ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

STUDY OF TURBOMACHINERY FLOWS USING OPEN SOURCE ANALYSIS SOFTWARE

M.Sc. THESIS

Büşra ÇETİN

Department of Defense Technologies

Defense Technologies Programme

DECEMBER 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

STUDY OF TURBOMACHINERY FLOWS USING OPEN SOURCE ANALYSIS SOFTWARE

M.Sc. THESIS

Büşra ÇETİN
(514191037)

Department of Defense Technologies

Defense Technologies Programme

Thesis Advisor: Prof. Dr. Fırat Oğuz EDİS

DECEMBER 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

AÇIK KAYNAKLI ANALİZ YAZILIMI KULLANARAK TURBOMAKİNE AKIŞLARININ İNCELENMESİ

YÜKSEK LİSANS TEZİ

Büşra ÇETİN
(514191037)

Savunma Teknolojileri Anabilim Dalı

Savunma Teknolojileri Programı

Tez Danışmanı: Prof. Dr. Fırat Oğuz EDİS

ARALIK 2022

Büşra Çetin, a M.Sc. student of İTU Graduate School student ID 514191037, successfully defended the thesis/dissertation entitled "STUDY OF TURBOMACHINERY FLOWS USING OPEN SOURCE ANALYSIS SOFTWARE", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Fırat Oğuz EDİS** ....
İstanbul Technical University

Jury Members : **Prof. Dr. Aydın MISIRLIOĞLU** ....
Istanbul Technical University

Doç. Dr. İlyas YILMAZ ....
Bilgi University

Date of Submission : 15 December 2022
Date of Defense : 19 December 2022

To my mother,

FOREWORD

I would like to express my sincere gratitude to my supervisor Prof. Dr. Fırat Oğuz Edis for his continuous interest and guidance during this study, and and for his support within that period.

I am also grateful to my friend and colleague Ece Saykan for her support and encouragement.

Computing resources used in this work were provided by the National Center for High Performance Computing of Turkey (UHeM) under grant number 4010432021.

December 2022

Büşra ÇETİN
(Researcher)

TABLE OF CONTENTS

Page

ABBREVIATIONS

CFD: Computational Fluid Dynamics

GUI: Graphical User Interface

CDP: Compressor Design Program

MCA: Multiple Circular Arc

NS: Navier-Stokes

RANS: Reynolds Averaged Navier-Stokes

SIMPLE: Semi-Implicit Method for Pressure-Linked Equations

BC: Boundary Condition

AMI: Arbitrary Mesh Interface

STL: Stereolithography

NOMENCLATURE

Latin Characters

c_{b1}, etc.	empirical constants in the turbulence model
c_p	specific heat for constant pressure
c_v	specific heat for constant volume
d	distance to the nearest wall for turbulence model
E	total energy
e	specific internal energy
g, **r**, $\tilde{S}$	intermediate variables in the turbulence model
H	total enthalpy
h	specific enthalpy
k	turbulent kinetic energy
Pr	Prandtl number
p	pressure
q	heat flux
s	strain-rate
u	velocity
T	temperature
t	viscous stress

Greek Characters

δ	Kronecker delta
γ	specific heat ratio
κ	Karman constant in the turbulence model
μ	dynamic viscosity
ν	kinematic viscosity
ψ	compressibility
ρ	density
σ	turbulent Prandtl number for turbulence model
τ	Reynolds stress

χ intermediate variable

ζ second viscosity

Superscripts

$\bar{q}$ Reynolds-averaged

q' fluctuation in Reynolds decomposition

$\tilde{q}$ Favre-averaged

q'' fluctuation in Favre decomposition

q^* estimated value by pressure correction

q' correction value in pressure correction equation

LIST OF TABLES

Page

LIST OF FIGURES

STUDY OF TURBOMACHINERY FLOWS USING OPEN SOURCE ANALYSIS SOFTWARE

SUMMARY

Construction of a proper mesh is a crucial stage of computational studies to obtain acceptable results as the quality of the mesh specifies accuracy, convergence and computational time. Solution is mainly restricted by imperfections on the geometry definition, low mesh quality and relatively coarse mesh structure. OpenFOAM as an open source software, which offers tools to generate grids for the solution domain in addition to execute CFD analysis. For complex geometries, mesh can be constructed by snappyHexMesh utility of OpenFOAM. snappyHexMesh creates an unstructured mesh over a structured state that is generated by blockMesh utility thus it presents an automatic mesh construction tool for engineering applications.

In this study, NASA Rotor 37 that is an axial flow, transonic compressor delivering a pressure ratio of 2.1 at design conditions, is analyzed. OpenFOAM was used for both mesh construction and CFD analysis. An unstructured mesh was generated by snappyHexMesh. Since the solution domain contains periodic boundaries, cyclic boundary condition was used. Generation of cyclic BCs on an unstructured mesh are a difficult process by an open source tool as two periodic boundaries must be identical in order to overlap grids on these boundaries. The cyclic surfaces were produced as circular patterns of each other. Quality of the geometry file is also an important factor for mesh construction. Because of unstructured results of snappyHexMesh with non conformal grids, cyclic boundary condition option could not be used solitarily. Non conformal periodic boundaries are matched by cyclicAMI boundary condition options of OpenFOAM that states Arbitrary Mesh Interface and contructs the mesh matching for non overlapping grids. Integration of cyclicAMI was applied by an exclusive OpenFOAM dictionary, createPatchDict. Thus, for obtaining a mesh automatically, geometry files must have high quality, and periodic boundaries must be geometrically identical.

Flow field of the rotor was evaluated in terms of total pressure and temperature ratios, and Mach number distributions. Spalart-Allmaras turbulence model was used. Computational results were compared with experimental measurements and other computational results. According to the results, method of preparation of the geometry and construction of the mesh affect accuracy considerably. Total temperature and pressure under predicted in comparison to experiment and other computational studies. It may be caused by rhoSimpleFoam that cannot capture shock, or by low quality geometry. Hence a proper solver and high quality geometry can give satisfying results.

AÇIK KAYNAKLI ANALİZ YAZILIMI KULLANARAK TURBOMAKİNE AKIŞLARININ İNCELENMESİ

ÖZET

Hesaplamalı akış problemlerinde çözümü etkileyen birçok unsur vardır. İncelemeye tabi tutulan geometrinin şekli ve eğer yüzeyi üçgenlerden ibaretse üçgenlerin yapısı, inşa edilecek örgü ağının mahiyetini belirler. Örgü ağı da çözümün doğru ilerlemesi için çok önemlidir. Uygun bir örgü ağının inşası, makul sonuçlar elde etmek için hesaplamalı çalışmaların mühim bir merhalesidir. Örülen ağın kalitesi sonuçların doğruluğunu, yakınsamasını ve hesaplama zamanını belirler. Ağ inşa etmek için kullanılan birçok yazılım programı mevcuttur. Açık kaynaklı bir yazılım olan OpenFOAM, akışkanlar dinamiği problemlerini çözmesinin yanında örgü ağı inşa etmek için yardımcı araçlar içerir. Basit geometriler için yapılandırılmış örgü ağı oluşturan blockMesh aracı, yeterli koordinat bilgisi girilirse karmaşık geometriler için de ağ örebilir. OpenFOAM karmaşık geometriler için örgü ağı inşa edebilen snappyHexMesh isimli bir araca sahiptir. Böylelikle mühendislik uygulamaları için otomatik bir ağ örme aracı sunar.

snappyHexMesh yapılandırılmış, arka plandaki bir ağ üzerinden yapısal olmayan bir örgü ağı oluşturur. Uygun formatta bir geometriyle ve onu içine alacak şekilde blockMesh veya başka bir araçla inşa edilmiş düzgün bir örgü ağıyla çalışmaya başlar. Geometri, kaleleştirme, yakalama, sınır ekleme ve ağ kalite kontrolü olmak üzere beş ana bölümden oluşur. Geometri bölümünde, geometri dosyası tanımlanır. Gerekirse düzlem, kutu gibi basit geometriler de burada tanımlanarak oluşturulabilir. Ardından gelen kaleleştirme bölümünde, arka plan ağında bulunan ve ana geometride olmayan gridler temizlenir. Bunun için geometrinin içinde kalan herhangi bir noktanın koordinat bilgisi girilmelidir. Kaleleştirmede tanımlanan bir diğer özellik örgü ağında yapılacak iyileştirmelerdir. Kenarlarda, yüzeylerde ve mevcutsa kapalı geometrilerin içinde veya dışında istenilen iyileştirme seviyeleri tanımlanır. Kaleleştirme sonucunda bölgesel iyileştirmelere tabi tutulmuş, ana geometriyi ortaya çıkaran bir yapı ortaya çıkar ancak kenar ve yüzey hatları tamamen düzeltilmiş değildir. Sonraki aşama olan yakalama bölümünde, ana geometrinin hatları yakalanır. Burada tanımlanan özelliklerde farklı iterasyon sayıları girilir. Kenarları ana geometriyle aynı olacak şekilde düzeltme ve geometriyi iyileştirme amacıyla belirlenen bu iterasyon sayıları artırılarak en iyi ağ yapısı elde edilir. Son ağ düzenleme bölümü olan sınır eklemede geometrinin istenilen bölgesi üzerine sınırlar eklenir. Eklenecek sınırların sayısı ve boyutları tanımlanır. Geometriye sınır eklenebilmesi için, diğer aşamaları geçen örgü ağı yapısal olarak kaliteli olmalıdır. Yani snappyHexMesh bölümlerindeki özellikler farklı geometriler için uyarlanır ve bu da yazılımın karmaşık geometriler için otomatik ağ örme aracı olduğu anlamına gelir.

Bu çalışmada transonik akışlı, tasarım şartlarında 2.1 basınç oranına ulaşan NASA Rotor 37 isimli kompresör rotoru tetkik edilmiştir. Rotor 37 hesaplamalı turbomakine

çalışmalarında yaygın olarak kullanılan bir rotordur. Çünkü deneysel sonuçları yayınlanmıştır ve bunun üzerine yapılan bir çok hesaplamalı çalışma da mukayeseli analiz imkanı sunar. Rotorun akış alanında şok, girdap ve sınır tabaka etkileşimlerinden doğan görece karmaşık yapılar ortaya çıkar. Bunları yakalayabilmek için analizde kullanılacak geometrinin kalitesi, örülen ağın yapısı ve kalitesi, kullanılan sayısal yöntemler ve türbülans modeli oldukça önemlidir. Bu unsurları sınamak için çeşitli çalışmalar yapılmıştır. Rotorun aerodinamik yapısını örnek alıp yeni bir kanatçık oluşturmak, çözüm yöntemlerinin grid sayısına ve yapısına olan bağımlılığını ölçmek, farklı sayısal yöntemlerden oluşan çözüm algoritmalarını test etmek gibi çeşitli amaçlarla yürütülen çalışmalarda rotorun akış alanı analiz edilmiştir.

Rotor geometrisi bilgisayarlı tasarım programında çizilmiştir. Kanatçığı çizmek için NASA tarafından yayınlanan raporlarda yer alan koordinat bilgileri kullanılarak nokta bulutu oluşturulmuştur.. Örgü ağı snappyHexMesh ile oluşturulmuştur. Basit bir geometri olmadığı için burada örgü ağı aracındaki özelliklerin çoğu kendisine göre uyarlanmıştır. Düzgün bir ağ oluşturulana dek geometride iyileştirmeler yapılmıştır. Ağ örülüp düzenlenirken geometriyi oluşturma yöntemleri de denenerek en uygun şekilde çizilmiştir. Geometriyi oluşturan üçgenlerin açıklık oranı ağın kalitesini oldukça etkiler. Üçgenler mümkün olduğu kadar eşkenar olmalıdır. Bu sınır koşullarının oluşturulmasını etkileyen bir faktördür.

Akış alanının sınırlarını oluşturan yüzeyler de akış özelliklerini yakalayacak şekilde belirlenip tasarlanmıştır. Giriş ve çıkış yüzeyleri, rotor kanatçığının alt ve üst yüzeyleri ve periyodik yüzeyler oluşturulmuştur. Yüzeylere uygun sınır koşulları tanımlanmıştır. Periyodik sınırlar için cyclic sınır şartı kullanılmıştır. İki periyodik yüzey, bu yüzeylerdeki gridlerin örtüşebilmesi için aynı olmalıdır. Rotorda 36 kanatçık olduğu için periyodik yüzeyler oluşturulurken, ilk yüzey 10° açıyla tekrar edilerek ikinci yüzey elde edilmiştir. Açık kaynaklı bir araçla cyclic sınır şartı oluşturmak dikkat gerektiren bir işlemdir. snappyHexMesh ile aynı geometri için aynı komutlar kullanılarak ağ örüldüğünde grid sayısının her sonuçta farklı olduğu görülür. Bu fark çok az olsa bile gridler tam olarak üst üste gelmediği için cyclic sınır şartının kullanımına engel olur. snappyHexMesh sonucu üretilen gridlerdeki bu düzensiz dizilimden dolayı cyclic sınır şartı tek başına kullanılamaz. Periyodik sınırların örtüşmeyen gridleri ayrı bir sınır şartı kullanılarak eşleştirilir. cyclicAMI isimli bu sınır şartını oluşturmak için createPatch isimli bir OpenFOAM aracı kullanılır. createPatch, periyodik yüzeylerdeki gridlerin merkezini hesaplayarak çalışır. Eğer iki merkez arası mesafe tolerans değerinden büyükse cyclicAMI sınırını oluşturamaz. Burada snappyHexMesh sonuçları ve onu da etkileyen, geometrinin yapısı çok önemlidir. Geometriyi oluşturan üçgenlerin eşkenar olması, açıklık oranlarının düşük olması periyodik yüzeyler için üretilen gridlerin örtüşmemesine neden olur. Ayrıca snappyHexMesh düzlemsel yüzeylerle daha rahat çalışır. Bu nedenle rotor kanatçığının etrafına yerleştirilen sınırlar mümkün olduğu kadar düzgün tasarlanmalıdır. Kavisli yapılar ortaya çıkan örgü ağının yapısal bir halden uzaklaşmasına neden olur. Bu nedenlerle otomatik olarak bir örgü ağı inşa etmek için geometriler kaliteli, periyodik sınırlar geometrik olarak özdeş olmalıdır.

Düzgün yapıda örgü ağı oluşturmak sayısal çözümlerin yürütülebilmesi için çok önemlidir. Örgü ağını oluşturan grid sayısı arttıkça sonuçlar düzelir ancak hesaplama maliyetleri artar. Bunların arasındaki dengeyi sağlayarak uygun sonuçlar elde edilir. snappyHexMesh ağ yapısında bölgesel iyileştirme seçenekleriyle geometriye ve akış alanına özel bir ağ üretme imkanı sunar.

Analiz için rhoSimpleFoam çözücüsü kullanılmıştır. rhoSimpleFoam daimi sıkıştırılabilir bir çözücüdür ve türbülanslı akışlar için kullanılır. Çözüm algoritmasında momentum, enerji ve basınç doğrulama denklemleri bulunur ve SIMPLE algoritmasını takip ederek çalışır. Bu denklemler açık bir sitem oluşturduğu için çözüm yapılamaz. Ek olarak hal denklemi ve türbülans modeli de eklenir. Çözümdeki momentum denklemi anlık olarak, enerji denklemiyse Favre ortalama konseptiyle yazılmıştır. Çözüm basınç tahminiyle başlar. Tahmini basıncın momentum denkleminde kullanılmasıyla hız ve yoğunluk değerleri bulunur ancak bunlar süreklilik denklemini sağlamaz. Daha sonra basınca bir doğrulama değeri eklenilerek kütle akısı için bir eşitlik oluşturulur. Kütle akısındaki hızlar momentum denkleminden, dolayısıyla basınç kullanılarak yazılır. Yoğunluk da basınçla arasında kurulan bir korelasyon sayesinde basınca bağlı olarak yazılır. Kütle akısı eşitliği kullanılarak süreklilik denkleminin sağlanıp sağlanmadığı kontrol edilir. Süreklilik sağlanana dek basınç doğrulama işlemi devam eder. Sağlandığındaysa enerji denklemi çözülerek sıcaklık ve türbülans değerleri bulunur. Sonuçlar yakınsarsa çözüm bitirilir aksi takdirde bu adımlar tekrar edilir. Böylece sonuçlar yakınsayana kadar iterasyonlar sürdürülür.

Bu çalışmada rhoSimpleFoam ile kullanılan türbülans modeli Spalart-Allmaras'tır. Modelde tam olarak viskozite olmayan ancak viskoziteyle arasında bir korelasyon bulunan bir değişken üretilmiştir. Bu değişken için deneysel katsayılar ve fonksiyonlar içeren bir taşıma denklemi oluşturulmuştur. Bu model kullanılarak çözüm algoritmasına bir taşıma denklemi eklenir ve açık olan sistem kapatılarak çözüm elde edilir.

Akış alanı, toplam basınç ve sıcaklık oranları ve Mach sayısı dağılımı bakımından değerlendirilmiştir. Sonuçlar, deneysel ölçümler ve diğer hesaplamalı sonuçlarla mukayese edilmiştir. Sonuçlara göre geometrinin hazırlanma ve örgü ağının inşa usulü hesapların doğruluğunu epeyce etkilemektedir. snappyHexMesh karmaşık geometriler için otomatik bir örgü ağı inşa eden bir araç olmasının yanında, parametrelerinin bilinçli bir şekilde kullanılması gerekir. Elde edilen akış alanı özelliklerine göre toplam sıkıştırma oranı deneysel ve hesaplamalı sonuçlara kıyasla düşük hesaplanmıştır. Ayrıca diğer sonuçlarda görülen eğiş şok, bu çalışmada görülememiştir. Bunun sebebi rhoSimpleFoam'un şokları yakalayamaması olabilir. Ayrıca oluşturulan geometrinin yapısı da örgü ağının kalitesini etkilediği için çözümlere oldukça etki eder. Uygun bir çözücü ve kaliteli bir geometriyle daha iyi sonuçlar elde edilebilir.

1. INTRODUCTION

Computational Fluid Dynamics (CFD) methods are numerical computer simulations executed on fluid flow fields. High rates of convergence and high quality results on reasonably sized grids are expected from numerical methods. Convergence is adjusted and achieved by numerical schemes while state of grids depend on meshing tool. Results of CFD studies executed for various flow fields are influenced by implemented numerical algorithm, initial and boundary conditions and structure of the mesh used.

There are many software to generate grids for common geometries and some of them are also available for complex geometries. OpenFOAM otherwise provides special meshing options to generate grids automatically. It is an open source, widely used software by CFD community as it both executes CFD analyses and constructs mesh. It offers manual usage to construct the mesh, to establish boundary conditions and solver, and to perform sampling for flow characteristics. Selection of discretization scheme and adjustment of tolerance and relaxation are made by user. Also solvers can be compiled for new flows but it is required advanced programing experience. The software has several meshing tools that may construct a structured or unstructured mesh depending upon the geometry.

The present effort that examines OpenFOAM's ability to generate grids automatically, evaluates flow field around a rotor blade. The rotor blade is three dimensional and has a twisted body, making mesh generation a complicated process. It has relatively low aspect ratio but even so, due to its curved shape, especially at leading and trailing edges, makes meshing process difficult. However there exists favorable results of experimental measurements and of computational studies then its geometry and flow characteristics are checked through by various researchers in terms of geometry optimization, development of software for fluid flow and examination of grid dependency of results.

1.1 OpenFOAM

OpenFOAM is an open source software, performing CFD analyses, generating mesh and post-processing results. Since it has no graphical user interface (GUI), it is difficult to use in terms of preparing geometry, constructing mesh, integrating additional software, setting up problem parameters, boundary conditions and solver options, and observing the computation process and post-process results. It has several meshing options for regular and complex geometries. OpenFOAM's utilities that generate mesh are mainly blockMesh and snappyHexMesh. It is also possible to import mesh from other tools.

The meshing utility of OpenFOAM which is utilized for present geometry is snappyHexMesh. This utility is an automatic meshing tool that enables to construct mesh for complex geometries. It generates grids inside, outside or around the geometry with adjustable refinement options. A structured base mesh is prepared in OpenFOAM or received from an external meshing tool then snappyHexMesh builts the mesh over the base by detecting the flow field, removing cells out of the domain, snapping remaining nodes to fit the original geometry, and layering if needed. Thus it can be considered as an automatic mesh generator. It refines mesh, adjusts grids to fit onto geometries, and adds required surface layers.

SnappyHexMesh works with certain geometry files that are composed of triangles. In order to obtain a smooth mesh structure, especially at the edges, triangles of the geometry must be uniform and nearly equilateral. Edge refinement becomes a difficult process for geometries with unfavorable triangularization. It is observed that acute angled edges cannot be snapped easily. Snapping process could not catch the edge of the actual geometry and chips away at corners. This is minimized by adjusting the refinement level and snap controls. However using geometries with equilateral triangles precludes troubles of edge refinement. In order to examine quality of the mesh, checkMesh utility is executed to observe mesh properties that are skewness, orthogonality, aspect ratio and few other aspects of cells. The quality of the mesh is one of the main determining factors through CFD analyses.

1.2 NASA Rotor 37

An experimental research program on axial flow compressors were executed by NASA in order to evaluate the performance characteristics of isolated stages. Four single stages that represent the inlet stage of an eight stage compressor, were designed to evaluate effects of varying rotor aspect and stage pressure ratios. According to the experiments, low aspect ratio stages performed better. Stage 37 was one of these stages and had low rotor aspect ratio and high stage pressure ratio. The stage consists of a rotor that is named as Rotor 37 and of a stator. The flow-path geometry and rotation speed were same for all stages. The design mass flow was 20.2 kg/s and the rotor tip speed was 455 m/s. Blade coordinates were calculated by CDP (compressor design program), a computer program that defines the blades along individual streamlines. CDP calculated the velocity diagrams at several locations on blades. It was aimed to reach the desired velocity diagrams with a minimum total pressure loss across the blade row. A loss model that breaks the total loss into profile and shock losses was used. All blade rows had multiple circular arc (MCA) blade profiles that are defined on an unwrapped conical surface. The rotor efficiency reached a maximum value of 0.876 at a pressure ratio of 2.056 and at mass flow of 20.74 kg/s that was 3 percent higher than the design value. The efficiency was slightly lower than design. The maximum value of efficiency was achieved as 0.93 at 70 percent of design speed.

NASA Rotor 37 is a commonly used transonic compressor test case. It was one of a series of four stages that were tested by NASA within the scope of turbomachinery development activities. The results of all stages were published in (Reid & Moore, 1978), at the end of experimental studies on axial compressors. Overall aerodynamic performance of the rotor was reported in (Moore & Reid, 1980). The rotor was damaged after a blind test case (Denton, 1997) where the results of experiments had been given to researchers just after their CFD analyses. Eleven blind cases were performed without knowing the measurements of the rotor, then predicted results of CFD and postdicted estimations were evaluated. A pressure deficit was calculated in experiments but could not captured in CFD analyses. Rotor 35 whose flow path is identical to Rotor 37, was used at following studies for experiments since it was not possible to repeat the experiments of Rotor 37. The case is still used in a widespread manner by researchers for verification and validation process of aerodynamic design and computational studies.

Many researchers utilize the rotor results to execute various studies that have different scopes. Several studies investigate exclusive flow properties ,e.g., shock/vortex interactions, leakage flow. The shock/vortex interaction in a high speed compressor was investigated in (Suder & Celestina, 1994). According to results the interaction leads to development of a high blockage in the blade passage and it may trigger tip stall. The effect of hub leakage on the performance of high speed compressor was observed both experimentally and computationally in (Shabbir, Celestina, Adamczyk, & Strazisar, 1997). It is stated that even a small leakage can cause a reduction in total pressure rise. Additional to investigation of flow properties, there exists CFD studies aimed for code validation. Two CFD codes were tested and results were in agreement with experiments (Dalbert & Wiss, 1995). The well-known blind test case demonstrated an incompatibility between experiments and computations in (Denton, 1997). Pressure was over predicted by CFD solvers. During a code validation process and a heat transfer solver was examined for turbomachinery simulation in (Ameri, 2010). The rotor is also used to validate design tools. A development study for software of design and analysis was executed in (Kündeş, Aksel, & Baran, 2019).

1.3 Scope of The Study

The objective of the present study is to examine the performance of OpenFOAM's snappyHexMesh utility with a twisted complex blade geometry, Rotor 37 and to decide if it is possible to use it as a functional automatic mesh generation tool. On the other hand the geometry which is analyzed, is widely available and has been used in many computational studies due to its experimental results. It was considered that experiments would usefully supplement and extend the computations. A variety of studies are used to assess the rotor. Many researchers have utilized the rotor to optimize geometry, to validate solution algorithm and to observe grid dependency of computations. Flow field of Rotor 37 is evaluated almost completely by an open source software where mesh generation and numerical simulation are automatic operations.

2. THEORETICAL BACKGROUND

This chapter is devoted to the formulation of the governing equations for compressible flow of the present flow field. Additionally incompressible Navier-Stokes (NS) equations are presented to show transition between averaging concepts from incompressible to compressible flow. Thus Favre averaged NS equations where fluctuations in density are eliminated, are described to introduce the solver that uses them rather than Reynolds-Averaged Navier-Stokes (RANS) equation. The equations which are utilized for numerical simulation are given in the following sections. Momentum and energy equations, and pressure correction equation based on the continuity are executive relations of the solution procedure. The numerical simulation is executed on a rotor of transonic compressor. An equation of state and Spalart-Allmaras turbulence model is used to close the system of equations.

2.1 The Governing Equations for Turbulent Flow

2.1.1 Incompressible flow

NS equation for turbulent flow is written by time averaging to express the variables as the sum of mean and perturbation parts. Time averaging is one of the Reynolds averaging concepts and appropriate for stationary turbulence. It is applied to incompressible form of the instantaneous continuity and momentum equations:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.1}$$

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial t_{ji}}{\partial x_j} \tag{2.2}$$

where t_{ij} is the viscous stress tensor $t_{ij} = 2\mu s_{ij}$, and s_{ij} is the strain-rate tensor:

$$s_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

$$t_{ij} = t_{ji} \qquad s_{ij} = s_{ji}$$

The equations are written in Cartesian tensor form with unit vectors i, j and k. A term with repeated indices such as continuity equation, means summation of this term for all indices. They are rewritten and arranged by Reynolds decomposition process where quantities are written by mean and fluctuating values:

$$u = \bar{u} + u'$$

$$p = \bar{p} + p'$$

$$s = \bar{s} + s'$$

Reynolds averaged values are shown with an overbar and letters with quotes express fluctuations. By inserting those into the equations and averaging them, the equations are rewritten:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \qquad (2.3)$$

$$\rho \frac{\partial \bar{u}_i}{\partial t} + \rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(2\mu \bar{s}_{ji} - \rho \overline{u_j' u_i'}\right) \qquad (2.4)$$

The second equation is generally named as RANS equation. The term $\rho\overline{u_j' u_i'}$ that is distinctive for RANS equation, is the Reynolds-stress tensor, τ_{ij} and its computation is important to determine all mean-flow properties.

2.1.2 Compressible flow

The RANS equation is based on incompressible flow assumption where fluctuations of density and temperature are ignored. For compressible flow these fluctuations have to be taken account. However expression of compressible NS equations by time averaging is a complicated process for correlation between density and velocity and therefore mass averaging, namely Favre averaging concept is used. The instantaneous equations are written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \qquad (2.5)$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial t_{ji}}{\partial x_j} \qquad (2.6)$$

$$\frac{\partial}{\partial t}\left[\rho\left(e+\frac{1}{2}u_iu_i\right)\right]+\frac{\partial}{\partial x_j}\left[\rho u_j\left(h+\frac{1}{2}u_iu_i\right)\right]=\frac{\partial}{\partial x_j}\left(u_it_{ij}\right)-\frac{\partial q_j}{\partial x_j}$$

(2.7)

where q_j is the heat flux vector, e is specific internal energy and h is specific enthalpy that is written as $h = e + p/\rho$.

The viscous stress tensor contains the second viscosity, ζ differently from incompressible flow:

$$t_{ij} = 2\mu s_{ij} + \zeta\frac{\partial u_k}{\partial x_k}\delta_{ij}$$

where δ_{ij} is Kronecker delta. Pressure, density and temperature are related by an equation of state which is specified as perfect gas law, $p = \rho RT$. The fluid is assumed to be calorically perfect then the relation is written for specific internal energy as $e = c_v T$, and for specific enthalpy as $h = c_p T$ due to constant specific heats. The flow properties are expressed by a mean and a fluctuating value:

$$u = \tilde{u} + u''$$

$$e = \tilde{e} + e''$$

$$h = \tilde{h} + h''$$

$$T = \tilde{T} + T''$$

The first term with tilde is mass-averaged while second one is fluctuation. As opposed to time averaging, mean of the fluctuation is not zero for mass averaging. The velocity equation is multiplied by density and averaged by time through Favre averaging where fluctuations of density is eliminated from the equations. It is applied to the governing equations for compressible flow and to the equation of state:

$$\frac{\partial\bar{\rho}}{\partial t}+\frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0$$

(2.8)

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{u}_i)+\frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{u}_i\right)=-\frac{\partial\bar{p}}{\partial x_i}+\frac{\partial}{\partial x_j}\left[\bar{t}_{ji}-\overline{\rho u_j''u_i''}\right]$$

(2.9)

$$\frac{\partial}{\partial t}\left[\bar{\rho}\left(\tilde{e}+\frac{\tilde{u}_i\tilde{u}_i}{2}\right)+\frac{\overline{\rho u_i''u_i''}}{2}\right]+\frac{\partial}{\partial x_j}\left[\bar{\rho}\tilde{u}_j\left(\tilde{h}+\frac{\tilde{u}_i\tilde{u}_i}{2}\right)+\tilde{u}_j\frac{\overline{\rho u_i''u_i''}}{2}\right]$$
$$=\frac{\partial}{\partial x_j}\left[-q_{L_j}-\overline{\rho u_j''h''}+\overline{t_{ji}u_i''}-\overline{\rho u_j''\frac{1}{2}u_i''u_i''}\right]$$
$$+\frac{\partial}{\partial x_j}\left[\tilde{u}_i\left(\bar{t}_{ij}-\overline{\rho u_i''u_j''}\right)\right]$$

(2.10)

$$\bar{p}=\bar{\rho}R\tilde{T}$$

(2.11)

These equations are named as Favre-averaged mean conservation equations, also known as compressible RANS equations. The turbulent kinetic energy per unit volume and the Favre averaged Reynolds stress tensor are expressed by relations below, respectively:

$$\bar{\rho}k=\frac{1}{2}\overline{\rho u_i''u_i''} \qquad \bar{\rho}\tau_{ij}=-\overline{\rho u_i''u_j''}$$

Total energy, E and total enthalpy, H are defined as:

$$E=\tilde{e}+\frac{\tilde{u}_i\tilde{u}_i}{2}+k \qquad H=\tilde{h}+\frac{\tilde{u}_i\tilde{u}_i}{2}+k$$

Since the number of equations are less than the number of unknowns they have to be closed in order to establish a solver for calculation of time averaged density, pressure and mass averaged velocity and total energy (Wilcox, 2006). Hence turbulence parameters, Reynolds stresses and heat transfer relations are added to obtain a closed set of partial differential equations then they are written in closed form as:

$$\frac{\partial\bar{\rho}}{\partial t}+\frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i)=0$$

(2.12)

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{u}_i)+\frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{u}_i+\bar{p}\delta_{ij}-\widetilde{t_{ji}^{tot}}\right)=0$$

(2.13)

$$\frac{\partial}{\partial t}(\bar{\rho}E)+\frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_jE+\tilde{u}_j\bar{p}+\widetilde{q_j^{tot}}-\tilde{u}_i\widetilde{t_{ij}^{tot}}\right)=0$$

(2.14)

These equations are formed by many assumptions. The viscous stress and the heat flux tensors are expressed with laminar and turbulence terms (CFD-Wiki, 2005):

$$\widetilde{t_{ij}^{tot}} \equiv \widetilde{t_{ij}^{lam}} + \widetilde{t_{ij}^{turb}}, \text{ where } \bar{t}_{ji} = \tilde{t}_{ji} + \overline{t_{ji}''}$$

$$\widetilde{t_{ij}^{lam}} \equiv \tilde{t}_{ij} = \mu\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \tilde{u}_i}{\partial x_i}\delta_{ij}\right)$$

$$\widetilde{t_{ij}^{turb}} \equiv -\overline{\rho u_i'' u_j''} \approx \mu_t\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \tilde{u}_i}{\partial x_i}\delta_{ij}\right) - \frac{2}{3}\bar{\rho}k\delta_{ij}$$

$$\widetilde{q_j^{tot}} \equiv \widetilde{q_j^{lam}} + \widetilde{q_j^{turb}}$$

$$\widetilde{q_j^{lam}} \equiv \tilde{q}_j \approx -c_p\frac{\mu}{Pr}\frac{\partial \tilde{T}}{\partial x_j} = -\frac{\gamma}{\gamma-1}\frac{\mu}{Pr}\frac{\partial}{\partial x_j}\left[\frac{\bar{p}}{\bar{\rho}}\right]$$

$$\widetilde{q_j^{turb}} \equiv c_p\overline{\rho u_j'' T} \approx -c_p\frac{\mu_t}{Pr_t}\frac{\partial \tilde{T}}{\partial x_j} = -\frac{\gamma}{\gamma-1}\frac{\mu_t}{Pr_t}\frac{\partial}{\partial x_j}\left[\frac{\bar{p}}{\bar{\rho}}\right]$$

$$\bar{p} = (\gamma-1)\bar{\rho}\left(E - \frac{\tilde{u}_i\tilde{u}_i}{2} - k\right)$$

In order to evaluate turbulent kinetic energy and turbulent viscosity a turbulence model is implemented to the system.

2.1.3 Turbulence modelling

The relations introduced in the previous section still constitute an open system due to insufficient equations. In order to close the system turbulence closure models are implemented corresponding to the turbulence model used. There exist mainly two classes of turbulence models that are eddy viscosity and Reynolds stress models. In the first class, relations for Reynolds stress tensor and turbulent heat flux are similar to the laminar flow relations. Second class models solve transport equations for stresses and generally they are not used for turbomachinery simulations.

Several eddy viscosity models exist and mostly called by numbers of equation in use for closure. Algebraic or zero equation turbulence models do not solve any additional equation and eddy viscosity is given by using the characteristics of boundary layer. One equation models solve one transport equation that may be for the turbulent kinetic energy or viscosity. Also two equation models are widely used for many engineering problems. They solve two transport equations for the turbulent kinetic energy and for another turbulence property corresponding to model.

Spalart-Allmaras model that is used in present study, is one equation model for high Reynolds numbers eddy viscosity, ν_t. The transport equation used in turbulence models is not unique then it is introduced by an empirical approach in this model. The eddy viscosity is expressed with Reynolds stress by $-\overline{u_i u_j} = 2\nu_t S_{ij}$. The model was constructed from the mean flow field and eddy viscosity. Thus variable generated through the model is not an exact viscosity but related with eddy viscosity.

The transport equation for the variable of the model is written as:

$$\frac{\partial \tilde{\nu}}{\partial t} + \frac{\partial}{\partial x_j}(\tilde{\nu} u_j) = c_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\left[\nabla \cdot \left((\nu + \tilde{\nu})\nabla\tilde{\nu}\right) + c_{b2}(\nabla\tilde{\nu})^2\right]$$
$$- \left[c_{\omega 1} f_\omega - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left[\frac{\tilde{\nu}}{d}\right]^2 + f_{t1}\Delta U^2 \tag{2.15}$$

where $\tilde{\nu}$ is introduced as working variable of the turbulence model and depends on eddy viscosity with a function, f_{v1}. It is written for compressible flow with turbulent dynamic viscosity:

$$\mu_t = \rho\tilde{\nu} f_{v1}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \qquad \chi \equiv \frac{\tilde{\nu}}{\nu}$$

The relations of the model have many parameters. f and c terms with different subscripts express empirical functions and empirical constants, respectively. σ is turbulent Prandtl number, κ is Karman constant, and d is the distance from the closest wall. χ is an intermediate value that gives the ratio between the variable of the model and the eddy viscosity. S is the magnitude of vorticity and introduced by relations below:

$$\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$

$$f_\omega = g\left[\frac{1 + c_{\omega 3}^6}{c_{\omega 3}^6}\right]^{\frac{1}{6}}$$

$$g = r + c_{\omega 2}(r^6 - r) \qquad r \equiv \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}$$

The new terms in the last equations, $\tilde{S}$, g and r are intermediate variables. The model provides a transport equation to the system of partial differential equations that are not convenient to solve due to surplus number of unknowns. Hence the system is solved by adding this transport equation (Spalart & Allmaras, 1994).

2.2 Solution Algorithm

The system of the equations for present fluid flow is solved by rhoSimpleFoam that is a steady state compressible solver for turbulent flow. It solves Favre averaged NS equations by the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm that is a guess and correct procedure. The SIMPLE algorithm is a commonly used process to solve the governing conservation equations of continuity and mass by an iterative procedure. It is called Semi-Implicit because momentum and pressure correction equations are solved implicitly while velocity correction is solved explicitly. The algorithm is robust and the instability is reduced by adding relaxation factors for variables. As the solver is pressure based high frequency oscillations or check board effect is avoided in the solution.

A first guess is made for pressure when starting the SIMPLE algorithm then an approximate velocity field is estimated. After that corrections are performed on the pressure to compromise the velocity field with the continuity equation. The determined velocity field from pressure guess do not satisfy generally the continuity equation and produce a mass source that should be removed by corrections. Correction equation for pressure is written by adding a correction quantity on the pressure guess then velocity corrections are calculated by the pressure correction. After that the continuity equation is arranged by substitution of the velocity corrections. New continuity is Poisson equation that is solved for pressure and then velocities are calculated. The pressure correction equation is used to dictate the continuity at each iteration and updates the velocity field. After obtaining correct pressure and velocity field, other unknowns of the system can be calculated (Patankar & Spalding, 1972).

It was explained a procedure for incompressible flow at previous section. For compressible flow density is also computed along with the velocity. rhoSimpleFoam contains the equations of momentum, energy and pressure correction (OpenFOAM Wiki, OpenFOAM Wiki, 2020). Firstly momentum equation is solved to obtain a velocity field and density that do not satisfy continuity equation. Then the energy equation is solved. After that a pressure equation is formed from continuity and momentum equations by adding correction terms to the velocity and density. By this equation pressure field is calculated to obtain divergence free velocity field. The corrections lasts up until the velocity field satisfies both continuity and momentum equations. Turbulence parameters are then solved. This procedure iterates up to convergence. So the equations are solved sequentially.

OpenFOAM uses special utilities for various flow fields. As turbomachinery applications contain rotating parts the solution domain is defined by a moving frame and a Coriolis force is added to the momentum equation. Since rhoSimpleFoam is a steady solver the transient terms vanish then the momentum equation becomes:

$$\frac{\partial}{\partial x_j}\left(\rho u_{Ri} u_j\right) + \rho \varepsilon_{ijk} \Omega_j u_k = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(t_{ji} + t_{ji}^{turb}\right) \tag{2.16}$$

where $\varepsilon_{ijk}$is the permutation symbol, u_R is the relative velocity and Ω is the angular velocity. The energy equation that is solved by rhoSimpleFoam, is written in Favre-averaged form as:

$$\frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\tilde{h} + 0.5\tilde{u}_i\tilde{u}_i\right)\tilde{u}_j + q_{tot}\right] = 0 \tag{2.17}$$

There exists a thermodynamic package, namely thermophysicalProperties where the basic thermophysical properties are specified. Since type of the energy equation is chosen as sensibleEnthalpy in thermophysicalProperties dictionary the total heat flux term in the equation (2.17) is defined as:

$$q_{tot} = -\frac{\mu_t}{Pr_t}\nabla h$$

Thermophysical model in the dictionary is chosen as hePsiThemo to correlate density with pressure via the compressibility, ψ. Firstly temperature is obtained from the equation (2.17) then dynamic viscosity and compressibility are computed.

The momentum equation is expressed in semi discrete form (2.18) in which estimated values of velocity, u_P^* and density, ρ_P^* are calculated by using pressure guess (OpenFOAM Wiki, 2022):

$$a_P^* u_P^* = H[u^*] - \nabla p_P^{n-1} \tag{2.18}$$

$$m_P^* = \rho_P^* u_P^*$$

$$m_p = (\rho_P^* + \rho_P')(u_P^* + u_P')$$
$$= (\rho_P^* + \rho_P')\left(\frac{H[u^*]}{a_P^*} + \frac{H[u']}{a_P^*}\right) - (\rho_P^* + \rho_P')\left(\frac{\nabla p_P^{n-1}}{a_P^*} + \frac{\nabla p_P'}{a_P^*}\right) \tag{2.19}$$

where a_P^* is the matrix coefficient that contains coefficients of terms in the momentum equation that are convection, Coriolis and diffusion terms. $H[u^*]$ expresses contribution of the forces from neighbor cells . ∇p_P^{n-1} is gradient of the pressure guess. Terms with asterisk(*) and with apostrophe(‘) express estimated and correction quantities, respectively. Estimated values of velocity and density do not satisfy the continuity equation then pressure correction approach is executed by adding correction quantity, p_P' to the pressure guess, p_P^{n-1}. The updated pressure value, p_P is used to introduce a mass flux relation to check the continuity equation. The new value of pressure is computed by summation of guess and correction then density correction is calculated by using :

$$p_P = p_P^{n-1} + p_P'$$

$$\nabla p_P = \nabla p_P^{n-1} + \nabla p_P'$$

$$\rho_P' = \psi p_P' = \psi p_P - \psi p_P^{n-1}$$

The third equation is written by compressibility due to correlation between pressure and density. These relations are substituted into the equation (2.19) to check the continuity. If it is satisfied turbulence parameters are updated. It is repeated till convergence.

3. COMPUTATIONAL EFFORT

CFD is a continuously developing aerodynamic tool that is built with numerical algorithms and discretization schemes of governing equations. These equations are the conservation of mass, momentum, and energy. Without any assumption for simplification, e.g., inviscid, incompressible, they are compressible NS equations. The algorithm is generated as result of theoretical. Also the governing equations are arranged by experimental results that give empirical quantities for certain relations. The solution algorithm is established and flow domain is divided into cells by mesh generation. Then the simulation is executed with proper boundary conditions. When solver converges within a specified margin, the simulation ends. After the simulation, solution is compared with certain data that obtained from experiments or other computations. This chapter describes how to solve and manage flow field of Rotor 37, including options to mesh construction and turbulence models, and how to monitor solution progress. Aerodynamic design parameters of the rotor shown in *Table 3.1* was presented in the first reports of the studies (Moore & Reid, 1980).

Table 3.1: Design parameters of Rotor 37.

Parameter	Design Value
Total Pressure Ratio	2.106
Total Temperature Ratio	1.270
Mass Flow, kg/s	20.188
RPM	17188.7
Number of Rotor Blades	36

3.1 Boundary Conditions

Boundary conditions (BC) are considered in two groups, physical and numerical BC. Physical BCs express the surfaces of the geometry, e.g., inlet, outlet and wall. Numerical BCs are specified for variables of the equation and differ by type of the expression. Dirichlet BC specifies a fixed value of the variable at the boundary while von Neumann BC specifies the gradient of the variable normal to the boundary of numerical domain. They are both used at the same boundary for Navier-Stokes equations. In many applications a fixed velocity is prescribed as a Dirichlet BC while zero gradient of pressure is prescribed as von Neumann BC at the inlet. Also at the outlet velocity is defined by a von Neumann BC and pressure is defined by a Dirichlet

BC. By this set up, the pressure equation provides overall continuity. In addition to those, special BCs are defined as symmetry plane BC and cyclic BC that are commonly used in turbomachinery simulations for periodic boundaries (Jasak, 1996).

Surfaces around the blade define the flow domain. Two surfaces are generic boundaries as inlet and outlet. At the hub and tip level two non-planar surfaces are formed, and they are named as top and bottom. Also periodic surfaces are introduced with 10deg space and they are named as AMI1 and AMI2 referring to their exceptional BCs. Type of BCs which are applied to boundaries properly, are given in *Table 3.2*. At the inlet total pressure is 101325 Pa and total temperature is 288.15 K.

Table 3.2: Types of boundary condition.

boundary	pressure	temperature	velocity
inlet	totalPressure	totalTemperature	pressureInletVelocity
outlet	fixedValue	fixedValue	pressureInletOutletVelocity
top	zeroGradient	zeroGradient	noSlip
bottom	zeroGradient	zeroGradient	noSlip
AMI1	cyclicAMI	cyclicAMI	cyclicAMI
AMI2	cyclicAMI	cyclicAMI	cyclicAMI
blade	zeroGradient	zeroGradient	noSlip

3.2 Automatic Mesh Generation

OpenFOAM presents several mesh utilities that could be applied to either straight or complex geometries. snappyHexMesh utility is a manageable meshing tool for complex geometries. It is defined as automatic split hex mesher that refines and snaps to surface. The geometry is in stereolithography (STL) format where surfaces are defined with triangles. With a proper geometry and a clear definition of snapping and refinement, a mesh structure with hexahedra and split hexahedra grids is created by snappyHexMesh.

The utility is controlled using the snappyHexMeshDict file that consists of five main sub dictionaries; geometry, castellatedMeshControls, snapControls, addLayersControl and meshQualityControls. snappyHexMesh works on a base, hexahedra mesh that is constructed by an external meshing tool or blockMesh utility of OpenFOAM. Geometry files with a correct extension are introduced in the geometry section. Common geometries such as box, sphere can be generated in this section. Once the geometries were assigned, castellation dictionary cleans out the redundant grids from the base mesh, and forms a rough mesh for the geometry. Refinement levels are defined in the castellation section for edges, surfaces and regions of the geometry. The mesh is hereby refined in a controlled manner. After the castellation, snapControls dictionary smooths out the castellated mesh in the same way as the original geometry. Then by switching on addLayersControl, layers could be added to boundaries of the surfaces. Results of executive stages are shown in *Figure 3.1*. The whole mesh is controlled by meshQualityControls section in terms of CFD solver requirements, e.g., skewness, orthogonality.

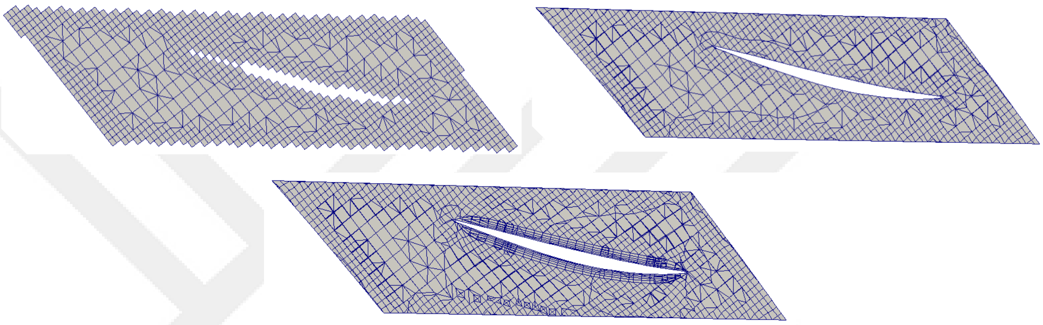

Figure 3.1: Three stages of snappyHexMesh; casteallation (upper left), snapping (upper right) and layering (below).

There are many factors that influence the results of snappyHexMesh. The surfaces in the STL file should interact with at least one edge of a cell of the base mesh. A coarse mesh without any intersection is not recognized by snappyHexMesh. On the other hand if the cell size is smaller than length of the gap between surfaces of the geometry, castellation could not be executed. Unused grids of base mesh are removed at the castellation stage. Since refinement level stated at this stage specifies size of the cells, the gap length should be smaller than the cell size at the edges to remove unused grids. In other words, for geometries consisting of separate surfaces, edges of the surfaces should overlap in order to get a fine mesh. Thus, refinement level of the base mesh and of the castellation stage are important to capture the surfaces. Moreover triangles of the STL file should have low aspect ratio and they should be almost equilateral. Otherwise surfaces could not coincide cells smoothly then it causes to break corners off such as *Figure 3.2*.

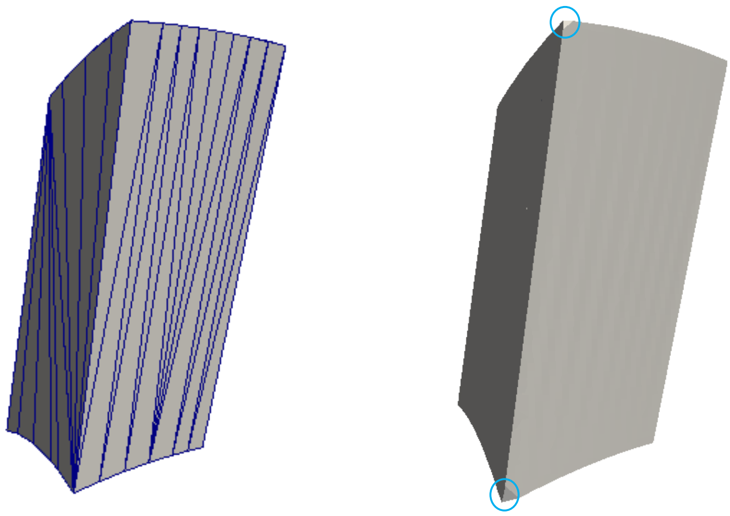

Figure 3.2: Surfaces with high aspect ratio triangles (left) and deformed corners (right).

A surface influences any surface near it corresponding to its individual refinement level if they are too close to allow interaction of cells. This disturbance is prevented by adjustment of distance and a special property, nCellsBetweenLevels. For instance, periodic boundaries of the domain should be placed at a correct distance to protect the mesh from cells of the blade; otherwise deformation on the surface may occur such as *Figure 3.3*.

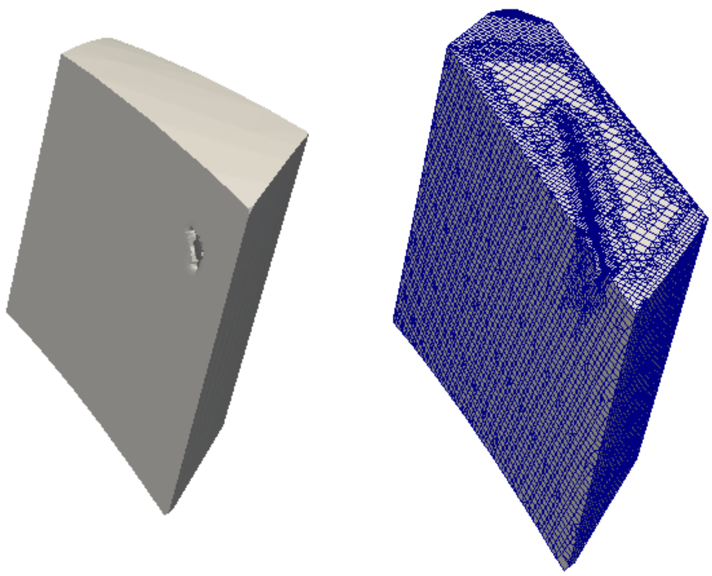

Figure 3.3: Surface deformation due to interaction of close surfaces.

The quality of the geometry is also significant for boundary determination. Type of periodic boundaries is defined as cyclic for constructed mesh in which grids and mesh structure are exactly same. However snappyHexMesh generates an unstructured mesh and vortices of cells are distributed arbitrarily then cyclic BC cannot be applied to. There exists another cyclic BC with arbitrary mesh interface (AMI) property and it is named as cyclicAMI. In order to define it for periodic boundaries of the domain, centers of the cells for two periodics must be near as possible, almost same. This is specified by distribution of cells that depends on the geometry file. In other words, structure of triangles in the STL file influences resulting mesh.

Refinement of a surface with curvature is time consuming and harder than flat surfaces. The curvature of the surface cannot be captured at moderate refinement levels; it is achieved by higher levels, and also produces increased number of cells. In addition to increased refinement level for curved parts of the surface, usage of finer base mesh also help to produce accurate representation of the boundary surface but again increases the cell numbers. In the *Figure 3.4*, refinement level was increased to improve the leading and trailing edge surface representation.

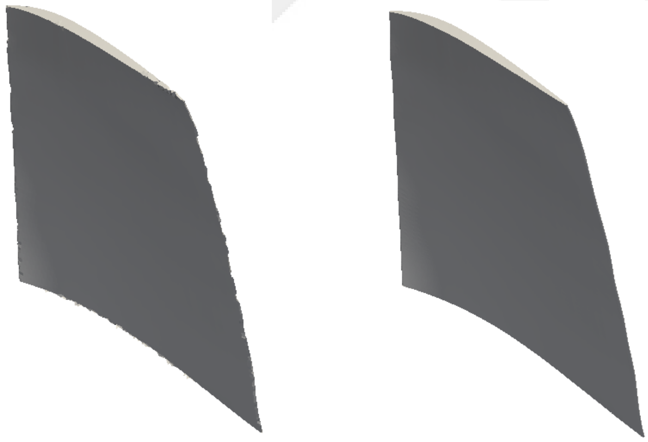

Figure 3.4: Blades with 12077 cells (left), and with 132213 cells (right).

4. RESULTS

4.1 Resulting Mesh

The output of the snappyHexMesh utility highly depends on the quality of the geometry file. It cannot generate clean mesh for CFD analysis for a low quality geometry then flow field cannot be solved. In this study several geometric domains were prepared until the optimum surfaces are determined. snappyHexMesh works with planar, regular surfaces well like other meshers. Thus surfaces were straightened as much as possible in order to obtain an useful computational domain.

Firstly blockMesh utility is executed to obtain the background mesh then snappyHexMesh is started to operate on this background mesh. The flow domain introduced in the geometry section by STL files, was castellated to appear the main lines and refinement levels for edges and surfaces were adjusted. Then it was snapped to obtain original geometry lines by arranging snapControls. Result of castellation and snapping are shown in *Figure 4.1* respectively.

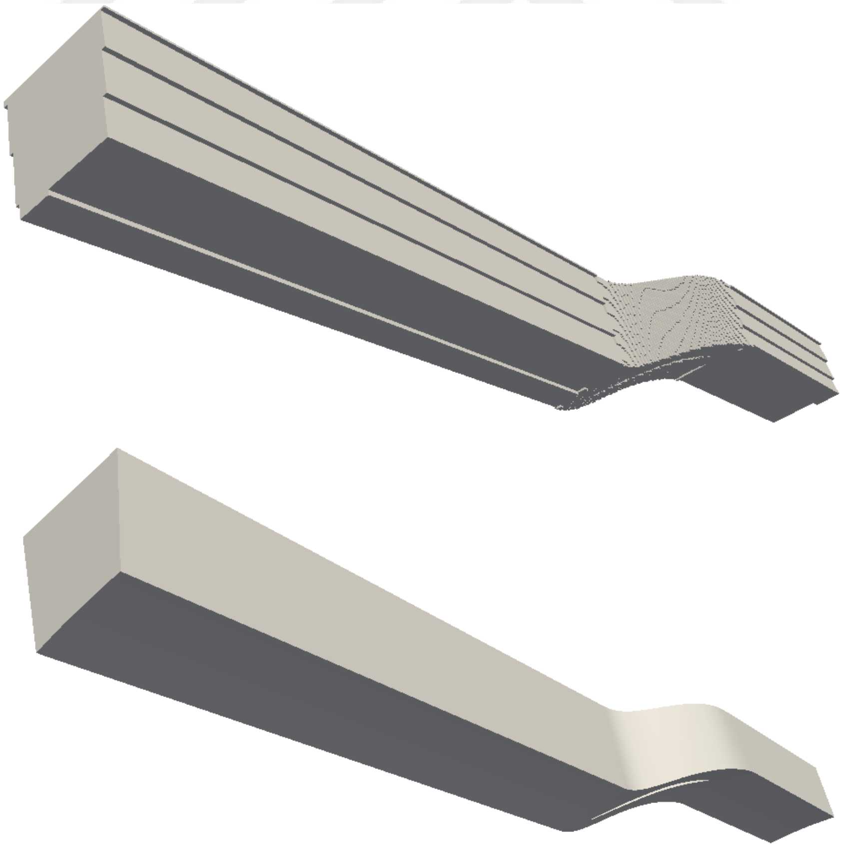

Figure 4.1: Castellation and snapping of the flow domain.

Refinement level were adjusted for separate regions where finer mesh is required due to developing flow properties. Surface of the blade is quite important to observe the flow field in terms of appearance of shock or separation of layer. An unfavorable mesh structure for blade causes the simulation to diverge. Mesh around the blade and at the leading edge are shown by figures *Figure 4.2* that resulted from castellation, snapping and layering processes.

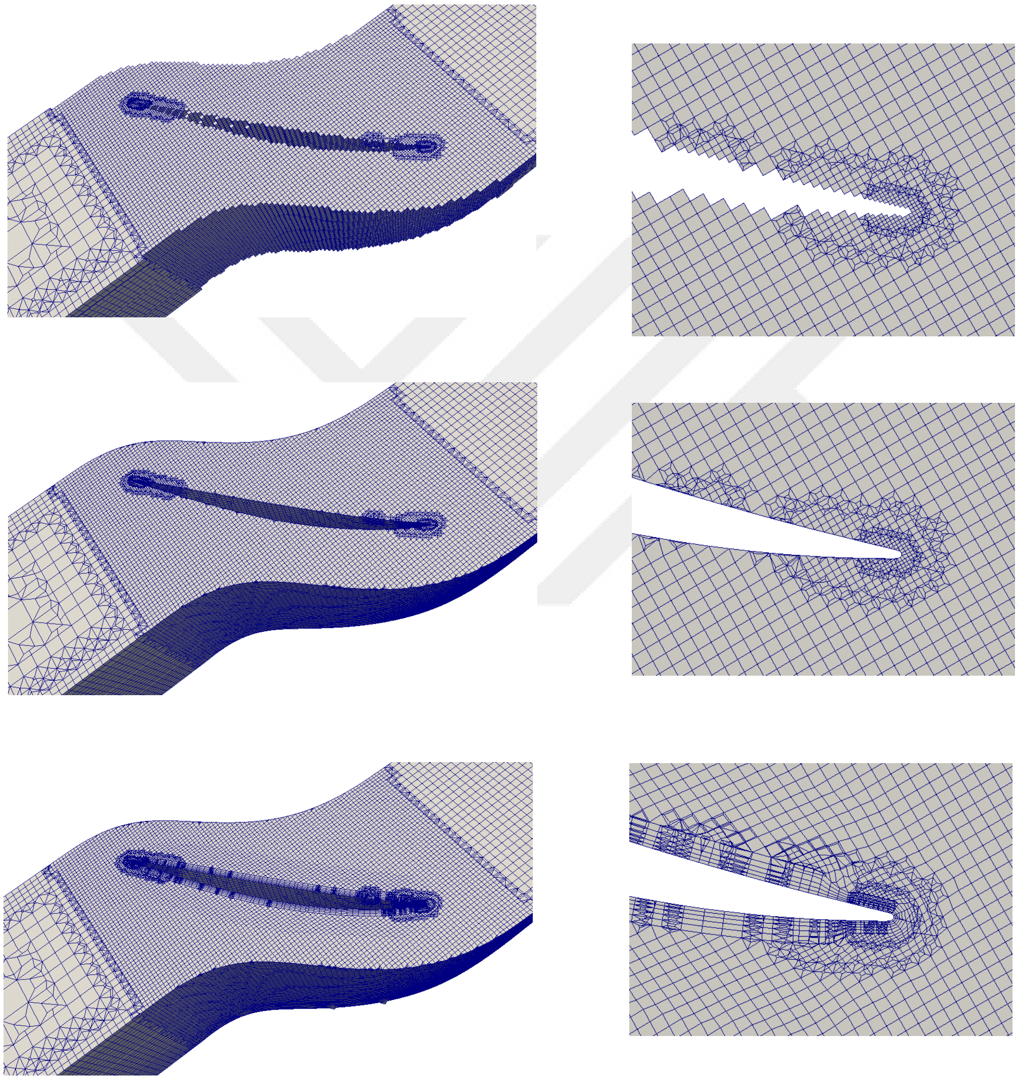

Figure 4.2: Cross section view of the blade and trailing edge.

4.2 Flow Field Characteristic

Rotor 37 was designed for aerodynamic optimization process that comprises to develop blade generating software and numerical methods for turbomachinery applications. In the experiments which were done through first studies the rotor was executed with stator. Overall performance parameters of the experiments were published in (Reid & Moore, 1978). Then the rotor was tested without a stator and measurements were compared with computational results that are blind test cases (Denton, 1997). According to comparison of flow field characteristics, total pressure and temperature ratios were over predicted by computations and it was considered that air leakage at the hub could cause the pressure deficit in the experiments. A study which used fully turbulent calculations, make an agreement with measured pressure ratios and it was determined that results of Rotor 37 depends on turbulence model quite (Chima, 2009).

In the present study, flow field of the rotor were calculated by rhoSimpleFoam that is compressible solver for turbulent flow. Spalart-Allmaras turbulence model was used. Resulting flow field properties were evaluated in terms of Mach number, pressure, temperature and density. According to results, at the blade to blade passage Mach number reaches maximum values then temperature becomes minimum till the separation. Contours of pressure and density tends to be similar. It is an expected results since thermophysical model applied by the solver correlates them via the compressibility that was explained in the section 2.2. Also it is observed that flow field properties were not presented with good resolution and any shock appearance was not observed but a strong shock was appeared within reference studies. This may be caused by the solver which was used during simulation. rhoSimpleFoam may be a solver that cannot capture shock then the results do not show any shock/vortex interaction due to missing the shock. Overall performance parameters are presented figures below. Relative and absolute Mach number, and static pressure contours are shown in *Figure 4.3*. Static and total temperature, and density contours are presented later in *Figure 4.4*.

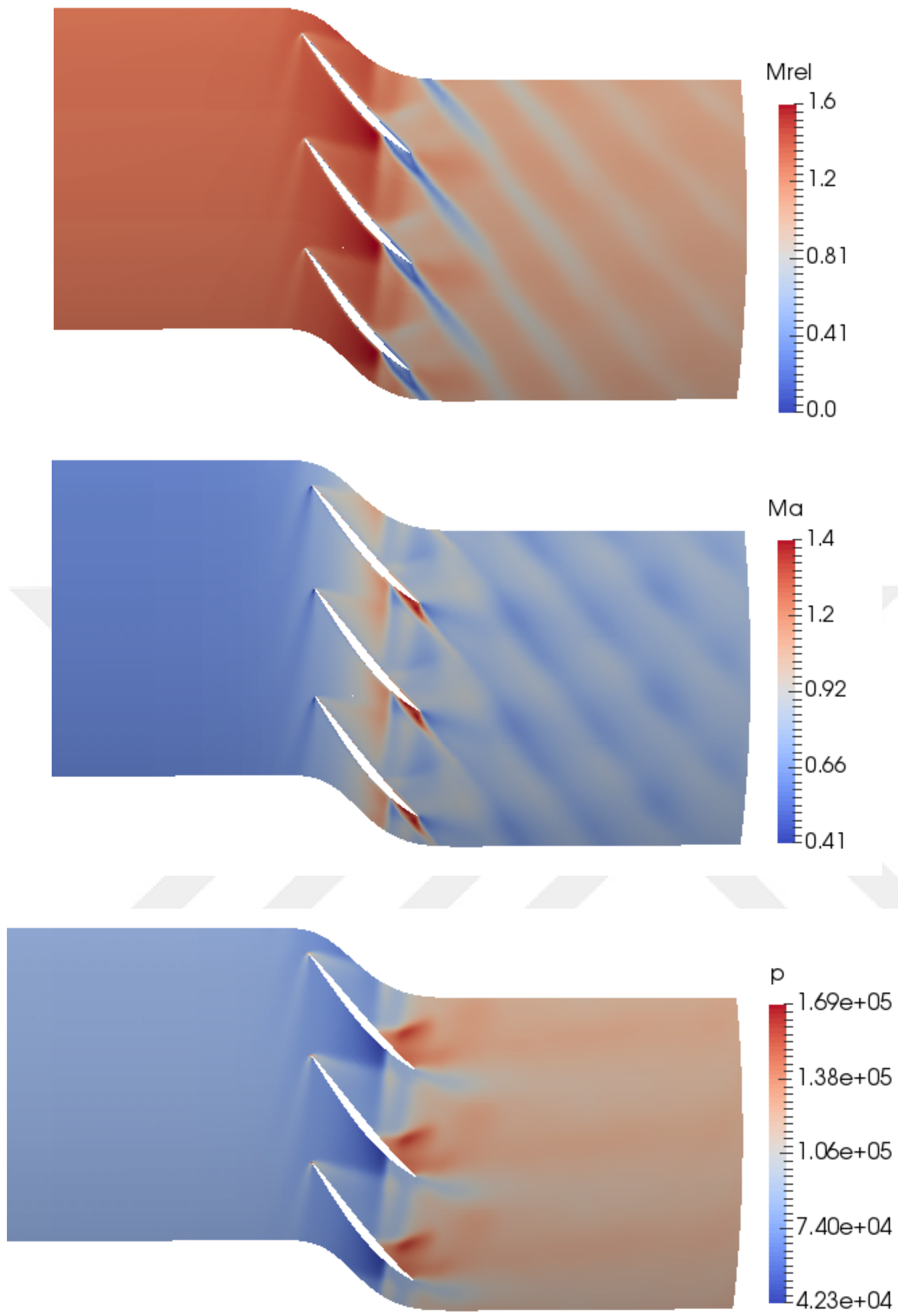

Figure 4.3 : Flow field properties; relative and absolute Mach numbers, and static pressure (Pa).

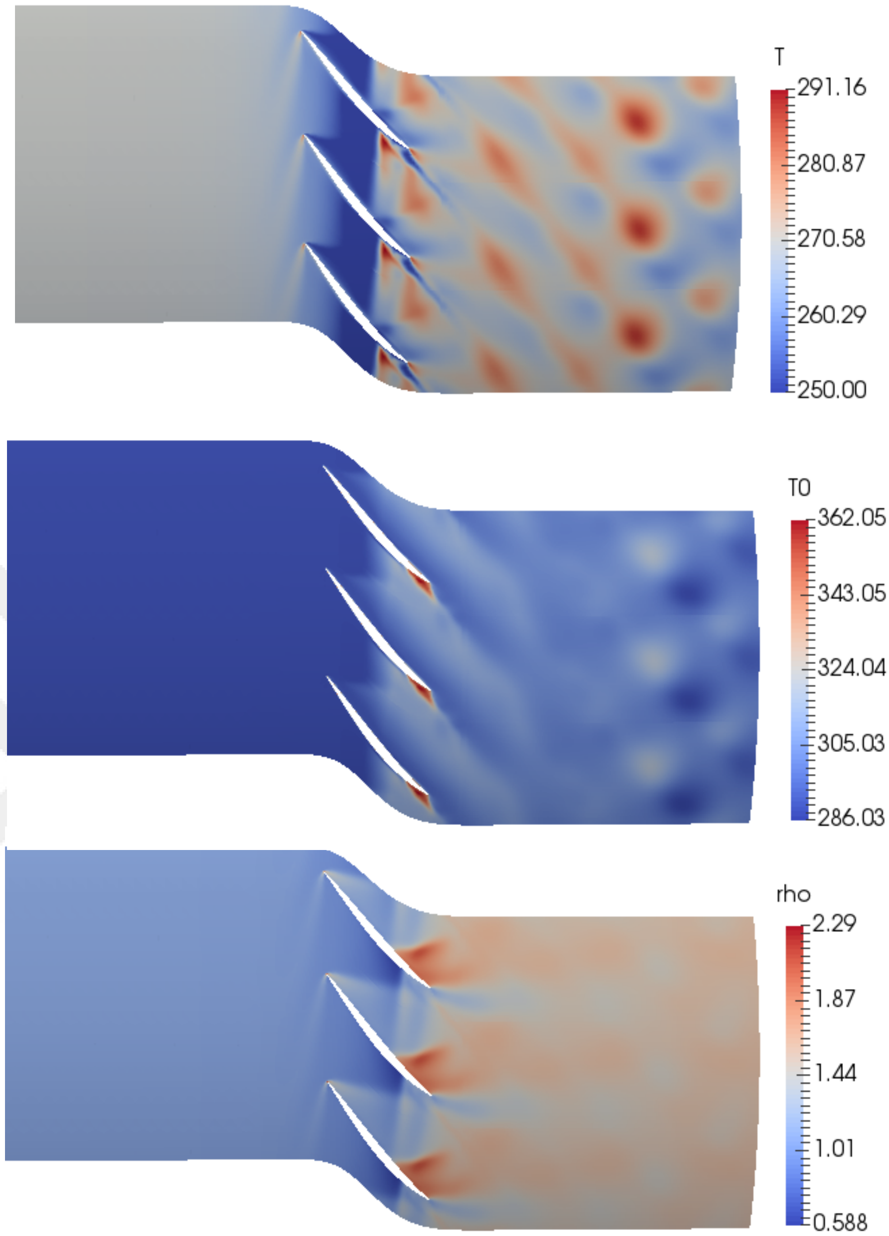

Figure 4.4 : Flow field properties; static and total temperatures (K), and density (kg/m^3).

4.3 Comparison

Resulting flow properties were compared with both experimental and computational studies. Total pressure and temperature ratios shown in *Figure 4.5* and were calculated after the simulation and compared with measurements of (AGARD, 1998). The ratios across the span wise changes similarly till %80 span level and they decreases under %20 span level according to measurements. It is observed that the ratios under predicted and computational values are smaller than measurements. Also pressure values at the tip increased suddenly since the tip leakage was not modelled during then simulation.

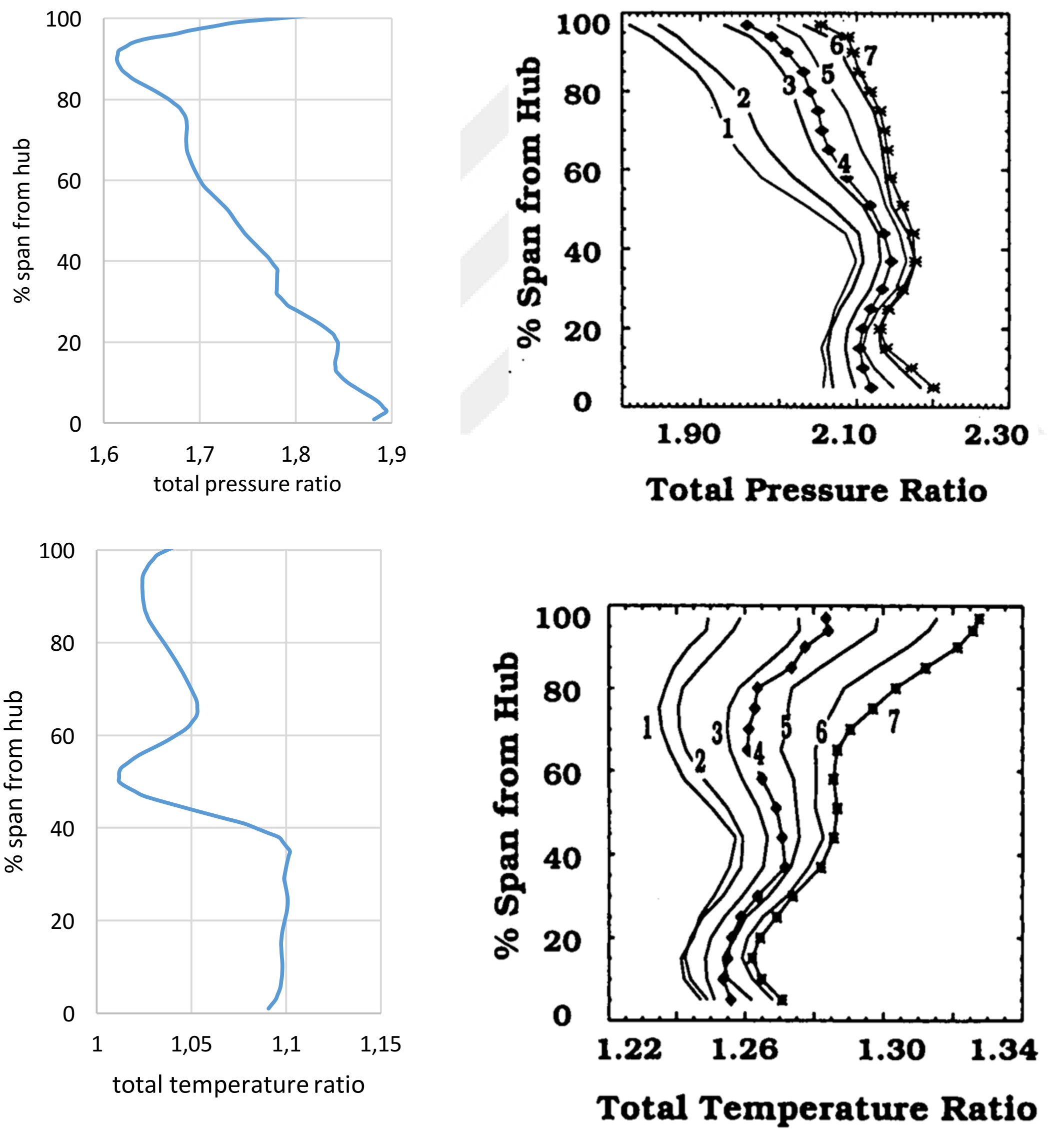

Figure 4.5 : Computed total temperature ratio across the span(left) and experimental measurement.

The results were also compared with other computational solutions for total pressure ratio. Compared computations were executed by two different solvers that was shown in *Figure 4.6*. BTOB3D code solves unsteady RANS equations by finite volume method and Baldwind-Lomax turbulence model is used. This software was developed especially for turbomachinery grading. TASCflow code is a general purpose CFD package and solves RANS while mass and energy equations are solved by time averaging within the solver. Finite volume method is used and k-ε turbulence model is implemented. Pressure at the exit was prescribed for TASCflow simulation and it had influence only at the exit region (Dalbert & Wiss, 1995). Total pressure ratio computed by present study is smaller than compared results across the whole span. It may be again a result of missing the shock due to the solver.

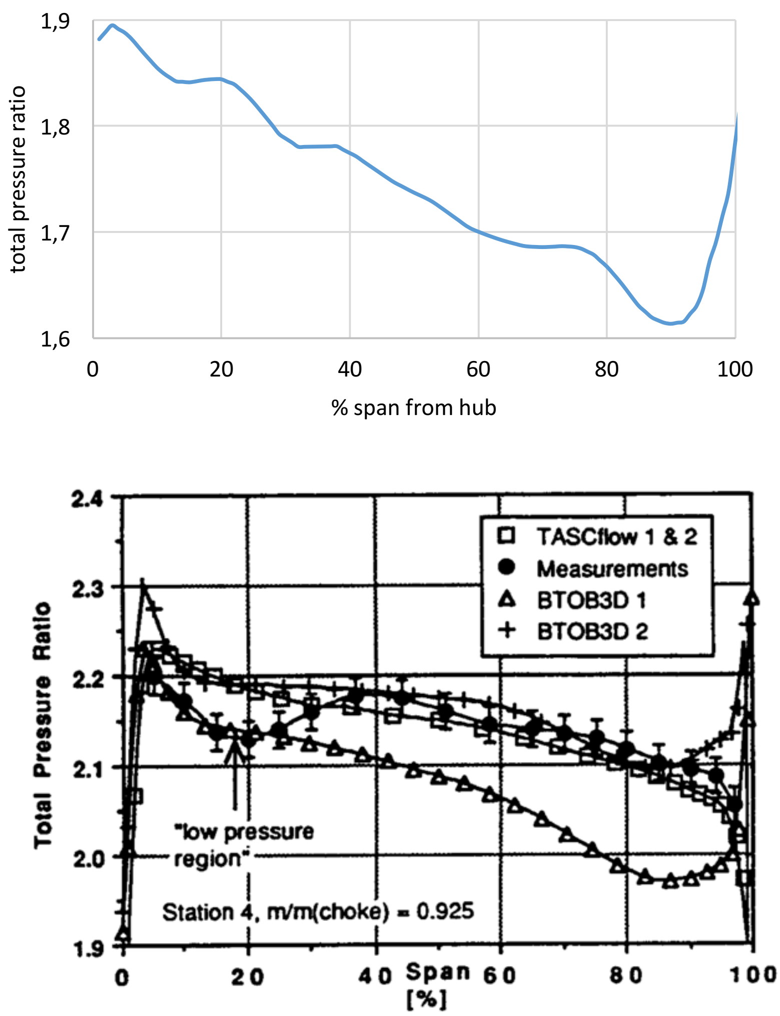

Figure 4.6 : Total pressure ratio obtained by rhoSimpleFoam(above) and by different solvers (below).

Another comparison were made based on a code validation study. The bow shock appears clearly within the computations by NASA Glenn-HT code that uses finite volume method and results are obtained by an k-ω turbulence model (Ameri, 2010). Thus it simulated the bow shock with good resolution as shown in *Figure 4.7*.

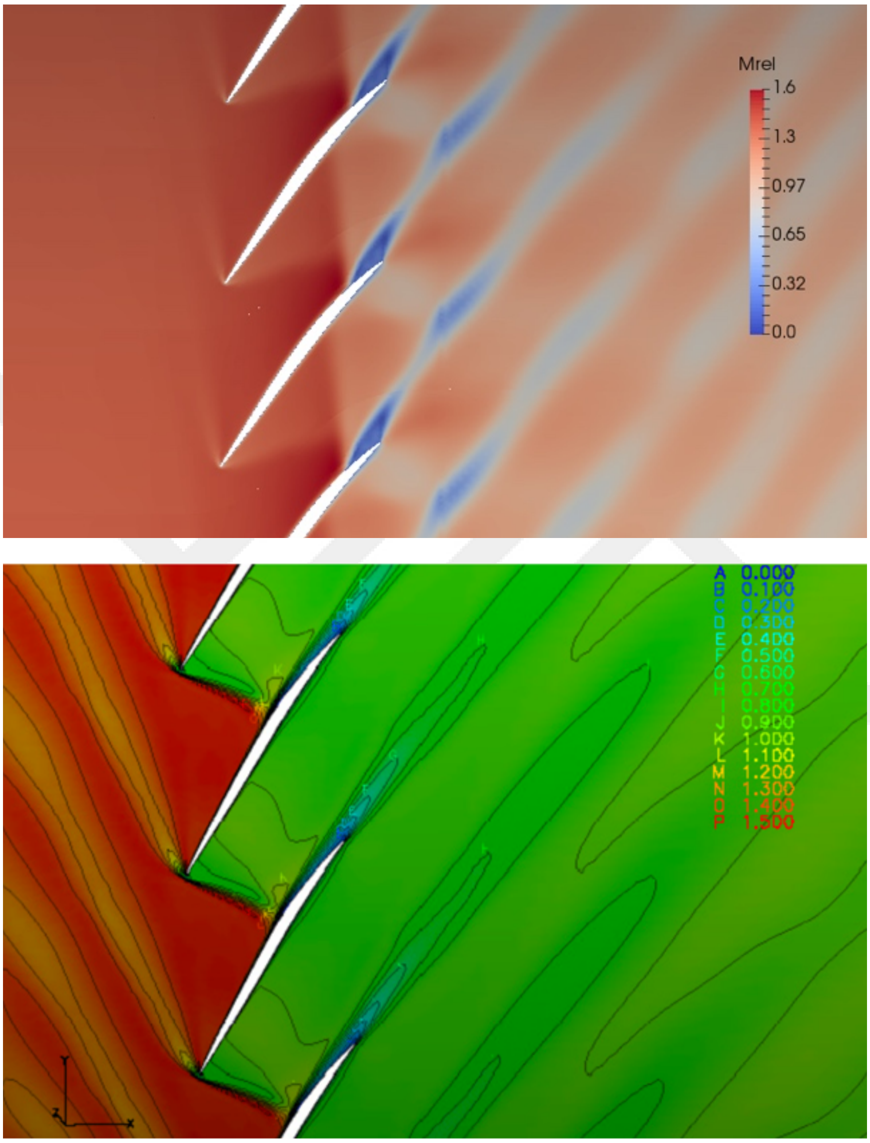

Figure 4.7 : Relative Mach number contours computed at the mid span within the present study (above) and within a code validation study (below).

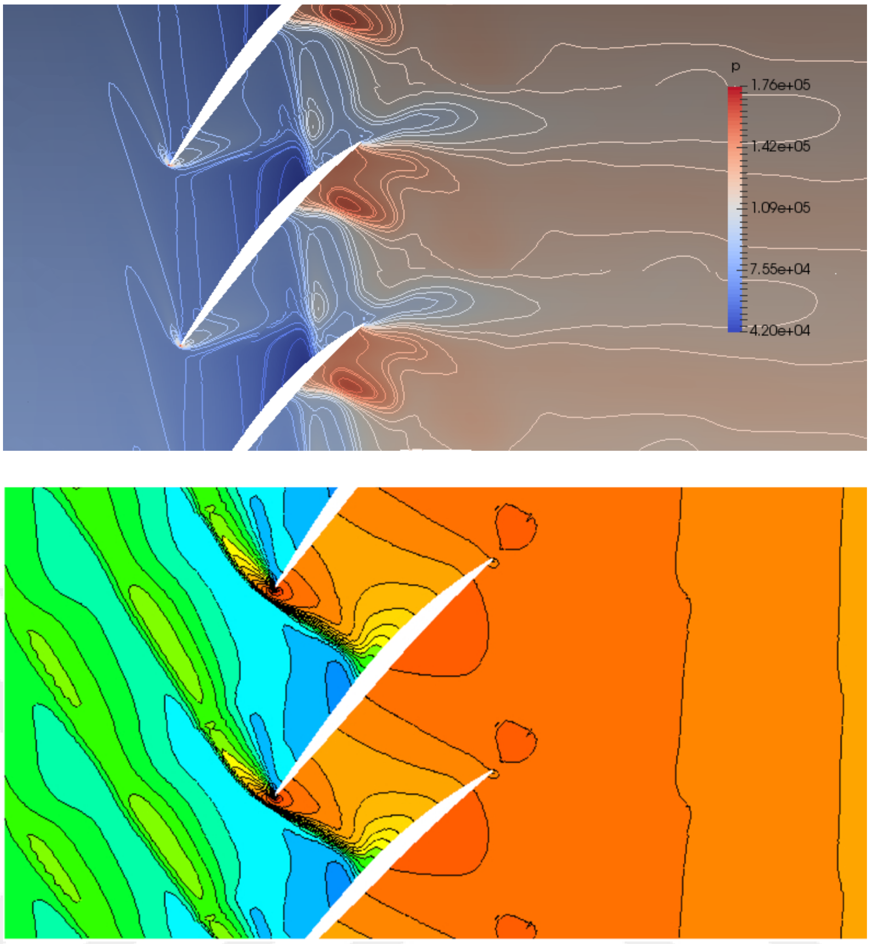

Figure 4.8 : Static pressure contours computed at the mid span within the present study (above) and within a code validation study (below).

Here static pressure contour is compared with another code validation study. According to the results, streamlines is compressed within rise of pressure as shown in *Figure 4.8*. Compared study demostrates a strong shock at the leading edge and there is no separation at the trailing edge unlike the contour of the present study where the pressure is not increasing enough to arise a shock. The results of code validation study was obtained by SWIFT turbomachinery analysis code that uses AUSM$^+$ upwind scheme with an k-ω turbulence model (Chima, 2009).

4.4 Residuals

In order to reach convergence many simulation parameters were examined. The simulation was executed with different shape of boundary surfaces, boundary distances from blade of the rotor, mesh refinement levels, time steps and boundary conditions. Some of them cause divergence of the solution. Result of the optimum setup is given in *Figure 4.9*. The solution was converged after 8000th iteration.

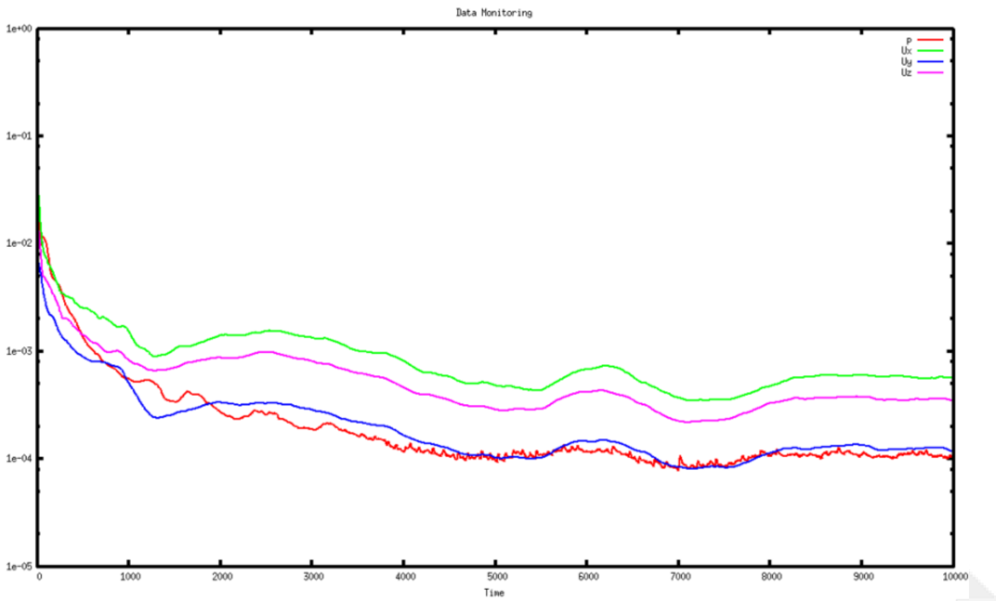

Figure 4.9 : Residuals for pressure and velocities.

5. CONCLUSION

OpenFOAM is used for various engineering applications due to its available solvers that offer different simulation types in terms of flow physics or geometry. In addition to different solvers, meshing utilities of the software system present rich options and different approaches to generate grids. The utility snappyHexMesh is used to automatically generate a suitable mesh.

On the other hand Rotor 37 is a widely used geometry to test and asses numerical solution schemes and approaches. There exists experimental measurements and many computational studies of the rotor, making it possible to compare and to evaluate results in different aspects.

The aim of this study was the assessment of the automatic mesh generation tool snappyHexMesh for problems involving turbomachinery. Rotor 37 is used as a test case to make this assessment.

It is observed that, despite being an automatic mesh generation tool, snappyHexMesh requires, careful construction of geometry, detailed consideration and educated selection of control parameters to generate meshes suitable for the type of analysis considered in this study. High fidelity geometrical representation in snapphHexMesh produced grids is proven to be possible only with proper STL file triangularization and proper selection of refinement levels for the surfaces and features.

Nevertheless, it is shown that, it is possible to automatically generate cartesian meshes for cyclic periodic analysis of flows in turbomachinery cascades using the snapphHexMesh utility of the OpenFOAM software system.

Results of analysis, performed using automatically generated mesh is presented along with experimental and computational results from the literature for comparison. While there is agreement in general structure of the flow with the literature, quantitative comparison reveals important differences. These differences are attributed to two distinct sources. These are proper definition of geometry both as STL surface and as the resulting mesh, and the solver used to analyze the challenging problems of correct shock capturing and shock-boundary layer interaction.

It is concluded that, while the automatic mesh generation tool snappyHexMesh is suitable for turbomachinery cascade flow analysis, currently used solver, rhoSimpleFoam, should be replaced with solvers of higher order of accuracy for quantitative agreement with reference solutions.

While concluding this study within the scope of the thesis, it is the aim of the author to extend this study to find a standard solver or develop a new solver which can be used in conjunction with snappyHexMesh generated meshes for transonic turbomachinery cascade flows.

REFERENCES

AGARD. (1998). *CFD Validation for Propulsion System Components.* Neuilly-sur-Seine: AGARD-AR-355.

Ameri, A. (2010). *NASA Rotor 37 CFD Code Validation.* Ohio: NASA.

CFD-Wiki. (2005, 11 24). *CFD Online*. Wiki: https://www.cfd-online.com/Wiki/Favre_averaged_Navier-Stokes_equations adresinden alındı

Chima, R. V. (2009). *SWIFT Code Assessment for Two Similar Transonic Compressors.* Ohio: NASA.

Dalbert, P., & Wiss, D. H. (1995). *Numerical Transonic Flow Field Predictions for NASA Compressor Rotor 37.* New York: ASME.

Denton, J. (1997). Lessons from Rotor 37. *Journal of Thermal Science, 6*(1), 1-13.

Jasak, H. (1996). *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows.* (PhD Thesis). Imperial College of Science, Technology and Medicine, London.

Kündeş, N., M.H., A., & and Baran, Ö. (2019). Development of an Automatic Design and Analysis Tool for Axial Flow Compressors. *Journal of Thermal Science and Technology*, 179-190.

Moore, R. D., & Reid, L. (1980). *Performance of Single-Stage Axial-Flow Transonic Compressor with Rotor and Stator Aspect Ratios of 1.19 and 1.26, respectively, and with Design Pressure Ratio of 2.05.* Ohio: NASA-TP-1659.

OpenFOAM Wiki, U. (2020, 12 12). *OpenFOAM Wiki*. RhoSimpleFoam: https://openfoamwiki.net/index.php/RhoSimpleFoam#cite_note-4 adresinden alındı

OpenFOAM Wiki, U. (2022, 9 16). https://openfoamwiki.net/index.php/SimpleFoam adresinden alındı

Patankar, S., & Spalding, D. (1972). A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-dimensional Parabolic Flows. *International Journal of Heat Mass Transfer, 15*, 1787-1806.

Reid, L., & Moore, R. (1978). *Design and Overall Performance of Four Highly Loaded, High-Speed Inlet Stages for an Advanced High-Pressure-Ratio Core Compressor.* Ohio: NASA-TP-1337.

Shabbir, A., Celestina, M., Adamczyk, J., & Strazisar, A. (1997). *The Effect of Hub Leakage Flow on Two High Speed Axial Flow Compressor Rotors.* New York: ASME Paper 97-GT-346.

Spalart, P., & Allmaras, S. (1994). A One-Equation Turbulence Model for Aerodynamic Flows. *La Recherche Aerospatiale, n1*, 5-21.

Suder, K., & Celestina, M. (1994). *Experimental and Computational Investigation of the Tip Clearance Flow in a Transonic Axial Compressor Rotor.* Ohio: NASA TM-106711.

Wilcox, D. C. (2006). *Turbulence Modeling for CFD* (3rd Edition b.). La Canada, California, US: DCW Industries.

CURRICULUM VITAE

Name Surname : **Büşra Çetin**

EDUCATION :

- **B.Sc.** : 2019, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Astronautical Engineering
- **M.Sc.** : 2022, Istanbul Technical University, Graduate School, Defense Technologies

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2018-2019 Turkish Aerospace
- 2020-2021 TÜMOSAN