to validate the implemented method. In general good agreement has been found, although further comparisons considering better meshes are required in order to draw relevant conclusions. It would be interesting as well to investigate other geometries to fully verify the implemented method.

5

Conclusions and recommendations

5.1. Conclusions

In this project the problem of stator and rotor interface has been investigated. The chosen alternative is the Mixing-Plane interface, which provides a good balance between accuracy and required computational time.

The Mixing-Plane interface with the quasi-3D Non-Reflective Boundary Conditions are implemented and the necessary approximations are made to make the method suitable for unstructured grids. The Non-uniform Direct Fourier Transform method is implemented and is validated among the DFT approach commonly used.

The method is validated by using experimental data found in literature, and to analyse more in detail and overcome the limitations of the data found in the literature, a comparison with a well known flow solver is performed and a good agreement is found.

Therefore, as a conclusion to this project, it can be said that the Mixing-Plane interface using the non-reflective boundary conditions for 3D test cases are implemented successfully and ready to perform calculations with unstructured meshes. This will allow to run the full 3D optimization of a multi-stage turbine in future projects.

5.2. Recommendations and future steps

As future steps regarding this project, some points can be made:

- Validation with a radial or mixed flow turbine, to prove that the NRBC are implemented correctly for this case.
- Validation with a compressor test case.
- Develop a good 3D unstructured mesh generator specialised in turbomachinery.
- And as last step, an adjoint calculation of a multi-stage 3D turbine to explore the full potential of the implementation of this options.

As further recommendations derived from this project, few points can be made:

- Preparation of the mesh can be time consuming and tricky. A good preparation for it and attention to details is needed in order to obtain reliable results from the simulation. High quality multi-block grids need a lot of pre-processing job before they can be simulated using SU2, therefore a preparation of a more automatic process would be preferred.
- A project that involves programming in an existing program can be overwhelming if the user has no previous experience. Therefore a good and realistic planning is necessary, and if time constrains are dominant, the objectives and the scope of the project should be revised regularly as they may need to change due to problems with obtaining good quality code or results.

Bibliography

[1] J. Bretheim and E. Bardy, "A Review of Power-Generating Turbomachines A Review of Power-Generating Turbomachines", in *2012 ASEE North Central Section Conference*, American Society for Engineering Education, 2012 (*see p. 1*).

[2] E. Logan and R. Roy, *Handbook of Turbomachinery*. 2003, p. 907 (*see p. 1*).

[3] A. P. Saxer and M. B. Giles, "Quasi-Three-Dimensional Nonreflecting Boundary Conditions for Euler Equations Calculations", *Journal of Propulsion and Power*, vol. 9, no. 2, 1993 (*see pp. 1, 2, 10, 11, 12, 14*).

[4] M. Pini, G. Persico, E. Casati, and V. Dossena, "Preliminary Design of a Centrifugal Turbine for Organic Rankine Cycle Applications", *Journal of Engineering for Gas Turbines and Power*, vol. 135, no. 4, p. 042 312, 2013 (*see p. 2*).

[5] S. Vitale, M. Pini, A. Ghidoni, and P. Colonna, "Fluid Dynamic Design and Analysis of a Highly Loaded Centrifugal Rotor for Mini ORC Power Systems", in *3rd International Seminar on ORC Power Systems*, Brussels, Belgium: ASME, 2015, pp. 1–11 (*see pp. 2, 6*).

[6] S. Vitale, M. Pini, G. Gori, A. A. Guardone, T. D. Economon, F. Palacios, and J. J. Alonso, "Extension of the SU2 Open Source CFD code to the simulation of turbulent flows of fluids modelled with complex thermophysical laws", in *22nd AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2015, pp. 1–22 (*see pp. 2, 6*).

[7] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "SU2: An Open-Source Suite for Multiphysics Simulation and Design", *AIAA Journal*, vol. 54, no. 3, pp. 828–846, Mar. 2016 (*see pp. 5, 6*).

[8] T. D. Economon, D. Mudigere, G. Bansal, A. Heinecke, F. Palacios, J. Park, M. Smelyanskiy, J. J. Alonso, and P. Dubey, "Performance optimizations for scalable implicit RANS calculations with SU2", *Computers and Fluids*, vol. 129, pp. 146–158, 2016 (*see p. 5*).

[9] T. A. Albring, M. Sagebaum, and N. R. Gauger, "Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2", in *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia: American Institute of Aeronautics and Astronautics, 2016 (*see p. 6*).

[10] B. Y. Zhou, T. A. Albring, N. R. Gauger, T. D. Economon, and J. J. Alonso, "An Efficient Unsteady Aerodynamic and Aeroacoustic Design Framework Using Discrete Adjoint", in *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia: American Institute of Aeronautics and Astronautics, Jun. 2016 (*see p. 6*).

[11] A. Ramezani, L. Remaki, and J. I. A. Suarez, "BBiped a SU2-based open-source : New Multiple Rotating Frame developments", in *Open Source CFD International Conference*, 2013, pp. 1–19 (*see pp. 6, 27*).

[12] J. D. Denton, "Some Limitations of Turbomachinery CFD", *ASME Turbo Expo 2010: Power for Land, Sea and Air*, pp. 1–11, 2010 (*see p. 7*).

[13] G. Fritsch and M. B. Giles, "An asymptotic analysis of mixing loss", *Journal of Turbomachinery-Transactions of the Asme*, vol. 117, no. 3, pp. 367–374, 1995 (*see p. 7*).

[14] G. Pullan, "Secondary Flows and Loss Caused by Blade Row Interaction in a Turbine Stage", *Journal of Turbomachinery*, vol. 128, no. July 2006, p. 484, 2006 (*see p. 7*).

[15] P. G. Tucker, "Trends in turbomachinery turbulence treatments", *Progress in Aerospace Sciences*, vol. 63, pp. 1–32, Nov. 2013. arXiv: `arXiv:1011.1669v3` (*see pp. 8, 9*).

[16] M. Giles, "UNSFLO: A Numerical Method For The Calculation Of Unsteady Flow In Turbomachinery", Tech. Rep. May, 1991 (*see p. 8*).

[17] F. Montomoli, H. P. Hodson, and L. Lapworth, "RANS-URANS in axial compressor, a design methodology", *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 225, no. 3, pp. 363–374, May 2011 (*see p. 8*).

[18] O. Uzol, Y.-C. Chow, J. Katz, and C. Meneveau, "Average Passage Flow Field and Deterministic Stresses in the Tip and Hub Regions of a Multistage Turbomachine", *Journal of Turbomachinery*, vol. 125, no. 4, p. 714, 2003 (*see p. 8*).

[19] V. Brost, A. Ruprecht, and M. Maihöfer, "Rotor-stator interactions in an axial turbine, a comparison of transient and steady state frozen rotor simulations", *Conference on Case Studies in Hydraulic Systems-CSHS03, Belgrade*, pp. 1–9, 2003 (*see p. 9*).

[20] J. D. Denton, "An Improved Time-Marching Method for Turbomachinery Flow Calculation", *Journal of Engineering for Power*, vol. 105, no. 3, p. 514, 1983 (*see pp. 9, 10*).

[21] J. D. Denton, "The Calculation of Three-Dimensional Viscous Flow Through Multistage Turbomachines", *Journal of Turbomachinery*, vol. 114, no. 1, pp. 18–26, 1992 (*see pp. 10, 11*).

[22] D. X. Wang, "An Improved Mixing-Plane Method for Analyzing Steady Flow Through Multiple-Blade-Row Turbomachines", *Journal of Turbomachinery*, vol. 136, no. 8, p. 081 003, 2014 (*see pp. 11, 12*).

[23] F. Gisbert and R. Corral, "A Novel Mixing Plane Method Using Non-Reflecting Boundary Conditions For Multi-Row Analysis in Turbomachines", in *ASME Turbo Expo*, Montréal, Canada: ASME, 2015, pp. 1–11 (*see p. 11*).

[24] P. Du and F. Ning, "Validation of a novel mixing-plane method for multistage turbomachinery steady flow analysis", *Chinese Journal of Aeronautics*, vol. 29, no. October, pp. 1563–1574, 2016 (*see pp. 11, 12*).

[25] L. Hanimann, L. Mangani, E. Casartelli, T. Mokulys, and S. Mauri, "Development of a Novel Mixing Plane Interface Using a Fully Implicit Averaging for Stage Analysis", *Trans. ASME J. Turbomachinery*, vol. 136, no. 8, pp. 1–14, 2014 (*see pp. 11, 43*).

[26] M. B. Giles, "UNSFLO: A numerical method for unsteady inviscid flow in turbomachinery", Gas Turbine Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, Tech. Rep. May, 1986, p. 91 (*see p. 14*).

[27] M. B. Giles, "Nonreflecting boundary conditions for Euler equation calculations", *AIAA Journal*, vol. 28, no. 12, pp. 2050–2058, Dec. 1990 (*see p. 18*).

[28] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "SU2: An Open-Source Suite for Multiphysics Simulation and Design", *AIAA Journal*, vol. 54, no. 3, pp. 1–19, 2015 (*see p. 22*).

[29] B. Stephan, H. E. Gallus, and R. Niehuis, "Experimental investigations of tip clearance flow and its influence on secondary flows in a 1-1/2 stage axial turbine", in *ASME TurboExpo 2000*, Munich, Germany: ASME, 2000, pp. 1–12 (*see p. 29*).

[30] R. E. Walraevens and H. E. Gallus, *Three Dimensional Structure of Unsteady Flow Downstream the Rotor in a 1-1/2 Stage Turbine*, 1995 (*see p. 29*).

[31] R. E. Walraevens and H. E. Callus, "Testcase 6: 1-1/2 Stage Axial Flow Turbine", Aachen, Germany, 1997 (*see pp. 41, 43*).

[32] M. Aubé and C. Hirsch, "Numerical Investigation of a 1-1/2 Axial Turbine Stage at Quasi-Steady and Fully Unsteady Conditions", in *Volume 1: Aircraft Engine; Marine; Turbomachinery; Microturbines and Small Turbomachinery*, ASME, Jun. 2001, V001T03A013 (*see pp. 41, 42, 43*).

[33] J. Yao, J. J. Alonso, A. Jameson, and F. Liu, "Development and Validation of a Massively Parallel Flow Solver for Turbomachinery Flows", *Journal of Propulsion and Power*, vol. 17, no. 3, pp. 659–668, 2001 (*see p. 43*).

[34] J. Yao, R. L. Davis, J. J. Alonso, and A. Jameson, "Massively Parallel Simulation of the Unsteady Flow in an Axial Turbine Stage", *Journal of Propulsion and Power*, vol. 18, no. 2, pp. 465–471, 2002 (*see p. 43*).

A

Aachen geometry definition

The Aachen turbine is a 1-1/2 stage turbine that consist of 2 stators and a rotor. Both the stator and the rotor are prismatic, so their profile is the same at the hub and the shroud.

Table A.1: Aachen turbine. Hub and shroud curve definition.

Hub		Shroud	
R [m]	Z [m]	R [m]	Z [m]
0.245	0.000	0.300	0.000
0.245	0.300	0.300	0.300

Table A.2: Aachen turbine stator profile

****Stator****					
Y [m]	Z [m]	Y [m]	Z [m]	Y [m]	Z [m]
-5.0610E-02	4.3620E-02	-5.5628E-04	1.7000E-02	-1.5374E-02	1.6000E-02
-5.0571E-02	4.3699E-02	-2.9970E-04	1.6000E-02	-1.5903E-02	1.7000E-02
-5.0521E-02	4.3771E-02	-1.2384E-04	1.5000E-02	-1.6463E-02	1.8000E-02
-5.0462E-02	4.3836E-02	-2.6107E-05	1.4000E-02	-1.7064E-02	1.9000E-02
-5.0394E-02	4.3891E-02	7.9000E-08	1.3000E-02	-1.7712E-02	2.0000E-02
-5.0318E-02	4.3936E-02	-2.9655E-05	1.2000E-02	-1.8406E-02	2.1000E-02
-5.0237E-02	4.3969E-02	-1.1410E-04	1.1000E-02	-1.9140E-02	2.2000E-02
-5.0152E-02	4.3991E-02	-2.6543E-04	1.0000E-02	-1.9910E-02	2.3000E-02
-5.0064E-02	4.4000E-02	-4.7852E-04	9.0000E-03	-2.0718E-02	2.4000E-02
-4.9976E-02	4.3996E-02	-7.4303E-04	8.0000E-03	-2.1575E-02	2.5000E-02
-4.9890E-02	4.3979E-02	-1.0635E-03	7.0000E-03	-2.2489E-02	2.6000E-02
-4.6219E-02	4.3000E-02	-1.4542E-03	6.0000E-03	-2.3458E-02	2.7000E-02
-4.2471E-02	4.2000E-02	-1.8637E-03	5.0000E-03	-2.4511E-02	2.8000E-02
-3.9056E-02	4.1000E-02	-2.3711E-03	4.0000E-03	-2.5631E-02	2.9000E-02
-3.5914E-02	4.0000E-02	-3.0261E-03	3.0000E-03	-2.6814E-02	3.0000E-02
-3.2996E-02	3.9000E-02	-3.7120E-03	2.0000E-03	-2.8100E-02	3.1000E-02
-3.0225E-02	3.8000E-02	-4.5463E-03	1.0000E-03	-2.9468E-02	3.2000E-02
-2.7588E-02	3.7000E-02	-5.1934E-03	3.7850E-04	-3.0927E-02	3.3000E-02
-2.5054E-02	3.6000E-02	-5.5754E-03	1.1141E-04	-3.2474E-02	3.4000E-02
-2.2615E-02	3.5000E-02	-6.2858E-03	4.4380E-06	-3.4085E-02	3.5000E-02
-2.0267E-02	3.4000E-02	-7.0069E-03	1.0712E-04	-3.5791E-02	3.6000E-02
-1.8013E-02	3.3000E-02	-7.3548E-03	3.7168E-04	-3.7562E-02	3.7000E-02
-1.5959E-02	3.2000E-02	-7.9728E-03	1.0000E-03	-3.9455E-02	3.8000E-02
-1.4065E-02	3.1000E-02	-8.7443E-03	2.0000E-03	-4.1428E-02	3.9000E-02
-1.2363E-02	3.0000E-02	-9.3932E-03	3.0000E-03	-4.3539E-02	4.0000E-02
-1.0813E-02	2.9000E-02	-9.9699E-03	4.0000E-03	-4.5796E-02	4.1000E-02
-9.3694E-03	2.8000E-02	-1.0501E-02	5.0000E-03	-4.8243E-02	4.2000E-02
-8.0541E-03	2.7000E-02	-1.1006E-02	6.0000E-03	-5.0276E-02	4.2831E-02
-6.8507E-03	2.6000E-02	-1.1445E-02	7.0000E-03	-5.0361E-02	4.2873E-02
-5.7511E-03	2.5000E-02	-1.1891E-02	8.0000E-03	-5.0438E-02	4.2929E-02
-4.7669E-03	2.4000E-02	-1.2308E-02	9.0000E-03	-5.0506E-02	4.2995E-02
-3.8694E-03	2.3000E-02	-1.2715E-02	1.0000E-02	-5.0562E-02	4.3072E-02
-3.0749E-03	2.2000E-02	-1.3123E-02	1.1000E-02	-5.0606E-02	4.3156E-02
-2.3868E-03	2.1000E-02	-1.3537E-02	1.2000E-02	-5.0636E-02	4.3246E-02
-1.8005E-03	2.0000E-02	-1.3964E-02	1.3000E-02	-5.0652E-02	4.3340E-02
-1.3052E-03	1.9000E-02	-1.4409E-02	1.4000E-02	-5.0652E-02	4.3435E-02
-8.9139E-04	1.8000E-02	-1.4876E-02	1.5000E-02	-5.0638E-02	4.3529E-02
				-5.0610E-02	4.3620E-02

Table A.3: Aachen turbine rotor profile

****Rotor****					
Y [m]	Z [m]	Y [m]	Z [m]	Y [m]	Z [m]
-4.3034E-02	2.5572E-01	-8.9062E-04	2.2500E-01	-1.4101E-02	2.1600E-01
-4.3010E-02	2.5578E-01	-5.9487E-04	2.2400E-01	-1.4134E-02	2.1700E-01
-4.2976E-02	2.5584E-01	-3.5269E-04	2.2300E-01	-1.4209E-02	2.1800E-01
-4.2932E-02	2.5589E-01	-1.7055E-04	2.2200E-01	-1.4327E-02	2.1900E-01
-4.2882E-02	2.5593E-01	-5.7723E-05	2.2100E-01	-1.4489E-02	2.2000E-01
-4.2825E-02	2.5596E-01	-3.9990E-06	2.2000E-01	-1.4697E-02	2.2100E-01
-4.2763E-02	2.5599E-01	-1.1524E-05	2.1900E-01	-1.4945E-02	2.2200E-01
-4.2699E-02	2.5600E-01	-8.5688E-05	2.1800E-01	-1.5240E-02	2.2300E-01
-4.2633E-02	2.5600E-01	-2.2754E-04	2.1700E-01	-1.5573E-02	2.2400E-01
-4.2568E-02	2.5599E-01	-4.3215E-04	2.1600E-01	-1.5925E-02	2.2500E-01
-4.2506E-02	2.5597E-01	-6.9027E-04	2.1500E-01	-1.6316E-02	2.2600E-01
-4.0175E-02	2.5500E-01	-1.0034E-03	2.1400E-01	-1.6749E-02	2.2700E-01
-3.7771E-02	2.5400E-01	-1.3808E-03	2.1300E-01	-1.7210E-02	2.2800E-01
-3.5545E-02	2.5300E-01	-1.8201E-03	2.1200E-01	-1.7693E-02	2.2900E-01
-3.3370E-02	2.5200E-01	-2.3266E-03	2.1100E-01	-1.8217E-02	2.3000E-01
-3.1297E-02	2.5100E-01	-2.9116E-03	2.1000E-01	-1.8782E-02	2.3100E-01
-2.9287E-02	2.5000E-01	-3.5858E-03	2.0900E-01	-1.9373E-02	2.3200E-01
-2.7305E-02	2.4900E-01	-4.3473E-03	2.0800E-01	-1.9985E-02	2.3300E-01
-2.5374E-02	2.4800E-01	-5.1759E-03	2.0700E-01	-2.0637E-02	2.3400E-01
-2.3464E-02	2.4700E-01	-6.1204E-03	2.0600E-01	-2.1334E-02	2.3500E-01
-2.1630E-02	2.4600E-01	-7.1842E-03	2.0500E-01	-2.2045E-02	2.3600E-01
-1.9895E-02	2.4500E-01	-8.3793E-03	2.0400E-01	-2.2792E-02	2.3700E-01
-1.8265E-02	2.4400E-01	-9.8127E-03	2.0300E-01	-2.3535E-02	2.3800E-01
-1.6734E-02	2.4300E-01	-1.0379E-02	2.0267E-01	-2.4349E-02	2.3900E-01
-1.5278E-02	2.4200E-01	-1.1501E-02	2.0219E-01	-2.5198E-02	2.4000E-01
-1.3882E-02	2.4100E-01	-1.2613E-02	2.0200E-01	-2.6055E-02	2.4100E-01
-1.2551E-02	2.4000E-01	-1.3649E-02	2.0219E-01	-2.6953E-02	2.4200E-01
-1.1299E-02	2.3900E-01	-1.4352E-02	2.0261E-01	-2.7875E-02	2.4300E-01
-1.0141E-02	2.3800E-01	-1.4715E-02	2.0300E-01	-2.8833E-02	2.4400E-01
-9.0558E-03	2.3700E-01	-1.5186E-02	2.0400E-01	-2.9838E-02	2.4500E-01
-8.0405E-03	2.3600E-01	-1.5243E-02	2.0500E-01	-3.0889E-02	2.4600E-01
-7.0880E-03	2.3500E-01	-1.5153E-02	2.0600E-01	-3.1974E-02	2.4700E-01
-6.2034E-03	2.3400E-01	-1.5021E-02	2.0700E-01	-3.3112E-02	2.4800E-01
-5.3952E-03	2.3300E-01	-1.4872E-02	2.0800E-01	-3.4304E-02	2.4900E-01
-4.6484E-03	2.3200E-01	-1.4717E-02	2.0900E-01	-3.5520E-02	2.5000E-01
-3.9578E-03	2.3100E-01	-1.4571E-02	2.1000E-01	-3.6809E-02	2.5100E-01
-3.3140E-03	2.3000E-01	-1.4437E-02	2.1100E-01	-3.8155E-02	2.5200E-01
-2.7145E-03	2.2900E-01	-1.4320E-02	2.1200E-01	-3.9554E-02	2.5300E-01
-2.1721E-03	2.2800E-01	-1.4223E-02	2.1300E-01	-4.1013E-02	2.5400E-01
-1.6847E-03	2.2700E-01	-1.4150E-02	2.1400E-01	-4.2473E-02	2.5500E-01
-1.2538E-03	2.2600E-01	-1.4107E-02	2.1500E-01	-4.2880E-02	2.5528E-01
				-4.2920E-02	2.5531E-01
				-4.2955E-02	2.5535E-01
				-4.2986E-02	2.5539E-01
				-4.3011E-02	2.5543E-01
				-4.3030E-02	2.5548E-01
				-4.3044E-02	2.5552E-01
				-4.3051E-02	2.5557E-01
				-4.3052E-02	2.5562E-01
				-4.3046E-02	2.5567E-01
				-4.3034E-02	2.5572E-01

B

SU2 Input 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= NAVIER_STOKES
%
% 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= 0.5
%
% 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= 100000.0
%
% Free-stream temperature (273.15 K by default)
FREESTREAM_TEMPERATURE=  300.0
%
% 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 (
    TEMPERATURE_FS) to initialize the solution
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 the solution (REYNOLDS, TD_CONDITIONS)
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= 287.058
%
% Critical Temperature (273.15 K by default)
CRITICAL_TEMPERATURE= 273.15
%
% Critical Pressure (101325.0 N/m^2 by default)
CRITICAL_PRESSURE= 101325.0
%
% Acentri factor (0.035 (air))
ACENTRIC_FACTOR= 0.035
%
%
% --------------------------- VISCOSITY MODEL ---------------------------------%
%
% Viscosity model (SUTHERLAND, CONSTANT_VISCOSITY).
VISCOSITY_MODEL= SUTHERLAND
%
% Molecular Viscosity that would be constant (1.716E-5 by default)
MU_CONSTANT= 1.716E-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= 0.0257
%
%
% -------------------- BOUNDARY CONDITION DEFINITION --------------------------%
%
% Navier-Stokes wall boundary marker(s) (NONE = no marker)
MARKER_HEATFLUX= (blade1, 0.0, blade2, 0.0,  blade3, 0.0, hub1, 0.0, shroud1, 0.0, hub2,
    0.0, shroud2, 0.0, hub3, 0.0, shroud3, 0.0)
%
% 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_angle_y-axis, rot_angle_z-axis, translation_x, translation_y, translation_z)
MARKER_PERIODIC= (per1, per2, 0.0, 0.0, 0.0, 0.0, 0.0, 10.0, 0.0, 0.0, 0.0, per3, per4,
    0.0, 0.0, 0.0, 0.0, 0.0, 8.780487805, 0.0, 0.0, 0.0, per5, per6, 0.0, 0.0, 0.0, 0.0,
    0.0, 10.0, 0.0, 0.0, 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 the machine (i.e, from the first blade to the last)
MARKER_TURBOMACHINERY= (inlet, outmix1, inmix1, outmix2, inmix2, outlet)
%
% Mixing-plane interface markers must be specified to activate the transfer of information
    between zones
MARKER_MIXINGPLANE_INTERFACE= (outmix2, inmix2, outmix1, inmix1)
%
% Non reflecting boundary condition for inflow, outfolw and mixing-plane
% Format inlet:  ( marker, TOTAL_CONDITIONS_PT, Total Pressure , Total Temperature, Flow
    dir-norm, Flow dir-tang, Flow dir-span, under-relax-avg, under-relax-fourier)
% Format outlet: ( marker, STATIC_PRESSURE, Static Pressure value, -, -, -, -, under-relax-
    avg, under-relax-fourier)
% Format mixing-plane in and out: ( marker, MIXING_IN or MIXING_OUT, -, -, -, -, -, -,
    under-relax-avg, under-relax-fourier)
MARKER_NRBC= (inlet, TOTAL_CONDITIONS_PT, 169458.44, 305.75, 1.0, 0.0, 0.0, 1.0, 1.0,
    outmix1, MIXING_OUT, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, inmix1, MIXING_IN, 0.0, 0.0,
    0.0, 0.0, 0.0, 1.0, 0.5, outmix2, MIXING_OUT, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.5, inmix2
    , MIXING_IN, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, outlet, STATIC_PRESSURE_1D, 110562.52,
    0.0, 0.0, 0.0, 0.0 , 1.0, 1.0)
%
SPATIAL_FOURIER= YES
%
%---------------------------- TURBOMACHINERY SIMULATION -----------------------------%
%
% Specify kind of architecture (AXIAL, CENTRIPETAL, CENTRIFUGAL, CENTRIPETAL_AXIAL)
TURBOMACHINERY_KIND= AXIAL AXIAL AXIAL
%
% Specify kind of interpolation for the mixing-plane (LINEAR_INTERPOLATION, NEAREST_SPAN,
    MATCHING)
MIXINGPLANE_INTERFACE_KIND= LINEAR_INTERPOLATION
%
% Specify option for turbulent mixing-plane (YES, NO) default NO
TURBULENT_MIXINGPLANE= NO
%
% 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, total number of iteration for the ramp)
RAMP_OUTLET_PRESSURE_COEFF= (140000.0, 10.0, 2000)
%
```

```
% Specify ramp option for rotating frame (YES, NO) default NO
RAMP_ROTATING_FRAME= NO
%
% Parameters of the rotating frame ramp (starting rotational speed, updating-iteration-
    frequency, total number of iteration for the ramp)
RAMP_ROTATING_FRAME_COEFF= (0.0, 1.0, 1000)
%
% Specify Kind of average process for linearizing the Navier-Stokes equation at inflow and
    outflow BC included mixing-plane
% (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, tollerance, max number of iterations)
MIXEDOUT_COEFF= (0.1, 1.0E-06, 60)
%
% Limit of Mach number below which the mixedout algorithm is substituted with a AREA
    average algorithm
AVERAGE_MACH_LIMIT= 0.03
%
%
% ------------------------ SURFACES IDENTIFICATION -----------------------------%
%
% Marker(s) of the surface in the surface flow solution file
MARKER_PLOTTING= (blade1, blade2, blade3)
%
%
% -------------------------- 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 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  -366.518833333 0.0
%
%
% ------------- 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= YES
%
% 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, 20.0)
%
%
% ------------------------ LINEAR SOLVER DEFINITION ---------------------------%
%
```

```
% Linear solver or smoother for implicit formulations (BCGSTAB, FGMRES, SMOOTHER_JACOBI,
    SMOOTHER_ILU0, SMOOTHER_LUSGS, SMOOTHER_LINELET)
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= 0.3
%
% 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
%
% 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
%
%
% ------------------- 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= 6001
% 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= aachen_1M_tip.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
% Writing restart type
WRT_BINARY_RESTART= YES
%
READ_BINARY_RESTART= YES
```