AN AUTOMATIC GEOMETRY AND MESH GENERATION TOOL FOR HELICOPTER ROTOR AERODYNAMIC DESIGN AND ANALYSIS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HALIT ELDEM UZUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2021

Approval of the thesis:

AN AUTOMATIC GEOMETRY AND MESH GENERATION TOOL FOR HELICOPTER ROTOR AERODYNAMIC DESIGN AND ANALYSIS

submitted by **HALIT ELDEM UZUN** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. M. A. Sahir Arıkan
Head of Department, **Mechanical Engineering** _____

Prof. Dr. Mehmet Haluk Aksel
Supervisor, **Mechanical Engineering, METU** _____

Assist. Prof. Dr. Özgür Uğraş Baran
Co-supervisor, **Mechanical Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Cüneyt Sert
Mechanical Engineering, METU _____

Prof. Dr. Mehmet Haluk Aksel
Mechanical Engineering, METU _____

Dr. Ali Karakuş
Mechanical Engineering, METU _____

Prof. Dr. Sinan Eyi
Aerospace Engineering, METU _____

Dr. Onur Baş
Mechanical Engineering, TEDU _____

Date: 08.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Halit Eldem Uzun

Signature :

ABSTRACT

AN AUTOMATIC GEOMETRY AND MESH GENERATION TOOL FOR HELICOPTER ROTOR AERODYNAMIC DESIGN AND ANALYSIS

Uzun, Halit Eldem
M.S., Department of Mechanical Engineering
Supervisor: Prof. Dr. Mehmet Haluk Aksel
Co-Supervisor: Assist. Prof. Dr. Özgür Uğraş Baran

September 2021, 86 pages

Helicopter rotors in various flight regimes constitute a fairly complex wing geometry and exhibit motion affected by the rotor wake with strong tip vortices. As a result, rotor motion creates highly three-dimensional flow patterns, and unlike fixed wings, flow around each rotor blade interacts with each other. Due to these complexities, the rotor flow analysis can be very challenging for CFD solvers. The challenge starts with very high-quality requirements on the computational mesh around the rotor geometry. A modern helicopter rotor geometry involves several different airfoil profile sections, profile blending, twists, and other parameters. Moreover, blade tip geometry can be selected from a wide variety of candidates. This thesis presents the development and results of fully automatized rotor geometry and solution domain generation/decomposition tool. A rotor with a very high complexity with any number of blades can be generated with the new tool. The rotor is assumed to be articulated for each blade, and the rotor blades are considered rigid. Very high-quality multi-block and conformal meshes in the flow domain can be generated automatically. The new tool is tested with known helicopter rotors in the literature, and CFD solutions are obtained.

Keywords: parametric geometry, automatic mesh generation, CFD

ÖZ

HELİKOPTER ROTOR TASARIM VE ANALİZİ İÇİN OTOMATİK BİR GEOMETRİ VE AĞ OLUŞTURMA ARACI

Uzun, Halit Eldem
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mehmet Haluk Aksel
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Özgür Uğraş Baran

Eylül 2021 , 86 sayfa

Helikopter rotorları çeşitli uçuş rejimlerinde oldukça karmaşık bir kanat geometrisine sahip olup güçlü pal ucu girdaplarının etkisi altında hareket eder. Sonuç olarak, rotor hareketi karmaşık üç boyutlu akış alanı oluşturur ve sabit kanatların aksine, her bir rotor kanadının etrafındaki akış birbiriyle etkileşime girer. Bu karmaşıklıklar nedeniyle rotor akış analizi, HAD (Hesaplamalı Akışkanlar Dinamiği) çözücüleri için zorlayıcı olabilir ve rotor geometrisi etrafında çok yüksek kaliteli ağ üretimini gerektirir. Modern bir helikopter rotor geometrisi, birkaç farklı kanat profili kesiti, aerodinamik ve geometrik burulma ve diğer parametreleri içerir. Ayrıca, farklı çeşitlerde kanat ucu şekilleri ile modellenebilir. Bu tez, tam otomatikleştirilmiş rotor geometrisi ve çözüm alanı oluşturma aracının geliştirilmesini ve sonuçlarını sunar. Bu yeni araç ile, farklı kanat sayılarıyla beraber yüksek karmaşıklığa sahip bir rotor geometrisi modellenebilir. Yüksek kaliteli çok bloklu konformal ağlar otomatik olarak oluşturulabilir. Yeni araç literatürde bilinen helikopter rotorları ile test edilerek HAD çözümleri elde edilmiştir.

Anahtar Kelimeler: parametrik geometri, otomatik çözüm ağı, HAD

To my mother and lovely girlfriend

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my supervisior, Prof. Dr. M. Haluk Aksel for his guidance and support.

I would like to present my gratitude to my co-supervisor Assist. Prof. Dr. Özgür Uğraş Baran for his support, guidance and endless motivation.

I want to present my sincerest thanks to my mother and my girlfriend for their encouragement, patience, support and unconditional love throughout this study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATONS

2D	Two Dimensional
3D	Three Dimensional
API	Application Programming Interface
BVI	Blade-Vortex Interaction
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
FM	Figure of Merit
MDAO	Multidisciplinary Analysis and Optimization
NURBS	Non-uniform Rational B-spline
RANS	Reynolds Average Navier Stokes

LIST OF SYMBOLS

SYMBOLS

Ω	Angular velocity
C_p	Pressure coefficient
C_T	Blade loading coefficient
C_Q	Torque coefficient
θ_c	Collective pitch angle
C^0	Line continuity
C^1	Line and slope continuity
ρ	Density
M	Mach number
N_b	Number of blades
N_l	Number of nodes at the leading edge
N_p, N_s	Number of nodes on pressure and suction curves
$N_{t,1}, N_{t,3}$	Number of nodes near the trailing edge
$N_{t,2}$	Number of nodes at the trailing edge
N_p	Number of panels
c_i	Chord distribution for i-th panel
$N_{g,i}$	Cross-section number for i-th panel
Γ_i	Dihedral angle distribution for i-th panel
t/c	Maximum thickness (%) distribution for i-th panel
b_i	Span for i-th panel
$\Lambda_{LE,i}$	Sweep angle distribution for i-th panel
ϵ_i	Twist angle distribution for i-th panel
θ	Pitching axis
β_0	Precone angle
r	Radial distance
P	Rotor power

R	Rotor radius
σ	Rotor solidity
T	Rotor thrust
$S_{l,j}$	Spacing inputs at the leading edge
$S_{p,j}, S_{s,j}$	Spacing inputs on pressure and suction curves
M_{tip}	Tip Mach number

CHAPTER 1

INTRODUCTION

The helicopter is an aircraft that utilizes rotating wings to supply lift, propulsion, and control forces. Blades of the main rotor, referred to as rotating wings, spin around a mostly-vertical axis, defining a disk in a nearly horizontal plane. The relative motion of a wing surface with respect to the surrounding air generates aerodynamic forces. Compared to the majority of the fixed-wing aircraft, which need a forward velocity to maintain flight, the helicopter's rotary wings can produce lift force even though the vehicle's velocity is zero. Hence, the helicopter has the capability of vertical flight, stationary hover with respect to ground, vertical take-off, and landing on nearly any terrain. The helicopter can perform a translational flight from hover to forward flight. Therefore, a method of providing propulsion by the rotor to enable forward flight while maintaining the lift is required. The thrust vector is tilted forward to obtain this propulsive force from the rotor. By employing a similar rotor disk mechanism, the helicopter can move backward and sideways. The rotor also generates the forces and moments that regulate the aircraft's attitude, position, and velocity. Slightly tilting the orientation of the rotor disk to fore and aft provides pitch control, whereas roll control is achieved by tilting it to left and right [1]. Notice that, this tilting operation alters the disk plane. Moreover, the collective pitch control may change the pitch angle of all the main rotor blades. As a result, the total lift produced by the rotor is adjusted.

Approximately thirty years after fixed-wing airplanes had flown, a rotating wing aircraft that could take-off, descend, hover vertically and carry out simple maneuvers were realized for the first time in the mid-1930s. Then, helicopters have been matured progressively as engine technology advances. In particular, the development of reciprocating and the turboshaft engines with high power-to-weight ratios lead

to a large variety of helicopter designs. The engine technology is accompanied by lightweight materials such as aluminum and lighter high-strength alloys of steel, that leads to high-performance helicopters. Also, the increasing demand for the helicopter in both civilian and military roles ultimately led to the beginning of mass production. Today, helicopters are essential in modern life; their roles involve search and rescue, firefighting, aerial observation, police surveillance, air ambulance, tourism, media, troop transport, and other uses through its unique flying features [2].

1.1 Aerodynamic Issues in Rotorcraft Design

The distinctive flying characteristics of helicopters are not achieved without costs of complex aerodynamic problems, high levels of noise, mechanical vibrations, and relatively higher power requirement than that of fixed-wing aircraft of the same weight [3]. Only aerodynamic flows around the rotors will be addresses within the scope of this thesis. An overview of some aerodynamic problems is given in the Fig. 1.1.

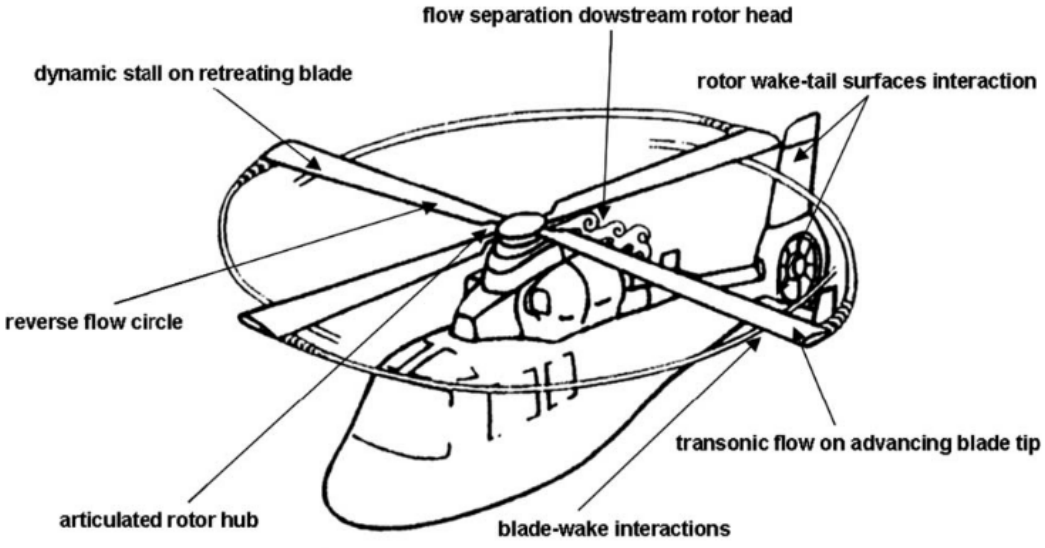


Figure 1.1: Overview of Aerodynamic Challenges in Rotorcraft [4].

The aerodynamics of modern helicopters are challenging. The flow field generated by the rotating blades is exceptionally complex and difficult to model and analyze. The accurate estimation of the rotor wake is one of the most critical flow features required to predict the rotorcraft performance. In general, the rotor wake consists of a vortex

sheet and a powerful helical tip vortex (Fig. 1.2). Strong vortices trailing from the tips of each blade dominate the rotor wake. The interaction of these strong shed vortices with another blade is known as the Blade-Vortex Interaction (BVI). These vortices remain close to rotor blades for several rotor revolutions and induce strong three-dimensional velocity field. As rotor blades encounter this induced velocity field, the effective angle of attack of the airfoil is changed, and unsteady fluctuating airloads are produced on the blade [4, 5]. Furthermore, helicopter rotor blade sections encounter a wide variety of velocities from root to tip, ranging from subsonic to transonic. This variation may have an impact on the stability and accuracy of the numerical flowfield simulations [6]. There are also other aerodynamic issues. One of them is the dynamic stall phenomena, which is particularly significant in the forward flight regimes [7]. The local velocity and dynamic pressure on the retreating blade are relatively low, and thus the blade is required to have a higher angle of attack to sustain lift. If the angle of attack become too large, then the dynamic stall may occur. Lastly, the flow may be transonic at the advancing blade tip which may stimulate shock-induced flow separation in high-speed forward flight speeds.

1.2 A Brief Review of Parametric Aircraft Sizing Tools

Aircraft manufacturers seek to enhance the efficiency and efficacy of their development processes. Utilizing comprehensive Computational Fluid Dynamics (CFD) tools and implementing multidisciplinary analysis and optimization (MDAO) frameworks have recently been developed to reduce the time and cost of design iterations. Incorporating automated geometry and grid generation capability into these tools may further accelerate the computations. Computer-Aided Design (CAD) tools that provide parametric geometries integrate into these aerodynamic and multidisciplinary analysis workflows to support automation and reduce the repetitive human effort. Parametric geometry generators allow practical trade-off studies or aerodynamic shape optimization processes. There are numerous parametric modeling studies for aircraft in the literature. Some noteworthy studies will be discussed in this section.

Some significant obstacles in using high-fidelity CFD solvers with CAD modeling tools are longer setup times and problems altering the geometry for trade-offs through-

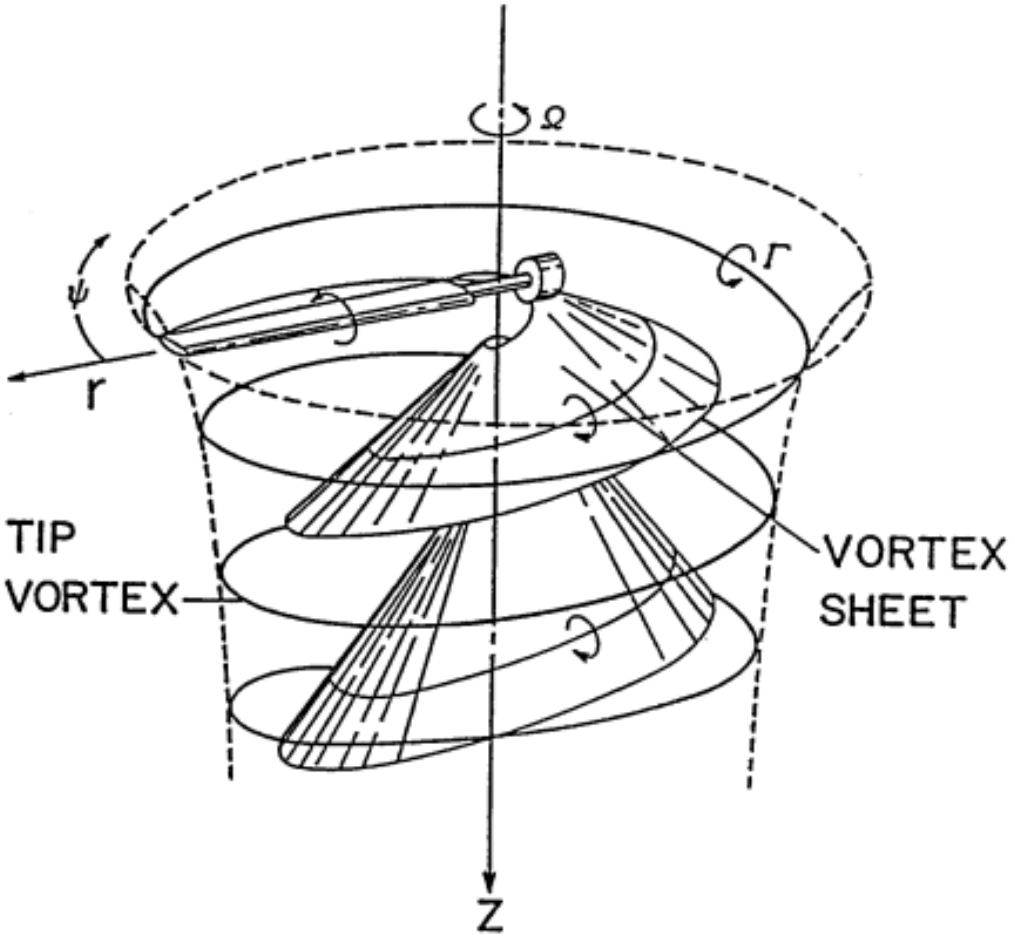


Figure 1.2: Sketch of a single blade rotor wake [5].

out the design processes, as illustrated in Fig. 1.3.Vandenbrande et al. [8] use the Boeing General Geometry Generator (GGG) as a parametric geometry modeler to automatize their design space exploration model. The GGG includes a particular spline method [9] to produce a continuous function of design parameters and support shape control to eliminate undesired bumps. This geometry generator can also perform advanced spline calculus while it is not readily applicable or practical in commercial CAD programs. Surfaces are built through the curves defined by two or more dependent variables between the predetermined section splines.

Vehicle Sketch Pad (VSP) is perhaps the most popular, publicly accessible parametric geometry modeling program with a GUI. This tool is developed by NASA to produce three-dimensional, water-tight aircraft geometries for the conceptual design phase [10]. VSP is built upon the experience with the Rapid Airplane Modeler (RAM [11])

tool. VSP extends geometry components of RAM along with parametrization capability, and provides more export formats. Compared to traditional CAD programs, the learning curve is significantly short for through a set of simple parameter forms and a real-time display window. While VSP is mainly designed to be used in interactive mode, it may also be programmed by a series of macros. This macro capability allows VSP to interface with external analysis and optimization tools without manual interruption. VSP also has a vast collection of predesigned aircraft geometries, mostly submitted by the global user community.

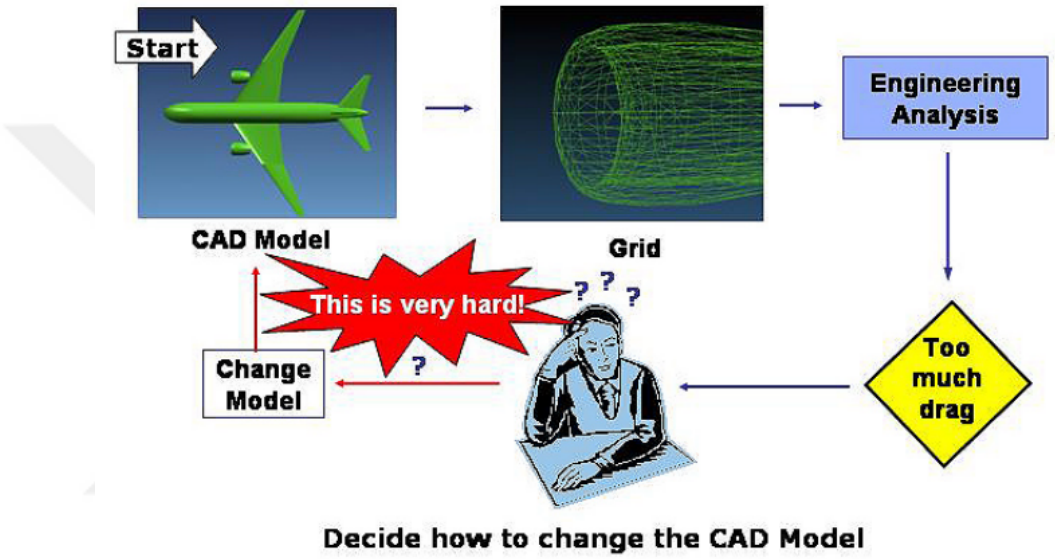


Figure 1.3: Difficulties in using CAD and CFD Tools [8].

Rodriguez and Sturdza developed a preliminary design tool in the Java programming language known as Rapid Geometry Engine (RAGE), which utilizes parametric geometry generation [12]. Essential airplane components such as fuselage, wings, and nacelles can be generated with the program. The majority of geometry components are constructed from sub-components; for example, fuselages are lofted from a stack of cross-sections. Wing surfaces can be obtained by lofting stacks of airfoil cross-sections in the spanwise direction. There are many lofting techniques available, including linear and spline lofting for spanwise curvature. There are also various wing tip designs to choose from, and blunt trailing edges are available. Several aerodynamic analysis methods ranging in terms of fidelity can be employed using the grid provided by the RAGE tool.

Iqbal et al. [13] used Microsoft Excel sheets to obtain the geometrical parameters such as aspect ratio, dihedral angle, taper ratio, sweep angle, and other necessary inputs. Then these geometry data is lofted into the required three-dimensional drawing by coupling Excel sheets to a commercial CAD package. Further coupling to Computer-Aided Engineering (CAE) tools enables performing aerodynamic and structural analysis.

TiGL is also a prominent [14] open-source geometry modeling library developed by the German Aerospace Research Centre (DLR) to ease the conceptual and preliminary aircraft and helicopter design workflows. TiGL produces complete three-dimensional representations of aircraft with movable parts and rotorcrafts from their Common Parametric Aircraft Configuration Schema (CPACS), a standardized data model for the communication in design processes encompassing the parametric description of aircraft and helicopters [15, 16]. TiGL's parametric modeling abilities extend to structural components, permitting it to create a wing structure model comprised of spars and ribs. The parametric profile and guide curves provided by CPACS form the network interpolated by B-Splined based Gordon surfaces that ensure accurate geometry representations; therefore, TiGL well suits high-fidelity CFD analyses. An example of curve network interpolations through the profile and guide curves in TiGL is illustrated in Fig. 1.4. TiGL implements profile curves from a set of three-dimensional points as well as analytical wing sections description via the Class Shape Transformation (CST) method [17]. TiGL is coded in C++, but it has bindings for a wide range of computer languages, like C, Python, MATLAB, and Java. The software also includes a straightforward graphical user interface (GUI) called TiGL Viewer, which can view CPACS geometries and export them to a range of common CAD formats. Furthermore, the TiGL Viewer has a scripting console that may be used to automate visualization operations.

Another noteworthy open-source study is conducted by De Marco et al. [18]. They have developed a high-fidelity parametric geometry templates of external aircraft shapes to be used at the preliminary design workflows. These templates include fuselage, wing, nacelle, and movable surfaces. The parameters that define the aircraft components are tabulated in great detail, and lofting operations through auxiliary cross-sections are shown clearly. The proposed parametric geometry template was

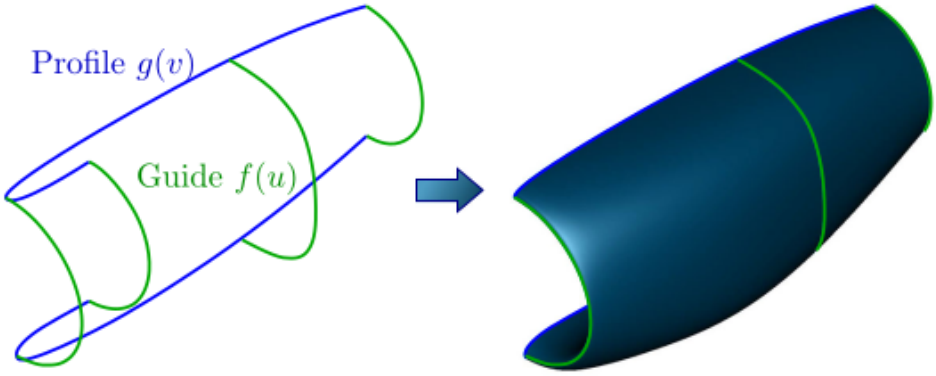


Figure 1.4: Curve network interpolation through profile and guide curves [14].

realized within the JPAD tool [19], a Java-based library conceived for aircraft designers. This tool involves a modeling module named JPADCAD to perform advanced CAD operations. The JPADCAD module is an application programming interface (API) established based on the OpenCASCADE CAD kernel [20].

The application of parametric geometries may be extended beyond the outer mold-line of the aircraft. Sanchez et al. [21] describe the requirements for enabling a parametric and automated linkage between a CAD modeler and the other disciplines in an MDAO workflow. The authors offer a system to facilitate integrating CAD modeler operations inside an MDAO process, allowing the incorporation of new disciplines into the aircraft conceptual design stage, such as a thermal risk analysis. The advantages of using a CAD geometrical modeler to perform a thermal risk evaluation of an aircraft equipment bay during the conceptual design phase are shown by a case study.

1.3 A Brief Review of Helicopter Rotor CFD

Problems in continuum mechanics, such as fluid dynamics, can be described by a set of partial differential equations such as Euler or Navier-Stokes equations, which need to be solved numerically for complex geometries. The numerical methods employed to solve the partial differential equations require a well-constructed computational grid that decomposes the physical domain into smaller, well defined volumes. The essential criteria for grids are a smooth distribution of grid points all over the flow-

field and a concentrated grid points cluster in regions where high gradients of flow quantities exist. The leading and trailing edges of the wing, and also regions where shock waves and separated flow emerge are among the reasons for the steep gradients. Considering the boundary layer where the viscous effect is significant in the vicinity of a surface, the grid must also be resolved extremely fine in the direction normal to the boundary surface since no-slip boundary condition results in increasing tangential velocity gradient with the decreasing wall distance. Also, grid cell orthogonality is one of the required features to solve equations accurately. Furthermore, unlike a fixed-wing aircraft, a helicopter rotor induces substantial flow velocities at vast distances from itself on account of the rotor wake structure [2]. This implies that the computational domain must be considerably bigger than similar fixed-wing ones, which increases computational cost. In addition, grid points should be appropriately distributed in the spanwise direction for a helicopter rotor considering the flight regime. Grid lines must also conform to non-linear variations such as blended profiles or other discontinuities on the blade. Moreover, the employed grid generation technique should allow refining the region where separated flow likely occurs. Therefore, a good computational mesh is essential to simulate complex flow around the rotor.

1.3.1 Mesh Generation Approaches

Mesh types can be classified into two categories based on their data structure and connectivity information between the points, known as structured and unstructured grids. Both structured and unstructured grids have individual benefits and drawbacks.

The idea behind the structured grids is that the grid points in the physical domain defined in the Cartesian x, y, z space are mapped onto a computational domain in ξ, η, ζ space. Adjacent grid points can be joined to form cubes in the computational domain and hexahedrons in physical space, and quadrilaterals are used in 2D grid discretization.Several topologies have been developed to provide desired grid properties for selected geometry models and numerical methods. H, O, and C-type grids are basic grid topologies that are frequently used in practice today.

Physical domains of these common grid topologies are illustrated in Fig. 1.5 with

their corresponding computational domains. H-type grids offer the capability to refine the grid near the surface of the blade or wings, but they may cause singularities for rounded edges such as the leading and trailing edges of airfoils. C and O type grids are very effective in obtaining accurate solutions of external flowfields with rounded shapes as they can be adjusted to conform to the curved airfoils. A combination of topologies such as C-O, H-O, and C-H, can also be used to streamline specific flow problems. To sum up, structured grids permit users to control interior point locations and cell sizes, as interior node positions are directly connected to the user-defined boundary nodes. Structured meshes are aligned in the flow direction leading to accurate results and a quick convergence behavior in CFD analyses as well. Furthermore, less memory and CPU time are required due to the well-organized data structures of structured grids. Another notable advantage is that imposed boundary conditions work well since orthogonal surfaces can be appropriately defined in a structured grid. Hexahedral meshes also provide the same accuracy with unstructured grids with significantly less number of cells. Lastly, structured grids facilitate the implementation of multigrid methods and high-order discretization schemes easier for fast convergence and accurate results. However, the adaptation of internal grid nodes to flow solutions is difficult, and generating grids for an arbitrary complex domain is challenging. [22].

Mixing of Cartesian and structured grids is also noteworthy approach. Cartesian grids are also a type of structured grid, which are comprised of cubes aligned with the Cartesian coordinate axes. They are generated by subdivision of an hexahedral domain several times, and removing the intersecting and outside cells. This mesh type can be generated, solved, and adapted to flow solutions easily. Nevertheless, the accuracy of the flow solution for curved shapes that are not aligned to Cartesian coordinates is abysmal. Implementing conforming structured curvilinear grids near curved geometries may be a remedy, as displayed in Fig. 1.6. However, special treatment is necessary for hanging nodes between the Cartesian and structured grids in the solver code.

In unstructured grids, cells and points are not ordered in a particular way such that adjacent cells or grid points cannot be recognized simply by their indices. The main benefit of the unstructured grids relies on the flexibility of triangular or tetrahedral

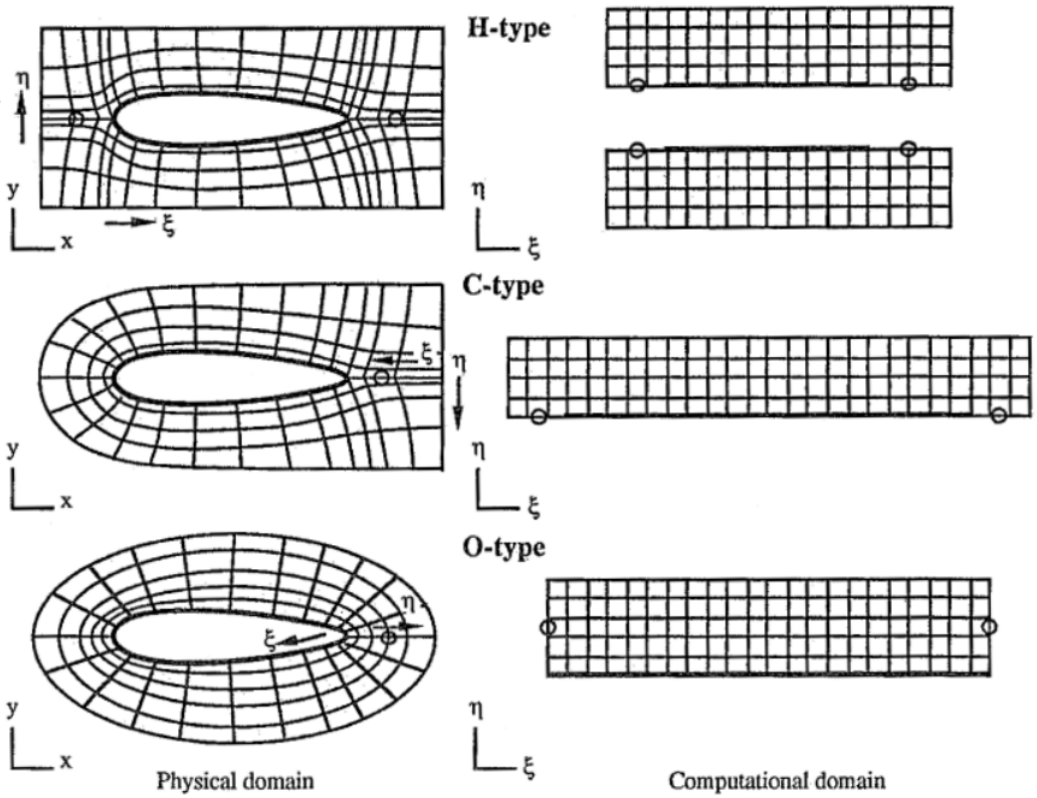


Figure 1.5: Common Grid Topologies [22].

grid elements that enables automatic grid generation regardless of the complexity of the geometry. For example, a complex three-element airfoil geometry from the structured grid point of view can be easily discretized through unstructured grids, as shown in Fig. 1.7, which is significantly difficult to generate with structured meshes. In addition, solution dependent grid adaptation can be handled seamlessly by removing or inserting grid points. The random order of grid points, on the other hand, requires more computation time and more memory for connectivity data; hence flow solvers built for unstructured grids are significantly slower compared to those for structured grids. Various element types such as hexahedrons, prisms, pyramids, and polyhedrons can further be combined with tetrahedrons to form hybrid grids For instance, hexahedrons can be employed to resolve viscous regions to preserve accuracy while exploiting the capability of tetrahedrons to treat complex geometries. Hybrid grids aim to achieve the advantages of both structure and unstructured methods [23].

Curvilinear structured grids may be employed to achieve accurate flow solutions in relatively simple geometries, yet creating a single structured block grid that correctly

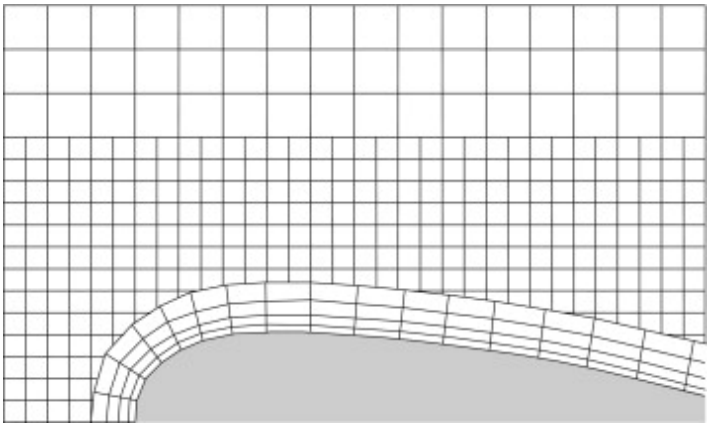


Figure 1.6: A mixed of cartesian and structured grid [23].

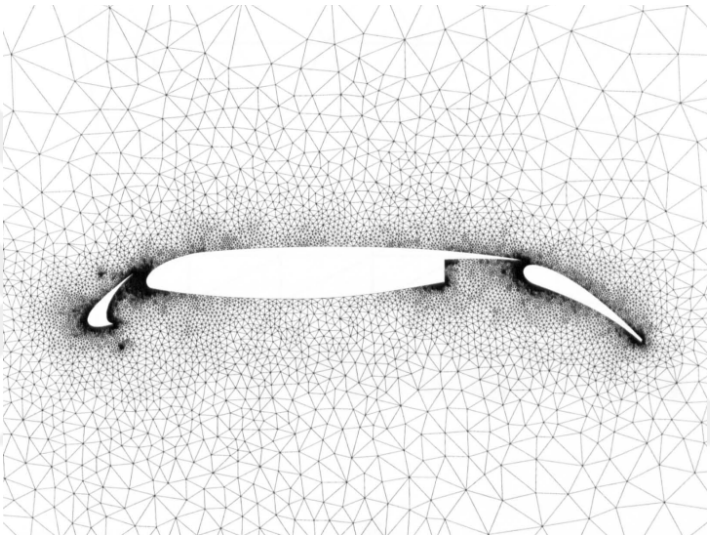


Figure 1.7: Unstructured grid around a three-element airfoil [24].

discretizes the physical domain is exceptionally challenging for complex geometries. The multiblock structured grid method has been developed as an alternative approach to address the problems related to complex geometries. In order to meet the desired grid point distributions in complex regions composed of several sub-components, the physical domain is divided into a series of conformally-connected structured blocks, which can be represented by different topologies. The divided blocks may subsequently be overlapped or patched with matched and non-matched boundaries. Indeed, the connectivity data between the sub-divided blocks increases the complexity of the solver because special treatment is needed to transfer physical quantities [25]. Fig. 1.8 demonstrates a close-up view of a multi-element airfoil discretized by the multiblock structured grid technique. The multiblock structured meshes may also be

exported to unstructured solvers. They provide excellent grid quality compared to unstructured grids.

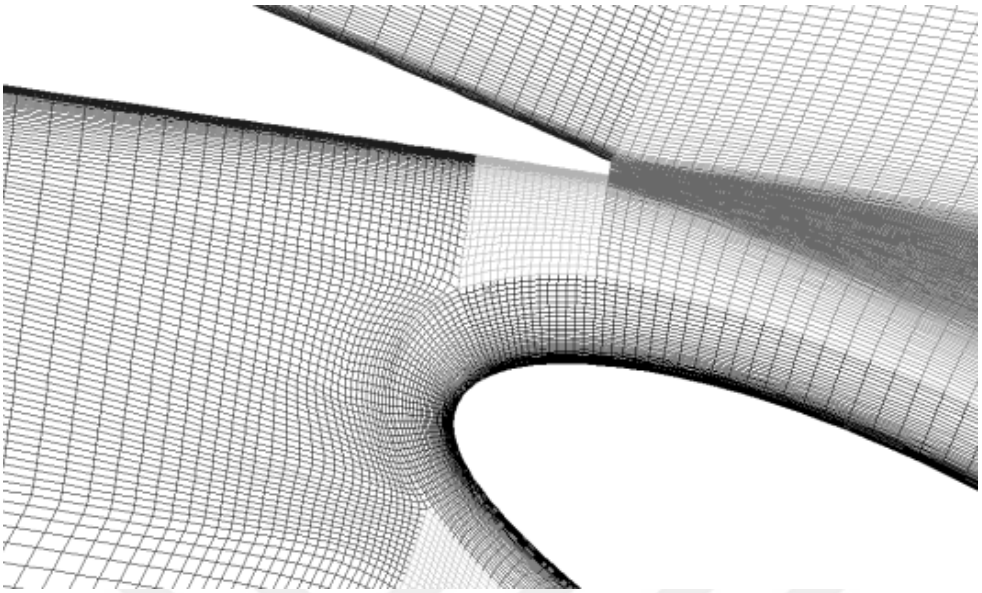


Figure 1.8: A close-up view of multiblock grid for a multi-element airfoil [26].

Excellent flexibility in the multiblock approach can be obtained via overset (Chimera) grids composed of entirely independent blocks that may overlap. Still, utilizing these grids in a solver is extremely expensive due to additional calculations required for overlapping regions as well as requirements for conserving flow quantities. Only a few solvers provide overset functionality. Non-overlapping multiblock meshes can be further categorized as discontinuous, C^0 continuity and C^1 continuity patched grids. Discontinuous patched grids are more customizable and simpler to construct since each block is generated individually without the matched grid point requirement. They may involve a varied grid point distribution. Nevertheless, the imposing of boundary conditions at common interfaces and the conservative interpolation required to transfer data across connected blocks are significant issues in adopting discontinuous grids. C^0 and C^1 continuity patches guarantee the conservation of governing equations as well as ease the implementation of boundary conditions, while continuity of grid lines between separate blocks limits the grid point distributions and the local refinement capability accordingly. C^0 (line continuity) and C^1 (line and slope continuity) continuity patches guarantee the conservation of governing equations as well as ease the implementation of boundary conditions, while continuity of grid lines between separate blocks limits the grid point distributions and the local refinement ca-

pability accordingly [22]. Types of multiblock structured grids are illustrated in Fig. 1.9.

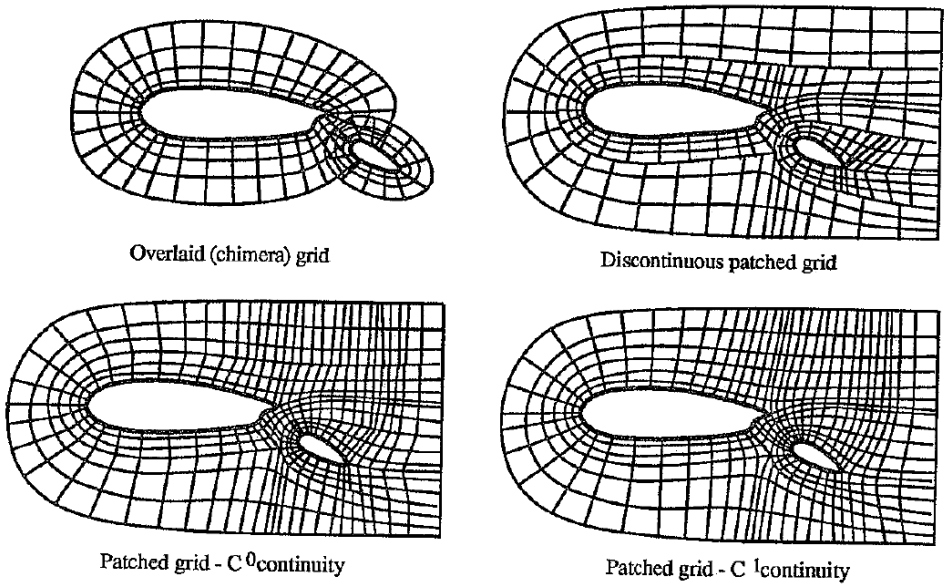


Figure 1.9: Multiblock grid types [22].

1.3.2 Template-Based Automatic Mesh Generation

Advances in CFD analysis methods and computer technology have reached a mature level, and today 3D Navier-Stokes simulations are frequently used for complex geometries in industrial environments. However, generating a high-quality grid around a complex geometry consumes more effort and time than analyzing and displaying the whole flowfield. There has been considerable interest in fast and robust mesh generation tools due to the high demand to cope with labor-intensive discretization tasks in aerodynamic analysis and optimization workflows. Several automatic grid generation systems have been developed to reduce human interaction in design loops. Most of the automation approaches are template-based for certain kinds of geometries.

Shih et al. [27] unified a parametric CAD modeler and mesh generator in the same platform to build templates. The CAD modeling and grid generating processes are based on the Geometry-Grid Toolkit library (GGTK) with a GUI, called MINICAD. Creating geometry and grid templates is possible with the Python scripting language. The multiblock structured capability is only shown for a shear coaxial element injec-

tor.

Pandya et al. [28] introduced a framework consisting of a set of scripting macros to automate structured overset mesh generation for rocket geometries from a CAD geometry model. Koseomur and Baran [29] presented an automated design code for missiles utilizing a multiblock structured grid generation method that relies on the geometry parameters provided by Missile DATCOM software.

Another area that employs the automatic multiblock technique is the turbomachinery. PADRAM is a design system developed by Shahpar and Lapworth [30], which includes a parametric design and rapid meshing system to be used in a turbomachinery optimization cycle with a high-fidelity CFD solver. A templated approach is presented that could create multiblock meshes for a set of problems and then be reused for a variety of similar geometries. Continuing on the topic of turbomachinery, Kundes [31] developed an automatic design tool for axial compressors by utilizing the multiblock technique. The generated grid is validated through performance analyses of Rotor 37 test case, and both designed compressor blade and original blade simulations fit remarkably well with the experimental data. Zhang and Barakos [32] present a tool chain for compound rotorcraft that comprises parametric design, automatic multiblock meshing, and CFD analysis, all of which are implemented by separate software packages. The geometry is parametrized through in-house codes, while ICEM CFD [33] with shell scripts realizes automatic multiblock structured grid generation. Various cases of duct propellers, an auxiliary propulsion component for compound rotorcraft, are examined and validated.

Shaw and Weatherill [34] describe a multiblock grid generation procedure with an automatic component-based block decomposition. This procedure is accompanied by a method that controls grid point distribution. The application of the method is demonstrated for a set of aircraft configurations. Euler equations are employed for flow simulations, and solutions are compared well with the experiment. Another automatic multiblock structured grid generation process with the aid of a parametric geometry modeler is presented by Ferrari [35]. The procedure entails creating a journal file (in .tcl format) in ICEMCFD that reads a parametric geometry created in the CATIA V5 tool. The journal file is editable for any geometry parameter. In addition,

some specific geometry functions are available in .tcl format that is not included in ICEMCFD. The automated method is shown for relatively complex geometries such as flap-track-fairing, wing fillet, and winglets.

There are also several studies of automatic multiblock grids in the external rotating machinery field. Allen [36] developed an automatic multiblock structured grid generation scheme based on a modified transfinite interpolation to obtain better orthogonality, smoothness, and spacing control. This automation method involves helicopter rotors and wings without movable surfaces and winglets. Surface patches and a target number of grid cells for the entire domain are sufficient inputs for the scheme. Thus it is suitable for a non-expert user. It can be also run in batch mode to be implemented in design and optimization studies. Doerffer and Szulc [37] present a simulation of flowfield around a hovering Caradonna-Tung rotor model in transonic regimes. A computational multiblock structured grid is built semi-automatically within the IGG (Numeca) software [38] via python scripts. Martinez et al. [39] developed a tool automating aerodynamic analysis stages of the flowfield around wind turbine rotor blades, named Aero-T, utilizing a set of python and shell scripts to drive IGG to generate block-structured grids. Besides, geometry modeling is also realized by a set of scripts enabling modification of various parameters. Another study using IGG to generate overlaid grids semi-automatically predicts helicopter main rotor flowfield in high-speed forward flight regimes [40]. Proceeding further with rotorcraft CFD, Xiang et al. [41] performed an unsteady flow simulation of an autorotating rotor in gyroplane level flight based on the semi-automatic multiblock division in ICEM.

Template-based studies targeted for a specific class of geometries have been mentioned up to now. Lu et al. propose a method to generate an automatic structured grid around different geometries for the boundary layer region using a surface grid as an input [42]. First, the outside frame of the boundary layer grid is built based on extracted geometry features. Then transfinite interpolation is employed to create interior grid points within the frame. Finally, the generality of the developed method is emphasized by applying it to three cases, namely typical concave and convex step geometries, the F6 wing-body model, and a four-rudder missile. The proposed multiblock method is further utilized in an interactive structured mesh generation computer program named NNW-GridStar [43].

Despite the expense in computational time and storage as well as the less accuracy, unstructured grids have much more tendency to be automated due to the flexibility in the case of complex geometries. Karman and Wyman [44] described a process that generates unstructured grids automatically from a geometry input. The Glyph scripting language, called AutoMesh, is used to creates meshes in commercial Pointwise software [45]. The geometry must be water-tight (closed) to generate volume meshes. Further commands, including viscous wall and endpoint spacings, can be delivered to Pointwise through geometry attribution. Several geometries involving a parametric aircraft model and rocket geometries demonstrate the capability of the AutoMesh.

1.4 Motivation and Scope Of The Thesis

The helicopter main rotor, as previously stated, plays a significant role in vertical flight capability, moving in any direction, and controlling the aircraft. Many analyses at various flight regimes and collective pitch angles must be performed throughout the design process of a helicopter rotor to achieve the desired requirements. A few days of setting up a new geometry model interactively and a couple of hours using a mesh generator to create an appropriate computational grid has been considered reasonable in the detailed design phase. Today, however, spending too much time for modeling and mesh generation is not acceptable in the conceptual and preliminary design stages. Thus, the main goal of this thesis is to reduce the time and repetitive human effort in the aerodynamic analysis workflows by developing a toolkit that automates the design of a helicopter rotor.

This thesis is a part of the comprehensive aerodynamic analysis tool developing by the Middle East Technical University and the Turkish Aerospace Industries company. The comprehensive framework comprises four main packages: a parametric geometry modeler, an automatic grid generation, a mesh deforming module, and a solver, as depicted in Fig. 1.10. The mesh deforming module is based on the spring analogy and has been developing to represent cyclic inputs in the forward flight. The solver part has been generated specialized for the multiblock structured grid. A high-order discretization scheme and grid adaptation method will be implemented into this solver to conserve the rotor wake accurately.

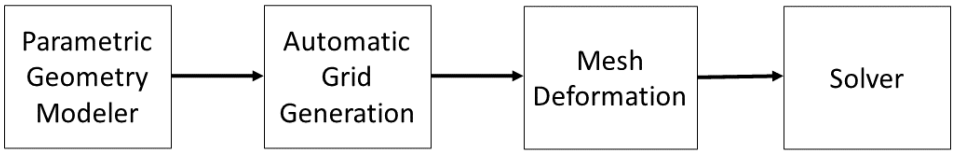


Figure 1.10: Comprehensive analysis tool.

This thesis includes the first two tasks, parametric geometry modeler and automatic grid generation, to design the helicopter main rotor. The flow chart of the developed design tool is shown in Fig. 1.11. The new tool must comply with design principles and practices of the helicopter design group of the company. Also, the new tool should fit other modules of the design framework. The mesh should be compatible to mesh deformation module, where the control inputs and rotor dynamics are handled. Also the generated mesh should comply with the solver requirements, the mesh format, turbulence-related requirements and high orthogonality.

The automation process starts with geometry that can be remodeled easily and rapidly in the design loop until performance criteria are met. In this study, the geometry generation of a helicopter rotor model is realized by a simple set of parameters: one or multiple airfoil definition, the wing span, the root and tip chords, and sweep, dihedral, twist angles. Trade studies on particular individual parameters can be performed effortlessly through a simple text input without dealing with CAD software throughout the design process. Also, this parametric geometry modeler can be the modeller foundation in a comprehensive analysis tool and even an optimization cycle in future studies.

The second step in the automatization method is generating a robust and accurate grid around the parametrized geometry. From the aforementioned literature, it is observed that the accuracy and convergence of any numerical flow field simulation are strongly dependent on the quality of the employed computational grid. It is known that skew, and non-orthogonal cells with high aspect ratio may decrease the spatial discretization order, so result in an increase in numerical dissipation.To fullfill these requirements, we have developed an automatic multiblock structured mesh generator for helicopter rotors. First, the computational domain generated in the first step is divided into

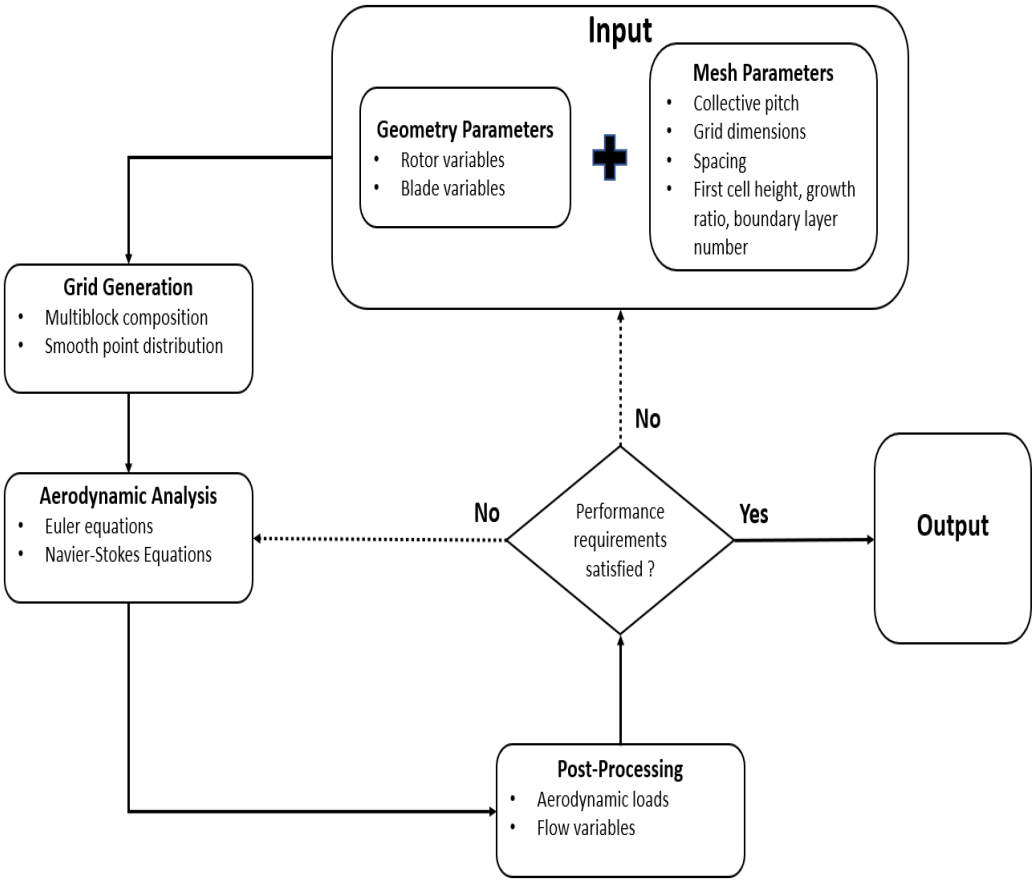


Figure 1.11: Flow chart of developed design tool.

blocks with matched boundaries automatically. Then, smooth grid distributions are ensured through stretching functions without human intervention. One can change the grid point spacing around the rotor blades within a simple text input, which is invisible to novice users.The boundary layer grid properties can be modified by layer number, growth ratio, and first cell height inputs.

As we have discussed earlier, several studies are conducted to accelerate the design stages of various aircraft and their components. However, only a few join the parametric geometry modeler and grid generator together into the same code. Fewer studies focus on helicopter rotors. Another aspect of the developed toolkit is the unification of the parametrized geometry and multiblock structured grid methods such that there is no need to exchange data.

After the mesh generation, the validation of the devised toolkit is presented. For this purpose, we have analyze the helicopter rotor in hover conditions with the computa-

tional volume and the grid generated by the developed toolkit, using an open-source solver, SU2. The performance requirements for the flight conditions can be checked, other flight conditions can be evaluated on automatically generated meshes with the same settings quickly. These settings include CFD boundary conditions and trim conditions. Also a new design cycle may be initialized easily by modifying geometry and mesh parameters..

This thesis is organized as follows. In Chapter 2, the parametric design process of a helicopter rotor is presented. Mathematical descriptions of the spline methods that form cross-sections of the rotor blades as well as the surface lofting techniques are explained. The theory behind multiblock structured grids with stretching functions is given in Chapter 3. In Chapter 4, the mathematical model of the governing equations are introduced. Then, in Chapter 5, validation and performance prediction of the test cases are realized. Finally, the conclusion and future works are presented in Chapter 6.

CHAPTER 2

PARAMETRIC ROTOR MODELING

The flow field around helicopter rotors heavily depends on the rotor wake structure. As previously described in Section 1.1, for helicopters in hover, the rotor wake and tip vortices remain close to blades for several revolutions and induce strong velocities that change the local incident angle of blades. This issue significantly affects the power loading, which is a performance parameter for rotors. Helicopters are typically designed to hover with the least amount of power needed for a given payload. This operation require high power loading, therefore hover condition is an initial sizing criterion. A modern helicopter rotor involves several different airfoil profile sections, complex profile blending, twists, and other complexities. Moreover, blade tip geometry can be selected from a wide variety of candidates. Influenced by these considerations, a well-established parametric rotor modeling system is developed, allowing linear and non-linear distribution of each parameter without disturbing others in an object-oriented data structure.

2.1 Object-Oriented Approach of the Modeler

An object-oriented approach to geometric modeling is beneficial for establishing definitions and rules. Thus the automation of the design cycles is enhanced by the well-organized code structure. Object-oriented programming is a software architecture wherein the programmer specifies both the data (attributes) and the methods (functions) operating on the data that are implemented in the objects. In this study, the object-oriented capabilities of the C++ programming language are exploited, including aggregation, inheritance, and polymorphism.

A functional geometry template is defined by all of the parameters required to create a three-view representation. The model provided herein proposes a wing composed of one or more watertight interconnected panels. In order to represent kink sections and other discontinuities accurately, panels can be built separately. Each panel is created as a child object inherited from the rotor class. Panel objects involve panel parameters, curve definitions, and editing functions. On the other hand, rotor parameters, surface definitions, and corresponding editing functions are included in the rotor class. An overview of the data groups is given in Fig. 2.1.

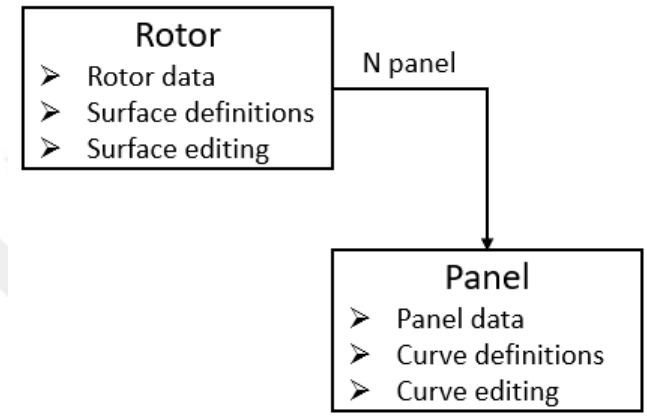


Figure 2.1: Organization of data groups in rotor modeling.

2.2 Parametric Rotor Geometry

The conceptual and preliminary design parameters of the helicopter rotor can be classified into the following groups.

- Rotor parameters: These involve determining the rotor diameter, number of blades, total number of panels per blade, and pitching axis. Several significant trade-off studies in performance with modifications in these parameters may be conducted.

- Panel parameters: These include planform dimensions such as chord and span for each panel. In addition, twist, dihedral, and sweep angles can be provided as both linear and non-linear distributions. Finally, airfoil selection with varying thickness distribution can also be given as a major design parameter to maxi-

mize the hover performance. The geometry modeller should handle all panel transitions.

2.2.1 Rotor Parameters

There is a trade-off between aerodynamic requirements and structural constraints for the rotor radius. Lower disk loadings and induced power needs are benefits of a larger rotor radius that leads to good hover performance. A larger radius also implies higher inertia and rotational kinetic energy, both required for safe autorotation capability. Initial main rotor design studies must consider autorotation characteristics, which need a certain amount of rotor inertia to satisfy certification requirements. On the other hand, a larger rotor radius would be detrimental to meet weight, speed, and maneuverability requirements. The rotor span is normalized with respect to rotor radius in the developed tool. Distances between far-field surfaces and the center of rotation are also normalized with respect to the rotor radius. The general rotor parameter inputs are given in Table 2.1.

Table 2.1: Rotor Parameters

Rotor radius	R
Number of blades	N_b
Number of panels	N_p
Pitching axis	θ

Lighter-weight helicopters generally have two blades, while the number of blades in larger helicopters varies from four to eight. It is possible to generate blades up to ten in the developed framework. The center of rotation point is taken as $(0, 0, 0)$ by default, and the pitching axis is the normalized distance between the center of rotation and the point 25% of the root chord back from the leading edge. An overview of rotor parameters as well as some of the panel parameters is shown in Fig 2.2. Since the objective of this study is to accelerate aerodynamic analysis workflows by providing a parametric CAD model for automatic grid generation, some typical design parameters such as rotor solidity are not direct inputs for the geometry, yet they can be satisfied with a combination of rotor and panel parameters input.

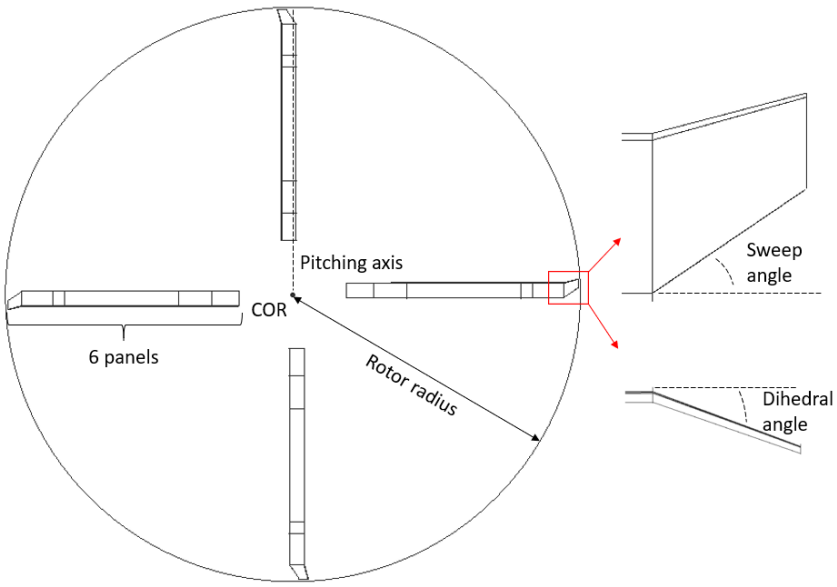


Figure 2.2: Overview of rotor parameters.

2.2.2 Panel Parameters

All necessary inputs for defining panels are presented in Table 2.2. The rotor blade is divided into N_p panels as defined in rotor parameters. Then, several control cross-sections with the corresponding number, N_g, can be equally placed along the panel span. For a panel, at least two cross-sections must be provided, one for the root of the panel and one for the tip.

Table 2.2: Panel Parameters

	i-th Panel
Span	b_i
Chord	c_i
Sweep angle	$\Lambda_{LE,i}$
Dihedral angle	Γ_i
Twist Angle	ϵ_i
Max. Thickness (%)	t/c
Cross-section number	$N_{g,i}$
Smooth	0 or 1
Airfoil curves	

Blade planform has a significant impact on blade lift distribution and, as a result, rotor performance. Slight amounts of taper in the blade tip region may substantially enhance the figure of merit (FM) in hover. In the developed tool, desired chord distribution can be scaled for each panel linearly as well as non-linearly with an additional input file. Sweep angle decreases the normal Mach number encountered by the leading edge, enabling the rotor to achieve a higher advance ratio, and prevents the increased drag and required power due to compressibility effects. Linear and non-linear varying sweep angle for each panel is also available in the developed tool.

Several blade tip geometries, including combinations of a sweep, taper, and anhedral, can be generated in the toolkit to carry out trade studies to improve the rotor performance. Some proposed blade tip designs shown in Fig. 2.3 can be easily created with the CAD modeler part of the developed tool.

An arbitrary rotor tip surface built in this CAD modeler with a mixed non-linear taper and the leading edge sweep angle is presented in Fig. 2.4. When used correctly, blade twist may increase the figure of merit in hover by a considerable amount. Thus, users are allowed to modify twist distribution through non-linear point clouds.

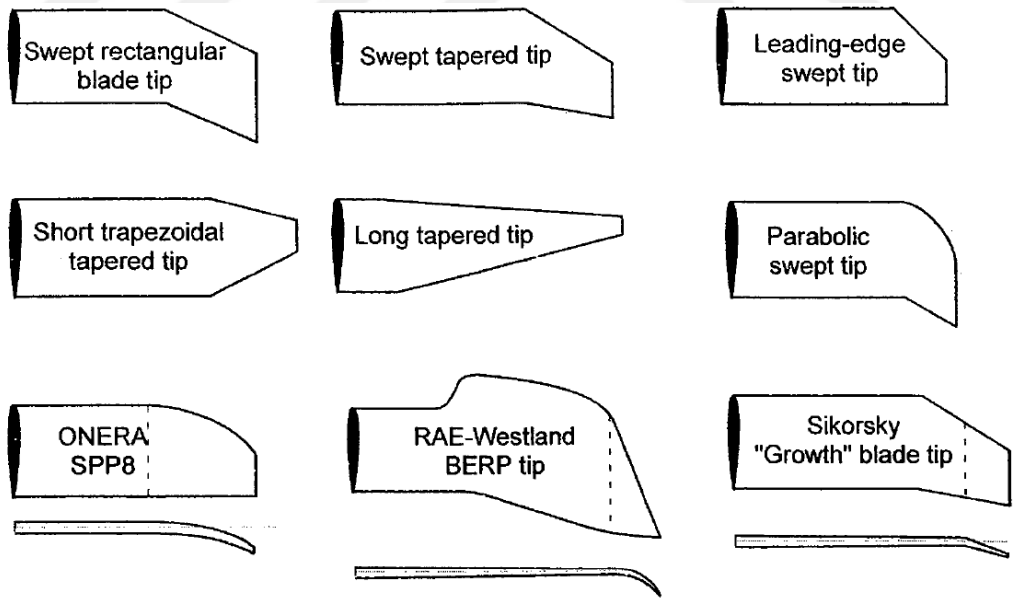


Figure 2.3: Some rotor tip designs [2].

Helicopters employ a range of airfoil sections that have been designed to meet the aerodynamic requirements of each station throughout the blade since helicopter rotors

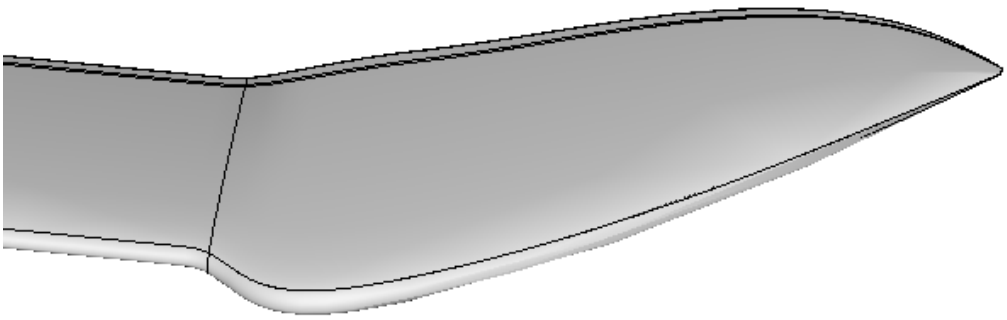


Figure 2.4: Rotor tip with blended notch.

are exposed to a wide variety of operating conditions and flight regimes from root to tip. Thus, various airfoil sections specified to each panel can be implemented into the tool with blending profiles. From a set of normalized, two-dimensional points given by the user, both B-Spline and NURBS curves can be constructed. Then, these curves are moved to the specified panel defined by the input. Corresponding scaling and rotation operations are applied to the panels. Moreover, the camber line of the curve is calculated automatically within the code, and blended profiles are achieved through the provided blade thickness distributions.

The continuity condition between two subsequent panels can be changed by the smooth attribute within the panel parameters. For example, a C^2 continuity can be imposed between two panels using a B-Spline interpolation. This feature is demonstrated in Fig. 2.5.

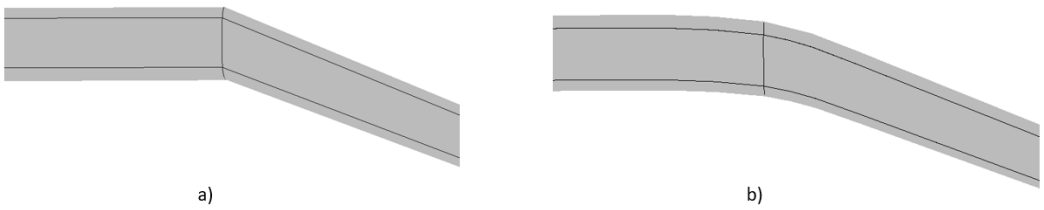
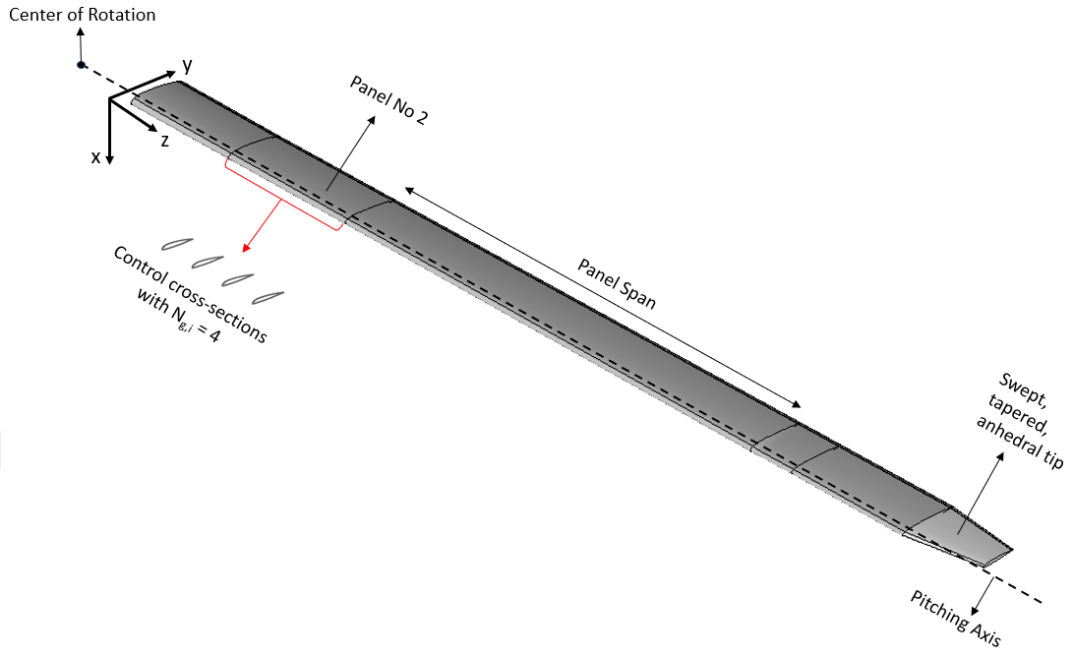


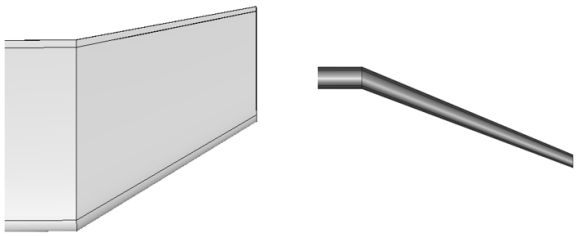
Figure 2.5: a) C^0continuity, b) C^2 continuity.

In order to present the capability of the developed parametric geometry modeler, a reasonably complex geometry ending with anhedral, tapered, and swept tip shape, the S-76 rotor model is chosen. Dimensions of the rotor are obtained by the experimental study of Balch and Lombardi [46]. Decomposition of the panels is realized according

to the discontinuities between the panel parameters. More control cross-sections are used to represent non-linear variations accurately than that of linear variations. S-76 model rotor with the developed tool is shown in Fig. 2.6.



(a) Isometric view of S-76 rotor



(b) Planform and side view of S-76 rotor

Figure 2.6: S-76 rotor modeled with the developed tool.

2.3 Generating Parametric Surface

Control cross-sections, namely airfoils, are built employing NURBS curves, then stacks of airfoils are lofted spanwise to generate rotor blades. Surface skinning can be realized linearly, or a B-spline of 3rd degree can be used. The B-spline and NURBS surface modeling algorithms applied in the developed tool are derived from formulations as described in [47, 48]. These algorithms and formulations are explained in the

following sections.

2.3.1 B-Spline Curves

Let $T = \{t_0, ..., t_m\}$ be a nondecreasing sequence of real numbers, such that $t_i < t_{i+1}$ for $i = 0, ..., m$. T is the knot vector, and t_i numbers are named as knots. Then, the i^{th} B-spline basis function of p^{th} degree, indicated as $N_{i,p}(t)$, is defined by

$$N_{i,0}(t) = \begin{cases} 1 & if\ u_i \leq u \leq u_{i+1} \\ 0 & otherwise \end{cases}$$
$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \tag{2.1}$$

A B-spline curve of degree p is defined by

$$C(t) = \sum_{i=0}^{n} N_{i,p}(t)\mathbf{P_i} \qquad a \leq t \leq b \tag{2.2}$$

where the $\mathbf{P_i}$ are the control points and $N_{i,p}(t)$ are the B-spline basis functions of degree p (Eq. 2.1) defined on the following knot vector with total $m + 1$ knots and $n + 1$ control points.

$$T = \{\underbrace{a, ..., a}_{p+1}, t_{p+1}, ..., t_{m-p-1}, \underbrace{b, ..., b}_{p+1}\}$$

Since a 3^{rd} degree B-spline curve is used in this study, the above form of knot vector ensures C^2 continuity for the interior control points. It is also assumed that $a = 0$ and $b = 1$. Number of knots can be calculated with the following relation:

$$m = n + p + 1 \tag{2.3}$$

A 3^{rd} degree B-spline definition is directly used as fitting curves without further interpolation or approximation for computational simplicity and fast solutions.

2.3.2 NURBS Curves

A pth degree NURBS curve is defined by

$$C(t) = \frac{\sum_{i=0}^{n} N_{i,p}(t) w_i \mathbf{P_i}}{\sum_{i=0}^{n} N_{i,p}(t) w_i} \qquad a \leq t \leq b \tag{2.4}$$

where the $\mathbf{P_i}$ are the control points, w_i are the weights, and $N_{i,p}(t)$ are B-spline basis functions of degree p defined on a knot vector, same form that is used in the case of B-spline. It is assumed that $a = 0$, $b = 1$, and for all i $w_i > 0$.

Defining a rational basis functions, $R_{i,p}(t)$:

$$R_{i,p} = \frac{N_{i,p}(t) w_i}{\sum_{j=0}^{n} N_{j,p}(t) w_j}$$

Then, Eq. 2.4 becomes

$$C(t) = \sum_{i=0}^{n} R_{i,p}(t) \mathbf{P_i} \tag{2.5}$$

A 3^{rd} degree NURBS definition is directly used as fitting curves without further interpolation or approximation for computational simplicity and fast solutions.

2.3.3 Curve Parametrization

There are three standard methods to parametrize curves implemented into the developed tool.

1. Uniform spacing method: It is the simplest method and produce undesirable curves for non-uniform data points.

$$t_i = \frac{i}{n} \qquad i = 1, ..., n-1 \tag{2.6}$$

2. Chord length method: Let s be the total chord length that is composed of the distances between given data points, D_i. This method gives better results for

non-uniform data points.

$$s = \sum_{i=1}^{n} \mid D_i - D_{i-1} \mid$$

Then,

$$t_i = t_{i-1} + \frac{\mid D_i - D_{i-1} \mid}{s} \qquad i = 1, ..., n-1 \tag{2.7}$$

3. Centripetal method: This method gives better results for sharp turns.

$$s = \sum_{i=1}^{n} \sqrt{\mid D_i - D_{i-1} \mid}$$

Then,

$$t_i = t_{i-1} + \frac{\sqrt{\mid D_i - D_{i-1} \mid}}{s} \qquad i = 1, ..., n-1 \tag{2.8}$$

CHAPTER 3

AUTOMATIC MESH GENERATION

3.1 Multiblock Structured Grid Generation

A patched multiblock structured grid generating code is developed within the scope of this thesis. The multiblock structured mesh method is computationally efficient, provides high-quality discretization for CFD analysis, and offers flexibility for moderately complex geometries. Although no automatic multi-block mesh generator exists and manual meshing is very time consuming, a template-based meshing is possible [38]. Thus, a grid template of this type is constructed and included in the aerodynamic analysis workflows of helicopter rotors. The developed code relies on Dener's [22] studies which are revised to produce automatic grids with the support of parametric geometries mentioned in Chapter 2.

First, an overview of the mesh generator will be presented with its object-oriented structure. Then, the theory used for grid generation, namely algebraic grid generation, which is a high-speed and straightforward method, will be explained. The mapping procedure of surface grids specified with parametric geometry will be described afterward. Lastly, the fully automatic grid generation around an arbitrary rotor model will be given as an exanple.

3.2 Object-Oriented Approach and Overview of the Mesh Generator

The physical domain of the flow field is divided into several structured blocks that are connected to each other with matching grid patches. Each structured block consists of six boundary faces. Boundary conditions for flow solvers can be defined on

these faces. Boundary faces are obtained from its four edges, where each edge includes a specified number of segments to represent discontinuities within the given geometry. Segments can be defined with a wide range of curve interpolation methods such as Lagrange, Hermite, B-spline, or NURBS functions. The subcomponents of a structured block are shown in Fig. 3.1.

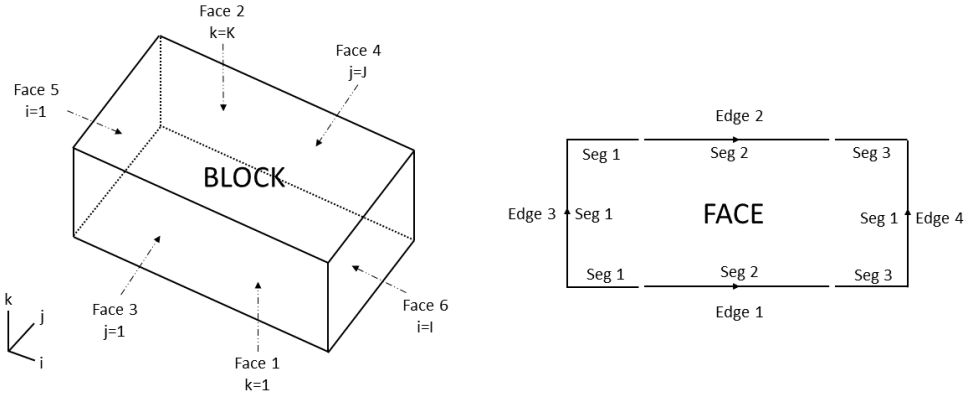


Figure 3.1: Block and face elements.

The curve object that is a child object of the segment data group processes the curve description produced by panel data to distribute grid points on the boundaries. Surface data is also transferred from the rotor to the surface object to achieve mapping operation together with the boundary curves. The flow chart given in Fig. 3.2 demonstrates the overview of the unified parametric geometry modeling and mesh generation framework.

3.3 Algebraic Grid Generation

Interpolation from specified coordinates or derivatives on the boundaries to find grid point coordinates inside the computational domain is referred to as transfinite interpolation. Grid point coordinates (x, y, z) within the physical domain can be described as normalized parametric coordinates (u, v, w), as demonstrated in Fig. 3.3. The transformation from the computational to physical domain is expressed by Eq. 3.1.

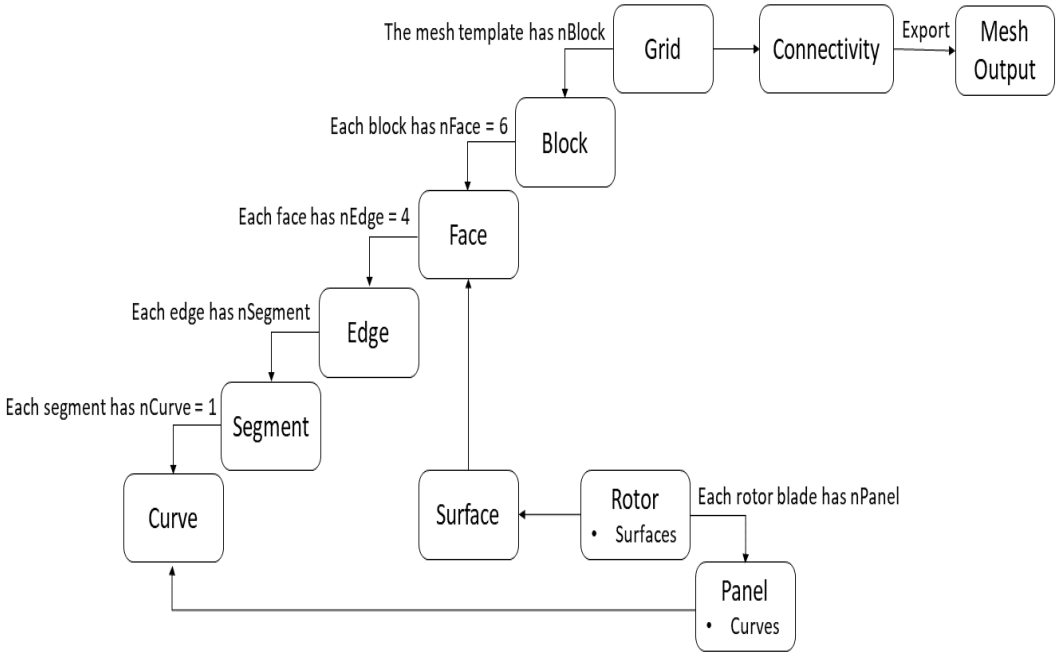


Figure 3.2: Data structure of the unified system.

$$\overrightarrow{r}(u,v,w) = \begin{bmatrix} x(u,v,w) \\ y(u,v,w) \\ z(u,v,w) \end{bmatrix} \tag{3.1}$$

where $0 \leq u \leq 1, \quad 0 \leq v \leq 1, \quad 0 \leq w \leq 1$.

Faces of a block in the computational domain can be described as shown in the Table 3.1.

Table 3.1: Definitions of Block Faces

$\overrightarrow{r}(0,v,w)$ and $\overrightarrow{r}(1,v,w)$ are faces where $u = 0$ and $u = 1$
$\overrightarrow{r}(u,0,w)$ and $\overrightarrow{r}(u,1,w)$ are faces where $v = 0$ and $v = 1$
$\overrightarrow{r}(u,v,0)$ and $\overrightarrow{r}(u,v,1)$ are faces where $w = 0$ and $w = 1$

Projectors, P_u, P_v , and P_w, can be employed to perform unidirectional interpolation in u, v, w directions to match the specified boundary grid points.

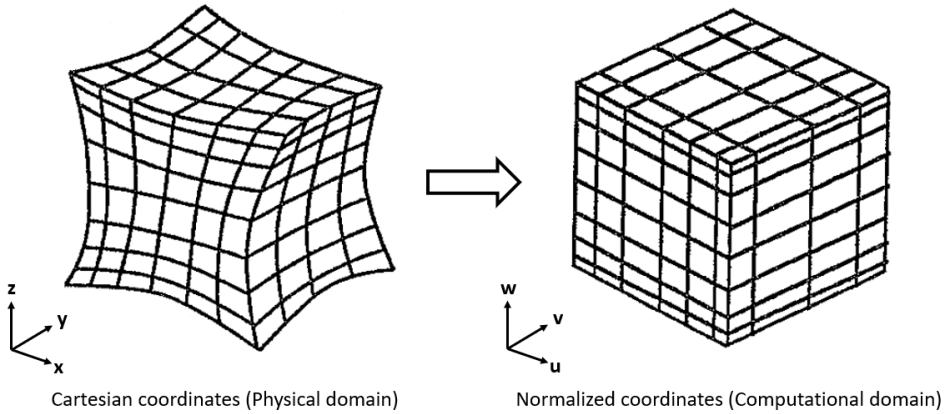


Figure 3.3: Notations for physical and computational domains.

$$
\begin{aligned}
P_u(\overrightarrow{r}) &= \sum_{i=0}^{1} L_i(u)\ \overrightarrow{r}(i,v,w) \\
P_v(\overrightarrow{r}) &= \sum_{j=0}^{1} L_j(v)\ \overrightarrow{r}(u,j,w) \\
P_w(\overrightarrow{r}) &= \sum_{k=0}^{1} L_k(w)\ \overrightarrow{r}(u,v,k)
\end{aligned} \tag{3.2}
$$

where, i, j, k indices denote boundary surfaces and $L_i(u)$, $L_j(v)$, $L_k(w)$ are the blending functions, which must satisfy the following condition.

$$
L_m(l) = \delta_{lm} \qquad \begin{cases} 1 & l = m \\ 0 & l \neq m \end{cases} \qquad for\ l, m = 0, 1 \tag{3.3}
$$

Interpolation functions may incorporate boundary derivatives to provide more control over the distribution of internal grid points. Then, projectors can be defined as:

$$P_u(\overrightarrow{r}) = \sum_{i=0}^{1}\sum_{n=0}^{1} H_i^n(u)\ \overrightarrow{r}^n(i,v,w)$$
$$P_v(\overrightarrow{r}) = \sum_{j=0}^{1}\sum_{n=0}^{1} H_j^n(v)\ \overrightarrow{r}^n(u,j,w)$$
$$P_w(\overrightarrow{r}) = \sum_{k=0}^{1}\sum_{n=0}^{1} H_k^n(w)\ \overrightarrow{r}^n(u,v,k) \tag{3.4}$$

where n shows the order of derivatives, $n = 1$ implies the first derivative in the specified grid point directions. $H_i^n(u)$, $H_j^n(v)$, $H_k^n(w)$ are the blending functions in separate u, v, w directions with the following condition:

$$H_m^n(l) = \delta_{lm}\delta_{nm} \qquad \begin{cases} 1 & l = m,\ n = m \\ 0 & l \neq m,\ n \neq m \end{cases} \qquad for\ l, m, n = 0, 1 \tag{3.5}$$

The transfinite interpolation relies on the combination of a series of unidirectional interpolations to establish coordinate transformation [49, 50]. Unidirectional interpolation methods are used to interpolate parametric coordinated in u, v, w directions to specified points on a curvilinear boundary curve. A wide range of methods can be employed. For simple curves, Lagrange and Hermite polynomials and Bezier curves based on Bernstein polynomials can be used. However, piecewise polynomials (cubic splines) give more accurate results, particularly for complex boundary curves [22]. In the following sections, firstly, unidirectional interpolation techniques and the functions that provide smooth grid distributions will be discussed. Then, details of the transfinite interpolation will be expressed.

3.3.1 Polynomial Unidirectional Interpolation

Polynomial interpolation is the process of matching a single polynomial to a set of grid point coordinates. Increasing the number of interpolation points to represent complex curves ultimately leads to extreme oscillations, and the shape of boundary curves can not be preserved. However, interpolation to relatively simple curves, polynomial methods provide high-speed solutions with minimum point data.

3.3.1.1 Lagrange Interpolation

The Lagrange interpolation function for specified points can be expressed as:

$$\overrightarrow{r}(u) = \sum_{i=0}^{I} L_i(u)\overrightarrow{r}(u_i) \qquad for\ 0 \leq i \leq I,\ and\ 0 \leq u \leq 1 \tag{3.6}$$

and Lagrange polynomial can be given as:

$$L_i(u) = \prod_{\substack{j=0 \\ j \neq i}}^{I} \frac{u - u_j}{u_i - u_j} \tag{3.7}$$

Linear blending functions can be expressed as:

$$\begin{aligned} L_0(u) &= (1-u) \\ L_1(u) &= u \end{aligned} \tag{3.8}$$

Then, the corresponding interpolation function becomes:

$$\overrightarrow{r}(u) = (1-u)\overrightarrow{r}(0) + u\overrightarrow{r}(1) \tag{3.9}$$

The interpolation function to three points, including the middle point of the boundary curve, is defined as:

$$\overrightarrow{r}(u) = (1 - 3u + 2u^2)\overrightarrow{r}(u_0) + 4(u - u^2)\overrightarrow{r}(0.5) + (2u^2 - u)\overrightarrow{r}(u_1) \tag{3.10}$$

3.3.1.2 Hermite Interpolation

The Hermite function interpolates both grid points ($\overrightarrow{r}(u)$) and derivatives ($\overrightarrow{r}'(u)$) on the boundaries to obtain better orthogonality at the boundaries. Definition of the Hermite interpolation can be given as:

$$\overrightarrow{r}(u) = \sum_{i=0}^{I} \sum_{n=0}^{1} H_i^n(u)\, \overrightarrow{r}^n(u_i) \tag{3.11}$$

where n denotes the order of derivative. The Lagrange polynomial may be used to derive Hermite functions as follows:

$$\begin{aligned} H_i^0(u) &= (1 - 2(1-u)L_i'(u))(L_i(u))^2 \\ H_i^1(u) &= (1-u)(L_i(u))^2 \end{aligned} \tag{3.12}$$

$L_i'(u)$ can be obtain by differantiating Eq. 3.7. The Hermite functions can be expressed as:

$$\begin{aligned} H_0^0(u) &= (1+2u)(1-u)^2 \\ H_0^1(u) &= u(1-u)^2 \\ H_1^0(u) &= (3-2u)u^2 \\ H_1^1(u) &= (u-1)u^2 \end{aligned} \tag{3.13}$$

Then, substituting of Eq. 3.13 into Eq.3.11 yields:

$$\overrightarrow{r}(u) = (1-3u^2+2u^3)\overrightarrow{r}(0) + u(1-u)^2\overrightarrow{r}'(0) + (u-1)u^2\overrightarrow{r}'(1) + (3-2u)u^2\overrightarrow{r}(1) \tag{3.14}$$

The grid lines have a tendency to fit the tangent directions as there are only two points.

3.3.1.3 Bezier Curve

Based on Bernstein polynomials, Bezier curves are also a polynomial interpolation method and form a basis to construct industry-standard piecewise splines [48]. For a specified set of control points, $\overrightarrow{r_i}$, Bezier curves can be written as:

$$\overrightarrow{r}(u) = \sum_{i=0}^{n} B_i^n(u)\, \overrightarrow{r_i} \tag{3.15}$$

where $0 \leq i \leq n$,and the Bernstein polynomial of degree n is denoted by:

$$B_i^n(u) = \frac{n!}{(n-i)!\, i!} u^i (1-u)^{n-i} \tag{3.16}$$

For $n = 3$, Bernstein polynomials become:

$$\begin{aligned} B_0^3(u) &= (1-u)^3 \\ B_1^3(u) &= 3u(1-u)^2 \\ B_2^3(u) &= 3(1-u)u^2 \\ B_3^3(u) &= u^3 \end{aligned} \tag{3.17}$$

Substituting of Eq. 3.17 into Eq. 3.15:

$$\overrightarrow{r}(u) = (1-u)^3 \overrightarrow{r_0}(u) + 3u(1-u)^2 \overrightarrow{r_1}(u) + 3(1-u)u^2 \overrightarrow{r_2}(u) + u^3 \overrightarrow{r_3}(u) \tag{3.18}$$

3.3.2 Piecewise Polynomial Interpolation

Curves consisting of a single polynomial require a high degree to fit polynomials passing through many points to represent complex shapes. However, high degree curves are numerically unstable and inefficient. Thus, piecewise polynomial curves can be utilized to fit curves with any complexity. NURBS and B-spline definitions are given in Eq. 2.4 and Eq. 2.2 with C^2 (two times continuously differentiable) continuity are used to interpolate in u, v, w directions to the given boundary grid point coordinates. The general shape of a curve may be determined by using control points to represent specified grid points. The parametric distribution of the interior points can also be achieved by one of the methods given in Eq. 2.6, Eq. 2.7 ,or Eq. 2.8.

3.3.3 Spacing Functions

Particular grid point distribution may be needed for the regions such as leading and trailing edges or boundary layers where a high gradient of flow variables exist. Desired grid point distributions for specific regions can be obtained with spacing functions, another form of interpolants used in algebraic grid generation. The general form of spacing functions for only two positions can be written as follows:

$$\overrightarrow{r}(u) = \overrightarrow{r}(0) + [\overrightarrow{r}(1) - \overrightarrow{r}(0)]\, s(u) \tag{3.19}$$

where s is the spacing function with the condition, $s(0) = 0$ and $s(1) = 1$. Hyperbolic tangent, hyperbolic sine, and geometric distribution functions have been widely used for grid point distributions.

3.3.3.1 Hyperbolic Tangent Function

The starting point spacing on an arc length given at one or both ends or an interior point can be used to define the hyperbolic tangent function [51]. The point distribution on the arc length is the spacing function. Let the arc length be $s(u)$, ranges from 0 to 1 where $0 \leq u \leq 1$.

- At both ends:

Let $\triangle s_1$ and $\triangle s_2$ be grid point spacing at the beginning and end, respectively, where $u = 0$ and $u = 1$. Then, the hyperbolic tangent function can be defined as:

$$s(u) = \frac{\Phi(u)}{A + (1 - A)\Phi(u)} \tag{3.20}$$

where $\Phi(u) = \frac{1}{2}\left\{1 + \frac{\tanh[(2u-1)\delta/2]}{\tanh(\delta/2)}\right\}$, A and B can be obtained in terms of specified end point spacing as follows:

$$A = \sqrt{\frac{\triangle s_1}{\triangle s_2}}, \qquad B = \frac{1}{\sqrt{\triangle s_1 \triangle s_2}} \tag{3.21}$$

Then, δ can be found with the following equation:

$$B = \frac{\sinh \delta}{\delta} \tag{3.22}$$

An approximate analytical solution for Eq. 3.22 yields:

For $B < 2.7829681$,

$$\delta = \sqrt{6w}(1 - 0.15w + 0.057321429w^2 - 0.024907295w^3 + 0.007742446w^4 - 0.0010794123w^5) \tag{3.23}$$

where $w = B - 1$.

For $B \geq 2.7829681$,

$$\delta = \vartheta + (1 + 1/\vartheta)\log 2\vartheta - 0.020417939 - 0.24902722w + 1.94964436w^2 - 2.6294547w^3 + 8.567959116w^4 \tag{3.24}$$

where $\vartheta = \log B$ and $w = 1/B - 0.028527431$.

- At the start point:

Let $\triangle s$ be specified grid point spacing at the beginning of the arc length, $u = 0$. Then, the hyperbolic tangent function can be described as:

$$s(u) = 1 + \frac{\tanh\left[(u-1)\delta/2\right]}{\tanh(\delta/2)} \tag{3.25}$$

Same procedure is followed to find hyperbolic tangent function, except the calculation of B, which is defined as:

$$B = 1/\triangle s \tag{3.26}$$

- At the end point:

Let $\triangle s$ be specified grid point spacing at the end of the arc length, $u = 1$. Then, the hyperbolic tangent function can be described as:

$$s(u) = \frac{\tanh\left(\frac{u\delta}{2}\right)}{\tanh(\delta/2)} \tag{3.27}$$

Same procedure can be followed to find hyperbolic tangent function, except the calculation of B, which is defined as in Eq. 3.26.

- At an arbitrary internal point:

Let $\triangle s$ be specified grid point spacing at an arbitrary internal point of the arc length, where the interior point $u = \chi$, and $s(\chi) = \sigma$. Then, the hyperbolic tangent function can be described as:

$$s(u) = \sigma \left\{ 1 + \frac{\sinh\left[\delta(u - \chi)\right]}{\sinh(\chi\,\delta)} \right\} \tag{3.28}$$

χ can be calculated as follows:

$$\chi = \frac{1}{\delta} \tanh^{-1} \left(\frac{\sinh\delta}{(1/\sigma) + \cosh\delta - 1} \right) \tag{3.29}$$

B is defined as in Eq.3.26, then δ can be obtained from the following expression:

$$1 + \left(\frac{1}{B\delta\sigma} \right)^2 = \left(\frac{\cosh\delta - 1 + (1/\sigma)}{\sinh\delta} \right)^2 \tag{3.30}$$

A simplification to the Eq. 3.30 may be achieved if the interior point is not too close to the start and endpoints and B is large enough. Assume $\exp(-2\delta) \ll 1$, then

$$\frac{\sinh(\delta/2)}{\delta/2} = 2B\sqrt{\sigma(1-\sigma)} \tag{3.31}$$

An analytical approximation used for Eq. 3.22 can be applied to Eq. 3.31.

3.3.3.2 Hyperbolic Sine Function

The hyperbolic sine function provides better control near the start and end points than that of the hyperbolic tangent function. Point distribution for specified spacing at an internal point is the same obtained with the hyperbolic tangent function. Let the arc length be $s(u)$, ranges from 0 to 1 where $0 \leq u \leq 1$.

- At both ends:

Let $\triangle s_1$ and $\triangle s_2$ be grid point spacing at both ends, respectively, where $u = 0$ and $u = 1$. Then, the hyperbolic sine function can be defined as:

$$s(u) = \frac{\Phi(u)}{A + (1 - A)\Phi(u)} \tag{3.32}$$

where $\Phi(u) = \frac{1}{2}\left\{1 + \frac{\sinh[(2u-1)\delta/2]}{\sinh(\delta/2)}\right\}$, A and B can be obtained as given in Eq. 3.21. Then, δ can be found by solving the following equation.

$$B = \frac{\tanh(\delta/2)}{\delta/2} \tag{3.33}$$

- At the start point:

For the specified spacing $\triangle s$ at the start point $u = 0$, the function becomes:

$$s(u) = \frac{\sinh(\delta u)}{\sinh(\delta)} \tag{3.34}$$

The same procedure is followed to find hyperbolic sine function, except the calculation of B, which is defined as $B = 1/\triangle s$.

- At the endpoint:

For the specified spacing $\triangle s$ at the endpoint spacing $u = 1$, then the distribution is defined as:

$$s(u) = 1 - \frac{\sinh(\delta(1-u))}{\sinh(\delta)} \tag{3.35}$$

The same procedure can be followed with $B = 1/\triangle s$.

3.3.3.3 Geometric Distribution Function

Smooth grid point distributions may be generated by providing a constant growth ratio, c, between each successive parametric interval, $\triangle u_i$. Initial spacing can further be specified for one or both ends. For the boundary layer region, the constant growth ratio, first cell height (initial spacing), and total layer number are taken as inputs within the code. The condition that is satisfied for geometric distribution is as follows:

$$\frac{\triangle u_i}{\triangle u_{i-1}} = c, \qquad for\ 1 \leq i \leq I$$

where, I is the total number of points on a boundary curve. Then, the geometric distribution function can be given as:

$$s(u_i) = \triangle u_0 \sum_{n=0}^{i} c^n$$

where, $\triangle u_0$ is the initial spacing.

3.3.4 Transfinite Interpolation

The unidirectional interpolation methods previously discussed are essentially utilized to construct boundary curves connecting the given points. First, the boundary points for four edges of a 2D domain or six faces of a 3D domain are fitted to specified points. Then, unidirectional interpolation in each coordinate direction is coupled to perform a multidirectional interpolation that satisfies all of the boundaries included. This multidirectional interpolation scheme is named as transfinite interpolation. Two, four, and six boundary interpolation techniques are the most common kinds of transfinite interpolation.

3.3.4.1 Two Boundary Interpolation

Projectors P_u, P_v, and P_w are defined to interpolate in u, v, w directions to grid coordinates or derivatives between two opposing boundaries (which is also described in Table 3.1) are shown in Fig 3.4.

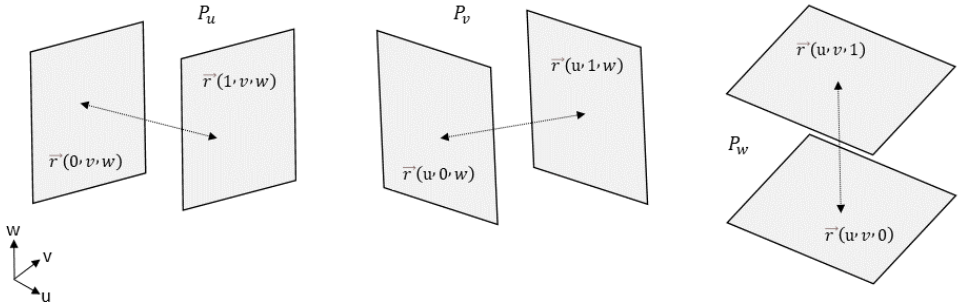


Figure 3.4: Projector definitions used in two boundary interpolations.

The linear two boundary interpolation can be used for simple geometries since it is the fastest type of transfinite interpolation. The linear or quadratic Lagrange functions can be implemented into projectors. The interpolation procedure can be written as:

$$
\begin{aligned}
P_u(\overrightarrow{r}) &= L_0(u)\,\overrightarrow{r}(0,v,w) + L_1(u)\,\overrightarrow{r}(1,v,w) \\
P_v(\overrightarrow{r}) &= L_0(v)\,\overrightarrow{r}(u,0,w) + L_1(v)\,\overrightarrow{r}(u,1,w) \\
P_w(\overrightarrow{r}) &= L_0(w)\,\overrightarrow{r}(u,v,0) + L_1(w)\,\overrightarrow{r}(u,v,1)
\end{aligned}
\tag{3.36}
$$

where, L_0 and L_1 are blending functions defined by Eq. 3.7. For regions that need better orthogonality, Hermite functions given by Eq. 3.13 can be utilized. Then, projectors are expressed as:

$$
\begin{aligned}
P_u(\overrightarrow{r}) &= H_0^0(u)\,\overrightarrow{r}(0,v,w) + H_0^1(u)\,\overrightarrow{r_u}(0,v,w) \\
&+H_1^1(u)\,\overrightarrow{r_u}(1,v,w) + H_1^0(u)\,\overrightarrow{r}(1,v,w) \\
P_v(\overrightarrow{r}) &= H_0^0(v)\,\overrightarrow{r}(u,0,w) + H_0^1(v)\,\overrightarrow{r_v}(u,0,w) \\
&+H_1^1(v)\,\overrightarrow{r_v}(u,1,w) + H_1^0(v)\,\overrightarrow{r}(u,1,w) \\
P_w(\overrightarrow{r}) &= H_0^0(w)\,\overrightarrow{r}(u,v,0) + H_0^1(w)\,\overrightarrow{r_w}(u,v,0) \\
&+H_1^1(w)\,\overrightarrow{r_w}(u,v,1) + H_1^0(w)\,\overrightarrow{r}(u,v,1)
\end{aligned} \tag{3.37}
$$

Subscripts denote the derivatives of boundary curves.

Spacing functions can be used to achieve desired grid density through redistributing points in related directions. The hyperbolic tangent, sine and geometric functions can be employed at one or both ends.

Interior parametric coordinates can be obtained by tensor product interpolation. For 2D domains, point distributions defined on two boundary curves connecting opposite edges are used to calculate internal grid points. For 3D domains, tensor product interpolation is applied to four edges connecting two opposite surfaces. The tensor product interpolation can be given as:

$$
P_s P_t = P_s(P_t(\overrightarrow{r})) \tag{3.38}
$$

where s and t are any two of the parametric coordinate indices.

The two boundary interpolation schemes by linear tensor product (bilinear interpolation) used within the code are expressed as:

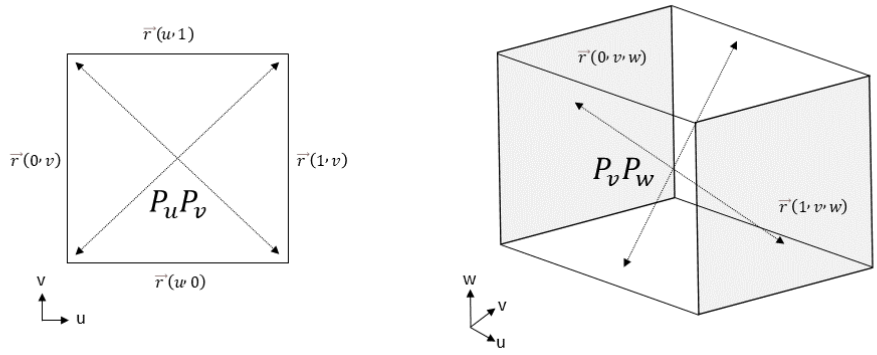


Figure 3.5: Tensor product interpolation using projectors.

$$
\begin{aligned}
P_uP_v &= L_0(u)\left[L_0(v)\overrightarrow{r}(0,0,w) + L_1(v)\overrightarrow{r}(0,1,w)\right] + \\
&\quad L_1(u)\left[L_0(v)\overrightarrow{r}(1,0,w) + L_1(v)\overrightarrow{r}(1,1,w)\right] \\
P_uP_w &= L_0(u)\left[L_0(w)\overrightarrow{r}(0,v,0) + L_1(w)\overrightarrow{r}(0,v,1)\right] + \\
&\quad L(u)\left[L_0(w)\overrightarrow{r}(1,v,0) + L_1(w)\overrightarrow{r}(1,v,1)\right] \\
P_vP_w &= L_0(v)\left[L_0(w)\overrightarrow{r}(u,0,0) + L_1(w)\overrightarrow{r}(u,1,1)\right] + \\
&\quad L(v)\left[L_0(w)\overrightarrow{r}(u,0,0) + L_1(w)\overrightarrow{r}(u,1,1)\right]
\end{aligned} \tag{3.39}
$$

Writing P_uP_v in the matrix form:

$$
P_uP_v = \begin{bmatrix} L_0(u) & L_1(u) \end{bmatrix} \begin{bmatrix} \overrightarrow{r}(0,0,w) & \overrightarrow{r}(0,1,w) \\ \overrightarrow{r}(1,0,w) & \overrightarrow{r}(1,1,w) \end{bmatrix} \begin{bmatrix} L_0(v) \\ L_1(v) \end{bmatrix} \tag{3.40}
$$

The tensor product projector using cubic Hermite functons (bicubic interpolation) in matrix form is given as:

$$
\begin{aligned}
P_uP_v = &\begin{bmatrix} H_0^0(u) & H_0^1(u) & H_1^1(u) & H_1^0(u) \end{bmatrix} \\
&\begin{bmatrix} \overrightarrow{r}(0,0,w) & \overrightarrow{r}_v(0,0,w) & \overrightarrow{r}_v(0,1,w) & \overrightarrow{r}(0,1,w) \\ \overrightarrow{r}_u(0,0,w) & \overrightarrow{r}_{uv}(0,0,w) & \overrightarrow{r}_{uv}(0,1,w) & \overrightarrow{r}_u(0,1,w) \\ \overrightarrow{r}_u(1,0,w) & \overrightarrow{r}_{uv}(1,0,w) & \overrightarrow{r}_{uv}(1,1,w) & \overrightarrow{r}_u(1,1,w) \\ \overrightarrow{r}(1,0,w) & \overrightarrow{r}_v(1,0,w) & \overrightarrow{r}_v(1,1,w) & \overrightarrow{r}(1,1,w) \end{bmatrix} \begin{bmatrix} H_0^0(v) \\ H_0^1(v) \\ H_1^1(v) \\ H_1^0(v) \end{bmatrix}
\end{aligned} \tag{3.41}
$$

Interpolation in other directions can be done by changing the indices defined in Eq. 3.40 and Eq. 3.41.

3.3.4.2 Four Boundary Interpolation

The four boundary interpolation can be shown via the Boolean sum of projectors in w, v, u directions, respectively, as follows:

$$\begin{aligned}\overrightarrow{r}(u,v,w) &= P_u \oplus P_v = P_u + P_v - P_u P_v \\ \overrightarrow{r}(u,v,w) &= P_u \oplus P_w = P_u + P_w - P_u P_w \\ \overrightarrow{r}(u,v,w) &= P_v \oplus P_w = P_v + P_w - P_v P_w \end{aligned} \tag{3.42}$$

Projectors are described by Eq. 3.36 and Eq. 3.37 including tensor products of projectors can be calculated by Eq. 3.40 and Eq. 3.41. The Boolean sum of projectors interpolates in two directions from the specified four boundary surfaces to obtain internal grid point coordinates.

3.3.4.3 Six Boundary Interpolation

Interpolation from all six boundary surfaces creates 3D block grids, as shown in Fig. 3.6. The six boundary interpolation can be defined with the Boolean sum of three projectors as follows:

$$\begin{aligned}\overrightarrow{r}(u,v,w) &= P_u \oplus P_v \oplus P_w \\ &= P_u + P_v + P_w - P_u P_v - P_u P_w - P_v P_w + P_u P_v P_w \end{aligned} \tag{3.43}$$

Projectors and double products can be calculated as in previous sections. The triple product, $P_u P_v P_w$, which interpolates grid points from eight corners of a 3D block, can be expressed as:

$$P_u P_v P_w = P_u(P_v(P_w(\overrightarrow{r}))) \tag{3.44}$$

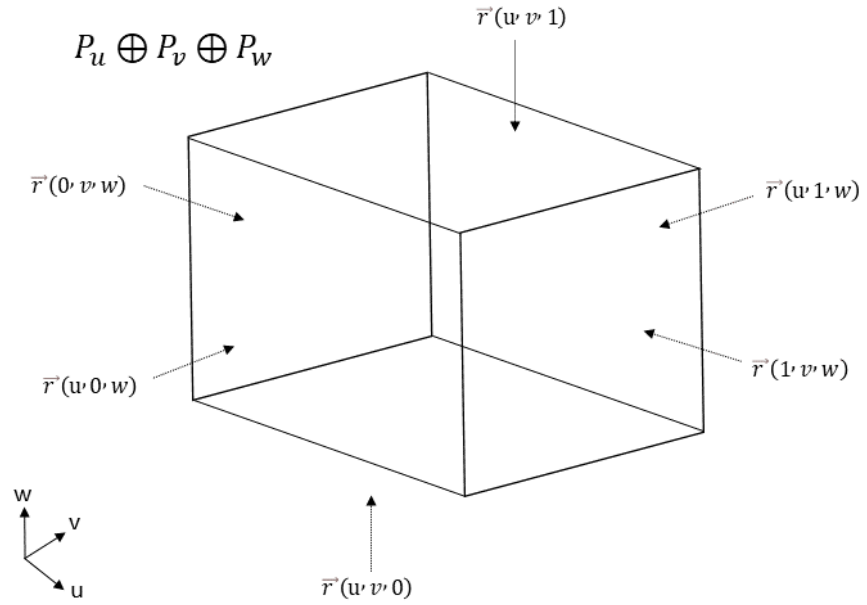


Figure 3.6: Six boundary interpolation with all block faces.

Then, linear triple product can be calculated

$$\begin{aligned} P_u P_v P_w = L_0(u)\,\{L_0(v)[L_0(w)\overrightarrow{r}(0,0,0) + L_1(w)\overrightarrow{r}(0,0,1)] + \\ L_1(v)[L_0(w)\overrightarrow{r}(0,1,0) + L_1(w)\overrightarrow{r}(0,1,1)]\} + \\ L_1(u)\,\{L_0(v)[L_0(w)\overrightarrow{r}(1,0,0) + L_1(w)\overrightarrow{r}(1,0,1)] + \\ L_1(v)[L_0(w)\overrightarrow{r}(1,1,0) + L_1(w)\overrightarrow{r}(1,1,1)]\} \end{aligned} \tag{3.45}$$

The cubic triple product, which involves the first, second, and third derivatives of $\overrightarrow{r}$, can be given as:

$$P_uP_vP_w = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1}[H_i^0(u)H_j^0(v)H_k^0(w)\overrightarrow{r}(i,j,k)+$$
$$H_i^1(u)H_j^0(v)H_k^0(w)\overrightarrow{r}_u(i,j,k) + H_i^0(u)H_j^1(v)H_k^0(w)\overrightarrow{r}_v(i,j,k)+$$
$$H_i^0(u)H_j^0(v)H_k^1(w)\overrightarrow{r}_w(i,j,k) + H_i^1(u)H_j^1(v)H_k^0(w)\overrightarrow{r}_{uv}(i,j,k)+$$
$$H_i^1(u)H_j^0(v)H_k^1(w)\overrightarrow{r}_{uw}(i,j,k) + H_i^0(u)H_j^1(v)H_k^1(w)\overrightarrow{r}_{vw}(i,j,k)+$$
$$H_i^1(u)H_j^1(v)H_k^1(w)\overrightarrow{r}_{uvw}(i,j,k)] \quad (3.46)$$

Assuming zero cross derivatives at the corners (C^1continuity), $\overrightarrow{r}_{uv}$, $\overrightarrow{r}_{uw}$, $\overrightarrow{r}_{vw}$, $\overrightarrow{r}_{uvw}$, provides a reasonable simplification to interpolation. Various blending functions such as Lagrange, Hermite, B-spline, and NURBS can be combined in different directions.

Another method for six boundary interpolation is the recursive algorithm defined as:

$$\begin{aligned}
\overrightarrow{r}_0(u,v,w) &= P_u(\overrightarrow{r}) \\
\overrightarrow{r}_1(u,v,w) &= P_v(\overrightarrow{r} - \overrightarrow{r}_0) \\
\overrightarrow{r}_2(u,v,w) &= P_w(\overrightarrow{r} - \overrightarrow{r}_0 - \overrightarrow{r}_1) \\
\overrightarrow{r}(u,v,w) &= \overrightarrow{r}_0 + \overrightarrow{r}_1 + \overrightarrow{r}_2
\end{aligned} \quad (3.47)$$

3.4 Rotor Grid Template

Automatic grid generation procedure for helicopter rotors is achieved by a template-based approach. An arbitrary four-blade rotor model is used to show the grid generation steps.

3.4.1 Surface Grid by Mapping

Surface grid generation for complex geometries can be obtained by the mapping process. The boundary segments can be copied from the given geometry model, point distribution on this segment can then be changed by spacing functions. Original surface data, consisting of a set of data points, is converted to a wireframe model. Grid

point coordinates can be calculated through bilinear interpolation in each cell of the wireframe model. These steps are demonstrated in Fig. 3.7. The parametric coordinates are searched based on data points then bilinear interpolation can be applied to four corner points within the cell to find interior grid coordinates as follows:

$$P_u P_v = L_0(u)[L_0(v)\overrightarrow{r}(0,0) + L_1(v)\overrightarrow{r}(0,1)]+ \\ L_1(u)[L_0(v)\overrightarrow{r}(1,0) + L_1(v)\overrightarrow{r}(1,1)] \quad (3.48)$$

Surface grids obtanied for a rotor blade is shown in Fig. 3.8.

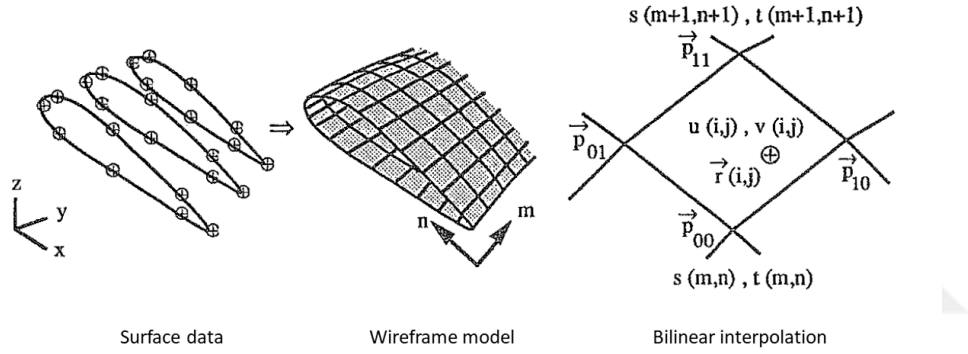


Figure 3.7: Surface grid generation by mapping [22].

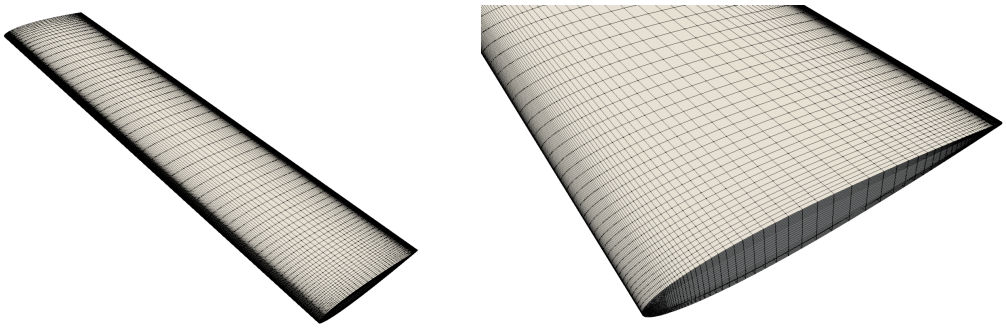


Figure 3.8: Surface grids for a rotor blade

The developed code currently supports only flat blade tips. The structured mesh at the blade tip template using butterfly topology is given in Fig. 3.9.

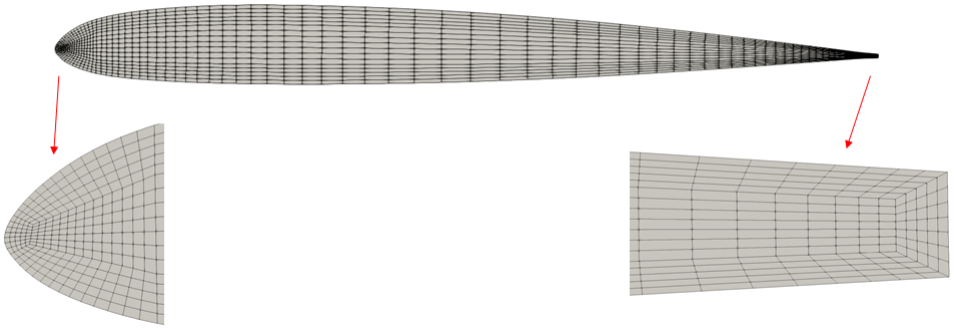


Figure 3.9: Structured grid at the blade tip.

3.4.2 Block Grids

O-type grid topology is applied around the rotor blade, then the computational domain is extended with H-type blocks. Boundary edges of O-type grid is divided into six segments to improve spacing control and orthogonality. Segments are divided as five percent of the chord from the leading and trailing edges. All grid dimensions (N) and spacing (S) inputs are named as shown in Fig. 3.10. Subscripts, l, t, p, s, denote leading, trailing, pressure and suction regions, respectively. Grid dimensions can be modified within the mesh parameters input as given in Table 3.2. Collective and precone angles may be also changed with the mesh parameters input. The following condition is satisfied for grid dimensions as:

$$N_l = N_{t,1} + N_{t,2} + N_{t,3} \tag{3.49}$$

An automatic grid spacing algorithm using the hyperbolic sine, hyperbolic tangent, and geometric function given in the Section 3.3.3.3 can be activated through mesh input file. An additional input file can be used to change individual spacing at each end point of divided segments. There is no spacing control for the trailing edge region (five percent of the chord), yet, it can be controlled through mesh dimensions with an automatic grid point distribution with geometric function.

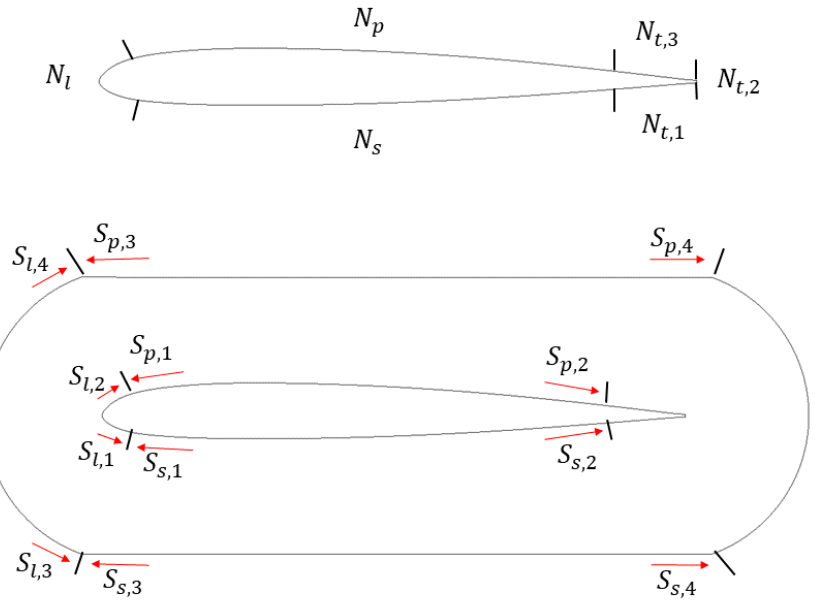


Figure 3.10: Dimensions and spacing.

Table 3.2: Mesh Parameters

Collective pitch angle	θ_c
Precone angle	β_0
Dimension at the trailing edge	$N_{t,2}$
Dimensions near the trailing edge	$N_{t,1}$, $N_{t,3}$
Dimensions at the middle segment	N_p, N_s
Spanwise dimensions (i, for panels)	$N_{span,i}$
Solver type (Euler = 0, N-S = 1)	1
Boundary layer number	15
Growth rate of layers	1.2
First cell height	1E-6
Scale factor for blocks (win tip to farfield)	3
Far-field distance (top)	6
Far-field distance (bottom)	6
Far-field distance (sides)	6
Automatic clustering (auto = 1, by user = 2)	1

High aspect ratio cell can be adjusted with boundary layer number, growth rate ,and first cell height parameters for Navier-Stokes solvers. Scale factor expand H-type blocks that fills the entire domain up to farfield. O-type grid around the blade with its conic extension at the blade tips is shown in Fig. 3.11.

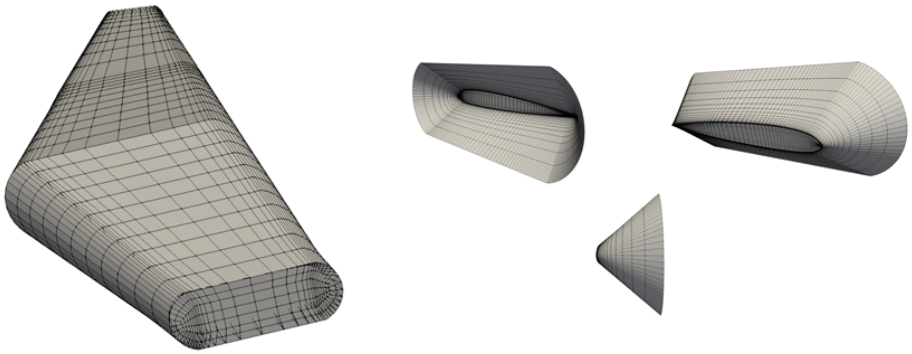


Figure 3.11: O-type grid around the blade with conic tip extension.

A plane view of the generated grid template is given in Fig. 3.12. The outline of the blocks in the template is demonstrated in Fig 3.13. H-type blocks with circular edges are created between successive blades.

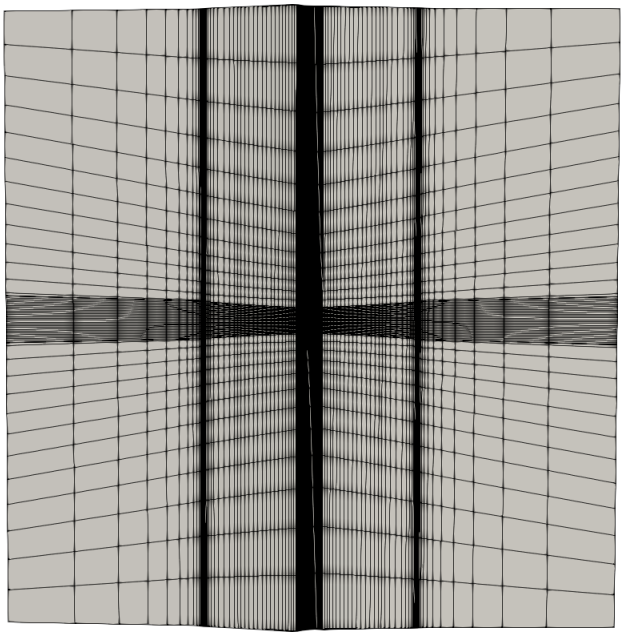


Figure 3.12: Plane view of the grid template.

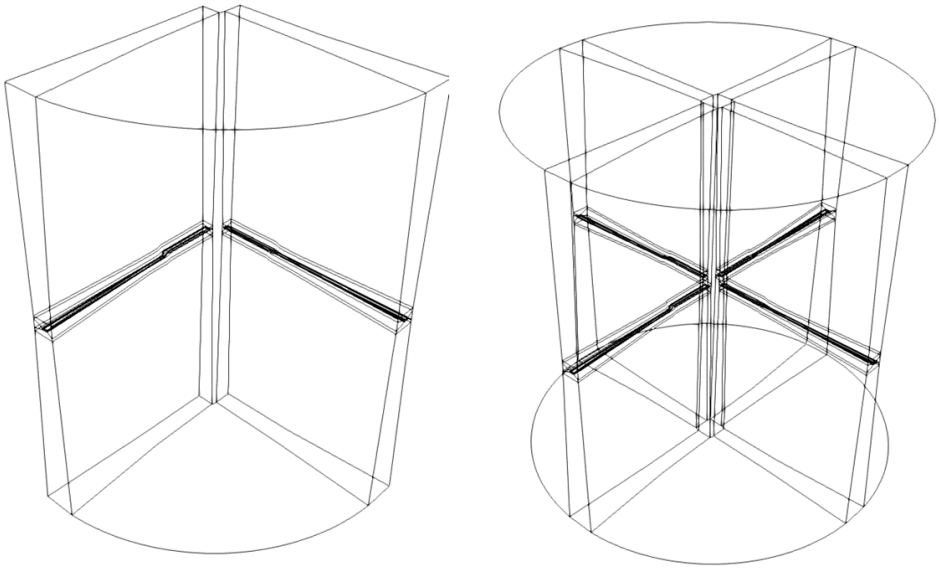


Figure 3.13: Generated grid template.

3.5 The Multiblock Assembly

General properties of the generated template has been explained. Now, the multiblock assembly process will be described step by step. In the first stage, the surface obtained from geometric modeling is mapped and the surface mesh shown in Fig. 3.14 is created.

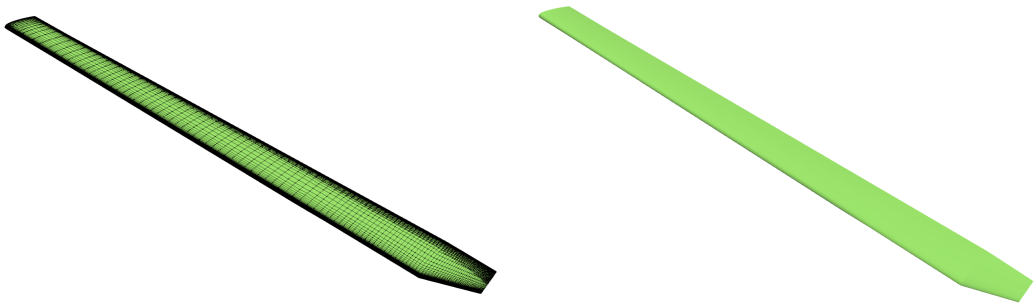


Figure 3.14: Surface grid

After the surface mesh is generated, the conformal block surrounding the rotor blade is obtained, as indicated in Fig. 3.15, using the O-type topology in the chordwise direction and the H-type topology in the spanwise direction.

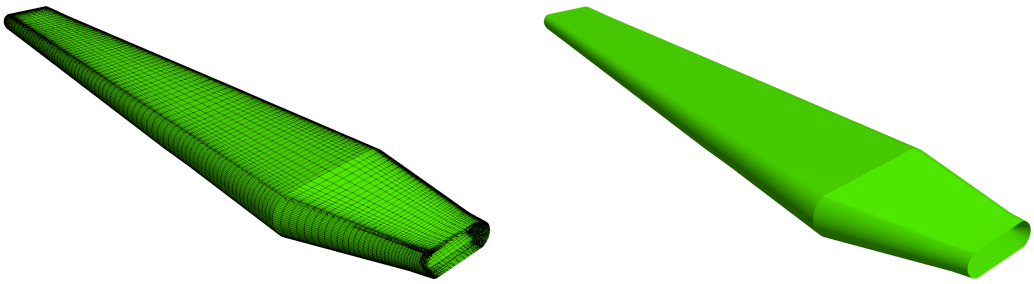


Figure 3.15: Block around the rotor blade

At the tip of the blade, two cone-shaped blocks are formed by using the butterfly topology (Fig. 3.16). Thus, the boundary layer volumes at the blade ends are expanded in the shortest possible distance without deteriorating the mesh quality. Conical expanding volumes follow a line of approximately 45 degrees with respect to the tip of the blade.

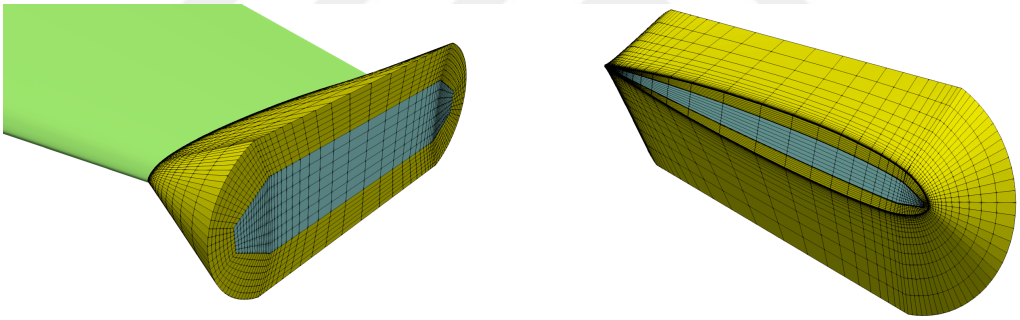


Figure 3.16: Butterfly grid at the blade tips

The block groups that completely cover the blade surface, the O-type block surrounding the blade and two conical blocks used at the blade ends, are shown in Fig. 3.17. The first cell height, growth ratio and layer number are specified by the user so grids sizes can be controlled in the turbulent region. The model is tested for very small y+ numbers and the mesh generator is proved to be robust for high aspect ratio cells.

In the next step, the computational volume is expanded to the far-field from the blade tip using two blocks. With the scale factor specified in the panel parameters, the volume of the blocks can be enlarged linearly up to the far-field. As can be seen in

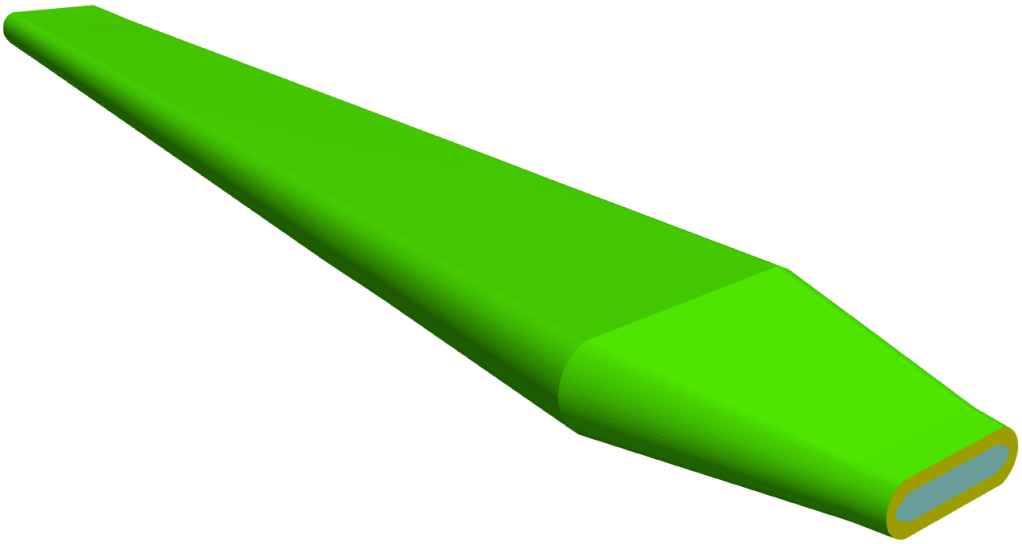


Figure 3.17: Group of blocks around the blade.

Fig. 3.18, the grid density is adjusted automatically so that no user intervention is required.

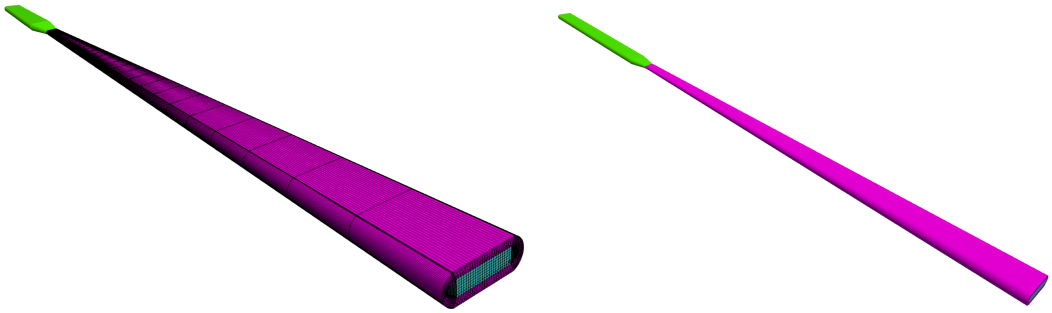


Figure 3.18: Block structure from blade tip to far-field

Then, four H-type blocks were used around the O-type block. The blocks created at this stage are designed for mesh deformation operations to represent the rotor dynamics effects in the forward flight. This developed H-O-H topology also allows the utilization of structured meshes between successive rotor blades. As can be seen from Fig. 3.19, grid densities are provided automatically and no user input is required. The scale factor specified in the mesh parameters is also applied on these blocks. In addition, the far field distance of the blocks in the radial direction is also specified in the

mesh inputs.

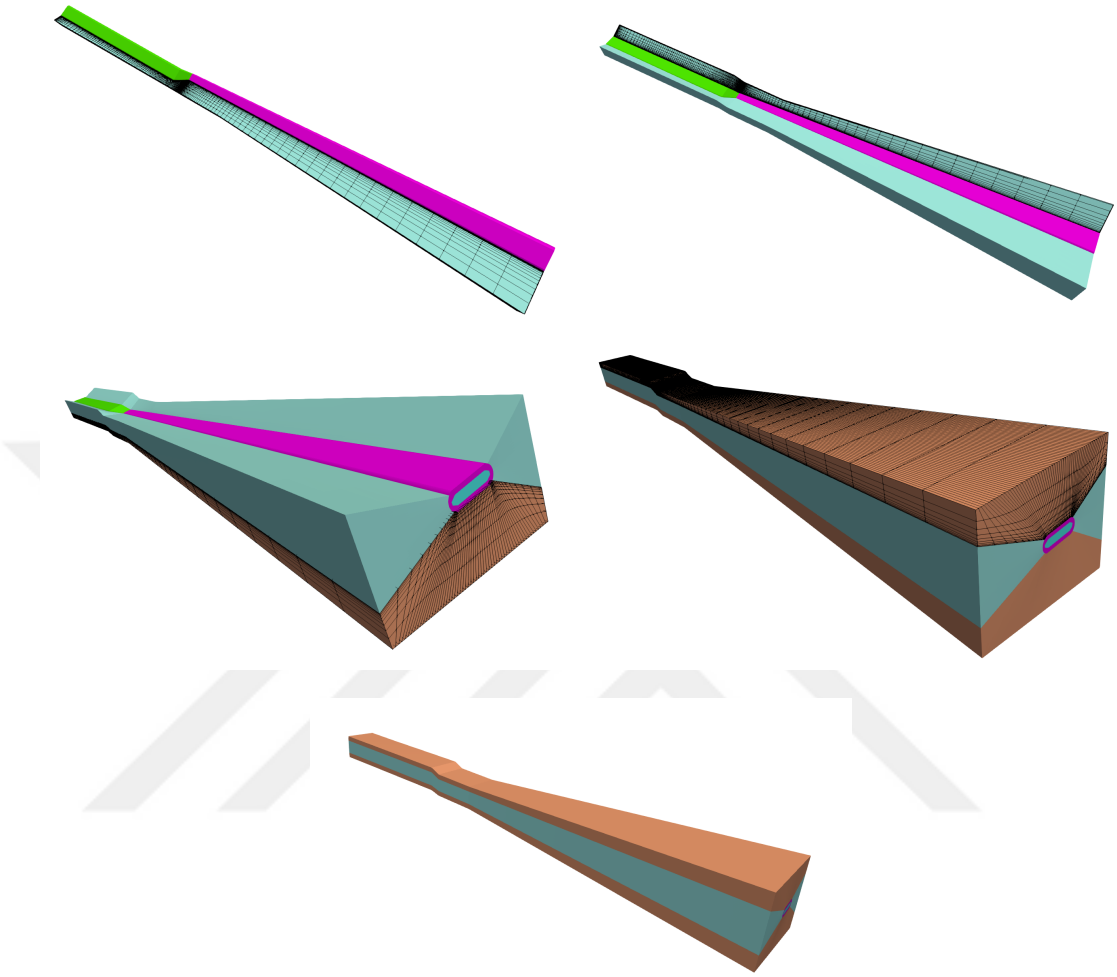


Figure 3.19: H-O-H topology around the blade

Next, the computational volume is expanded using H-type blocks to the lower and upper surfaces of the cylindrical far-field. Distances between the lower-upper surfaces and the center of rotation are specified in terms of rotor radius in the mesh parameters input. The value given in the scale parameter is also applied for these blocks. Grid densities are adjusted automatically. At the end of this stage, the mesh extending from the lower and upper surfaces of the blade to the far-field is obtained, as seen in Fig. 3.20.

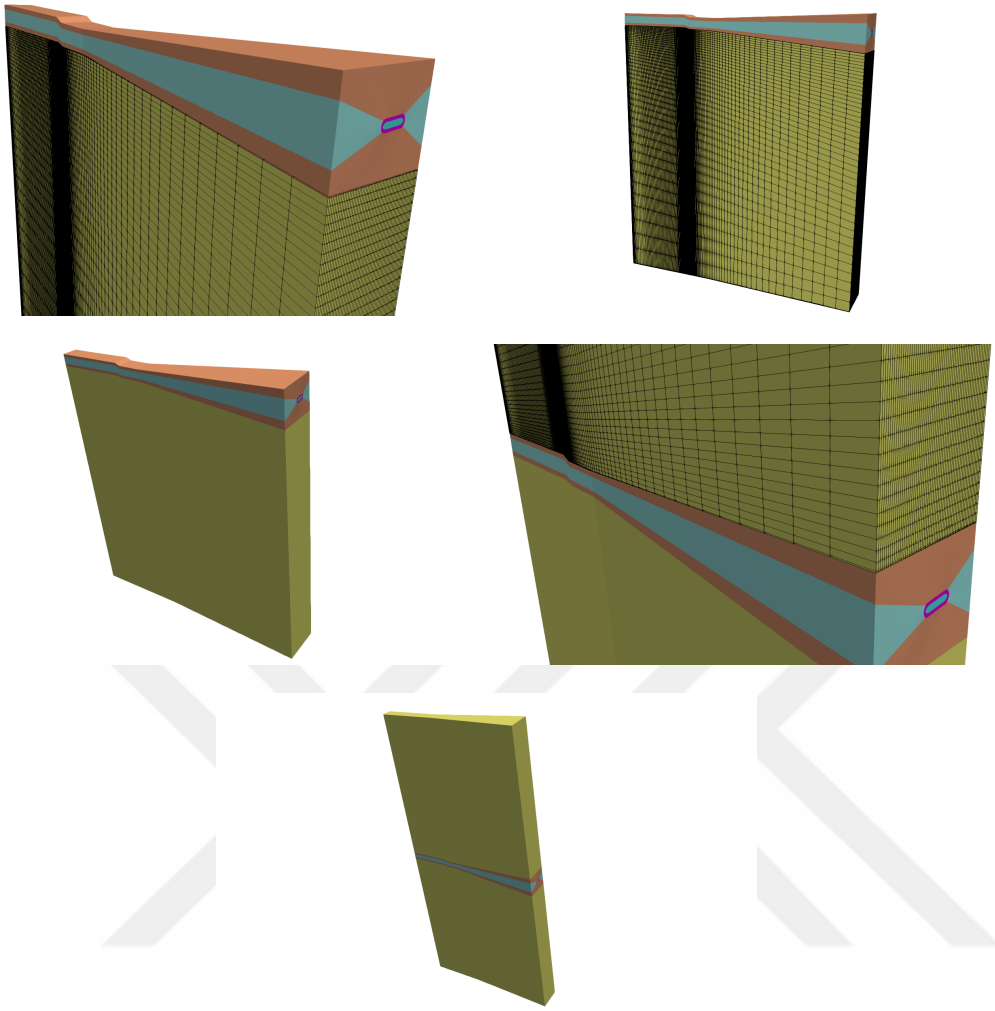


Figure 3.20: Blocks from blade surface to far-field

All the aforementioned steps are applied simultaneously for the other blades and H-type cylindrical blocks are formed by using arc-shaped edges between the blocks extended to the far-field for each blade (Fig. 3.21).

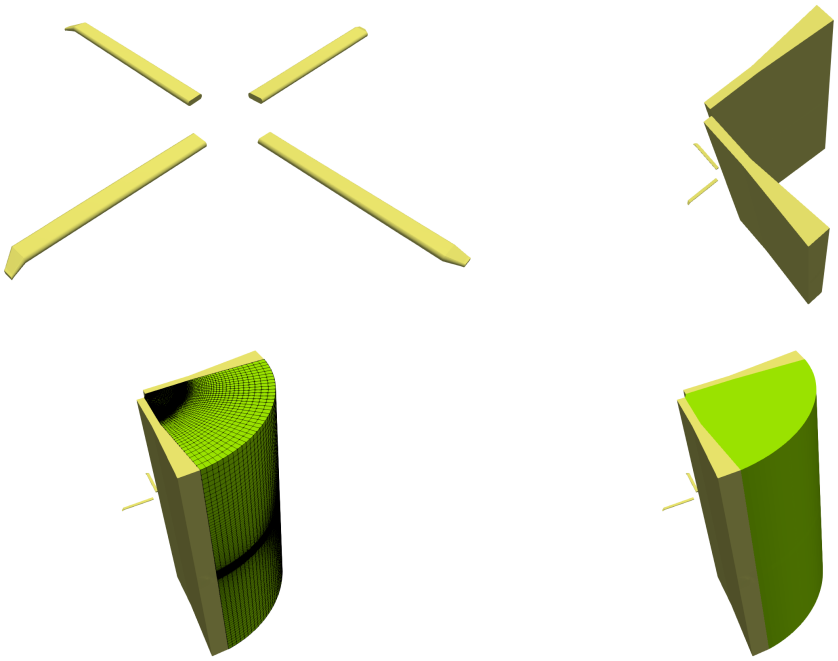


Figure 3.21: Cylindrical blocks between successive blades

After all the steps, the automatic mesh template shown in Fig. 3.22 is obtained.

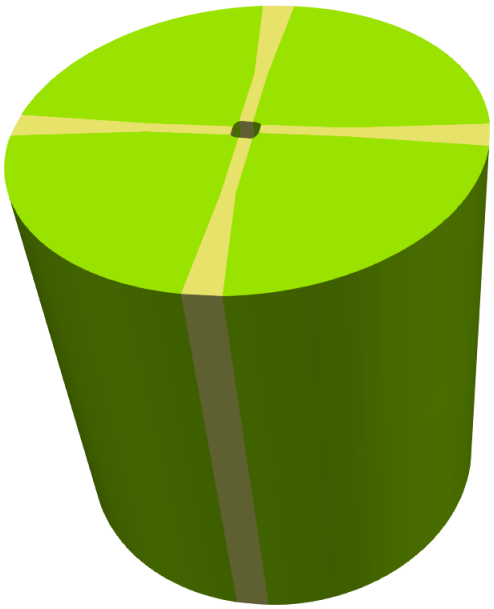


Figure 3.22: The template.

3.6 Mesh Quality Metrics

The mesh quality metrics for a very coarse mesh are given in Table 3.3. Grids with better quality can be achieved by increasing node numbers in the chordwise and spanwise direction (5.1, 5.2).

Table 3.3: Mesh Quality Metrics

Total cells	560 K
First Cell Height	0.00001
Growth Ratio	1.2
Boundar layers	12
Max. aspect ratio	96700
Non-orthogonal cells ($> 70°$)	800
Max. skewness	2.32

CHAPTER 4

ANALYSIS METHODOLOGY

Computational aerodynamic analyses of helicopter main rotors in both hover and forward flight can be performed with several commercial and open-source software. In this thesis, the open-source SU2 code, an open source CFD analysis and optimization software, is utilized. The flow simulations are conducted to calculate.. rotor performance in hover. The SU2 suite can solve both Euler and Navier-Stokes equations.

4.1 Governing Equations

The governing equations of fluid mechanics consist of continuity, momentum, and energy equations. Collecting them into one system of equations, 3D compressible unsteady RANS, Favre-averaged Navier-Stokes, are given in cartesian coordinates as follows:

$$\frac{\partial \overrightarrow{Q}}{\partial t} + \frac{\partial(\overrightarrow{F_c} - \overrightarrow{F_v})}{\partial x} + \frac{\partial(\overrightarrow{G_c} - \overrightarrow{G_v})}{\partial y} + \frac{\partial(\overrightarrow{H_c} - \overrightarrow{H_v})}{\partial z} = S \tag{4.1}$$

where Q represents conservative flow quantities, $\overrightarrow{F_c}, \overrightarrow{G_c}, \overrightarrow{H_c}$ are convective flux vectors, $\overrightarrow{F_v}, \overrightarrow{G_v}, \overrightarrow{H_v}$ are viscous flux vectors, and S is the source term.

The conservative flow variables and and flux vectors can be defined as:

$$\overrightarrow{Q} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}\tilde{u} \\ \bar{\rho}\tilde{v} \\ \bar{\rho}\tilde{w} \\ \bar{\rho}\tilde{e_t} \end{bmatrix}$$

$$\overrightarrow{F_c} = \begin{bmatrix} \bar{\rho}\tilde{u} \\ \bar{\rho}\tilde{u}^2 + \bar{p} \\ \bar{\rho}\tilde{u}\tilde{v} \\ \bar{\rho}\tilde{u}\tilde{w} \\ (\bar{\rho}\tilde{e_t} + \bar{p})\tilde{u} \end{bmatrix}, \quad \overrightarrow{G_c} = \begin{bmatrix} \bar{\rho}\tilde{v} \\ \bar{\rho}\tilde{v}\tilde{u} \\ \bar{\rho}\tilde{v}^2 + \bar{p} \\ \bar{\rho}\tilde{v}\tilde{w} \\ (\bar{\rho}\tilde{e_t} + \bar{p})\tilde{v} \end{bmatrix}, \quad \overrightarrow{H_c} = \begin{bmatrix} \bar{\rho}\tilde{w} \\ \bar{\rho}\tilde{w}\tilde{u} \\ \bar{\rho}\tilde{w}\tilde{v} \\ \bar{\rho}\tilde{w}^2 + \bar{p} \\ (\bar{\rho}\tilde{e_t} + \bar{p})\tilde{w} \end{bmatrix}$$

$$\overrightarrow{F_v} = \begin{bmatrix} 0 \\ \bar{\tau}_{xx} \\ \bar{\tau}_{xy} \\ \bar{\tau}_{xz} \\ \bar{\tau}_{xx}\tilde{u} + \bar{\tau}_{xy}\tilde{v} + \bar{\tau}_{xz}\tilde{w} - \dot{\tilde{q}}_x \end{bmatrix}$$

$$\overrightarrow{G_v} = \begin{bmatrix} 0 \\ \bar{\tau}_{yx} \\ \bar{\tau}_{yy} \\ \bar{\tau}_{yz} \\ \bar{\tau}_{yx}\tilde{u} + \bar{\tau}_{yy}\tilde{v} + \bar{\tau}_{yz}\tilde{w} - \dot{\tilde{q}}_y \end{bmatrix}$$

$$\overrightarrow{H_v} = \begin{bmatrix} 0 \\ \bar{\tau}_{zx} \\ \bar{\tau}_{zy} \\ \bar{\tau}_{zz} \\ \bar{\tau}_{zx}\tilde{u} + \bar{\tau}_{zy}\tilde{v} + \bar{\tau}_{zz}\tilde{w} - \dot{\tilde{q}}_z \end{bmatrix} \tag{4.2}$$

where $\tilde{e}_t$ is the total energy, $\bar{\rho}$ is the density, p is the pressure and u, v, w are the components of the velocity vector, $\overrightarrow{V} = \begin{bmatrix} \tilde{u} & \tilde{v} & \tilde{w} \end{bmatrix}$, in the cartesian coordinates.

The overbar indicates a time-averaged (Reynolds-averaging) flow variable. Let ϕ be any flow variable, then Reynolds-averaging can be defined as:

$$\bar{\phi} = \frac{1}{T}\int_{T} \phi dt \tag{4.3}$$

The tilde indicates a density-weighted (Favre-averaging) flow variable, defined as follows:

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}} \tag{4.4}$$

where T denotes a time interval.

Shear stresses in the viscous flux vector can be written as:

$$\bar{\tau}_{xx} = 2\mu\frac{\partial\tilde{u}}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial\tilde{u}}{\partial x} + \frac{\partial\tilde{v}}{\partial y} + \frac{\partial\tilde{w}}{\partial z}\right)$$

$$\bar{\tau}_{yy} = 2\mu\frac{\partial\tilde{v}}{\partial y} - \frac{2}{3}\mu\left(\frac{\partial\tilde{u}}{\partial x} + \frac{\partial\tilde{v}}{\partial y} + \frac{\partial\tilde{w}}{\partial z}\right)$$

$$\bar{\tau}_{zz} = 2\mu\frac{\partial\tilde{w}}{\partial z} - \frac{2}{3}\mu\left(\frac{\partial\tilde{u}}{\partial x} + \frac{\partial\tilde{v}}{\partial y} + \frac{\partial\tilde{w}}{\partial z}\right)$$

$$\bar{\tau}_{xy} = \bar{\tau}_{yx} = 2\mu\left(\frac{\partial\tilde{u}}{\partial y} + \frac{\partial\tilde{v}}{\partial x}\right)$$

$$\bar{\tau}_{xz} = \bar{\tau}_{zx} = 2\mu\left(\frac{\partial\tilde{u}}{\partial z} + \frac{\partial\tilde{w}}{\partial x}\right)$$

$$\bar{\tau}_{yz} = \bar{\tau}_{zy} = 2\mu\left(\frac{\partial\tilde{w}}{\partial y} + \frac{\partial\tilde{v}}{\partial z}\right) \tag{4.5}$$

where μ represents the dynamic viscosity.

The heat flux terms from Fourier's law can be expressed as:

$$\dot{\bar{q}}_x = -k\frac{\partial\tilde{T}}{\partial x}$$

$$\dot{\tilde{q}}_y = -k\frac{\partial \tilde{T}}{\partial y}$$

$$\dot{\tilde{q}}_z = -k\frac{\partial \tilde{T}}{\partial z} \tag{4.6}$$

where k is the thermal conductivity with the following definition:

$$k = \frac{c_p \mu}{Pr} \tag{4.7}$$

where c_p is the specific heat at constant pressure and Pr is the Prandtl number.

Assuming a perfect gas with gas constant R and a ratio of specific heats γ, temperature can be obtained as:

$$\tilde{T} = \frac{\bar{p}}{\bar{\rho} R} \tag{4.8}$$

The total energy variable, $\tilde{e_t}$, can be calculated as:

$$\tilde{e_t} = \frac{\bar{p}}{\bar{\rho}(\gamma - 1)} + \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) \tag{4.9}$$

The specific heat at constant pressure is written as:

$$c_p = \frac{\gamma R}{\gamma - 1} \tag{4.10}$$

4.2 Turbulence Modeling

According to the standard approach to turbulence modeling, which is based on the eddy viscosity hypothesis of Boussinesq, the impact of turbulence may be expressed as increased viscosity. Thus, the viscosity can be divided into dynamic and turbulent viscosity as follows:

$$\mu = \mu_{dyn} + \mu_{tur} \tag{4.11}$$

An appropriate turbulence model, which includes a new set of variables in addition to the flow variables, can be used to determine turbulent viscosity. The Spalart-Allmaras and the Menter shear stress transport model, which are also included in the SU2 solver, are the most well-known and broadly used models for analyzing rotorcraft.

4.3 Rotating Reference Frame

A helicopter is able to hover, climb and descend vertically in axial flight. During these flight regimes, the flow around the main rotor may be considered as a steady rotation. Transforming the system of governing equations into a reference frame that rotates with a constant angular velocity, the unsteady motion of a rotor can be assumed as steady. This transformation is implemented by Economon et al. [52] into the SU2 solver.

4.4 Boundary Conditions

Any numerical simulation can only examine a portion of the actual physical area. The reduction of the domain results in the creation of artificial boundaries at which specific physical quantities must be prescribed. Moreover, any geometry borders exposed to the flow field are natural boundaries of the physical domain. The numerical implementation of the boundary conditions requires special attention since an incorrect treatment may lead to inaccurate results.

4.4.1 Solid Wall Boundary Condition

The velocity vector is tangent to the surface in the case of an inviscid solid wall, $\overrightarrow{V}.\overrightarrow{n} = 0$. Hence, the convective flux vectors reduce the pressure terms on an inviscid wall boundary. In the case of viscous wall conditions, the relative velocity between the solid geometry and the fluid is assumed to be zero, $\tilde{u} = \tilde{v} = \tilde{w} = 0$.

The convective flux vectors reduce the pressure terms again, and only heat flux terms remains in viscous flux vectors.

4.4.2 Far-field Boundary Condition

The numerical simulation of a helicopter rotor has to be performed within a bounded domain, so an artificial far-field is required. In comparison to the infinite domain, the domain truncation should have no significant effect on the flow field solution. Distances between the center of rotation and boundaries located at the sides, top, and bottom are given as three rotor radii within the code; it can easily be modified in terms of rotor radius through the input file. Furthermore, disturbances from the outside of the domain must not be reflected back into the flow field. Free stream Mach number, static temperature and pressure, and flow direction are defined at the far-field boundary.

CHAPTER 5

VALIDATION AND RESULTS OF THE DEVELOPED TOOL

The ability of the developed code to simulate three-dimensional, inviscid and viscous flows around rotor models in hover regimes is demonstrated through validating the well-known experimental results of S-76 [46, 53] and Caradonna-Tung [54] rotors. Both Euler and RANS computations are performed for different flight conditions with varying collective pitch angles.

The inner cylindrical surface of the computational domain has remained empty without a hub model, and an inviscid wall boundary condition is imposed at this region. A no-slip boundary condition with zero heat-flux (adiabatic wall) is used for the surface of the blade, and a far-field boundary condition is applied to the outer side of the domain.

5.1 The Caradonna-Tung Rotor

Caradonna and Tung conducted an experimental and analytical investigation of a hovering model helicopter rotor. The experimental research included simultaneous blade pressure measurements and tip vortex assessments. Test cases in this study are widely used in the helicopter community to validate CFD codes to solve rotorcraft issues. The rotor model comprises two rectangular, untwisted, and untapered rigid blades with NACA0012 airfoil sections. The rotor aspect ratio, AR , is 6, defined as the ratio of rotor radius to blade chord. The model has a radius, R, of $1.143\,m$ and a chord length, c, of $0.1905\,m$. Various collective pitch angles, θ_c, and tip Mach numbers, M_{tip}, ranging from 0° to 12° and 0.225 to 0.89 are set, respectively. In the experiment, surface pressure distributions were obtained at five rotor blade sec-

tions, $r/R = 0.5, 0.68, 0.80, 0.89$ and 0.96. This study uses three specific test cases of the Caradonna-Tung rotor for validation. For the simulations Jameson-Schmidt-Turkel scheme is used for the convective numerical method. For the viscous part Spalart-Allmaras turbulence model is utilized. The first order upwind discretization is choosen for turbulent numerical method. Weighted least squares is applied for viscous fluxes. For all cases, maximum y^+ values are less than 1, which proves the compatibility of mesh generator to the CFD solvers.

- Test Case 1: The first test case is a transonic flow where the collective pitch angle is $8°$, and the tip Mach number is 0.877. A mesh study is conducted for different cell numbers as given in Table 5.1, and results are shown in Fig. 5.1 for pressure distributions at the normalized radial distance of 0.96. There is no important difference between the medium and fine mesh. The fine mesh requires less than a minute to compute.

Table 5.1: Mesh Study

Mesh	Number of Volumes
Coarse	500 K
Medium	1.4 M
Fine	2.3 M

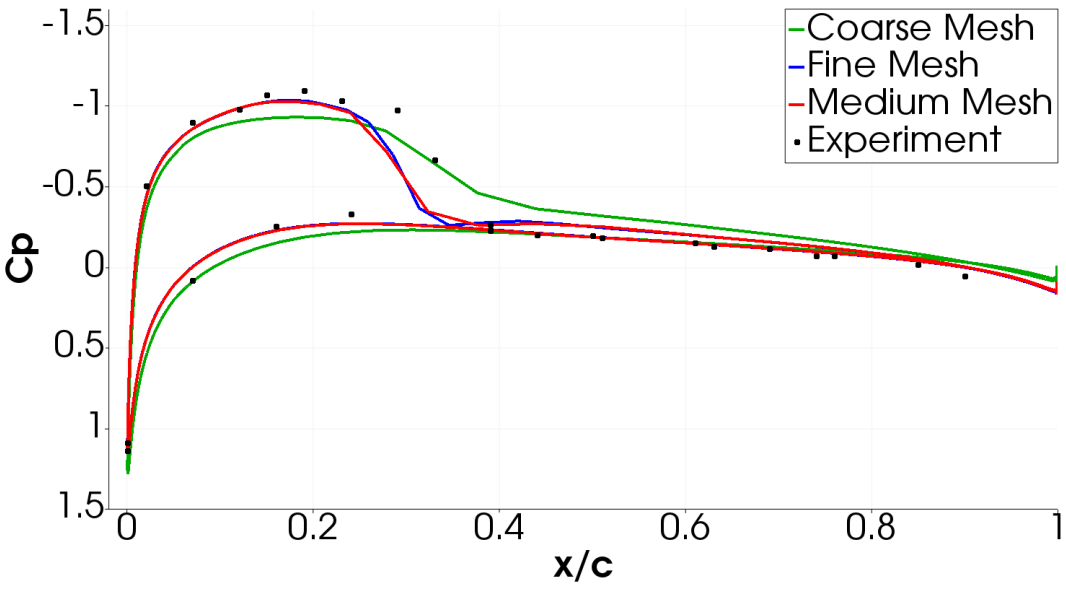
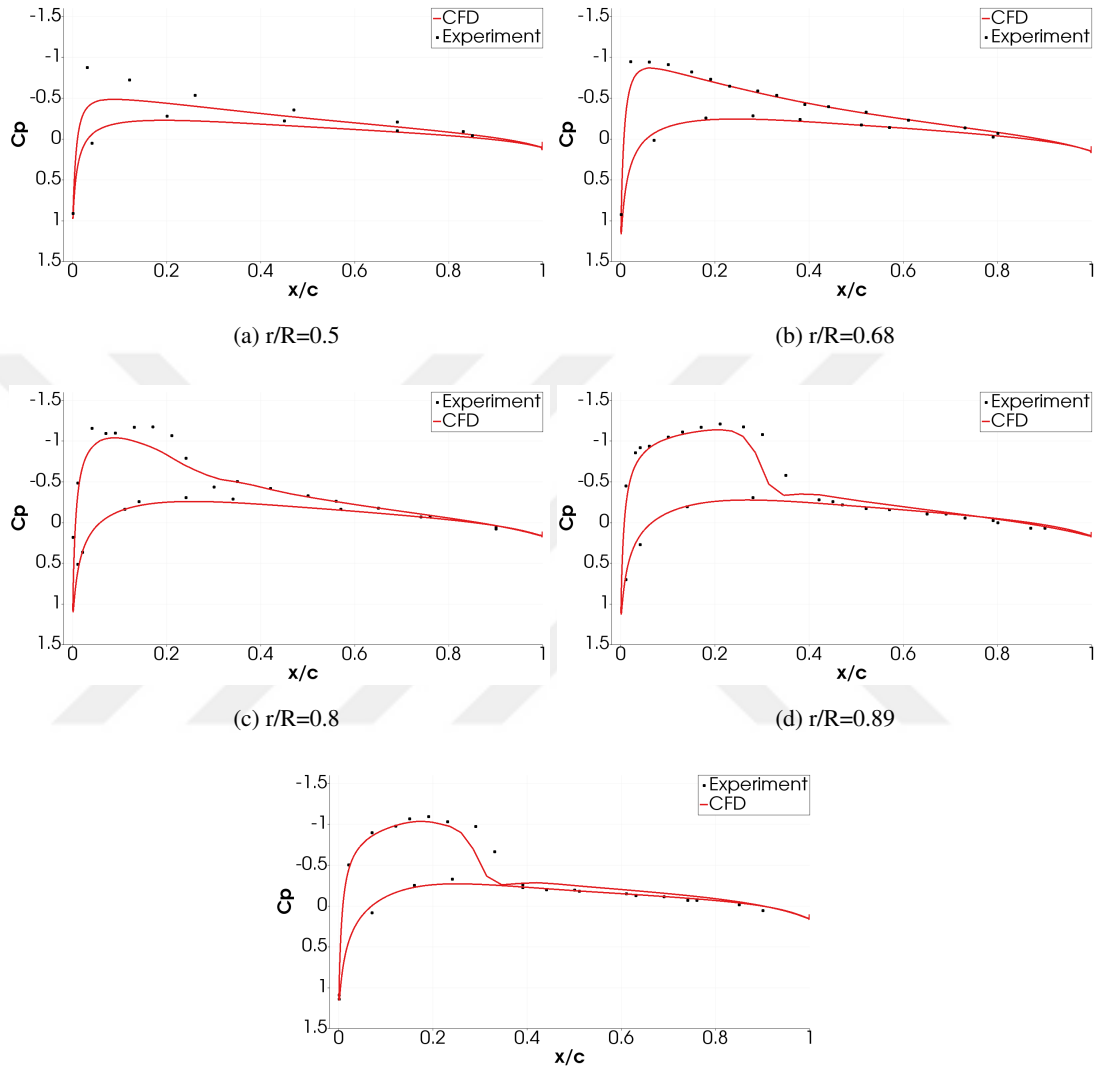


Figure 5.1: Mesh independency study.

Pressure distributions for this test case are given in Fig. 5.2. The flow is entirely subsonc at$r/R = 0.5$ and 0.6, while a shock formation develops on the outer 20% of the blade, $r/R = 0.80$, 0.89, and 0.96. Except for the leading edge acceleration at $r/R = 0.5$, the overall agreement with experimental data is acceptable.



(a) r/R=0.5

(b) r/R=0.68

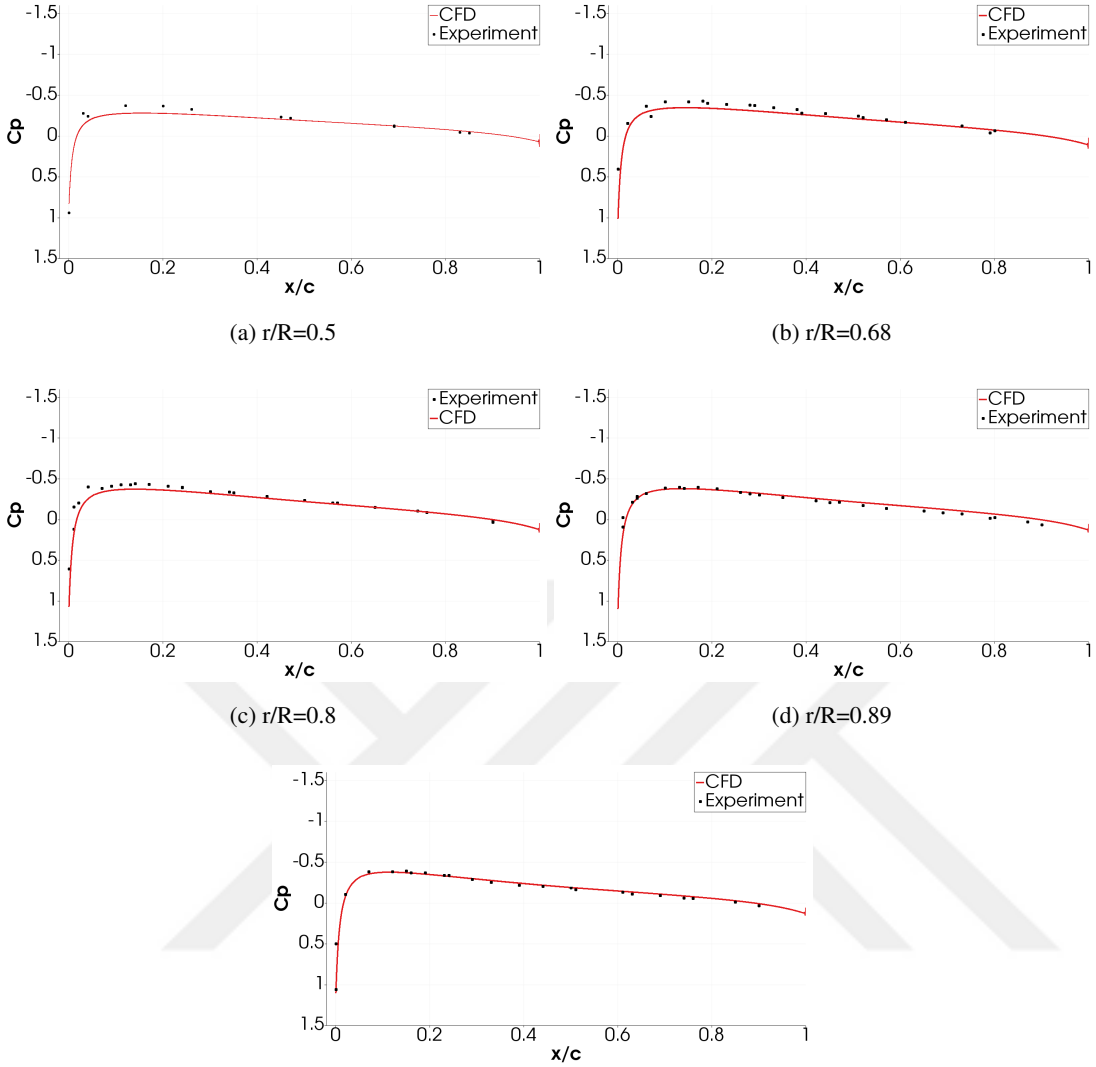
(c) r/R=0.8

(d) r/R=0.89

(e) r/R=0.96

Figure 5.2: C_p distributions for Test Case 1

- Test Case 2: The second test case is non-lifting at zero collective pitch angle with a tip Mach number of 0.52. Pressure coefficient distributions are shown in Fig. 5.3. In this case, the agreement between results from the viscous simulation and the experimental data for both the pressure and suction surface is satisfactory.



(a) r/R=0.5

(b) r/R=0.68

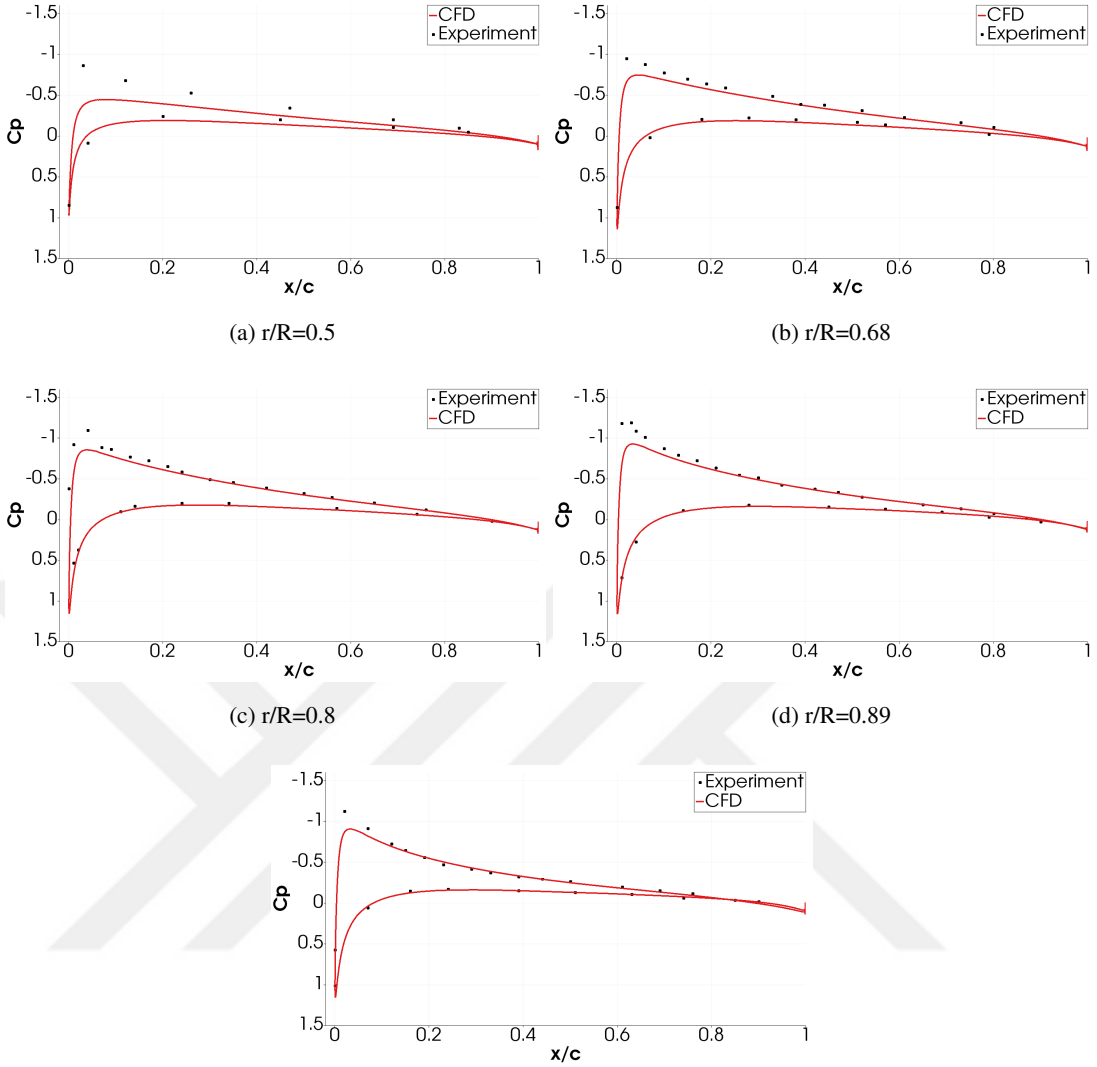
(c) r/R=0.8

(d) r/R=0.89

(e) r/R=0.96

Figure 5.3: C_p distributions for Test Case 2

- Test Case 3: The collective pitch angle is set to 8 degrees with a tip Mach number of 0.439 for the first test case. Pressure coefficient distributions are given in Fig. 5.4. The figures show a good agreement between the experimental and computed results for the pressure surface. Pressure distributions are slightly under-predicted at the leading edge of suction surfaces for all stations except the radial station, $r/R = 0.5$. Increasing grid points in the spanwise direction may give better results for that station. The surface of constant vorticity in Fig. 5.5 shows a tip vortex that is produced at the tip of the blades and follows a helical shape.



(a) r/R=0.5

(b) r/R=0.68

(c) r/R=0.8

(d) r/R=0.89

(e) r/R=0.96

Figure 5.4: C_p distributions for Test Case 3

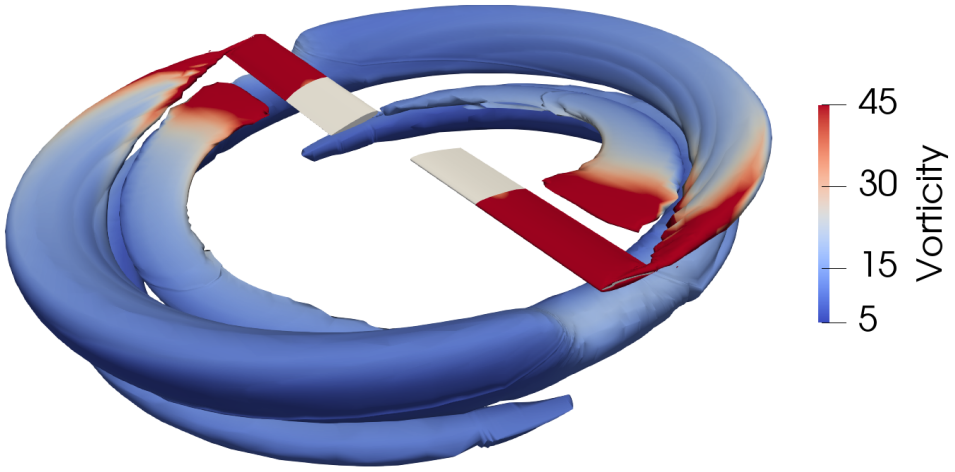


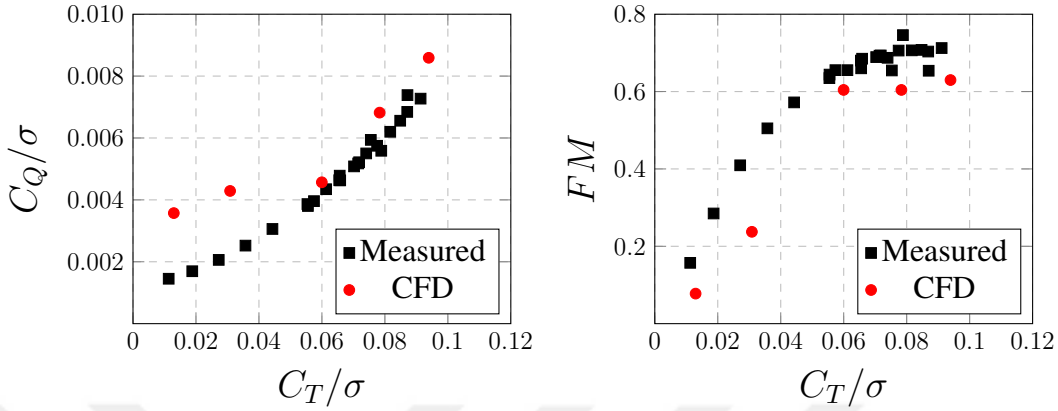
Figure 5.5: Isosurface of vorticity.

5.2 The S-76 Rotor

Hover performance on a $1/4.71$ model-scale Sikorsky S-76 main rotor is predicted for various collective pitch angles at a tip Mach number of 0.65. The rotor model possesses four blades and a linear geometric twist of $-10°$. Blades have been generated using six panels, and zero twist stations are located at the 75% of rotor radius. The blade tip with 60% taper, 35 degrees swept, and 20 degrees anhedral tip is investigated. Collective pitch angles are set to $2°$, $4°$, $6°$, $8°$, and $10°$. For the simulations Jameson-Schmidt-Turkel scheme is used for the convective numerical method. For the viscous part Spalart-Allmaras turbulence model is utilized. The first order upwind discretization is choosen for turbulent numerical method. Weighted least squares is applied for viscous fluxes. The mesh generation process requires less than 2.5 minutes for a mesh size of twelve million cells. Similarly, maximum y^+ values are kept less than 1 for all collective angles, which proves the compatibility of mesh generator to the CFD solvers.

Figure of Merit, FM, and torque coefficients, C_Q/σ, as function of the blade loading coefficient, C_T/σ is presented in Fig 5.6. The measured trends are captured for high collective pitch settings, the results represent an under-predicted Figure of Merit for

the collective angles of $2°$and $4°$, mainly due to the increased artificial dissipation for fast convergence. We are expecting better results with the new high-order solver being developed in the same project.



(a) (b)

Figure 5.6: (a) Torque coefficient versus blade loading coefficient (b) Figure of Merit

CHAPTER 6

CONCLUSION

In this thesis, an automatic high-fidelity aerodynamic analysis tool, which consists of parametric geometry modeling and robust structured multiblock conformal mesh generation, is developed to be included in the conceptual and preliminary design phases of a helicopter.

The parametric geometry modeler within the code requires a set of design inputs that describe a rotor. Complex rotor geometries with any number of blades can be generated through linear or non-linear distribution of specified parameters. Stacks of airfoil sections by combinations of blended profiles, taper ratio, twist, dihedral, and sweep angle distributions are lofted to create desired rotor blades.

High-quality multiblock structured and conformal grid generation code is developed for helicopter rotors. The surface mesh of a rotor blade is realized by mapping using geometry attributes. The rotor blade is surrounded by an O-type grid topology, and the computational domain is subsequently expanded with H-type blocks. A set of inputs, including collective pitch and precone angle, grid dimensions, as well as spacing control for both the O-type block and boundary layer region, far-field distances, can be modified by the user. The resulting mesh has high orthogonality and low skewness. The boundary layer cells comply the solver requirements, where y^+ values and inflation rate are controlled for accurate boundary layer calculations.

The resulting computational meshes are very high quality and they can be exported to various unstructured mesh formats. One possible caveat of the multiblock meshing is the mesh-bleeding problem. The small cells generated near the tip of the blade preserve its size distribution. This results in unnecessarily small cells at the far field

and an increase in the final cell count. This problem will be addressed in the future versions.

The mesh generation is very quick. The entire meshing process, from surface to multi-block mesh generation requires less than 1.5 minutes for a five million volume mesh on a standard processor utilizing single processor core. The rotor design changes and flight controls can be applied immediately and a new, high quality mesh can be generated in similar time compared to commercial software packages. Meshing time changes linearly with the mesh count. Up to 15 million cell count is experimented. Therefore, the new tool achieves its goal as a fast and flexible geometry modeler and mesh generator for helicopter rotors.

The parametric geometry modeler and the multiblock structured mesh generator are unified in the same environment. Therefore, as soon as the geometry parameters are defined, automatic mesh generation occurs instantly due to the specified grid parameters without the need to exchange data.

SU2, an open-source CFD solver and optimization tool developing by a worldwide community, is used to simulate the flow field around the rotor blade. Inviscid and viscous flow solutions on the generated grid template are validated by comparing with the well-known experimental results of Caradonna-Tung and S-76 rotors. The acquired results are quite close to those achieved in the experiments.

Even though the developed tool may now be used for the conceptual and preliminary design of a helicopter rotor, the following enhancements can be made to get better results and simulate other flight regimes.

- An in-house solver code specialized for the multiblock structure grid generator may be developed for faster convergence.
- A high-order numerical scheme or a mesh adaptation method to flow solutions can be implemented to conserve the rotor wake in the simulations.
- An alternative C-type topology around rotor blades may be added to the code for better resolution of the wake behind airfoil sections.
- Mesh bleeding problem should be addressed.

REFERENCES

[1] W. Johnson, *Helicopter Theory*. Princeton, N.J: Princeton University Press, 1980.

[2] J. Leishman, *Principles of Helicopter Aerodynamics*. Cambridge New York: Cambridge University Press, 2006.

[3] F. Caradonna, "Developments and challenges in rotorcraft aerodynamics," in *38th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, jan 2000.

[4] M. Costes, T. Renaud, and B. Rodriguez, "Rotorcraft simulations: a challenge for CFD," *International Journal of Computational Fluid Dynamics*, vol. 26, pp. 383–405, jul 2012.

[5] A. T. Conlisk, "MODERN HELICOPTER AERODYNAMICS," *Annual Review of Fluid Mechanics*, vol. 29, pp. 515–567, jan 1997.

[6] Y. Tanabe and S. Saito, "Significance of all-speed scheme in application to rotorcraft cfd simulations," 10 2009.

[7] R. C. Strawn, F. X. Caradonna, and E. P. N. Duque, "30 years of rotorcraft computational fluid dynamics research and development," *Journal of the American Helicopter Society*, vol. 51, no. 1, p. 5, 2006.

[8] J. Vandenbrande, T. Grandine, and T. Hogan, "The search for the perfect body: Shape control for multidisciplinary design optimization," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, jan 2006.

[9] T. A. Grandine and T. A. Hogan, "A parametric quartic spline interpolant to position, tangent and curvature," *Computing*, vol. 72, pp. 65–78, apr 2004.

[10] A. Hahn, "Vehicle sketch pad: A parametric geometry modeler for conceptual aircraft design," in *48th AIAA Aerospace Sciences Meeting Including the New*

Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics, jan 2010.

[11] J. Gloudemans, P. Davis, and P. Gelhausen, "A rapid geometry modeler for conceptual aircraft," in *34th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, jan 1996.

[12] D. Rodriguez and P. Sturdza, "A rapid geometry engine for preliminary aircraft design," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, jan 2006.

[13] L. Iqbal and J. Sullivan, "Application of an integrated approach to the UAV conceptual design," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, jan 2008.

[14] M. Siggel, J. Kleinert, T. Stollenwerk, and R. Maierl, "TiGL: An open source computational geometry library for parametric aircraft design," *Mathematics in Computer Science*, vol. 13, pp. 367–389, jul 2019.

[15] A. Rizzi, M. Zhang, B. Nagel, D. Boehnke, and P. Saquet, "Towards a unified framework using CPACS for geometry management in aircraft design," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, jan 2012.

[16] B. Nagel, D. Böhnke, V. Gollnick, P. Schmollgruber, A. Rizzi, G. LaRocca, and J. Alonso, "Communication in aircraft design: Can we establish a common language?," *28th International Congress of the Aeronautical Sciences, Brisbane, Australia*, vol. 1, 01 2012.

[17] B. M. Kulfan, "Universal parametric geometry representation method," *Journal of Aircraft*, vol. 45, pp. 142–158, jan 2008.

[18] A. D. Marco, M. D. Stasio, P. D. Vecchia, V. Trifari, and F. Nicolosi, "Automatic modeling of aircraft external geometries for preliminary design workflows," *Aerospace Science and Technology*, vol. 98, p. 105667, mar 2020.

[19] A. De Marco, V. Cusati, V. Trifari, M. Ruocco, F. Nicolosi, and P. Vecchia, "A java toolchain of programs for aircraft design," 10 2017.

[20] Open Cascade, "Open Cascade Technology." https://www.opencascade.com/. Last accessed: 2021-07-26.

[21] F. Sanchez, S. Liscouët-Hanke, and A. Tfaily, "Improving aircraft conceptual design through parametric CAD modellers – a case study for thermal analysis of aircraft systems," *Computers in Industry*, vol. 130, p. 103467, sep 2021.

[22] C. Dener, *Development of an Interactive Grid Generation and Geometry Modeling System with Object Oriented Programming*. PhD thesis, Vrije Universiteit Brussel, 1991.

[23] J. Blazek, "Principles of grid generation," in *Computational Fluid Dynamics: Principles and Applications*, pp. 357–393, Elsevier, 2015.

[24] A. C. Haselbacher, *A Grid-Transparent Numerical Method For Compressible Viscous Flows On Mixed Unstructured Grids*. PhD thesis, Loughborough University, 1994.

[25] H. K. Versteeg, *An introduction to computational fluid dynamics : the finite volume method*. Harlow, England New York: Pearson Education Ltd, 2007.

[26] L. Tysell, "Hybrid grid generation for viscous flow computations around complex geometries," 2009.

[27] A. Shih, S. Gopalsamy, Y. Ito, D. Ross, M. Dillavou, and B. Soni, "Automatic and parametric mesh generation approach," 01 2005.

[28] S. Pandya, W. Chan, and J. Kless, "Automation of structured overset mesh generation for rocket geometries," in *19th AIAA Computational Fluid Dynamics*, American Institute of Aeronautics and Astronautics, jun 2009.

[29] O. Z. Köseömür and O. Baran, "Automatic mesh generator integrated missile design tool development," 09 2019.

[30] S. Shahpar and L. Lapworth, "PADRAM: Parametric design and rapid meshing system for turbomachinery optimisation," in *Volume 6: Turbo Expo 2003, Parts A and B*, ASMEDC, jan 2003.

[31] N. A. Kündeş, "Development of an automatic design and analysis tool foraxial compressors," Master's thesis, Middle East Technical University, 2017.

[32] T. Zhang and G. N. Barakos, "Development of simulation tools for high fidelity analysis of compound rotorcraft," in *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, jan 2020.

[33] Ansys Inc, "ANSYS Meshing." `https://www.ansys.com/products/meshing`. Last accessed: 2021-08-03.

[34] J. Shaw and N. Weatherill, "Automatic topology generation for multiblock grids," *Applied Mathematics and Computation*, vol. 52, pp. 355–388, dec 1992.

[35] M. Ferrari, "Automatic method for multiblock structured grid generation," in *2018 AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, jan 2018.

[36] C. B. Allen, "Towards automatic structured multiblock mesh generation using improved transfinite interpolation," *International Journal for Numerical Methods in Engineering*, vol. 74, no. 5, pp. 697–733, 2008.

[37] P. Doerffer and O. Szulc, "Numerical simulation of model helicopter rotor in hover," *TASK QUARTERLY*, vol. 12, pp. 227–236, 09 2008.

[38] Cadence Design Systems Inc, "OMNIS™/AUTOGRID." `https://www.numeca.com/product/omnis-autogrid`. Last accessed: 2021-08-04.

[39] J. Martinez, P. Doerffer, O. Szulc, and F. Tejero, "Aerodynamic analysis of wind turbine rotor blades," *TASK Quarterly*, vol. 19, pp. 129–140, 2015.

[40] O. Szulc, P. Doerffer, F. Tejero, J. Żółtak, and J. Małecki, "Numerical analysis of high-speed impulsive (hsi) noise of pzl w3-a "sokół" (falcon) helicopter main rotor in forward flight," *TASK Quarterly*, vol. 19, pp. 181–196, 2015.

[41] C. Xiang, X. Yang, B. Xu, H. Han, and L. Liu, "Numerical simulation of unsteady flow past autorotating rotor in gyroplane level flight," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, jun 2015.

[42] F. Lu, Y. Pang, X. Jiang, J. Sun, Y. Huang, Z. Wang, and J. Ju, "Automatic generation of structured multiblock boundary layer mesh for aircrafts," *Advances in Engineering Software*, vol. 115, pp. 297–313, jan 2018.

[43] F. Lu, L. Qi, X. Jiang, G. Liu, Y. Liu, B. Chen, Y. Pang, and X. Hu, "NNW-GridStar: Interactive structured mesh generation software for aircrafts," *Advances in Engineering Software*, vol. 145, p. 102803, jul 2020.

[44] S. L. Karman and N. J. Wyman, "Automatic unstructured mesh generation with geometry attribution," in *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, jan 2019.

[45] Pointwise Inc, "Pointwise." `https://www.pointwise.com/`. Last accessed: 2021-08-03.

[46] D. T. Balch and J. Lombardi, "Experimental study of main rotor tip geometry and tail rotor interactions in hover. volume 1. text and figures," 1985.

[47] D. F. Rogers, *An Introduction to Nurbs: With Historical Perspective*. MORGAN KAUFMANN PUBL INC, July 2000.

[48] W. T. Les Piegl, *The NURBS Book*. Springer Berlin Heidelberg, Nov. 1996.

[49] W. J. Gordon and C. A. Hall, "Construction of curvilinear co-ordinate systems and applications to mesh generation," *International Journal for Numerical Methods in Engineering*, vol. 7, no. 4, pp. 461–477, 1973.

[50] A. Rizzi and L. Eriksson, "Transfinite mesh generation and damped euler equation algorithm for transonic flow around wing-body configurations," in *5th Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, jun 1981.

[51] J. Thompson, *Numerical grid generation : foundations and applications*. New York: North-Holland Elsevier Science Pub. Co. distributor, 1985.

[52] T. D. Economon, F. Palacios, and J. J. Alonso, "A viscous continuous adjoint approach for the design of rotating engineering applications," in *21st AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, jun 2013.

[53] D. T. Balch and J. Lombardi, "Experimental study of main rotor tip geometry and tail rotor interactions in hover. volume 1. run log and tabulated data," 1985.

[54] F. Caradonna and C. Tung, “Experimental and analytical studies of a model helicopter rotor in hover,” 1980.

APPENDIX A

INPUT FILE FOR MODELING

```
//Rotor parameters
1.43                    //Rotor radius
4                       //Blade number
6                       //Total Panel number
0.25                    //Pitching axis
//Panel i parameters
1                       //Panel number (i=1)
5                       //Cross section number
SC1013R8pres            //Pressure curve (at root)
SC1013R8suc             //Suction curve (at root)
SC1095R8pres            //Pressure curve (at tip)
SC1095R8suc             //Suction curve (at tip)
0.189 0.285             //Panel root tip span in terms of rotor radius
1                       //Aero Twist (=1 yes, =0 no)
1                       //=0 linear, =1 non lin
thickness               //File for non lin dist.
13.0 10.74              //Values for lin dist.
1.5                     //Chord Dist (=0 dist, = constant val.)
0                       //=0 linear, =1 non lin
chordDist               //File for non lin dist.
3.12 5                  //Values for lin dist.
35                      //Sweep Dist (=0 dist, = constant val.)
0                       //=0 linear, =1 non lin
sweepDist               //File for non lin dist.
0 5                     //Values for lin dist.
20                      //Dihedral Dist (=0 dist, = constant val.)
0                       //=0 linear, =1 non lin
dihedralDist            //File for non lin dist.
55 65                   //Values for lin dist.
0                       //GeoTwist Dist (=0 dist, = constant val.)
0                       //=0 linear, =1 non lin
geoTwist                //File for non lin dist.
4.01 4.5                //Values for lin dist.
```

APPENDIX B

INPUT FILE FOR MESH GENERATION AND SPACING

B.1 Mesh File

```
8.0                     //Collective pitch angle
0.5                     //Precone angle
20                      //Node number near the trailing edge
10                      //Node number at the trailing edges
55                      //Node number at the middle segment
8                       //Node number for panel i
1                       //Solver type (0 = Euler, 1= NS)
30                      //Boundary layer number
1.2                     //Boundary Layer growth rate
0.000005                //First Cell Height
2                       //Scale factor for H type blocks
1                       //=0 uniform, =1 auto clust, =2 by user
3                       //Far field distance from top
3                       //Far field distance from bottom
3                       //Far field distance from sides
```

B.2 Spacing File

```
0.0232                  // S l ,1
0.0232                  // S l ,2
0.0236                  // S l ,3
0.0187                  // S l ,4
0.0233                  // S p ,1
0.0232                  // S p ,2
0.0321                  // S p ,3
0.0321                  // S p ,4
0.0321                  // S s ,1
0.0321                  // S s ,2
0.0401                  // S s ,3
0.0401                  // S s ,4
```