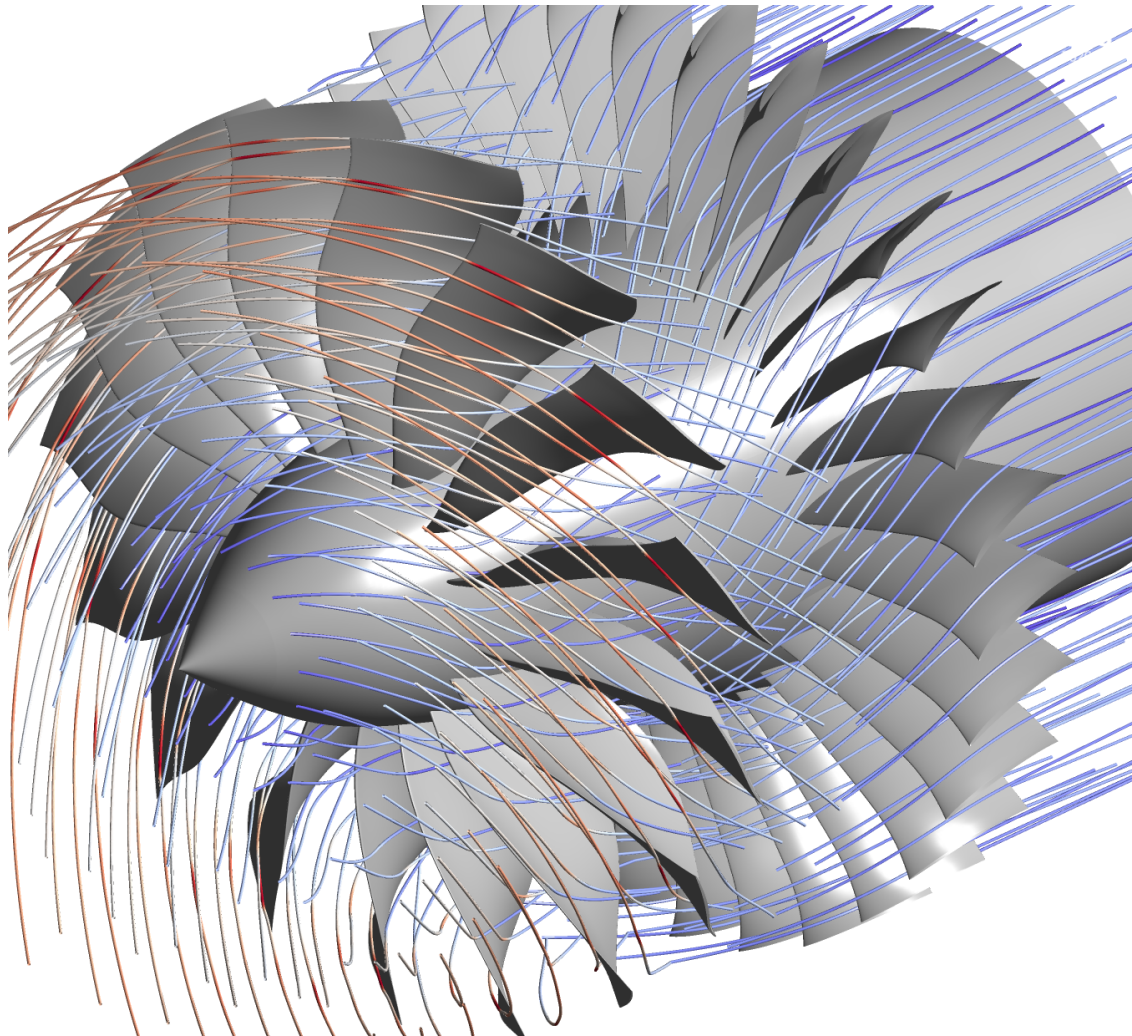




CHALMERS
UNIVERSITY OF TECHNOLOGY



Developing of a numerical framework for transonic axial compressor analysis

Master's thesis in Applied Mechanics

ROBERT HESSMAN RANMAN

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

MASTER'S THESIS 2024

**Developing of a numerical framework
for transonic axial compressor analysis**

ROBERT HESSMAN RANMAN



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Fluid Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Developing of a numerical framework for transonic axial compressor analysis
ROBERT HESSMAN RANMAN

© ROBERT HESSMAN RANMAN, 2024.

Supervisor: Oliver Sjögren, Department Of Mechanics and Maritime Sciences
Examiner: Carlos Xisto, Department Of Mechanics and Maritime Sciences

Master's Thesis 2024
Department of Mechanics and Maritime Sciences
Division of Fluid Mechanics
Name of research group (if applicable)
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Streamline representation of flow through CATANA fan design.

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Developing of a numerical framework for transonic axial compressor analysis
ROBERT HESSMAN RANMAN
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Modern engineering in the aerospace field of operations is transitioning from real experimental testing to heavily relying on computer-aided engineering (CAE). Computational fluid dynamics (CFD) is a great tool for obtaining estimates of real operations but computational limitations often lead to the implementation of simplified models of the full governing equations, rendering estimates with a limited accuracy. Validating numerical results to experimental data is a great way of bridging the gap between virtual and real operation, and further provides weight to the validity of the results to the aeronautical engineering community.

At *Ecole Centrale de Lyon* a carbon fiber fan stage has been designed as a reference test case for state-of-the-art fan stage technology, in collaboration with the engine manufacturer Safran. The test case named CATANA is intended as a platform for collaboration between universities and research agencies and to support validation of simulation codes by providing experimental data.

In this thesis, the numerical and experimental results generated at *Ecole Centrale de Lyon* on the CATANA test case is used to establish and validate a numerical framework developed to further be used in a parametric study of fan blade design. The framework is presented in three phases, grid generation, simulations, and validation with CATANA data. The baseline case is established with a low Reynolds number RANS turbulence model with the assumption of steady-state.

To further reduce the overall computational load of any given blade design the spacial resolution requirements of a low and high Reynolds's number turbulence models are investigated by integrating a low-Re $k-\varepsilon$ model in the framework in parallel to a high-Re $k-\omega$ SST model. The thesis covers an in-depth presentation of meshing routines and numeric setup for both models as well as the implementation of custom external convergence criteria. As a final part of the thesis, the effects of removing the stator from the fan stage are investigated by comparing spanwise distributions and mass flow averaged data downstream of the rotor.

Keywords: compressor, rotor, stator, wall-functions, RANS, convergence, CFX, Turbogrid, Pointwise

Acknowledgements

To start of with i would like to express my gratitude for the opportunity to working in a project closely in line with my primary field of interest and with such relevance to my academic track within the masters program of applied mechanics. For this i extend my gratitude towards Oliver sjögren for offering me this thesis and to Carlos Xisto for agreeing to be my examiner. Further i would like to extend this gratitude to Oliver for his weakly advice and counsel throughout the duration of this project. Lastly i would like to express my gratitude towards Anders at GKN Aerospace for attending a collective meeting each weak and showing interest and offering his guidance to my work.

Robert Ranman, Gothenburg, 02 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ACA	Area Circumferential Average
APG	Adverse Pressure Gradient
CFX	Computational Fluid Dynamics
DF	Diffusion Factor
DNS	Direct Numeric Simulation
FSI	Fluid Structure Interaction
GCI	Grid Convergence Index
LES	Large Eddy Simulation
MFA	Mass Flow Average
ML	Machine Learning
OGV	Outlet Guiding Vanes
PR	Pressure Ratio
RA	Rolling Average
RANS	Reynolds Averaged Navier Stokes
RE	Rotor Exit
RMS(D)	Root Mean Square (Deviation)
SE	Stage Exit
SGS	Sub Grid Scale
SST	Shear Stress Transport
TR	Temperature Ratio
URANS	Unsteady Reynolds Averaged Navier Stokes

Nomenclature

Below is the nomenclature that have been used throughout this thesis unless otherwise specified in the text.

Roman upper case letters

S_{ij}	strain rate tensor
T	static temperature
T_0	total temperature
\tilde{T}_0	reference total temperature
P_k	turbulent production rate
\tilde{P}_0	reference total pressure
M	Mach number
U	mean velocity field

Roman lower case letters

\mathbf{u}	velocity vector
u_i	velocity component
u_τ	frictional velocity
u^+	velocity scale in standard wall functions
u^*	velocity scale in scalable wall functions
h	static specific enthalpy
h_0	total specific enthalpy
i	incidence angle
\dot{m}	mass flow
p	static pressure
p_0	total pressure
k	turbulent kinetic energy
k_n	averaging window
y^+	dimensionless wall distance
\bar{y}^*	limited dimensionless wall distance
x_2^+	dimensionless wall distance

Greek letters

γ	heat capacity ratio
η	efficiency
δ_{ij}	Kronecker's delta
μ	dynamic viscosity
μ_t	turbulent (eddy) dynamic viscosity
ν	kinematic viscosity
ν_t	turbulent kinematic viscosity
ρ	density
ε	turbulent dissipation rate
ω	turbulence frequency
τ_w	shear stress at wall
κ	von Karman constant
τ_w	wall shear stress

Contents

List of Acronyms	x
Nomenclature	xiii
List of Figures	xvii
List of Tables	xix
1 Introduction	1
1.1 Aim	1
1.2 Limitations	2
1.3 Literature Review	2
2 Theory	5
2.1 Governing Equations	5
2.1.1 k - ε Turbulence model	5
2.1.2 k - ω SST Turbulence model	7
2.2 Procedure for Estimation of Discretization Error	9
2.3 Aerodynamic Instabilities and High Speed Effects in Compressors	10
3 Methods	13
3.1 Baseline Case	13
3.2 Mesh Generation	14
3.3 Boundary Conditions	18
3.4 Computational Fluid Dynamics	20
3.4.1 Solver Settings	20
3.4.2 Convergence Criteria	21
4 Results	25
4.1 Grid Independence Study	26
4.1.1 Low-Re model grid	26
4.1.2 Wall function grid	29
4.2 Design Point Analysis	32
4.3 Off-Design Performance	36
4.4 Stand alone Rotor	37
4.5 Temporal Gain Relative Numerical Model and Compressor Core Configuration.	39

5 Conclusion	41
5.1 Thoughts About Current Work	41
5.2 Further Discussion on Rotor Independence (ML Estimates)	42
5.3 Future Work	44
Bibliography	45
A Appendix 1	I
B Appendix 2	III

List of Figures

3.1	Machine core view of the annular with integrated measurement planes for performance analytics. IN-stage inlet, RE-rotor exit, SE-stage exit	13
3.2	Rotor, initial topology and blade surface mesh.	14
3.3	Blade surface mesh with edge split.	15
3.4	Rotor tip topology definition.	15
3.5	Inlet grid.	17
3.6	Assembled mesh with inlet, rotor and stator domain.	17
3.7	Meridional view of the computational domain, illustrating assigned boundary conditions.	18
3.8	Top view of the computational domain illustrating the boundary conditions applied to the tangential patches.	18
3.9	Test run of convergence algorithm with a tolerance of $1E - 4$ for Pressure ratio (PR), mass flow rate (\dot{m}), and isentropic compression efficiency (η_c).	22
3.10	CFD flow chart.	23
4.1	Variance in design parameters relative the representative grid size, h , for pressure ratio, mass flow rate, total to total isentropic efficiency, and temperature ratio.	27
4.2	y^+ distribution at rotor suction side and hub surface for low-Re model grid.	29
4.3	Variance in design parameters for wall function mesh relative to the representative grid size, h , for pressure ratio, mass flow rate, total to total isentropic efficiency, and temperature ratio.	30
4.4	y^+ distribution at rotor suction side and hub surface for high-Re (wall function) model grid.	32
4.5	Radial distributions of normalized total pressure, total temperature, and exit flow angle α [deg]. Measured at the rotor exit (RE) plane (see Figure 3.1).	33
4.6	Relative Mach number number at 10/50/90 % span.	34
4.7	Static pressure profiles at 10/50/90 % span.	35
4.8	Off-design performance (speedline) for pressure ratio and mass flow rate over the CATANA fan stage.	36
4.9	Relative mach number at 90%-span, stall operating point.	37
4.10	Off-design performance (speedline) for pressure ratio and mass flow rate over rotor row.	38

4.11	Radial distributions at rotor exit (RE) line for normalized pressure ratio, normalized temperature ratio, and exit flow angle α	39
5.1	Liner-SVR ML-algorithms, trained and tested on data obtained from full stage Low- and High-Re models.	43
5.2	Liner-SVR ML-algorithms applied to CATANA experimental and URANS data compared to rotor only cases.	43

List of Tables

3.1	Design point data for baseline ECL5/CATANA test rig.	13
3.2	Baseline Mesh Parameters.	16
3.3	Boundary conditions.	19
3.4	Interface models applied at transition planes.	20
3.5	Solver settings.	21
4.1	Low-Re model grid data	26
4.2	Low-Re model discretization error.	28
4.3	Final grid cell count for configuration with rotor and stator, and stand alone rotor, low-Re grid.	28
4.4	High-Re model grid data.	30
4.5	High-Re model discretization error.	31
4.6	Final grid cell count for configuration with rotor and stator, and stand alone rotor, wall function grid.	31
4.7	CATANA fan operation point data.	32
4.8	Average iteration time relative compressor core configuration and nu- merical setup.	40

1

Introduction

Air-breathing jet engines operate by imparting and extracting energy from surrounding fluid by means of compression and expansion. These processes are almost exclusively achieved with turbomachines such as fans, compressors, and turbines. A fundamental part of a turbomachine is the rotor that transfers energy from a shaft to the fluid, or vice versa, through interactions between the fluid and moving blades. In high-bypass turbofan engines, the propulsive efficiency and generated thrust is therefore related to the work imparted by the fan stage on the fluid and its efficiency in doing so. To further reduce the overall environmental impact of the engine, size and weight are limiting aspects in the design process impacting the overall drag and weight of the aircraft. Fan stages with high efficiency and high mass flow per unit area are therefore desired while at the same time minimizing weight and ensuring sufficient aerodynamic stability. Computational fluid dynamics (CFD) and structural analysis with the finite element method (FEM) are widely utilized in the aeronautical field to analyze and optimize performance for this purpose. Computational methods rely on assumptions and models of real physical conditions and a degree of accuracy between actual and simulated performance is introduced. Experimental data is therefore important to aid engineers in development and validation of fan blade designs. At *Ecole Centrale de Lyon* a carbon fiber fan stage has been designed as a reference test case for state-of-the-art fan stage technology, in collaboration with the engine manufacturer Safran. The test case named CATANA is intended as a platform for collaboration between universities and research agencies and to support validation of simulation codes by providing experimental data.

1.1 Aim

With access to experimental data and geometry provided by CATANA the aim of this project is to validate and develop a CFD based framework for fan stage design and performance analysis. With reference CATANA blade design and experimental data as a starting point, the initial aim of the project is to develop and validate a numeric framework for performance of any fan blade design suitable for large scale applications such as optimization. Important aspects such as temporal scale and general data limitations are therefore to be considered when establishing the framework. Challenging aspect in developing the framework is to figure out at what scale in the flow field performance defining phenomena occur i.e. a framework that can accurately capture large scale performance analytics that can propagate from a much smaller scale in the flow field with the least amount of spacial resolution.

If further time is available a parametric study is to be carried out evaluating interdependencies between efficiency, flow per unit area, aerodynamic stability, and specific work input with established framework.

1.2 Limitations

The flow will be solved assuming steady-state in frame of references relative to the blade passages. The assumption of steady-state is a simplification of the actual flow field and leads to worse estimates of stall but reduces time needed to reach a converged solution. Further the system is considered adiabatic. The second limitation is to restrict the resolution of the turbulent properties of the flow to RANS scale, i.e fully modeling the turbulent length scales according to the Boussinesq assumption with the introduction of the superimposed eddy viscosity. The last limiting factor is to not consider the structural validity and deformations of the optimized blade design by the means of structural analysis or fluid structure interaction (FSI). The impact of geometric deformations during operation is therefore not considered.

1.3 Literature Review

Previous experimental work and data from the creator of the fan design [1] is to be utilized in the project to compare and validate the framework. The article presented at the ASME conference of 2023 compares models to experimental data including performance data of relevance to this project. Performance data contributed to the article suggests that a steady state RANS model for the turbulent flow is sufficient for numeric investigating of the design and is suited for the problem at hand. Further validation of the CFD solution with experimental data will be supported by general guidelines from the literature [2].

In regards to reducing the overall computational load the difference between implementing a high- reps. low-Reynolds number model is investigated. In [3], an article primarily investigating the effects of tip clearance flows in transonic compressors, but relevant with its implementation of wall functions. The authors utilised the realizable $k-\varepsilon$ turbulence model with wall functions and Chien's low Reynolds number model comparing the two. In the section covering the computation grid generated for wall functions it is stated that the y^+ resolution at the walls is kept around 30 allowing the solver to stay in the logarithmic region of the boundary layer, allowing for some variation in y^+ pending on the operating point. The final optimized wall function grid consisted of 373008 grid cells while the higher resolution low Reynolds number grid consisting of 1701280 grid cells, illustrating that the implementation of wall functions can reduce the overall computational load substantially. Comparing total pressure ratio data at operating points following a compressor map for both cases it is shown that the low Reynolds model induces a higher overall pressure ratio converging with the high Reynolds number data approaching stall but diverging approaching choke. It is further illustrated that the Chien low Reynolds number model renders results closer to experimental data but indicating the general trend

of underestimating the pressure ratio. The conclusion of the authors is that both models underestimate the total pressure ratio compared to experimental results, but is still able to capture the overall trend of the compressor map, and in terms of polytropic efficiency both models agree well with experimental results.

In regards to wall functions in the presence of shocks on the blade surface, an interesting article is written by P. Catalano [4]. In the article he evaluates the commonly used $k - \varepsilon$ and $k - \omega$ turbulence model for transonic flow over an airfoil with the presence of a shock. Both models is used with wall functions and wall integration for a set of three grid resolutions. From the illustrated data it can be seen that a low resolution wall function $k - \varepsilon$ turbulence model renders results with minor discrepancies to that of a high resolution wall integrated model and experimental data in terms of pressure coefficient distribution. It can further be seen that in terms of pressure recovery downstream of the shock the high Reynolds number model renders results with minor differences compared to the low Reynolds number model.

An extension to P.Catalanos work in [4], is an article he wrote with P.L. Vitagliano [5]. In the article, they explore the implementation of a scalable wall function approach for high Reynolds number flow, derived from the logarithmic region of the boundary layer. In case of fluid separation, the standard wall functions struggle to predict the flow downstream of the separation point since the utilized velocity scale becomes singular. The scalable wall functions remedy this by reformulating the velocity scale making it suitable for high-Re flows, flowing in the direction of an adverse pressure gradient, typically the case of a high speed compressor. The author was able to correctly predict separation using the scalable wall function approach relative to a low-Re wall integration model, for shock-induced separation in transonic flow over an airfoil. The results indicate that the scalable wall functions could potentially be applied to the current compressor configuration for this thesis.

2

Theory

2.1 Governing Equations

Steady-state simulations used to render the data presented in this thesis are solved using the pressure-based ANSYS CFX solver for a compressible flow field. Additionally, the k - ω SST (two-equation) turbulence model is applied to render data for a low Reynolds number grid, and the k - ε (two-equation) turbulence model is applied to a high Reynolds number grid. CFX solves equations for the time-averaged mass, momentum, and energy together with respective transport equations of turbulent properties associated with the respective model. The system of equations can be expressed in the general form,

$$\frac{\partial \mathcal{Q}}{\partial t} + \frac{\partial \mathcal{F}_j}{\partial x_j} = \mathcal{H} \quad (2.1)$$

Where \mathcal{Q} is the state vector, \mathcal{F}_j is the flux vector, and \mathcal{H} is the source term vector. A thermally perfect Newtonian fluid is assumed in the following sub-sections, and buoyant forces are neglected.

2.1.1 k - ε Turbulence model

The two governing equations in addition to mass, momentum, and energy in the k - ε turbulence model are equations for the turbulent kinetic energy, k , and turbulent dissipation rate, ε . Rendering the state vector \mathcal{Q} (Reynolds decomposed equations),

$$\mathcal{Q} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho} \bar{u}_i \\ \bar{\rho} \bar{h}_0 - \bar{p} \\ \bar{\rho} \bar{k} \\ \bar{\rho} \bar{\varepsilon} \end{bmatrix} \quad (2.2)$$

Flux vector \mathcal{F}_j ,

$$\mathcal{F}_j = \begin{bmatrix} \bar{\rho} \bar{u}_j \\ \bar{\rho} \bar{u}_j \bar{u}_i + \bar{p} \delta_{ij} - \tau_{ij} + \overline{\rho u'_j u'_i} \\ \bar{\rho} \bar{u}_j \bar{h}_0 - \lambda \frac{\partial T}{\partial x_j} - \frac{\mu_t}{Pr_t} \frac{\partial \bar{h}}{\partial x_j} - \bar{u}_i \tau_{ij} + \bar{u}_i (\overline{\rho u'_j u'_i}) \\ \bar{\rho} \bar{u}_j k - \left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \\ \bar{\rho} \bar{u}_j \varepsilon - \left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \end{bmatrix} \quad (2.3)$$

Source term vector \mathcal{S} ,

$$\mathcal{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ P_k - \rho\varepsilon \\ \frac{\varepsilon}{k}(C_{\varepsilon 1}P_k - C_{\varepsilon 2}\rho\varepsilon) \end{bmatrix} \quad (2.4)$$

The Reynolds decomposition of mass, momentum, and energy equations introduces a closure problem in the sets of equations as a new term appears called the Reynolds (turbulent) stresses. Eddy viscosity-based turbulence solver aims to model these stresses by introducing an additional set of equations(s), in this case, two, and solving for the modeled property μ_t , known as the eddy (turbulent) viscosity. The modeled eddy viscosity is introduced in the momentum equations with the Boussinesq assumption, replacing the unknown turbulent stresses with the turbulent viscosity, resulting in the definition for the modeled Reynolds stress equation,

$$-\overline{\rho u'_i u'_j} = \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} + \frac{2}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \bar{\rho} k \delta_{ij} \quad (2.5)$$

The k - ε model assumes that the eddy viscosity is linked to the turbulent kinetic energy and turbulence dissipation rate via the relation,

$$\mu_t = C_\mu \rho \frac{k^2}{\varepsilon} \quad (2.6)$$

With the calculated eddy viscosity, the turbulent production, P_k , can be solved using the following equation,

$$P_k = \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} - \frac{2}{3} \frac{\partial \bar{u}_k}{\partial x_k} \left(3\mu_t \frac{\partial \bar{u}_k}{\partial x_k} + \rho k \right) \quad (2.7)$$

The standard k - ε turbulence model with wall functions classifies as a high Reynolds number model, referring to the local Reynolds number in the wall adjacent cell. The turbulence model is configurable for a low-Re approach but requires non-linear damping functions in the boundary layer as it has been proven inconsistent in estimating the near wall eddy viscosity[6]. Conventionally the k - ε is configured with the high-Re approach and is utilized as such in this thesis to reduce the overall computational load associated with the discretization of the domain.

In CFX the high-Re numerical setup is coupled with **Scalable Wall Functions**, which is an extension of the Launder and Spalding [18] approach, based on the logarithmic relation of viscous and inertial forces in the boundary layer. The main objective of the scalable approach is firstly to replace the standard formulation of the dimensionless near wall velocity scale, u^+ , with a new velocity scale that does not become singular at separation points, and lastly to limit the calculated dimensionless wall distance, y^+ , to the log-layer valid range. The equation for the alternate velocity scale reads,

$$u^* = C_\mu^{1/4} k^{1/2} \quad (2.8)$$

u^* does not go to zero as the wall adjacent tangential velocity, u_t , approaches zero, and can further be used to calculate the wall shear stress as,

$$\tau_w = \rho u^* u_\tau \quad (2.9)$$

Where u_τ is the frictional velocity defined as,

$$u_\tau = \frac{u_t}{\frac{1}{\kappa} \ln(\bar{y}^*) + C} \quad (2.10)$$

Where C is log-layer constant dependant on wall roughness, and \bar{y}^* is the limited dimensionless wall distance calculated as,

$$\bar{y}^* = \max(y^*, 11.06) \quad (2.11)$$

y^* is the calculated real dimensionless wall distance,

$$y^* = \frac{\rho u^* \Delta y}{\mu} \quad (2.12)$$

Where Δy is the real distance to the wall adjacent cell center and u^* is calculated from Equation .2.8. The limiter of 11.06 in Equation 2.11 represents the intersection point between the linear (viscous sub-layer) and logarithmic (log-law) near wall profiles, ensuring that a local cell with a $y^+ < 11.06$ is overwritten and a valid y^+ is always used to estimate the wall adjacent flow properties.

With the calculated wall shear stress, derivatives for the velocity can be solved for, and with the calculated velocity scale u^* and wall distance \bar{y}^* , the boundary condition for turbulent dissipation rate can be calculated as (valid for the logarithmic region),

$$\varepsilon = \frac{\rho u^*}{\bar{y}^*} \frac{C_\mu^{3/4}}{\kappa} k^{3/2} \quad (2.13)$$

The remaining constant required for establishing Equation 2.3 to 2.13:

$$[C_{\varepsilon 1}, C_{\varepsilon 2}, C_\mu, \sigma_k, \sigma_\varepsilon, \kappa] = [1.45, 1.9, 0.09, 1.0, 1.3, 0.40]$$

2.1.2 k - ω SST Turbulence model

The k - ω SST (Shear Stress Transport) turbulence model solves in addition to the mass, momentum, and energy equation for the turbulent kinetic energy, k , and turbulent frequency, ω . The model is a hybrid of the Wilcox k - ω model and the standard k - ε known for rendering accurate predictions of the onset of flow separation under adverse pressure gradient flows [14]. The model is typically considered a low Reynolds number model but can be combined with wall functions and used as a high-Re model. The low-Re implementation is used for the current work and is presented as such. The flux, state, and source term vector in Equation 2.1 for the k - ω SST model reads,

$$\mathcal{Q} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}u_i \\ \bar{\rho}h_0 - \bar{p} \\ \bar{\rho}k \\ \bar{\rho}\omega \end{bmatrix} \quad (2.14)$$

$$\mathcal{F}_j = \begin{bmatrix} \bar{\rho}u_j \\ \bar{\rho}u_j\bar{u}_i + \bar{p}\delta_{ij} - \tau_{ij} + \overline{\rho u'_j u'_i} \\ \bar{\rho}u_j\bar{h}_0 - \lambda\frac{\partial T}{\partial x_j} - \frac{\mu_t}{Pr_t}\frac{\partial \bar{h}}{\partial x_j} - \bar{u}_i\tau_{ij} + \bar{u}_i(\overline{\rho u'_j u'_i}) \\ \bar{\rho}u_j k - \frac{\partial k}{\partial x_j} \left(\mu + \frac{\mu_t}{\sigma_{k3}} \right) \\ \bar{\rho}u_j \omega - \frac{\partial \omega}{\partial x_j} \left(\mu + \frac{\mu_t}{\sigma_{\omega 3}} \right) \end{bmatrix} \quad (2.15)$$

$$\mathcal{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ P_k - \beta' \rho k \omega \\ (1 - F_1)2\rho\frac{1}{\sigma_{\omega 2}\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} + \alpha_3\frac{\omega}{k}P_k - \beta_3\rho\omega^2 \end{bmatrix} \quad (2.16)$$

This is an eddy viscosity model that aims to model the turbulent stresses with the Boussinesq assumption equivalent to the k - ε model, but here the eddy viscosity is assumed to be linked to k and ω ,

$$\mu_t = \rho \frac{\sqrt{C_\mu k}}{\max(\sqrt{C_\mu \omega}, (\bar{S}_{ij}\bar{S}_{ij})^{0.5}F_2)} \quad (2.17)$$

Where \bar{S}_{ij} is the strain rate tensor,

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (2.18)$$

With the known eddy viscosity, the turbulent production rate can be calculated according to Equation 2.7. In Equations 2.16 and 2.17 two blending functions occur, F_1 and F_2 , that are responsible for switching between k - ω and k - ε dependent on the wall-normal distance and flow variables. The blending function F_1 dictates which model is being used, equal to one in the boundary layer and zero in the free stream, resulting in the k - ω model applied in the boundary layer, and the k - ε turbulence model applied in the free stream. The choice of k - ω in the boundary layer is due to its ability to better predict the eddy viscosity near the wall, but is outperformed in the free stream by the k - ε model [19]. The F_1 blending function is defined as,

$$F_1 = \tanh(a_1^4) \quad (2.19)$$

$$a_1 = \min \left[\max \left(\frac{\sqrt{k}}{\beta'\omega y_{wall}}, \frac{500\nu}{y_{wall}^2\omega} \right), \frac{4\rho k}{CD_{kw}\sigma_{\omega 2}y_{wall}^2} \right] \quad (2.20)$$

$$CD_{kw} = \max \left(2\rho\frac{1}{\sigma_{\omega 2}\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-10} \right) \quad (2.21)$$

Where y_{wall} is the distance to the closest wall. The blending function F_2 in Equation 2.17 dictates which model is being used in the prediction of the eddy viscosity defined as,

$$F_2 = \tanh(a_2^2) \quad (2.22)$$

$$a_2 = \max\left(\frac{2\sqrt{k}}{\beta'\omega y_{wall}}, \frac{500\nu}{y_{wall}^2\omega}\right) \quad (2.23)$$

The remaining model constants required to establish Equations 2.16 to 2.23 are obtained as linear combinations of model constants from underlying models using the blending function F_1 , or regular empirically generated constants. The new linear constants are denoted with subscript 3 (e.g. σ_{k3}), and are defined as,

$$\Phi_3 = F_1\Phi_1 + (1 - F_1)\Phi_2 \quad (2.24)$$

With Equation 2.24 the remaining model constants are:

$$[\alpha_1, \alpha_2, \beta_1, \beta_2, \beta', \sigma_{k1}, \sigma_{k2}, \sigma_{\omega1}, \sigma_{\omega2}] = [5/9, 0.44, 0.075, 0.0828, 0.09, 2, 1, 2, 1/0.856]$$

The low-Re version of the k - ω SST model integrates down to the wall, resolving gradients by piece-wise linear integration, instead of estimating gradients from wall functions. To achieve a good linear representation of the gradients in the boundary layer a $y^+ < 2$ is recommended [14].

2.2 Procedure for Estimation of Discretization Error

CFD solvers that utilize the Eulerian frame of reference to model any stream of fluid are inherently dependent on the special resolution of the flow path to accurately represent the flow field. To solve for a flow field a grid has to be generated superimposed on the domain creating smaller discrete control volumes. Solving for the transport of fluid properties is further done by integrating a general transport equation solving for the fluxes across faces and further interpolating to get a cell center value by linear or non-linear means. The accuracy of the integration is dependent on the resolution of the domain and its ability to accurately represent local gradients .i.e the order of linearity of the flow field in the control volume is to be accurately represented by the interpolation technique. To evaluate the error associated with the discretization of the domain *the Fluids Engineering Division of ASME* [2] has developed an extrapolation technique to approximate the solution for infinite resolution. The 1st step is to define a representative grid size h for an initial resolution,

$$h = \left[\frac{1}{N} \sum_{i=1}^N (\Delta V_i) \right] \quad (2.25)$$

Valid for three dimensions where ΔV_i is the cell volume of the i th cell and N is the total number of cells in the domain. The 2nd step is to refine the grid two (or

more) times with a relative grid refinement factor of $r = h_{coarse}/h_{fine}$ greater than 1.3 while keeping the cells geometrically proportional. Simulations are to be carried out for each grid monitoring key parameters, ϕ , important to the objective of the framework and extracted for a fully converged solution. Step 3 is to calculate the apparent order p of the method for $h_1 < h_2 < h_3$ with equations,

$$p = \frac{1}{\ln(r_{21})} |\ln|\varepsilon_{32}/\varepsilon_{21}| + q(p)| \quad (2.26)$$

$$q(p) = \ln \left(\frac{r_{21}^p - s}{r_{32}^p - s} \right) \quad (2.27)$$

$$s = 1 \cdot \text{sgn}(\varepsilon_{32}/\varepsilon_{21}) \quad (2.28)$$

Where,

$$r_{21} = \frac{h_2}{h_1} \quad r_{32} = \frac{h_3}{h_2} \quad \varepsilon_{21} = \phi_2 - \phi_1 \quad \varepsilon_{32} = \phi_3 - \phi_2$$

The 4th step is then to calculate the extrapolated values from,

$$\phi_{ext}^{21} = (r_{21}^p \phi_1 - \phi_2)(r_{21}^p - 1) \quad (2.29)$$

$$\phi_{ext}^{32} = (r_{32}^p \phi_2 - \phi_3)(r_{32}^p - 1)$$

The 5th step is to calculate the approximate relative error, e_a , extrapolated relative error, e_{ext} , and fine-grid convergence index, GCI_{fine} .

$$e_a^{21} = \left| \frac{\phi_1 - \phi_2}{\phi_1} \right|$$

$$e_{ext}^{21} = \left| \frac{\phi_{ext}^{12} - \phi_1}{\phi_{ext}^{12}} \right| \quad (2.30)$$

$$GCI_{fine}^{21} = \frac{1.25e_a^{21}}{r_{21}^p - 1}$$

The apparent order p along with e_a , e_{ext} , and GCI_{fine} can be used to motivate the final grid resolution. The framework does not however offer any modeling metric for turbulent flows and general recommendations regarding grid resolution in critical regions specific to the utilised model require further validation.

2.3 Aerodynamic Instabilities and High Speed Effects in Compressors

Off-design performance of a compressor or turbine is important to consider in the design phase if the turbomachine is to be capable of operating under a range of conditions such as part load during take-off (for aero engines). For high-speed machines operating at transonic/supersonic speeds additional caveats have to be considered in the design due to the presence of shocks. The operating range of a

compressor is commonly described by the nomenclature surge and choke denoting the upper and lower limit in which the design is capable of operating. In addition, the phenomena of stall is often used to mark a performance deteriorating state ([7]).

Choke is the limit of the operating range in which maximum mass flow through the passage is reached. Choke is induced by reducing the pressure ratio and increasing the Mach number (over the stage). As subsonic flow is accelerated the density reduction is not a linear correlation to the flow speed as the flow velocity, c , increases more rapidly than the density, ρ . This holds true for subsonic flow $M < 1$ but not for supersonic flow or $M = 1$. At supersonic conditions, the Mach number and flow velocity increase but the mass flow per unit area decreases as the compressible effects become more prominent. There is therefore a maximum mass flow rate per unit area that occurs at $M = 1$, and this is achieved in the compressor passage once $M = 1$ across the section of minimum flow area. Beyond this point, the mass flow will initially stagnate and eventually decrease.

Surge is a state in which the flow is completely reversed in the machine. Natural convection occurs in the direction of a negative pressure gradient, counteracted in compressors by adding kinetic energy to the fluid by the means of rotating blades and/or inducing a velocity at the inlet. However, there is a point at which the adverse pressure gradient is too high to overcome, and natural convection forces the fluid to reverse. The phenomenon is called surge and leads to a state in which emergency shutdown of the compressor is necessary to prevent damage to the blades as the surge can occur in periodic patches creating frequencies harmful to the construction.

Stall is a phenomena that is non-harmful to the design but reduces the efficiency of the blade row. Stall occurs when the boundary layer separates from the blade reducing the flow area in the passage between the blades. This will lead to the fluid being redirected upstream of the blade leading edge to the adjacent blade and effectively increase the incidence. At this point, the boundary layer will separate from the adjacent blade as the surface velocity is increased by the increase in incidence (the local stage loading and diffusion factor is increased by the increase in turning). The separation on the adjacent blade will then force fluid back into the original blade passage and the flow will reattach but induce separation on the blade adjacent to the original adjacent blade. Stall patches is therefore a phenomena that periodically "jump" around in the blade row and reduce the overall efficiency from the separation of boundary layers affecting the exit flow angle. Since the separation of a boundary layer is correlated to the pressure ratio over a stage, stall will occur as the design is approaching surge.

If a compressor operates at transonic speeds additional loss sources occur from shock formation. If the free stream Mach number upstream of the blade row is close to subsonic speeds $M \geq 0.7$, the relative local Mach number over the blade suction side will reach much larger values and shocks will form on the blade surface. Shock boundary layer interaction will induce a thickening of the boundary layer and induce greater viscous losses. There is however some advantages to high-speed compressors as it induces a greater mass flow per unit area and a high work input per stage.

3

Methods

3.1 Baseline Case

In CATANA article [1] baseline geometrical features and design point data is defined for experimental setup used in obtaining performance data for its operating range at 100% design speed. Experimental and RANS data obtained from the article is used for cross-validating the current framework for equivalent mass flow at design shaft speed.

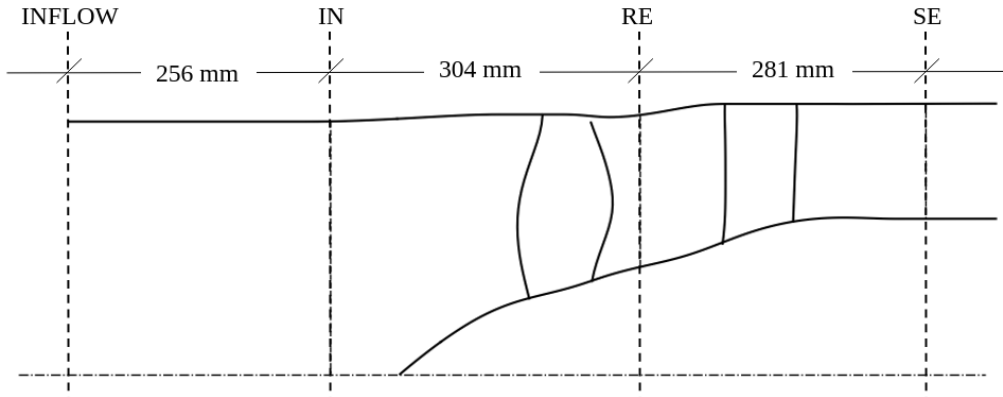


Figure 3.1: Machine core view of the annular with integrated measurement planes for performance analytics. IN-stage inlet, RE-rotor exit, SE-stage exit

ECL5/CATANA DESIGN DATA	
Rotor diameter	508 [mm]
Number of rotor blades	16
Number of stator vanes	31
Design mass flow	36 [kg/s]
Design total pressure ratio	1.35
Design shaft speed	11000 [rpm]
Design tip Mach number	1.02
Design isentropic efficiency	92.6 [%]

Table 3.1: Design point data for baseline ECL5/CATANA test rig.

Inlet (IN), rotor exit (RE), and stage exit (SE) planes are located at equivalent probe locations for CATANA experimental setup.

3.2 Mesh Generation

Figure 3.1 illustrates three separate domains of the machine core i.e. inlet, rotor, and stator. Grid generation of each domain is carried out separately and treated as modules. Grid generation of the inlet is carried out in Fidelity Pointwise 2022.2 and exported to pointwise gridgen (*.grd) format for compatibility with Ansys CFX. Grid generation of the rotor and stator domain is carried out in Ansys Turbogrid 2021 R1. The turbogrid user guide [12] is used as a reference and for Pointwise the digital user-manual [17]. Initial baseline mesh is established in the rotor domain and further used as a reference for meshing of the inlet and stator domain. The Hub and casing of the entire annulus are further represented by B-splines of the original ECL5 geometrical data.

Baseline rotor domain grid is generated in Turbogrid with hexahedron elements applied to the internal topology generation for the blade surface. Wall refinement is implemented at the hub, casing and blade with the *First Element Offset* method specifying an approximate Reynolds number and y^+ offset for the wall adjacent cells and an expansion ratio in wall normal direction. The global cell count is specified with the *Global Size Factor* method adjusting the overall mesh to fit defined wall resolution.

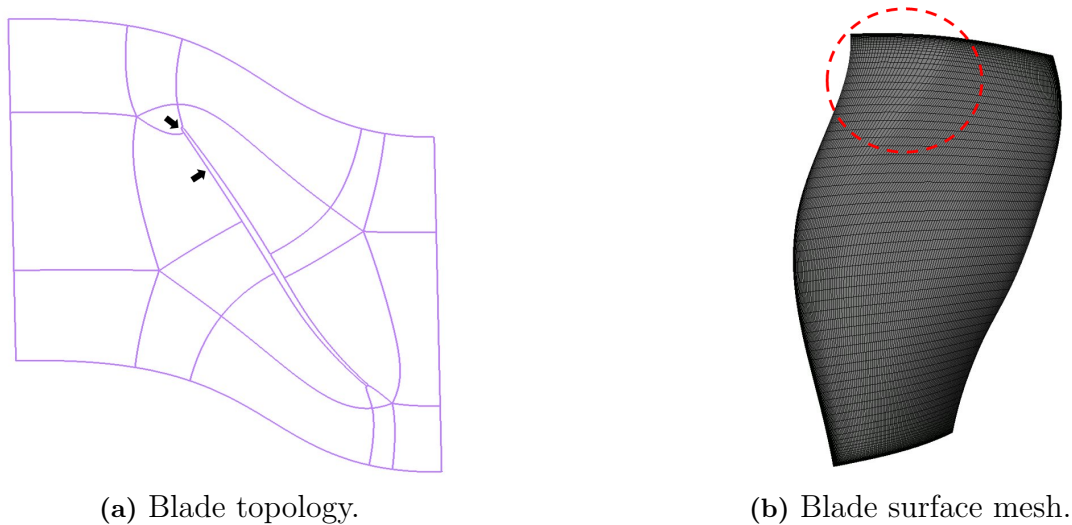


Figure 3.2: Rotor, initial topology and blade surface mesh.

Figure 3.2a and 3.2b illustrates native turbogrid topology and resulting streamwise blade surface mesh. The red circle in figure 3.2 points at a low relative resolution generated by the native topology in a highly sensitive area with assumed large gradients in the flow field. turbogrid is locked to using its own native topology generation and to reduce its local effects on the mesh, two *Edge Split Factors* is applied on the topology lines marked by the black arrows.

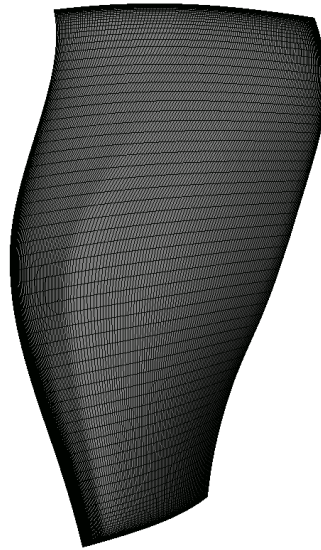
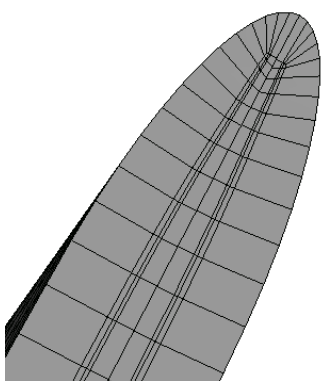
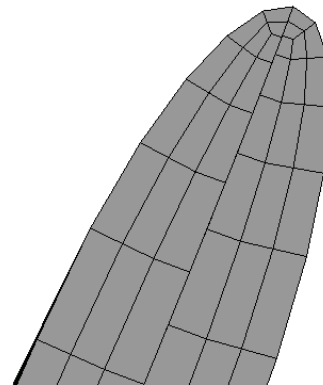


Figure 3.3: Blade surface mesh with edge split.

Figure 3.4 illustrates the blade surface mesh with applied *Edge Splits Factors* larger than unity. The resolution can be seen improved near the leading edge at the tip but induce a sharp gradient in streamwise expansion rate near the hub. Limitations present in turbogrid hinders this problem to be fully solved and the solution is considered acceptable as an assumed more sensitive region is better resolved.



(a) ATM Optimized.



(b) H-Grid Not Matching.

Figure 3.4: Rotor tip topology definition.

Figure 3.4a and 3.4b illustrates the tip topology and resulting mesh for two separate casing tip topology definitions. Figure 3.4a represent the default tip topology generated by the internal ATM optimized method generating a skew mesh with large expansion rates in the leading edge region. Figure 3.4b illustrates the solution obtained by editing in the command editor interface for the blade profile altering the *casing Tip Topology Definition* variable from "ATM Optimized" to "H-Grid Not Matching". Radial cell count in the tip region is defined by a set number of cells utilising the *uniform* setting, globally independent of the *Global Size Factor* vari-

able. Tip radial cell count is set to a value rendering proportional cells to that of the wall adjacent cells in blade normal direction for defined y^+ .

The grid for the stator domain is further generated with equivalent values to that of the rotor domain in regards to the near wall regions. The *Global Size Factor* is then modified trying to match the radial and axial distributions to that of the rotor domain exit interface. The stator is connected to both the hub and casing meaning that no tip gap modifications is necessary. Alterations of the baseline mesh in the rotor and stator domain dependant on use of a **High** or **Low Reynolds number** numeric turbulence model is done by only changing the *Global Size Factor*, *Wall Approximate y^+* , *Radial Tip Cell Count* and *Boundary Layer Expansion rate*.

Baseline Meshing Parameters		
Domain:	Rotor	Stator
Low Reynolds Number Model		
Global size factor	2.0	1.2
Boundary layer offset y^+	0.5	0.5
Target maximum expansion rate	1.2	1.2
Boundary layer expansion rate	1.2	1.2
Hub wall y^+	0.5	0.5
Casing wall y^+	*	0.5
Casing tip number of elements	80	*
High Reynolds Number Model		
Global size factor	0.92	0.7
Boundary layer offset y^+	40	40
Target maximum expansion rate	1.2	1.2
Boundary layer expansion rate	1.2	1.2
Hub wall y^+	120	0.5
Casing wall y^+	*	120
Casing tip number of elements	10	*

Table 3.2: Baseline Mesh Parameters.

The inlet grid is then further generated in Pointwise to match grid proportions at the rotor inlet interface. The grid is generated with the idea of reducing the computational load of the compressor by assuming minimal tangential flow upstream of the rotor. This allows for a small circumferential extent of the inlet and a rotation angle of 5° with 5 cells in tangential direction is selected. Tip gap refinement applied in the rotor is further created in the inlet domain by a topology box with equivalent wall normal extent and uniform cell distribution. A growth rate is then applied to the wall adjacent topology line to match the growth rate in the radial direction to that of the rotor. Further the inlet extends radially down to the center line of rotation introducing wedge elements adjacent to the rotational axis. No refinement is applied here as the boundary represents free-flowing fluid. The hub connecting the center line of rotation and the outflow boundary of the inlet requires additional refinement and is generated by encapsulating the region with two topology boxes,

further locally refining the mesh at the hub.

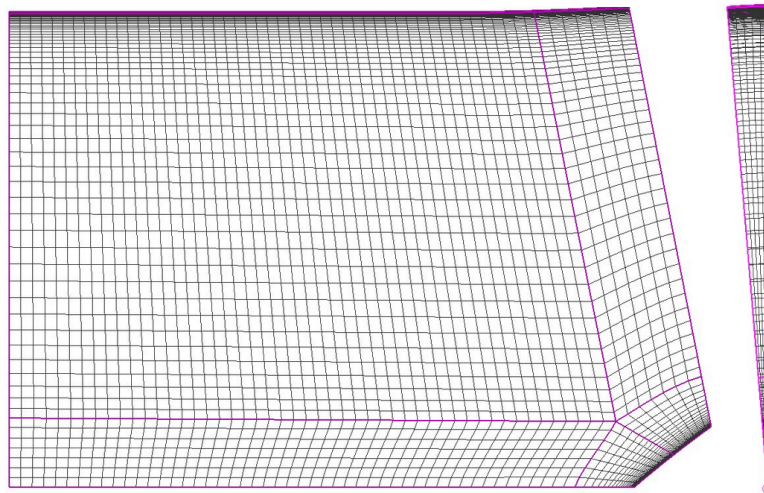


Figure 3.5: Inlet grid.

Figure 3.5 illustrates the grid distribution in the inlet domain, as well as the topology utilised to achieve required growth rate at hub and shroud with minimum skewness.

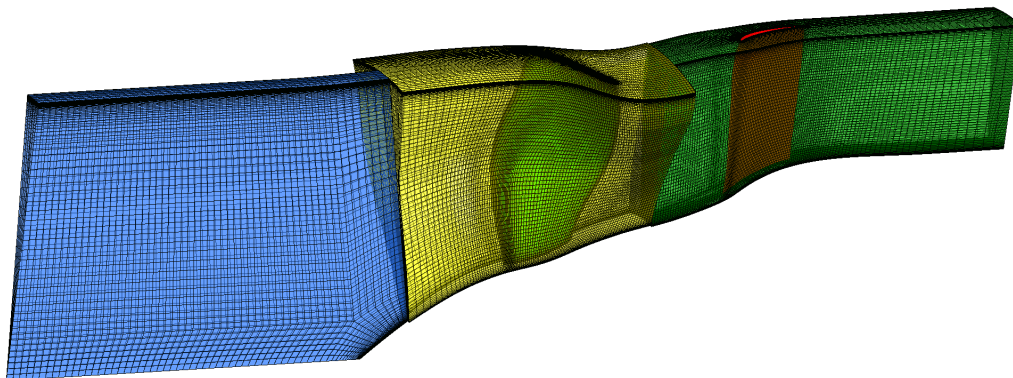


Figure 3.6: Assembled mesh with inlet, rotor and stator domain.

Figure 3.6 illustrates the mesh for the assembled domains, including the stator.

3.3 Boundary Conditions

The computational domain consists of the inlet, rotor, and stator that in 3D renders a need for a minimum of six boundary conditions on each domain. Coupling of the domains is done by assigning an interface at the connecting surface, transporting the fluid across the faces. The inlet and stator are treated as stationary domains while the rotor is rotating, rendering a change in the relative frame of reference across the interfaces. A transition model is therefore assigned at each interface, modeling new properties downstream of the interface.

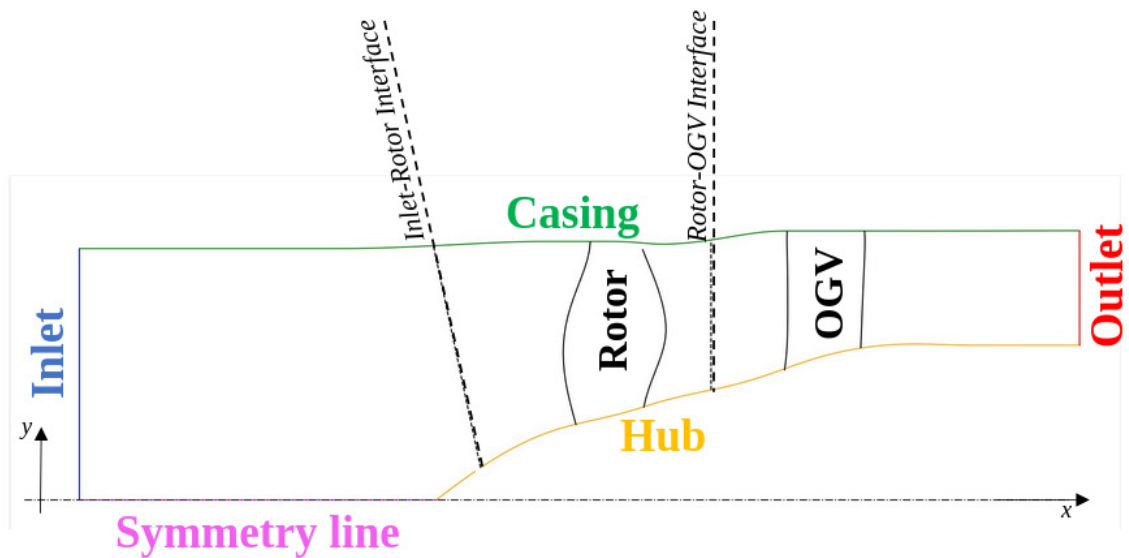


Figure 3.7: Meridional view of the computational domain, illustrating assigned boundary conditions.

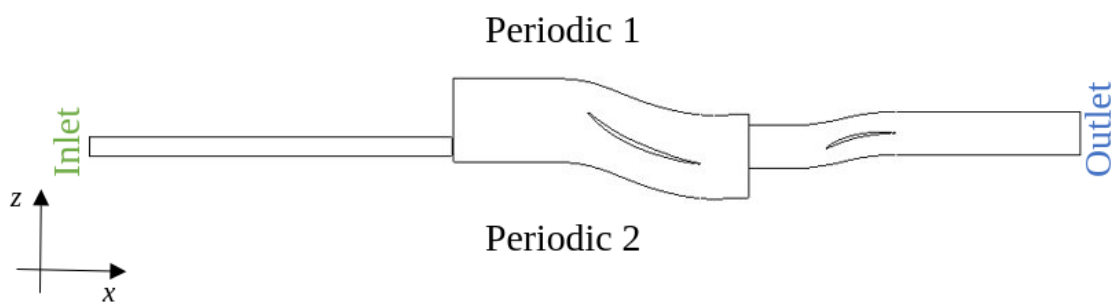


Figure 3.8: Top view of the computational domain illustrating the boundary conditions applied to the tangential patches.

Figure 3.7 and 3.8 illustrate the computational domain in 2D from two perspectives with boundary locations and nomenclature. The inlet originates at the rotational axis and is therefore assigned a symmetry condition at the center of rotation. The remaining part of the lower boundary within the inlet domain connects to the hub in the rotor domain and is assigned a no-slip rotating wall boundary condition,

rotating at equivalent speed to the rotor in the opposite direction. The casing in the inlet domain is assigned a no-slip wall boundary condition and a total pressure boundary is assigned to the inlet face. In the rotor domain both the hub, casing, and rotor are assigned a no-slip wall condition, while the casing boundary is further assigned a counter-rotational velocity to that of the rotor. Further in the stator domain both the hub, OGV, and casing are considered stationary and assigned a no-slip condition. At the outlet a corrected mass flow outlet boundary condition is assigned, computed as [14],

$$\dot{m}_{corr} = \dot{m} \sqrt{\frac{\tilde{T}_{0, exit}/T_{ref}}{\tilde{P}_{0, exit}/P_{ref}}} \quad (3.1)$$

Where $\tilde{T}_{0, exit}$ and $\tilde{P}_{0, exit}$ is the mass-averaged total temperature and pressure at the outlet, and T_{ref} and P_{ref} reference conditions. both the corrected mass flow \dot{m}_{corr} and reference conditions are assigned at the boundary and constant in the equation. The resulting mass flow rate at the boundary is therefore proportional to the exit total pressure and inversely proportional to the square root of the exit total temperature. This allows the boundary condition to adapt dynamically to a range of operating conditions while remaining stable as the flow develops from the initial guess.

Boundary Conditions			
Domain:	Inlet	Rotor	Stator
Hub	No-slip rotating wall	No-slip wall	No-slip wall
Casing	No-slip wall	Counter rotating wall	No-slip wall
Inlet	Total Pressure	Interface	Interface
Outlet	Interface	Interface	Corrected Mass Flow
Blade	*	No-slip wall	No-slip wall
Symmetry line	symmetry	*	*
Tangential planes	Periodic	Periodic	Periodic

Table 3.3: Boundary conditions.

Table 3.3 represents named boundary conditions for the computational domain applied, consulting the CFX-Pre users guide [13] and CFX-Solver modelling guide [15] for implementation. Further added to the table is the boundary condition applied to any tangential plane in the domain. A periodic boundary condition is assigned to these boundaries as the simulated domain is limited to one blade passage. The purpose of the periodic boundary is to mimic the presence of adjacent blades in the circumferential direction, meaning that any fluid exiting one boundary enters the other. As any gravitational forces are neglected in the analysis the current blade passage can represent any passage in the actual machine core and the presence of adjacent blades is modeled by the implementation of the periodic boundaries.

Further interfaces connecting the domains are assigned different transition models for the relative frame change occurring entering or leaving the rotor domain. At the *Inlet-Rotor interface*, a frozen rotor boundary condition is applied ([15] Chap.

5.3.3.1.2). The frozen rotor method translates any vectorial flow property to rotating properties in the rotating frame of reference. This is accomplished by decomposing the vector in the absolute frame to the relative frame of reference consisting of one relative part of the absolute vector and one translational. The velocity formulation reads,

$$\bar{v} = \bar{v}_r + (\bar{\omega} \times \bar{r}) + \bar{v}_t \quad (3.2)$$

Where $\bar{\omega}$ is the rotational velocity vector of the domain. The frozen rotor interface applies this to the upstream interface adjacent cells and transports the variable across the interface to the rotating frame of reference at each time step. Any scalar properties are directly copied across the interface such as density and static pressure or temperature.

The *Rotor-OGV interface* is assigned a mixing-plane model ([15] Chap. 5.3.3.1.3). This model works equivalent to the frozen rotor model, but further applies a circumferential average to the interface adjacent cells over a prescribed time step interval, meaning that it will always transport a steady-state solution. Downstream of the rotor a phenomena called vortex shedding can occur inducing transient periodic oscillations in a steady state flow field downstream of the blade trailing edge. Implementing a mixing-plane model downstream of the rotor will therefore stop the oscillations from entering the stator region as it averages the data over a set of time steps before transporting it across the interface.

Interface Models	
Inlet-Rotor interface	Frozen rotor model
Rotor-OGV interface	Mixing plane model

Table 3.4: Interface models applied at transition planes.

3.4 Computational Fluid Dynamics

3.4.1 Solver Settings

Configuration of the CFD solver is firstly executed in the graphical interface and recorded to a state file. Modelling assumptions and settings is applied consulting with the CFX users manual [16] and modelling guide [15]. Steady-state is chosen as the temporal frame of reference and the morphology of the fluid is set to continuous (eulerian). The fluid used is Air, assumed a calorically perfect ideal gas, reducing the specific heat, C_p and C_v , to constant values with a fixed ratio, γ . Heat transfer throughout the domain is calculated from the total energy equation, appropriate for high-speed compressible flows, and the viscous work term (heat generation due to internal viscous interactions in the fluid) is not assumed negligible and accounted for in the equation. High Resolution is used as both advection scheme and turbulence numerics, resulting in CFX attempting to find the highest order scheme possible while keeping the solution bounded (all properties are "bounded" by their boundary values), theoretically resulting in better stability in the solver. For the low-Re model

the $k-\omega$ SST turbulence model is used for its ability to accurately predict boundary layer flows with high adverse pressure gradients (APG) if $y^+ \leq 1$ at the walls. Automatic wall functions are selected for the low-Re model but never utilized as the y^+ mesh requirement is scaled throughout the domain and the flow properties are integrated down to the wall. The high-Re model utilizes $k-\varepsilon$ as a turbulence model coupled with CFX internal scalable wall functions, switching from linear interpolation at the wall to the log law at a y^+ of 11.56. An initial timescale factor of 1.0 is used and the auto time scale control is switched on. As steady-state is assumed, the time scale factor can be interpreted as a pseudo time step factor and acts as an implicit relaxation factor, internally adjusted by CFX to progress the simulation while maintaining numerical stability.

Numeric Setup		
Setting	Low-Re Model	High-Re Model
Temporal Frame of Reference	Steady State	
Morphology	Continuous Fluid	
Fluid	Air	
Fluid Model	Ideal Gas	
Heat Transfer Equation	Total Energy (incl. viscous work term)	
Advection Scheme	High Resolution	
Turbulence Model	$k-\omega$ SST	$k-\varepsilon$
Turbulence Numerics	High Resolution	
Wall Function	Automatic	Scalable
Timescale Control	Auto Timescale	
Initial Timescale Factor	1.0	

Table 3.5: Solver settings.

3.4.2 Convergence Criteria

CFX offers an internal convergence criteria that exits the simulation whenever a residual target is achieved in transport equations for turbulent properties, u, v, w -momentum, and continuity. However achieving a residual target at a specific pseudo-time-step does not ensure that steady-state (in pseudo-time-steps) is achieved for critical design parameters such as pressure, temperature, and isentropic efficiency. To work around this a custom convergence criteria is developed in python and integrated in the solver. The algorithm continuously extract time-step history for pressure ratio, isentropic efficiency, and mass flow rate. For the current time-step history with n entities of each parameter ϕ , the rolling average (RA) is calculated for the RMS values of the dataset with an averaging window of k_n entities at each time-step, t , for $t \geq k_n$.

$$RA_t^{RMS\phi} = \sqrt{\frac{1}{k_n} \sum_{i=n_t-k_n}^{n_t} \phi_i^2} \quad (3.3)$$

What this does is that it continuously renders a curve that represents the mean RMS value of ϕ at any given time step for the previous k_n entities in the dataset. When

this curve starts to stabilize the parameter is stable in time, and thus convergence can assumed. To further quantify a tolerance of what is considered stable, to numerically capture the behavior of the curve, the rolling average of the root mean square deviation (RMSD) is calculated.

$$RA_t^{RMSD_\phi} = \frac{\sqrt{\frac{1}{k_n} \sum_{i=n-k_n}^n (RA_i^{RMS_\phi} - RA_n^{RMS_\phi})^2}}{\frac{1}{k_n} \sum_{i=n-k_n}^n RA_i^{RMS_\phi}} \quad (3.4)$$

Here the most recent time step in the dataset of k_n entities is used as the baseline value, and the rest is used to compute the deviance. This graph renders a descending curve as the parameter starts to stabilize and the preciously calculated RA RMS values deviate less from the baseline (latest) entity. By applying a tolerance value to the RA RMSD value, it is possibly to capture when the RA RMS value is stable in time and the solution can be considered converged. For this case multiple parameters is used so the condition for considering convergence in the solver is when the RA RMSD value for all parameters is bellow the tolerance.

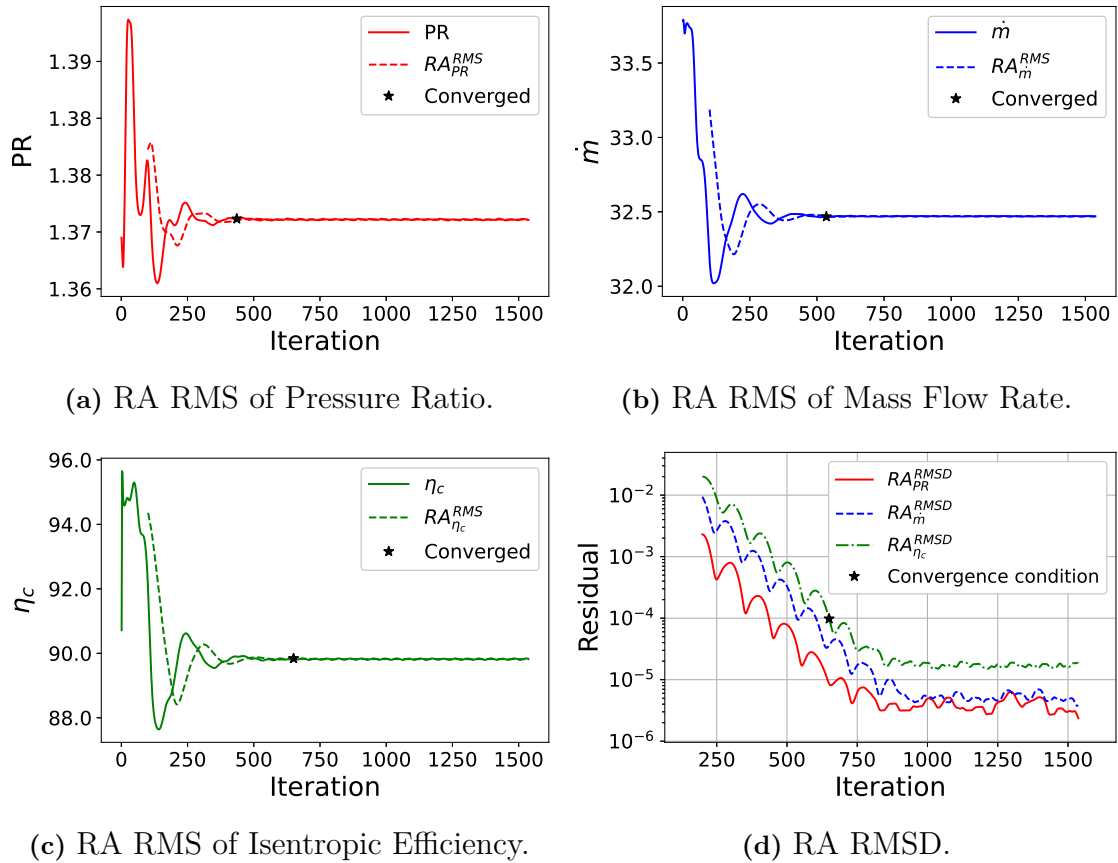


Figure 3.9: Test run of convergence algorithm with a tolerance of $1E - 4$ for Pressure ratio (PR), mass flow rate (\dot{m}), and isentropic compression efficiency (η_c).

Figures presented in 3.9 illustrates a test run of the implemented convergence algorithm with a tolerance of $1E - 4$ and averaging window size of 100 data points, and further serves the purpose of illustrating how the equations work on the parameter

data. For this case the pressure ratio (PR), mass flow rate (\dot{m}), and isentropic compression efficiency (η_c) is utilised. Figures 3.9a - 3.9c illustrates the raw time step data and the RA RMS filtered data, as well as the iteration number at which the parameter is considered converged. Figure 3.9d illustrates the RA RMSD residuals for monitored parameters and the iteration number at which the convergence condition is achieved. The implementation of the algorithm in the overall framework is handled in python coupled with native CFX commands to communicate with the solver.

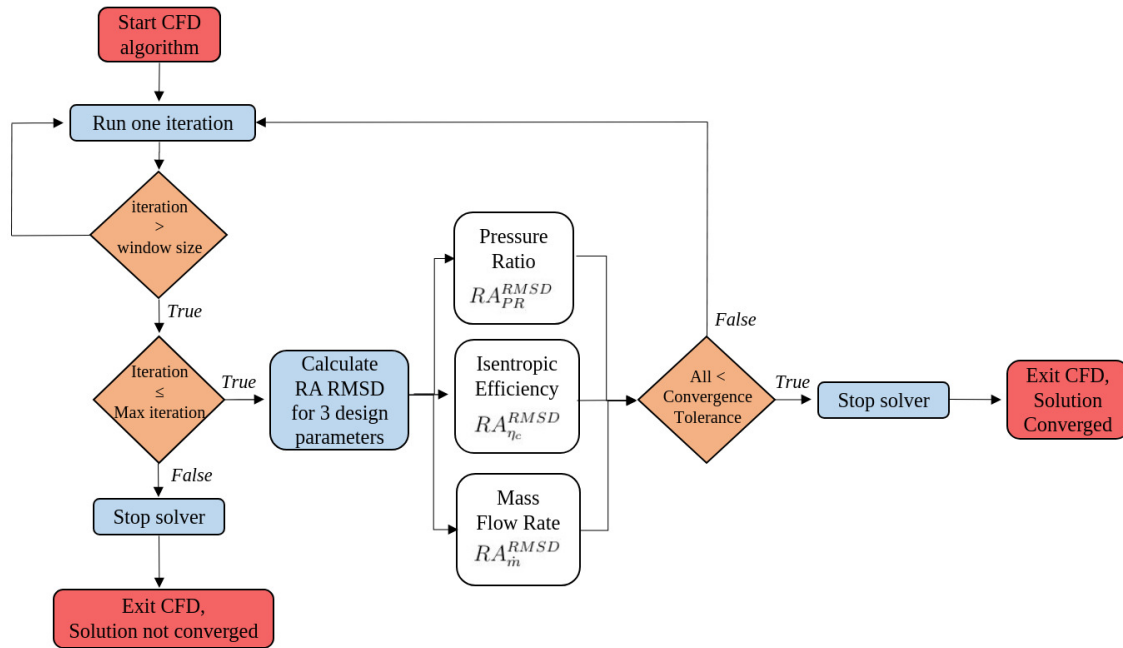


Figure 3.10: CFD flow chart.

Figure 3.10 illustrates the structure of how a case (one design at a specific operating condition) is set up and how the convergence algorithm is implemented in the solution procedure. For a full reference of CFX commands utilised for Python integration, see Appendix 1. For a pseudo code representation of the convergence algorithm and integration in speedline code, see Appendix 2.

4

Results

Assembly of the domain for a confirmed stable and accurate discretization allows for the generation of a compressor map for the framework, generated by matching correct boundary conditions to achieve similar results to data obtained from ECL5 experimental and URANS data, in regards to fan face mach number, mass flow rate, and pressure ratio. This is firstly done on a steady-state RANS numeric setup with the low-Re turbulence model $k-\omega$ -SST to compare with the ECL5 URANS data and obtain an equivalent framework to that of the original authors. The self obtained RANS data is then used as a baseline for evaluating the impact of reducing the wall resolution, pairing wall functions with a high Reynolds number $k-\varepsilon$ turbulence model. The impact of utilizing a high Reynolds number model is measured by comparing self-obtained compressor maps to that of the original ECL5 data. Lastly the impact of removing the OGV from the simulated domain is analyzed by replacing the OGV with an extended rotor outlet domain. Cross-validation of removing the rotor is done by measuring critical parameters and radial distributions at a surface downstream of the rotor, comparing to data extracted from previous cases at equivalent locations. Lastly, the overall computational load is evaluated by calculating the average iteration time for the CFX solver.

4.1 Grid Independence Study

A grid independence study is carried out for two separate cases, firstly for the low-Re model with an overall greater cell density due to y^+ requirement at the walls, and lastly, the high-Re model where the grid is scaled to switch to wall functions in the boundary layer. All numeric simulations are executed with equivalent boundary values and an exit-corrected mass flow rate of 28.2 kg/s. Data compared in selection process of the final grid resolution is obtained at the inlet, rotor exit, and stator exit plane, see Figure 3.7. Discretization error of each grid is measured in pressure ratio, temperature ratio, isentropic efficiency, and mass flow rate as these parameters are critical to the overall study and directly comparable to the source material for the CATANA fan design. The procedure for obtaining a set of grids for each case is done by first establishing a baseline grid resolution (see section 3.2) and then generating a coarser and refined version by altering the global size factor in Turbogrid, while all other parameters are kept constant. The inlet grid is not further altered as it only serves the purpose of developing the flow field upstream of the rotor. The approximate cell count of the respective coarser and refined grid is obtained by establishing a suitable growth rate in representative grid size, r , and solving for,

$$r = \frac{h_{coarse}}{h_{fine}} = \frac{\left[\frac{1}{N} \sum_{i=1}^N (\Delta V_i) \right]_{coarse}^{1/3}}{\left[\frac{1}{N} \sum_{i=1}^N (\Delta V_i) \right]_{fine}^{1/3}} = \left\{ \sum_{i=1}^N (\Delta V_i) |_{fine} \approx \sum_{i=1}^N (\Delta V_i) |_{coarse} \right\} = \frac{N_{fine}^{1/3}}{N_{coarse}^{1/3}} \quad (4.1)$$

The overall goal is to use a $r \geq 1.3$ as recommended in [2]. All grids are evaluated at a fully converged solution in accordance to the established convergence criteria in section 3.4.2 with a tolerance of $1e - 4$.

4.1.1 Low-Re model grid

A total of three grids are generated in evaluating the discretization error associated with the spatial resolution of the flow field when solved for with $k-\omega$ turbulence model. A baseline mesh was established with 7,839,398 grid cells and the respective approximate cell count of the refined and coarse grid is derived from establishing a growth rate in representative grid size of $r \approx 1.3$ (see Equation 4.1).

Grid data	
N_1	17,154,060
N_2	7,839,398
N_3	3,630,311
r_{21}	1.298
r_{32}	1.293

Table 4.1: Low-Re model grid data

In table 4.1 the total cell count and exact representative growth rate, r , for the

selected grids are shown. The corresponding global size factor in Turbogrid is 2.9, 2.0, and 1.2 for the rotor domain, and 1.75, 1.2, and 0.6 for the stator domain, for N_1 , N_2 , and N_3 respectively.

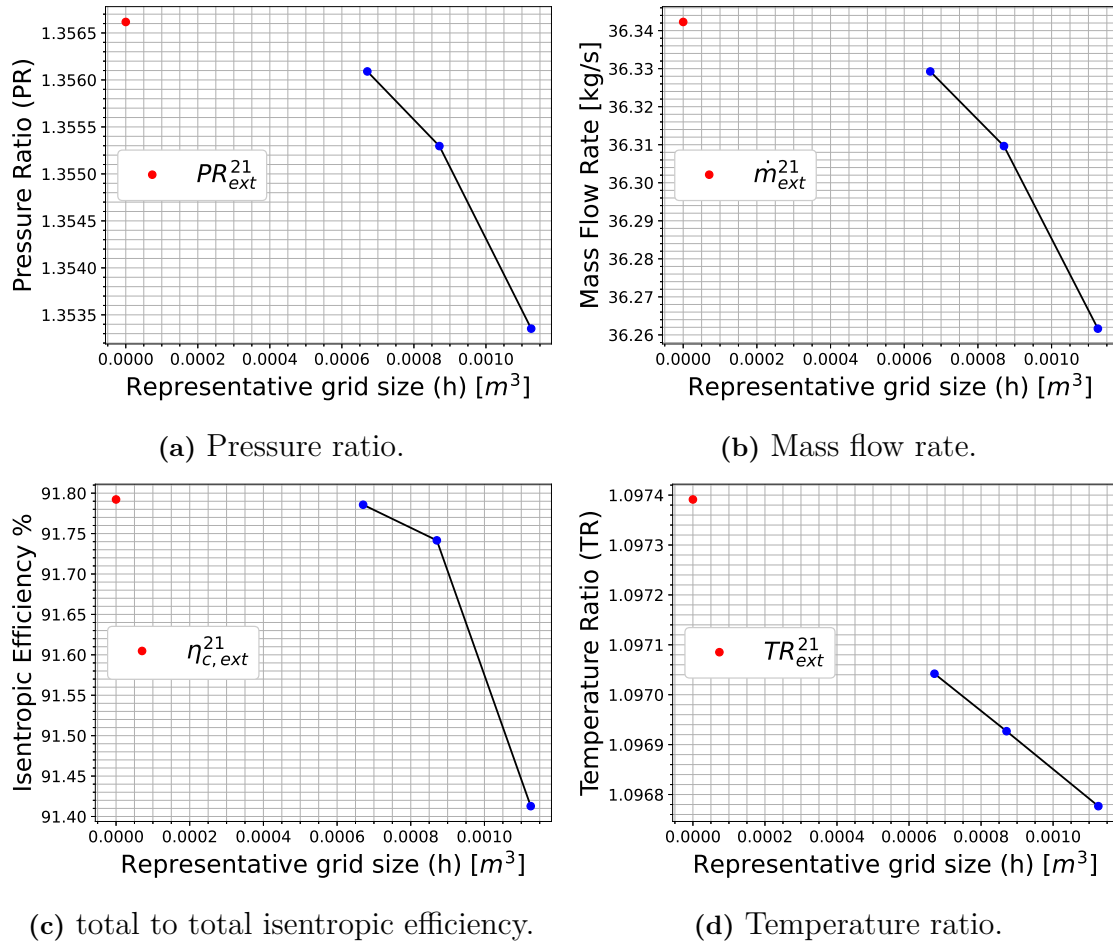


Figure 4.1: Variance in design parameters relative the representative grid size, h , for pressure ratio, mass flow rate, total to total isentropic efficiency, and temperature ratio.

Figure 4.1 illustrates the governed parameters for the grid study and the corresponding extrapolated values ϕ_{ext}^{21} , approximating the parameter at an infinite grid resolution. Temperature ratio and pressure ratio (figure 4.1a and 4.1d) indicates that the fines grid resolution obtains an approximate error of order 10^{-4} relative to the extrapolated value and the mid-resolution produces an error of order 10^{-3} relative the extrapolated value. Mass flow rate and isentropic efficiency (figure 4.1b and 4.1c) indicate a relative error of order 10^{-2} . Important to note is that the perceived smaller variance in pressure and temperature ratio is inconclusive to the overall results as it is only an indicator relative to the extrapolated value and represents a ratio in discretization error of scalar properties between two planes. Isentropic efficiency and mass flow rate are scalar properties measured at a downstream plane and therefore represent the cumulative error of the upstream flow path. Any individual plot is therefore not decisive of the final resolution but rather serves the purpose of

illustrating the relative difference noted while varying the spatial resolution, indicating a decline in the variance at each refinement.

Parameter calculations of discretization error.				
	Pressure Ratio	Mass flow rate	Isentropic efficiency	Temperature ratio
ϕ_1	1.3533	36.2616 [kg/s]	91.4128 [%]	1.0968
ϕ_2	1.3566	36.3096 [kg/s]	91.7922 [%]	1.0969
ϕ_3	1.3560	36.3293 [kg/s]	91.7587 [%]	1.0970
p	3.52	3.52	7.84	1.09
ϕ_{ext}^{21}	1.3566	36.3423 [kg/s]	91.7921 [%]	1.0974
e_a^{21}	0.005%	0.005%	0.004%	0.001%
e_{ext}^{21}	0.003%	0.004%	0.0007%	0.003%
GCI_{fine}^{21}	0.004%	0.004%	0.0008%	0.004%

Table 4.2: Low-Re model discretization error.

Table 4.2 presents the exact calculated grid related parameters for each solution established in section 2.2. The extrapolated error e_{ext}^{21} and the relative error e_a^{21} can be seen to indicate that the mid resolution, N_2 , renders a suitable grid resolution as the calculated margin of error is of order 10^{-3} within the extrapolated value. To expand on the analysis and investigate the effect of removing the stator from the configuration, a mesh for the stand alone rotor configuration is created by removing the stator domain and extending the rotor mesh to the outlet plane from the previous rotor-stator interface. The extended rotor domain mesh is created without applying any growth rate and is 1:1 proportional to the previous interface adjacent cells in the rotor domain.

Final cell count.	
Configuration	Cell count
rotor and stator	7,839,398
stand alone rotor	7,563,652

Table 4.3: Final grid cell count for configuration with rotor and stator, and stand alone rotor, low-Re grid.

In table 4.3 values for the final cell count for configuration with and without stator are given.

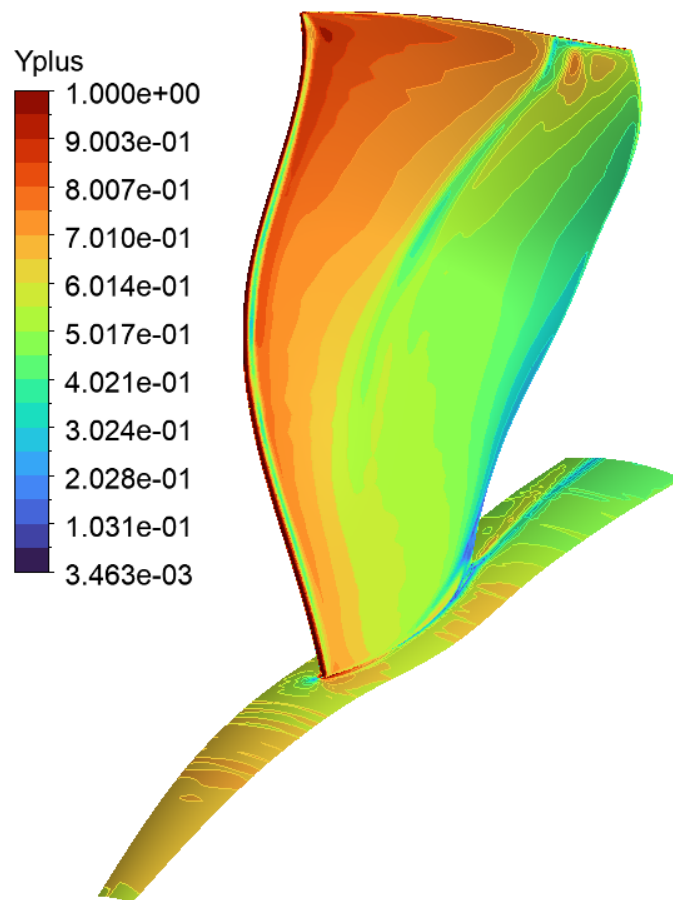


Figure 4.2: y^+ distribution at rotor suction side and hub surface for low-Re model grid.

Figure 4.2 illustrates the y^+ distributing at the blade rotor suction side and hub. Local regions of $y^+ \geq 1$ can be observed at the blade leading edge and tip. Small localized regions with a y^+ deviation from the maximum recommended value are expected and proven difficult to remedy in Turbogrid. Illustrated distribution is used in the simulations for the low-Re model.

4.1.2 Wall function grid

A total of three grids are used to evaluate the discretization error in the flow domain for a solver coupled with $k-\varepsilon$ turbulence numerics. Firstly a baseline mesh is established with a cell count of 780,737 and a target $30 < y^+ < 300$ conforming to the general bounds of the log-law. Target y^+ is achieved in greater parts of the wall adjacent cells but in localized zones the value can drop below the intersection point of $y^+ < 11.06$, i.e. the \bar{y}^* limiter of the scalable wall function model is applied. A coarse and fine mesh is further created from the baseline mesh approximating the overall cell count with Equation 4.1 and a relative refinement factor of $r = 1.4$. A larger relative refinement factor is used compared to the low-Re grid as the baseline mesh is a factor of 10^{-1} smaller and the overall computational load is reduced. Established parameters rendering desired y^+ and proportions of the baseline mesh

4. Results

are not altered for the three grids and the wall adjacent resolution is kept constant, only the global size factor is altered.

Grid data	
N_1	2,024,375
N_2	780,737
N_3	292,867
r_{21}	1.374
r_{32}	1.387

Table 4.4: High-Re model grid data.

In table 4.4 the respective cell count and exact relative refinement factor for the fine, mid, and coarse grid are presented, corresponding to a global size factor in the rotor domain of 1.8, 1.2, 0.6, and the stator domain of 0.9, 0.6, 0.3, respectively.

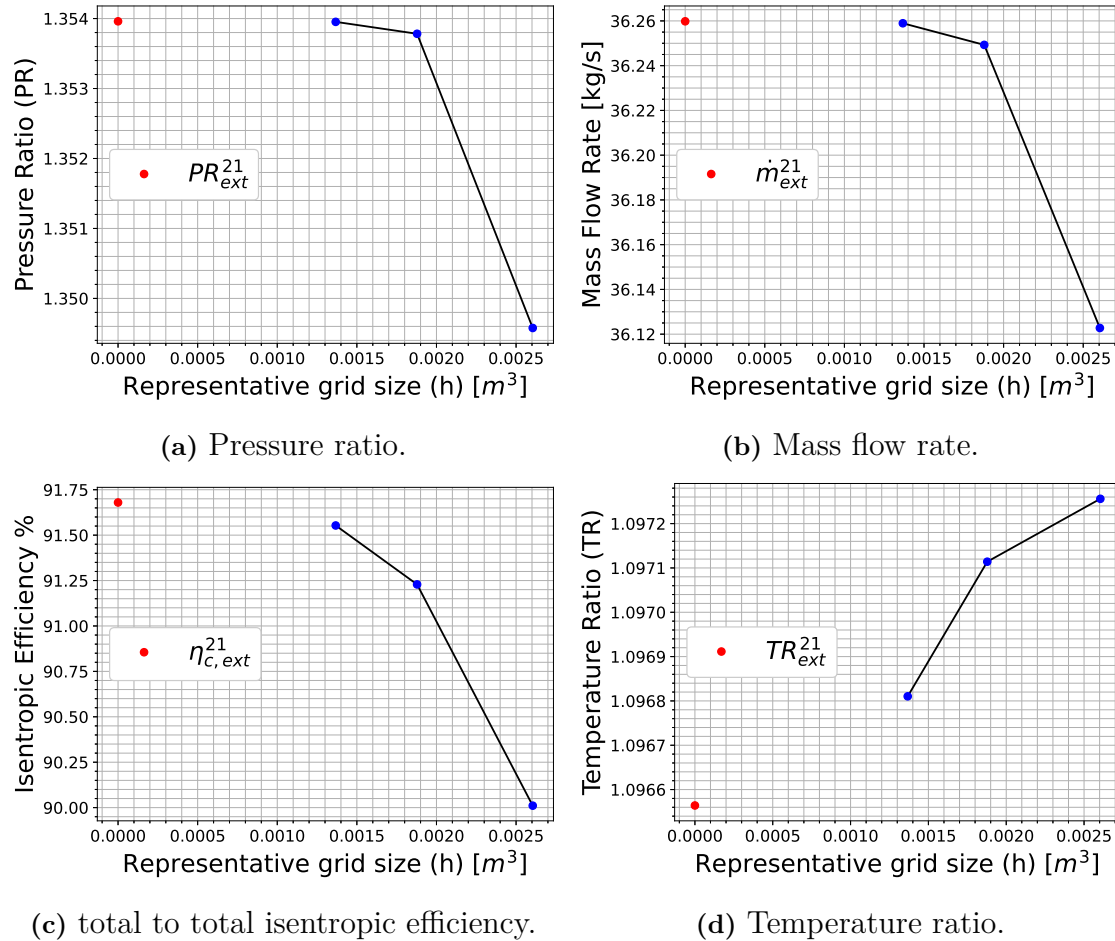


Figure 4.3: Variance in design parameters for wall function mesh relative to the representative grid size, h , for pressure ratio, mass flow rate, total to total isentropic efficiency, and temperature ratio.

Figure 4.3 illustrates the discretization error represented by the pressure ratio, mass flow rate, isentropic efficiency, and temperature ratio, and the extrapolated value

approximated at an infinite resolution. Variance in the pressure and temperature ratio (figure 4.3a and 4.3d) between fine and mid resolution can be seen to be of order 10^{-4} . Variance in the isentropic efficiency and mass flow rate (figure 4.3b and 4.3c) can be seen to be of order 10^{-1} and 10^{-2} respectively.

Parameter calculations of discretization error.				
	Pressure Ratio	Mass flow rate	Isentropic efficiency	Temperature ratio
ϕ_1	1.3495	36.1228 [kg/s]	90.0107 [%]	1.0973
ϕ_2	1.3537	36.2493 [kg/s]	91.2287 [%]	1.0971
ϕ_3	1.3539	36.2589 [kg/s]	91.5532 [%]	1.0968
p	9.79	7.85	4.0	2.53
ϕ_{ext}^{21}	1.3540	36.2598 [kg/s]	91.6795 [%]	1.0965
e_a^{21}	0.001%	0.002%	0.0354%	0.0027%
e_{ext}^{21}	0.000006%	0.00002%	0.0138%	0.0022%
GCI_{fine}^{21}	0.000007%	0.00003%	0.0173%	0.0028%

Table 4.5: High-Re model discretization error.

In table 4.5 the exact grid-related parameters established in section 2.2 are presented with the values observed for each flow property. The mid-resolution can be seen to indicate a small relative error, e_a^{21} , in pressure ratio, temperature ratio, and mass flow rate of $\sim 0.002\%$. A greater error can be observed in the isentropic efficiency of $\sim 0.035\%$, indicating numerical diffusion in the transport of enthalpy. Further, observing the extrapolated relative error of the fine grid, N_1 , the error in pressure ratio and mass flow rate is within $1e-4\%$ of the extrapolated value, and the isentropic efficiency and temperature ratio is within 0.02% of the extrapolated value. The data indicates that the fine grid, N_1 , generates a solution in critical design parameters that are within $\sim 0.014\%$ of the infinite resolution grid.

With a relative error $e_a^{21} \leq 0.0354\%$ for all design parameters. and an extrapolated relative error of $e_{ext}^{21} \leq 0.0138$, the mid resolution is considered suitable for the task. Equivalent to what was done to the low-Re grid, a mesh is created for a stand alone rotor configuration with a cell distribution in the extended region proportional to the previously interface adjacent cells.

Final cell count.	
Configuration	Cell count
rotor and stator	780,737
stand alone rotor	650,883

Table 4.6: Final grid cell count for configuration with rotor and stator, and stand alone rotor, wall function grid.

In table 4.6 values for the final cell count dependent on configuration (with and without stator) are given.

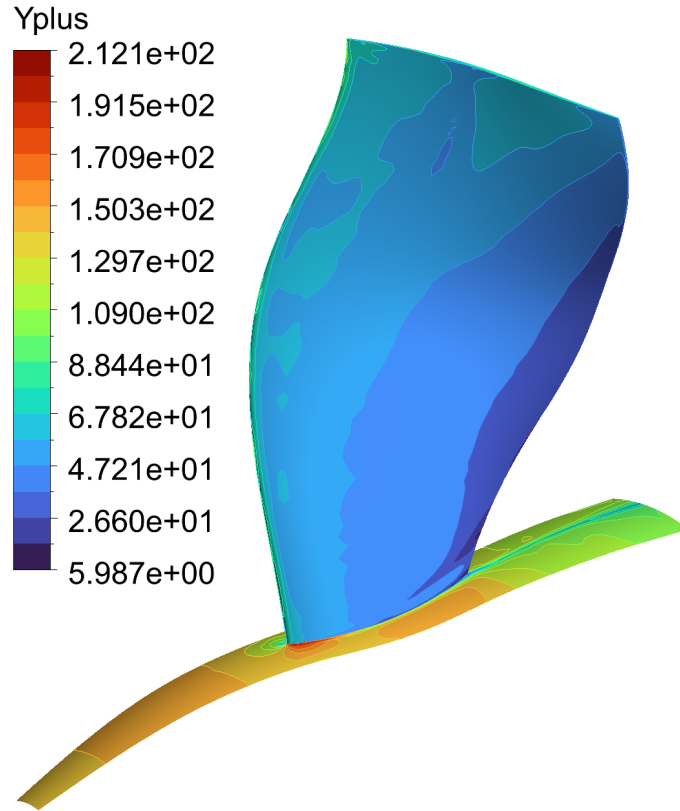


Figure 4.4: y^+ distribution at rotor suction side and hub surface for high-Re (wall function) model grid.

Figure 4.4 illustrates the y^+ distribution achieved for the high-Re model. Local resolutions rendering a $y^+ \leq 11.06$ can be observed meaning that the limiter of the scalable wall function approach is applied in these regions, most prominent near the trailing edge close to the hub.

4.2 Design Point Analysis

The final grid resolution obtained in section 4.1 for the low and high Reynolds number model is simulated at design mass flow and speed obtained from the CATANA publication (see Table 3.1). Experimental design mass flow is declared at 36 kg/s and later illustrated at ~ 35.61 kg/s for URANS data in publication [1]. For this thesis and the sake of comparability, experimental target mass flow is used.

	Operating point data			
	CATANA Exp.	CATANA RANS	Low-Re Model	High-Re Model
Pressure Ratio (MFA) [-]	1.350	1.356	1.355	1.352
Temperature Ratio (MFA) [-]	*1.097	*1.098	1.098	1.098
Isentropic Efficiency [-]	0.925	0.920	0.925	0.916
Mass Flow Rate [kg/s]	36.0	35.61	35.88	35.80
Fan Face Mach [-]	-	-	0.545	0.543

Table 4.7: CATANA fan operation point data.

Performance data at design speed and mass flow rate is presented in table 4.7 for the CATANA test case and simulated annular. Values with the (*) prefix are not present in the CATANA publication and are manually calculated from the data presented in the paper. Fan face Mach number data is a performance metric utilized in the early stages of the project and is collected at radially spaced data points located at an equivalent axial distance from the rotor leading edge. Scaling the outlet boundary condition to find the target mass flow rate is manually iterated and a final corrected mass flow rate of 27.8 [kg/s] is selected as the design condition.

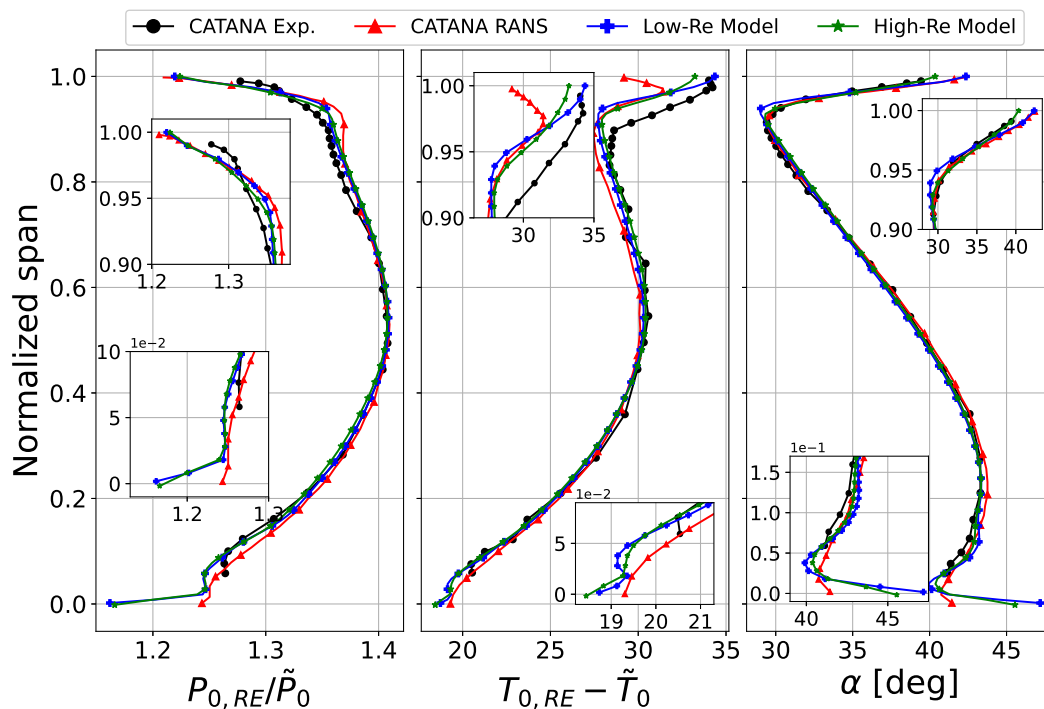


Figure 4.5: Radial distributions of normalized total pressure, total temperature, and exit flow angle α [deg]. Measured at the rotor exit (RE) plane (see Figure 3.1).

Figure 4.5 illustrates span-wise distributions of normalized total pressure, normalized total temperature, and exit flow angle α at the *RE* plane located downstream of the rotor leading edge. $P_{0,RE}$ and $T_{0,RE}$ is the area circumferential average (ACA) of the total pressure and temperature at the *RE* plane, and \tilde{P}_0 and \tilde{T}_0 is the reference total pressure and temperature. The α parameter illustrates the circumferential flow angle of the flow exiting the rotor trailing edge and assuming axial flow upstream of the rotor leading edge, the parameter represents the absolute turning exerted on the fluid by the rotor design in the stationary frame of reference. The spanwise distributions are illustrated for the low- and high-Re model compared to RANS and experimental CATANA data extracted from the CATANA/ECL5 publication. Experimental data at the walls is not part of the publication and the full dataset is illustrated in figure 4.5. Free stream parameters of the numeric solutions match the experimental data at the RE line with deviations forming in the direction of the hub and casing.

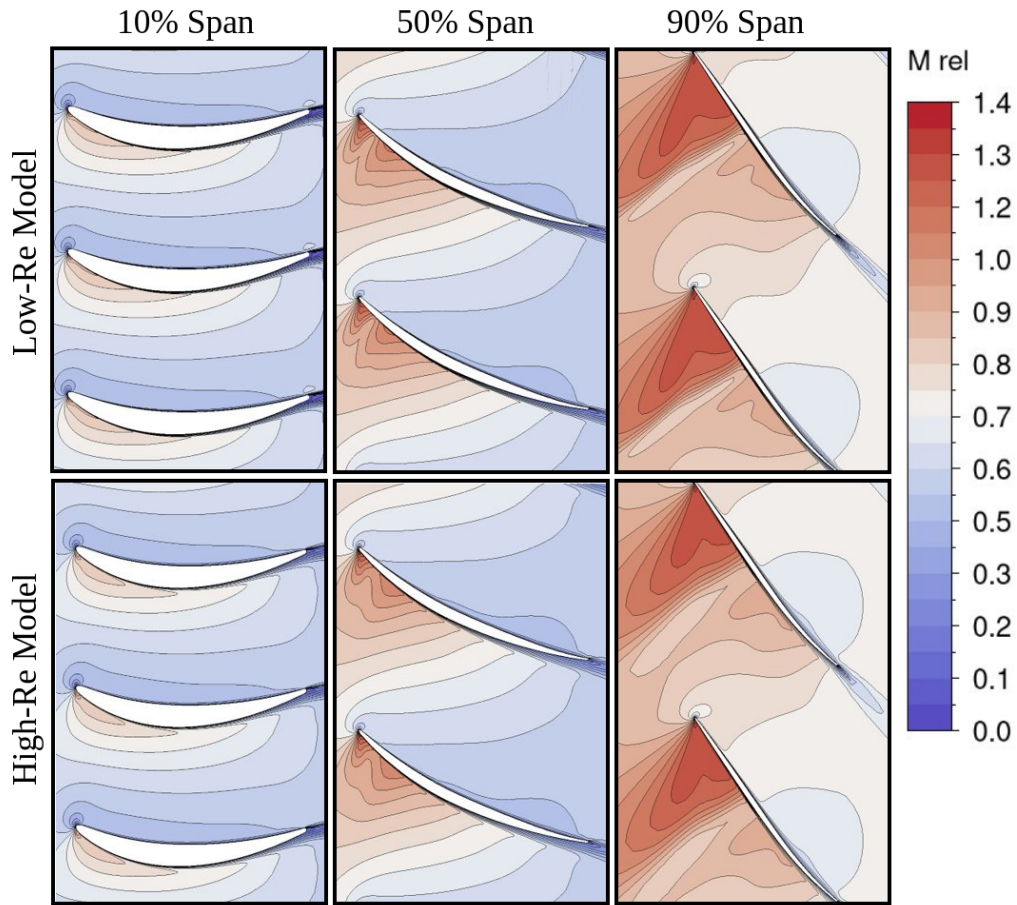


Figure 4.6: Relative Mach number number at 10/50/90 % span.

Figure 4.6 illustrates contours of the relative Mach number at a meridional stream surface projected on a 2D plane located at 10/50/90 % span of the rotor. At 10 and 50 % span the adverse pressure gradient can be observed to induce a thickening of the boundary near the trailing edge on the suction side of the blade. At 50 % span early onset of a shock can be observed on the suction side downstream of the blade leading edge. At 90% span, a normal shock occurs at mid-span on the suction side. At the shock location, the profile is designed with minimal local turning to aid in the recovery of the surface velocity on the blade. The bulk of the turning is therefore located further downstream closer to the trailing edge. This design can be seen to induce a shock recovery with low indications of problematic boundary layer thickness regarding numerical instabilities or separation. On the pressure side, however, the increase in turning near the trailing edge can be seen to induce a high pressure zone with a noticeable thickening of the boundary layer. Comparing wall models the high-resolution low-Re model can be seen to induce a relative delay in the boundary layer development, moving it further downstream on the blade surface. The earlier onset of thickening of the boundary layer on the blade surface with the high-Re model implementation indicates higher profile losses in the form of greater pressure drag. The indication of a thicker boundary layer in the high-Re model could further lead to numerical instabilities as the wake profile thickness increases, indicating that the high-Re model will approximate an earlier

onset of stall at off-design conditions. Further, the implementation of the low-Re model can be observed to delay the normal shock at 90% span on the blade surface relative to the high-Re model, inducing a shock with greater magnitude and blade wall normal extent.

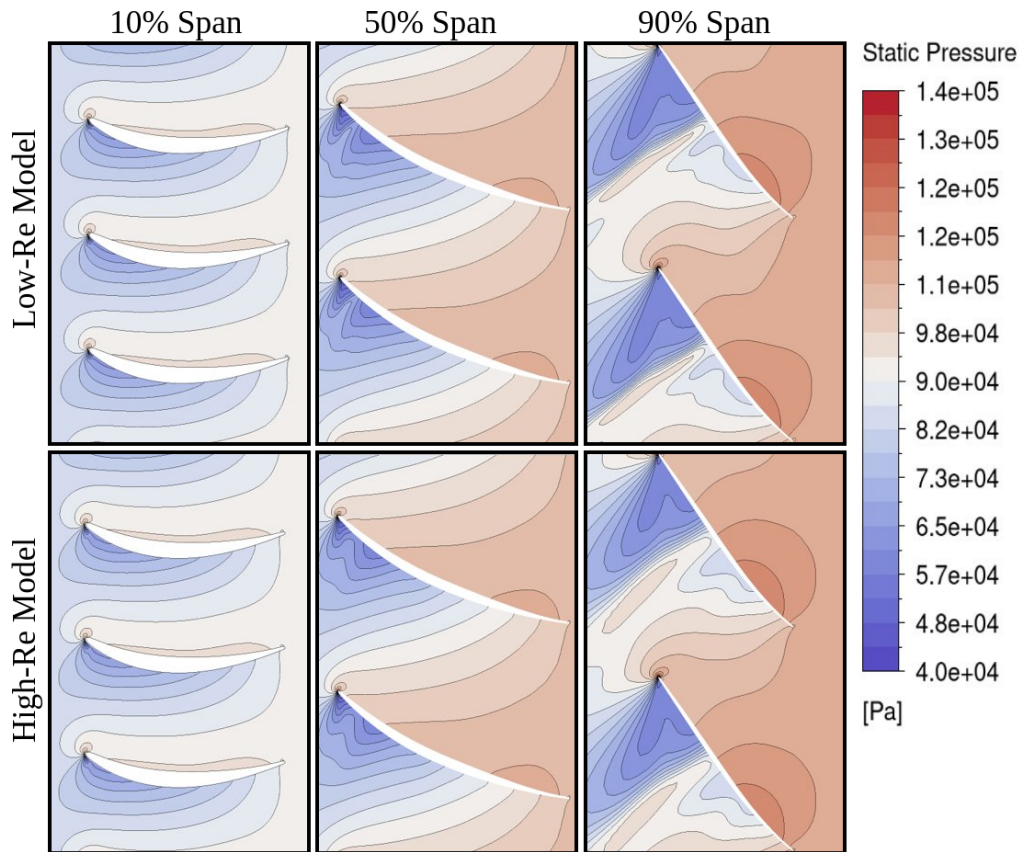


Figure 4.7: Static pressure profiles at 10/50/90 % span.

Figure 4.7 illustrates contours of the static pressure distribution at 10/50/90 % span of the rotor for the high- and low-Re model. The difference in resulting static pressure distribution due to the wall modeling technique can be observed at 90% span as the low resolution of the high-Re model illustrates a lower static pressure in the blade passage and an extended shock transition region indicated by the larger high pressure zone downstream of the normal shock on the suction side of the blade before the starts accelerating. Equivalent behavior can be observed at 50% span close to the leading edge of the blade but at a reduced scale.

4.3 Off-Design Performance

The speedline algorithm referenced in section 3.4.2 is utilized on the low and high Reynolds number models for modeling the boundary layer flow. The algorithm is capable of estimating numerical stall in the compressor design by sequentially lowering the outlet corrected mass flow rate boundary condition, inducing a greater pressure ratio over the fan stage. The choke point of the design is manually estimated from CATANA publication and defined at a value that renders a successful initial solution without a defined prior known flow field at the inlet.

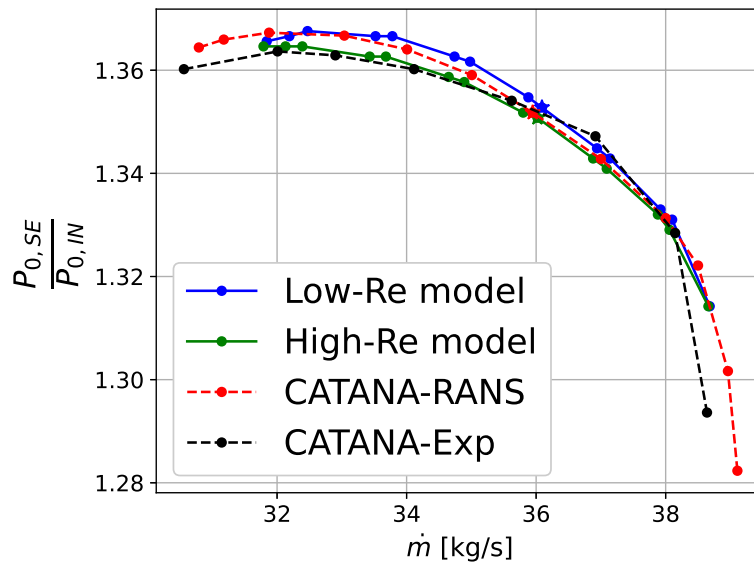


Figure 4.8: Off-design performance (speedline) for pressure ratio and mass flow rate over the CATANA fan stage.

Figure 4.8 illustrates the generated speedline for the fan stage with incorporated low and high Reynolds number models. CATANA experimental and URANS data is extracted via plot digitize, for exact data see original publication [1]. The initial approximation of choke point in the compressor design is found at a corrected mass flow rate of 30 [kg/s] for the current configuration. Stall illustrated in Fig 4.8 is firstly approximated utilizing the low-Re high-resolution model, the data illustrated for the high-Re low-resolution model is cut on an equivalent corrected mass flow rate as the model was capable of operating at a higher pressure ratio. The low-Re model can be seen to render off-design performance in line with CATANA URANS data, deviating from the experimental setup for a higher pressure ratio. The high-Re model can be observed to fit experimental data a high pressure ratio operating conditions, further deviating at high mass flow conditions approaching the solution of the CATANA URANS and low-Re model data.

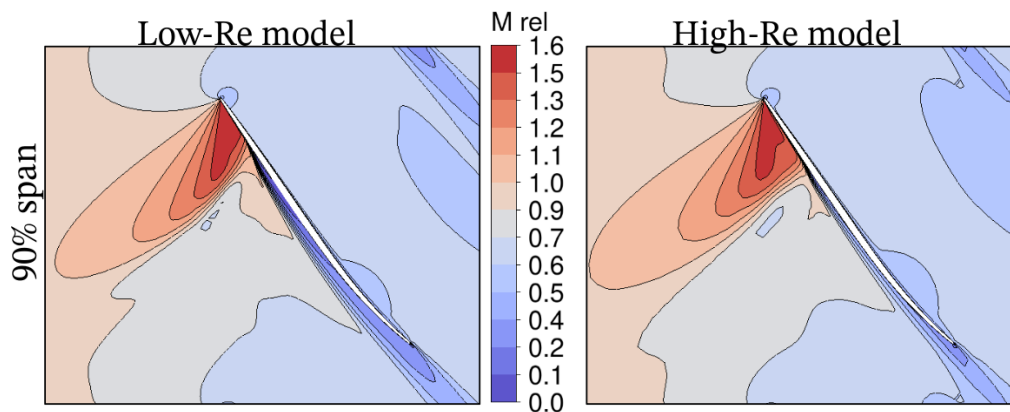


Figure 4.9: Relative mach number at 90%-span, stall operating point.

Figure 4.9 illustrates the flow field across the rotor row at 90% span (close to the tip of the blade) obtained for the low and high Reynolds number models. The high-Re model solution was capable of operating at a higher pressure ratio compared to the low-Re model. The cause is believed to be due to the reduced resolution and logarithmic approximation of the viscous forces in the wall adjacent flow. In both models separation of the boundary layer can be observed downstream of the normal shock on the suction side of the blade. In the high-Re model, a relative delay of the shock location can be observed inducing a smaller blade normal extent of the separated flow. As mentioned in section 2.3 the rotating stall patches developed by the reduced flow area from separated flow, are proportional to the area reduction, meaning that a thicker wake downstream of the separation point induces a greater contraction of the passage flow and thus increases the numerical instabilities. The illustration therefore indicates that the high-Re model predicts stall at a higher pressure ratio. Considering that the CATANA URANS data illustrates (figure 4.8) a framework capable of operating at a higher pressure ratio compared to current low-Re model implementation, the capability of the high-Re model to operate at a higher pressure ratio can be beneficial for a direct comparison with the experimental/CATANA URANS data at the stall point. The high-Re model however overestimates the stall point beyond what is illustrated in the speedline, indicating that it is less suited for stall estimates.

4.4 Stand alone Rotor

A study was carried out in which the stator was removed from the fan stage. The main objective is to try and analyze how the downstream positioned stator affects the flow across the rotor, with the intent to see if it is possible to remove the stator in future optimization studies of the rotor blade. Including the stator in any optimization of the design will lead to a larger number of design variables and further increase the overall resource requirement. When altering the current rotor design variations in the radial distribution of exit flow angles will change and thus alterations to the stator would be necessary to fit the new design. The idea is therefore to see if it is possible to exclude the stator from the framework while still

obtaining a fair/relevant estimate of the rotor performance.

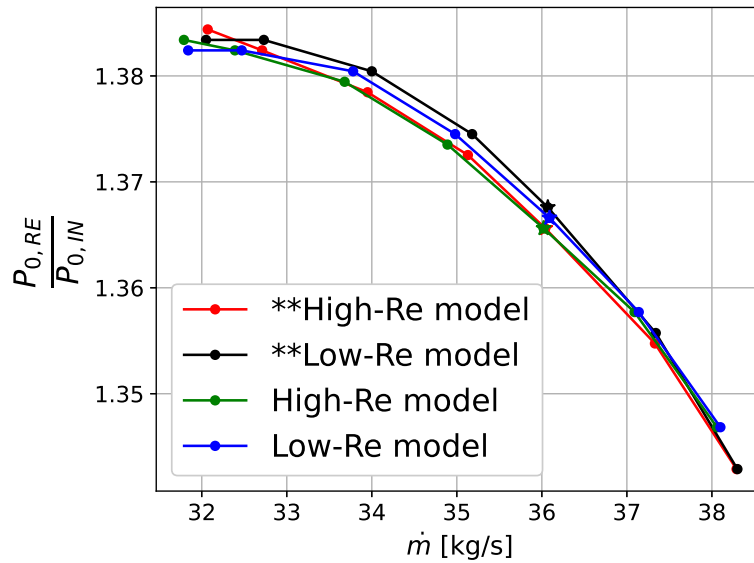


Figure 4.10: Off-design performance (speedline) for pressure ratio and mass flow rate over rotor row.

Figure 4.10 illustrates speedlines for the pressure ratio over the rotor. The mass flow averaged total pressure is extracted at the inlet (IN) plane and rotor exit (RE) plane, see figure 3.1, located at equivalent locations for all configurations. In the figure legend, the models noted with "**" denote the values obtained for the rotor only configuration, while the other data lines illustrate data extracted from the full fan stage but at equivalent axially located planes. Data points illustrated with a **star**, denote values at the design operation point. Data for original CATANA experimental/URANS data is not available at the rotor exit and is excluded from the illustration, but is further discussed in the *Conclusion* chapter.

At design condition, a small relative difference can be observed within each model, while the larger difference is present in the data when switching from a high resolution model to a low resolution model. Removing the stator from high-Re model can be observed to have minimal impact on the rotor for better part of the operation range, while approaching stall at high pressure ratios induces the data to deviate. The low-Re model can be observed to coincide with the full stage data at high mass flow but deviate when approaching the stall region. The model however indicates a similar trend when approaching stall and it is clear from the data that the pressure ratio is starting to stagnate. For the high-Re model the full stage curve can be seen to indicate the early onset of stall, while the gradient of the only rotor configuration is larger in magnitude. The deviance within each model is in general of a small magnitude and can be visually exaggerated by the fact that the off-design data points are collected at slightly different operating conditions, and the interpolation between the data points is linear.

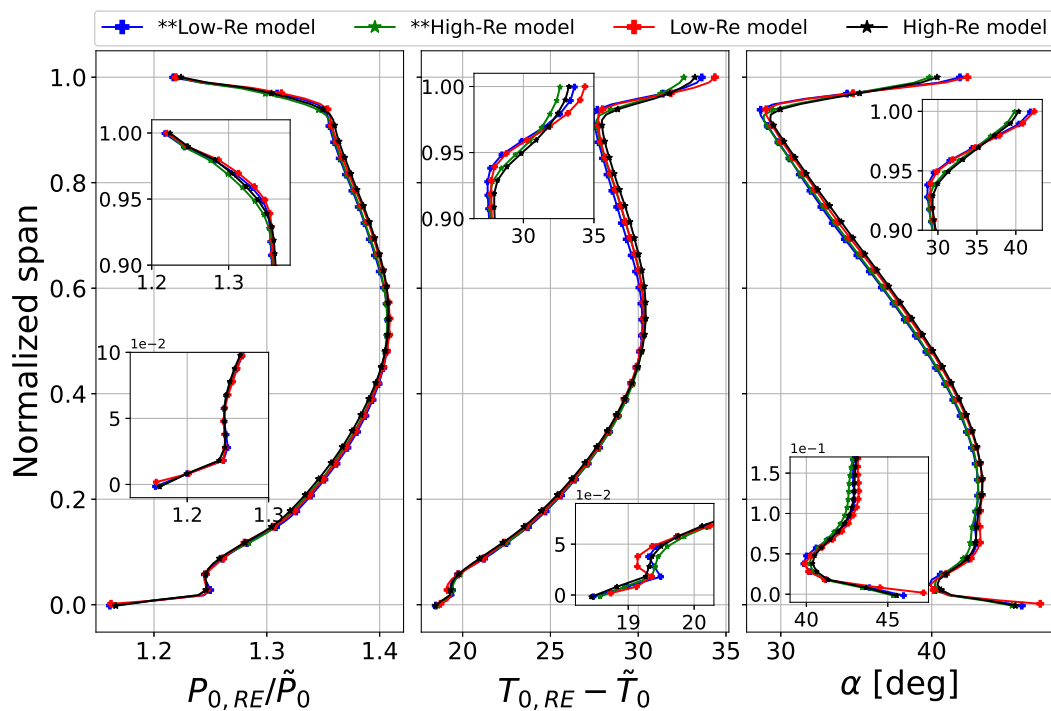


Figure 4.11: Radial distributions at rotor exit (RE) line for normalized pressure ratio, normalized temperature ratio, and exit flow angle α .

Figure 4.11 illustrates radial distributions at the rotor exit (RE) line for established numerical models for fan configuration with and without stator implementation in the flow passage. Configuration without stator is marked with "*" in the figure legend, entries without the marker are data collected from fan configuration with stator. The absence of a stator can be observed to induce a small variance in the flow field downstream of the rotor. In the free stream, small deviations can be observed, while in the wall-adjacent flow, larger deviations can be observed.

4.5 Temporal Gain Relative Numerical Model and Compressor Core Configuration.

The temporal scale for each numerical model, low- and high-Re, and configuration, with and without stator, is obtained for a PC configuration with an AMD Ryzen Threadripper 3960x (stock configuration) CPU. Each case (model+configuration) is simulated with 20 dedicated CPU partitions (physical cores), and Intel MPI partitioning is utilized. The simulations are run in TUI (Text User Interface) mode on a LINUX OS (Fedora 40). Time scale data generated by CFX Solver Manager is expressed in *CPU seconds*, corresponding to the summation of the active time of each dedicated physical CPU core, defined for one iteration as,

$$t_{CPU \text{ seconds}} = \sum_{i=1}^{partitions} t_i \quad (4.2)$$

Time scale data of actual interest is the real average iteration time for a converged solution, and is obtained by dividing the total amount of *CPU seconds* with the number of iterations and partitions,

$$\langle t_{iter} \rangle = \frac{CPU\ seconds}{iter * partitions} \quad (4.3)$$

Average iteration time	
Configuration	Avg. iter time [s]
Low-Re rotor and stator	25.87
Low-Re stand alone rotor	20.48
High-Re rotor and stator	1.96
High-Re stand alone rotor	1.58

Table 4.8: Average iteration time relative compressor core configuration and numerical setup.

The average iteration time data presented in table 4.8 are averaged over 600 iterations. The tabulated data can be observed to indicate that removing the stator from the configuration reduced the overall computational load by a factor of ~ 1.2 , while switching to a high-Re model reduced the overall computational load by a factor of ~ 13.2 . In the larger scope of integration in a parametric design study of the rotor and stator blade design, removing the stator would impose a larger gain in terms of computational load.

Presented data is highly sensitive to computer/cluster/server configuration and does not reflect the background processes, such as writing to files and GUI (Graphical User interface) related processes, but does however reflect the bulk of the simulation time which is the iteration time for the linear solver.

5

Conclusion

5.1 Thoughts About Current Work

The task at hand was to generate a framework capable of evaluating rotor performance at design and off-design operating conditions with an initial guess of operating condition at the outlet boundary, for future implementation in a parametric rotor design study. The current framework is capable of evaluating a design at multiple operating conditions for an established geometry, discretization, and numeric setup without any user input required approaching stall. Further Meshing, numeric, and post-evaluation software are linked through intermittent code developed in Python. So far the code is capable of estimating stall by gradually lowering the corrected mass flow at the annular outlet and utilizing the previously converged flow field as an initial guess to aid in numerical stability at critical operating points. The converged solution is at the current stage evaluated with modified convergence criteria geared towards establishing a minimum stable residual in critical parameters to ensure all solutions are evaluated at an equivalent order of accuracy.

Automating and developing connectivity between multiple CFD software to work seamlessly during an extended amount of time is tricky in many situations and requires a lot of time and experience to make it work. Sharp testing of the framework is very time consuming and often reveals a lot of quirks related to developing inter-connections between the software. The general workflow is however challenging and fun, and when it runs and work as expected the feel of accomplishment is rewarding.

At the start of the thesis, the goal was to achieve a small parametric study, testing the framework for multiple rotor designs. Unfortunately, the parametrization code could not be finished in time and the overall scope and challenges associated with developing the framework were more time consuming than initially thought. The framework is however tested for the CATANA rotor design with different numerical models and works as expected.

In the thesis, the validity of implementing wall functions in high speed numerics is further investigated to see if it is possible to reduce the overall computational load generally associated with higher resolution wall modeling techniques. The implementation of wall models was shown to reduce the overall cell count of a factor ~ 10 , reducing the overall computational time by a factor ~ 13.2 , for the established computer configuration in section 4.5.

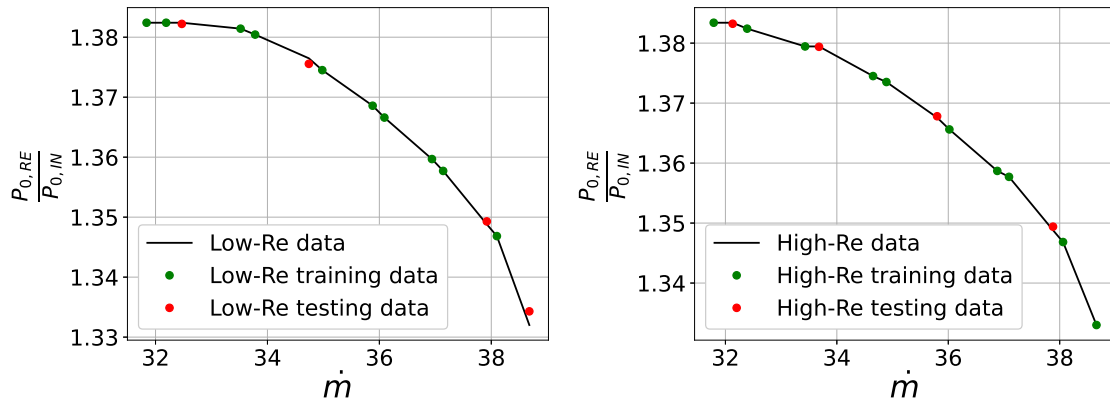
The implementation of wall functions is considered successful and is currently being implemented by others in a full scale parametric study. As the error associated with the wall model relative the RANS model is smaller than the error generated by the RANS model relative the experimental data, the high-Re model is considered an acceptable trade-off in regards to the otherwise higher computational load as equivalent trends can be observed in generated compressor maps.

Regarding the integration of CFX in Python. Creating an equivalent framework to that of the current one is difficult in the sense that it heavily pushes the limits of what the CFX integration framework is capable of. For instance, one such problem is working around the sequential execution of code. Code is always executed line by line, and in this instance, we needed Python to start a CFX-Solver session, and further start the convergence algorithm, while regaining basic control of the solver so it was possible to communicate with the process and inject further commands. This was only one of many problems faced when doing sharp testing of the framework. My experience after having worked with this for the last couple of months is that it works well enough for simple "one-off" tasks via terminal integration but gets rather tricky once the dynamic aspect is introduced and on top of that adding layers of parallel code executions. This is by no means due to the lack of possibilities present in the CFX toolbox, but rather a residual of utilizing the program beyond its current vision.

For the readers who are interested in the basic structure of the framework or interested in the CFX commands used for this thesis, Appendix 1 & 2 shows basic commands and reason for implementation and contain a full pseudo-code representation of the speedline and convergence algorithm.

5.2 Further Discussion on Rotor Independence (ML Estimates)

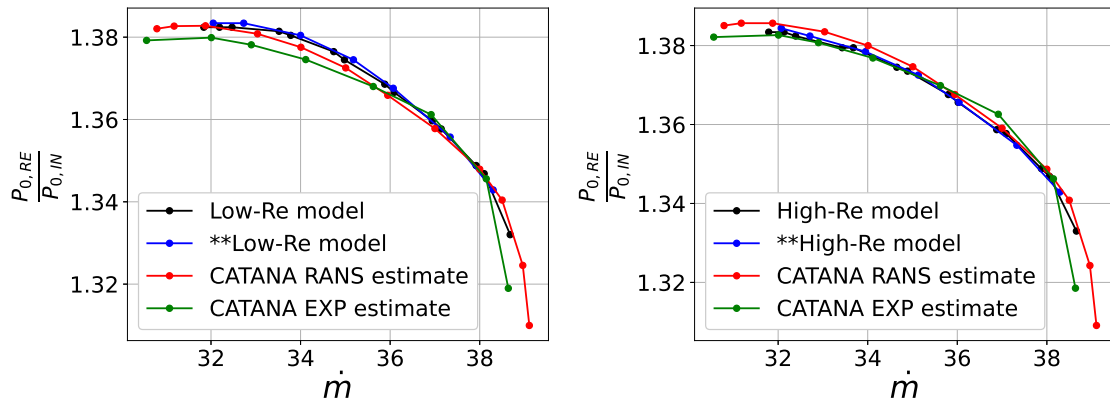
Evaluating the full extent of removing the stator from the duct is tricky, in regards to fully capturing downstream properties at the stator propagating to the rotor domain. In this thesis, a simple evaluation was conducted on comparing radial distributions and local pressure ratio over the rotor, and the results show with the current data, a possibility of removing the stator from the framework to reduce the overall computational load in a parametric study of the rotor design. To further validate the current stand alone rotor framework to the original CATANA data, radial distributions of the flow field are needed at the rotor exit plane. The data is not currently available and is therefore excluded from the overall results presented in this thesis. However, it is possible to use the current off-design data obtained for the stage design with the stator, and train a machine learning (ML) algorithm to estimate properties downstream of the rotor, and further apply the ML algorithm to the original CATANA data. As a proof of concept, a linear support vector regression (Linera-SVR) ML algorithm was trained on the obtained off-design data over the rotor and stator, and applied to the CATANA data.



(a) Low-Re Linear-SVR ML-algorithm. (b) High-Re Linear-SVR ML-algorithm.

Figure 5.1: Liner-SVR ML-algorithms, trained and tested on data obtained from full stage Low- and High-Re models.

Figure 5.1 illustrates two Liner-SVR ML-algorithms trained on estimating local pressure ratio over the rotor from data for pressure ratio over rotor and stator. Figure 5.1a illustrates ML-algorithm trained and tested on data extracted from the Low-Re framework. Figure 5.1b illustrates ML-algorithm trained and tested on data extracted from High-Re framework. The graphs illustrate the original local pressure ratio obtained from the low- and high-Re models (black line), and the data used for training the algorithm (green dots) as well as the testing samples (red dots). The algorithm can be seen to estimate a local rotor pressure ratio close to that of the original data for the four testing samples. With a validated model it can be applied to the original CATANA URANS and experimental data to estimate the local pressure ratio over the rotor.



(a) Low-Re Linear-SVR ML-algorithm. (b) High-Re Linear-SVR ML-algorithm.

Figure 5.2: Liner-SVR ML-algorithms applied to CATANA experimental and URANS data compared to rotor only cases.

Figure 5.2 illustrates trained ML-algorithms applied to CATANA experimental and URANS data extracted from publication, as well as the rotor pressure ratio obtained

from rotor only cases with low- and high-Re models. Legend entries denoted with "***" indicates rotor pressure ratio obtained for cases with a removed stator from the flow channel. Figure 5.2a illustrates the ML-model trained on low-Re data applied to CATANA data compared to original speedlines obtained from full stage design and rotor only design. Figure 5.2b illustrates ML-model trained on high-Re data applied to CATANA data compared to speedlines obtained from rotor only and full stage simulations.

ML-models can be seen to indicate that the original CATANA data fits well with generated compressor maps for the different cases. The validity of data can be debated as the models is trained on steady-state solutions and applied to transient data, but is only meant to indicate and further validate that removing the stator from the domain renders minimal impact on the rotor evaluation. The data set is also rather small to accurately train the mode, which could be further improved with a larger data set.

5.3 Future Work

Expanding on the current work involves integrating choke point analysis in compressor map evaluation and integrating the framework in a parametric study for different blade designs. The currently generated Python based code is developed as a standalone module for evaluating a specific blade design and can be easily integrated into different frameworks as long as CFX is to be used as the primary simulation tool. The framework is specifically developed with a future parametric study in mind but can be integrated in a range of development tools.

Further independence of the rotor design is probably needed as the data presented in the thesis only concerns a specific blade design. Altercations to the current design could potentially induce flow phenomena in the stator domain propagating upstream to the rotor. The concept of hot and cold design could also further be investigated to account for deformation during operation, and thus integration FSI into the framework.

Bibliography

- [1] Alexandra P. Schnider, Anne-Lise Fiquet, Benoit Paoletti, Xavier Ottavy, Christoph Brandstetter. (2023) Université de Lyon, École Centrale de Lyon. **Experiments on Tuned UHBR Open-Test-Case Fan ECL5/CATANA: Performance and Aerodynamics**, GT2023-103629.
- [2] Ismail B. Celik, Urmila Ghia, Patrick J. Roache, Christopher J. Freitas, Hugh Coleman, Peter E. Raad. (2008) Journal of Fluids Engineering, Transaction of ASME. **Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications**. Vol. 130 / 078001-1
- [3] Marcus Lejon, Niklas Andersson, Lars-Erik Eriksson, Lars Ellbrant. (2015) Proceedings of ASME. **Simulations of Tip-Clearance Effects In a Transonic Compressor**, GT2015-43033.
- [4] P. Catalano. **Transonic flow simulations using wall functions**, European Conference For Aerospace Sciences, Italian Aerospace Research Center, Italy.
- [5] P. Catalano, P.L. Vitagliano. **A Scalable Wall Function Approach for High Reynolds Number Flows**, European Conference For Aerospace Sciences, Italian Aerospace Research Center, Italy.
- [6] L. Davidsson. **Fluid mechanics, turbulent flow and turbulence modeling**. Chalmers University of Technology, Department of Mechanics and Maritime Sciences, 2023.
- [7] S.L. Dixon, C.A Hall. (2014). **Fluid Mechanics and Thermodynamics of Turbomachinery** (Seventh Edition), Elsevier Inc.
- [8] H.K. Versteeg, W. Malalasekera. (2007). **An Introduction to Computational Fluid Dynamics The Finite Volume Method** (Second Edition), Pearson Education Limited.
- [9] Jhon D. Anderson. (2021). **Modern Compressible Flow With Historical Perspective** (Fourth Edition), McGraw-Hill Education.
- [10] H.I.H. Saravanamuttoo, G.F.C. Rogers, H. Cohen, P.V. Straznicky, A.C. Nix. (2017). **Gas Turbine Theory** (Seventh Edition), Pearson Education Limited.
- [11] Saeed Farokhi. (2022). **Aircraft Propulsion Cleaner, Leaner, and Greener** (Third Edition), John Wiley & Sons, Inc.

- [12] Ansys, Inc. (2021). **Ansys TurboGrid User's Guide (R2)**, Ansys, Inc. and Ansys Europe, Ltd. https://ansyshelp.ansys.com/account/secured?returnurl=/Views/Secured/corp/v201/en/tg_user/tg_user.html
- [13] Ansys, Inc. (2021). **Ansys CFX-Pre User's Guide (R2)**, Ansys, Inc. and Ansys Europe, Ltd. https://ansyshelp.ansys.com/account/secured?returnurl=/Views/Secured/corp/v201/en/cfx_pre/cfx_pre.html%23cfx_pre
- [14] Ansys, Inc. (2021). **Ansys CFX-Solver Theory Guide (R2)**, Ansys, Inc. and Ansys Europe, Ltd. https://ansyshelp.ansys.com/account/secured?returnurl=/Views/Secured/corp/v212/en/cfx_thry/cfx_thry.html
- [15] Ansys, Inc. (2021). **Ansys CFX-Solver Modeling Guide (R2)**, Ansys, Inc. and Ansys Europe, Ltd. https://ansyshelp.ansys.com/account/secured?returnurl=/Views/Secured/corp/v212/en/cfx_mod/cfx_mod.html
- [16] Ansys, Inc. (2021). **Ansys CFX-Solver Manager User's Guide (R2)**, Ansys, Inc. and Ansys Europe, Ltd. https://ansyshelp.ansys.com/account/secured?returnurl=/Views/Secured/corp/v212/en/cfx_solv/cfx_solv.html
- [17] Cadence, Learning & Support. (2023). **Fidelity Pointwise User Manual**, Cadence, Learning & Support. https://ansyshelp.ansys.com/account/secured?returnurl=/Views/Secured/corp/v212/en/cfx_solv/cfx_solv.html
- [18] Launder, B.E. and Spalding, D.B. **Comp Meth Appl Mech Eng**, **3:269-289**, **1974**, *The numerical computation of turbulent flows*
- [19] Menter, F.R. **AIAA-Journal.**, **32(8)**, pp. **1598 - 1605**, **1994**, *Two-equation eddy-viscosity turbulence models for engineering applications*

A

Appendix 1

This section of the Appendix covers implemented CFX batch commands used for CFX integration in python.

Command:

```
·cfx5pre -batch <session-file>
```

The command is used to start a CFX-Pre session in batch mode that executes the commands given in the session-file. The session file is generated by a separate script that takes the mesh files as input, as well as some solver settings, and integrates the input in to a template and outputs a case specific session file. The **output** of the command is a case definition file that is used as input for the CFX solver.

```
·cfx5solve -def <definition-file> -initial-file <results-file> -start-method Intel MPI  
Local Parallel -par-local -partition <partition number> -batch -chdir <case-directory>
```

The command is used to start a CFX-Solve session with a few specific input arguments. *definition-file* is the file generated by the *cfx5pre* command. *results-file* is part of the *-initial-file* argument and tells the solver to start with an initial guess for the flow field extracted from a previously converged solution, the input file needs to be a **.res* or **.bak* file. The *-start-method* argument tells the solver to start with a specific MPI method that is configured on the system. The *-par-local* argument tells the solver that the user is to manually allocate CPU partitions, and *-partition* is the partition number, limited to the amount of CPU cores available. *-batch* runs the program in batch mode, and *-chdir* is utilised to change the directory in which the solver operates and files are saved.

```
·cfx5post -batch <post-state-file> <results-file>
```

The command is used to start a CFX-Post session and post process the data generated by the CFX solver. The input is a *post-state-file* that contains a template for which commands CFX-post is to execute, and *results-file* is the **.res* or **.bak* file generated by CFX-Solve.

```
·cfx5mondata -dir <run-directory> -varrule CATEGORY=<variable-name> -out  
<user-out-file>
```

The command is used to extract monitor data from current CFX-Solve session and is used by the convergence algorithm to continuously monitor the currently running simulation. The *-dir* argument is a pointer to the working directory generated by the *cfx5solve* command. The *-varrule* argument is a pointer to the variable that is to be extracted from the solver, where variable name is the user input in the argument. This command is a bit restricting in that it can only extract variables from the equivalent CFX category, so for this use case it was called three times to extract data for all three variables. The *-out* argument tells the command the file name to which the data is to be piped to.

```
·cfx5control <run-direcotory> -inject-commands <ccl file> -no-backup » <log-file>
```

This command is a part of generating a speedline for a given compressor design. The command is used to inject commands to currently running solver, allowing for alterations to current solver settings, and thus allowing for changes to the operating condition for the design during run-time. The command is integrated in the framework due to a request from the project supervisor. The intent of the command is to allow for dynamic alterations of the currently running solution, to avoid exiting the solver and having to wait in the cluster queue to run the design at a new operating point. The initial input argument is the solver working directory and the *inject-commands* argument is a pointer to a **.ccl* file containing the relevant changes to solver. The caveat to this argument is that it requires a full *ccl* file with all solver settings, regardless if changed or not. The best way of obtaining a *ccl* file template is to start a solution in the GUI interface and manually make some alterations to the solver. A backup *ccl* file will then be created with all relevant settings for the solver. In order to make this dynamic one can use the backup *ccl* file as a template in a python script and format it to fit a new case, or for a one time use case for an equivalent case setup, one can manually alter the file and use as input in the *inject-commands* argument. For this thesis a dynamic implementation was required and therefore a separate script was created. The *-no-backup* argument is used to explicitly tell the solver to not save a backup of the currently running case. This is done to avoid the CFX naming integration of output **.bak* files. CFX names its backup files based on a randomly generated string of numbers, which for post processing integration is not ideal. In order to avoid this the command is firstly executed with only a *-backup* flag when the current operating point is converged, and when this command is done, the backup file is dynamically moved with UNIX commands and renamed so it can be associated with its respective operating point. When these lines of code is successfully executed the algorithm waits for the backup file to finish writing so to not overlap data in the files, and then the full command is executed to alter the currently running simulation. The *» <log-file>* is not part of the CFX native commands, but is there to log the CFX response whenever the new *ccl* file is injected in the solver.

This command is further used with the flag *<run-directory> -stop* to exit a solver whenever necessary.

B

Appendix 2

This section of the appendix includes pseudo-code for for the current speedline and convergence algorithms developed in python.

Algorithm 1 Convergence algorithm

```
1: while CFX-solver is running do
2:   if number of CFD iterations  $\geq$  RA window size then
3:     Call "cfx5mondata..."  $\rightarrow \eta_c, \dot{m}, pr$  time step history
4:      $RA_{pr}^{RMSD} \leftarrow pr$ 
5:      $RA_{\eta_c}^{RMSD} \leftarrow \eta_c$ 
6:      $RA_{\dot{m}}^{RMSD} \leftarrow \dot{m}$ 
7:     if all( $RA_{pr}^{RMSD}, RA_{\eta_c}^{RMSD}, RA_{\dot{m}}^{RMSD}$ )  $\leq$  convergence tolerance then
8:       Stop CFX-solver
9:       Solver converged  $\leftarrow True$ 
10:    else if CFD iterations = max iterations then
11:      Stop CFX-solver
12:      Solver converged  $\leftarrow False$ 
13:    end if
14:  end if
15: end while
```

Algorithm 2 Speedline algorithm

```

1: case_number = i
2: outlet_bc = X
3: delta_bc = Y
4: min_delta_bc = Y/4
5: current_iterations = 0
6: RA_window_size = n
7: Generate CFX-pre session file → *.pre file
8: Run "cfx5pre -*.pre ..." → *.def file
9: Run "cfx5solve -*.def ..."
10: CFX-solver is running is True
11: while CFX-solver is running do
12:   if current_iterations ≥ RA_window_size then
13:     Call "cfx5mondata..." →  $\eta_c, \dot{m}, pr$ , time step history (t)
14:     current_iterations = length(t)
15:      $RA_{pr}^{RMSD} \leftarrow pr$ 
16:      $RA_{\eta_c}^{RMSD} \leftarrow \eta_c$ 
17:      $RA_{\dot{m}}^{RMSD} \leftarrow \dot{m}$ 
18:     if all( $RA_{pr}^{RMSD}, RA_{\eta_c}^{RMSD}, RA_{\dot{m}}^{RMSD}$ ) ≤ convergence tolerance then
19:       Call "cfx5control <run-directory> -backup" ← Generate backup
20:       while Backup is writing do
21:         Wait...
22:       end while
23:       case_i_X.bak ← Rename *.bak file
24:       outlet_bc = outlet_bc − delta_bc
25:       case_i_X.ccl ← Generate *.ccl file with new outlet boundary condition
26:       Call "cfx5control <run-directory> -inject-commands <ccl-file> -no-backup"
27:       current_iterations = 0
28:     end if
29:     if current_iterations = max iterations then
30:       if delta_bc = min_delta_bc then
31:         Stop CFX-solver
32:         Solver converged ← False
33:         CFX-solver is running is False
34:       else:
35:         Stop CFX-solver
36:         Solver converged ← False
37:         delta_bc = delta_bc/2
38:         Generate CFX-Pre file for new boundary condition → *.pre file
39:         Run "cfx5pre -*.pre ..." → *.def file
40:         Run "cfx5solve -*.def ..."
41:         current_iterations = 0
42:       end if
43:     end if
44:   end if
45: end while
46: Call "cfx5post.... for all *.bak files" → post data

```

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY