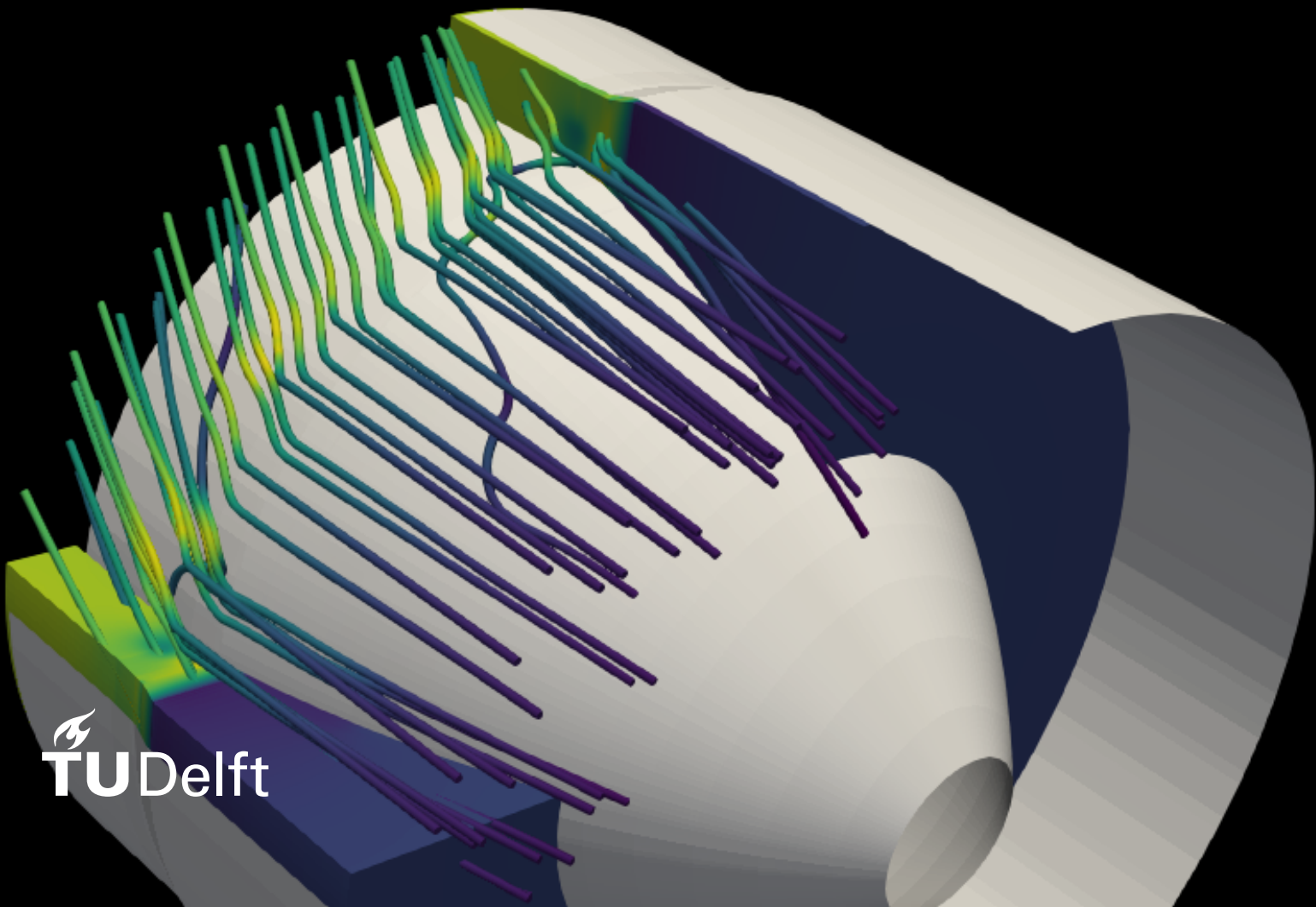


Turbomachinery Analysis and Design Using Body-Force Modeling in SU2

E. C. Bunschoten

Master Thesis Report

Aerospace Propulsion and Power Engineering
15-10-2020



Turbomachinery Analysis and Design Using Body-Force Modeling in SU2

by

E. C. Bunschoten

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday October 15, 2020 at 14:00

Student number:	4455584
Project duration:	January 28, 2020 – October 15, 2020
Thesis committee:	Prof. dr. ir. M. Pini, TU Delft, supervisor
	ir. N. Anand, TU Delft, supervisor
	Prof. dr. ir. P. Colonna TU Delft
	dr. R. P. Dwight TU Delft

Preface

This report represents the final research project conducted in the Master curriculum of Aerospace Propulsion and Power Engineering. This research report is intended for those who seek insight in the workings of body-force models, the implementation of Thollets body-force model into SU2 and its capabilities. Information regarding the implementation can be found in Chapter 3, Section 3.3. Additionally, readers who are interested in using body-force modeling in a design context could find inspiration in this document and utilize the methods posed. For those interested in the design analysis workflow, an example analysis case can be found in Appendix A. Here, a GitHub link to the method can also be found.

Including literature study, it has taken about a year to complete this research, in which the world has changed drastically. The situation regarding the global pandemic caused several delays and hindered communication, as physical meetings with my supervisors were prohibited. Despite these difficulties, I always felt well-supported. Matteo Pini, the professor who guided me through the research was a joy to work with. He was always curious regarding updates and offered valuable advice when major decisions had to be made regarding the research. Special thanks also goes to Nitish Anand, a PHD candidate within the Propulsion and Power research group at TU Delft. His expertise in the SU2 code structure was of crucial help while implementing the body-force model into SU2. Along side that, he was (nearly) always available for video chat to resolve technical issues, which was highly appreciated. Finally, my gratitude goes to William Thollet and Luis Lopez de Vega, of Airbus. They originally developed the body-force model used in this thesis and offered their advice regarding the implementation of it.

Abstract

As aircraft manufacturers push for better fuel economy, novel aircraft concepts, like the propulsive fuselage concept have emerged. One of the prime component of such concepts is the boundary layer ingestion (BLI) engine. Such engine operates by ingesting the low-momentum fluid emerging from the surface of the aircraft fuselage which enables to enhance its propulsive efficiency. The design of a BLI engine is fundamentally different than that of a conventional aircraft engine as the BLI fan experiences distorted airflow at every revolution. Therefore, when its performance are analyzed by means of CFD, single passage simulations become insufficient to represent the flow characteristics. and extremely expensive simulation of the entire annulus become necessary for analysis and design purposes. Especially during conceptual design, where several BLI engine configurations must be investigated in combination with their airframe integration, new reduced-order methods are required to alleviate the computational burden while retaining a sufficient level of accuracy.

An alternative, computationally efficient approach capable of coping with the full-annulus simulation is the well-established body-force modelling (BFM). BFM is a technique through which the physical blade rows are replaced by a volumetric force field which mimics the flow turning imparted by the physical blade rows. Thanks to this simplification, BFM does not require the physical blades to be accurately reproduced in the mesh, thus offering substantial computational cost reduction. Due to the associated gains in computational efficiency, this method is commonly used for the analysis and off-design performance prediction of full-annulus geometries, which are computationally demanding through conventional methods. Although the method has been established for analysis purposes, its capability as a design tool was never assessed.

Implementation of BFM in a CFD code that can be automatically differentiated to attain design sensitivities via the adjoint method may enable efficient BFM-based design of complex engine configurations.

In this research, the BFM model originally developed by Ref. [1] and improved by Ref. [2] was implemented into open-source CFD software SU2, which is equipped with a discrete adjoint solver based on operator overloading. In addition, parallel force and a metal blockage source term were added to the existing formulation to increase the fidelity of the BFM. The implementation was validated by comparing the results obtained from the BFM and the physical blade computation on an exemplary axial turbine test case.

The results show that the BFM was capable of providing static and stagnation pressure and temperature trends with an accuracy of approximately 94%. Furthermore, the absolute flow deflection by the stator and rotor rows were found to deviate by 5° and 17° . The BFM was found to be 3 orders of magnitude faster than the equivalent physical blade computation.

Table of Contents

Preface	i
Abstract	ii
List of Symbols	ii
List of Figures	iv
List of Tables	v
1 Introduction	1
1.1 Motivation and Research Objective	1
1.2 Original Contributions	2
1.3 Outline	3
2 Background Knowledge	4
2.1 Body-force modeling	4
2.2 Applications and limitations of body-force modeling	7
2.3 Original BFM implementation in SU2	8
2.4 Thollet's BFM	10
3 Research Methodology	13
3.1 Ray-Cast Interpolation	13
3.2 3D BFM transformation	14
3.3 BFM solver structure	16
3.4 Automated Workflow for BFM Computations	16
4 BFM Verification	19
4.1 Verification of Parallel Force Implementation	19
4.2 Metal Blockage Model Verification	21
5 BFM Case Study	26
5.1 Aachen Turbine Blade Computation Setup	26
5.2 BFM Computation Setup	28
6 Results	30
6.1 Verification of Parallel Force Implementation	30
6.2 Verification of Metal Blockage Model Implementation	32
6.3 Aachen Turbine Case Study Results	36
6.3.1 Results as a Function of Axial Node Density	36
6.3.2 Tangential node count study	49
6.3.3 Grid convergence	49
7 Conclusions	52
8 Final Remarks and Perspectives	54
8.1 Final Remarks	54
8.2 Perspectives	54
A Design Work Flow Demonstration	56
A.1 Design variable selection	56

B Radial section data	60
References	63

List of Symbols

Abbreviations

ADM	Actuator Disc Model
BFM	Body-Force Model
BLI	Boundary-Layer Ingestion
NS	Navier-Stokes
RANS	Reynolds Averaged Navier-Stokes
SU2	Stanford University Unstructured
TSFC	Thrust Specific Fuel Consumption
URANS	Unsteady Reynolds Averaged Navier-Stokes

Roman Symbols

b	Blockage factor	[-]
c_p	Specific enthalpy	[J kg ⁻¹ K ⁻¹]
F	Body-force	[kg m ⁻² s ⁻²]
f	Body-force, momentum vector	
h	Enthalpy	[J kg ⁻¹ K ⁻¹]
n	Camber normal vector component	
N_x	Node count in axial direction	[-]
R^*	Degree of reaction	[-]
T	Temperature	[K]
u	Absolute velocity	[m s ⁻¹]
R	Gas constant	[J kg ⁻¹ K ⁻¹]

Greek Symbols

β	Blade metal angle	[°]
γ	Specific heat ratio	[-]
Ω	Rotation rate	[rad/s]
ϕ	Flow coefficient	[-]
ψ	Work coefficient	[-]
ξ	Vortex-free twist parameter	[-]

Subscripts

θ	Tangential
----------	------------

ax	Axial
n	Normal
p	Parallel
r	Radial
s	Static conditions
t	Stagnation conditions
hub	At height of the hub
LE	At the leading edge
shroud	At height of the shroud
TE	At the trailing edge

List of Figures

2.1	Different blade shapes interpolated to the same mesh	7
2.2	Hall's BFM illustration	9
2.3	Definition of effective pitch and blade pitch	11
2.4	Illustration of the parallel body-force component	12
3.1	Illustration of the ray-casting principle	14
3.2	SU2 BFM solver structure	16
3.3	BFM preprocessing workflow diagram	17
4.1	axial-radial slice of parallel force verification domain	21
4.2	Prismatic cross section used for metal blockage verification	22
4.3	3D mesh used for the blockage verification blade computation	23
4.4	Axial-tangential slice of blade computation domain	24
4.5	Axial-radial slice of blade computation domain	24
4.6	Axial-radial slice of BFM computation domain	24
4.7	Axial-tangential slice of BFM computation domain at mid-span	25
5.1	Aachen turbine flow domain	26
5.2	Aachen stator cross section	27
5.3	Aachen rotor cross section	27
5.4	Axial node count definition	28
5.5	Tangential node count definition	29
6.1	Density residual convergence trend	30
6.2	Comparison between entropy gradient and parallel force	31
6.3	Comparison between pressure gradient and parallel force	31
6.4	Comparison between pressure gradient and parallel force with inclusion of kinetic energy gradient	32
6.5	Comparison between interpolated and analytical blockage factor distribution	32
6.6	Comparison between interpolated and analytical blockage factor distribution	33
6.7	Mach number distribution in BFM results (radius=0.2875 m)	33
6.8	Mach number distribution in the blade computation results(radius=0.2875 m)	34
6.9	Mach number comparison between blade computation and BFM	34
6.10	Mass flux comparison between blade computation and BFM	35
6.11	Normalized static pressure comparison between blade computation and BFM	35
6.12	Deviation between BFM mass flux and blade computation mass flux trends	36
6.13	Convergence trend comparison between the different mesh densities	37
6.14	Total convergence time for each level of mesh refinement	37
6.15	Comparison of nomalized static pressure trends	38
6.16	Static pressure deviation	39
6.17	Comparison of nomalized total pressure trends	39
6.18	Stagnation pressure deviation	40
6.19	Comparison of nomalized total temperature trends	40
6.20	Stagnation temperature deviation	41
6.21	Comparison of absolute flow angle trends	41
6.22	Absolute flow angle deviation	42
6.23	Comparison of entropy trends	43
6.24	Comparison of normalized entropy trends	43
6.25	Comparison of mass flux trends	44

6.26	Mass flux deviation	44
6.27	Axial station definition	45
6.28	Radial static pressure trend between first stator and rotor	46
6.29	Radial static temperature trend between first stator and rotor	46
6.30	Radial static pressure trend between rotor and second stator	47
6.31	Radial static temperature trend between rotor and second stator	47
6.32	Radial absolute flow angle trend between the first stator and rotor	48
6.33	Radial absolute flow angle trend between rotor and second stator	48
6.34	Static pressure N_θ comparison	49
6.35	Static and stagnation pressure and temperature RMS errors	50
6.36	Mass flux RMS error trend	51
A.1	Mid-span rotor cross section	57
A.2	Mid-span stator cross section	57
A.3	Isometric view of unstructured mesh	58
A.4	Side view of unstructured mesh	58
A.5	Axial, mass-flux-averaged absolute flow angle trend	59
A.6	Axial, mass-flux-averaged entropy increase trend	59
A.7	Axial, mass-flux-averaged stagnation pressure trend	59
A.8	Axial, mass-flux-averaged stagnation temperature trend	59
B.1	Radial absolute flow angle near the inlet	60
B.2	Radial absolute flow angle near the first blade leading edge	60
B.3	Radial absolute flow angle in the first row gap	61
B.4	Radial absolute flow angle in the second row gap	61
B.5	Radial absolute flow angle near the last blade trailing edge	62
B.6	Radial absolute flow angle near the outlet	62

List of Tables

4.1	Parallel force verification boundary conditions	21
4.2	Geometric parameters of blade geometry used for blockage verification	22
4.3	Boundary conditions used for the blockage verification simulations	25
5.1	Aachen blade computation boundary conditions	27
5.2	Aachen blade computation direct solver settings	27
6.1	Tabulated total computation time for each level of mesh refinement	38
A.1	Design variable selection for demonstration compressor case	56
A.2	<i>Parabla</i> thickness parameters	56
A.3	<i>Meangen</i> blade geometry output	57
A.4	Rotor row <i>Meangen</i> output	57
A.5	Stator row <i>Meangen</i> output	57

1 Introduction

1.1 Motivation and Research Objective

Climate change is real and global carbon emissions are one of the main causes for this crisis. This encourages transport manufacturers to rely less on hydrocarbon fuels. The aircraft industry is no exception to this change. Until electric or hydrogen powered aircraft are feasible, engineers have to come up with ways to reduce the fuel consumption of future aircraft. One of the most effective ways to achieve this is to reduce the Thrust Specific Fuel Consumption(TSFC) of aircraft propulsors. For this, there are several options and one is to make use of Boundary Layer Ingestion(BLI). Here, the engine is placed behind or close to the fuselage or wing, which allows for the thrust jet to directly energize the low-momentum wake of the aircraft. This can lead to TSFC reductions of around 10% [3, 4]. A drawback of using this engine configuration is that the air flowing into the engine fan is distorted due to the fuselage boundary layer. This distortion decreases the efficiency of the fan and causes flow distortions downstream of the fan, which can have negative consequences on downstream components. Current high-fidelity analysis methodologies are capable of resolving these distortions, but are unfeasible in a design context due to the high associated computational costs [5]. This makes numerical design optimization impractical using conventional methods, as subsequent design iterations would require vast computational resources and time. Alternative methods would have to be utilized which require less computational resources, while maintaining a competent level of accuracy. One analysis method which has high potential in this field is the Body-Force Model (BFM).

A BFM works by replacing the blade rows by a volumetric force field, which locally models the flow turning and losses, while requiring several orders of magnitude less computation time [6] and maintaining reasonable accuracy. Various BFM's have been developed over the course of more than 50 years [7] and have been used successfully for off-design BLI engine analysis and parameter studies [8]. What has not been tested yet, is the viability of this method in a design optimization context. Studies have proven that BFM's can approach RANS and URANS simulation results for a reasonable degree on a single design point [9, 2, 10], but it has not been developed for and applied in a design context. Such a method which is capable of producing results quickly and with relative accuracy would be a powerful tool in the optimization of (fan) blades.

The objective of this research project is to set up a workflow process which allows for effective turbomachinery analyses using body-force modeling in SU2. In order to reach this objective, a set of research questions had to be answered.

1. **What level of accuracy in terms of loss generation and flow obstruction modeling can be achieved through implementation of respective models?** This main research question was posed for verification purposes. As the existing BFM formulation in SU2 was being extended in this research, obtaining a measure of accuracy of the additions to the BFM was crucial. Subsequently, errors during later validation could be attributed to specific parts of the BFM if they were to be of equal order of magnitude. The sub-questions of this main research question delve into the specific additions to the BFM.
 - (a) **With what level of accuracy does the 2D interpolation method interpolate the blade shape to the mesh?** The 2D interpolator is the first part of the BFM accessed by the solver and projects the body-force parameters to the mesh. The accuracy of this functionality is therefore paramount.

- (b) **With what level of accuracy does the loss generation method approximate entropy generation with respect to theoretical values?** The parallel force addition to the BFM was put in place to generate losses due to friction. This model had to be implemented correctly in order to generate these losses appropriately.
 - (c) **With what level of accuracy can flow obstruction due to blade thickness be simulated using metal blockage modeling?** In order to assess the flow obstruction due to blade thickness, a metal blockage model was implemented. Before applying the BFM to a case study, the implementation and accuracy of this model had to be verified.
2. **What are the pros and cons of body-force modeling in SU2 with respect to physical blade computations?** This question regards the effectiveness of the implemented BFM formulation in a design context. Here, by “effectiveness” a combination of computational efficiency and relative accuracy is meant. These results were obtained from a predefined simulation case. The sub-questions of this main research question go more in depth regarding the computational efficiency, relative accuracy of the BFM results and the grid refinement level required for grid convergence on the flow domain.
- (a) **What level of computational efficiency can be achieved through body-force modeling in SU2 compared to blade computations?** This question was posed in order to quantify the computational costs of the BFM of the set test case compared to those associated with the physical blade computation.
 - (b) **With what level of accuracy do the BFM results approximate those obtained through blade computation?** The BFM was expected to be substantially more efficient in terms of computational requirements compared to the blade computation, but be penalized in terms of absolute accuracy. This question delves into the accuracy by which the BFM is able to reproduce the trends of several relevant flow variables.
 - (c) **What level of mesh refinement is required for the BFM to reach grid convergence?** One of the main advantages of using a BFM instead of a physical blade computation for analysis is the relative low mesh density associated with body-force modeling. This question was asked to pursue the answer to what level of grid refinement was required until the results no longer experience any visible change and the BFM therefore produces reliable results.

1.2 Original Contributions

The original contributions of this research work can be summarized as follows:

- **3D Euler BFM in SU2:** The BFM formulation in SU2 was expanded to be viable for design applications of turbomachinery components. The method allows for arbitrary geometries to be analyzed using single-zone structured or unstructured meshes. The fidelity of the method was also increased through addition of a model for flow obstruction due to metal blockage and a model simulating boundary-layer losses. The respective formulations of these models was taken from literature, but were never before implemented into SU2.
- **Open-source automated design workflow:** Body-force modeling has been applied in analyzing fan performance in BLI conditions, but not in a design context. During the current research, an automated workflow process was set up, which allows for the parameterization of axial turbomachinery geometries and performance estimation using body-force modeling. The user can specify a set of design variables, which are translated

into input parameters for the BFM and an unstructured mesh. All software used in the workflow method is open-source and the code was written in Python.

1.3 Outline

Some background knowledge on BFM's and their functionality can be found in Chapter 2. In this chapter, the Navier-Stokes equations are derived for the application of body-force modeling and the BFM formulation used in this research is given. Additionally, the original and current implementation of the BFM formulation in SU2 are discussed. The methodologies used regarding the BFM implementation in SU2 and the automated turbomachinery design workflow are outlined in Chapter 3. In Chapter 4, the numerical experiments used for verification of the metal blockage model and loss generation model are described. The BFM capabilities were assessed using a case study, where the BFM results and computational efficiency were compared to those of a RANS simulation of the same geometry. This case study was performed on a selection of meshes with varying levels of cell density in order to explore the scaling of accuracy and computational efficiency. The set-up used for these numerical experiments are given in Chapter 5. The data resulting from the experiments outlined in Chapter 4 and Chapter 5 are given in Chapter 6. These data are partially commented upon, while the rest of the data after postprocessing can be found in the various appendices. Based on these results, answers are given on the research questions in Chapter 7 and the validity of these results is discussed in Chapter 8. In the same chapter, recommendations for future research and improvements on the current research are listed.

2 Background Knowledge

Body-force modeling is a throughflow method for turbomachinery applications with a relatively high computational efficiency compared to current high-fidelity methods. This chapter provides some fundamental knowledge on the working principles of body-force modeling, as well as the current applications, benefits and limitations of the method. In Section 2.1, a general description of body-force modeling is given, which explains the fundamentals behind the method and how it compares to more conventional methods. Section 2.2 provides some insight on how body-force modeling came to be and some of the current applications and limitations of the method. Finally, a description of the body-force model implemented at the start of this research is given in Section 2.3.

2.1 Body-force modeling

In conventional turbomachinery simulations, a set of flow equations is being resolved around the blades, whereas the interface between rotor and stator rows is often represented by a mixing plane, especially in the context of design optimization. Throughout this report, this type of simulation is referred to as ‘*blade computation*’ or ‘*physical blade computation*’. The mesh used to discretize the computational domain for blade computations requires a high level of refinement near the blade surfaces and endwalls in order to resolve boundary layers. Due to the fact that the flow in turbomachinery is predominantly turbulent, the Navier-Stokes(NS) equations with turbulent source terms have to be employed, either in RANS or URANS form in order to account for turbulent phenomena. The high mesh density and complex flow solver make turbomachinery CFD analyses relatively expensive to perform. A body-force model(BFM) does not require mesh densities of such magnitude and, depending on the formulation and analysis case, operate using more simple flow solvers, resulting in a significantly shorter computational time.

There are several reasons as to why computation times associated with body-force modeling are lower compared to blade computations. In body-force modeling, the physical blade rows are replaced by a volumetric force field. Within this field, momentum and energy source terms act forces on the flow, which turn the flow according to the local blade shape and flow conditions. The computational grid requires a low mesh densities at grid convergence, due to the lack of boundary layers to be resolved around the blade surface. Another aspect of BFMs is that the presence of the blades is ‘smeared out’ over the annulus, which theoretically allows for the mesh to be only 1 cell wide in pitch-wise direction. The lack of boundary layer refinements and single-cell mesh width result in substantially lower cell counts to mesh an equivalent domain in body-force modeling compared to blade computations. Additionally, a BFM does not require a mixing plane to distinguish rotating blade rows from stator rows. This increases the computational efficiency and eliminates the associated artificial mixing losses.

There are alternatives to body-force modeling when computational efficiency is desired. An example of a popular low-fidelity method used to predict blade disc performance is the actuator disc method (ADM). This method allows for the prediction of flow turning and sometimes stagnation pressure losses across a blade row using low computational resources. The BFM and actuator disc model differ mainly in terms of fidelity. The ADM turns the flow instantaneously upon entering the disc, based on the blade angle of attack and linearized lift curve slope. This makes the analysis rather crude and requires calibration data from either experiments or simulations to operate properly. The BFM turns the flow according to the local blade shape, using interpolated data on the blade geometry. Since the BFM uses a finite-volume method to compute the flow field, phenomena such as flow separation, flow obstruction and secondary flows can

be accounted for, which is not the case in ADMs. BFM are computationally more expensive than ADMs, but they allow for more complex blade shapes to be analyzed and depending on the BFM formulation, don't require any calibration data.

How does a BFM work mathematically? The BFM models the presence of the blades through the introduction of source terms in the governing equations which dictate the flow physics. Within the bladed sections of the domain, the BFM computes the magnitude and direction of the body-forces on each node. The body-forces turn the flow by acting as momentum source terms. The unsteady equation for conservation of mass, flow momentum and energy in continuum, adiabatic flow conditions are given below.

$$\frac{\partial}{\partial t} \iiint_V \rho dV + \oint_S \rho \mathbf{u} \cdot d\mathbf{S} = 0 \quad (2.1)$$

$$\frac{\partial}{\partial t} \iiint_V \rho \mathbf{u} dV + \oint_S (\rho \mathbf{u} \cdot d\mathbf{S}) \mathbf{u} + \oint_S p d\mathbf{S} = \iiint_V \mathbf{F}_{\text{BFM}} dV + \mathbf{F}_{\text{viscous}} \quad (2.2)$$

$$\frac{\partial}{\partial t} \iiint_V \rho \left(e + \frac{|\mathbf{u}|^2}{2} \right) dV + \oint_S \rho \left(e + \frac{|\mathbf{u}|^2}{2} \right) \mathbf{u} \cdot d\mathbf{S} = \iiint_V \rho \dot{E}_{\text{BFM}} dV - \oint_S p \mathbf{u} \cdot d\mathbf{S} + \dot{W}_{\text{viscous}} \quad (2.3)$$

Here, Equation 2.1 is the unsteady continuity equation, Equation 2.2 the three-dimensional momentum equation and Equation 2.3 the unsteady, adiabatic energy equation without internal heat generation. V is the control volume and S the surface of the domain boundaries. \mathbf{u} is the three-dimensional absolute velocity vector and ρ and p the static fluid density and pressure respectively. The BFM acts as a source term in the momentum- and energy equation through \mathbf{F}_{BFM} and \dot{E}_{BFM} , where \mathbf{F}_{BFM} is the three-dimensional body-force vector and \dot{E}_{BFM} the energy source term according to the BFM formulation. In blade computations, these terms are usually equal to zero, unless gravity is applied as a body-force. When applying divergence theorem to Equation 2.1, Equation 2.2 and Equation 2.3, and setting the integrant to zero for an arbitrary control volume, the following set of governing equations were derived.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.4)$$

$$\frac{\partial (\rho u_i)}{\partial t} + \nabla \cdot (\rho u_i \mathbf{u}) + \frac{\partial p}{\partial x_i} = F_{\text{BFM},i} + F_{\text{viscous},i} \quad (2.5)$$

$$\frac{\partial}{\partial t} \left[\rho \left(e + \frac{|\mathbf{u}|^2}{2} \right) \right] + \nabla \cdot \left[\rho \left(e + \frac{|\mathbf{u}|^2}{2} \right) \mathbf{u} + p \mathbf{u} \right] = \rho \dot{E}_{\text{BFM}} + \dot{W}'_{\text{viscous}} \quad (2.6)$$

Here, $\dot{W}'_{\text{viscous}}$ is the heat addition due to viscous processes per unit volume. The first and second term on the left hand side of Equation 2.6 can be replaced by the change in stagnation energy and divergence in stagnation enthalpy respectively, resulting in the following formulation for the energy equation.

$$\frac{\partial}{\partial t} (\rho e_t) + \nabla \cdot (\rho h_t \mathbf{u}) = \rho \dot{E}_{\text{BFM}} + \dot{W}'_{\text{viscous}} \quad (2.7)$$

Equation 2.4, Equation 2.5 and Equation 2.7 are the well-established Navier-Stokes equations, which are resolved by the flow solver. By grouping the equations into vectors, the flow solver attempts to resolve the following set of equations.

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_t \end{bmatrix} + \frac{\partial}{\partial x_i} \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + \delta_{i1} p \\ \rho u_i u_2 + \delta_{i2} p \\ \rho u_i u_3 + \delta_{i3} p \\ \rho u_i h_t \end{bmatrix} = \begin{bmatrix} 0 \\ F_{\text{BFM},1} + F_{\text{viscous},1} \\ F_{\text{BFM},2} + F_{\text{viscous},2} \\ F_{\text{BFM},3} + F_{\text{viscous},3} \\ \rho \dot{E}_{\text{BFM}} + \dot{W}'_{\text{viscous}} \end{bmatrix} \quad (2.8)$$

In viscous flow solvers, $\mathbf{F}_{\text{viscous}}$ and \dot{W}_{viscous} are approximated using turbulence models. In turbomachinery design, these terms are often modeled using RANS turbulence models. Resolving these forces allows for the approximation of boundary layers and viscous mixing phenomena. In combination with a BFM, the flow turning by the blades is achieved through implementation of the source term F_{BFM} , while endwall boundary layers are still resolved through the viscous forces. The computation of the viscous forces by the solver significantly increases the overall computational costs. Therefore, if resolving the endwall boundary layers is not a priority, an inviscid flow solver can be used. Here, the inviscid BFM reads:

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_t \end{bmatrix} + \frac{\partial}{\partial x_i} \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + \delta_{i1} p \\ \rho u_i u_2 + \delta_{i2} p \\ \rho u_i u_3 + \delta_{i3} p \\ \rho u_i h_t \end{bmatrix} = \begin{bmatrix} 0 \\ F_{\text{BFM},1} \\ F_{\text{BFM},2} \\ F_{\text{BFM},2} \\ \rho \dot{E}_{\text{BFM}} \end{bmatrix} \quad (2.9)$$

The energy source term, \dot{E}_{BFM} depends on the BFM formulation. In this research, the following expression was used:

$$\rho \dot{E}_{\text{BFM}} = r \Omega F_{\text{BFM},\theta} \quad (2.10)$$

Here, $F_{\text{BFM},\theta}$, r and Ω are the body-force vector component in tangential direction, the local radius and rotation rate respectively. This translates to the energy source term being zero in stator rows. This is consistent, as the overall stagnation energy remains unchanged throughout stationary blade rows. In case of a rotor blade row, stagnation energy is added or subtracted, depending on the orientation of the tangential force with respect to the direction of rotation. If the tangential body-force and rotation are aligned, stagnation enthalpy increases. This is the case for compressors, while the opposite is true for turbines.

The BFM essentially consists of two parts: the numerical flow solver and the BFM formulation. These two algorithms pass information over to one another. During the solving process, the numerical flow solver attempts to resolve the governing equations as given in Equation 2.8 or Equation 2.9. During each iteration, the values of the absolute velocity, pressure and temperature are calculated for each node in the domain, after which the body-force values are calculated by the BFM, which are subsequently stored on each node. The solver then adds the body-forces as source terms to the set of governing equations and resolves them prior to proceeding to the next iteration.

Up to now, the interaction between the BFM and the numerical flow solver has been discussed. This does not explain the dependency of the body-forces on the shape of the blade. The magnitude and direction of the body-force vector depends on the formulation of the BFM, the local flow conditions as calculated by the flow solver and the local blade shape. The BFM therefore requires information on the orientation of the blade surface or camber line with respect to the flow. In blade computations, this information is provided by the mesh and boundary conditions. In body-force modeling, this information is stored onto each node, usually as a normal or tangent unit vector of the blade camber line. By assigning a set of blade shape parameters through interpolation to each node prior to solving, the mesh becomes independent of the blade shape. A stem cell analogy can therefore be made for the nodes within the domain, where each node can be part of a blade row or not, depending on the blade geometry. The properties of each node depend on a virtual blade shape which is interpolated onto the mesh nodes. This eliminates the need of the definition of rotating mesh zones, as the energy source term will only be dependent on the value for Ω assigned to the local mesh node. Therefore, different blades can be analyzed on the same mesh by storing geometric parameters on different

nodes. For example, in Figure 2.1, two different meridional geometries are shown on the same mesh.

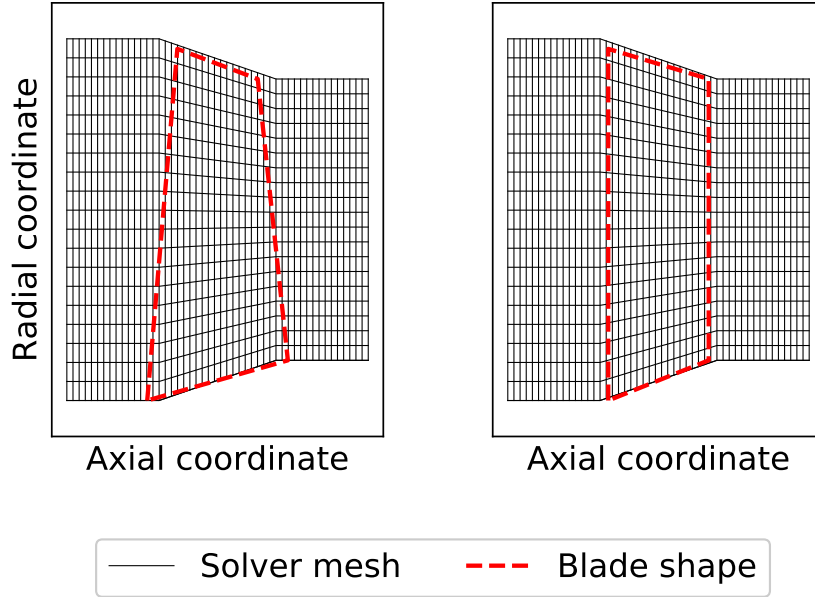


Figure 2.1: Different blade shapes interpolated to the same mesh

Within the red, dotted border, information regarding the local blade shape is interpolated onto the respective nodes. In case of a rotor row, a rotation rate is assigned to the nodes within the bladed region. The BFM calculates body-forces for every node in the domain, but for the nodes outside of the bladed region these body-forces are multiplied by zero.

2.2 Applications and limitations of body-force modeling

The main benefit body-force modeling offers over conventional methods is its relative computational efficiency. This computational efficiency comes at a cost of lower fidelity, where the accuracy depends on the BFM formulation and calibration of its governing equations. Whether a BFM can be effectively used depends on the objective of the analysis and the application. This section describes some of the cases in which body-force modeling has been applied successfully, as well as some limitations of the method.

BFMs are not new. The original idea was developed by Marble in the 1960s[7] and many variations exist today. Originally, the method was developed as an extension of radial equilibrium theory, allowing for interesting flow phenomena to be resolved throughout the blade row, while limiting the computational costs. This was done with the intent to allow designers to perform flow simulations on turbomachinery geometries in a time when blade computations were impractical for this purpose due to their high computation costs.

This intended purpose fell to the wayside due to technological advancements. The increasing capability of modern computers allowed for blade shapes to be optimized using physical blade computations, which in a way undermined the original purpose of the BFM by Marble. However, as modern, binary computers are approaching their processing limits set by physics, the BFM might be employed in its intended way. In recent times, body-force modeling has sparked interest among aircraft manufacturers. One of the reasons for this is the move towards synergizing the propulsion system with the airframe, where the exhaust jet from the propulsor is

located inside or close to the wake of the aircraft fuselage. This has been shown to yield propulsive efficiency gains of up to 10% [3, 4, 2, 1], which is significant and can initiate a snowball effect in which total aircraft weight could be reduced substantially. Utilizing this phenomenon requires the propulsor intake to be placed in close proximity to the airframe surface, which causes the aircraft boundary layer to be ingested into the propulsor. This intake configuration is called boundary-layer ingestion (BLI) and has several drawbacks compared to a clean intake configurations. An intake fan using BLI experiences a distorted inflow. The airflow close to the airframe surface has a momentum deficiency compared to the free stream, which causes the fan blades to experience varying intake conditions throughout each revolution. Such conditions can cause downstream flow distortion, fan efficiency losses and flutter of the fan blades. In order to cope with these phenomena, the fan and subsequent stator blades would have to be designed accordingly, which requires full-annulus analyses of the flow field throughout the fan and transient solutions to account for unsteady phenomena. Physical blade computations of such analysis cases can take weeks to converge using hundreds of processors [5]. While designing propulsor intake fans, the design variable count can be numerous, requiring thousands of design iterations to be performed during conceptual design optimization. This would not be feasible using physical blade computations, even when utilizing advanced techniques such as adjoint solving. This is where body-force modeling could be applied in the near future, as the method allows for comparatively cheap analyses, while maintaining the fidelity required for conceptual design. Currently, BFM have mostly been applied in the (off-)design analysis of BLI fans on single design points and parameter studies [9, 2, 10], in which the effects of altering certain design parameters on BLI performance were quantified.

Although body-force modeling is a promising BLI design methodology, it does have some limitations with respect to physical blade computations. Over the years, many different BFM formulations have been developed, each with their own range of applications. Many BFMs require calibration from blade computation data, which partially negates the computational benefit body-force modeling offers.

The fact that a BFM ‘smears out’ the blade geometry over the annulus has the benefit of allowing for single cell wide meshes for axisymmetric simulations, but causes some relevant phenomena to not be resolved. In transonic blade rows, shock wave interaction between blades can decrease efficiency and cause damage to the blades. These interactions cannot be modeled through BFM.

Most BFM formulations lack a way to model blade metal blockage. This results in the mass flux to be overestimated by the BFM and the effects of varying blade thickness throughout the blade row to not be modeled. There are BFM formulations which are capable of accounting for metal blockage to an extent. One formulation which is capable of doing this was developed by Thollet [2] and is discussed in more detail in Section 2.4.

2.3 Original BFM implementation in SU2

In 2019, a BFM formulation based on Hall’s BFM formulation was implemented in SU2 [6]. SU2 (Stanford University Unstructured) is an open-source CFD solver platform developed in collaboration between Stanford University and Delft University of Technology, among many other institutes. The results of this implementation [6] demonstrated that SU2 was capable of body-force modeling and a possible future coupling with the adjoint solver in SU2. The BFM formulation implemented was based on Hall’s original BFM [1], which was developed for BLI flow distortion analysis purposes. This BFM formulation was limited to only modeling flow turning, while lacking terms for loss generation or flow obstruction due to blade thickness. The reason

for leaving these phenomena unresolved was their perceived lack of influence on downstream flow distortion.

The body-force model developed by Hall is one of the simplistic BFM formulations, yet being very efficient and maintaining reasonable accuracy regarding flow deflection. The body-forces acting on the flow consisted of a single vector normal to the flow. The mathematical formulation of this normal force is

$$F_n = \rho\pi W^2 \delta \frac{1}{s|n_\theta|} \quad (2.11)$$

Here, W , δ and s are the relative velocity magnitude, local angle of attack and blade pitch respectively. n_θ is the camber normal vector component in tangential direction. The local angle of attack was calculated through

$$\delta = \sin^{-1} \left(\frac{\mathbf{n} \cdot \mathbf{w}}{W} \right) \quad (2.12)$$

where \mathbf{n} is the local camber normal vector and \mathbf{w} the relative velocity vector. For a stator blade row, W and \mathbf{w} are replaced with the absolute velocity magnitude and absolute velocity vector respectively. This force acts normal to the blade surface and relative flow, which defines its direction in Cartesian space. An illustration of the geometric terms in Equation 2.11 can be seen in Figure 2.2.

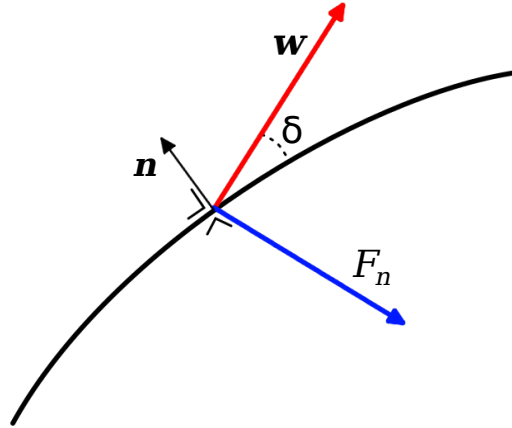


Figure 2.2: Hall's BFM illustration

The black, curved line is the camber line of the local blade cross section. The red arrow is the airflow, which has an angle of attack of value δ with respect to the camber line. By calculating δ according to Equation 2.12, the body-force normal to the airflow, F_n was calculated. This body-force is visualized in Figure 2.2 as the blue arrow. The normal force is oriented in such a way that it turns the flow towards the camber line. Through algebraic transformation, this normal body-force could be defined in Cartesian space, from which the respective components could be substituted into Equation 2.9.

Hall's BFM is simple in terms of formulation and implementation, but lacks the fidelity required for applications in turbomachinery design. For instance, the effects of blade thickness could not be modeled and no losses were generated. This would lead to over-estimations in mass flow rate and efficiency. In 2017, additions were made to this BFM formulation[2], which allowed for

metal blockage and loss modeling, as well as the effects of high Mach numbers on flow turning. Throughout this report, this BFM formulation will be referred to as ‘Thollet’s BFM’, named after the scientist who originally implemented the blockage and loss models[2]. A 2D version of this BFM formulation was implemented into SU2 without the loss generation and metal blockage model[6]. The normal force was formulated according to

$$F_n = K_{\text{Mach}} \rho \pi W^2 \delta \frac{1}{s |n_\theta|} \quad (2.13)$$

where

$$K_{\text{Mach}} = \begin{cases} K' & , K' \leq 3 \\ 3 & , K' > 3 \end{cases} \quad (2.14)$$

$$K' = \begin{cases} \frac{1}{\sqrt{1-M_{\text{rel}}^2}} & , M_{\text{rel}} < 1 \\ \frac{2}{\pi \sqrt{M_{\text{rel}}^2-1}} & , M_{\text{rel}} > 1 \end{cases}$$

Here, M_{rel} is the relative, local Mach number, which in case of a stator blade row is the absolute Mach number. This factor approaches 1.0 for low Mach numbers, reaches a ceiling of 3.0 near sonic conditions and then decreases again in supersonic flow.

The applicability of this BFM formulation and implementation was limited, as the blade shape parameters had to be hard-coded into the solver source code. Information on the blade geometry could therefore not be provided externally and required compilation of the solver code when analyzing a different blade geometry. This made the method in its current state ill-suited for design applications in its current state, as the computation time associated with a single geometry analysis would include the time required for the solver source code to compile.

2.4 Thollet’s BFM

The method described in Section 2.3 was based on Thollet’s method, but lacked the metal blockage source term and loss generation. In this section, these sub-models are elaborated upon. In blade computations, the physical presence of the blades forces the fluid to turn and acts as an obstruction in the channel. This obstruction is caused by the non-zero thickness of the blade, which results in fluid acceleration at locations of varying thickness. Apart from flow acceleration, this obstruction reduces the flow rate throughout the blade row. Due to the physical blades not being present in body-force modeling, these phenomena are not modeled. The governing equations for continuum flows, given in Equation 2.1, Equation 2.2 and Equation 2.3 included the source terms for momentum and energy present in body-force modeling, but lack any source term associated with metal blockage. A geometric parameter was added to Hall’s BFM, which will be referred to as ‘blockage factor’, which was added by Thollet[2] and is defined as

$$b = \frac{s'}{s} \quad (2.15)$$

Here, b is the metal blockage factor, s' the local, effective pitch and s the actual blade pitch. The blade pitch is equal to

$$s = \frac{2\pi r}{N_b} \quad (2.16)$$

where r is the local radius and N_b the number of blades in the blade row. The effective pitch, s' is essentially the blade pitch minus the local, tangential blade thickness. This is illustrated in Figure 2.3 for a 2D cascade.

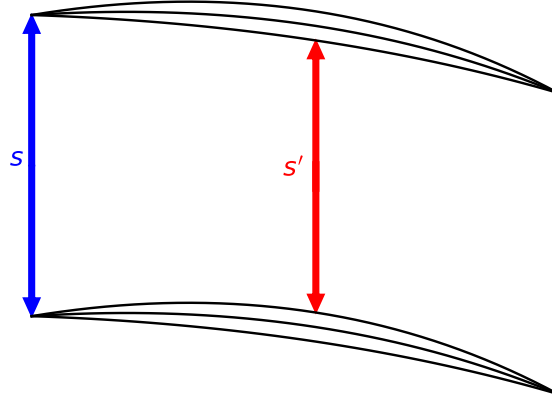


Figure 2.3: Definition of effective pitch and blade pitch

The ratio between the local, effective pitch and blade pitch translates to the available throughflow area and the area of the annulus. If the local blade thickness is high, the blockage factor reduces, while outside of the blade row, it attains a value of 1.0. This makes the naming ‘blockage factor’ to be a bit misleading, as low values translate to high flow obstruction and vice versa. The source terms implemented by Thollet to account for flow obstruction were based on the gradient in blockage factor. If the blockage factor decreases along the flow direction, flow rate is added, which results in the flow accelerating in sections of increasing blade thickness. If the blockage factor increases along the flow direction, the flow decelerates. This source term was added to the right hand side of the steady form of Equation 2.9, resulting in the following BFM formulation:

$$\frac{\partial}{\partial x_i} \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + \delta_{i1} p \\ \rho u_i u_2 + \delta_{i2} p \\ \rho u_i u_3 + \delta_{i3} p \\ \rho u_i h_t \end{bmatrix} = \begin{bmatrix} 0 \\ F_1 \\ F_2 \\ F_3 \\ F_{\theta} r \Omega \end{bmatrix} - \frac{1}{b} \begin{bmatrix} \rho u_i \frac{\partial b}{\partial x_i} \\ \rho u_1 u_i \frac{\partial b}{\partial x_i} \\ \rho u_2 u_i \frac{\partial b}{\partial x_i} \\ \rho u_3 u_i \frac{\partial b}{\partial x_i} \\ \rho h_t u_i \frac{\partial b}{\partial x_i} \end{bmatrix} \quad (2.17)$$

The set of equations in Equation 2.17 can be solved using standard, time-marching techniques used in blade computations. This formulation proved to be able to capture flow acceleration due to blade thickness very well, but requires the distribution of the metal blockage factor throughout the bladed regions to work.

BFM formulations such as Hall’s BFM lack any form of loss generation. In physical blade rows, most losses are generated through boundary layers, shocks and endwall losses. The respective magnitude of these components depend on the blade geometry and flow regime. These phenomena can be simulated in blade computations, as the physical boundary of the blade surface allows for boundary layers to form and shocks to impinge upon. In body-force modeling, only the endwalls could generate such phenomena, but as the physical blade surface is absent, losses would be underestimated by the BFM. In order to cope with this, an additional body-force component was added to the BFM. This component acts parallel to the flow, in opposite direction. This force component can be seen in relation to the normal body-force in Figure 2.4.

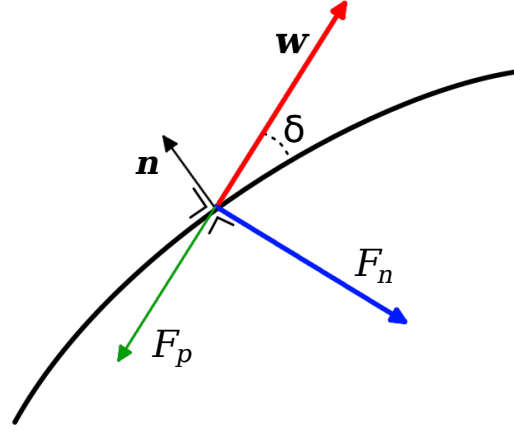


Figure 2.4: Illustration of the parallel body-force component

In the earliest document on body-force modeling[7], a relation between the magnitude of the body-force component parallel to the flow and entropy generation was derived. This derivation can be found in Chapter 3. In order for a BFM to generate losses, a parallel body-force component was required according to this derivation. The formulation for the normal and parallel force used by Thollet were

$$F_n = K_{\text{Mach}} \rho \pi W^2 \delta \frac{1}{sb|n_\theta|} \quad F_p = \rho C_f W^2 \frac{1}{sb|n_\theta|} \quad (2.18)$$

where

$$C_f = \frac{0.0592}{Re_x^{-0.2}} \quad Re_x = \frac{\rho W(x - x_{LE})}{\mu} \quad (2.19)$$

The parallel force formulation was based on a simple, flat-plate boundary layer loss model, where C_f was used as a friction factor. The coefficient of 0.0592 was found through calibration on reference data. $x - x_{LE}$ is the axial distance between the leading edge and the local coordinate and μ is the fluid kinematic viscosity. In stator blade rows, the relative velocity magnitude, W is replaced by the absolute velocity magnitude. Both the parallel and normal force magnitudes were divided by the blockage factor, which translates to the body-force magnitude increasing near the thicker regions of the blade. By projecting the normal and parallel body-forces into 3D, Cartesian space, the body-forces can be substituted into the first right-hand side term in Equation 2.17.

3 Research Methodology

In this chapter, the methodologies used to obtain the answers to the research questions are discussed. In order to answer the research questions and solve the problems posed in Chapter 1, additions had to be made to the existing BFM in SU2 and these additions had to be verified through numerical experiments. The BFM was to be expanded to 3D, assess the effects of blade blockage on the flow and loss generation. Formulations for these expansions were found in Thollet's BFM, which were described in Chapter 2. In order for the BFM and metal blockage model to function, the camber normal vector and blockage factor distribution associated with the blade geometry had to be interpolated to the mesh nodes. This was achieved through implementation of a 2D interpolation algorithm into SU2. The working principles behind this method are explained in Section 3.1. The normal and parallel force generated by the BFM had to be transformed to 3D, Cartesian components in order to be substituted into the body-force source term vector. This transformation process is outlined in Section 3.2. The additions to the existing BFM formulation in SU2 had to be implemented in a numerically efficient manner. The integration of the BFM formulation into the SU2 solver structure is described in Section 3.3. In order to allow for the analysis of parameterized turbomachinery geometries using body-force modeling, an efficient and automated workflow process had to be set up. This would allow the user to specify a set of design variables, defining one or multiple blade row geometries. The geometric parameters required for the BFM to function, as well as the mesh would have to be calculated according to this geometry prior to analysis using the BFM. This workflow process is explained in Section 3.4.

3.1 Ray-Cast Interpolation

The BFM requires the camber normal vector components and blockage factor to generate body-forces and the metal blockage source terms. Since these terms are independent of the mesh geometry and vice-versa, they had to be interpolated onto the mesh coordinates and stored on each node. The 2D BFM implemented by M.Latour used linear interpolation for this. This was possible, as the geometric parameters only varied in axial direction, allowing for 1D interpolation. For a 3D BFM, this method is insufficient, as the geometric parameters vary in 2D (in axial and radial direction). A custom 2D interpolator was added to SU2, which made use of a mathematical concept called ray-casting.

What is ray-casting theory? Ray-casting theory is based on the concept that a vector drawn from any point outside a region to a location inside a region intersects an odd number of times with the boundary of that region. This is illustrated in Figure 3.1.

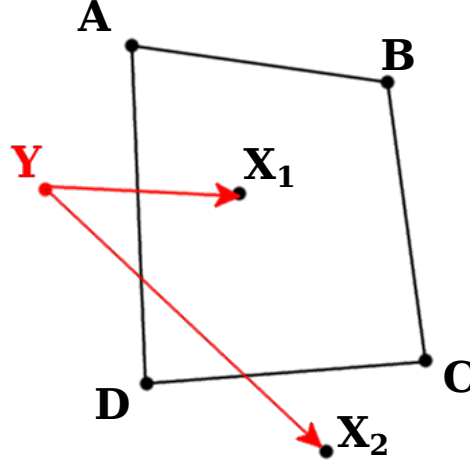


Figure 3.1: Illustration of the ray-casting principle

Here, the black box is the boundary of the region in question, point \mathbf{X}_1 is located inside and \mathbf{X}_2 is located outside of the region. By drawing a line from point \mathbf{Y} to point \mathbf{X}_1 , it intersects with the boundary once, while drawing a line from \mathbf{Y} to \mathbf{X}_2 results in two intersections. According to the theory, \mathbf{X}_1 is correctly located inside the region, while \mathbf{X}_2 is not.

This principle can be used to interpolate the geometric parameters of the blade to the mesh without determining separate zones in the mesh. For example, \mathbf{X}_1 and \mathbf{X}_2 are nodes in the mesh, while points A-D form the corners of the blade and \mathbf{Y} is the origin of the coordinate system. By counting the number of times a vector drawn from the origin intersects with the virtual blade boundary, it can be determined whether the node is inside of the blade region or not. If the node is located inside the region ABCD and geometric parameters are defined on points A, B, C and D, the interpolated parameters could be determined through the *distance-weighted-average* of the parameters at the corner points. By subdividing the blade region into quadrilaterals, calculating the camber normal components and blockage vectors on those points and providing this information to the solver through an input file, the interpolation algorithm could interpolate arbitrary blade shapes onto arbitrary meshes.

3.2 3D BFM transformation

Thollet's BFM generates a value for the normal and parallel body-force components, according to Equation 2.18. In order to transform these values into a source term vector in the governing equations, these body-forces had to be transformed to a force vector in 3D Cartesian space. In this section, the steps taken in this transformation are outlined.

The body-force components calculated according to Thollet's BFM were defined with respect to the local, relative flow vector and camber normal vector. This allowed for the body-forces to be transformed to cylindrical space, of which the transition to Cartesian coordinates is trivial. The first step in the transformation process was to define the relative velocity in cylindrical coordinates.

$$\mathbf{w}_{\text{cyl}} = \begin{bmatrix} w_{ax} \\ w_{\theta} \\ w_r \end{bmatrix} \quad (3.1)$$

By defining the rotation axis of the analysis case as a unit vector, the axial and radial coordinates could be defined at each node.

$$\mathbf{x}_{ax} = \frac{\mathbf{i}_\Omega \cdot \mathbf{x}}{\mathbf{i}_\Omega \cdot \mathbf{i}_\Omega} \mathbf{i}_\Omega \quad \mathbf{x}_{rad} = \mathbf{x} - \mathbf{x}_{ax} \quad (3.2)$$

Here, \mathbf{x}_{ax} is the parallel projection of the cell coordinate vector \mathbf{x} onto the rotation axis \mathbf{i}_Ω . \mathbf{x}_{rad} is the radial coordinate vector. Through normalization, these vectors could be transformed into unit vectors.

$$\mathbf{i}_{ax} = \frac{\mathbf{x}_{ax}}{|\mathbf{x}_{ax}|} \quad \mathbf{i}_{rad} = \frac{\mathbf{x}_{rad}}{|\mathbf{x}_{rad}|} \quad \mathbf{i}_\theta = \frac{\mathbf{i}_\Omega \times \mathbf{x}_{rad}}{|\mathbf{x}_{rad}|} \quad (3.3)$$

The unit vector in tangential direction, \mathbf{i}_θ was defined as the normalized cross product of the rotation axis and the radial coordinate. The absolute velocity vector in Cartesian space could now be defined in cylindrical space.

$$\mathbf{u}_{cyl} = \begin{bmatrix} u_{ax} \\ u_\theta \\ u_{rad} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{Cart} \cdot \mathbf{i}_{ax} \\ \mathbf{u}_{Cart} \cdot \mathbf{i}_\theta \\ \mathbf{u}_{Cart} \cdot \mathbf{i}_{rad} \end{bmatrix} \quad (3.4)$$

The relative velocity vector could now be calculated through subtraction of the induced velocity due to rotation from the tangential velocity component.

$$\mathbf{w}_{cyl} = \begin{bmatrix} w_{ax} \\ w_\theta \\ w_{rad} \end{bmatrix} = \begin{bmatrix} u_{ax} \\ u_\theta \\ u_{rad} \end{bmatrix} - \begin{bmatrix} 0 \\ \Omega |\mathbf{x}_{rad}| \\ 0 \end{bmatrix} \quad (3.5)$$

Here, w_{ax} , w_θ and w_{rad} are the relative velocity components in axial, radial and tangential direction respectively. In order to translate the normal and parallel body-forces to cylindrical space, the relative velocity parallel to the camber line was required.

$$\mathbf{w}_p = \mathbf{w}_{cyl} - \mathbf{w}_n \quad (3.6)$$

$$\mathbf{w}_n = \mathbf{w}_{cyl} \cdot \mathbf{n} \begin{bmatrix} n_{ax} \\ n_\theta \\ n_r \end{bmatrix} \quad (3.7)$$

Here, \mathbf{w}_p and \mathbf{w}_n are the relative velocity projection vectors parallel and normal to the camber line. With the relative velocity vector parallel to the camber line, the normal body-force could be transformed to cylindrical space. This transformation was trivial the parallel body-force, as it acts in opposite direction to the relative flow. By adding the normal and parallel body-force components in cylindrical space, the cylindrical body-force vector was calculated.

$$\mathbf{F}_{cyl} = \begin{bmatrix} F_{ax} \\ F_\theta \\ F_r \end{bmatrix} = F_n \left(-\cos(\delta) \mathbf{n} + \sin(\delta) \frac{\mathbf{w}_p}{|\mathbf{w}_p|} \right) - F_p \frac{\mathbf{w}_{cyl}}{|\mathbf{w}_{cyl}|} \quad (3.8)$$

Here, δ is the deviation angle between the relative flow and camber line, which is defined in Equation 2.12. Using the Cartesian unit vectors in cylindrical space, as defined in Equation 3.3, the cylindrical body-force vector could be defined in Cartesian space.

$$\mathbf{F}_{Cart} = F_{ax} \mathbf{i}_{ax} + F_\theta \mathbf{i}_\theta + F_r \mathbf{i}_{rad} \quad (3.9)$$

The resulting body-force vector could then be used as the momentum source term in Equation 2.17 and the tangential component of Equation 3.8 could be used to compute the respective energy equation source term.

3.3 BFM solver structure

In this section, the solver structure set up in SU2 to facilitate body-force modeling is described. The process is illustrated in Figure 3.2. The red blocks are processes which were already incorporated into SU2. The blue blocks were added or modified. The process starts with the geometry preprocessing in block 0, where SU2 defines the mesh boundaries, boundary conditions and cell connectivity. The coordinates of the nodes are then passed to the interpolator function in block 1.1. Here, the camber normal vectors and blockage factor values are interpolated to the node coordinates and stored for each node.

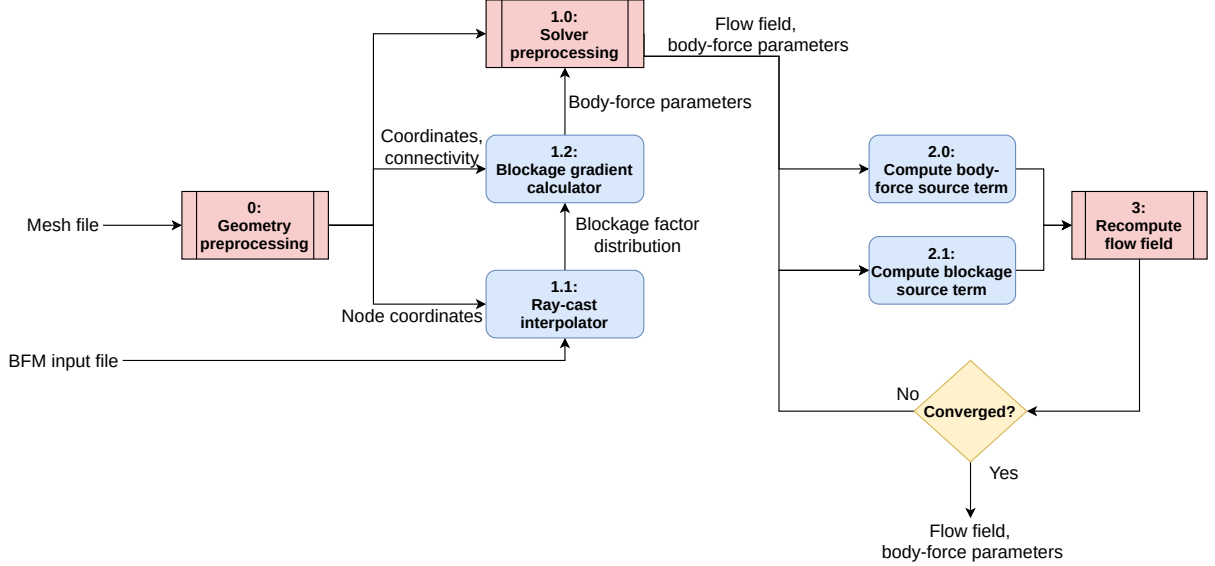


Figure 3.2: SU2 BFM solver structure

In order to calculate the metal blockage source term during the solution process, the metal blockage model requires the spatial gradients of the blockage factor. These are calculated in block 1.2, using the *finite-difference-method* used by the solver to calculate the gradients in flow variables, but using the interpolated blockage factor values as input.

After passing through blocks 1.1 and 1.2, all geometric parameters required for the BFM to calculate the body-forces were calculated and stored on each node. The actual solving process could initialize after block 1.0, where the body-force and blockage source terms are calculated and added to the governing equations. The flow field is recomputed and this process is repeated until convergence. The flow variables, body-forces and interpolated geometric parameters are included in the solver output for postprocessing.

3.4 Automated Workflow for BFM Computations

In order to prepare the BFM for design optimization in future studies, an automated workflow translating a set of design variables to SU2 input was required. In this section, the automated workflow process developed for this purpose is outlined.

Due to the complex shape of modern turbomachinery blades, a high number of design variables is often used for blade optimization. However, at a first-level conceptual stage, this is often reduced to a set of duty coefficients, of namely the flow coefficient(ϕ), work coefficient(ψ) and degree of reaction(R^*). These three duty coefficients allow for the definition of the blade inlet and outlet angles and therefore allow for the creation of a simplified geometry with a low number

of variables. The method put in place allowed for expansion to higher variable counts if desired.

The goal of this workflow process was to take these high-level design variables and translate them to a mesh and the inputs required for the BFM. This would result in a streamlined design process. Figure 3.3.

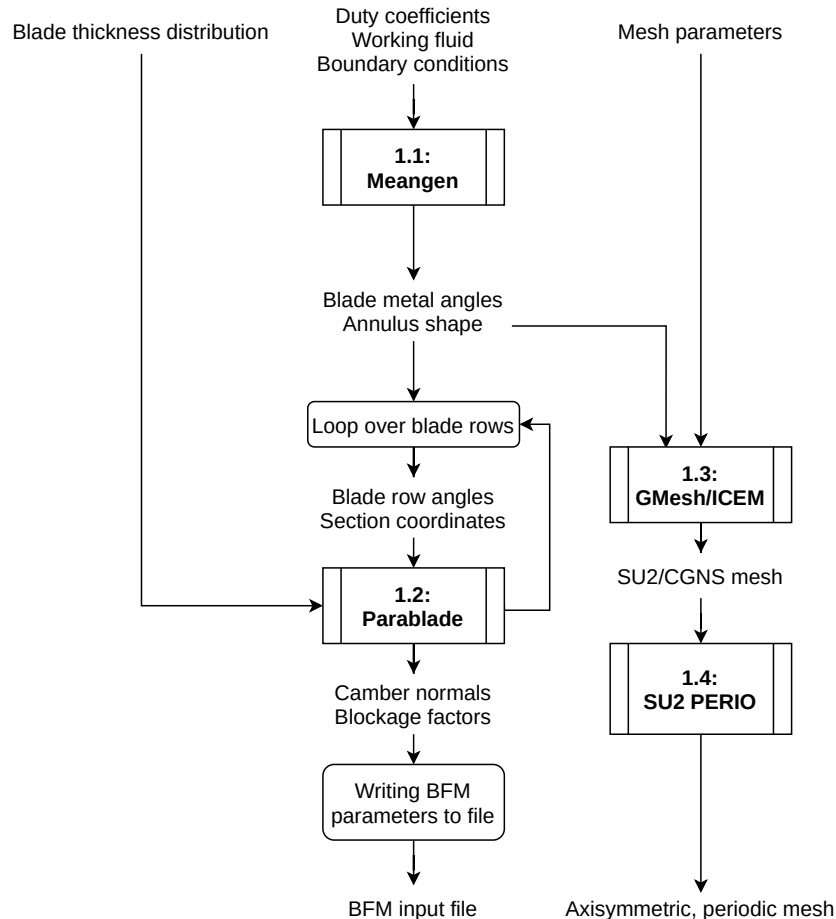


Figure 3.3: BFM preprocessing workflow diagram

The BFM preprocessing starts with the definition of design variables, working fluid parameters, boundary conditions, blade thickness distribution and mesh parameters. The process initializes with a program called *Meangen*. *Meangen* is able to convert high-level design variables into a simplified geometry, which includes the blade metal angles and meridional blade shape for a specified number of stages. *Meangen* can be used for axial, radial and mixed-flow machines, but in this workflow, only axial machines are considered. The BFM requires a distribution of camber normal and metal blockage data, for which a more detailed blade geometry is required. This detailed geometry is generated through a program called *Parablade*.

Parablade is a Python-based code developed within the Propulsion and Power group at TU Delft. It is used to generate detailed blade shapes which can be used to create meshes for turbomachinery analyses. *Parablade* allows the user to specify an almost arbitrary number of design variables, making it a powerful tool in shape optimization of turbomachinery blade rows. Additionally, the code is able to generate geometry sensitivity distributions based on the specified design variables, which allows for the computation of the design sensitivities in adjoint optimization. The *Parablade* code was modified to allow for BFM preprocessing as well. Here, the axial and radial distribution of camber normal vectors and blockage factor are calculated,

which can be used in SU2 for BFM computation.

After *Meangen* returns the crude blade geometry, *Parablade* is initialized for each consecutive blade row to produce the distribution of camber normals and blockage factors. It stores this information in a file which can be read by the 2D interpolation algorithm in SU2. The input parameters of *Parablade* include the blade metal angles and annulus shape from *Meangen* output and the blade thickness distribution defined by the user. *Meangen* also uses blade thickness parameters as input, but these could not be translated to the blade thickness parameterization used in *Parablade*, which is why these had to be defined during initialization. The blade parameterization in *Parablade* is based on Bezier-curves. The blade camber line is defined at several sections along the blade span. The blade surface is generated by a curve offset by half the local thickness, normal to the camber line. This camber normal vector distribution was used as input for the BFM in SU2. The blockage factor distribution was calculated through interpolation of the blade upper and lower surface and calculating the effective pitch trend in axial direction. By dividing these by the blade pitch, the blockage factor distribution could be defined. After looping through all the blade rows, the camber normal and blockage distribution of all blade rows were merged into a single input file which could be read by SU2.

With the annulus shape defined through *Meangen*, the flow domain could be discretized into a mesh. The BFM meshing procedure was relatively simple, as the detailed blade shape was not required. The only inputs required were the annulus shape produced in step 1.1 by *Meangen* and mesh size parameters provided by the user. Two variations in mesh generation were incorporated into the process. The program used for unstructured meshing was *GMesh*. *GMesh* is a meshing software which can be controlled through Python as a module to produce unstructured meshes for relatively simple geometries. The meshing parameters the user can provide are the number of cells in axial direction for each blade row, mesh density reduction factors for the inlet and outlet section of the mesh and the number of cells in pitch-wise direction. The structured meshing option was *ANSYS ICEM*. Here, the only mesh parameter which was used for this research project was the axial node count. Additional refinement was applied near the leading and trailing edges of the blade rows, while using exponential cell size relations were used in the far field.

Structured and unstructured meshes have respective benefits and drawbacks with respect to body-force modeling. The unstructured mesh is fully independent of the meridional blade shape, allowing for the mesh to be reusable given that the annulus shape remains constant. A drawback is that the solver process is less efficient and the total node count is not optimized for the geometry. Using a structured mesh allows the user to reduce the total number of nodes, as the cell aspect ratio allows for axial refinement, while keeping the number of cells in radial direction constant. Additionally, the interpolation onto the mesh nodes results in more smooth distributions and a better approximation of the gradient in blockage factor. The meshes generated by *ICEM* or *GMesh* can be read by SU2, but are not suitable for axisymmetric BFM analysis yet. SU2_PERIO is used to transform the mesh produced by *GMesh* or *ICEM* into a periodic mesh.

Upon finishing this research project, the unstructured *GMesh* procedure was fully automated, while that involving *ICEM* still required manual adjustment. In the near future, also this procedure is expected to be automated. This preprocessing workflow proved to be a valuable tool in performing BFM analyses as it was easy to use and very efficient. Even for multiple stages, the entire process takes less than one minute when defining an axisymmetric case. In the future, this tool could be expanded to include full-annulus meshing and design capabilities through adjoint-solving.

4 BFM Verification

Before comparing the performance of the BFM to the results of high-fidelity blade computations, its correct implementation had to be verified. In this section, the methods used to verify the implementation of the body-force model are described, in particular the metal blockage and parallel force additions to the exiting BFM formulation. The verification processes were carried out through result comparison of theoretical trends or blade computation results. In Section 4.1, the verification methodology of the parallel force addition is given, while that of the blockage source term addition and 2D interpolation can be found in Section 4.2.

4.1 Verification of Parallel Force Implementation

The parallel force addition to the BFM allows for entropy generation in the BFM simulation. This capability had to be verified to ensure its proper functionality and implementation. The experiment used for verifying the parallel force functionality was to compare the BFM output of a flat plate stator geometry to the theoretical solution.

The governing equations used for this simulation were the steady Euler equations, with the working fluid being air. In order to verify the parallel force implementation, the simulation entropy and pressure gradient results are compared to the theoretical trends they would have for a flat plate case. During the upcoming derivations, the following set of assumptions were applied.

- **Incompressible flow:** Free-stream Mach number is below 0.3, constant density, constant axial velocity.
- **Perfect gas:** Gas constant $R = 287.1 \text{ J kg}^{-1} \text{ K}^{-1}$, specific enthalpy $c_p = 1004.7 \text{ J kg}^{-1} \text{ K}^{-1}$, specific heat ratio $\gamma = 1.4$
- **Constant kinematic viscosity:** $\mu = 1.716 \times 10^{-5} \text{ kg m}^{-1} \text{ s}^{-2}$
- **Adiabatic flow:** No heat transfer between the geometry and the flow or the flow and the endwalls.
- **Parallel force only:** Stator angle is zero degrees, no flow deflection in tangential direction, camber normal components in axial and radial direction are equal to zero, $n_\theta = 1.0$.
- **Inviscid flow:** No turbulent source terms in the governing equations. The only viscous terms are introduced through the parallel force.
- **Axial, 1D, steady flow:** No unsteady terms in the governing equations, flow only in axial direction.
- **No metal blockage:** Blockage factor set to 1.0, blockage gradient throughout the flow field is zero.

Due to the inviscid assumption, the Euler equations were used for the derivation. The Euler momentum equation in x-direction is equal to

$$\frac{\partial(\rho u_x^2)}{\partial x} + \frac{\partial(\rho u_y u_x)}{\partial x} + \frac{\partial(\rho u_z u_x)}{\partial x} + \frac{\partial p}{\partial x} = F_x \quad (4.1)$$

Here, F_x is introduced by the BFM. Due to the flow being axial, all velocity components in y and z-direction are set to zero. Also, due to the flow being assumed incompressible, the density

was assumed constant. Finally, the combination of the flow being incompressible, the cross section being constant and the case being steady, the axial velocity component is constant. This simplifies the axial momentum equation to

$$\frac{d(\rho u_x^2)}{dx} + \frac{dp}{dx} = F_x \rightarrow \frac{dp}{dx} = F_x \quad (4.2)$$

Since only the parallel force acts on the flow in opposite direction to the velocity, the body-force can be rewritten as

$$F_x = -F_p \quad (4.3)$$

This would mean that the pressure gradient in axial direction would match the negative parallel force trend. If significant differences in the shape of the trends were to be found, it would mean that either the set of stated assumptions didn't apply or the parallel force implementation was faulty.

The parallel force was implemented to model entropy generation. The following derivations were carried out in order to predict the entropy change throughout the blade row. For the entropy change across any blade row, the following equation is considered.

$$Tds = dH - vdp = c_p dT - \frac{RT}{p} dp \quad (4.4)$$

Dividing by dx to get entropy gradient in axial direction:

$$T \frac{ds}{dx} = c_p \frac{dT}{dx} - \frac{RT}{p} \frac{dp}{dx} \quad (4.5)$$

The pressure gradient was already computed according to Equation 4.2. To get the temperature gradient, the body-force source term in the energy equation needs to be considered. The source term in the energy equation for the BFM is equal to $r\Omega F_\theta$. In case of the flat plate stator, this term is equal to zero, resulting in the temperature gradient to be zero.

$$\frac{\partial(\rho u_x h_t)}{\partial x} + \frac{\partial(\rho u_y h_t)}{\partial y} + \frac{\partial(\rho u_z h_t)}{\partial z} = \Omega r F_\theta \quad (4.6)$$

$$\rho u_x \frac{dh_t}{dx} = 0 \rightarrow \rho u_x \frac{d(c_p T)}{dx} = 0 \quad (4.7)$$

$$\rho u_x c_p \frac{dT}{dx} + \frac{\rho u_x}{2} \frac{d(u_x^2)}{dx} = 0 \quad (4.8)$$

Due to the flow being incompressible with a constant flow area, the velocity gradient is zero. This simplifies the temperature gradient to

$$\rho u_x c_p \frac{dT}{dx} = 0 \rightarrow \frac{dT}{dx} = 0 \quad (4.9)$$

Since the pressure gradient is unaffected by the change in energy source term, the entropy gradient equation becomes

$$T \frac{ds}{dx} = -\frac{1}{\rho} \frac{dp}{dx} = \frac{1}{\rho} F_p \quad (4.10)$$

By comparing the pressure and entropy gradient to the parallel force trend across the flow domain, the correct implementation of the parallel force can be verified. If the entropy gradient and negative pressure gradient equal the parallel force, the BFM source term was implemented correctly.

The flow simulation domain used for the verification of these equations can be seen in Figure 4.1. The mesh was set up to be a structured mesh, generated in ICEM. The body-forces acted in the red region, where the mesh was refined to ensure grid convergence. The mesh was also refined near the leading and trailing edges of the bladed region, which ensured a more smooth solution. The mesh was set one cell wide in pitch-wise direction, as the BFM theoretically doesn't require mesh refinement in that direction for axisymmetric simulations. Whether this is true was tested in a different experiment.

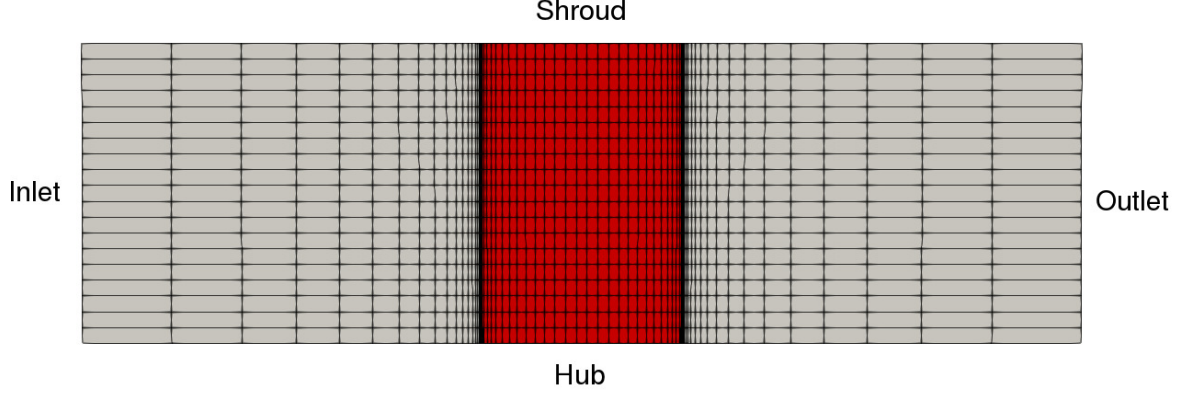


Figure 4.1: axial-radial slice of parallel force verification domain

The simulation used the following set of boundary conditions, as can be seen in Table 4.1. The total-to-static pressure ratio between the domain inlet and outlet was set to 1.04, which is lower than the sonic pressure ratio for air, which is about 1.8. With this pressure ratio, no compressible flow within the domain was expected.

Table 4.1: Parallel force verification boundary conditions

Boundary	Boundary conditions
Inlet	$p_t = 105\,325\text{ Pa}, T_t = 288.7\text{ K}$
Outlet	$p_s = 101\,325\text{ Pa}$
Hub, Shroud	slip-wall

After convergence of the solution, the values for entropy generation and static pressure trends were extracted from the solution flow field through averaging over the radial-tangential slice. The entropy and pressure gradients were obtained through finite-difference. As no significant radial and tangential flow components were expected, the parallel force magnitude could be assumed equal to the negative body-force magnitude in axial direction. With these data, Equation 4.2 and Equation 4.10 could be visualized, through which the parallel force implementation could be verified.

4.2 Metal Blockage Model Verification

The main goal of the metal blockage term implementation was to model the flow obstruction by the blades. Before analyzing the performance of the BFM, the functionality of the metal blockage implementation had to be validated. This was done through comparison of blade computation and BFM flow simulation data of a symmetric, prismatic stator geometry with a nonzero thickness distribution. This resulted in a blockage factor distribution within the blade section with values lower than 1 and a nonzero blockage gradient distribution. Especially by

comparing the mass flux trends in axial direction, the flow obstruction capability of the metal blockage model could be assessed. In Figure 4.2, the cross section of the blade can be seen. The thickness distribution was chosen to be parabolic for simplicity and also to test the accuracy of the interpolator, as the blockage factor distribution can be expressed analytically.

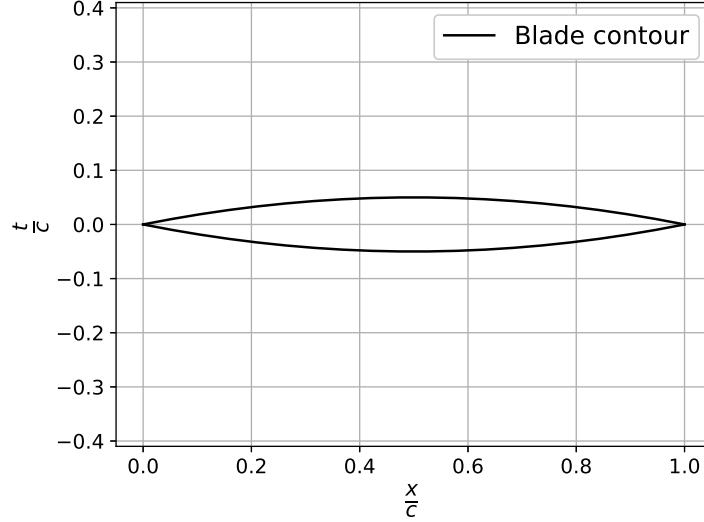


Figure 4.2: Prismatic cross section used for metal blockage verification

The validation run was performed in 3D through an axisymmetric turbomachinery simulation. In Table 4.2, the dimensions of the blade geometry can be found. Here, t_{\max} is the maximum thickness of the blade, c the axial chord and N_{blade} the number of blades in the blade row. For simplicity, no hub or tip gaps were applied.

Table 4.2: Geometric parameters of blade geometry used for blockage verification

Parameter	Dimension
r_{hub}	0.25 m
r_{shroud}	0.325 m
t_{\max}	5 mm
c	0.05 m
N_{blade}	30

The blockage factor distribution could be expressed analytically, due to the cross section being parabolic. For a parabolic cross section with its maximum thickness in the middle, the thickness as a function of axial coordinate is equal to

$$t(x) = 4\frac{t_{\max}}{c}x - 4\frac{t_{\max}}{c^2}x^2 \quad (4.11)$$

The blockage factor is equal to the ratio between the local channel width and the blade pitch. In this case, the channel width is equal to the pitch minus the blade thickness. The theoretical blockage factor distribution could therefore be expressed as

$$b(x, r) = \frac{\frac{2\pi r}{N_{\text{blade}}} - 4\left(\frac{t_{\max}}{c}x - \frac{t_{\max}}{c^2}x^2\right)}{\frac{2\pi r}{N_{\text{blade}}}} \quad (4.12)$$

This expression was used to verify the implementation of the 2D interpolation algorithm, by comparing the interpolator output for the blockage factor to Equation 4.12. By plotting the deviation and calculating the root-mean-square(RMS) error of the interpolated blockage field, the accuracy of the interpolator could be determined. If the error was found to be below an acceptable limit, the interpolation algorithm was deemed to be implemented correctly and functioning properly.

In order to quantify the accuracy of the metal blockage model, the results obtained through blade computation and body-force modeling had to be compared. This was achieved through setting up a blade computation and BFM flow domain of the same blade geometry with similar boundary conditions. For both simulations, an Euler solver scheme was used. In an Euler simulation, no boundary layers are resolved. Therefore, the only flow obstruction in the blade computation would be due to metal blockage and no boundary layer blockage. In the BFM simulation, the parallel force was disabled. This way, the only source terms acting on the flow would be introduced by the metal blockage model. In Figure 4.3, the mesh can be seen which was used to obtain the results of the blade computation of the stator geometry. In Figure 4.4 and Figure 4.5, 2D slices of the blade computation domain can be seen, in which the fluid boundaries are identified.

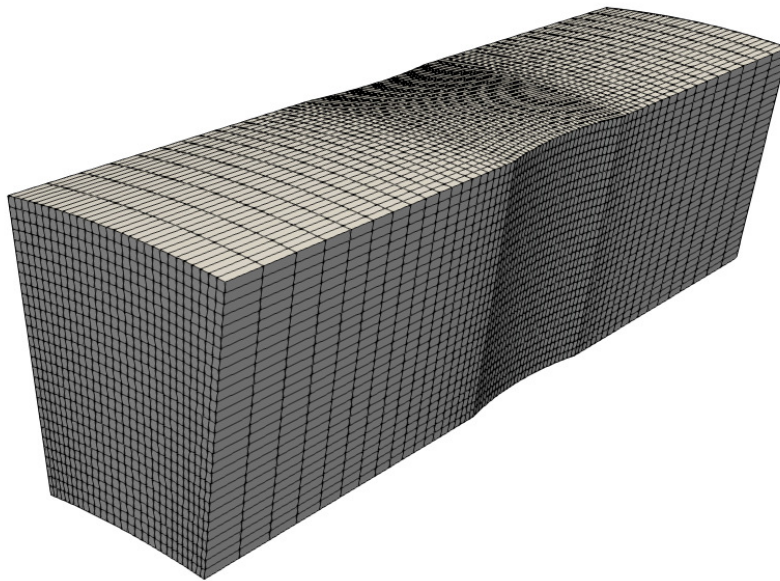


Figure 4.3: 3D mesh used for the blockage verification blade computation

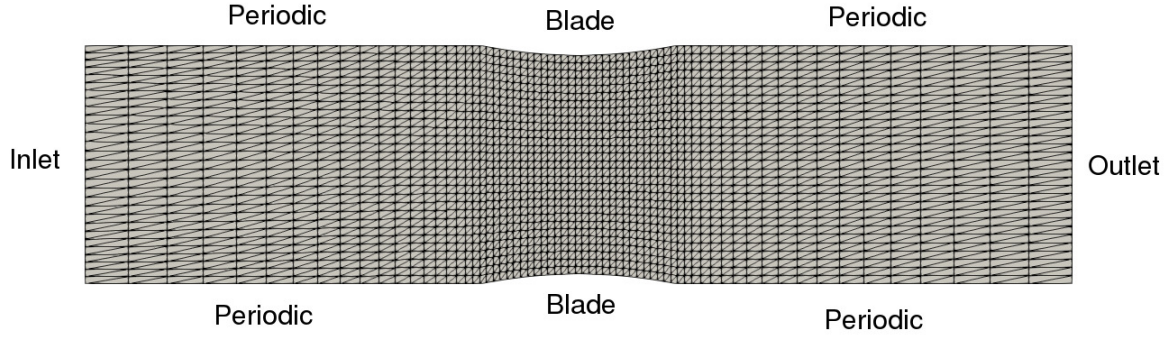


Figure 4.4: Axial-tangential slice of blade computation domain

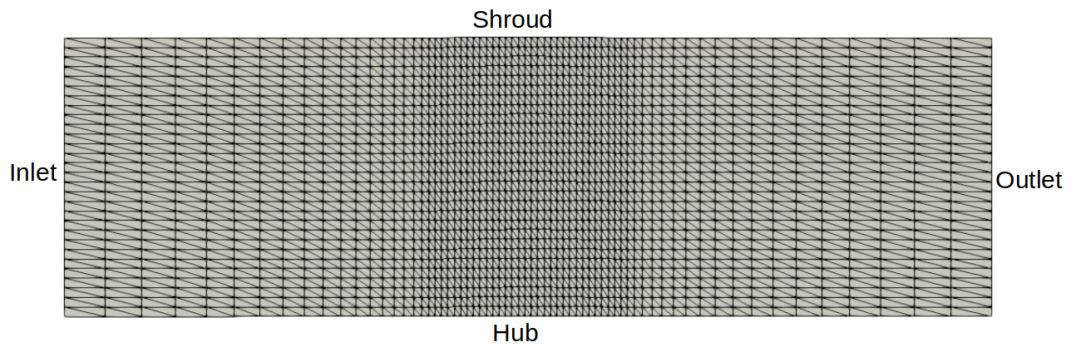


Figure 4.5: Axial-radial slice of blade computation domain

For the blade computation and BFM simulation, the same boundary conditions were used, apart from the blade surface, as this was not present in the BFM simulation. The BFM boundaries are visualized and identified in Figure 4.6 and Figure 4.7, where the red region is where the blockage source terms are added to the solution.

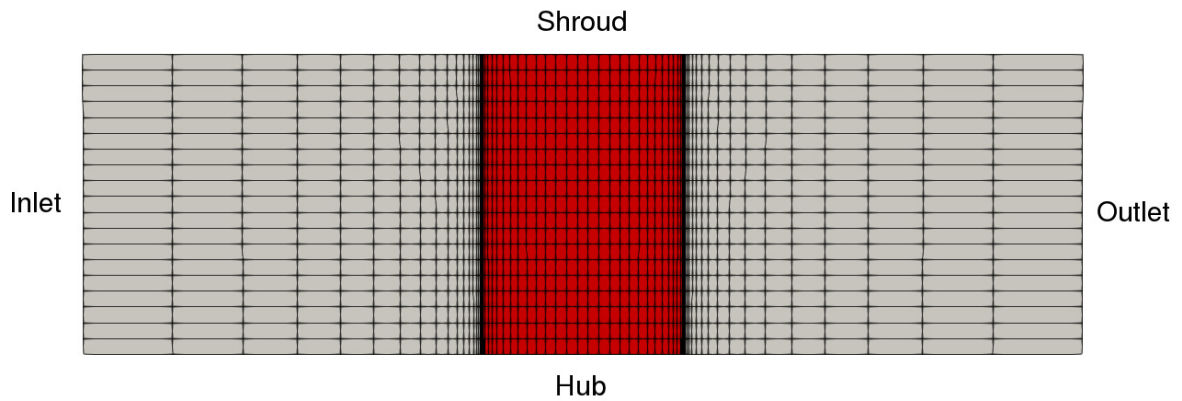


Figure 4.6: Axial-radial slice of BFM computation domain

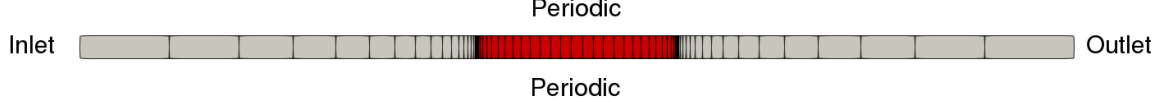


Figure 4.7: Axial-tangential slice of BFM computation domain at mid-span

In Table 4.3, the boundary conditions for the boundary markers in Figure 4.4, Figure 4.5, Figure 4.6 and Figure 4.7 are listed. The working fluid in these simulations was set to be standard air, with the same properties as listed in the assumptions for the parallel force derivations in Section 4.1.

Table 4.3: Boundary conditions used for the blockage verification simulations

Boundary	Boundary conditions
Inlet	$p_t = 105\,325\text{ Pa}, T_t = 288.7\text{ K}$
Outlet	$p_s = 101\,325\text{ Pa}$
Hub and shroud	slip-wall
Blade surface	slip-wall

By comparing the averaged pressure and axial mass flux trends of the BFM and blade computation results, the accuracy of the blockage source term model could be found. This was done through plotting the error along axial direction to see where the deviation was the highest and by computing the RMS. If the RMS was found to be below an acceptable limit, the blockage source term model was considered to work correctly.

5 BFM Case Study

After verifying the proper implementation of the BFM parallel force and blockage implementation, it was possible to compare BFM data to that of blade computations of actual turbomachinery geometries. The case chosen for this purpose was the Aachen turbine case. This is an axial, low speed turbine geometry with one rotor row in between two stator rows. The hub and shroud surfaces were defined as free slip walls and no tip gap was present in the rotor row. This made the case suitable for validation purposes, as annulus boundary layers are not modeled by the BFM and no tip gap modifications were implemented. By comparing the flow data from the Aachen turbine blade computation and those resulting from the BFM, the accuracy of the BFM could be quantified. In order to quantify the computational efficiency, the total computation times required for convergence were compared between the methods. This was done for several variations in mesh density for the BFM in order to visualize the trade-off between accuracy and computational efficiency.

5.1 Aachen Turbine Blade Computation Setup

The blade rows of the Aachen turbine consisted of prismatic blades with no tip or hub gap, with 41 blades in each row and a rotor rotation speed of 3500 rpm. The flow domain used for the blade computation can be seen in Figure 5.1. In Figure 5.2 and Figure 5.3, the cross sections of the Aachen turbine stator and rotor can be seen respectively.



Figure 5.1: Aachen turbine flow domain

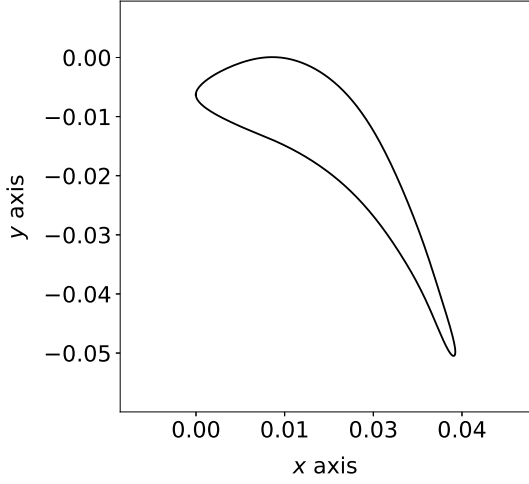


Figure 5.2: Aachen stator cross section

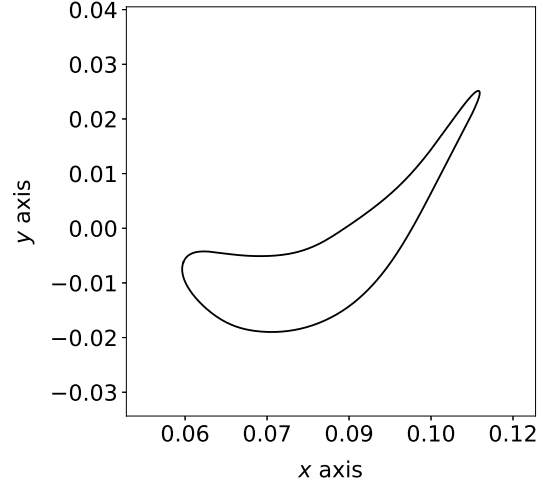


Figure 5.3: Aachen rotor cross section

The blade computation performed on this geometry was a RANS computation using a $\kappa - \omega$ -SST turbulence scheme with the boundary conditions listed in Table 5.1. The total number of cells in the mesh was in the order of 1.8×10^6 .

Table 5.1: Aachen blade computation boundary conditions

Boundary	Boundary conditions
Inlet	$p_t = 155\,000\text{ Pa}$, $T_t = 308\text{ K}$
Outlet	$p_s = 99\,614.9\text{ Pa}$
Blade surfaces	no-slip wall
Hub and shroud	symmetry

Table 5.2: Aachen blade computation direct solver settings

Setting	Type	Value(s)
Turbulence model	$\kappa - \omega$ -SST	
Fluid model	Ideal gas	$R = 287.058\text{ J kg}^{-1}\text{ K}^{-1}$ $\gamma = 1.4$
Viscosity model	Sutherland	$C_S = 110.4$ $\mu_{\text{ref}} = 1.716 \times 10^{-5}\text{ kg m}^{-1}\text{ s}^{-2}$ $T_{\text{ref}} = 273.15\text{ K}$
Conductivity model	Constant Prandtl number	$C_{KT} = 0.0257$
Spatial gradient model	Weighted least squares	
Linear solver	FGMRES	Error= $1e-4$, iterations=15
Flow numerical method	JST	Sensor coefficients=(0.3,0.18), Relaxation factor=0.75
Turbulent numerical method	Scalar upwind	Relaxation factor=0.95

5.2 BFM Computation Setup

One of the advantages of using a BFM compared to a blade computation is the relatively coarse mesh associated with BFM computations. By performing a mesh study on the BFM mesh used for the Aachen turbine case study, it was possible to find the level of mesh refinement required to reproduce the results from the blade computation.

The inlet, outlet, hub and shroud boundary conditions of the blade computation were carried over to the BFM simulation in order to ensure case similarity. The flow solver conducting the BFM simulation was the Euler solver. The reason why this solver was used instead of the RANS solver applied in the blade computation was the lack of viscous phenomena to be resolved. The annulus walls of the domain were identified as slip-walls, along which no boundary layers form. Therefore, the only relevant viscous phenomenon occurring in the flow would be the boundary layer on the blade surface. This phenomena is modeled by the parallel force model in the BFM, which therefore made the necessity for a viscous flow solver obsolete.

The two mesh parameters used in the mesh study were the number of nodes in axial direction per blade section and the mesh wedge angle. Both are illustrated in Figure 5.4 and Figure 5.5 respectively.

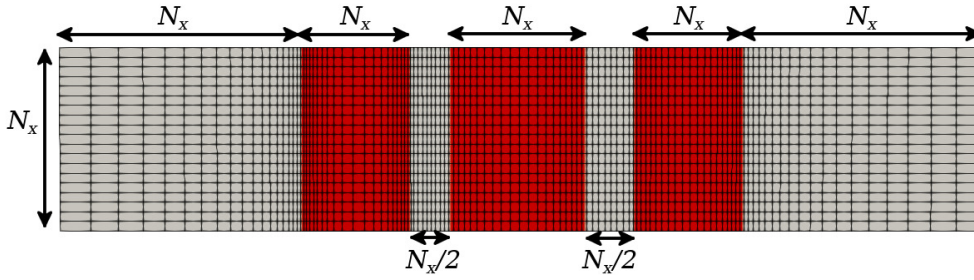


Figure 5.4: Axial node count definition

In Figure 5.4, the first mesh study parameter is illustrated. The red regions are the meridional projections of the turbine blades. By varying the axial node count (N_x) in those regions, the overall mesh density was adjusted. A high value for N_x therefore resulted in a more refined mesh, while the opposite was true for low values of N_x . In this part of the mesh study, N_x was varied between 10 and 60 nodes for all three blade rows. One extra run with a value of 100 for N_x was performed in order to have a reference trend for (assumed) grid convergence. The convergence criterion for both the BFM and blade computation results was set to a density residual value of 10^{-16} . By comparing the computation times of the blade computation and BFM simulations, the relative computational efficiency of the BFM was determined.

The mesh wedge angle parameter was varied to verify the independence of the BFM on the domain pitch. In order to verify this property, solutions of BFM simulations using varying cell counts in pitch-wise direction were compared. The mesh parameter used for this study is illustrated in Figure 5.5.

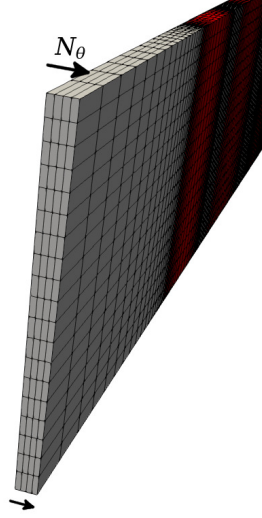


Figure 5.5: Tangential node count definition

Here, N_θ is the node count in tangential direction. By varying the pitch angle and applying one cell per degree of pitch angle, the number of cells along the arrows was varied. By performing BFM computations on meshes with varying wedge angle parameters, while maintaining a constant value for N_x , it was possible to verify the BFM independence on pitch-wise node count. This would be the case if the averaged result trends of the different wedge angles overlap. If this would be the case, it would mean that axisymmetric BFM simulations could be performed on meshes of one cell wide, which substantially increases computational efficiency due to the limited cell count in the mesh. The tangential node count was varied between $N_\theta = 1$ and $N_\theta = 4$ with a constant $N_x = 20$.

6 Results

In this chapter, the results from the various verification steps and mesh study are given. The various methodologies in Chapter 3 were set up to provide insight into the correct functionality of the BFM. The results of these tests inspire confidence in the workings of the new additions to the BFM if they comply to reference data. Once verified, the BFM was to be applied in an actual turbomachinery analysis case, where the BFM results were compared to blade computation data in order to determine the accuracy and computational efficiency with respect to the blade computation. The results regarding the parallel force addition can be found in Section 6.1. The results of the blockage factor verification experiment and interpolator quality are in the subsequent Section 6.2. Finally, the results regarding the Aachen turbine mesh study can be found in Section 6.3.

6.1 Verification of Parallel Force Implementation

In this section, the results of the parallel force verification are presented. The numerical setup and boundary conditions can be found in Section 4.1. With a constant CFL number of 8.0, the simulation converged steadily. The density residual convergence trend can be seen in Figure 6.1. The trend somewhat fluctuated initially while descending, but after about 3000 iterations, this changed to a nearly linear descend.

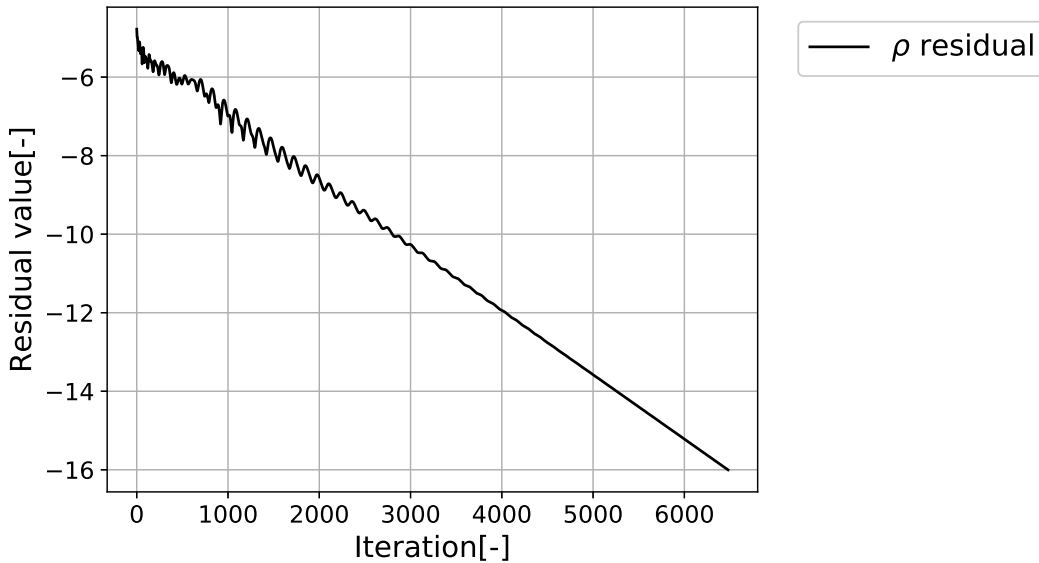


Figure 6.1: Density residual convergence trend

After convergence, the flow data was processed. In order to verify the parallel force implementation through Equation 4.2 and Equation 4.10, the pressure and entropy gradients had to be obtained. This was achieved through numerical differentiation on the *area-averaged* pressure and entropy trends. In Figure 6.2, the comparison between the entropy trend and parallel force can be seen. The trends overlap for the most part, apart from near the leading and trailing edge of the blade section. This fluctuation is probably caused by the high gradient in body-force magnitude. In the cells right outside the body-force zone, no source terms are added to the governing equations, while in the adjacent cells, this term is nonzero. This behaviour was also found in Latour's work[6]. Apart from the fluctuations near the leading and trailing edge, the entropy gradient seemed to have been calculated correctly by the BFM, as the trends overlap completely.

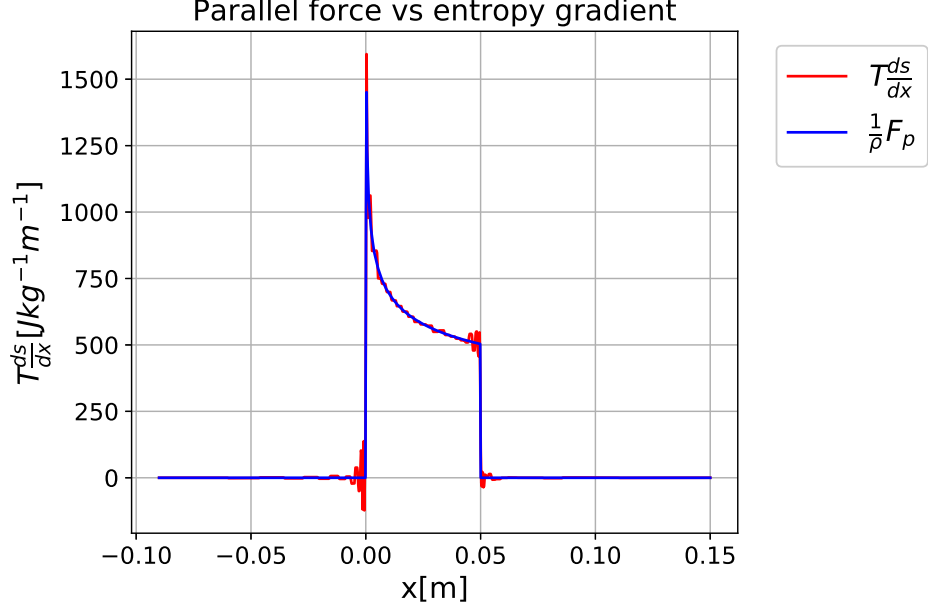


Figure 6.2: Comparison between entropy gradient and parallel force

In Figure 6.3, the comparison between the parallel force and pressure gradient can be seen. Similar to Figure 6.2, some fluctuation of the pressure gradient can be seen near the blade edges. In this case, the two trends are offset, although the behaviour seems to agree.

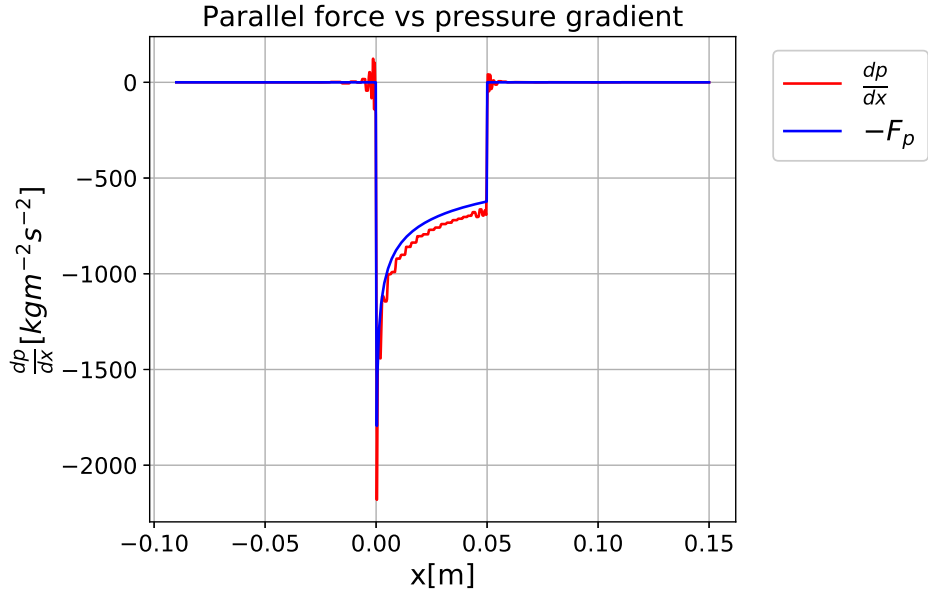


Figure 6.3: Comparison between pressure gradient and parallel force

The reason for the offset had to do with the flow not being fully incompressible. The average Mach number was about 0.24, which is generally considered to be on the edge of the compressible regime. This caused the gradient in kinetic energy to be significant. Therefore, the momentum term on the left hand side of Equation 4.2 should be taken into account. When subtracting the gradient in kinetic energy from the parallel force, the trends overlap completely, as can be seen in Figure 6.4. The results of the flat plate run with the parallel force term implemented resulted in Equation 4.2 and Equation 4.10 to be satisfied, therefore verifying its functionality.

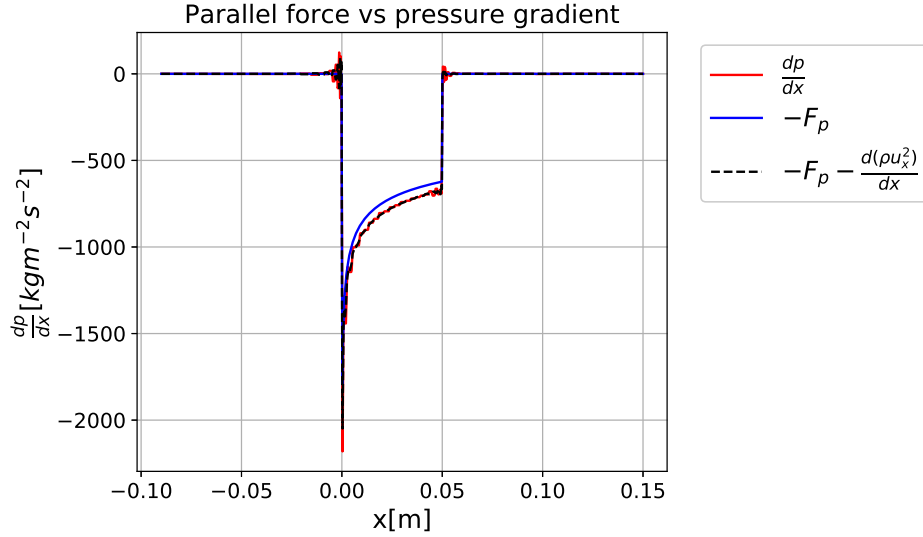


Figure 6.4: Comparison between pressure gradient and parallel force with inclusion of kinetic energy gradient

6.2 Verification of Metal Blockage Model Implementation

The main goal of the blockage factor verification was to verify the BFM's ability to model flow obstructions. If the model was to operate accordingly, the flow should accelerate in the blade region, reaching a maximum velocity at the height of the thickest portion of the blade. As was said in the experiments description, the parallel force was disabled for this run, as friction was not considered in the blade computation.

After interpolating the body-force parameters, the accuracy of the 2D interpolation algorithm could be determined. In Figure 6.5, the blockage factor distribution along the axial and radial axis throughout the flow field can be seen, alongside the distribution according to Equation 4.12.

Analytical blockage factor distribution

Interpolated blockage factor distribution

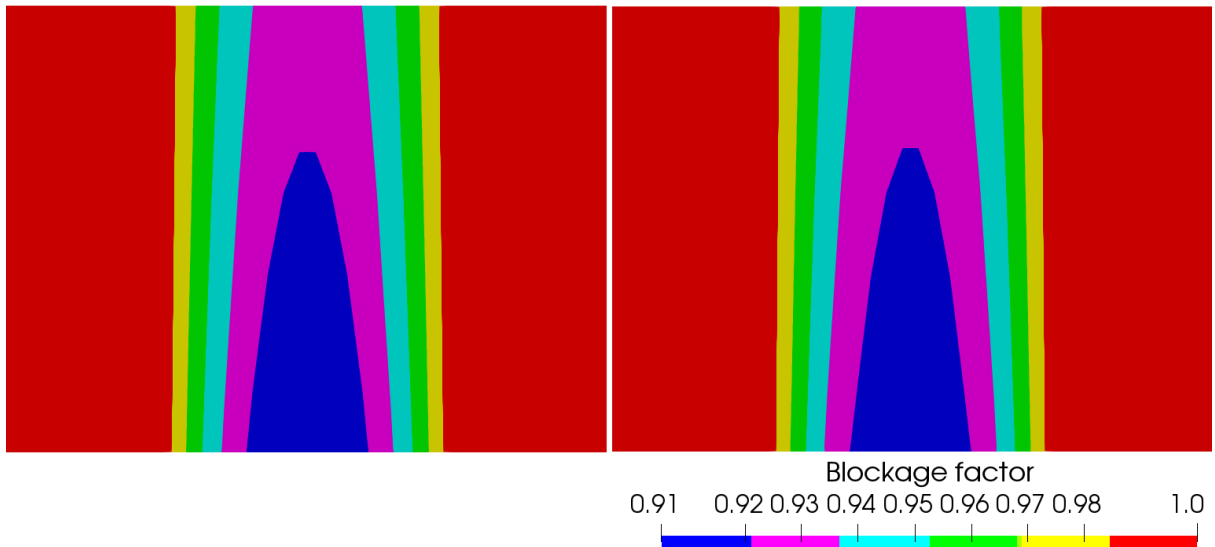


Figure 6.5: Comparison between interpolated and analytical blockage factor distribution

From a qualitative point of view, these distributions seem nearly identical. In order to obtain a quantitative measure for the level of accuracy of the interpolator, the deviation between the theoretical and interpolated blockage factor was calculated and visualized. In Figure 6.6, the deviation between the blockage factor field as it was interpolated and the theoretical blockage factor distribution is displayed.

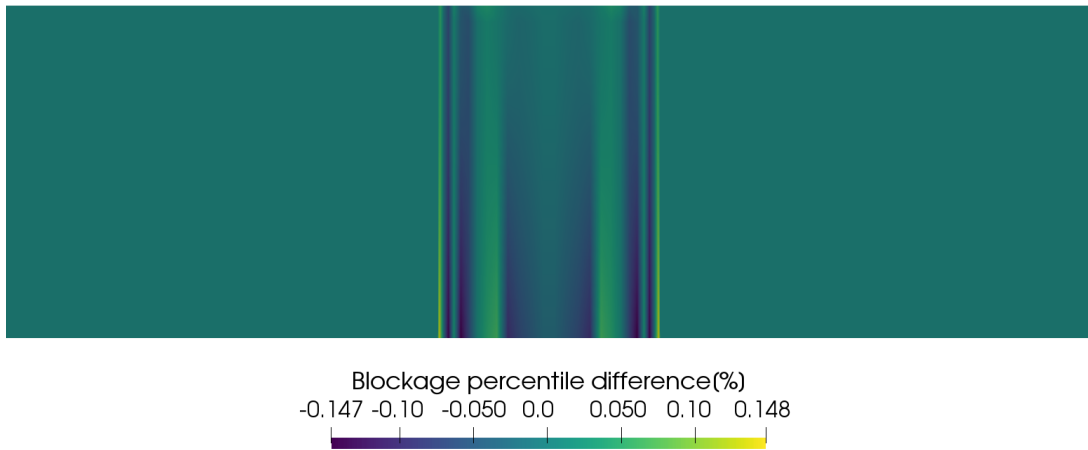


Figure 6.6: Comparison between interpolated and analytical blockage factor distribution

The deviation fluctuated between -0.15% and 0.15% throughout the blade region. The RMS error within the bladed region was calculated to be 0.041%, indicating that the error fluctuation was relatively high compared to the average value. There were some differences expected between the theoretical and interpolated distributions, as the mesh was relatively coarse (20 nodes in axial direction within the bladed region) and the mesh used a different axial node distribution (biexponential) compared the blade shape provided as input(uniform). Despite these differences, the interpolator was able to approach the theoretical solution accurately. The RMS difference of 0.041% was deemed low enough to continue using the method for further analyses.

After completion of the BFM and blade computation simulations, the results were analyzed and compared. Both the BFM and blade computation converged to their respective convergence targets set to a density residual of 10^{-16} . In Figure 6.7 and Figure 6.8, the Mach number distribution of the BFM and blade computation can be seen respectively. Both figures show the contour of the Mach number on the blade-to-blade plane at mid-span.

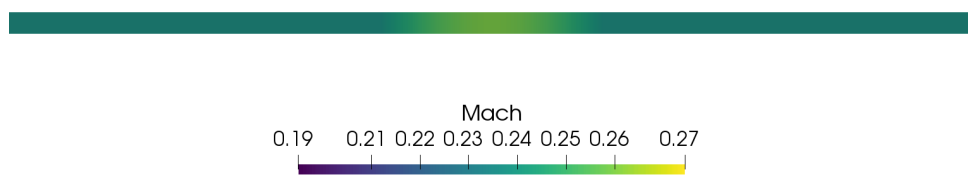


Figure 6.7: Mach number distribution in BFM results (radius=0.2875 m)

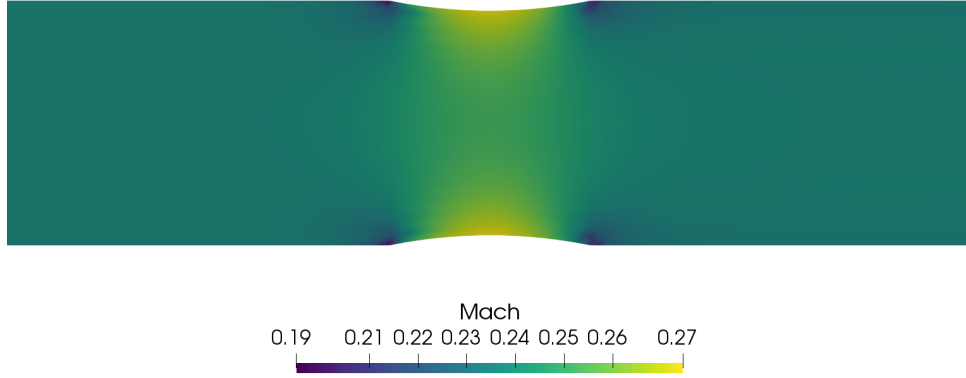


Figure 6.8: Mach number distribution in the blade computation results(radius=0.2875 m)

When looking at Figure 6.8, the flow-through area contracts and expands in the blade row, resulting in respective flow acceleration and deceleration, especially near the blade surface. Since the BFM replaces the blade by a volumetric force field, the effective presence of the blade row is constant in tangential direction. This, combined with the flow conditions being uniform upon entering the domain causes the flow to be uniform in tangential direction.

In order to check whether the metal blockage model performed correctly, the area averaged Mach number, mass flux and pressure trends of the BFM and blade along the rotation axis computation were compared. In Figure 6.9, Figure 6.10 and Figure 6.11 the respective averaged Mach number, mass flux and normalized static pressure trends are displayed. The black, dotted line represents an error margin of 1% with respect to the blade computation results.

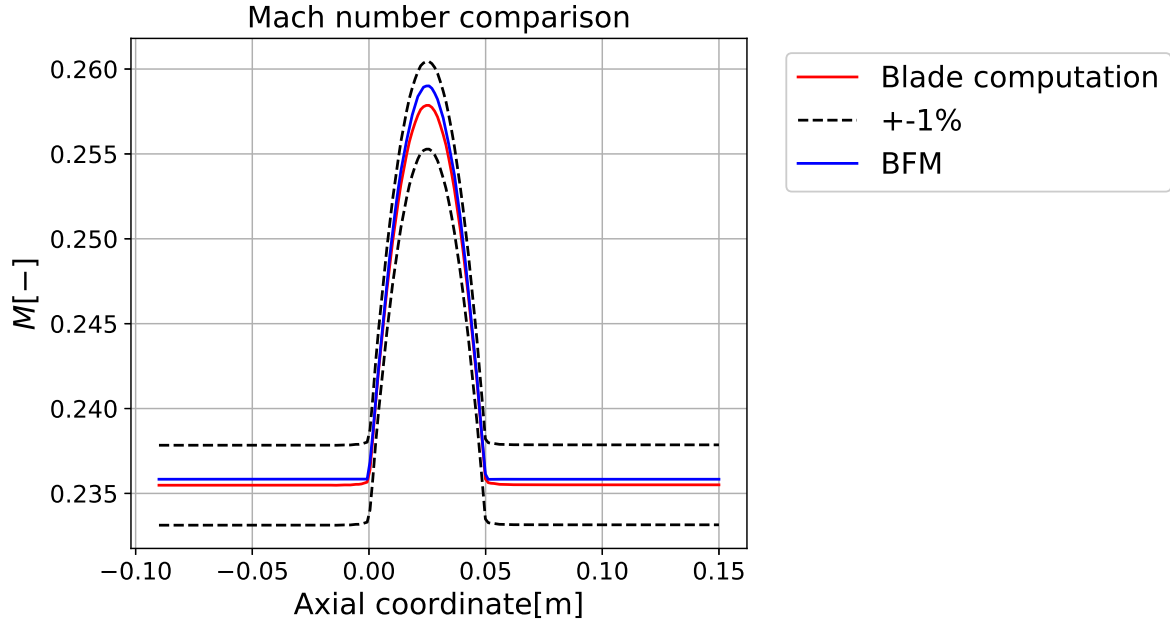


Figure 6.9: Mach number comparison between blade computation and BFM

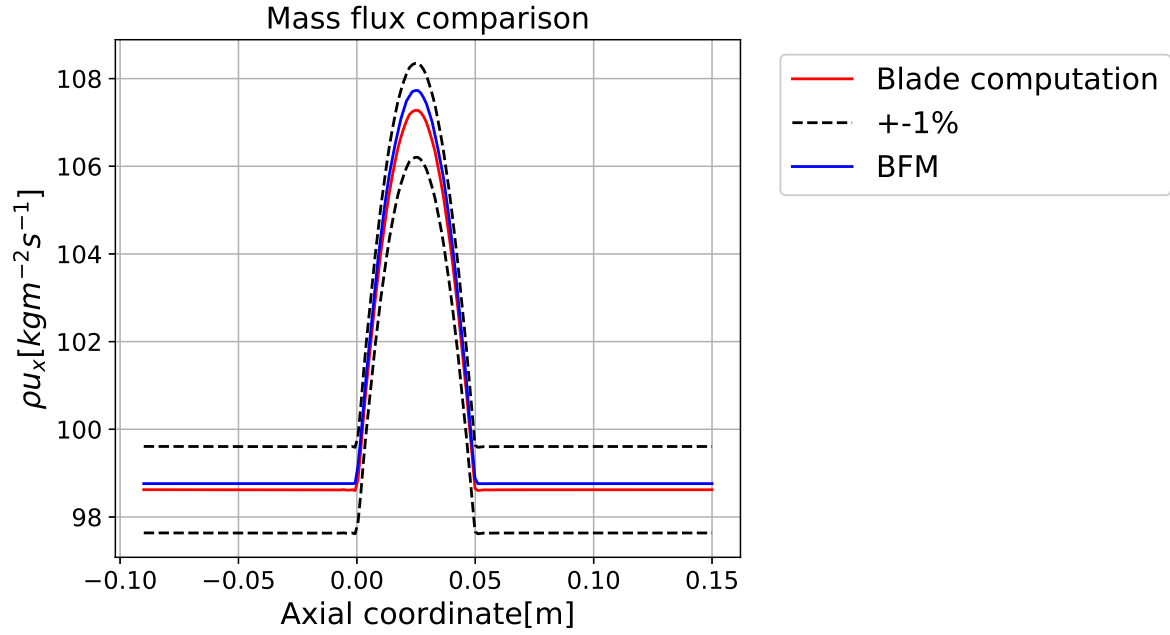


Figure 6.10: Mass flux comparison between blade computation and BFM

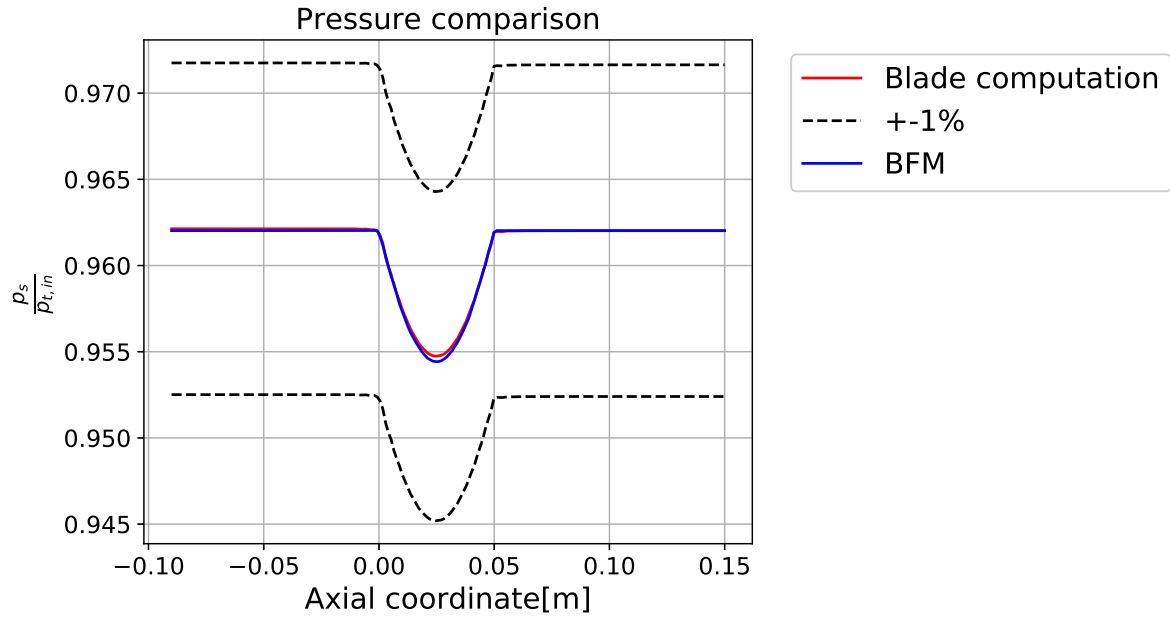


Figure 6.11: Normalized static pressure comparison between blade computation and BFM

For all three trends, the averaged BFM and blade computation results are in good agreement. The Mach number increases in the converging section of the blade and then decreases in the diverging section, returning to the same value the flow had upon entering the blade row. In Figure 6.12, the deviation between the mass flux trends of the BFM and blade computation results can be seen, alongside 1% margins shown, indicated by the black, dotted lines.

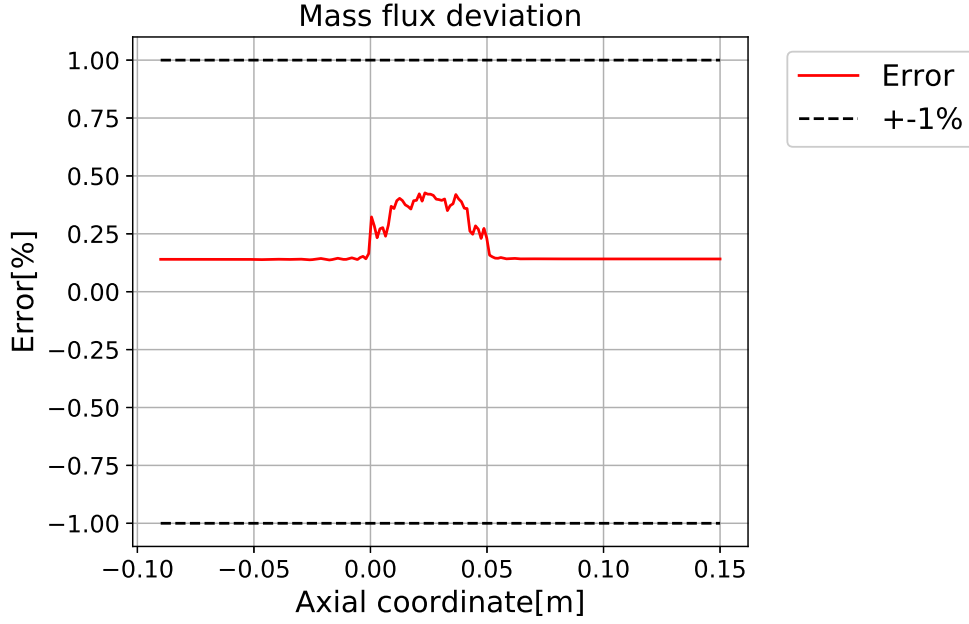


Figure 6.12: Deviation between BFM mass flux and blade computation mass flux trends

In the far field regions, the BFM overestimated mass flux by about 0.12%, while in the bladed region, this difference increased to 0.35%. Overall, the RMS error of the mass flux was calculated to be 0.18%. This deviation is about an order of magnitude higher than the interpolation RMS error, so the error was likely not caused by the inaccuracy introduced by the interpolation method.

6.3 Aachen Turbine Case Study Results

In this section, the results of the Aachen turbine mesh study are presented. A comparison was made between the blade computation and BFM results, where the mesh density and number of cells in axial and tangential direction were varied respectively. The accuracy of the BFM was measured through axial trend comparison of several relevant flow variables. These were mass-flux-averaged over the radial-tangential plane in order to produce clear trends. Through computation of the deviation between the BFM and blade computation results, it was possible to determine the magnitude and location of inaccuracies of the BFM.

6.3.1 Results as a Function of Axial Node Density

The BFM simulations converged successfully to their respective convergence target. In Figure 6.13, the convergence trends of the different meshes can be seen. All BFM simulations were performed using a single processor, with a constant CFL number of 8.0. The mesh with the highest density, having 100 nodes in axial direction per blade row, took about 220 minutes to converge, while that of the least dense mesh took less than 30 seconds. The blade computation took about 52 hours wall clock time on 20 processors to converge to the same level. This demonstrates the high computational efficiency of the BFM. This means that the BFM overall converged 3 orders of magnitude faster than the blade computation.

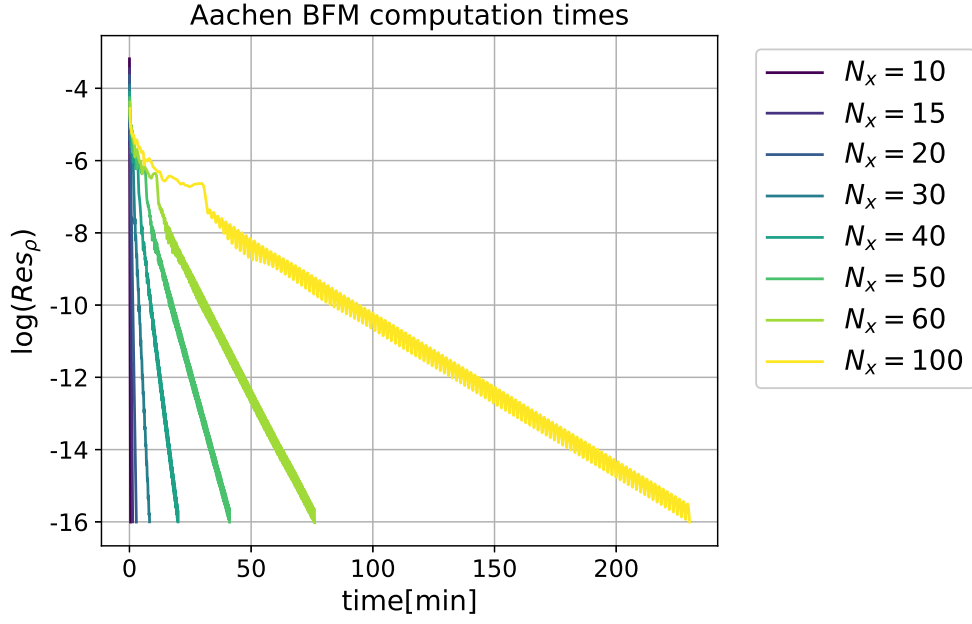


Figure 6.13: Convergence trend comparison between the different mesh densities

In Figure 6.14, the trend of total convergence time for each level of mesh refinement can be seen. The values are tabulated in Table 6.1. The trend is nearly parabolic, which was expected, as the number of cells in the mesh scaled by N_x^2 . The reason why the computation time increases linearly between 60 and 100 had to do with the mesh of $N_x = 100$ did not have 100 nodes in radial direction. A segmentation fault prevented the use of more than 70 nodes in radial direction. Therefore, the node count did not scale exactly scale quadratic between 60 and 100 nodes.

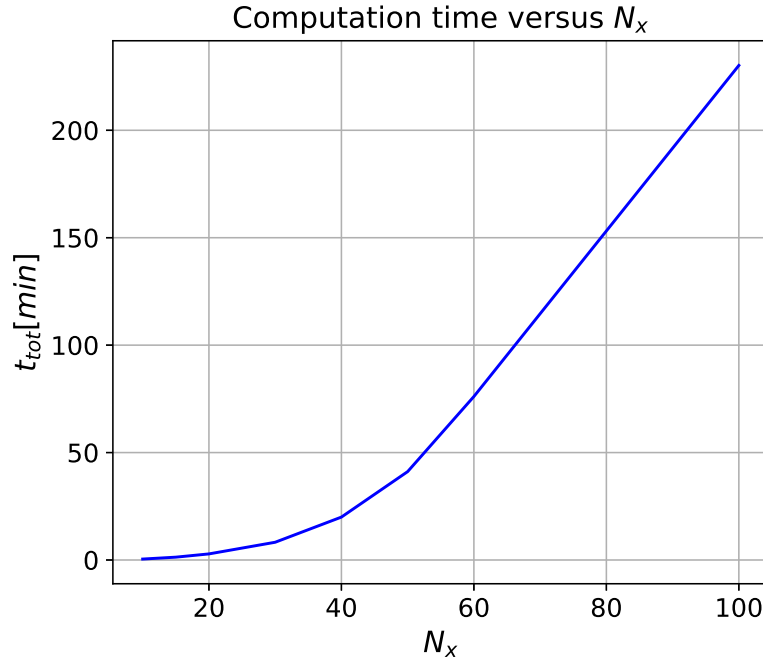


Figure 6.14: Total convergence time for each level of mesh refinement

Table 6.1: Tabulated total computation time for each level of mesh refinement

N_x	10	15	20	30	40	50	60	100
Time[min, sec]	0, 26	1, 19	2, 50	8, 16	19, 58	41, 8	76, 6	230, 11

The relatively high computational efficiency of the BFM with respect to the blade computation was considered to be a significant benefit of using body-force modeling in design applications. However, for such applications, relative accuracy is equally as important. In Figure 6.15, the mass flow averaged trends of the normalized static pressure according to the BFM and blade computation simulations can be seen. When globally comparing the BFM and blade computation results, the BFM appears to agree very well to the blade computation results, always remaining within 5% error margin.

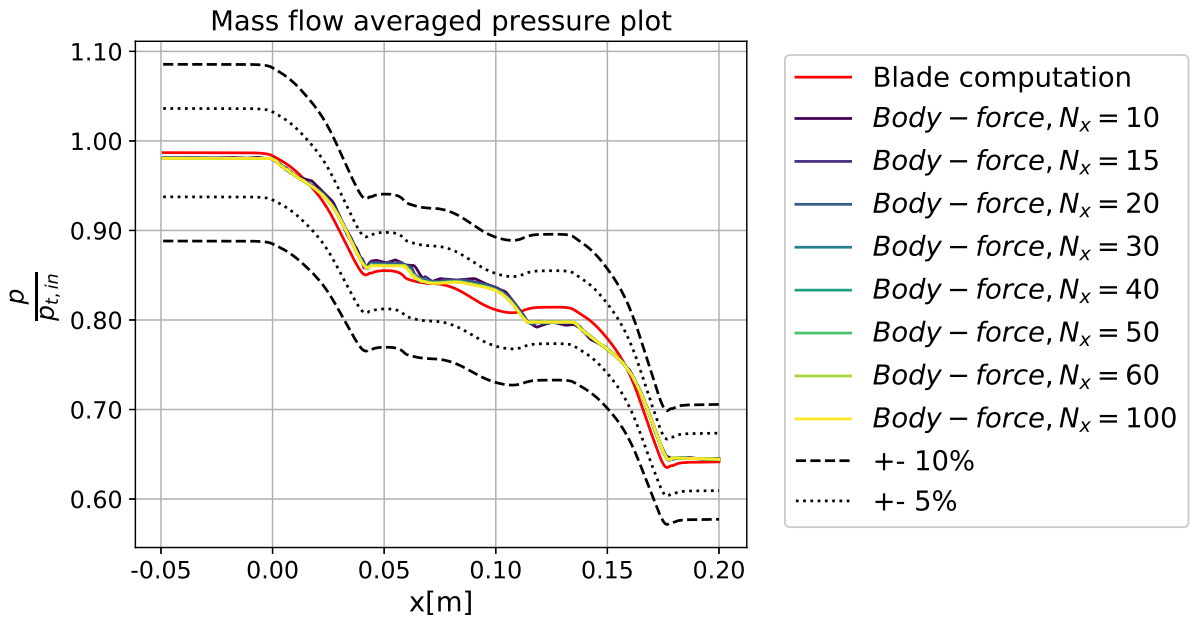


Figure 6.15: Comparison of nomalized static pressure trends

In Figure 6.16, the exact percentile deviation of the BFM static pressure trends can be seen. The BFM overestimated the static pressure by a maximum of 3% near the trailing edge of the rotor, after which it underestimated it in the second stator row.

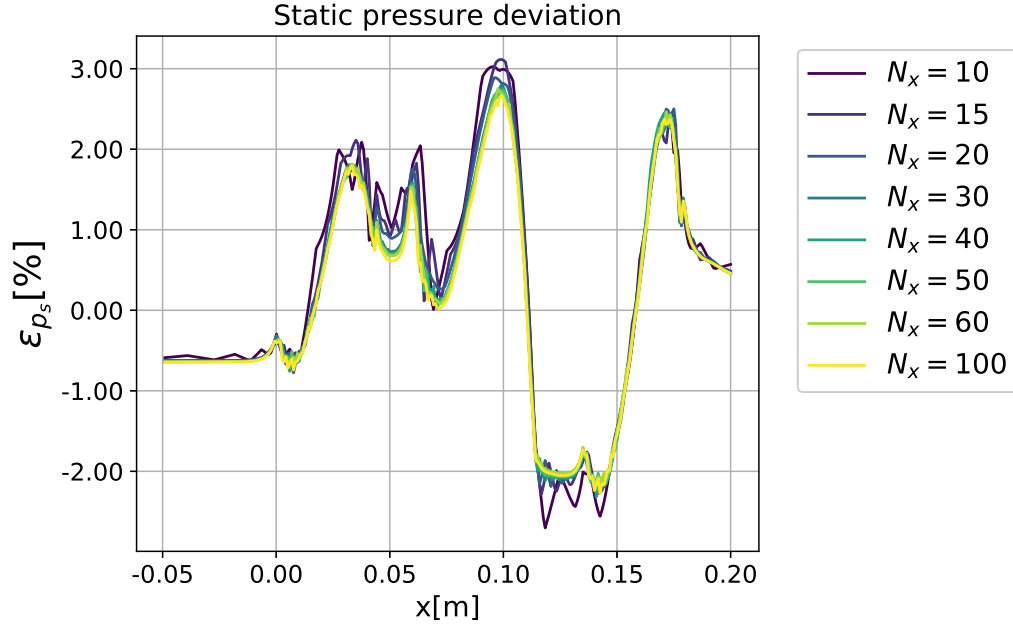


Figure 6.16: Static pressure deviation

The stagnation pressure trend behaved similarly to the static pressure trend, which can be seen in Figure 6.17. There does seem to be some difference between the stagnation pressure trends within the rotor row. Whereas the trend of the blade computation immediately decreased with a near constant gradient, the BFM initially shows a more gradual decrease and ends up at a slightly lower stagnation pressure value compared to that of the blade computation. The stagnation pressure deviation with respect to the blade computation results is shown in Figure 6.18. Due to the parallel force, a total pressure drop across the two stator rows can be seen and the outlet total pressure is approximated within 1%. The highest deviation is seen throughout the rotor, where the deviation peaks at around 5.5%, after which the deviation remains within 1%.

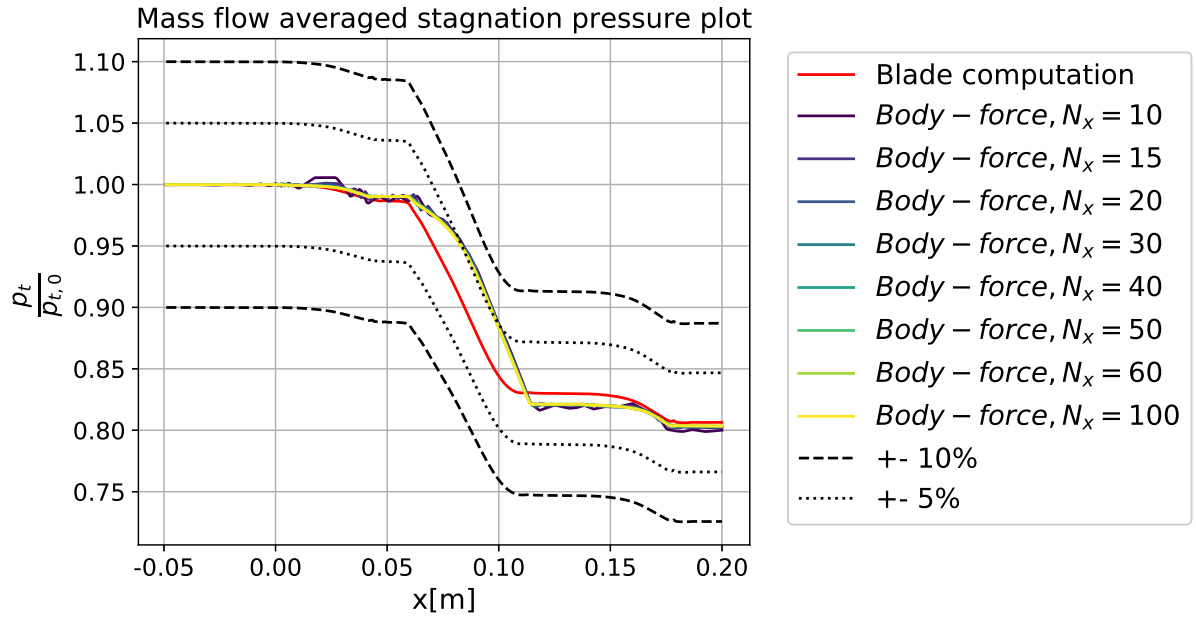


Figure 6.17: Comparison of nomalized total pressure trends

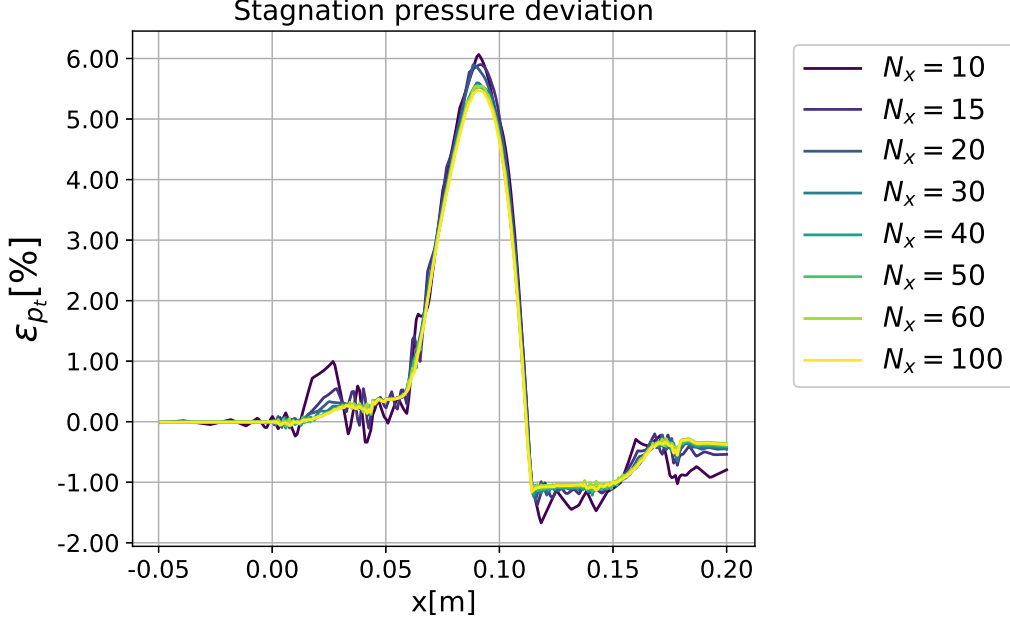


Figure 6.18: Stagnation pressure deviation

The stagnation temperature comparison can be seen in Figure 6.19. The BFM correctly shows no stagnation temperature drop in the stator rows, as no work is extracted in those regions. Over the rotor row, the stagnation temperature drops. There seems to be somewhat of a difference between the shape of the trends there, but the overall stagnation temperature drop over the rotor row is approximated well within 1%.

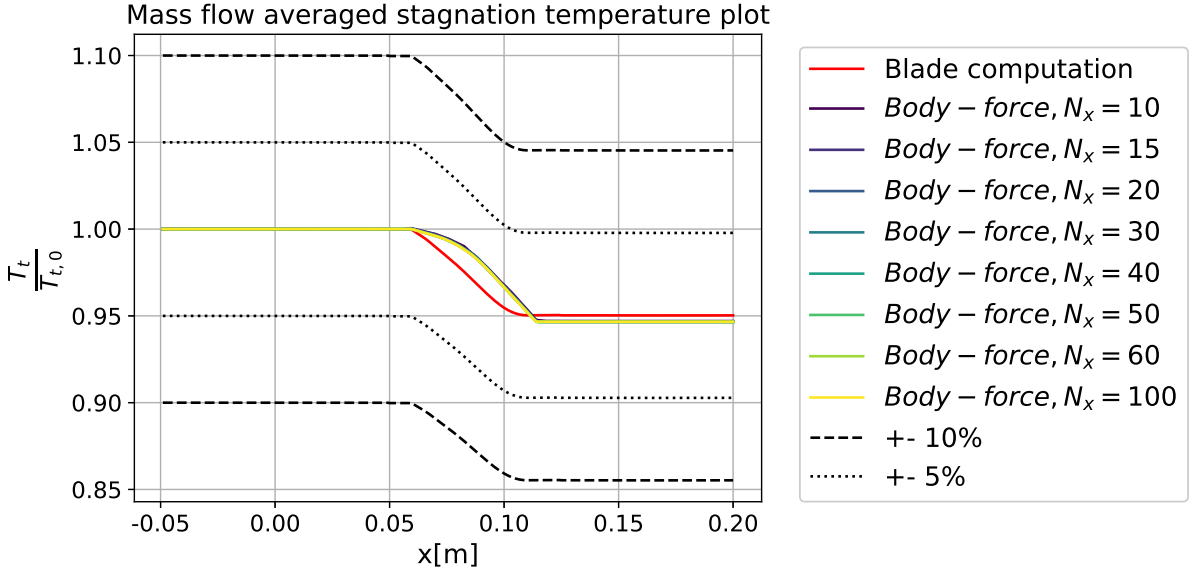


Figure 6.19: Comparison of nomalized total temperature trends

The stagnation temperature trend was reproduced more accurately by the BFM compared to the stagnation pressure. In Figure 6.18, the stagnation temperature deviation trends can be seen. The shape of the deviation is similar to the stagnation pressure deviation trend in Figure 6.18, but it peaks at around 1.5% instead of 5.5%. When looking at the outlet stagnation pressure deviation, it underestimated the extracted work within 0.5%.

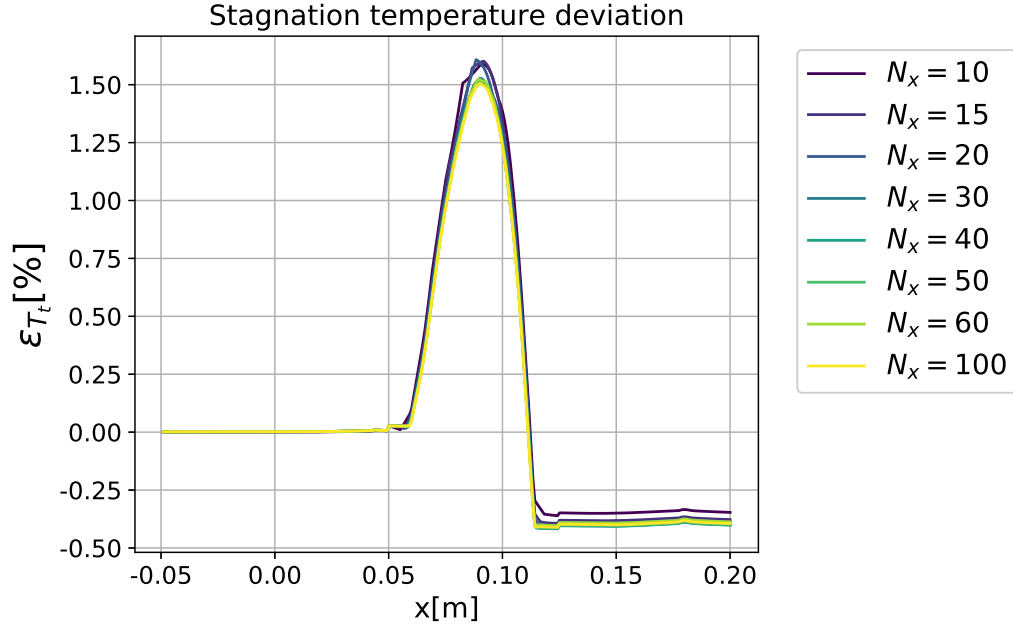


Figure 6.20: Stagnation temperature deviation

For the stagnation and static pressure and temperature trends, the BFM approximated the blade computation results rather well. This was not the case for all flow variables however. In Figure 6.21, the comparison between the absolute flow angle trends can be seen.

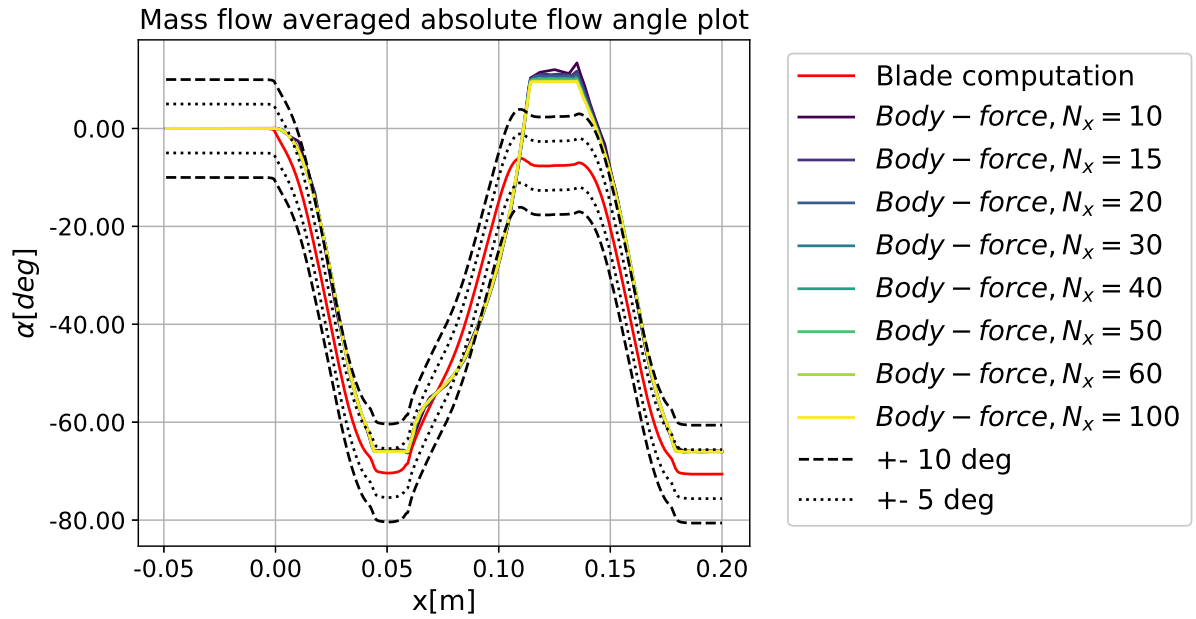


Figure 6.21: Comparison of absolute flow angle trends

The BFM trend for flow deflection was in good agreement with the blade computation results throughout the first stator row, with a slightly higher outflow angle compared to the blade computation. Contrary, the rotor flow deflection was overestimated. The mass flow averaged rotor outflow angle was about 17° higher than that of the blade computation. The second stator row returned the flow angle to a value in better agreement with that of the blade computation.

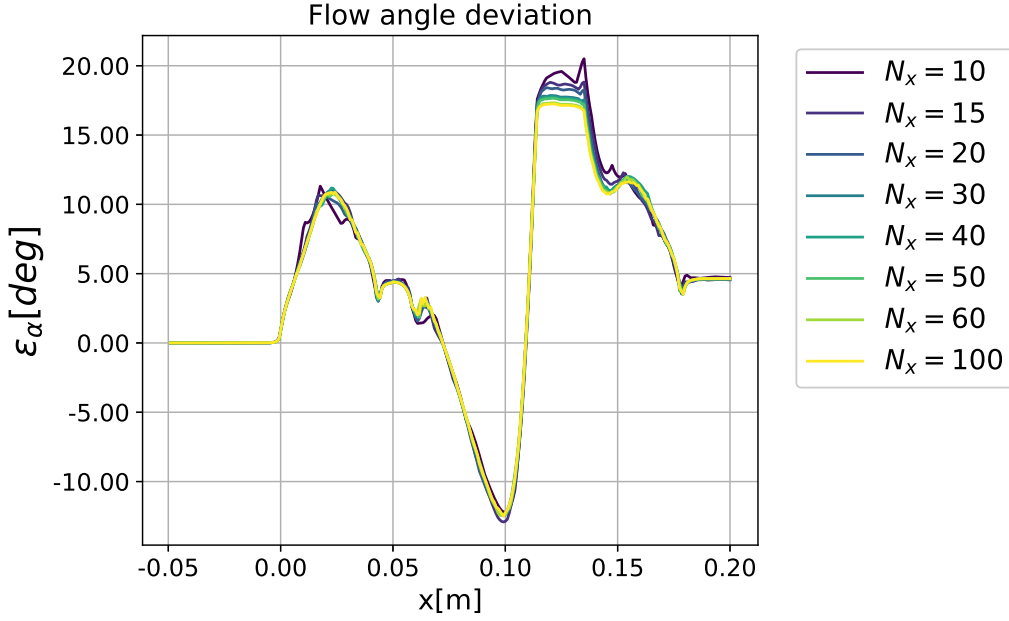


Figure 6.22: Absolute flow angle deviation

In Figure 6.22, a more detailed view of the flow angle deviation is given. Within the first stator row, the BFM overestimated the flow deflection, with the stator outlet flow angle being about 5° higher compared to the blade computation. The flow turning within the rotor row seems to be off, as it initially underturns the flow, while the rotor outlet flow angle was around 17° higher than that of the blade computation. The second stator row brings this deviation back to a deviation of 5° , similar to that of the first stator row.

Up to now, there has not been much variation in the BFM results on the different meshes densities. However, when looking at the trends of entropy increase (Figure 6.23), the trends become more distinct. Near the leading and trailing edges of each blade row, the entropy trend of the BFM computations fluctuate. For lower mesh densities, these fluctuations increase. For the lowest mesh density, being 10 nodes in axial direction, the fluctuations in entropy cause the overall trend to be hard to recognize, while for 20 nodes, the trend seems to have converged.

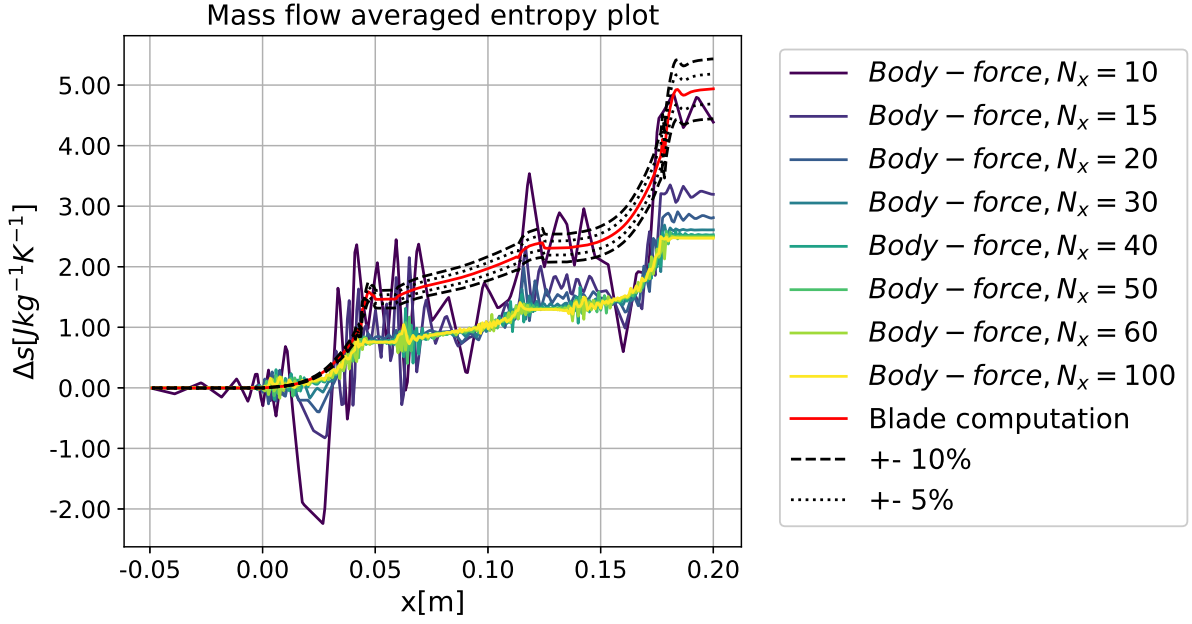


Figure 6.23: Comparison of entropy trends

When globally comparing the entropy trends of the blade computation and BFM simulation, the absolute values seem to be off. However, when comparing the relative increase in entropy per blade row compared to the total entropy increase, the trends are much closer. This can be seen in Figure 6.24. Here, the entropy trends were divided by the total entropy increase over the turbine.

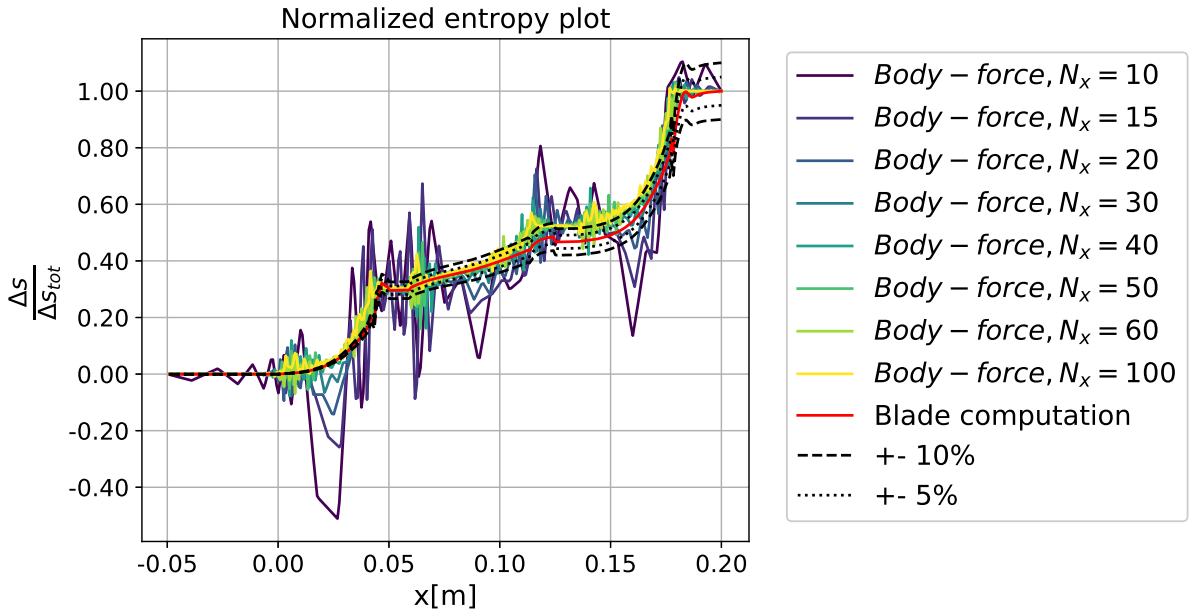


Figure 6.24: Comparison of normalized entropy trends

The BFM trends with mesh densities of 15 nodes or higher seem to approximate the blade computation trend rather well, especially through the first stator and rotor row. Near the trailing edge of the rotor, the entropy trend seems to deviate more than 10%, after which this difference decreases again in the second stator row.

In Figure 6.25, the mass flux trends in axial direction can be seen. Here, there seems to be a significant difference between the blade computation and BFM results. Over the entire domain, the mass flux according to the BFM is about 20% higher with respect to the blade computation trend.

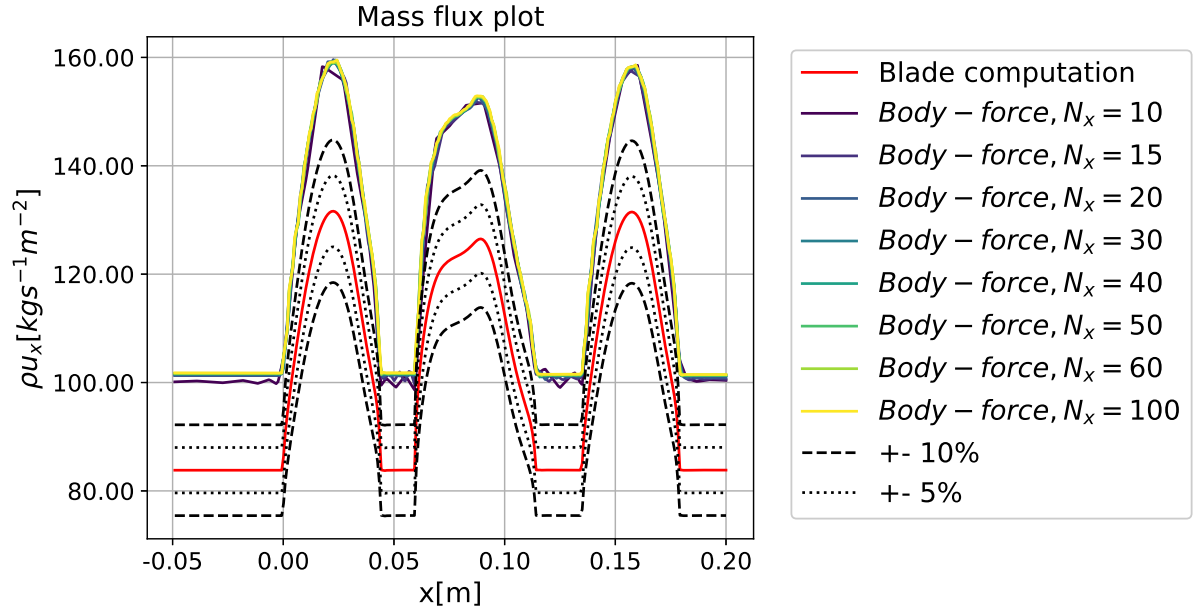


Figure 6.25: Comparison of mass flux trends

This overestimation of mass flux seemed to be almost constant throughout the turbine, as can be seen in Figure 6.26. Near the leading and trailing edges of the blade rows, a short peak can be seen, after which the trend provides values of the mass flux that are overestimated by 20%. When comparing the different levels of mesh density, the only outlier seemed to be the trend of $N_x = 10$.

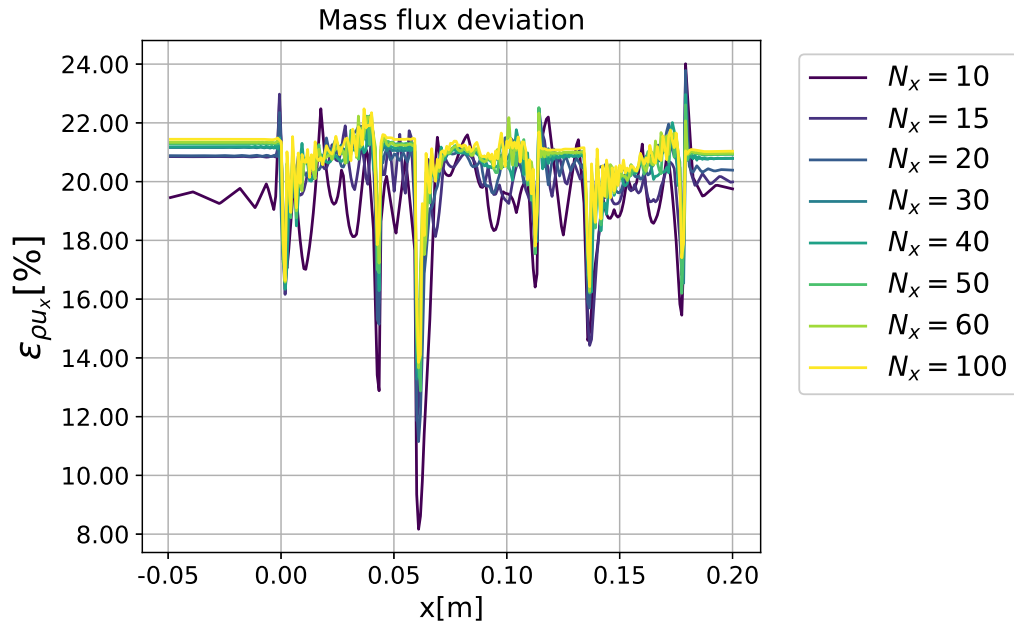


Figure 6.26: Mass flux deviation

A possible explanation for these differences in mass flux could be obstruction due to boundary layers. The source term in the continuity equation introduced by the BFM only took metal blockage into account, but not the flow blockage introduced by the displacement boundary layer on the blade surface. The Aachen turbine has a mid-span pitch solidity of 1.0 and a span solidity of about 0.6, meaning that the channels in between the blades within each blade row are relatively narrow. For such geometries, the obstruction due to boundary layers can be significant compared to the obstruction due to metal blockage alone.

In order to get a better understanding of the detailed flow behaviour, plots were made of flow variable trends in radial direction at several stations in the turbine. In Figure 6.27, relevant sections are identified at which flow variables in radial direction were extracted and compared to those from the blade computation.

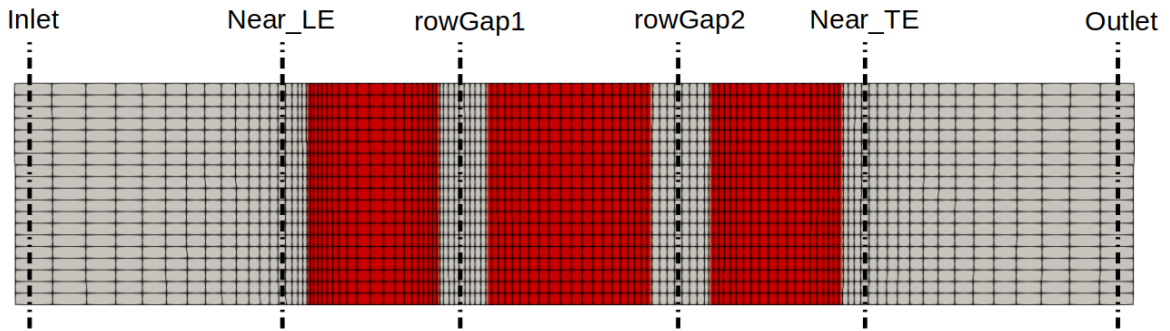


Figure 6.27: Axial station definition

From the mass flow averaged plots, it became clear that the highest deviations between the blade computation and BFM results occurred in the two row gaps. In Figure 6.28 and Figure 6.29, the radial static pressure and static temperature trends in the first row gap can be seen. For both of these quantities, the BFM results agree quite well with the blade computation results. The trends remain well within the 5% margin and have a shape similar to the blade computation trend.

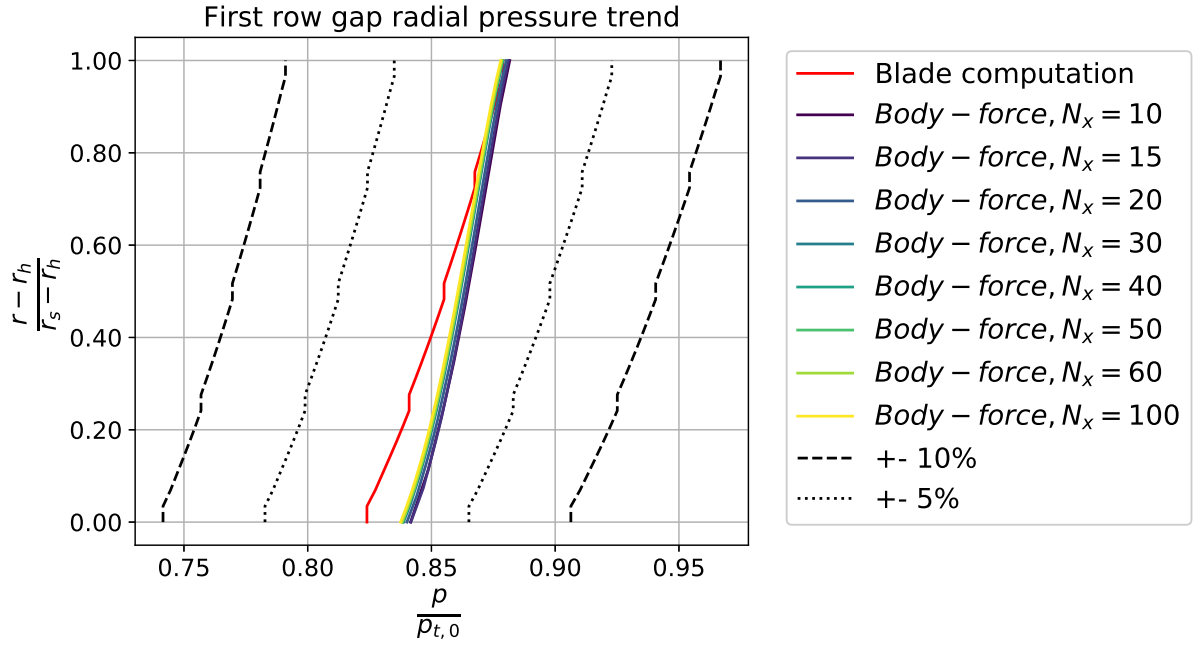


Figure 6.28: Radial static pressure trend between first stator and rotor

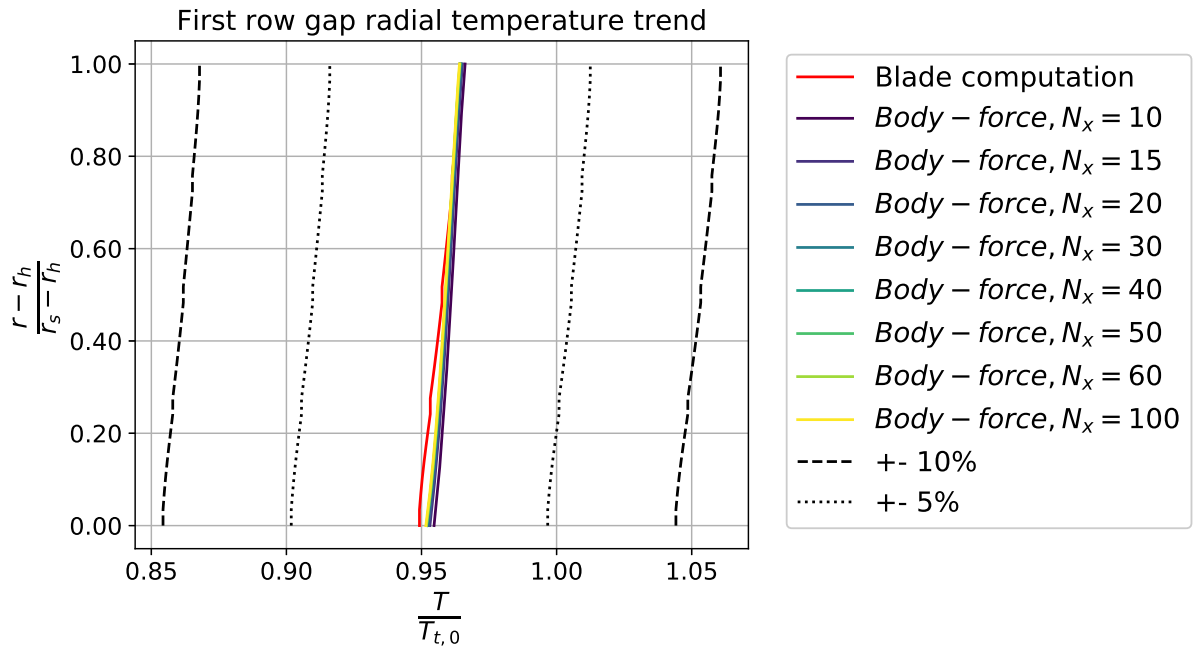


Figure 6.29: Radial static temperature trend between first stator and rotor

This pattern is repeated in the second row gap. Both the blade computation and BFM results show near constant values for static pressure and static temperature in radial direction and the BFM results remain well within 5% error margin.

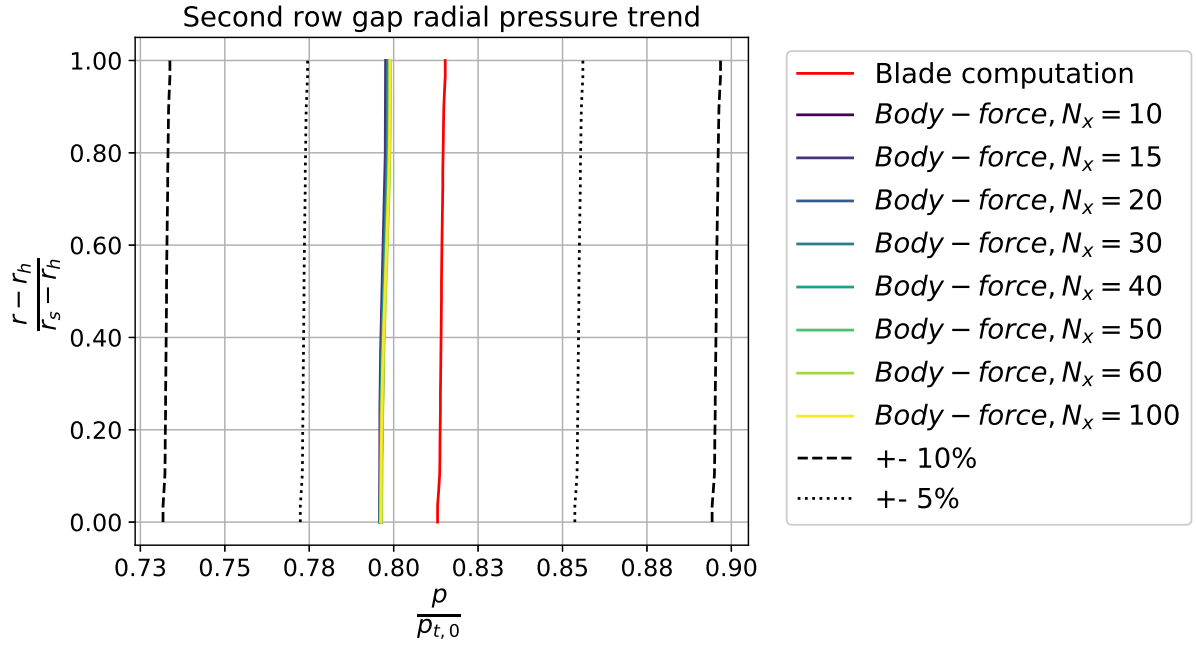


Figure 6.30: Radial static pressure trend between rotor and second stator

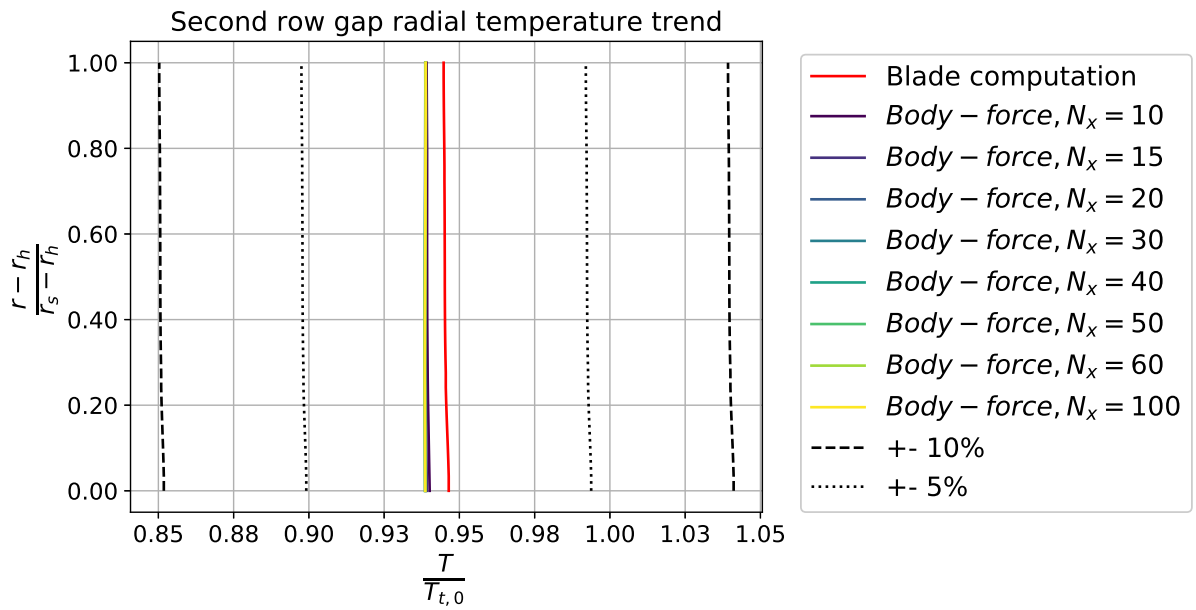


Figure 6.31: Radial static temperature trend between rotor and second stator

In Figure B.3 and Figure B.4, the radial absolute flow angle trends at the first and second row gap can be seen. The blade computation trend in the first row gap shows the flow deflection to be higher near the hub and gradually decreasing towards the shroud. The BFM trends show a nearly constant flow deflection over the blade height, offset by about 5° on average.

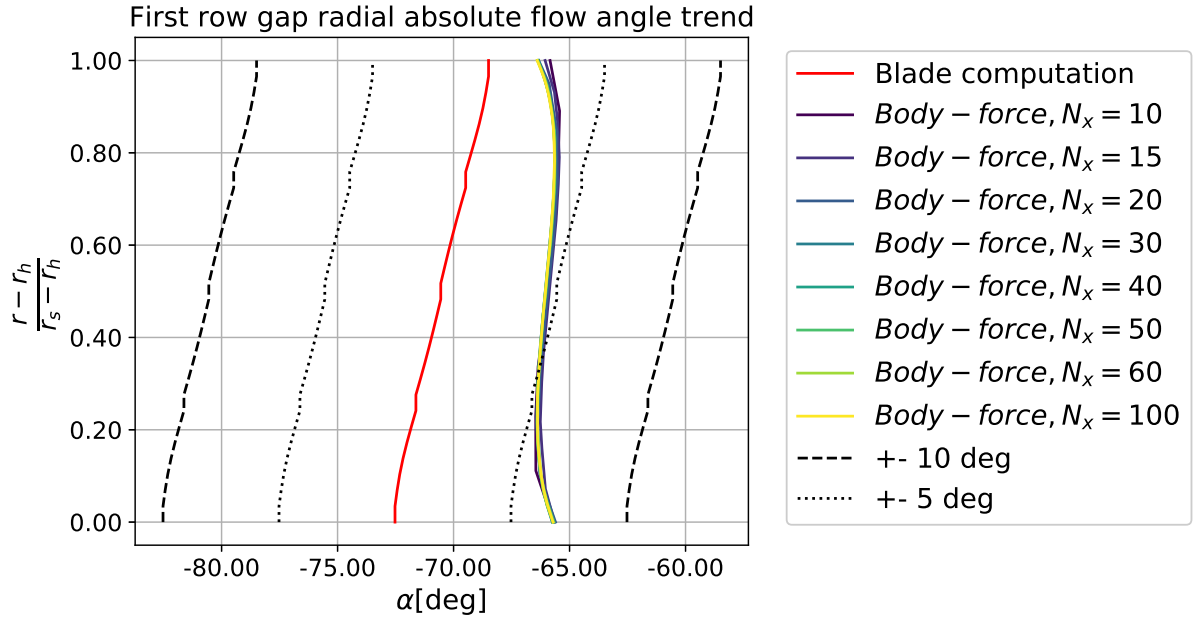


Figure 6.32: Radial absolute flow angle trend between the first stator and rotor

The flow angle trends in the second blade row have a more similar shape, but show an offset of about 20° . The overturning near the hub and underturning near the shroud are nicely captured by the BFM.

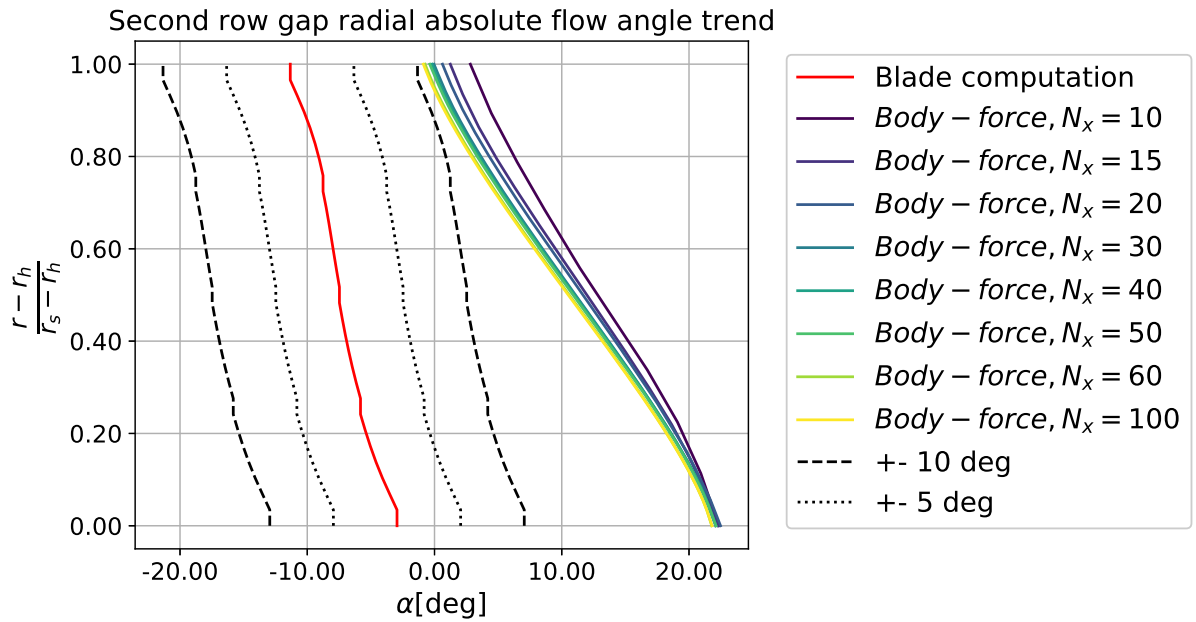


Figure 6.33: Radial absolute flow angle trend between rotor and second stator

Overall, the BFM seems to provide trends of flow quantities well in agreement with the blade computation results, with a few exceptions. The largest differences were found in the rotor row, where the stagnation pressure and stagnation temperature trends over the rotor row seemed showed a different shape compared to that of the blade computation. Also, the rotor outlet flow angle deviated more than 15° , which is significant. Finally, the mass flux throughout the turbine was overestimated by the BFM. This is probably due to the lack of boundary layer blockage in the BFM, as it only uses metal blockage as a source term. The trend of the mass

flux trends was respected very well.

6.3.2 Tangential node count study

In this section, the results of the second part of the mesh study are presented. Here, the goal was to demonstrate the BFM independence on pitch-wise cell count by performing BFM computations with varying tangential cell counts and see whether the respective trends are similar. The axial node count was kept constant at $N_x = 20$ for this study. In Figure 6.34, the normalized static pressure trends for the selection of tangential cell counts, as well as the blade computation results can be seen.

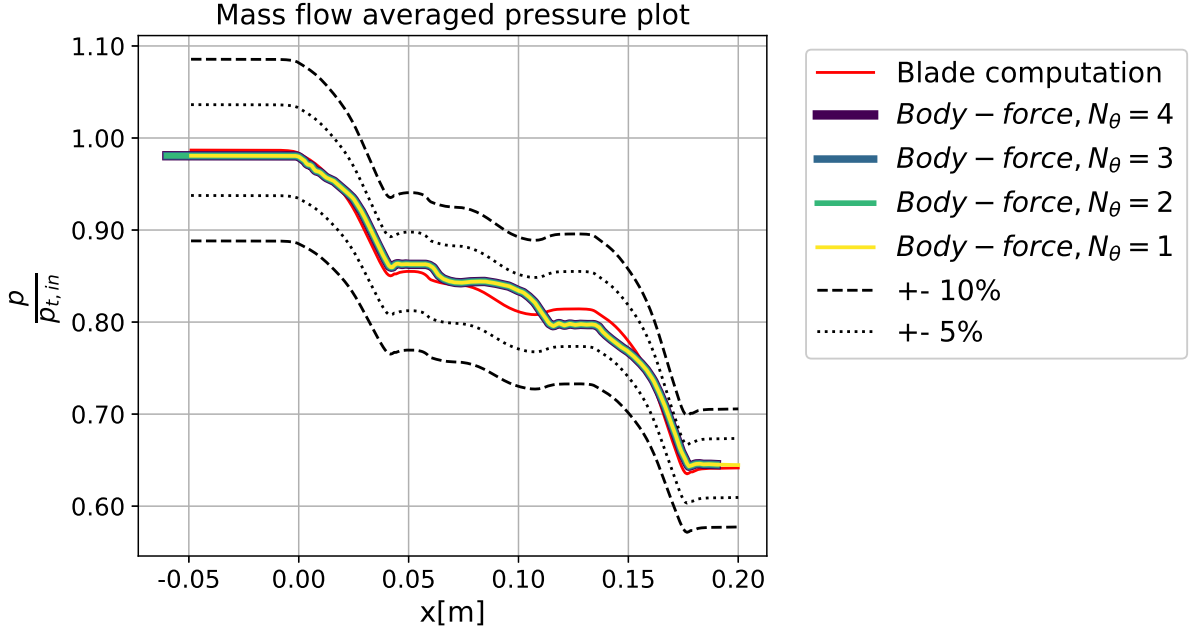


Figure 6.34: Static pressure N_θ comparison

The static pressure trends overlap completely for the different meshes. This was also found to be true for the other flow variables. This confirms that the BFM is indeed independent of tangential mesh count for axisymmetric simulations.

6.3.3 Grid convergence

For most results in the previous sections, the BFM trends of the different levels of mesh density remained relatively close together. This made it difficult to check for grid convergence on a qualitative point of view. In this section, root mean squared (RMS) error trends are provided for a series of significant flow variables. Here, the RMS is defined as

$$RMS_y = 100\% \sqrt{\frac{\sum_{i=0}^{i=n} (\frac{y_i}{\hat{y}_i} - 1)^2}{n}} \quad (6.1)$$

Here, y_i is the *mass-flux-averaged* flow quantity, \hat{y}_i the respective quantity according to blade computation results and n the number of stations in axial direction. By comparing the RMS values at varying levels of grid refinement, the point of grid convergence could be determined. In Figure 6.35, the RMS trends of the static and stagnation pressure and temperature are shown.

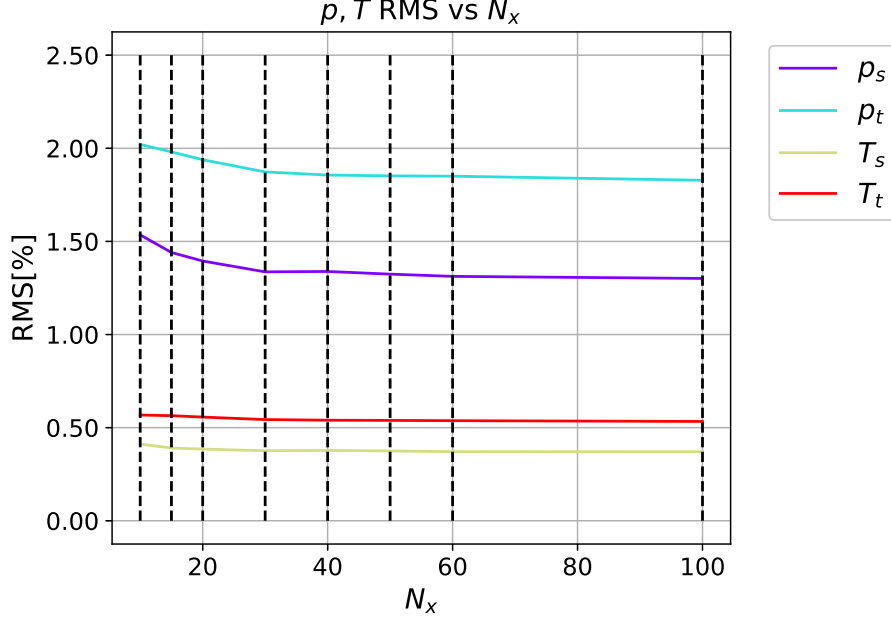


Figure 6.35: Static and stagnation pressure and temperature RMS errors

The RMS errors of these quantities were within the same order of magnitude, well below 5%. At the lower axial node counts of 10, 15 and 20, the static and stagnation pressure trends are visually declining. After an N_x of 30, all trends seem to remain nearly constant. This translates to mesh refinement levels above $N_x = 30$ resulting in little to no improvement in terms of overall result quality.

The RMS error for mass flux was overall about one order of magnitude higher compared to the RMS of pressure and temperature. In Figure 6.36, the RMS trend of the mass flux can be seen. Surprisingly, the RMS is actually lower for the coarser meshes, after which it maintains a near constant value of 21% after an N_x of about 30. This could also be seen in Figure 6.26, where the mass flux trend of $N_x = 10$ was quite distinct from the other curves, yielding an overall lower mass flux value.

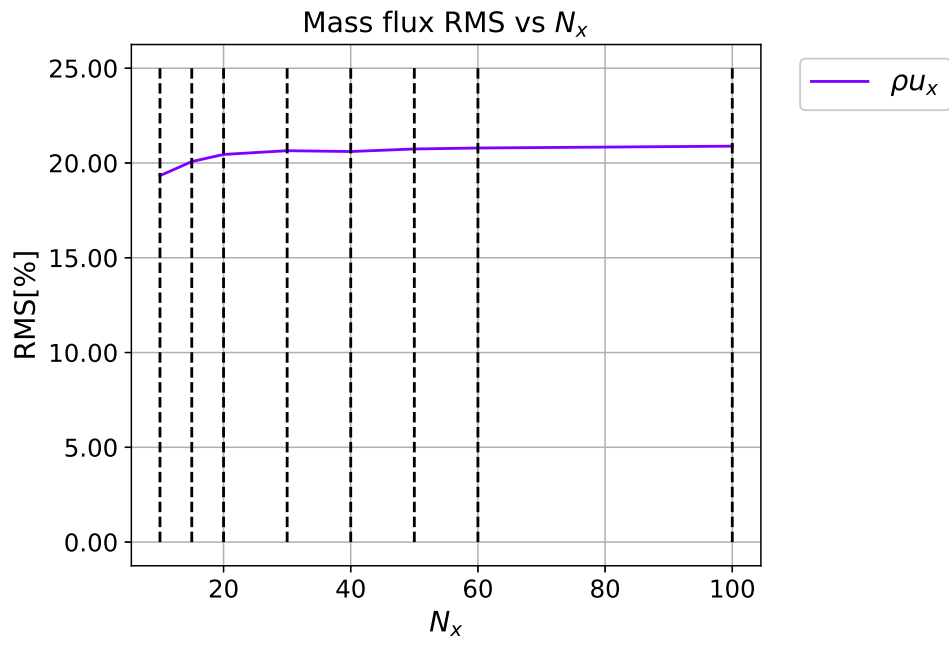


Figure 6.36: Mass flux RMS error trend

7 Conclusions

The goal of this research project was to set up a workflow which would enable design optimization using Body-Force Modeling in SU2. In order to do this, the existing 2D BFM was expanded upon by adding a parallel force formulation, which enabled the estimation of losses and a metal blockage source term, which took metal blockage into account. Additionally, the BFM had to be expanded to 3D to allow for higher fidelity. These added functionalities had to be properly tested to verify their performance. Additionally, the overall performance of the BFM was compared to the results obtained from blade computations. Based on the obtained results, the following conclusions were drawn.

1. **What level of accuracy in terms of loss generation and flow obstruction modeling can be achieved through implementation of respective models?**
 - (a) **With what level of accuracy does the 2D interpolation method interpolate the blade shape to the mesh?** The ray-cast interpolation algorithm implemented in SU2 was able to interpolate the blockage factor distribution during the blockage factor verification test case with an overall RMS error below 0.05%, which was considered acceptable for application in further simulations.
 - (b) **With what level of accuracy does the loss generation method approximate entropy generation with respect to theoretical values?** The parallel force model showed perfect agreement between the theoretical pressure and entropy gradient and the parallel force. This confirmed that the parallel force model was implemented correctly. When applied to the case of the Aachen turbine, the entropy generation was underestimated by the BFM by an almost constant factor. The relative increases in entropy for each blade row, with respect to the total entropy increase was captured accurately.
 - (c) **With what level of accuracy can flow obstruction due to blade thickness be simulated using metal blockage modeling?** The metal blockage model showed good agreement (mass flux RMS below 0.2%) with Euler simulation data of a symmetrical stator blade. When compared to the Aachen reference case, the flow rate was found to be overestimated by an almost constant 21%, although the shape of the trend was reproduced accurately by the BFM. It is speculated that this overestimation in flow rate is due to boundary layer blockage, which is not modeled by the metal blockage model. This hypothesis could be tested through repeating the symmetrical stator test case using a RANS solver and comparing the mass flux trends.

The additions to the BFM consisted of a 2D interpolation algorithm, a parallel force model for loss generation and a metal blockage source term to model flow obstruction due to blade thickness. Each of these additions was verified through respective test case simulations. Within these test cases, it was found that the parallel force model, 2D interpolator and blockage source model resulted in results within acceptable error margin.

2. **What are the pros and cons of body-force modeling in SU2 with respect to physical blade computations?**
 - (a) **What level of computational efficiency can be achieved through body-force modeling in SU2 compared to blade computations?** The test case used for answering this question was a 1.5 stage axial turbine called the Aachen turbine. BFM simulations were performed at varying levels of mesh refinement in order to check for grid convergence and observe the scaling of computation time with mesh refinement.

The BFM simulation performed on the mesh with the lowest refinement level took less than 30 seconds, while that of the highest mesh refinement level took 220 minutes, each on a single processor. Comparatively, the physical blade computation took about 54 hours on 20 processors. This meant that the BFM was between 2 and 6 orders of magnitude faster than the blade computation. The simulation at the level of grid convergence took 8 minutes, being approximately 8100 times faster than the blade computation.

- (b) **With what level of accuracy do the BFM results approximate those obtained through blade computation?** The BFM was able to reproduce the trends of static and stagnation temperature and pressure rather well, although the shape of the trend was somewhat different throughout the rotor row. In terms of entropy generation, the absolute values were off, but the relative increases in entropy per section with respect to the entropy increase throughout the turbine were captured accurately. The main differences were found in the mass flux and absolute flow angle. The mass flux was overestimated by the BFM by an almost constant 21%, while the shape of the mass flux trend was captured accurately. The outlet flow angles of the stator rows were underestimated by 5° and for the rotor by 17° . The BFM is therefore capable of reproducing absolute performance in terms of pressure, temperature and work rather well. The absolute values for mass flow rate and efficiency are off, but can be evaluated in a relative sense. The results on flow deflection yield the highest degree of uncertainty and adjustments would have to be made to the BFM in order to reproduce the absolute flow angle more accurately.
- (c) **What level of mesh refinement is required for the BFM to reach grid convergence?** The mesh parameter used in this research was the number of axial nodes across each blade row section. An equal number of nodes was applied in radial direction. No visible changes in RMS for several flow variables were found beyond an axial node count of 30, at which point grid convergence was reached. This translates to an average cell cross-sectional area of 3.92 mm^2 . The number of cells in tangential direction did not affect the results. Therefore, meshes using a single cell in tangential direction of arbitrary width can be used in axisymmetric BFM simulations.

The BFM was found to be significantly more efficient in terms of computation time (8100 times faster) with respect to the physical blade computation of the same geometry. At grid convergence, the BFM converged to a density residual of 10^{-16} in 8 minutes and 25 seconds on a single processor for a structured grid. This allows for multiple stages to be optimized for a high number of design variables using limited computational resources in less time compared to RANS CFD analysis methods. In terms of accuracy, deviations in absolute values for several flow variables were found frequently. However, the trends were often in agreement with those of the blade computation results. This led to believe that the BFM would be a powerful tool in turbomachinery design. During conceptual design, relative performance has to be estimated accurately, as the objective function is often a parameter normalized with respect to a reference value. In order to conclusively answer this question, design sensitivity results between the BFM and physical blade simulation results would have to be compared.

8 Final Remarks and Perspectives

In this chapter, the quality of the results is discussed, as well as the soundness of the recommendations. Additionally, recommendations are made for future research and improving the quality of the efforts made in this research.

8.1 Final Remarks

The goal of this research was to evaluate the application of body-force modeling in the context of design optimization. One rather obvious test for this would be to compare the design sensitivities resulting from blade computations to those produced by the BFM. Unfortunately, partially due to logistical issues involving the ongoing pandemic, the computational resources required were unavailable for the majority of the research period. It was in the final weeks that access was available to the software to produce the meshes required for blade computations. The blade computation data on the Aachen turbine was readily available, as the simulation data was generated in other research. Setting up the automated work flow was an important step in streamlining the design process using body-force modeling, but it doesn't support any hypothesis on whether body-force modeling is or is not a viable design tool.

When comparing the flow variable trends of the blade computation and BFM, there were similarities and differences, especially throughout the rotor blade row. The shape of for example the static and stagnation pressure in this region was quite off, even though the values at the rotor outlet agreed pretty well. In previous work involving the BFM formulation used in this research, the similarity with blade computations was higher. In Thollet's PHD thesis [2], none of the BFM resulting trends exceeded the error margin of 5% with respect to blade computation results and even the flow rate computed by the BFM was comparable to that found through blade computation. In terms of flow deflection, the BFM results in this thesis were off quite a bit at the rotor outlet. This was not observed in Thollet's BFM results of intake fans. It could be that this BFM formulation is not as suitable for turbines compared to intake fans. The scalar factor used for determining the friction factor could for instance be recalibrated to yield better results regarding entropy generation and perhaps a different formulation for the compressibility factor should be utilized in such cases. Additionally, Thollet made use of a RANS solver combined with the BFM in his work, while an Euler solver was used in this research. It could be that turbulence affects the solution in such a way that the trends compare better and it would have been interesting to compare the results obtained by the Euler solver to those produced through RANS computation.

8.2 Perspectives

- **RANS Solver Implementation:** The first recommendation is based on the discussion on flow solvers in the previous section. At the start of this research project, the plan was to compare different flow solvers in combination with the BFM to make a trade-off between accuracy and computational efficiency. Time and circumstances prevented this from being set up. With relatively minor adjustments to the existing BFM, it should be possible to allow for body-force modeling using a RANS solver in SU2. These adjustments include the functions defining the body-force and metal blockage source terms to be defined within the RANS solver structure in SU2.
- **Increase Interpolation Efficiency:** The interpolator function used for the interpolation of the body-force parameters onto the mesh nodes was found to be lacking in terms of

efficiency. Currently, the interpolator loops over all the nodes in the mesh and loops over the points defined in the BFM input file within this loop. Although this method works for coarse meshes, it could become computationally intensive for full-annulus simulations for example. In order to resolve this, the interpolation could be performed prior to the solution process. Efficient interpolation functions such as the "griddata" function in Python could be used to interpolate the body-force parameters onto the mesh coordinates after mesh generation. By directly storing these interpolated data points into the BFM input file, the values could directly be read from the file in SU2 and stored in a single for-loop.

- Coupling with Adjoint Solver:** As was mentioned in Chapter 2, the combination of body-force modeling and adjoint solving could be very powerful for design purposes. To facilitate this, the adjoint solver native to SU2 would have to be able to use the camber normal vectors and blockage factors as input, instead of mesh coordinates. In the final weeks of this research project, attempts were made to get this functionality to work, but was not ready in time to be published in this work, although the method is believed to be close to being operational. If work continues on coupling these methods, the design sensitivities produced by the BFM-adjoint method could be compared to those produced by physical blade computations. If these are found to be comparable, the method is believed to have considerable potential in the optimization of fans with distorted inlet conditions, as the nearly unlimited design freedom the adjoint method provides would be combined with the high computational efficiency of the BFM.
- System-Level Solving:** Another interesting expansion on the BFM implemented into SU2 would be to facilitate flow simulations of the complete turbine or compressor. By adding a combustion chamber model and placing it in between a compressor and turbine, both represented by the BFM, it would allow for the analysis of the effects of upstream components on downstream components, while limiting the associated computation costs. Turbine and compressor rows can already be placed in succession of one another with the current BFM formulation. The only addition would be a combustion chamber model. It is expected that this addition would not be very challenging, as it would involve a volumetric mass and energy source field in the region of the combustion chamber.
- Aeroelastic Analysis and Design:** As was mentioned in Chapter 2, the BFM allows for different blade geometries to be analyzed on the same mesh. One interesting application of this property would be aeroelastic simulations. Here, the need of remeshing after each iteration would be eliminated, as the blade can move independently of the mesh. The body-forces could directly be applied to a finite-element method which calculates the respective deformation of the blade. Recalculating the distribution of camber normal vectors and blockage factors and the interpolation process is computationally less intensive compared to remeshing, especially when a more efficient interpolation function is used. This would especially be interesting for BLI design cases, as blade flutter is more likely to occur for distorted inlet conditions.

A Design Work Flow Demonstration

In this appendix, the BFM design workflow outlined in Chapter 3 is demonstrated by example of an axial intake fan. The steps taken by the method, as well as the most relevant user inputs are provided in order to give the user an idea of the possibilities regarding design variables. The tools used within this workflow can be downloaded from GitHub using the following link:

[git@github.com:EvertBunschoten/Meangen2BFM.git](https://github.com:EvertBunschoten/Meangen2BFM.git)

A.1 Design variable selection

In order for the BFM to work properly, it requires an axial-radial distribution of camber normal and blockage factor values, as well as the annulus shape which defines the domain boundaries. The shapes of the blades and annulus can be defined through numerous design variables. This can be problematic during conceptual design, when the overall shape of the annulus and blades have not been defined yet. During conceptual design, the number of design variables is often limited to a few parameters such as the three duty coefficients (R^* , ϕ , ψ).

Table A.1: Design variable selection for demonstration compressor case

Design variable	Value
n_{stage}	1
Machine type	Compressor
\dot{m}	200 kg s ⁻¹
Ω	4000 rpm
R_{gas}	287.15 J kg ⁻¹ K ⁻¹
γ	1.4
$p_{t,\text{in}}$	1.5 bar
$T_{t,\text{in}}$	280 K
p_{out}	1.89 bar
$N_{\text{b, rotor}}, N_{\text{b, stator}}$	30, 50
R^*	0.75
ϕ	0.6
ψ	0.5
ξ	0.0
r_{design}	0.70 m

Table A.2: *Parablade* thickness parameters

Thickness parameter	Rotor	Stator
t_{LE}	0.003	0.003
t_{TE}	0.005	0.005
d_1	0.40	0.40
d_2	0.35	0.35
t_1	0.015	0.015
t_2	0.017	0.017
t_3	0.019	0.019
t_4	0.024	0.024
t_5	0.030	0.030
t_6	0.020	0.020

The design variables listed in Table A.1 are used by *Meangen* to calculate the blade inlet and outlet metal angles at three spanwise sections for each blade row. These angles are calculated using the values specified for R^* , ϕ , ψ and ξ , where ξ is a twist parameter which allows for vortex-free design. A ξ value of 0.0 translates to a prismatic blade, while 1.0 results in a twisted blade geometry compatible with vortex-free design. For more detailed design, the user can also specify values for leading and trailing edge deviation angles and sweep angles for the leading and trailing edge.

The annulus shape is determined by *Meangen* through estimation of the static density and prescribed mass flow rate. The blade metal angles are also calculated by *Meangen*. With the user input as listed in Table A.1 and Table A.2, two blade rows are defined with the following geometric parameters. The leading and trailing edge radii, as well as the respective metal angles can be found in Table A.4 and Table A.5.

Table A.3: *Meangen* blade geometry output

Table A.4: Rotor row *Meangen* output

Table A.5: Stator row *Meangen* output

Meangen Output	Value	Meangen Output	Value
$r_{\text{hub, LE}}$	0.622 m	$r_{\text{hub, LE}}$	0.639 m
$r_{\text{hub, TE}}$	0.638 m	$r_{\text{hub, TE}}$	0.642 m
$r_{\text{shroud, LE}}$	0.777 m	$r_{\text{shroud, LE}}$	0.761 m
$r_{\text{shroud, TE}}$	0.762 m	$r_{\text{shroud, TE}}$	0.758 m
$\beta_{\text{m, LE}}$	-59.0°	$\beta_{\text{m, LE}}$	39.8°
$\beta_{\text{m, TE}}$	-39.8°	$\beta_{\text{m, TE}}$	0.0°

These geometric parameters, alongside the thickness parameters in Table A.2 are passed to *Parablade*, which generates the detailed blade shape. In Figure A.1 and Figure A.2, the 2D cross-sections of the rotor and stator blade rows can be seen.

Parablade also generates the input file containing information on the camber normal and blockage factor distributions. After generating an input file for each respective blade row, the files are combined into a single input file for the entire stage. This input file is used by the BFM while interpolating the camber normal and blockage factor values onto the mesh nodes.

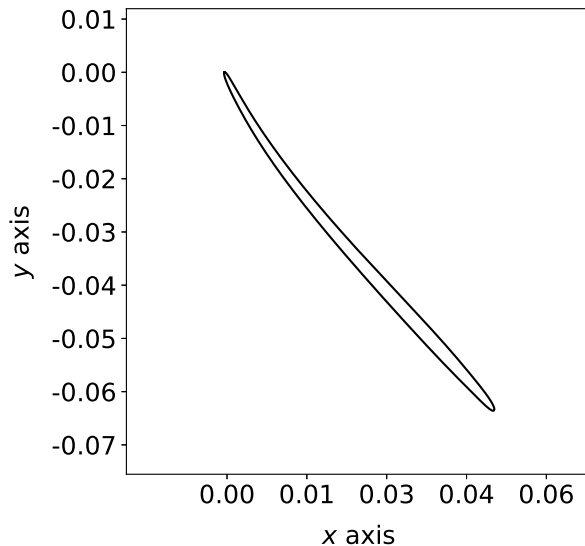


Figure A.1: Mid-span rotor cross section

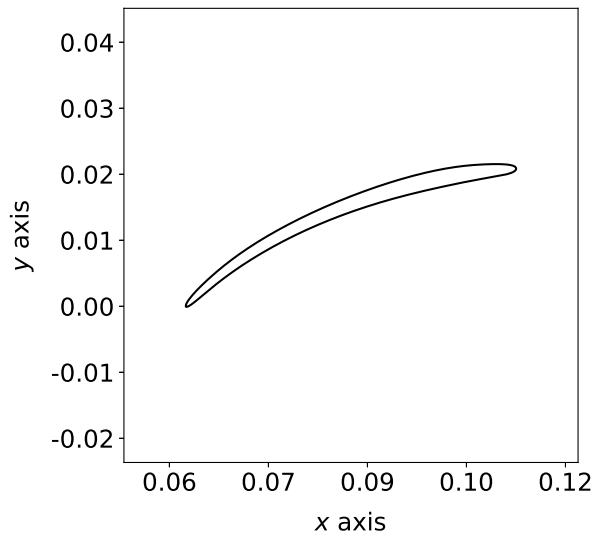


Figure A.2: Mid-span stator cross section

The final step in preparation to the solution process is mesh generation. Currently, the workflow is only able to generate unstructured meshes using *GMesh*. It is expected that in the near future, a structured meshing option will be implemented using *ANSYS ICEM* or the structured mesh options within *GMesh*.

An isometric view of the unstructured mesh for the compressor geometry in question can be seen in Figure A.3. The individual boundaries of the domain are colored and named accordingly. The mesh is kept relatively coarse near the inlet and outlet, while the body-force regions are refined. This can be seen more clearly in Figure A.4.

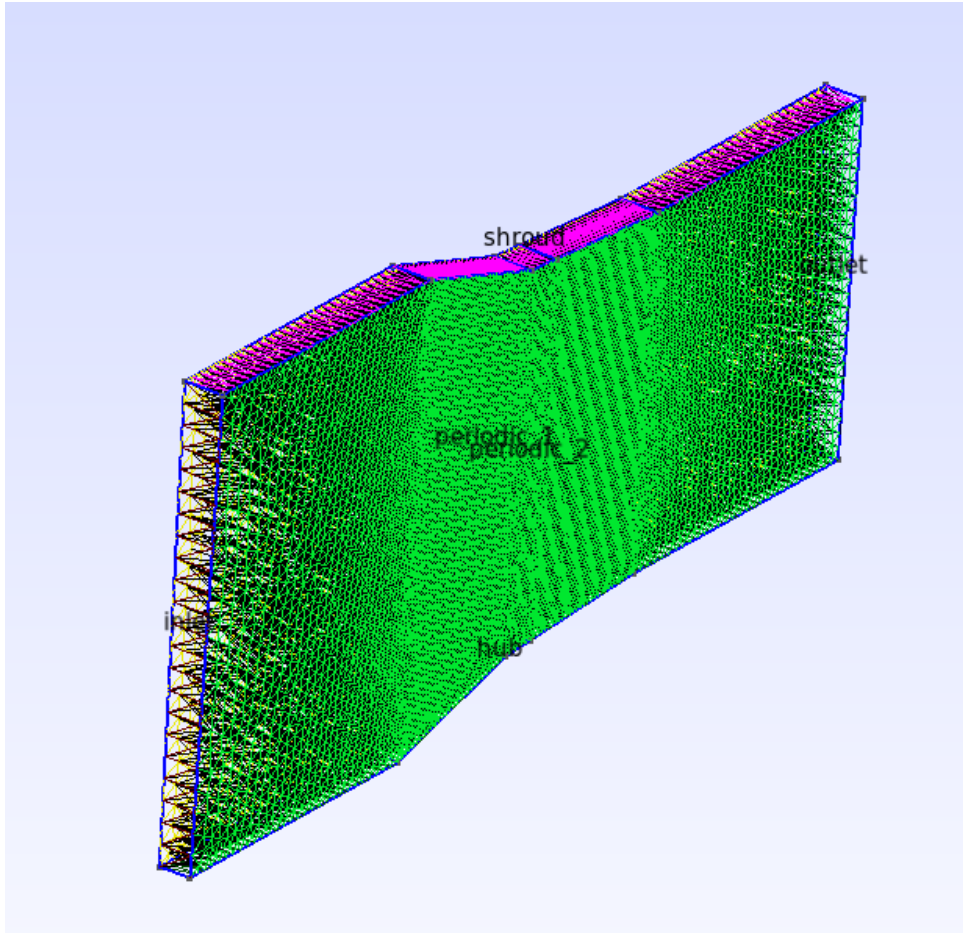


Figure A.3: Isometric view of unstructured mesh

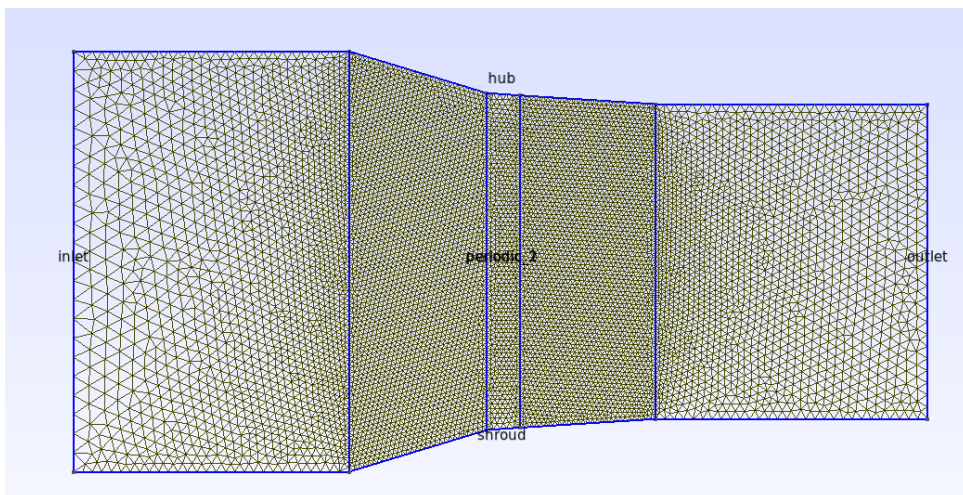


Figure A.4: Side view of unstructured mesh

After rewriting the mesh output by *GMesh* into a periodic mesh, the simulation can be initial-

ized. An SU2 configuration file is written, containing all the information regarding boundary conditions, input files and convergence parameters. In the workflow input file, the user can specify whether to run the simulation automatically or manually. If ran automatically, SU2 will initialize after writing the periodic mesh and SU2 input file and post-process the data once converged. If for instance different convergence parameters than the default values are preferred, the user can choose for the solver option to be manual. In this case, the workflow will terminate after writing the SU2 input file and mesh and allow the user to initialize SU2 manually.

After solving, the flow data is written in *Paraview .vtk* format. Currently, this is the only option regarding output file format, as the postprocessing function uses the *Paraview* Python environment. The postprocessor function creates a directory called 'Performance_Data', in which axial flow data and radial flow data trends are stored. Additionally, performance objectives such as total-to-total efficiency and power are calculated and stored in a file called 'Machine_Objectives.txt'. The mass-flux-averaged data trends are also plotted and saved as *.eps* figures. In Figures A.5, A.6, A.7 and A.8, the trends for absolute flow angle, entropy increase, stagnation pressure and stagnation temperature can be seen respectively.

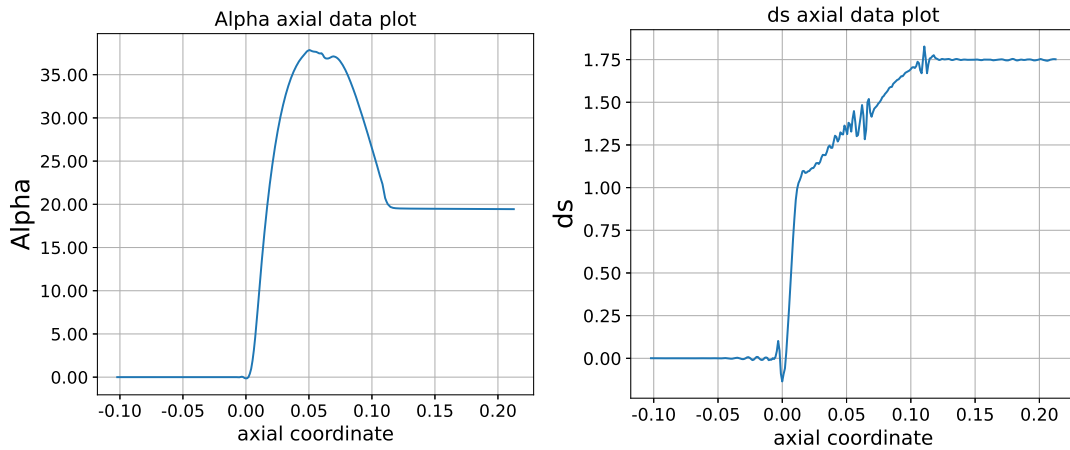


Figure A.5: Axial, mass-flux-averaged absolute flow angle trend

Figure A.6: Axial, mass-flux-averaged entropy increase trend

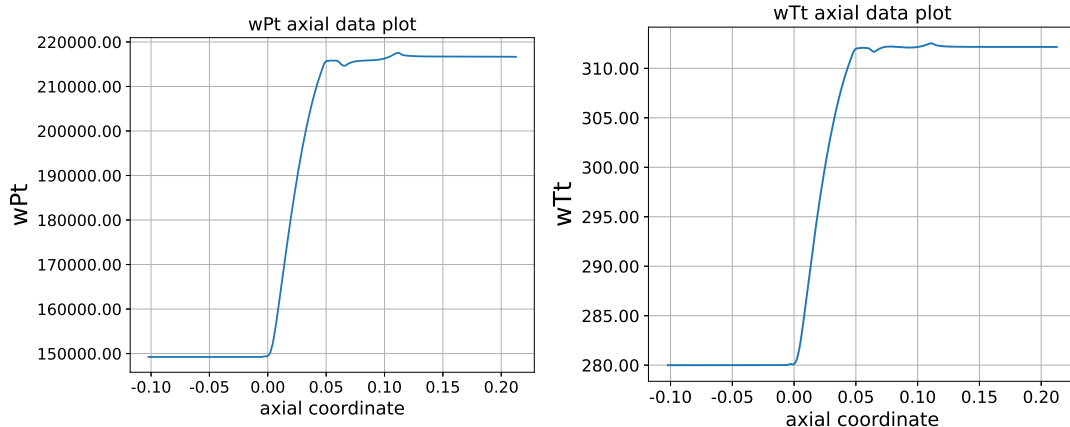


Figure A.7: Axial, mass-flux-averaged stagnation pressure trend

Figure A.8: Axial, mass-flux-averaged stagnation temperature trend

B Radial section data

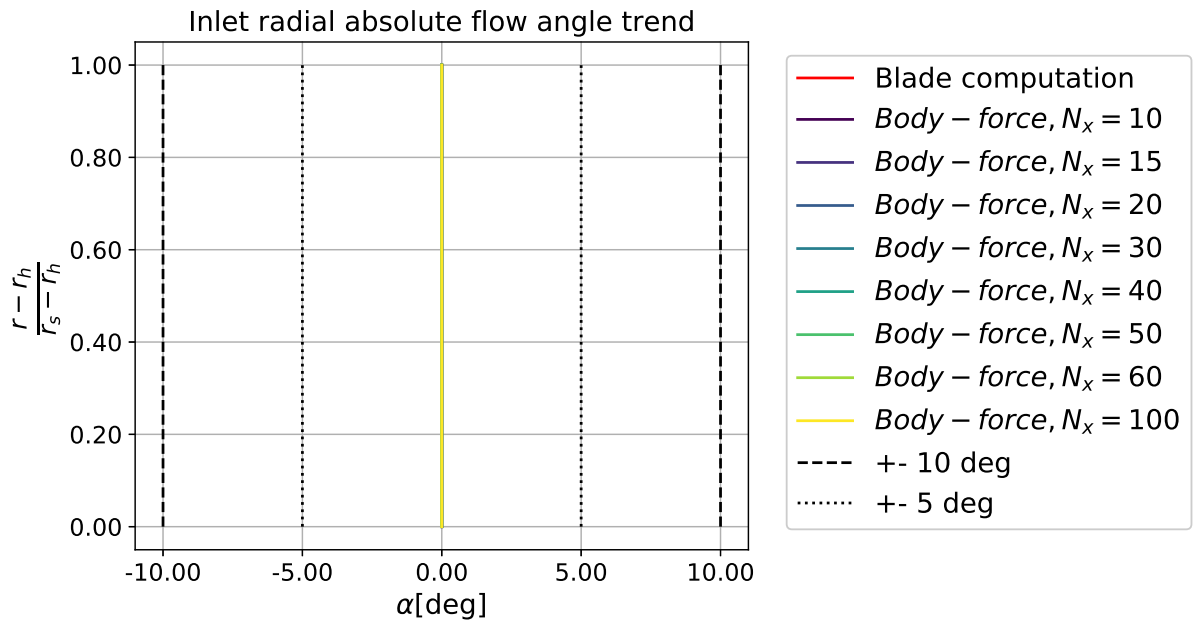


Figure B.1: Radial absolute flow angle near the inlet

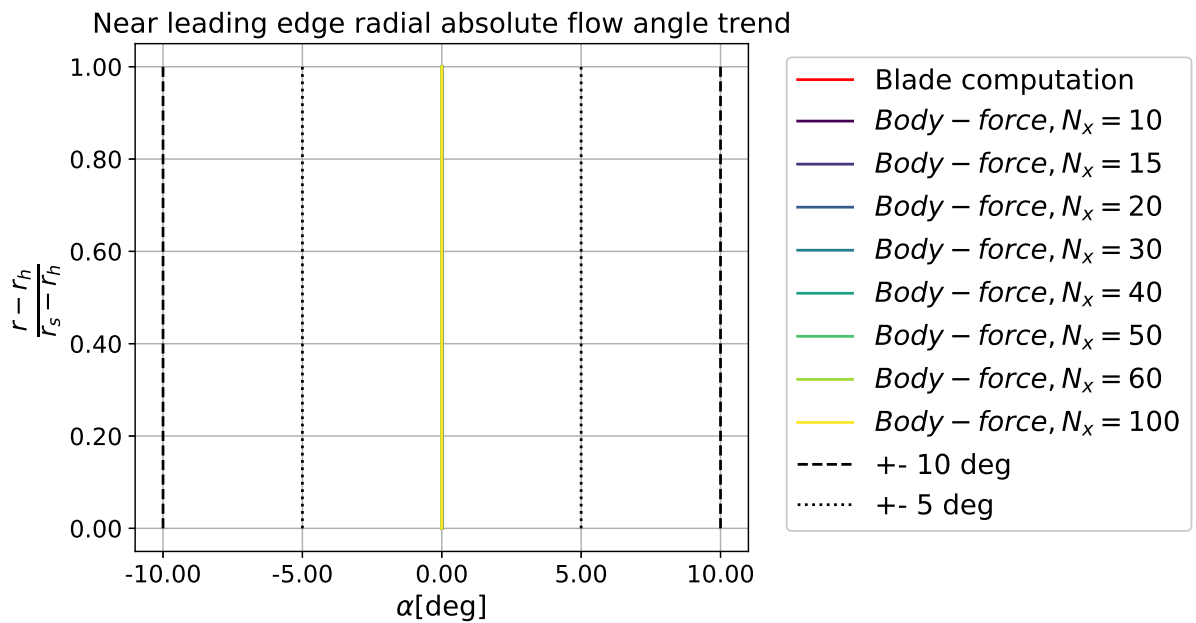


Figure B.2: Radial absolute flow angle near the first blade leading edge

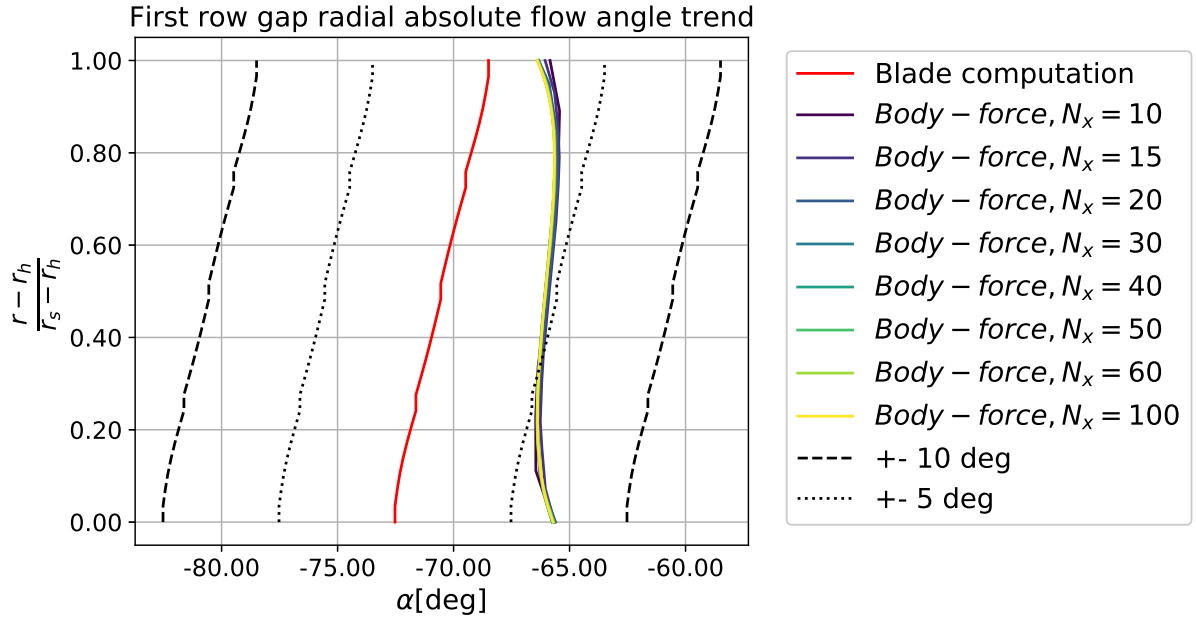


Figure B.3: Radial absolute flow angle in the first row gap

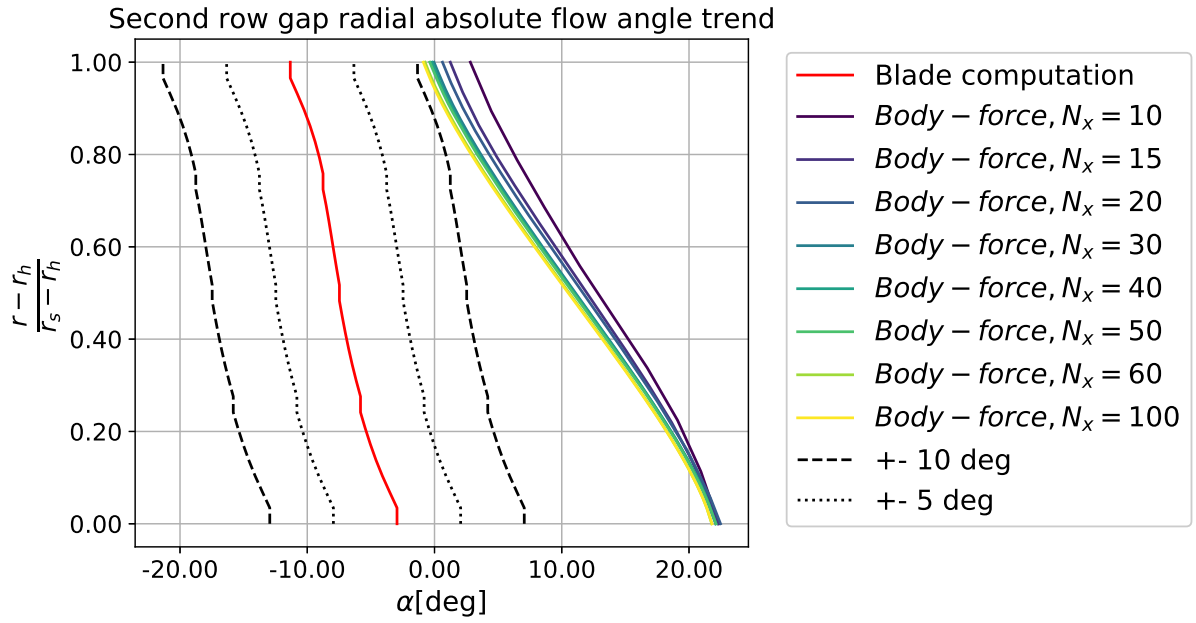


Figure B.4: Radial absolute flow angle in the second row gap

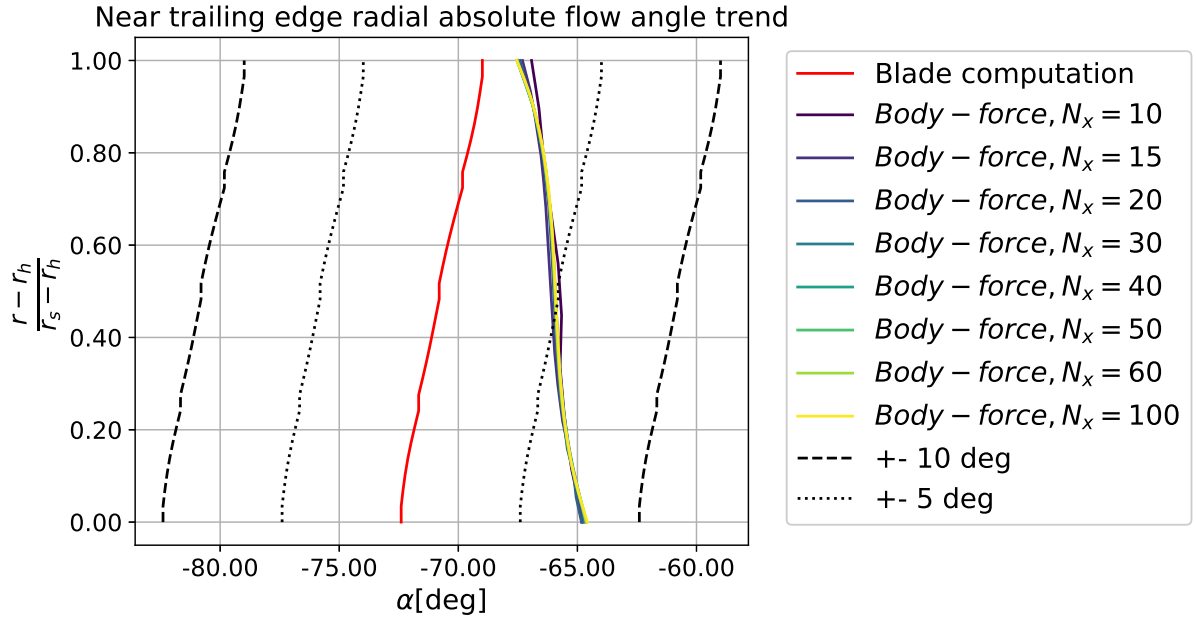


Figure B.5: Radial absolute flow angle near the last blade trailing edge

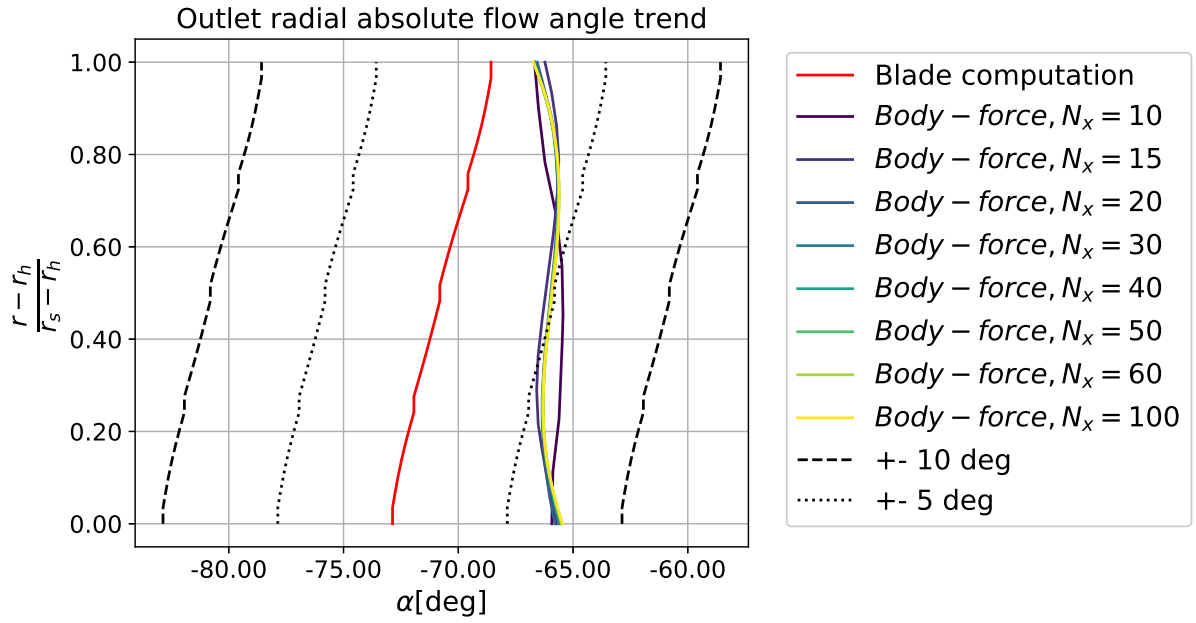


Figure B.6: Radial absolute flow angle near the outlet

References

- [1] D. K. Hall. “Analysis of civil aircraft propulsors with boundary layer ingestion”. PhD thesis. Massachusetts Institute of Technology, 2015, pp. 1–116. URL: <https://dspace.mit.edu/handle/1721.1/97353>.
- [2] W. Thollet. “Body force modeling of fan – airframe interactions”. PhD thesis. Université Fédérale Toulouse Midi-Pyrénées, 2017, pp. 1–157.
- [3] Olivier Atinault et al. “Numerical and experimental aerodynamic investigations of boundary layer ingestion for improving propulsion efficiency of future air transport”. In: *31st AIAA Applied Aerodynamics Conference* June 2013 (2013). DOI: 10.2514/6.2013-2406.
- [4] Alejandra Uranga et al. “Preliminary experimental assessment of the boundary layer ingestion benefit for the D8 aircraft”. In: *52nd Aerospace Sciences Meeting* January 2017 (2014). DOI: 10.2514/6.2014-0906.
- [5] Jin Guo and Jun Hu. “Development of body force model for steady inlet distortions in high-speed multistage compressor”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 231.9 (2017), pp. 1650–1659. ISSN: 20413025. DOI: 10.1177/0954410016656880.
- [6] M T Latour. “Modeling axial compressor fan blades using a body force model”. PhD thesis. Delft University of Technology.
- [7] F Marble. “Three Dimensional Flow in Turbomachines”. In: *High Speed Aerodynamics and Jet Propulsion* X (1964), pp. 83–166.
- [8] D. K. Hall, E. M. Greitzer, and C. S. Tan. “Analysis of fan stage conceptual design attributes for boundary layer ingestion”. In: *Journal of Turbomachinery* 139.7 (2017), pp. 1–10. ISSN: 15288900. DOI: 10.1115/1.4035631.
- [9] David J. Hill and Jeffrey J. Defoe. “Innovations in body force modeling of transonic compressor blade rows”. In: *International Journal of Rotating Machinery* 2018.1 (2018). ISSN: 15423034. DOI: 10.1155/2018/6398501.
- [10] Luis López de Vega, Guillaume Dufour, and Nicolás García Rosa. “A fully coupled body force-engine performance methodology for boundary layer ingestion”. In: *AIAA Propulsion and Energy Forum and Exposition, 2019* August (2019), pp. 1–15. DOI: 10.2514/6.2019-3828.