ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DEVELOPMENT OF ARBITRARY LAGRANGIAN - EULERIAN ALGORITHM FOR COMPRESSIBLE FLOW PROBLEMS

M.Sc. THESIS

Erol AKSOY

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Engineering Programme

JANUARY 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DEVELOPMENT OF ARBITRARY LAGRANGIAN - EULERIAN ALGORITHM FOR COMPRESSIBLE FLOW PROBLEMS

M.Sc. THESIS

Erol AKSOY
(511211121)

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Engineering Programme

Thesis Advisor: Prof. Dr. Mehmet ŞAHİN

JANUARY 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SIKIŞTIRILABİLİR AKIŞLAR İÇİN KEYFİ LAGRANYEN EULERYEN ALGORİTMA GELİŞTİRİLMESİ

YüKSEK LİSANS TEZİ

Erol AKSOY
(511211121)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mehmet ŞAHİN

OCAK 2024

Erol AKSOY, a M.Sc. student of ITU Graduate School student ID 511211121, successfully defended the thesis entitled "DEVELOPMENT OF ARBITRARY LAGRANGIAN - EULERIAN ALGORITHM FOR COMPRESSIBLE FLOW PROBLEMS", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Mehmet ŞAHİN** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. M. Adil YUKSELEN** Istanbul Gelisim University	.....
	Assoc. Prof. Bayram CELIK Istanbul Technical University	.....

Date of Submission : 5 January 2024
Date of Defense : 25 January 2024

To my family,

FOREWORD

As someone who believes in the apprentice-master relationship as an essential aspect of education, I am extremely honored and proud to work with my advisor, Prof. Dr. Mehmet ŞAHİN. During the long hours spent in his office, I have witnessed many situations where I can look up to both his personality and academic expertise. I would like to express my sincere gratitude to him for the support he has provided in both my undergraduate and graduate studies.

I would also like to thank Prof. M. Adil YÜKSELEN, with whom I had the opportunity to work during my undergraduate years. I have learned a lot from him, and I have enjoyed our conversations immensely. I am grateful for the significant place he holds in my life.

I extend my thanks to my esteemed teachers and friends Hasan KARALİ, Yusuf DEMİROĞLU, Şıhmehmet YILDIZ, Utkun Erinç MALKOÇOĞLU and Samet UZUN, whom I had the pleasure of meeting through Prof. YÜKSELEN. They have been supportive in every situation.

I want to express my gratitude to my close friends, Huriye Nur TOPRAK, Talha ATEŞ and the HEMLAB community, who endured my hardships and shared my feelings of anger, despair, and joy throughout this process.

Furthermore, I would like to gratefully acknowledge the financial support from the Turkish National Scientific and Technical Research Council (TÜBİTAK) through the 1001 project under the project number of 122M598. This acknowledgement also extends to the National Center for High Performance Computing of Turkey (UHeM) and TR-Grid for the use of their computer resources.

Finally, I would like to thank my dear family for raising me and providing unwavering support in all circumstances.

January 2024

Erol AKSOY
(Aeronautics Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

AMR	**:** Adaptive Mesh Refinement
ALE	**:** Arbitrary Lagrangian Eulerian
CFD	**:** Computational Fluid Dynamics
DNS	**:** Direct Numerical Simulation
DES	**:** Detached Eddy Simulation
DGCL	**:** Discrete Geometric Conservation Law
FVM	**:** Finite Volume Method
LES	**:** Large Eddy Simulation
N-S	**:** Navier-Stokes Equations

SYMBOLS

$\mathbf{Q}$	: Conservative variables vector
$\mathbf{u}$	: Velocity vector
u, v, w	: Velocity components
u_n	: Normal velocity
$\dot{\mathbf{x}}$	: Grid velocity vector
$\dot{x}, \dot{y}, \dot{z}$	: Grid velocity components
$\dot{x}_n$	: Grid normal velocity
γ	: Specific heat ratio
Re	: Reynolds number
Pr	: Prandtl number
μ	: Dynamic viscosity
μ_t	: Turbulent viscosity
ρ	: Density
T	: Temperature
$\mathbf{F}^c$	: Convective flux vector
$\mathbf{F}^v$	: Viscous flux vector
τ	: Shear stress tensor
$\mathbf{n}$	: Surface normal
$\mathbf{J}$	: Jacobian matrix
Δt_p	: Physical time step
Δt_{CFL}	: Pseudo time step
V	: Volume of the cell
C_T	: Thrust coefficient
C_Q	: Torque coefficient
FM	: Figure of merit

LIST OF FIGURES

Page

DEVELOPMENT OF
ARBITRARY LAGRANGIAN - EULERIAN ALGORITHM
FOR COMPRESSIBLE FLOW PROBLEMS

SUMMARY

With the increasing computer power, CFD have become the main part of the design phase in industry. To design novel geometries and to estimate the aerodynamic forces, RANS equations are widely used to simulate flow around a body. Even there are increasing attempts to coalesce RANS solutions to design optimization. Therefore, robust and accurate RANS solvers are required.

In this study, we extend the capabilities of in-house HEMLAB solver with Arbitrary Lagrangian-Eulerian approach to deal with moving and deforming boundaries. The fluxes due to grid motion is calculated in accordance with the Discrete Geometric Conservation Law. In the scope of this thesis, HEMLAB algorithm is integrated with anisotropic mesh adaptation library pyAMG. The interface between pyAMG and HEMLAB is designed in order for pyAMG to project adapted surface mesh onto original one. By this way, we ensure to preserve the geometry. In addition, residual smoothing is applied. HLLC and Roe scheme is used in the simulations. To enhance the convergence and effectivity of preconditioner, we scaled the equations in order to have eigenvalues lower than unity.

We approved the algorithm on 2D heaving and pitching airfoil cases which is proposed under HiOCFD4 Workshop. In this problem, RANS equations are solved on rigidly moving meshes. Then, S76 main rotor configuration is assessed in hover. Firstly, Euler equations are solved for rotor analysis while applying anisotropic mesh adaptations. Effect of differrent sensor functions are assessed. Entropy and Mach sensor functions are used. Entropy sensor function captures the wake layer, providing an advantage in terms of prediction of drag coefficient. Generally, we can say that the application of mesh refinement is beneficial to simulate complex flows because the refinement of wake and tip vortices are promising. Also, RANS equation is solved on a fix mesh without using Adaptive Mesh Refinement.

SIKIŞTIRILABİLİR AKIŞLAR İÇİN KEYFİ LAGRANYEN EULERYEN ALGORİTMA GELİŞTİRİLMESİ

ÖZET

Endüstrileşme ile başlayan teknoloji hareketi sonucu dünyada teknik ve bilimsel bir çok konuda atılım yaşanmıştır. Bu atılımlar sonucu insanoğlu gözünü semaya dikmiş ve kuşlara özenmiştir. Ancak çok önemli bir problem halen daha önümüzde durmaktadır. Hava araçlarının tasarımı için araç etrafındaki hava hareketinin anlaşılması gerekmektedir. Bu çok düşük hızlarda ve düşük hücum açılarında potansiyel akış denklemleri yardımıyla kolayca halledilebilirken artan teknoloji ile birlikte hava aracının hızının da artmasıyla ve daha ekstrem koşullarda çalışabilecek tasarımlar yapmak amacıyla, incelenmesi gereken hava akış fenomeni de karmaşıklaşmıştır. Daha genel halde Navier-Stokes denklemleri ile yönetilen akışkan fiziğini, bu denklemleri analitik olarak çözmek imkansız olduğundan (en azından şimdiye kadar), anlamak için nümerik yöntemler kullanılmaktadır. Bu yöntemler bütününe Hesaplamalı Akışkanlar Dinamiği (HAD) denilmektedir. Yıllar içerisinde gelişen bilgisayar teknolojisi ile bu yöntemler de yaygınlaşmış ve geliştirilmeye devam edilmiştir.

İlk olarak Euler denklemleri çözülerek başlanılmış ve 20 yıl içerisinde RANS denklemlerinin çözülebilmiştir. Şu anda çok daha yoğun ağ yapıları üzerinde DES, LES ve DNS çözümleri yapabiliyor olsak da RANS denklemleri endüstriyel uygulama alanında halen popülerliğini sürdürmektedir. Bu çalışmada da RANS denklemleri hareketli ağlar üzerinde çözülecektir.

Bu çalışmada daha önceden geliştirilmiş bir algoritma olan HEMLAB çözücüsü, hareketli ağ yapıları üzerinde çalışacak şekilde Keyfi Lagranyen-Euleryen(ALE) yaklaşımla genişletilmiştir. ALE yaklaşımında hareketli sınır yüzeylerde Lagranyen yaklaşımla ağ noktaları yüzey hareketini takip ederken bu hareket akışkan alanı içerisine ağ yapısını fazla deforme etmeyecek şekilde iletilmektedir. Bu şekilde akışkan alanındaki noktalar tam olarak Lagranyen bir hareket yapmamakta ancak Euleryen olarak uzayda sabit bir koordinatta da kalmamaktadır. Dolayısıyla bu yönteme Keyfi Lagranyen-Euleryen denilmesi bundan ibarettir. Bu çalışmada uygulandığı haliyle ALE yaklaşımı sadece taşınım terimlerine ek olarak bir ağ hızı terimini ilave etmektedir ve integral formulasyonda bu terime bağlı olarak ek bir akı hesabı ortaya çıkmaktadır. Bu formülasyonun en büyük avantajı viskoz akı terimlerinin ve uzaydaki türevlerin değişmemesidir.

Bahsi geçen HEMLAB çözücüsü, nokta-tabanlı bir sonlu hacimler yöntemi uygulamaktadır. Kullandığı özel bir veri yapısı sayesinde, önbellekdeki verileri etkin bir şekilde kullanmakta ve bu da algoritmayı hızlandırmaktadır. HEMLAB

çözücüsünde Euler ve RANS denklemleri yardımıyla sıkıştırılabilir akış problemleri çözülebilmektedir. RANS denklemlerinde türbülans modeli olarak tek denklemli Negatif Spalart-Allmaras türbülan modeli kullanılmaktadır. Türbülans modeli ile korunum denklemleri tam bağdaşık bir şekilde çözülmektedir.

Viskoz akılar Green-Gauss yöntemiyle hesaplanmaktadır. Temel değişkenlerin türevlerinin çok daha pratik hesapları olmasına karşın bu yöntemin kullanılmasının birkaç nedeni bulunmaktadır. Öncelikle bu yöntem tek-çift ayrışmaya (odd-even decoupling) yol açmamaktadır. Ayrıca bu yöntemle Jakobyen tam bir şekilde elde edilmiş olup bu da algoritmayı daha gürbüz hale getirmektedir. Dolayısıyla daha yüksek CFL sayılarında algoritmanın çalışmasına imkan sağlamaktadır.

Taşınım akılarının ayrıklaştırılması için Roe, HLLC ve AUSM$^+-$up akı şemaları kullanılmaktadır. Bu şemalar çalışma kapsamında hareketli ağlar için genişletilmiştir. Roe yönteminde Jakobyen matrisi analitik olarak da hesaplanabilirken her üç yöntem için de Tapenade yazılımı kullanılarak hazırlanmış otomatik türev rutinleri kullanılabilmektedir.

Daha önce belirtildiği üzere taşınım akılarında hareketli ağların ek bir terimi bulunmaktadır. Bu terimin hesaplanması adına Ayrık Geometrik Korunum Yasası (DGCL) adı verilen bir yöntem uygulanmaktadır. Bu yöntemde bir hareketli yüzeyin akısı o yüzeyin zaman içerisinde taradığı hacimle belirlenmektedir. Sıkıştırılamaz akışlarda DGCL, kütle korunumunun makine hassasiyetinde sağlanması için gerekli iken sıkıştırılabilir akışlarda böyle bir rolü yoktur. Ancak literatürde belirtildiği üzere sıkıştırılabilir akışlarda ALE yönteminin kararlılığı ile yakından ilgilidir.

Yüksek mertebeden türevler için her ağ noktası üzerinde korunumlu değişkenlerin türevleri hesaplanıp Taylor serisi yardımıyla arayüzlere interpolasyon yapılmaktadır. Bu yöntemdeki türevler en küçük kareler metoduyla hesaplanmaktadır. Ayrıca eğer tercih edilirse ağırlılı en küçük kareler yöntemi de kullanılabilir. Ancak kuvvet katsayılarının yakınsaması düşünüldüğünde interpolasyon yaparken en küçük kareler kullanılması daha uygundur. Taylor serisi kullanarak türevler ile interpolasyon yapılırken türevler çok yüksek olduğunda ve ağ rezolüsyonu bu yüksek türevler kompanse edebilecek kadar yeterli olmadığında çözümde osilasyonlar ortaya çıkmaktadır. Hatta bu osilasyonlar çözümü ıraksatabilmektedir. Bunun önüne geçilebilmesi adına Venkatakrishnan limitleyicisi Wang modifikasyonu ile birlikte kullanılmaktadır.

Ortaya çıkan lineer-olmayan denklem takımı, Newton iterasyonuları vasıtasıyla lineerleştirilip çözülmektedir ancak bu yöntem yeterince küçük zaman adımlarında uygulanabilir olduğundan İkili Zaman Adımı adı verilen bir yöntem uygulanmıştır. Bu yöntemde her fiziksel zaman adımı, sahte zaman adımlarına bölünerek bu sahte zaman adımlarında denklem daimi hale erişiyormuş gibi çözüm yapılmaktadır. Matematiksel açıdan ise matrisin diyagonaline bir terim eklenmektedir. Ve bu terim CFL sayısına bağlı olarak hesaplanmaktadır. Fakat bu yöntemin de bazı kusurları vardır. Bunlardan en önemlisi denklemin sağ tarafı ile Jakobyeni uyumsuz yapmasıdır ki bu nedenle düşük CFL sayılarında yakınsama yavaşlamaktadır. Bundandır ki böyle yöntemlere yarı-Newton iterasyonu denmektedir. Bu yakınsama problemi Artan Düzgünleştirme uygulanarak bir nebze olsun giderilebilmektedir. Bu yöntemde denklemin sağ tarafındaki eksik terim bir yaklaşımla hesaplanıp eklenmektedir.

Ayrıca bu çalışma kapsamında rotor etrafındaki akış çözülürken isotropik olmayan ağ adaptasyonu kullanılacaktır. HAD çözümlerinde çözümün ağ yapısına olan bağlılığı detaylıca incelendiğinden *apriori* yöntemlerle ağ yapısı oluşturması çok pahalı olabilmektedir. Ayrıca ağ adaptasyonu yapmak istenilen fiziksel durum için optimal bir çözüm sunabilmektedir. Bunlardan dolayı pyAMG kütüphanesi HEMLAB çözücüsüne entegre edilerek ağ adaptasyonu uygulanmaktadır.

Anlatılan bu yöntemlerin uygunluğu bir 2-boyutlu problem üzerinde test edilmiştir. Bu problem NACA 0012 profilinin hem dikey doğrultuda hem de rotasyonel hareket ederkenki durumunu baz almakta olup Yüksek Mertebeden Hesaplamalı Akışkanlar Dinamiği Çalıştayında önerilmiştir. HEMLAB çözücüsünün sonuçları ile bu çalıştayda elde edilen sonuçlar iyi bir uyum sergilemektedir.

Son olarak tezin ana odak noktalarından biri olan S76 helikopterinin ana rotoru etrafındaki akış incelenmiştir. Bu problem için 3 adet simülasyon gerçekleştirilmiştir. Bu simülasyonlarda akı şeması ve farklı adaptasyon fonksiyonları test edilmiştir. Entropi sensörü iz bölgesini de yakalayabildiğinden sürükleme tahmininde fayda sağlayacağı düşünülmektedir. Ancak bu sensör potansiyel akış bölgelerinde ağ yapısının çok gevşek istiflenmesini sağlamaktadır. Mach sensörü ise bu bölgelerde entropiye kıyasla daha iyi sonuçlar vermektedir. Ayrıca bu çalışmada Roe şeması ve HLLC şeması arasında rotorun uç bölgesinde bir fark olduğu gözlemlenmiştir. Bu farktan dolayı Roe ile elde edilen çözümde Harten entropi düzeltmesi artırılarak sonuç elde edilebilmiştir. Genel olarak görülmüştür ki ağ adaptasyonu uygulaması kanat ucu vortexleri ve iz bölgesinin yakalanması adına iyi bir opsiyon sağlamaktadır.

Ayrıca son kısımda S76 rotoru etrafındaki akış RANS denklemleri yardımıyla çözülmüştür. Bu analizde ağ adaptasyonu uygulanmamıştır.

1. INTRODUCTION

After the industrialization of the world, humankind have gone further in amount that had never been seen in the world. As a consequence of this period, people designed and produced numbers of tools either to make life easier and protect humankind against difficulties or to destroy other nations. Whatever purpose is behind the development of high technology, people are now at the edge of science and engineering.

The demand for aerial applications of science and engineering emerged in this kind of athmosphere. Initially, fundemental understanding of Aerodynamics was constructed by great minds of history: Ludwig prandtl, Blasius, Theodore Von Karman and their friends and students. However, the mistery behind the Navier-Stokes equations still preserves its state. Firstly potential flow solutions are prepared and used to design airfoils. These solutions includes Prandtl's Lifting Line Theory, Vortex Lattice Theory and many others. They are now very easy and practical to implement and they are still very beneficial to be exploited in design optimization algorithms [5].

However, with the increasing power of computer technology, we became able to do much more than potential codes because it is hard to take viscous and nonlinear effects into account with the potential codes. In addition, flow around complex shapes can not even be solved with these tools. On the other hand, CFD tools have these capabilities. However, when CFD first appeared, computer memories were not sufficient to handle high numbers of data. It is stated in [6] that CFD solvers were capable of solving 1 million nodes in the 1990s and 64 million mesh points in the 2000s. Nowadays, the number is getting closer to 1 trillion mesh points. Therefore, CFD tools have become the main tool in the design phase of a vehicle.

There are several methodologies in CFD to solve Navier-Stokes equations. With the order of increasing complexity and decreasing applicability, they can be listed as Euler simulations, Reynolds Averaged Navier-Stokes simulations, Large Eddy Simulations, Detatched Eddy Simulations and Direct Numerical Simulations. In the 1980s, 3D Euler equations started to be solved by using CFD techniques and the solution of 3D

RANS equations followed it in the 1990s [6]. Today, RANS is widely used in practical applications because it provides sufficient amount of information, not including some details about the turbulence. It is not as naive as the Euler simulations however it is not as detailed as the LES, DES and DNS. Since LES, DES and DNS requires much more computer power and time for now for especially complex geometries, RANS appears as the best choice in order to simulate flow around a complex body and obtain information about flow physics.

In this study, an existing algorithm, HEMLAB Solver, is extended to handle with moving boundary problems. An Arbitrary Lagrangian-Eulerian algorithm is developed for compressible RANS equations. The developed algorithm is tested on 2D validation cases that are proposed under High Order CFD Workshop and It is utilized to solve flow field around S-76 main rotor that is proposed under AIAA Rotorcraft hover Prediction Workshop. To solve flow around the rotor, an anisotropic mesh adaptation strategy is exploited, which is provided by pyAMG library [7].

1.1 Literature Review For The Rotor Problem

The hover performance of a helicopter is key to the efficiency of its rotor system. In this respect, some preliminary methods such as Blade Element Theory, Blade Element Momentum Theory, Free and Prescribed Wake methods were developed to quickly analyze the hover performance. Although these methods are very fast, they do not directly take the viscous and nonlinear effects into account and are not applicable to complex geometries because of the simplifications made in these methods. On the other hand, CFD tools have high reliability and can calculate complex flow properties on complex geometries, although they require very high computational times compared to the simple theories mentioned above.

Since the flow-field around a rotor includes sharp velocity gradients, vorticities, complex phenomenas, such as vortex-blade interaction and shock waves, and since the rotors may have complicated geometries, CFD usage becomes inevitable for these types of analysis. Even if CFD tools have sufficient capabilities to capture these types of complex flow behaviours, it is still difficult to thoroughly capture tip vortices. Because the regions where the tip-vortices occur in the flow-field requires very fine grid structures. As stated by Chaderjian, the wake grid resolution has a vital importance to

capture tip vortices and, also, the grid resolution of the region under the rotor identifies the dissipation location of the wake shear layer [8]. Unfortunately, tip vortices has certain effects on the in-flow ratio of a rotor. Therefore, thorougly capture of the tip-vortices is a must for designer to correctly predict the thrust and power values of helicopter. Thankfully, CFD tools have been continuously being improved due to the developments in the field of computer science and the parallel clusters. They can now perform calculations on larger computational grids with lesser computational time than in the early years of CFD. It is stated in the Ref. [9] that flow solvers were able to conduct solutions on 1 million mesh points in the 1990s and on 64 million mesh points in the 2000s and, in recent years, they have reached the capability to work on grids with several hundred million mesh points. In addition, some advanced techniques such as Adaptive Mesh Refinement(AMR) can be used instead of creating very fine mesh at the beginning, allowing CFD tools to correctly simulate flow properties with minimum mesh element numbers.

S-76 helicopters have attracted attention to be utilized in experimental and CFD studies due to their commercial purposes since their first development in 1977. Thus, numerous data sets about the helicopter were produced both by experimental studies and, in relation to that, by CFD analysis. In this section, firstly some of the CFD studies and, then, some experimental studies are explained briefly.

One of the CFD studies is Chaderjian's study Ref. [8]. In this study, the author gives some important perspectives about the physics of the flow around the rotor and explains how this physics can be captured by CFD tools. Chaderjian performed the analysis for the Bell-Boeing V-22 Osprey tiltrotor and the UH-60 Blackhawk helicopter rotor via OVERFLOW CFD code. Then the results are interpreted comparing to the experimental data. An important result from the study states that the wake grid resolution has vital importance to capture tip vortex structures. Besides, the refined region under the rotor identifies the dissipation location of the wake shear layer if Adaptive Mesh Refinement (AMR) procedure is not used.

Another study was conducted by Jain [10] for the 1/4.71 scale S-76 rotor geometry by using the HELIOS CFD code. The study investigated the rotor hover analysis for different rotor tip Mach numbers and different tip shapes such as swept-tapered, rectangular, and swept-tapered-anhedral tips. As well as the effects of various

turbulence models that are the SA and the Shear Stress Transport (SST) models, the hub and coning effects were examined. The results imply that coning and hub have very little impact on figure of merit (FM) calculation. And at the Mach number of 0.55 and 0.60, FM results were close to each other but at the Mach number of 0.65, the rotor experienced a significant decrease in FM due to the increased drag at the outboard. The author, also, emphasized that swept-tapered-anhedral tip showed a little improvement of FM at the large thrust values although the experiments showed larger improvements for all thrust ranges.

In a further study of Jain [9], two solvers, which are CREATETM-AV Helios and standalone OVERFLOW, were compared to each other investigating only swept-tapered-anhedral tip. The analyses were set on various grid types that are structured and unstructured blade meshes. Various wake grid resolutions, as well, were practiced to assess the dissipation of the wake depending on the mesh coarseness. The HELIOS solver was utilized for two cases while one was structured blade mesh and the other one was unstructured. Both of them reached almost the same FM accuracy. However, some distinct characteristics were observed in the results of OVERFLOW. While one of the analyses accomplished by using OVERFLOW gave a good correlation with the HELIOS solutions, the other exhibited results in lesser correctness. Because the wake region in the latter mesh was more coarse than the former one which applies the AMR procedure. In addition to that, a 2nd order spatial scheme was utilized for the latter case where it was 5th order for the former.

Tadghighi [11], also, has done a study to predict the hover performance of the S-76 rotor and to validate the NSU3D module of the HELIOS solver against the experimental data [12]. Unsteady Reynolds Averaged Navier-Stokes Equations were solved using the NSU3D module of the HELIOS. To analyze, the 1/4.71 scale model of S-76 was used with swept-tapered tip geometry in conformity with the experimental study in the Ref. [12]. Unstructured near-body and structured off-body meshes were employed. The near-body flow was assumed as fully turbulent and was modeled by utilizing Spalart-Almaras model. Besides that, the off-body flow was presumed as inviscid to protect the wake from excessive dissipations. An interesting outcome of this research is that the author indicates the sudden increases in the thrust and

torque loadings near the %90-95 radius of the span are the indications of vortex-blade interactions.

To compare the capabilities of HELIOS and FUN3D codes, a research on the hover performance of the S-76 rotor was carried out in the Ref. [1]. The investigated rotor in the study has swept-tapered blade tip geometry. And four different grids were generated to perform analysis for the rotor. These computational grids differ from each other in refinement location and grid size. For the HELIOS solver, the near-body grids were created as an unstructured grid, while off-body grids were structured. However, a fully unstructured mesh was generated for the FUN3D solver. Also, the SA turbulence model is implemented into the solution with a rotation correction term in FUN3D while SA was employed alone in HELIOS. An AMR procedure was also performed during the solution in HELIOS solver. As a result, the differences between the two analysis are specified in the paper.

An attempt to contribute the AIAA Rotor Simulation Discussion Group on hover performance, was done by Narducci [13]. This group was established in 2013 and aimed to benchmark the capabilities of CFD tools for capturing the flow properties of rotors in hover condition with the experimental study of Balch and Lombardi [12]. Various tip geometries and coning effects were examined by using OVERFLOW CFD code in the study of Narducci. The flow around the body was assumed as fully turbulent and turbulence is modeled via Spalart-Almaras turbulence model with rotational correction. Also, the solution was in 2nd order accuracy in time. The results show that Figure of Merit (FM) was better predicted for Swept-Tapered-Anhedral tip geometry.

All of the CFD simulations including the studies indicated above require experimental studies for validation. Some of the main experimental studies which are accomplished for S-76 hover performance are denoted in the followings. One of them is the research of Balch and Lombardi [12]. The researchers tested various scaled main and tail rotor configurations. However these lines are merely focusing on the S-76 main rotor variants examined in the study. These variants are the main rotors with tip geometries of swept, tapered, swept-tapered, swept-tapered-anhedral and rectangular. The tests were generated on Sikorsky Model Hover Test Facility using the Basic Model Test Rig (BMTR). As a result of the experiments, a wide range of data including different

tip shape effects was provided to the literature. Another study is Johnson's study [14] which is on full scale helicopter rotors with different tip shapes. The aim of this study is to compare the results from the scaled models of the rotors and the full scale rotors with indicated advance ratio range from 0.075 to 0.4 and tip Mach range from 0.64 to 0.965. In addition to these studies, an experiment in Ref. [15] was conducted through obtaining performance of the S-76 main rotor. The tests in this study were carried out in the NASA Ames 80- by 120- Foot Wind Tunnel which is an open circuit wind tunnel. Its aim is to acquire low Mach number rotor performance and comparison with existing data sets and analytical solutions. For this purpose, the experiments for S-76 rotor in hover condition were designed where the shaft angle varied between -15 and 15 degree, the C_T/σ varied between 0.02 and 0.12 and the tip Mach number was equal to 0.605.

Regarding all the passages above, CFD have become essential analysis method to solve complex flows such as flows over a rotor in hover condition due to its complex vortical structures and naturally time dependent-being etc.. As a result of this, many studies were conducted in this respect. Even a working group was established under the AIAA to develop and validate CFD tools through capturing the complex flow phenomenas of rotor configurations. The studies mentioned above simulate various conditions on various rotor geometries and they will be used to validate the present study. In this respect, some parameters calculated in the studies for the main rotor of S76 helicopter with swept-tapered tip are summarized at the Table **??**. Thrust and torque coefficients and Figure of Merit(FM) values can be seen on the table for different analysis. As a convention, the coefficients and FM calculations are

$$C_T = \frac{T}{\rho A V_{tip}^2} \qquad C_Q = \frac{Q}{\rho A V_{tip}^3} \qquad FM = \sqrt{\frac{C_T}{2}}\frac{C_T}{C_Q}$$

where A is the disk area of the rotor, ρ indicates density and V_{tip} is the velocity at the tip of the rotor. FM is an indicator of rotor efficiency. It implies ideal power of the rotor, which excludes non-ideal effects such as viscosity and assumes uniformly distributed inflow ratio over the disk area, divided by actual power. Therefore, the value of unity is desired for FM. However it is impossible to attain unity, while FM takes a value between zero and unity.

In the present study, an anisotropic "Adaptive Mesh Refinement (AMR)" procedure is utilised during the solution process to overcome mentioned difficulty to capture tip-vortices. For this purpose, HEMLAB solver is integrated with an anisotropic adaptive mesh refinement library "pyAMG" [7], and used to analyse hover performance of the main rotor of S-76 helicopter in the present study. As a result, it is expected that thorugly capture of tip vortices around the rotor and predict the figure of merit value within an acceptable error range.

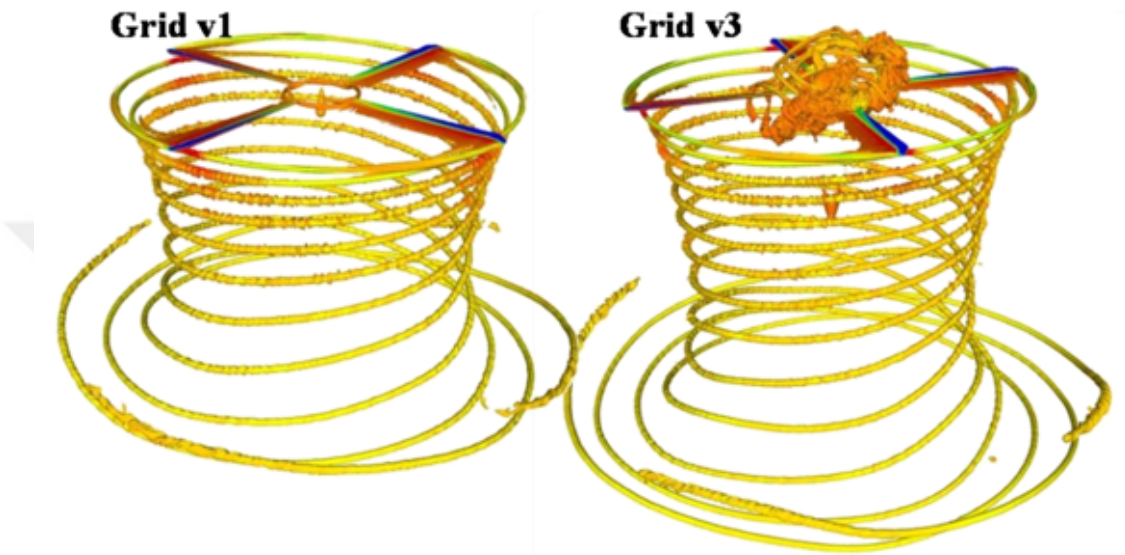

Figure 1.1 : Tip vortex captured by using AMR around the S76 main rotor from Abras' study [1].

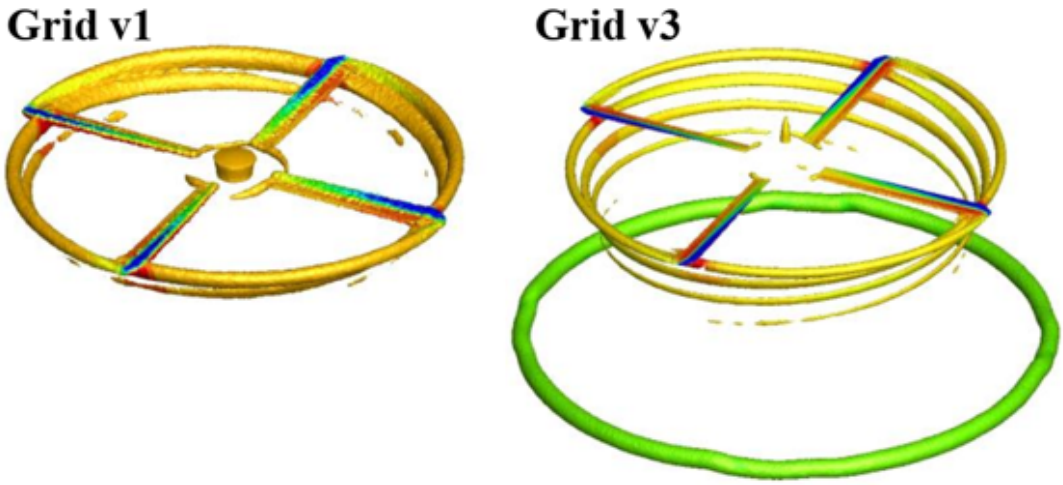

Figure 1.2 : Tip vortex captured without AMR around the S76 main rotor from Abras' study [1].

Adaptive refinement process plays a critical role to capture the wake around a rotor. Wake region requires fine resolution mesh because it is dissipated easily in the case of corser grids. Hence, It demands very high numbers of vertices when used without

AMR. However, in the case of use of AMR, high gradient regions, which one of them is the wake region, are simultaneously refined step-by-step during the solution process. Therefore it is possible with the AMR to capture these regions without excessive vertex numbers in the computational domain. This situation can be better understood by looking at the Figure 1.1 and 1.2. These figures are obtained from the study of Abras [1]. In the Figure 1.1, HELIOS solver is practised in cooperation with an AMR procedure for two type of grid structure. The other figure (Fig. 1.2) is from an analysis conducted on FUN3D solver without AMR for similar grid structures as the former one. As comparing these figures, the former one shows better capture of tip vortex structure than the latter one due to the AMR process. Therefore, the AMR procedure brings an advantage to capture vortex structures around a rotor and it is also benefited in the present study.

Another study of the author of this thesis shows the advantages to apply an-isotropic mesh adaptation to solve flow around a rotor. In the study, SU2 software was utilized with pyAMG library. The importance of baseline solution was implied. The mesh structures, wake region contours and tip vortices are shown in Figures 1.3, 1.4 and 1.5.

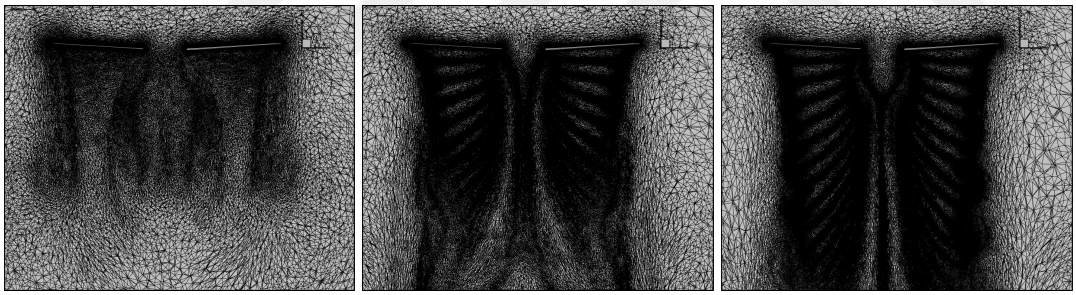

Figure 1.3 : Wake region mesh visualization (side-view) when an-isotropic mesh adaptation is applied with entropy sensor (Revolution 11, 15 and 21 from left to right) [2].

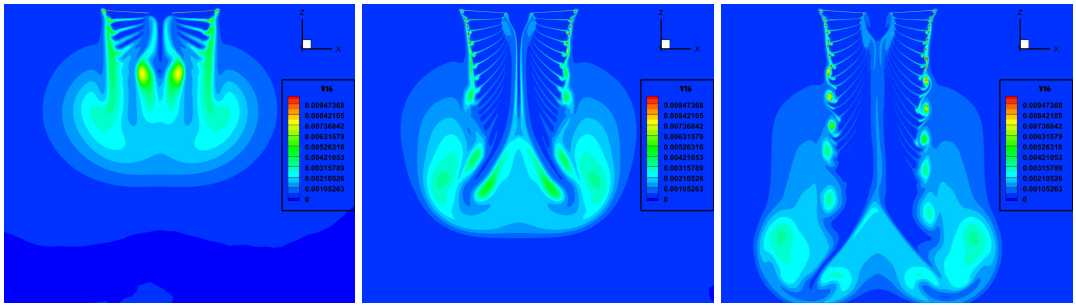

Figure 1.4 : Wake region visualization by entropy contour when an-isotropic mesh adaptation is applied with entropy sensor (Revolution 11, 15 and 21 from left to right) [2].

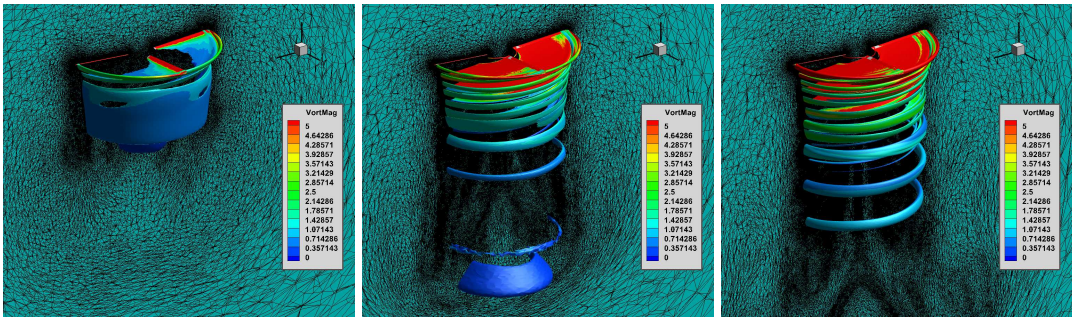

Figure 1.5 : Mesh and tip vortices with vorticity magnitude contours when an-isotropic mesh adaptation is applied with entropy sensor (Revolution 11, 15 and 21 from left to right) [2].

As a summary, application of anisotropic mesh adaptation to solve flow around a rotor, which is one of the main objective of the present study, provides a great opportunity to decrease time required for mesh generation and to improve the speed of solution by decreasing the amount of elements in the computational domain while achieving better accuracy. Because it allows to refine only necessary regions in the computational domain. However, some problems can occur when doing mesh adaptation.

1.2 Outline of The Thesis

In this section, outline of the thesis is explained briefly. In the thesis, firstly, the equations that governs the fluid physics will be given and examined. In this chapter, Arbitrary Lagrangian-Eulerian kinematics is also explained and applied to the Reynolds Averaged Navier-Stokes equations. Then, in the third chapter, numerical approaches to solve the governing equations are explained in detail, following a structure. The fourth chapter is allocated for the validation of developed algorithm on a 2D airfoil case. The main focus of the thesis will come right after the validation in the fifth chapter. The chapter explains the rotor problem and depicts the simulations that was governed.

2. GOVERNING EQUATIONS AND THEIR CHARACTERISTICS

In this section, we will discuss first the general equations that govern the flow physics in continuum, giving beneficial informations relevant to the numerical techniques. Then, we will introduce Arbitrary Lagrangian Eulerian kinematics which will be used to extent our in-house solver to solve moving boundary problems. Laslty, Reynolds Averaged Navier-Stokes equations that is used under the scope of this thesis will be given in integral form.

2.1 General Form of Governing Equations

The Navier-Stokes (N-S) equations constructs a special team of equations that governs fluid motion as a result of Newton's Law of motion. Despite of its very wide use and application in engineering and scientific fields, its general solution is unkown due to non-linearity. However, some solution of NS equations were found under special boundary conditions [16]. Even though we do not know the exact solution, we hope to find one that is suitable. However, it is a common topic among mathematicians to study that blow-up solutions can be found for N-S equations [17, 18].

2.1.1 Navier-Stokes equations

N-S Equations are a special sub-division of continuum material equations that is governed by Cauchy Equations. Under linear stress assumptions and Stokes' hypothesis, N-S Equations for Newtonian fluids can be written in conservative form as

$$\partial_t \mathbf{Q} + \nabla \cdot \mathbf{F}^c = \nabla \cdot \mathbf{F}^v \tag{2.1}$$

where $\mathbf{Q} = (\rho, \rho\mathbf{U}, E)$,

$$\mathbf{F}^c = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E+p)u \end{bmatrix} \mathbf{i} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (E+p)v \end{bmatrix} \mathbf{j} + \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (E+p)w \end{bmatrix} \mathbf{k} \tag{2.2}$$

and

$$\mathbf{F}^v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx}+v\tau_{xy}+w\tau_{xz}-q_x \end{bmatrix}\mathbf{i} + \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx}+v\tau_{yy}+w\tau_{yz}-q_y \end{bmatrix}\mathbf{j} + \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx}+v\tau_{zy}+w\tau_{zz}-q_z \end{bmatrix}\mathbf{k} \tag{2.3}$$

In this formulation, the linear stress tensor is written like

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} = \lambda(\nabla\cdot\mathbf{u})\mathbf{I} + \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}\right), \tag{2.4}$$

the heat fluxes are

$$\begin{aligned} q_x &= -\frac{M_\infty}{(\gamma-1)\,\mathrm{Re}_\infty \mathrm{Pr}_\infty}\mu\frac{\partial T}{\partial x} \\ q_y &= -\frac{M_\infty}{(\gamma-1)\,\mathrm{Re}_\infty \mathrm{Pr}_\infty}\mu\frac{\partial T}{\partial y} \\ q_z &= -\frac{M_\infty}{(\gamma-1)\,\mathrm{Re}_\infty \mathrm{Pr}_\infty}\mu\frac{\partial T}{\partial z} \end{aligned} \tag{2.5}$$

To satisfy the stationary conditions according to Stokes' hypothesis, λ becomes

$$\lambda = -\frac{2}{3}\mu. \tag{2.6}$$

However, It is worth to note that there are doubts about the Stokes' hypothesis to apply [19]. In the cases of shock formation in the flow field, the hypothesis can be problematic to determine the place of shock. Even, there are attempts to correct right-hand side of N-S Equations in order to avoid Stokes' hypotesis [20].

Although we do not know the exact eigenstructure of N-S Equations, general intention to understand N-S Equations approachs the terms seperately as convective and viscous terms. The convective terms that appears as $\mathbf{F}^c$ are the source of non-linearity and its continuos eigenstructure can not be defined. However, using quasi-linear formulation, we can conduct differential analysis of the waves. Since the convective terms are hyperbolic in time, the solution consists of waves that propagate in finite speed. So that, it allows discontinuities and discretization of convective terms needs special attention in order to find zone of influence and zone of dependence.

On the other hand, the viscous terms are appeared as pure Laplacian for incompressible in the right hand side of the equation. Laplacian serves as a smoother for PDE's like heat equation [21]. Therefore, one can think of viscous terms as a smoothing operation

over the discontinuities caused by convective terms. In addition, since it is parabolic in time, discretizations can be done easily using central schemes.

2.1.2 Euler equations, quasi-linear form and eigenstructure

Since it has practical applications and covers an important part of this study, Euler equations are examined here. 1D Euler equations can be written in the following form.

$$\partial_t \mathbf{Q} + \partial_x \mathbf{F}^c = 0 \tag{2.7}$$

In this form Euler equations defines a divergence-free field in space-time that is applied on vector field (Q,F). It should be noted that the term "space-time" here does not refers to space-time in General Relativity under Minkowskian metric. It just reflects a ground that is constructed by 3 space and 1 time dimensions under the standart metric. Writing the equation in quasi-linear form

$$\partial_t \mathbf{Q} + \mathbf{J}\partial_x \mathbf{Q} = 0 \tag{2.8}$$

where the Jacobian $\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{Q}}$. Diagonalizing the Jacobian, eigenstructure can be obtained like

$$\partial_t \mathbf{Q} + T\Lambda T^{-1}\partial_x \mathbf{Q} = 0. \tag{2.9}$$

If we multiply the equation by T^{-1}, it becomes

$$\partial_t \mathbf{W} + \Lambda \partial_x \mathbf{W} = 0 \tag{2.10}$$

where $\frac{\partial \mathbf{W}}{\partial \mathbf{Q}} = T^{-1}$. W is called *Riemann Invariants*, since they are invariant along the characteristic line as it is shown below, and the equations for these variables are de-coupled. This form is widely utilized and very practical to develop numerical schemes for Compressible Euler Equations.

Invariance of the variables W along the characteristic curves can be obtained in the following way. Let's write the total derivative and divide it by dt,

$$\frac{d\mathbf{W}}{dt} = \frac{\partial \mathbf{W}}{\partial t} + \frac{dx}{dt}\frac{\partial \mathbf{W}}{\partial x} \tag{2.11}$$

The resemblence between Equations 2.10 and 2.11 stands out. If $\frac{dx}{dt} = \Lambda$, $d\mathbf{W}/dt$ becomes zero. Since the vise versa is true for $d\mathbf{W}/dx$, we say that the curve of $\frac{dx}{dt} = \Lambda$ is the characteristic curve of PDE and $\mathbf{W}$ is invariant along the curve. Implication of this

as a condition on divergence-free vector fields brings Lax's Jump condition that states $d(\mathbf{F}-\Lambda\mathbf{Q})=0$ across the discontinuity. It is a crucial condition for numerical schemes like HLLC because the scheme calculates fluxes using state variables estimated using the relations for discontinuities.

The mentioned objects in this section like T and Λ will be adressed in the following sections when the numerical Jacobian is given. In addition, It might be appropriate to write Riemann invariants here. They constructs a team of variables like [22]

$$\begin{aligned} &\text{1-Riemann invariants:} \quad s, \quad u+\frac{2c}{\gamma-1}, \\ &\text{2-Riemann invariants:} \quad u, \quad p, \\ &\text{3-Riemann invariants:} \quad s, \quad u-\frac{2c}{\gamma-1}. \end{aligned}$$

2.2 Arbitrary Lagrangian Eulerian Kinematics

Every numerical algorithm that has a background mesh describes the problem in three-frames. One of the frames is called Lagrangian or material frame in which each coordinate pairs consist of the material and time. The other one is Eulerian frame in which the coordinate pairs consist of the location of points on background space and time. The third one is the mesh domain in which coordinate pairs are mesh points and time. This is illustrated in Figure 2.1 where X is material domain, x is spatial domain

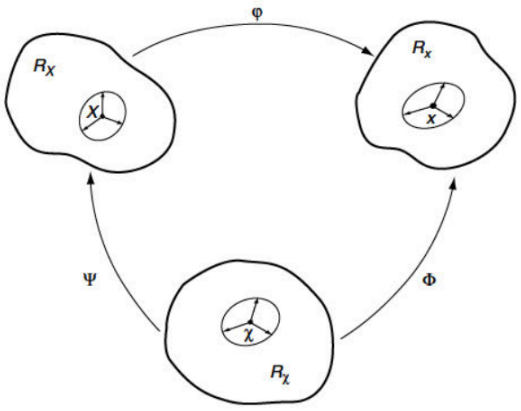

Figure 2.1 : Three frames (material, spatial and mesh) and their mapping between each other [3].

and χ is mesh domain.

In the material domain, a property f only depends on the t time. So that,

$$\frac{df}{dt}=\left(\frac{\partial f}{\partial t}\right)_X \tag{2.12}$$

Using the mapping ϕ between spatial and material domain, derivative a property f defined on material domain can be written in spatial domain as

$$\frac{df}{dt} = \left(\frac{\partial f}{\partial t}\right)_x + \left(\frac{\partial f}{\partial x}\right)_t \frac{dx}{dt} \tag{2.13}$$

where dx/dt is the material velocity on space.

Before going further, I would rather to interrupt the mathematical description to explain why ALE is needed. In Eulerian algoritms, mesh domain is identical with the spatial one. So that, the locations of mesh points are fixed in space. Otherwise, in Lagrangian algorithms, mesh domain is matched with the material domain. So that, the locations of the mesh points follows the material. As one might expect that, if the material is highly deformed, Lagrangian techniques might not behave well since mesh is also deformed. On the other hand, Eulerian definition of motion can easily handle large deformation with the expense of loosing the accuracy on the interface. Arbitrary Lagrangian-Eulerian techniques coalesce the advantages of Eulerian and Lagrangian approachs. The boundaries are treated as Lagrangian whereas inner points of mesh is moved suitably while preserving the precise shape of interface and quality of mesh.

To obtain ALE formulation, we exploit a very beneficial approach: we define the time derivative on mesh domain whereas spatial derivatives are defined on spatial domain. So that the derivative in ALE kinematics becomes

$$\frac{df}{dt} = \left(\frac{\partial f}{\partial t}\right)_\chi + \left(\frac{\partial f}{\partial x}\right)_t \left(\frac{dx}{dt} - \left(\frac{\partial x}{\partial t}\right)_\chi\right) \tag{2.14}$$

where $(\partial x/\partial t)_\chi$ is the mesh velocity. The advantage of using this formulation is that we can integrate the property on a specified mesh point in time by using only spatial derivatives because the time derivative is defined on the mesh domain whereas spatial derivatives are still in the spatial domain. Besides, we do not need to modify viscous terms due to this characteristic of the derivatives. The only modification on the original formulation is the mesh velocity. Details about the derivation of the formulation can be found in the Reference [3]. If the property f is replaced by the velocity, Equation 2.14 represents the acceleration. Therefore, ALE formulation for N-S equations will be derived in the following section by using this form.

2.3 RANS Equations Under ALE Kinematics

Since it is hard to capture all scales in a flow field that is related to the turbulence, the common approach is to use turbulence models rather than direct simulations. In this study, Reynolds Averaged Navier-Stokes (RANS) equations are employed. RANS equations consist of Navier-Stokes equations to predict the mean flow and additional equations to model the turbulence. In this study, Spalart-Allmaras one equation model is utilized for turbulence. It is a model based on Boussinesq hypothesis in which turbulence viscosity is solved explicitly and added to the dynamic mean viscosity.

The RANS equations in integral form are written as

$$\frac{\partial}{\partial t}\iiint_{\Omega}\mathbf{Q}dV+\oint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F}^{c}dS-\oint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F}^{v}dS=0 \tag{2.15}$$

where $\mathbf{Q}$ is the same vector as in Equation 2.1 and $\mathbf{n}$ is the normal vector of the surface S. The Ω is the volume where as $\partial\Omega$ is the boundaries enclosing the volume. The convective fluxes, $\mathbf{F}^{c}$, are changed in order to include fluxes due to moving mesh where the only difference from Equation 2.2 is the grid velocities as shown below.

$$\mathbf{F}^{c}=\begin{Bmatrix}\rho(u-\dot{x})\\ \rho u(u-\dot{x})+p\\ \rho v(u-\dot{x})\\ \rho w(u-\dot{x})\\ (E(u-\dot{x})+pu\end{Bmatrix}\mathbf{i}+\begin{Bmatrix}\rho(v-\dot{y})\\ \rho v(v-\dot{y})\\ \rho v(v-\dot{y})+p\\ \rho w(v-\dot{y})\\ (E(v-\dot{y})+pv\end{Bmatrix}\mathbf{j}+\begin{Bmatrix}\rho(w-\dot{z})\\ \rho u(w-\dot{z})\\ \rho v(w-\dot{z})\\ \rho w(w-\dot{z})+p\\ (E(w-\dot{z})+pw\end{Bmatrix}\mathbf{k} \tag{2.16}$$

As explained in the Section 2.2, the derivatives with respect to spatial coordinates are not changed. Therefore, viscous fluxes $\mathbf{F^v}$ are same as in the Equation 2.3. However, the stress terms are modified under Boussinesq approximation in order to model turbulence. The viscosity in the Equation 2.4 is modified as the sum of turbulent and dynamic viscosity. So that, the stress terms in RANS equation can be written

$$\begin{aligned}\tau_{xx}&=\frac{M_\infty}{Re_\infty}\left[2\mu^*\frac{\partial u}{\partial x}+\lambda^*\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)\right]\\ \tau_{yy}&=\frac{M_\infty}{Re_\infty}\left[2\mu^*\frac{\partial v}{\partial y}+\lambda^*\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)\right]\\ \tau_{zz}&=\frac{M_\infty}{Re_\infty}\left[2\mu^*\frac{\partial w}{\partial z}+\lambda^*\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)\right]\\ \tau_{xy}&=\frac{M_\infty}{Re_\infty}\left[\mu^*\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]\\ \tau_{xz}&=\frac{M_\infty}{Rc_\infty}\left[\mu^*\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)\right]\\ \tau_{yz}&=\frac{M_\infty}{Rc_\infty}\left[\mu^*\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)\right]\end{aligned}, \tag{2.17}$$

where non-dimensional viscosity

$$\mu^* = \frac{\mu + \mu_t}{\mu_\infty}$$

and, using Stokes' hypothesis

$$\lambda^* = -\frac{2}{3}\mu^*$$

Here, dynamic viscosity is found using Sutherland's law,

$$\frac{\mu}{\mu_\infty} = T\sqrt{T}\frac{1 + S/T_\infty}{T + S/T_\infty} \tag{2.18}$$

Similarly, heat transfer can be written using Fourier heat transfer law

$$\begin{aligned} q_x &= -\frac{M_\infty}{(\gamma-1)Re_\infty Pr_\infty}\left[\mu^*\frac{\partial T}{\partial x}\right] \\ q_y &= -\frac{M_\infty}{(\gamma-1)Re_\infty Pr_\infty}\left[\mu^*\frac{\partial T}{\partial y}\right] \\ q_z &= -\frac{M_\infty}{(\gamma-1)Re_\infty Pr_\infty}\left[\mu^*\frac{\partial T}{\partial z}\right] \end{aligned} \tag{2.19}$$

The turbulence viscosity, μ_t, is defined by the turbulence model. In this study, negative Spalart-Allmaras Turbulence Model is utilized in a monolithic approach with the Navier-Stokes equations.

2.3.1 Negative Spalart-Allmaras model on moving meshes

For moving meshes, the Negative Spalart-Allmaras model with no transition equation in non-dimensional form becomes

if $\hat{\nu} \geq 0$ then

$$\begin{aligned} \frac{\partial \hat{\nu}}{\partial t} + (\mathbf{u} - \dot{\mathbf{x}})\cdot\nabla\hat{\nu} = {} & c_{b1}\hat{S}\hat{\nu} \\ & - \frac{M_\infty}{Re}\left[c_{w1}f_w\right]\left(\frac{\hat{\nu}}{d}\right)^2 \\ & + \frac{M_\infty}{Re\sigma}\left[\nabla\cdot\left[(\hat{\nu}f_n + \nu)\nabla\hat{\nu}\right] + c_{b2}\nabla\hat{\nu}\cdot\nabla\hat{\nu}\right] \end{aligned} \tag{2.20}$$

if $\hat{\nu} < 0$ then

$$\begin{aligned} \frac{\partial \hat{\nu}}{\partial t} + (\mathbf{u} - \dot{\mathbf{x}})\cdot\nabla\hat{\nu} = {} & c_{b1}|\omega|\hat{\nu} \\ & + \frac{M_\infty}{Re}c_{w1}\left(\frac{\hat{\nu}}{d}\right)^2 \\ & + \frac{M_\infty}{Re\sigma}\left[\nabla\cdot\left[(\hat{\nu}f_n + \nu)\nabla\hat{\nu}\right] + c_{b2}\nabla\hat{\nu}\cdot\nabla\hat{\nu}\right] \end{aligned} \tag{2.21}$$

where the turbulence viscosity is

$$\mu_t^* = \begin{cases} \rho \hat{\nu} f_{v1}, & \hat{\nu} > 0 \\ 0, & \text{otherwise} \end{cases}$$

if $\nu < 0$, f_n function becomes

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3} \tag{2.22}$$

If $\nu > 0$ then;

$$f_n = 1 \tag{2.23}$$

where $c_{n1} = 16$. $\hat{S}$ is determined as

$$\hat{S} = \begin{cases} |\omega| + \bar{S}, & \text{if } \bar{S} \geq -c_{v2}|\omega| \\ |\omega| + |\omega| \frac{(c_{v2}^2 |\omega| + c_{v3} \bar{S})}{(c_{v3} - 2c_{v2})|\omega| - \bar{S}}, & \text{otherwise} \end{cases} \tag{2.24}$$

where $c_{v2} = 0.7$ and $c_{v3} = 0.9$.

and other parameters are

$$\begin{aligned} \chi &= \frac{\hat{\nu}}{\nu^*} = \frac{\hat{\nu}\rho}{\mu^*} \\ r &= \min\left[10, \frac{M_\infty}{Re_\infty} \frac{\hat{\nu}}{\hat{S}^2 d^2}\right] \\ g &= r + c_{w2}\left(r^6 - r\right) \\ f_w &= g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6} \end{aligned} \tag{2.25}$$

$$\sigma = 2/3 \quad c_{b1} = 0.1355 \quad c_{b2} = 0.622$$

$$\kappa = 0.41 \quad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma} \quad c_{w2} = 0.3 \quad c_{w3} = 2c_{v1} = 7.1.$$

3. NUMERICAL MODELS AND DISCRETIZATIONS

A new algorithm to solve moving boundary problems in compressible flow regimes is developed based on the in-house HEMLAB algorithm. HEMLAB solver [23] uses a vertex based median-dual finite volume approach. It employs an efficient edge based data structure. It utilises pointers to reach a specific data lying on the facet-edge data types belonging each edge of a face of each volume. Using the pointers *org, dest, sym and rot*, whole domain can be swept starting from an arbitrary point. Due to this unique data structure, spatial locality in physical memory is achieved. Therefore, the algorithm works cache-efficiently. HEMLAB can solve Euler and RANS equations. For turbulent flows, it utilises negative one equation Spalart-Allmaras model [24]. The resulting system is solved in a monolithic manner with Navier-Stokes equations. Implicit solutions are obtained by employing Newton's sub-iterations for implicit Euler formulation. PETSc [25] library is exploited to solve implicit equation system with FGMRES and Generalized Conjugate Residual (GCR) Krylov subspace methods. Time dependent simulations are conducted by first order dual time-stepping approach [26]. HEMLAB solver employs Roe's flux difference scheme [27] to discretize convective terms. Recently, its capability has been enhanced by HLLC [28] and $AUSM^{+} - up$ [29] schemes where Jacobian matrices of the flux schemes are calculated using automatic differentiation library Tapenade [30, 31]. The second order accuracy in spatial discretizations are achieved by using gradients computed at nodes. The gradients are calculated by unweighted second-order upwind least square approach [32] and left and right state can, then, be obtained using Taylor series expansion with second order accuracy. As well as the Jacobian free option, second-order and first-order Jacobian can be constructed. In Jacobian free approach, first order Jacobian is used as preconditioner. Due to the time considerations, in the current study, the Jacobian is calculated using first order approach. Following the recent advencements in the adaptive mesh refinement techniques, pyAMG anisotropic mesh adaptation library

is implemented into HEMLAB solver. In this section the details of the algorithm and HEMLAB solver will be discussed.

3.1 Finite Volume Discretization

HEMLAB solver utilizes a nose-based Finite Volume discretization. In this formulation, Equation 2.15 is shown in semi-discrete form as

$$\frac{\partial}{\partial t}(\mathbf{Q_i}V_i) + \sum_{j \in N(i)} \mathbf{n}_{ij} \cdot \mathbf{F}^C_{ij}(\mathbf{Q}) S_{ij} = \sum_{j \in N(i)} \mathbf{n}_{ij} \cdot \mathbf{F}^V_{ij}(\mathbf{Q}) S_{ij} \tag{3.1}$$

where $\mathbf{Q_i}$ and V_i are the conservative state vector and the volume of control volume surrounding the node i. $N(\mathbf{i})$ is the set of neighbouring points of vertex i. And $\mathbf{n_{ij}}$ indicates the normal vector of the surface S_{ij} (edge length in 2D) that is constructed at the interface of control volumes of vertices i and j. The discretization is shown in the Figure 3.1.

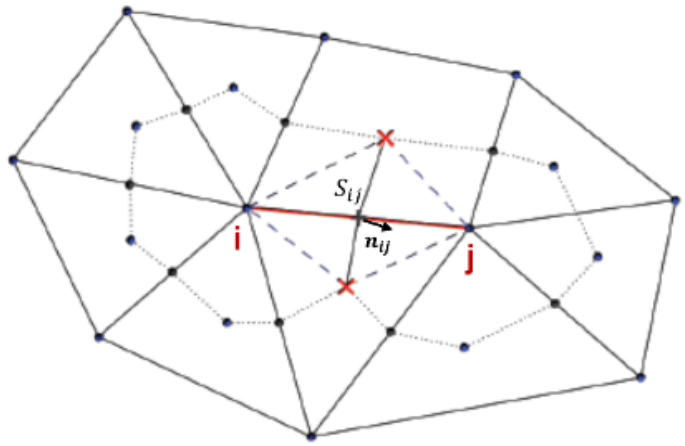

Figure 3.1 : Control volumes constructed around nodes i and j.

The geometrical properties are stored in related variables as a part of the *node* data type. Therefore, we need to calculate convective $\mathbf{F}^c_{ij}(\mathbf{Q})$) and viscous $\mathbf{F}^{\mathrm{V}}_{ij}(\mathbf{Q})$ fluxes at the interfaces.

3.2 Convective Flux Calculation

3.2.1 Roe's flux difference scheme

Roe scheme [27], is made of a stabilization term in addition to the second order central interpolation and it is written as

$$\mathbf{F_{ij}}\left(\mathbf{Q_L},\mathbf{Q_R}\right)=\frac{1}{2}\left[\mathbf{F}\left(\mathbf{Q_L}\right)+\mathbf{F}\left(\mathbf{Q_R}\right)-\hat{T}|\hat{\Lambda}|\hat{T}^{-1}\left(\mathbf{Q_R}-\mathbf{Q_L}\right)\right] \tag{3.2}$$

where Q_L and Q_R is the conservative state vectors that are in the left and right of the interface. $\dot{\mathbf{x}}\cdot\mathbf{n_{ij}}$ is the normal component of the grid velocity. If the left and right state vectors are just the state vectors on the vertices i and j, the discretization becomes first-order. However, we can obtain the lest and right states using Taylor series expansion to achieve second-order accuracy. This process will be explained later.

$\hat{T}$ is the right eigen-vectors of the Jacobian of Roe's stabilization term whereas $\hat{\Lambda}$ is the eigenvalues. These quantities are calculated using the Roe's linearization where

$$\begin{aligned}
\hat{\rho}&=\sqrt{\rho_L\rho_R}\\
\hat{\mathbf{u}}&=\frac{\sqrt{\rho_L}\mathbf{u}_L+\sqrt{\rho_R}\mathbf{u}_R}{\sqrt{\rho_L}+\sqrt{\rho_R}}\\
\hat{H}&=\frac{\sqrt{\rho_L}H_L+\sqrt{\rho_R}H_R}{\sqrt{\rho_L}+\sqrt{\rho_R}}\\
\hat{a}&=\sqrt{(\gamma-1)\left[\hat{H}-0.5\hat{u}^2\right]}.
\end{aligned} \tag{3.3}$$

Using these values in the left eigenvectors shown as

$$T^{-1}=\begin{bmatrix}
n_x a_5-(n_z v-n_y w)\rho^{-1} & n_x a_1 u c^{-2} & n_x a_1 v c^{-2}+n_z\rho^{-1} & n_x a_1 u c^{-2}-n_y\rho^{-1} & -n_x a_1 c^{-2}\\
n_y a_5-(n_x w-n_z u)\rho^{-1} & n_y a_1 u c^{-2}-n_z\rho^{-1} & n_y a_1 v c^{-2} & n_y a_1 w c^{-2}+n_x\rho^{-1} & -n_y a_1 c^{-2}\\
n_z a_5-(n_y u-n_x v)\rho^{-1} & n_z a_1 u c^{-2}+n_y\rho^{-1} & n_z a_1 v c^{-2}-n_x\rho^{-1} & n_z a_1 w c^{-2} & -n_z a_1 c^{-2}\\
a_2(\phi-cV) & -a_2(a_1 u-n_x c) & -a_2(a_1 v-n_y c) & -a_2(a_1 w-n_z c) & a_1 a_2\\
a_2(\phi+cV) & -a_2(a_1 u+n_x c) & -a_2(a_1 v+n_y c) & -a_2(a_1 w+n_z c) & a_1 a_2
\end{bmatrix} \tag{3.4}$$

and in the right eigenvectors shown as

$$T=\begin{bmatrix}
n_x & n_y & n_z & a_3 & a_3\\
n_x u & n_y u-n_z\rho & n_z u+n_y\rho & a_3(u+n_x c) & a_3(u-n_x c)\\
n_x v+n_z\rho & n_y v & n_z v-n_x\rho & a_3(v+n_y c) & a_3(v-n_y c)\\
n_x w-n_y\rho & n_y w+n_x\rho & n_z w & a_3(w+n_z c) & a_3(w-n_z c)\\
n_x a_6+\rho(n_z v-n_y w) & n_y a_6+\rho(n_x w-n_z u) & n_z a_6+\rho(n_y u-n_x v) & a_3(a_4+cV) & a_3(a_4-cV)
\end{bmatrix}, \tag{3.5}$$

the stabilization term can be calculated where wave speeds are

$$\begin{aligned}\Lambda_1 &= \Lambda_2 = \Lambda_3 = V - V_t \\ \Lambda_4 &= V - V_t + c \\ \Lambda_5 &= V - V_t - c\end{aligned} \tag{3.6}$$

and other abbreviations to calculate eigenstructure can be listed as below.

$$\begin{aligned}a_1 &= \gamma - 1 \quad & a_6 &= \tfrac{\phi}{\gamma-1} \\ a_2 &= \tfrac{1}{\rho c\sqrt{2}} \quad & V &= n_x u + n_y v + n_z w \\ a_3 &= \tfrac{\rho}{c\sqrt{2}} \quad & \phi &= \tfrac{1}{2}(\gamma-1)\left(u^2+v^2+w^2\right) \\ a_4 &= \tfrac{\phi+c^2}{\gamma-1} \\ a_5 &= 1 - \tfrac{\phi}{c^2}\end{aligned} \tag{3.7}$$

3.2.1.1 Jacobian of Roe scheme

The Jacobians are written as

$$\begin{aligned}\mathbf{J}_L &= \frac{1}{2}(A_L + \hat{A}) \\ \mathbf{J}_R &= \frac{1}{2}(A_R - \hat{A})\end{aligned} \tag{3.8}$$

where $\hat{A} = \hat{T}|\hat{\Lambda}|\hat{T}^{-1}$ is calculated using Roe's linearization. A_L and A_R are the Jacobians of the left and right fluxes

$$\begin{aligned}A_L &= \frac{\partial \mathbf{F_L}}{\partial \mathbf{Q_L}} \\ A_R &= \frac{\partial \mathbf{F_R}}{\partial \mathbf{Q_R}}\end{aligned}. \tag{3.9}$$

The Jacobian matrix A for moving grids is written as [1]

$$A = \begin{bmatrix} -V_t & n_x & n_y & n_z & 0 \\ n_x\phi - uV & V - V_t - a_3 n_x u & n_y u - a_2 n_x v & n_z u - a_2 n_x w & a_2 n_x \\ n_y\phi - vV & n_x v - a_2 n_y u & V - V_t - a_3 n_y v & n_z v - a_2 n_y w & a_2 n_y \\ n_z\phi - wV & n_x w - a_2 n_z u & n_y w - a_2 n_z v & V - V_t - a_3 n_z w & a_2 n_z \\ V(\phi - a_1) & n_x a_1 - a_2 uV & n_y a_1 - a_2 vV & n_z a_1 - a_2 wV & \gamma V - V_t \end{bmatrix} \tag{3.10}$$

[1] Jacobian is obtained from Reference [33] in the Appendix 7 Convective Flux Jacobian. However, there were some typos and they are fixed in the current study.

where

$$
\begin{aligned}
a_1 &= \gamma E - \phi \\
a_2 &= \gamma - 1 \\
a_3 &= \gamma - 2 \\
V &= n_x u + n_y v + n_z w \\
V_t &= \dot{x} n_x + \dot{y} n_y + \dot{z} n_z \\
\phi &= \frac{1}{2}(\gamma - 1)\left(u^2 + v^2 + w^2\right)
\end{aligned}
\tag{3.11}
$$

On the other hand, automatic differentiation library Tapenade [30,31] is also utilised to obtain jacobian of the Roe scheme. This approach is found to have better convergence rates.

3.2.2 HLLC flux scheme

In the context of this study, the HLLC scheme in the literature [28] has been modified by considering the ALE equations. In the employed HLLC scheme, wave speeds are treated as suggested by Batten and colleagues [34]. Accordingly, the HLLC flux is calculated as follows:

$$
\mathbf{F}_{ij}^{HLLC}\left(\mathbf{Q_i}, \mathbf{Q_j}, \mathbf{n}_{ij}\right) = \begin{cases} \mathbf{F}\left(\mathbf{Q_i}\right) \cdot \mathbf{n}_{ij} & \text{if } \lambda_i > 0 \\ \mathbf{F}\left(\mathbf{Q_i^\star}\right) \cdot \mathbf{n}_{ij} & \text{if } \lambda_i \leq 0 < \lambda_M \\ \mathbf{F}\left(\mathbf{Q_j^\star}\right) \cdot \mathbf{n}_{ij} & \text{if } \lambda_M \leq 0 \leq \lambda_j \\ \mathbf{F}\left(\mathbf{Q_j^\star}\right) \cdot \mathbf{n}_{ij} & \text{if } \lambda_j < 0 \end{cases}
\tag{3.12}
$$

Various parameters in this formulation are expressed in the followings. It should be noted that in all the equations here, $u_n = (\mathbf{u} - \dot{\mathbf{x}}) \cdot \mathbf{n}$ is assumed, where $\dot{\mathbf{x}}$ represents the grid velocity. Additionally, λ_M is the speed of the contact wave, and λ_i and λ_j are the maximum and minimum acoustic wave speeds.

$$
\begin{aligned}
\mathbf{Q}^\star &= \frac{1}{\lambda - \lambda_M} \begin{pmatrix} \rho\left(\lambda - u_n\right) \\ \rho \mathbf{u}\left(\lambda - u_n\right) + \left(p^\star - p\right)\mathbf{n} \\ \rho E\left(\lambda - u_n\right) + p^\star \lambda_M - p u_n \end{pmatrix} \\
p^\star &= \rho\left(\lambda - u_n\right)\left(\lambda_M - u\_n\right) + p \\
\lambda_M &= \frac{\rho_j u_{n_j}\left(\lambda_j - u_{n_j}\right) - \rho_i u_{n_i}\left(\lambda_i - u_{n_i}\right) + p_i - p_j}{\rho_j\left(\lambda_j - u_{n_j}\right) - \rho_i\left(\lambda_i - u_{n_i}\right)} \\
\lambda_i &= \min\left(u_{n_i} - c_i, \widetilde{u_n} - \tilde{c}\right) \text{ ve } \lambda_j = \max\left(u_{n_j} + c_j, \widetilde{u_n} + \tilde{c}\right)
\end{aligned}
\tag{3.13}
$$

The acoustic waves $\widetilde{u_n} - \tilde{c}$ and $\widetilde{u_n} + \tilde{c}$ within the above wave speeds are calculated using the Roe linearization.

The Jacobians of HLLC scheme is calculated using Tapenade [31] automatic differentiation library.

3.2.3 Grid velocity and discrete geometric conservation law

In order to calculate the fluxes due to mesh movement, a special attention is paid to satisfy the Discrete Geometric Conservation Law (DGCL) [35]. The DGCL implies that the volumetric increase of a moving element should be equal to the sum of the volumes scanned by the surfaces surrounding that volume. Thus, the flux of the mesh movement is calculated as follows for first-order accuracy:

$$\dot{\mathbf{x}}_{125}^{n+1} \cdot \mathbf{S}_{125}^{n+1} = \frac{\Delta V_{125}^{n+1/2}}{\Delta t} \tag{3.14}$$

It is straightforward to extend this for second-order formulation, as applied in the reference [4], and can be written as follows. However, in this study, a first-order calculation is used to match with the time discretization.

$$\dot{\mathbf{x}}_{125}^{n+1} \cdot \mathbf{S}_{125}^{n+1} = \frac{3}{2\Delta t}\Delta V_{125}^{n+1/2} - \frac{1}{2\Delta t}\Delta V_{125}^{n-1/2} \tag{3.15}$$

Here, $\Delta V_{125^{n+1/2}}$ represents the volume scanned by the area S_{125} between time intervals n and $n+1$, and $\Delta V_{125^{n-1/2}}$ represents the volume scanned by the area S_{125} between time intervals $n-1$ and n. The volume scanned by the area S_{125} between time intervals n and $n+1$ is calculated by dividing the same volume into three tetrahedral elements, as shown in Figure 3.2:

$$\begin{aligned}
\Delta V_{125}^{n+1/2} = & \frac{1}{3}\left[\frac{1}{2}\left(\mathbf{x}_2^{n+1} - \mathbf{x}_5^{n+1}\right) \times \left(\mathbf{x}_1^{n+1} - \mathbf{x}_5^{n+1}\right)\right] \cdot \left(\frac{\Delta \mathbf{d}_2^{n+1} + \Delta \mathbf{d}_1^{n+1}}{2}\right) \\
& + \frac{1}{3}\left[\frac{1}{2}\left(\mathbf{x}_2^{n} - \mathbf{x}_5^{n+1}\right) \times \left(\mathbf{x}_1^{n} - \mathbf{x}_5^{n+1}\right)\right] \cdot \left(\frac{\Delta \mathbf{d}_2^{n+1} + \Delta \mathbf{d}_1^{n+1}}{2}\right) \\
& + \frac{1}{3}\left[\frac{1}{2}\left(\mathbf{x}_2^{n} - \mathbf{x}_5^{n}\right) \times \left(\mathbf{x}_1^{n} - \mathbf{x}_5^{n}\right)\right] \cdot \left(\Delta \mathbf{d}_5^{n+1}\right),
\end{aligned} \tag{3.16}$$

Here, the displacement vector is defined as

$$\Delta \mathbf{d}_i^{n+1} = \mathbf{x}_i^{n+1} - \mathbf{x}_i^{n} \tag{3.17}$$

This approach ensures DGCL at the discrete level and maintains a uniform flow solution independent of mesh movement within the existing ALE algorithm. Even

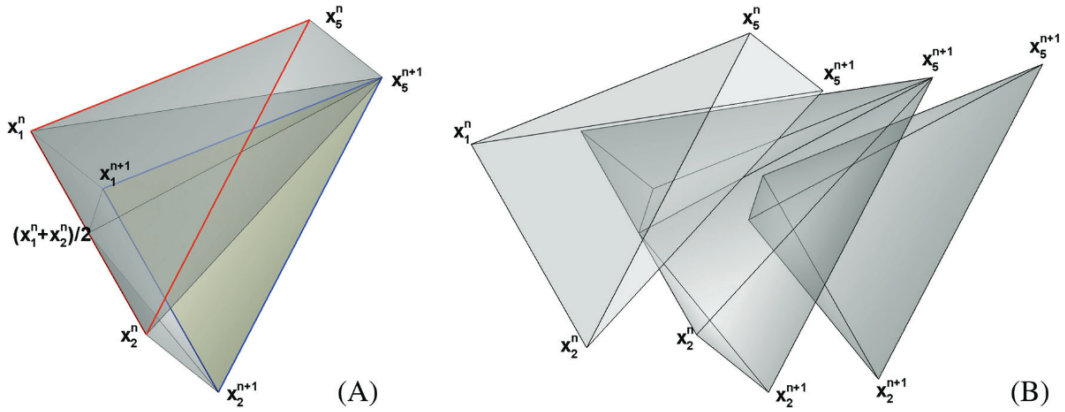

Figure 3.2 : The three-dimensional volume scanned by the area S_{125} is divided into three tetrahedra (B) [4].

though it does not guarantee exact mass preservation in compressible formulation as in the incompressible case, Farhat and colleagues [36] state that DGCL is a necessary and sufficient condition for the nonlinear stability of the ALE algorithm.

In addition, the technique to divide the swept volume into tetrahedras guarantees that the volume is correctly calculated even for non-planar hexahedral elements. With this approach, the velocity of the surface at each interface between two points is calculated. As seen in Figure 3.3, for each interface, it can be easily applied, as each interface can be divided into triangles with three points: the midpoint of the edge intersected by the surface, the midpoint of any surface connected to the edge (red points in Figure 3.3), and the centroids of the volume elements connected to the surface (blue points in Figure 3.3).

3.2.4 High-order interpolation

To achieve high-order solutions, Taylor's expansion based interpolation is applied from the points where state variable vectors are defined to the surfaces of the control volume created around that point (at the midpoint of the edge). Gradients are calculated at each point, and then, using the Taylor expansion as shown below, second-order and higher-order solutions can be obtained. It should be noted that for a third-order solution, the Hessian matrix must be calculated at the points.

$$q_{\mathrm{ij}} = q_i + \nabla q_i \left(\mathbf{r}_{ij} - \mathbf{r}_i\right) + \frac{1}{2}\left(\mathbf{r}_{ij} - \mathbf{r}_i\right)^{\top} \left[\nabla\left(\nabla\left(\mathrm{q_i}\right)\right)\right]\left(\mathbf{r}_{ij} - \mathbf{r}_i\right) + O\left(\Delta r^3\right) \qquad (3.18)$$

In this study, a second-order solution is used and will be explained in detail below.

3.2.4.1 Calculation of Fluxes at Interfaces

As previously explained, the calculation of fluxes on interfaces requires the "left and right" conserved state variable vectors. When performing second-order solution, the vectors of left and right state variables will be calculated using the Taylor series. For example, for the flux on the interface S_{ij} shown in Figure 3.1:

$$\begin{aligned} q_{\mathrm{L}} &= q_i + \nabla q_i \left(\mathbf{r}_{ij} - \mathbf{r}_i\right) \\ q_{\mathrm{R}} &= q_j + \nabla q_j \left(\mathbf{r}_{ij} - \mathbf{r}_j\right) \end{aligned} \tag{3.19}$$

Here, gradients at control points are calculated using the "least squares" or "weighted least squares" methods.

3.2.4.2 Calculation of gradients at vertices

The "least squares" or "weighted least squares" methods can be used for gradient calculation for interpolation. For this, the following overdetermined system of equations is written between one point and its neighboring points where ω_{ij} is the weight.

$$\begin{bmatrix} \omega_{i1}(\mathbf{x}_1 - \mathbf{x}_i) \\ \omega_{i2}(\mathbf{x}_2 - \mathbf{x}_i) \\ \omega_{i3}(\mathbf{x}_3 - \mathbf{x}_i) \\ \omega_{i4}(\mathbf{x}_4 - \mathbf{x}_i) \\ \vdots \\ \omega_{in}(\mathbf{x}_n - \mathbf{x}_i) \end{bmatrix} \begin{bmatrix} \frac{\partial q}{\partial x} \\ \frac{\partial q}{\partial y} \\ \frac{\partial q}{\partial z} \end{bmatrix} = \mathbf{A}_i \mathbf{x} = \mathbf{b}_i = \begin{bmatrix} \omega_{i1}(\mathbf{q}_1 - \mathbf{q}_i) \\ \omega_{i2}(\mathbf{q}_2 - \mathbf{q}_i) \\ \omega_{i3}(\mathbf{q}_3 - \mathbf{q}_i) \\ \omega_{i4}(\mathbf{q}_4 - \mathbf{q}_i) \\ \vdots \\ \omega_{in}(\mathbf{q}_n - \mathbf{q}_i) \end{bmatrix} \tag{3.20}$$

Therefore, the solution can be expressed as follows:

$$\begin{aligned} \left.\frac{\partial q}{\partial x}\right|_i &= \sum_{j=1}^{n} W_{ij}^{x} \left(q_i - q_j\right) \\ \left.\frac{\partial q}{\partial y}\right|_i &= \sum_{j=1}^{n} W_{ij}^{y} \left(q_i - q_j\right) \\ \left.\frac{\partial q}{\partial z}\right|_i &= \sum_{j=1}^{n} W_{ij}^{z} \left(q_i - q_j\right) \end{aligned} \tag{3.21}$$

where the index "j" represents the points neighboring the point "i." Solving this system of equations through matrix-vector operations is prone to round-off errors because multiplying the equation by the transpose of matrix "A" to solve the system squares round-off errors. Fortunately, an analytical solution based on QR iterations has been

proposed in the literature for solving this system [32]. The analytical solution for 2D problems is extended for weighted least squares as follows:

$$\begin{aligned} W_{ij}^{x} &= \omega_{ij}\left(\frac{\Delta x_{ij}}{r_{11}^{2}} - \frac{r_{12}}{r_{11}r_{22}^{2}}\left[\Delta y_{ij} - \Delta x_{ij}\frac{r_{12}}{r_{11}}\right]\right) \\ W_{ij}^{y} &= \omega_{ij}\left(\frac{1}{r_{22}^{2}}\left[\Delta y_{ij} - \Delta x_{ij}\frac{r_{12}}{r_{11}}\right]\right) \end{aligned} \tag{3.22}$$

$$\begin{aligned} r_{11} &= \sqrt{\sum_{j=1}^{N_A}\left(\omega_{ij}\Delta x_{ij}\right)^{2}} \\ r_{12} &= \frac{1}{r_{11}}\sum_{j=1}^{N_A}\omega_{ij}^{2}\Delta x_{ij}\Delta y_{ij} \\ r_{22} &= \sqrt{\sum_{j=1}^{N_A}\omega_{ij}^{2}\left[\Delta y_{ij} - \Delta x_{ij}\frac{r_{12}}{r_{11}}\right]^{2}} \end{aligned} \tag{3.23}$$

And for 3D problems:

$$\begin{aligned} W_{ij}^{x} &= \omega_{ij}\left(\alpha_{ij,1} - \frac{r_{12}}{r_{11}}\alpha_{ij,2} + \beta\,\alpha_{ij,3}\right) \\ W_{ij}^{y} &= \omega_{ij}\left(\alpha_{ij,2} - \frac{r_{33}}{r_{22}}\alpha_{ij,3}\right) \\ W_{ij}^{z} &= \omega_{ij}\left(\alpha_{ij,3}\right) \end{aligned} \tag{3.24}$$

$$\begin{aligned} \alpha_{ij,1} &= \frac{\Delta x_{ij}}{r_{11}^{2}} \\ \alpha_{ij,2} &= \frac{1}{r_{22}^{2}}\left(\Delta y_{ij} - \frac{r_{12}}{r_{11}}\Delta x_{ij}\right) \\ \alpha_{ij,3} &= \frac{1}{r_{33}^{2}}\left(\Delta z_{ij} - \frac{r_{23}}{r_{22}}\Delta y_{ij} + \beta\Delta x_{ij}\right) \\ \beta &= \frac{r_{12}r_{23} - r_{13}r_{22}}{r_{11}r_{22}} \end{aligned} \tag{3.25}$$

$$
\begin{aligned}
r_{11} &= \sqrt{\sum_{j=1}^{N_A} \left(\omega_{ij}\Delta x_{ij}\right)^2} \\
r_{12} &= \frac{1}{r_{11}} \sum_{j=1}^{N_A} \omega_{ij}^2 \Delta x_{ij}\Delta y_{ij} \\
r_{22} &= \sqrt{\sum_{j=1}^{N_A} \left(\omega_{ij}\Delta y_{ij}\right)^2 - r_{12}^2} \\
r_{13} &= \frac{1}{r_{11}} \sum_{j=1}^{N_A} \omega_{ij}^2 \Delta x_{ij}\Delta z_{ij} \\
r_{23} &= \frac{1}{r_{22}} \left(\sum_{j=1}^{N_A} \omega_{ij}^2 \Delta y_{ij}\Delta z_{ij} - \frac{r_{12}}{r_{11}} \sum_{j=1}^{N_A} \omega_{ij}^2 \Delta x_{ij}\Delta z_{ij} \right) \\
r_{33} &= \sqrt{\sum_{j=1}^{N_A} \left(\omega_{ij}\Delta z_{ij}\right)^2 - \left(r_{13}^2 + r_{23}^2\right)}
\end{aligned}
\tag{3.26}
$$

It is a common approach to use least-square approach to interpolate the state vector. However, as pointed out in literature [37], it suffers from accuracy on highly anisotropic meshes. However, as Mavriplis states [37], the interpolated values can still be used when flow aligns with the grid points. This alignment happens in the boundary layer when it is constructed using inflation layers. However, for arbitrarily anisotropic meshes, it is recommended to use weighted least-square with the weights [38]

$$
\omega_{ij} = \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|^t}. \tag{3.27}
$$

Increasing values of t in Equation 3.27 leads to slow convergence. Therefore, it is recommended to use $t \in [0,2]$ [39].

In HEMLAB, we generally use least-square to calculate the gradient for reconstruction. However, some experiences with reconstruction schemes agree with the literature. Weighted least square gives more stable and robust solutions. However, it degrades in the convergence of the performance parameters.

3.2.4.3 Venkatakrishnan gradient limiting (slope limiter)

Taylor's expansion based interpolation is performed by calculating gradients for high-order solutions. In cases where there are very high gradients (for example, in the direction perpendicular to a shock wave and in the case of discontinuity) and the grid resolution is not sufficiently fine to capture this high frequency solutions, oscillations

can occur, and these oscillations can even lead solution to diverge. To prevent this, gradients are limited by a local maximum of the state variables. The limiting operation ensures that the highest interpolated value is restricted between the local minimum and the local maximum values. This ensures preserving the monotonicity. The limiting function, Φ, modifies the interpolation equation as follows:

$$q_{ij} = q_i + \Phi_i \nabla q_i \left(\mathbf{r}_{ij} - \mathbf{r}_i\right) \tag{3.28}$$

In the HEMLAB solver, the Venkatakrishnan limiter with the Wang modification is used, and this method is explained below.

$$\Phi_i^{VK} = \min_{ij \in \{c_i\}} \begin{cases} \phi^{VK}(y), & q_i \neq q_{ij} \\ 1, & q_i = q_{ij} \end{cases} \tag{3.29}$$

$$\phi^{VK}(y) = \frac{y^2 + 2y + \frac{\varepsilon^2}{\left(q_{ij} - q_i\right)^2}}{y^2 + y + 2 + \frac{\varepsilon^2}{\left(q_{ij} - q_i\right)^2}} \tag{3.30}$$

In the formulation, when the value of $q_{ij} - q_i$ approaches zero numerically, the limitier function produces erroneous results. Thus, ϕ^{VK} is multiplied by the term $\left(q_{ij} - q_i\right)^2$ in both the numerator and the denominator. Here, c_i is the set of midpoints of edges connected to point i, and y is defined as:

$$y = \begin{cases} \frac{q_{\min} - q_i}{q_{ij} - q_i}, & q_i \geq q_{ij} \\ \frac{q_{\max} - q_i}{q_{ij} - q_i}, & q_i < q_{ij} \end{cases} \tag{3.31}$$

Here, $q_{\min}$ and $q_{\max}$ are the minimum and maximum q values among neighboring points of vertex i:

$$q_{\min} = \min_{j \in \{N_i\}} q_j, q_{\max} = \max_{j \in \{N_i\}} q_j \tag{3.32}$$

$\{N_i\}$ is the set of neighboring points of point i. Additionally, the value q_{ij} in the above formulation corresponds to the unbounded interpolated value, i.e.

$$q_{ij} = q_i + \nabla q_i \left(\mathbf{r}_{ij} - \mathbf{r}_i\right) \tag{3.33}$$

The ε parameter is the Wang modification and is calculated following Wang's modification:

$$\varepsilon = K_v \left(q_{\text{gmax}} - q_{\text{gmin}}\right) \tag{3.34}$$

In this equation, q_{gmax} and q_{gmin} are the minimum and maximum values over the entire solution domain globally. However, in the original formulation of Venkatakrishnan limiter, ε parameter is calculated as

$$\varepsilon^2 = (K_v \Delta x)^3 \tag{3.35}$$

where $\Delta x = 2\sqrt{V_i/\pi}$.

3.3 Calculation of Viscous Fluxes

HEMLAB solver employs a technique that exploits the Green-Gauss Gradient method to calculate gradient at the interfaces. Although there is a much simpler technique to calculate the gradients that uses Weighted Least-Square approach and the Green-Gauss method is relatively more expensive, it prevents odd-even decoupling problem. Additionaly, the viscous jacobian can be fully obtained, making the numerical algorithm more robust to reach higher numbers of CFL.

Using Green-Gauss theorem, gradient integrated over a volume is reduced to integration of the same variable along the surface of dual-volume. Dual volume construction is explained as below.

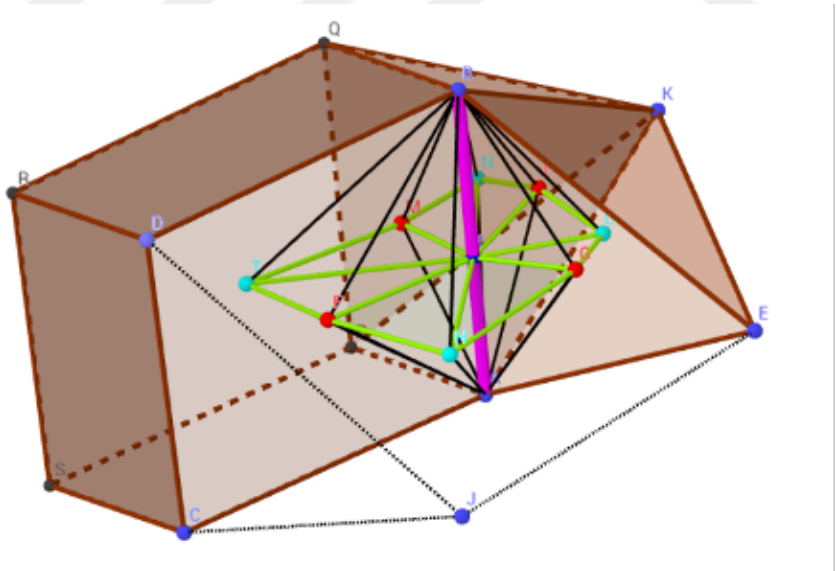

Figure 3.3 : The dual volume constructed around the edge on polyhedral mesh.

In the Figure 3.3, green edges determines the interface between nodes of *pink* edge. Centroids of every faces that shares the edge is marked with *red* points whereas the centroids of the cells that share the edge are marked with *blue* points. Connecting them in one direction determines the interface.

The dual volume is created by connecting the vertices that determine interfaces and and the vertices of the edge on which the gradient will be calculated. Therefore, one of the points of the of *pink* edge, the centroid of a face that shares the edge and centroid of a cell of which shares the face determines a face of the dual volume. For example, the face BFH is one of them. Therefore, the gradient of any variable can be written as a surface integral over the boundaries of dual volume such as

$$\nabla\phi_{ij} = \frac{1}{V_{d_{ij}}} \oint \phi \mathbf{n} dS \tag{3.36}$$

where $V_{d_{ij}}$ is the dual volume and S is the surface enclosing the dual volume. Numerically, this would become

$$\nabla\phi_{ij} = \frac{1}{V_{d_{ij}}} \sum_{\mathbf{k}\in S(ij)} \phi_k \mathbf{n}_\mathbf{k} S_k \tag{3.37}$$

where $S(ij)$ is the surfaces that enclose the dual volume around the edge ij. To obtain values at the center points of surfaces of the dual volume, we interpolate the values from vertices. Since dual volume is constructed using the vertices that include element centroids and face centroids, the values are interpolated from the vertices of elements and faces. For example, for a point at the element centroid, values at the nodes of that element are used to interpolate. The same notion is applied for faces. Therefore, the formulation of Green-Gauss gradient implies that the gradient of a variable can be written as a weighted average of the values at the vertices that constructs the cells sharing the edge on which the gradient is calculated. This is shown in Equation 3.38

$$\begin{aligned}
\partial\phi/\,\partial x|_{AB} &= \frac{1}{V^{(d)}} \sum_{i=1}^{N} W_i^x \phi_i \\
\partial\phi/\,\partial y|_{AB} &= \frac{1}{V^{(d)}} \sum_{i=1}^{N} W_i^y \phi_i \\
\partial\phi/\,\partial z|_{AB} &= \frac{1}{V^{(d)}} \sum_{i=1}^{N} W_i^z \phi_i
\end{aligned} \tag{3.38}$$

where N is number of the nodes of cells that shares the edge, ϕ is just a scalar variable and $V^{(d)}$ is the dual-volume.

In HEMLAB algorithm, gradients of the primitive variables are calculated in accordance with the Green-Gauss approach explained above and they are used in Equations 2.17 and 2.19. Once the weights of each node that is in the neighbouring of an edge are calculated and stored, the viscous flux and Jacobian can be calculated.

This brings an advantage in terms of speed of the algorithm with the expense of use of additional RAM.

3.3.1 Jacobian of the viscous fluxes

The total viscous flux is calculated by integrating the gradient over the surfaces of the control volume shaped around a point, as explained above. The Jacobian of the viscous flux can be calculated as follows.

For the x-momentum equation:

$$\begin{aligned}
&= \sum \tau_{xx} S_x^c + \tau_{xy} S_y^c + \tau_{xz} S_z^c \\
&= \sum \mu \left[(2+\lambda^*) \, \partial u/\partial x + \lambda^* \partial v/\partial y + \lambda^* \partial w/\partial z \right] S_x^c + \mu [\partial u/\partial y + \partial v/\partial x] S_y^c \\
&+ \mu [\partial u/\partial z + \partial w/\partial x] S_z^c
\end{aligned} \tag{3.39}$$

For the y-momentum equation:

$$\begin{aligned}
&= \sum \tau_{yx} S_x^c + \tau_{yy} S_y^c + \tau_{yz} S_z^c \\
&= \sum \mu [\partial u/\partial y + \partial v/\partial x] S_x^c + \mu \left[\lambda^* \partial u/\partial x + (2+\lambda^*) \, \partial v/\partial y + \lambda^* \partial w/\partial z \right] S_y^c \\
&+ \mu [\partial v/\partial z + \partial w/\partial y] S_z^c
\end{aligned} \tag{3.40}$$

For the z-momentum equation:

$$\begin{aligned}
&= \sum \tau_{zx} S_x^c + \tau_{zy} S_y^c + \tau_{zz} S_z^c \\
&= \sum \mu [\partial u/\partial z + \partial w/\partial x] S_x^c + \mu [\partial v/\partial z + \partial w/\partial y] S_y^c \\
&+ \mu \left[\lambda^* \partial u/\partial x + \lambda^* \partial v/\partial y + (2+\lambda^*) \, \partial w/\partial z \right] S_z^c
\end{aligned} \tag{3.41}$$

And for the energy equation:

$$\begin{aligned}
&= \sum \left[u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\partial T/\partial x \right] S_x^c \\
&+ \left[u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\partial T/\partial y \right] S_y^c \\
&+ \left[u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k^{\partial T}/\partial z \right] S_z^c \\
&= \sum \left[u \left[(2+\lambda^*) \, \partial u/\partial x + \lambda^* \partial v/\partial y + \lambda^* \partial w/\partial z \right] + v[\partial u/\partial y + \partial v/\partial x] + w[\partial u/\partial z + \partial w/\partial x] \right. \\
&\left. + k^* \partial T/\partial x \right] \mu S_x^c \\
&+ \left[u[\partial u/\partial y + \partial v/\partial x] + v \left[\lambda^* \partial u/\partial x + (2+\lambda^*) \, \partial v/\partial y + \lambda^* \partial w/\partial z \right] + w[\partial v/\partial z + \partial w/\partial y] \right. \\
&\left. + k^* \partial T/\partial y \right] \mu S_y^c \\
&+ \left[u[\partial u/\partial z + \partial w/\partial x] + v[\partial v/\partial z + \partial w/\partial y] + w \left[\lambda^* \partial u/\partial x + \lambda^* \partial v/\partial y + (2+\lambda^*) \, \partial w/\partial z \right] \right. \\
&\left. + k^* \partial T/\partial z \right] \mu S_z^c
\end{aligned} \tag{3.42}$$

Here, $k^* = \frac{M_\infty}{(\gamma-1)\mathrm{Re}_\infty \mathrm{Pr}_\infty}\mu^*$ and the gradients, as explained earlier, are the gradients of the fundamental variables calculated using the Green-Gauss method. Then, the following transformation is applied to obtain the conserved variables and the Jacobian.

Conserved variables:

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \tag{3.43}$$

Primitive variables:

$$\mathbf{W} = \begin{bmatrix} \rho \\ u \\ v \\ w \\ T \end{bmatrix} \tag{3.44}$$

The Jacobian transformation is as follows:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} = \frac{\partial \mathbf{R}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{Q}} \tag{3.45}$$

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}} = \begin{bmatrix} \frac{\gamma-1}{2}\left(u^2+v^2+w^2\right) & -(\gamma-1)u & -(\gamma-1)v & -(\gamma-1)z & (\gamma-1) \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ \frac{\gamma(\gamma-1)}{\rho}\left(u^2+v^2+w^2-\frac{E}{\rho}\right) & -\gamma(\gamma-1)\frac{u}{\rho} & -\gamma(\gamma-1)\frac{v}{\rho} & -\gamma(\gamma-1)\frac{w}{\rho} & \frac{\gamma(\gamma-1)}{\rho} \end{bmatrix} \tag{3.46}$$

3.4 Discretization in Time Dimension

In the HEMLAB solver, first-order backward discretization is applied for time discretization.

$$\frac{\partial}{\partial t}\left(\mathbf{Q}_i V_i\right) = \frac{\left(\mathbf{Q}_i V_i\right)^{n+1} - \left(\mathbf{Q}_i V_i\right)^n}{\Delta t_p} \tag{3.47}$$

Here, Δt_p is the physical time step.

3.5 Boundary Conditions

For the Euler equations, slip and farfield boundary conditions are imposed in HEMLAB. On the other hand, for RANS simulations, noslip conditions are imposed.

3.5.1 Slip boundary condition

On the slip boundary, we impose

$$\mathbf{n} \cdot (\mathbf{u} - \dot{\mathbf{x}}) = 0. \tag{3.48}$$

So that, the slip boundary flux becomes

$$\mathbf{F}^{Slip} \cdot \mathbf{n} = \begin{bmatrix} 0 \\ pn_x \\ pn_y \\ pn_z \\ p(\mathbf{n} \cdot \dot{\mathbf{x}}) \end{bmatrix} \cdot S \tag{3.49}$$

3.5.2 Noslip boundary condition

Noslip kinematic boundary condition is strongly imposed using the relations below instead of the momentum equations.

$$\mathbf{u} - \dot{\mathbf{x}} = 0 \tag{3.50}$$

The jacobian of the equation is inserted into the matrix. Note that there are pressure flux and viscous term contributions to the energy equation due to the moving boundary. The pressure flux term added to the energy equation is same as the slip boundary condition. Viscous flux contribution to the energy equations must be evaluated same as the internal edges.

3.5.3 Farfield boundary conditions

In HEMLAB solver, two different implementations for farfield boundary conditions are employed as explained in this section.

3.5.3.1 Weak-Riemann farfield-value

The weak-Riemann ∞-value farfield boundary condition is imposed in the code. In this approach, left state variables are determined as the state variable of corresponding vertex. The right state vector comes from the prescribed freestream conditions. So that, the flux on the farfield can be evaluated as

$$\mathbf{F}^{Farfield} = \mathbf{H}(\mathbf{Q}_L, \mathbf{Q}_\infty) \tag{3.51}$$

where $\mathbf{H}$ is an approximate Riemann solver such as Roe and HLLC.

3.5.3.2 Weak-prescribed Riemann invariant

In this technique, inflow and outflow conditions are identified and flux is calculated in order to preserve Riemann invariants. In order to do that, right state vectors are modified in a procedure explained below [40].

The Riemann invariants on moving grid given in 2.1.2 are determined as

$$R_+ = V_{in} + \frac{2c_{in}}{\gamma - 1}, \quad R_- = V_\infty - \frac{2c_\infty}{\gamma - 1} \tag{3.52}$$

where subindice *in* indicates the internal node, ∞ is freestream conditions and c is the speed of sound. The velocities V_{in} and V_∞ are

$$\begin{aligned} V_{in} &= \mathbf{n} \cdot (\mathbf{u}_{in} - \dot{\mathbf{x}}) \\ V_\infty &= \mathbf{n} \cdot (\mathbf{u}_\infty - \dot{\mathbf{x}}) \end{aligned}. \tag{3.53}$$

The Riemann invariants are modified, If $(V_\infty + c_\infty < 0)$ then

$$R_+ = V_\infty + \frac{2c_\infty}{\gamma - 1} \tag{3.54}$$

and if $(V_\infty - c_\infty > 0)$, then

$$R_- = V_{in} - \frac{2c_{in}}{\gamma - 1} \tag{3.55}$$

After that, on the boundary

$$V_b = \frac{1}{2}(R_+ + R_-). \tag{3.56}$$

If it is an outflow $(V_b > 0)$,

$$\begin{aligned} \mathbf{u}_b &= \mathbf{u}_{in} + (V_b - V_{in})\,\mathbf{n} \\ s_b &= \frac{c_{in}^2}{\gamma \rho_{in}^{\gamma - 1}} \end{aligned} \tag{3.57}$$

otherwise,

$$\begin{aligned} \mathbf{u}_b &= \mathbf{u}_\infty + (V_b - V_\infty)\,\mathbf{n} \\ s_b &= \frac{c_\infty^2}{\gamma \rho_\infty^{\gamma - 1}} \end{aligned} \tag{3.58}$$

Therefore, the right state vector can be obtained as

$$\mathbf{Q}_R = \left\{ \begin{array}{c} \rho_b \\ \rho_b \mathbf{u}_b \\ E_b \end{array} \right\} \tag{3.59}$$

where

$$\rho_b = \left(\frac{c_b^2}{\gamma s_b} \right), \quad p_b = \frac{\rho_b c_b^2}{\gamma} \tag{3.60}$$

3.6 Solution Of The Equation System In Implicit Form

In the final state, the following nonlinear system of equations appears:

$$\frac{(\mathbf{Q}_i V_i)^{n+1} - (\mathbf{Q}_i V_i)^n}{\Delta t_p} + \sum_{j \in N(i)} \mathbf{n}_{ij} \cdot \left[\mathbf{F}^{\mathbf{I}}_{ij}(\mathbf{Q}) - \mathbf{F}^{\mathbf{V}}_{ij}(\mathbf{Q}) \right] S_{ij} = 0 \tag{3.61}$$

As it is known, iterative methods are used for the solution of non-linear equation systems. The HEMLAB solver uses the Newton's iteration method. The Newton's iteration scheme for the system of equations is written as follows:

$$\left(\frac{V_i^m}{\Delta t_p} \mathbf{I} + \mathbf{J} \right) \left(\mathbf{Q}^{m+1} - \mathbf{Q}^m \right) = - \left[\frac{(\mathbf{Q}_i V_i)^m - (\mathbf{Q}_i V_i)^n}{\Delta t_p} + \sum_{j \in \mathbf{N}(i)} \mathbf{n_{ij}} \cdot \left[\mathbf{F}^{\mathbf{I}}_{ij}(\mathbf{Q}^m) - \mathbf{F}^{\mathbf{V}}_{ij}(\mathbf{Q}^m) \right] S_{ij} \right] = -\mathbf{R}_i(\mathbf{Q}^m) \tag{3.62}$$

Here, $\mathbf{I}$ is the identity matrix, and $\mathbf{J} = \frac{\partial \mathbf{R}_i(\mathbf{Q}^m)}{\partial \mathbf{Q}^m}$. By solving Equation 3.62, $\Delta \mathbf{Q}^{m+1} = \left(\mathbf{Q}^{m+1} - \mathbf{Q}^m \right)$ is obtained. When $\Delta \mathbf{Q}^{m+1}$ converges to zero, which means the convergence of $\mathbf{R}_i(\mathbf{Q}^m)$ to zero, the value of $\mathbf{Q}^{m+1}$ converges to the value of $\mathbf{Q}^{n+1}$.

On the other hand, Equation 3.62 has a weakness. As Δt_p increases, the contribution of $\frac{V_i^m}{\Delta t_p}$ to the matrix diagonal decreases. If the contribution from the stabilization term of convective flux in the equation is also low, in which case the convection term discretization will approach second-order central discretization, the matrix diagonal will approach zero for high Reynolds numbers. This makes the numerical solution of the matrix-vector system difficult, even impossible, because iterative methods and preconditioners generally perform diagonal division. In addition, diagonal dominance is another problem. Therefore, with this solution method, a solution can only be obtained with sufficiently small time steps. To overcome this, Jameson proposes a method based on a pseudo-time step called "Dual Time Stepping" [26]. However, the application of dual time stepping introduces a mismatch between the right-hand side of the equation and the Jacobian. To address this, a technique called "Residual Smoothing" can be applied.

After all these considerations, the resulting equation is solved using the PETSc library. The PETSc library [25] is a parallel scientific computing library that includes many different matrix-vector operations. In the solution of the equation system, the additive Schwarz preconditioner is used, and in each block, ILU(0) is used in three dimensions,

and ILU(4) is used in two dimensions as a preconditioner. The FGMRES Krylov subspace method is used for the solution of the matrix-vector system.

3.6.1 Dual-time stepping

In the very basis of the technique, It just adds a term to the diagonal of the matrix. This makes the solution of matrix-vector system more robust. Therefore, the algorithm allows higher numbers of physical time steps, hastening the convergence to steady-state. With this additional term on diagonal, the linear equation of Newton's iterations becomes

$$\left(\frac{V_i^m}{\Delta t_p}\mathbf{I}+\frac{V_i^m}{\Delta t_{CFL}}\mathbf{I}+\mathbf{J}\right)\Delta\mathbf{Q}^{m+1}=-\mathbf{R}_i\left(\mathbf{Q}^m\right). \tag{3.63}$$

The pseudo-time step is calculated locally by adopting Courant-Freiedrich-Lewy (CFL) number. This process is explained in Section 3.6.2.

Besides all of the advantages of the Dual-Time technique, It degrades the convergence for small CFL numbers. Since the right hand-side of the Iteration is not mathched with the Jacobian in the left hand-side, the convergence behaviour of exact Newton's iteration can not be preserved. Thus, these types of techniques are called as *quasi-Newton Iterations*. In these types of techniques, the pseudo-time step should approach to infinity to preserve desirable convergence characteristics of the scheme.

3.6.2 Courant-Friedrichs-Lewy (CFL) number

As previously mentioned, local time stepping approach is applied to reach state-state fast, and therefore, the Δt_{CFL} value is different in each control volume, and this value will be calculated using the CFL number. Understanding the CFL number is necessary for its calculation.

While the understanding of the CFL number is relatively straightforward for the Finite Differences method, various approachs can be encountered in the literature for Finite Volumes. From the perspective of Finite Differences, the CFL number is related to how far a wave leaving a point can travel within the solution step. In Finite Volume, it represents the proportion of volume that the wave covers in addition to its relationship with the distance traveled.

In the Finite Volumes method, two different CFL calculations can be made. Considering only the convective terms, the first one calculates the time that it takes for the waves originated at the midpoints of each control volume to reach the surface of that volume, similar to the method used in Finite Differences. The second method calculates the proportion of volume covered by the waves on the surface of each volume. Therefore, the CFL number calculated with the first method can be given as follows:

$$CFL = \frac{\max\left(\lambda_i\right) \cdot \Delta t_{CFL}}{\min\left(l_{ij}\right)} \tag{3.64}$$

Here, λ_i represents the wave speeds, and l_{ij} is the distance between the vertex " i " and the midpoint of the surfaces. Therefore, if CFL $= 1$ is satisfied, this equation guarantees that the wave does not exit the control volume.

In the second method, the CFL number can be calculated as follows:

$$CFL = \frac{\Delta t_{CFL}}{V_i} \sum_{j \in N(i)} \max\left(\lambda_{ij}\right) \cdot S_{ij} \tag{3.65}$$

The difference here is that λ_{ij} is the speed of the waves on the control volume surfaces. Thus, the average wave speeds in the flux scheme are used. S_{ij} represents the area of the relevant surface of the control volume. It has been observed that the second method accelerates convergence in HEMLAB so far, and this method is also used in this study. It is reminded here that the calculated time step using these approaches considers only the convective terms. Therefore,

$$\Delta t_{CFL}^{C} = \frac{CFL \cdot V_i}{\sum_{j \in \mathbf{N}(i)} \max\left(\lambda_{ij}\right) \cdot S_{ij}} \tag{3.66}$$

is calculated.

On the other hand, If RANS equations are solved, there are also viscous terms. A time step based on the viscous terms can be calculated as follows:

$$\Delta t_{CFL}^{V} = \frac{V_i^2}{2\sum_{j \in \mathbf{N}(i)} \left(\lambda_{ij}^{V}\right) \cdot S_{ij}^2} \tag{3.67}$$

where

$$\lambda_{ij}^{V} = \left(\frac{4}{3}\frac{\mu_{ij}}{\rho_{ij}} + \frac{\gamma}{Pr}\right)\frac{M_\infty}{Re_\infty} \tag{3.68}$$

Therefore, time step is calculated as

$$\Delta t_{CFL} = \min(\Delta t^{C}_{CFL}, \Delta t^{V}_{CFL}). \tag{3.69}$$

3.6.3 Residual smoothing

To address the problem mentioned in Section 3.6.1, a term should be added to the right-hand side of the equation so that the matrix on the left hand-side can be obtained when its derivative is taken, making the exact Newton's iteration as

$$\left(\frac{V_i^m}{\Delta t_p}\mathbf{I} + \frac{V_i^m}{\Delta t_{CFL}}\mathbf{I} + \mathbf{J}\right)\Delta\mathbf{Q}^{m+1} = -\mathbf{R}_i(\mathbf{Q}^m) + \frac{V_i^m}{\Delta t_{CFL}}\left(\mathbf{Q}^{m+1} - \mathbf{Q}^m\right) \tag{3.70}$$

However, $\left(\mathbf{Q}^{m+1} - \mathbf{Q}^m\right)$ is unknown. Fortunately, there are methods to approximate this value. One such method is the Residual Smoothing technique proposed by Mavriplis [41].

According to this technique, the term $\left(\mathbf{Q}^{m+1} - \mathbf{Q}^m\right)$ can be approximated as follows:

$$\mathbf{Q}^{m+1} - \mathbf{Q}^m \approx -\mathbf{D}^{-1}\mathbf{R}_i(\mathbf{Q}^m) \tag{3.71}$$

Here, $\mathbf{D}^{-1}$ is the inverse of the relevant diagonal block of the Jacobian matrix. Thus, the equation system to be solved in the final step is transformed into the following form:

$$\left(\frac{V_i^m}{\Delta t_p}\mathbf{I} + \frac{V_i^m}{\Delta t_{CFL}}\mathbf{I} + \mathbf{J}\right)\Delta\mathbf{Q}^{m+\mathbf{1}} = -\mathbf{R}_i(\mathbf{Q}^m) - \frac{V_i^m}{\Delta t_{CFL}}\mathbf{D}^{-1}\mathbf{R}_i(\mathbf{Q}^m) \tag{3.72}$$

Mavriplis also emphasized that the term added to the right side guarantees a decrease in the error (residual) in each Newton iteration [41]. Therefore, it serves as a line-search direction in Newton's iterations. In addition, it is called some kind of smoothing because the term added actually corresponds to one-step Block-Jacobi solution.

3.7 Grid Adaptation and Use of pyAMG

Interesting studies subjecting mesh optimization shows that the effect of grid can be more significant than we tend to think [42, 43]. It is clear in these studies to conclude that very slight move of problematic vertices can improve the stability of Finte Volume schemes. Even, it may lead the convergence of diverging solution. As well, what is so amazing about the studies is that very small numbers of vertices in the domain

have been moved to improve the stability. Therefore, it is clear that CFD solutions are highly dependent on grid structures. Especially in cases where the physics of the phenomenon needs to be resolved, the grid structure may become very complex as high resolutions are required at different locations. Since it is not deterministically known where resolution of grid are needed to be refined or coarsened *a priori*, the grid structure is usually prepared as an estimate in a human-dependent process. As a result, this can lead to unnecessary grid vertices or undesirable results. In addition, an iterative process becomes inevitable in the mesh generation phase. Therefore, it is crucial to adapt the grid structure a posteriori and modify it along with the solution. The technique called mesh adaptation allows to obtain accurate solutions using fewer points and to obtain solutions adapted to the problem being solved, as indicated in the literature [44]. In this study, disadvantages arising from the complex nature of the flow around the rotor will be resolved with the help of grid adaptation. The pyAMG library is used as the grid adaptation algorithm. For this purpose, a Python script connecting the HEMLAB and pyAMG libraries has been written.

The pyAMG library [7, 45] is an anisotropic grid adaptation library. This library controls the interpolation error for a given solution parameter. The minimization of this error arises as a term related to the determinant of the parameter's Hessian matrix, expressing a metric on the Riemann manifold. The goal of the algorithm is to create a triangle consisting of unit-length edges on the manifold determined by this metric. The parameter determining the Hessian matrix is called the sensor. Theoretically, the sensor can be any parameter, but in practice, it varies depending on the problem being solved. For example, Mach number can be a good parameter for capturing shock waves [44]. In a study by Fidkowski and Roe [46], entropy is mentioned as a general purpose sensor variable. In this study, Mach number and entropy are chosen as sensor variables. Since the flow values in the farfield region of the flow field are very small, rounding errors can lead to large values in the Hessian matrix. Therefore, to prevent grid refinement in this region, the sensor value is multiplied by a function dependent on the distance from the wall. The sensor functions used in the study are shown below.

$$Mach \quad sensor = M \times e^{-0.1r} + 0.5 \tag{3.73}$$

$$\begin{aligned} Entropy \quad sensor &= e^{-0.1r} \times \Delta s + 0.5 \\ \Delta s &= \frac{R_g}{\gamma - 1} \ln \frac{T}{T_\infty} - R_g \ln \frac{\rho}{\rho_\infty} \end{aligned} \tag{3.74}$$

Here, "s" is entropy, and M represents the Mach number. R_g and γ are the gas constant and the ratio of specific heat capacities. T and ρ are temperature and density, and when used with the ∞ index, they correspond to values in the farfield.

Another important aspect for grid adaptation is the preservation of surface geometry because the adaptation algorithm also makes adaptations on the surface grid. Therefore, it is necessary to ensure that the adapted surface grid preserves the original geometry. In SU2, the pyAMG library disrupts the geometry on the surface. To preserve the geometry, the preservation of edges on the surface is necessary. For this purpose, it is generally recommended to make adaptations on the surface by dividing it into as many parts as possible. However, even if edges are preserved, there may be indentations and protrusions on the surface between the edges.

To overcome this difficulty, the "*adap_back*" option of the pyAMG library is used. This option allows the newly created surface grid width to be projected onto the surface grid in the provided file. For this, the Gamma ".mesh" file format is used because this format can store edge and corner information. Most of the mesh generation programs do not write this information to the grid file. The "*adap_back*" option requires a surface grid file with a dense grid structure on the surface; it is printed in MeshLink format to obtain edge information. MeshLink is an open-source library used for the projection of grid files onto geometry (CAD). Later, with another Python script, this information is extracted and added to the grid file obtained in the ".mesh" format. In this way, the grid file is given to the pyAMG code with the "*adap_back*" option, ensuring that the surface grid is projected onto the original geometry in each adaptation.

4. VALIDATION OF THE ALGORITHM ON A 2D PROBLEM

As part of the validation study, heaving NACA 0012 solutions proposed under the Fourth High Order Computational Fluid Dynamics Workshop (HiOCFD4) were utilized as well as the heaving and pitching NACA 0012 solutions. For both of these problems, RANS equations are solved on moving domains and the motion equations are predefined on the configuration shown in Figure 4.1. The grids are rigidly moved in these problems. In both problems, the flow was initially solved at a 0° angle of attack. The state vector for this solution is specified with a Mach number of 0.2 as follows:

$$[\rho, \rho u, \rho v, E] = \left[1, 1, 0, 0.5 + \frac{1}{M^2 \gamma(\gamma - 1)}\right] \tag{4.1}$$

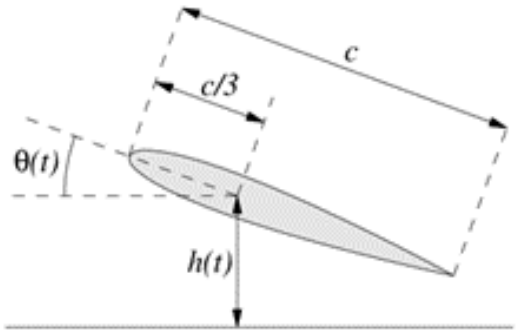

Figure 4.1 : NACA 0012 motion frame.

4.1 Heaving NACA 0012 Airfoil

The mesh motion defined for the Heaving NACA0012 is shown below.

$$h(t) = \frac{t^2(3-t)}{4} \tag{4.2}$$

Upon solving the problem, the force coefficients were compared with the results obtained in the HiOCFD4 workshop, and a good agreement between the results was observed, as depicted in the Figures 4.2. In addition, vorticity contours that are shown in Figure 4.3 matches with the results in literature [47].

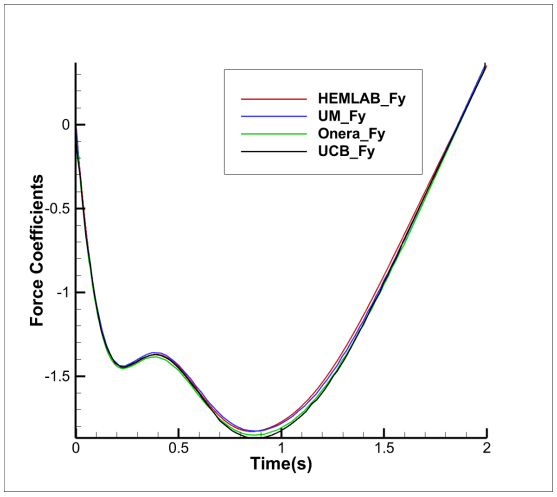

Figure 4.2 : Comparison of results for heaving airfoil case between different codes including HEMLAB.

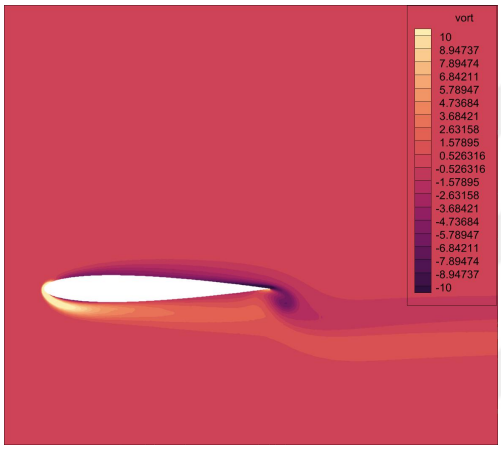

Figure 4.3a : $t = 0.5s$

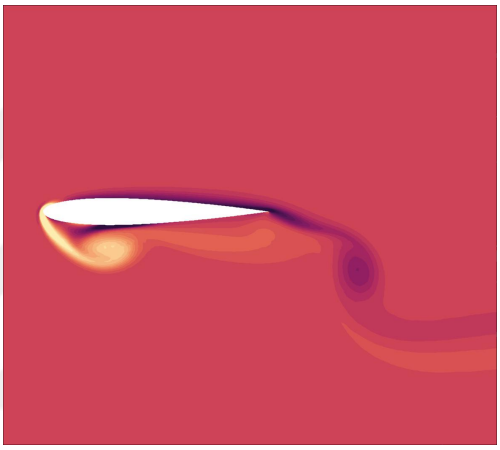

Figure 4.3b : $t = 1.0s$

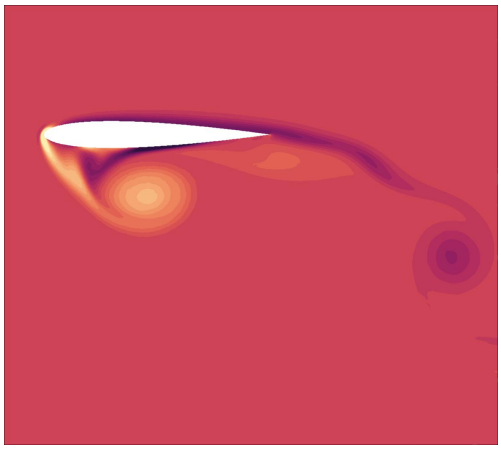

Figure 4.3c : $t = 1.5s$

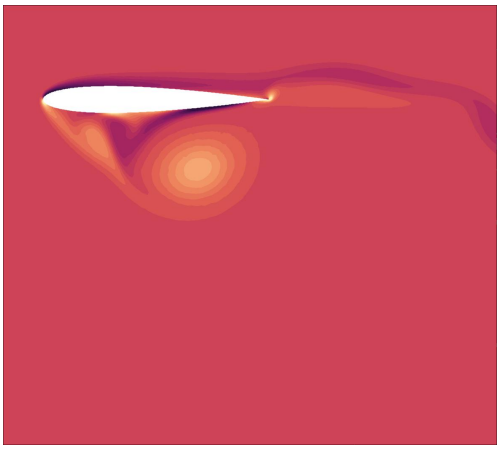

Figure 4.3d : $t = 2.0s$

Figure 4.3 : Vorticity contours for heaving NACA 0012.

4.2 Heaving and Pitching NACA 0012 Airfoil

The motion equations for the airfoil undergoing heaving and pitching motion are provided below. Similar to the previous problem, solutions for this case also exhibit compatibility with other codes as shown Figures 4.4 and the vorticity contours are shown in Figure 4.5.

$$h(t) = \frac{t^2(3-t)}{4} \tag{4.3}$$

$$\theta(t) = \frac{\pi}{3}\left(t^4 - 4t^3 + 4t^2\right) \tag{4.4}$$

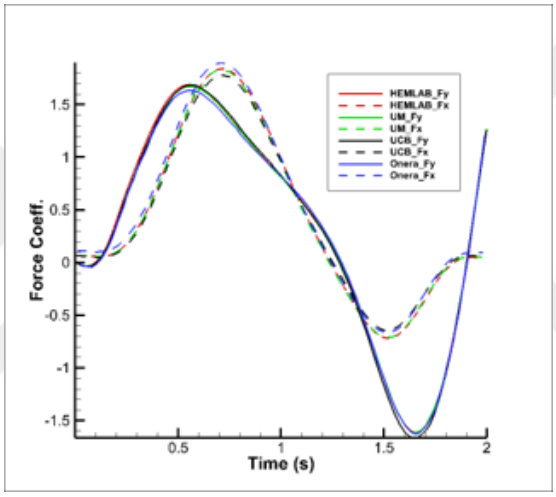

Figure 4.4 : Comparison of results for heaving and pitching airfoil case between different codes including HEMLAB.

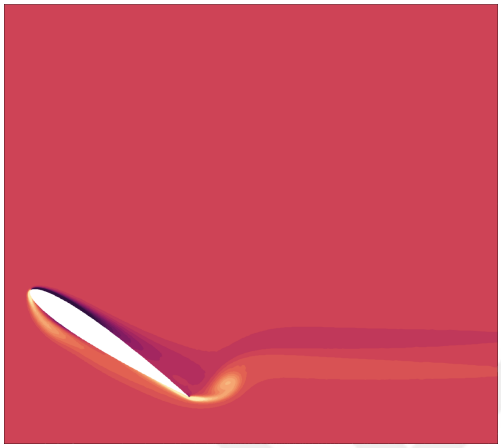

Figure 4.5a : $t = 0.5s$

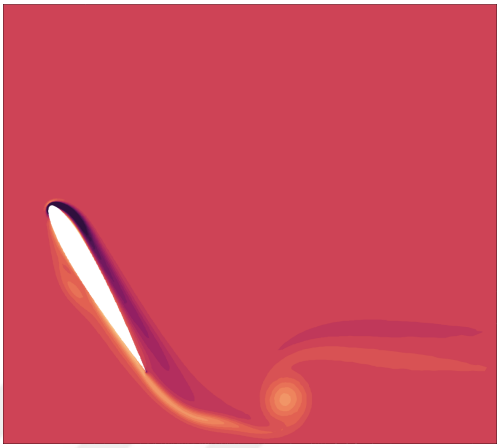

Figure 4.5b : $t = 1.0s$

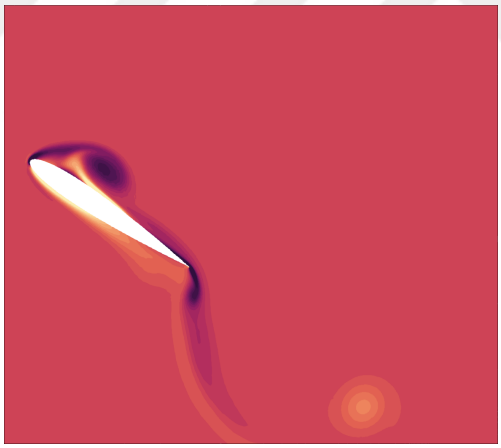

Figure 4.5c : $t = 1.5s$

Figure 4.5 : Vorticity contours for heaving and pitching NACA 0012.

5. SOLUTION OF THE FLOW FIELD AROUND THE S76 MAIN ROTOR USING EULER EQUATIONS

The current study focuses on simulating the S-76 main rotor with a swept-tapered tip, using the specifications specified by AIAA Hover Prediction Workshop (HPW). The rotor geometry file is provided on the HPW website.

5.1 Geometry of The S76 Main Rotor Blade

The rotor is equipped with three distinct airfoils over its span: SC1013R8, SC1095R8, and SC1095. The taper ratio and sweep angle of the tip area are also displayed in Table 1. The Figure 5.1 illustrates the geometry of the rotor blade and the distribution of twist over its span.

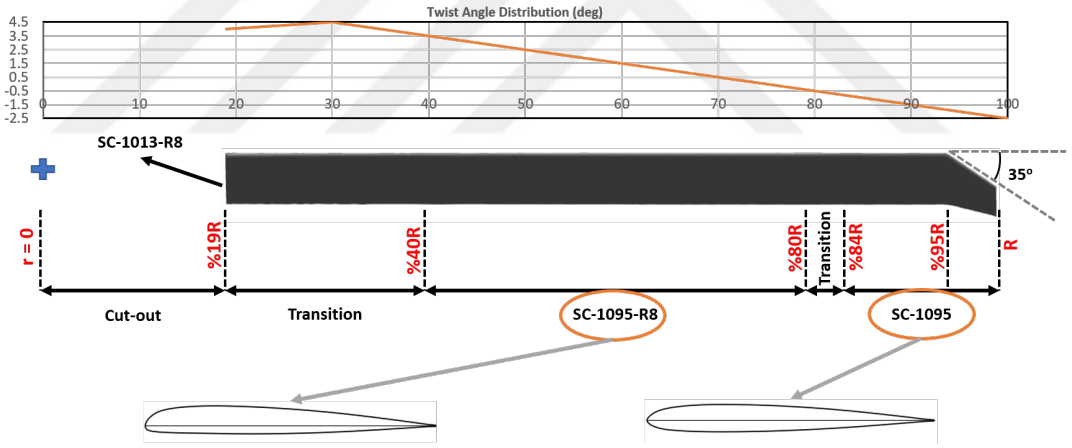

Figure 5.1 : S76 Blade geometry.

5.2 Baseline Mesh

The grid structure has a direct impact on the findings. As mentioned before, this effect can become apparent in a sophisticated flow analysis. However, the present analyses employ anisotropic mesh adaption. Therefore, by employing rather simple mesh topologies during the early phase (baseline solution before the adaptation), it is

possible to attain a high levels of accuracy. The following passages gives information about baseline mesh structures.

The unstructured mesh of the blade surface is depicted in Figure 5.2. The span contains a total of 235 points, including 59 chord-wise vertices at the root and 44 at the tip. The span-wise points are frequently aligned at the root and tip to effectively capture the vortices formed at the tip and root. Surface capture is challenging to do with unstructured grids due to the high twist rate and varying shapes along the span. This difficulty can only be overcome by using an excessive number of nodes. Consequently, we generated surface meshes resembling a boundary layer originated from both the front and back edges. These are elongated along the chord.

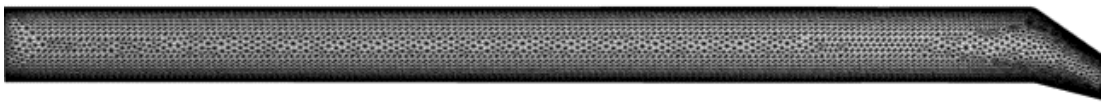

Figure 5.2 : S76 Blade surface mesh for baseline solution.

Figure 5.3 illustrates the volume mesh topology. The spacings progressively increase as one moves further away from the object. The near-body zone is characterized by a cylindrical shape with a height of 1.3 times the radius (R) and a diameter of 2 times the radius (2R). A constant spacing of 0.015 times the radius (R) is employed in this zone. The region located away from the body, with a distance of 0.2 times the radius (R), also demonstrates a cylindrical form. Currently, the cylinder has a height of 5.4 times the radius (R) and a diameter of 6 times the radius (R). The spacings at the farfield have a value of R, where the farfield cylinder extends 40R in the longitudinal direction and 20R in the radial direction. Consequently, the total number of vertices contained within the computational volume is 1,043,679.

5.3 Analysis Conditions

The rotor was used for analysis at 3.5° cone angle and 9° collective angle. The Mach number at the tip is 0.65, while the Reynolds number is 1.18 million. So far, only Euler analyses have been performed, and Mach and entropy sensor functions, Roe and HLLC schemes, have been tested. A total of three different simulations were performed. Full convergence was not achieved due to the use of "limiter". In the simulations, the rotor rotated a total of 21 revolutions. The first simulation used the Mach sensor function

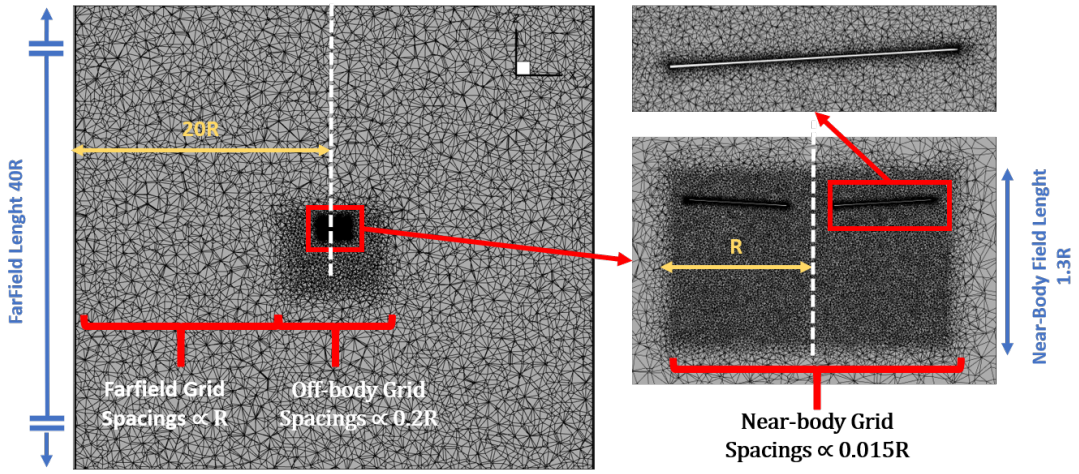

Figure 5.3 : S76 volume mesh for baseline solution.

and the Roe scheme. In the second simulation, the Mach sensor function and the HLLC scheme are used. In the third simulation, we used the Entropy sensor. The third simulation started with Roe as a convevtive scheme, but continued with HLLC as the simulation diverged. The reason behind this is tried to be explained in the following sections.

5.4 Baseline Solution

Initially the baseline mesh was subjected to a simulation for 10 revolutions (equal to 10,000 time steps) in order to obtain converged solutions before the refinement process occurred. This phase was crucial in facilitating the dissipation of the initial vortex ring. The vortex structures corresponding to the initial solution are shown in Figure 5.4. In the initial rotation, the formation of the vortex ring structure occurs and progressively moves beneath the rotor. However, after the tenth revolution, it weakens and disappears because of the dissipative impact of the coarse mesh in the initial setup. Notwithstanding this dissipation, the root vortex remains present until the tenth revolution due to the low values of induced velocity at the root.

5.5 Simulation Results

The first simulation utilized the Roe flux scheme, whereas the mesh refinement method incorporated the Mach sensor function. The adaption size (complexity) for the initial four refinement levels was established at 400,000. Following that, the subsequent four

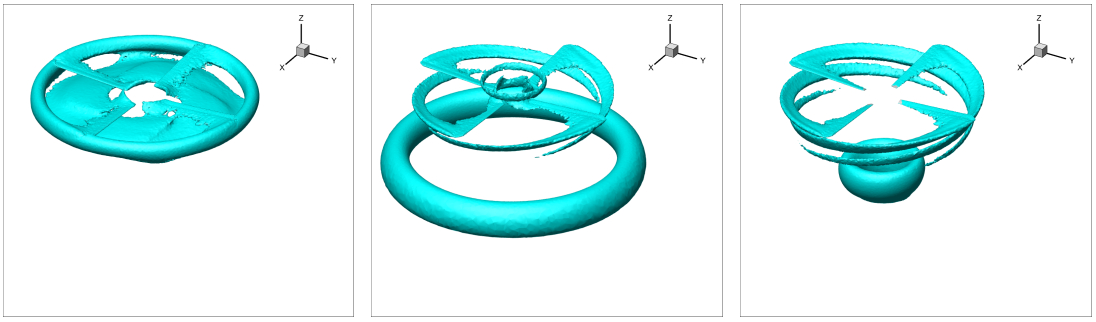

Figure 5.4 : Initial vortex ring structures (Revolution 1, 5 and 10 from left to right).

refining stages utilized a complexity of 800,000, whereas the last two levels employed a complexity of 2,400,000. The final iteration resulted in a total of 8,249,524 nodes. As anticipated, the Mach sensor successfully detected the tip vortices, but, the wake region was not adequately resolved. Figure 5.5 and 5.6 exhibits the vortex formations and wake region, whilst Figure 5.7 illustrates the mesh structures created from this simulation.

Figure 5.5 : Simulation 1 (Mach sensor and Roe scheme) vortex structures (Revolution 11, 15 and 21 from left to right).

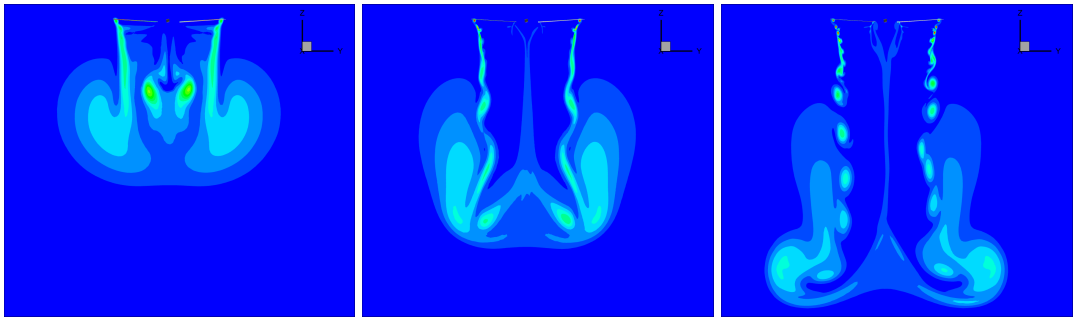

Figure 5.6 : Simulation 1 (Mach sensor and Roe scheme) wake region (Revolution 11, 15 and 21 from left to right).

A strong rarefaction wave emerges at the rotor's tip, as the flow attempts to curl upwards from the lower surface to the upper surface of the rotor. The Roe method permits the occurrence of expansion shocks and, as a result, does not satisfy the condition of entropy. To resolve this problem, it is necessary to implement an entropy

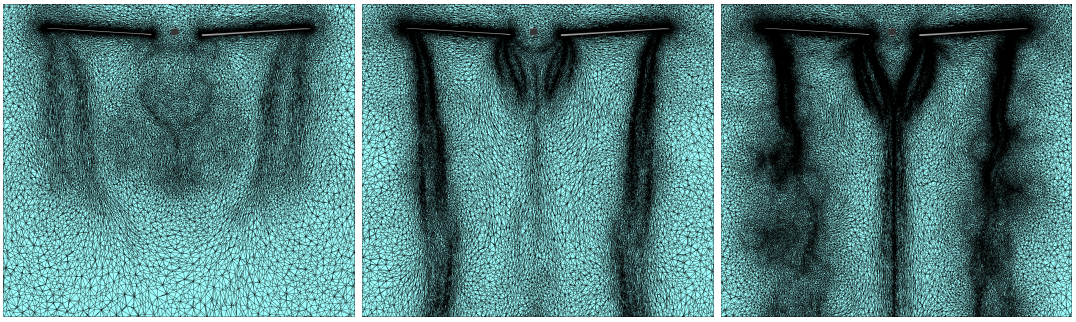

Figure 5.7 : Simulation 1 (Mach sensor and Roe scheme) mesh topology (Revolution 11, 15 and 21 from left to right).

fix in conjunction with the Roe scheme. The HEMLAB Solver applies the Harten's entropy correction. Nevertheless, due to the strong rarefaction discontinuity and the lack of positivity preservation in the Roe scheme, a negative density was detected in the expansion zone. To successfully complete the first simulation, we enhanced the impact of Harten correction. In the second simulation, we employed the HLLC scheme, renowned for its ability to preserve positivity [34, 48]. As a result, the problem of negative density, which was seen in the first simulation, did not occur while employing the HLLC technique.

Figure 5.8, 5.9 and Figure 5.10 illustrate the results and grid layouts produced from simulation 2. The same adaptation sizes as in simulation 1 were used, except for the last three levels which had an adaptation size of $2,400,000$. At the final iteration, the number of nodes reached $9,870,963$.

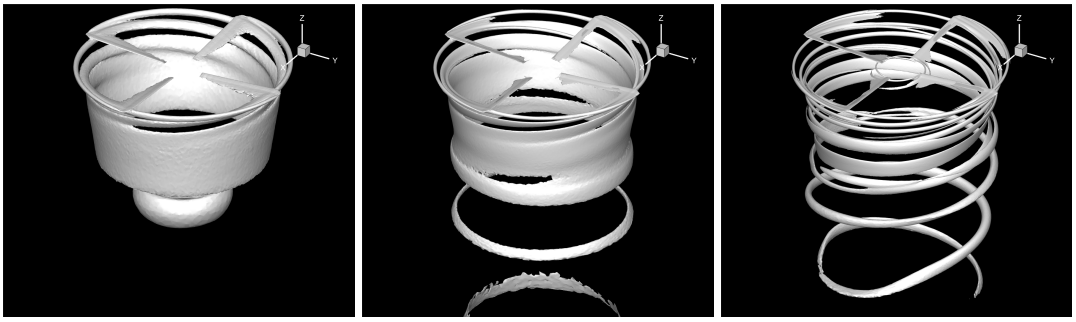

Figure 5.8 : Simulation 2 (Mach sensor and HLLC scheme) vortex structures (Revolution 11, 15 and 21 from left to right).

The third simulation initially utilized the Roe scheme with the entropy sensor function. Nevertheless, this combination resulted in non-physical solutions as a consequence of the negative density issue. Unfortunately, the Harten entropy correction was unable to remedy the issue. In response to this, a transition to the HLLC scheme became

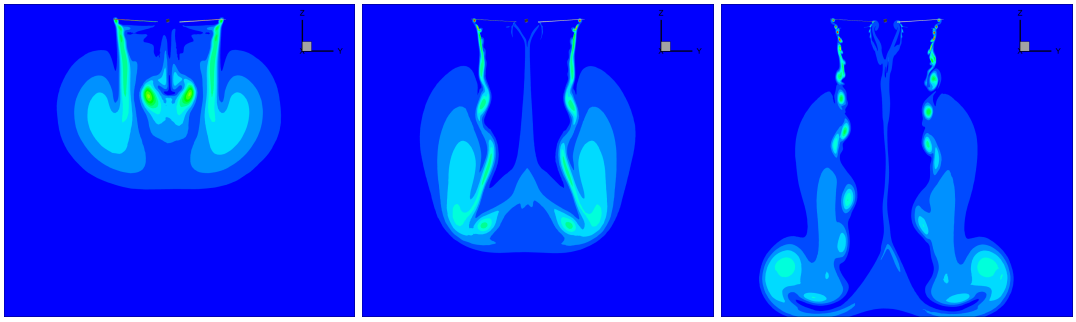

Figure 5.9 : Simulation 2 (Mach sensor and HLLC scheme) wake region (Revolution 11, 15 and 21 from left to right).

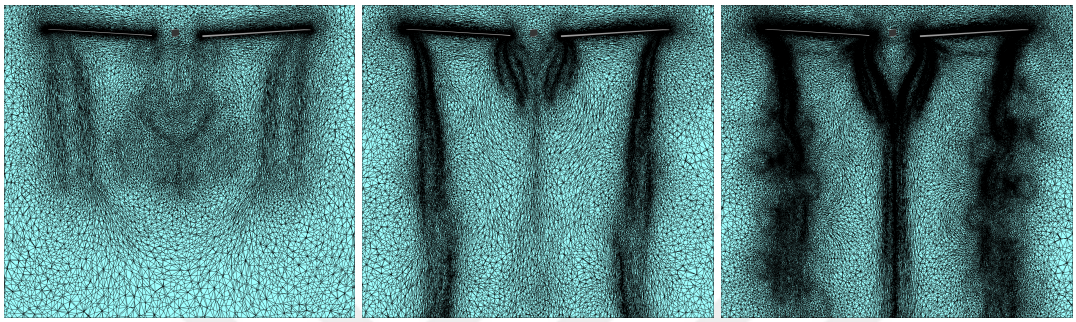

Figure 5.10 : Simulation 2 (Mach sensor and HLLC scheme) mesh topology (Revolution 11, 15 and 21 from left to right).

essential after the 15th revolution. The mesh structures and vortex formations can be noticed in Figures 5.11, 5.12 and 5.13, respectively.

The refinement levels in this simulation were built in a manner analogous to simulation 2, and at the concluding time step, the domain encompassed a total amount of $23,127,097$ nodes. The entropy sensor effectively detected the wake sheet at the trailing edge, resulting in a notably greater number of nodes compared to the other two simulations. The unique feature of the entropy sensor can detect both the trailing edge wake sheet and tip vortices by employing adaptive mesh refinement distinguishes this study from previous studies. According to prior research [2], the entropy sensor has been found to be successful in detecting the wake layer.

As seen in the Figure 5.14, a shock wave forms at the tip and moves with the rotor blade. The surface mesh and mesh slice shows that adaptive mesh refinement captures the shockwave.

Figure 5.15 depicts the progression of the performance parameters (thrust and torque coefficients) over the time steps. Remarkably the thrust coefficients produced from the Mach sensor simulations show a stronger correspondence with the experimental

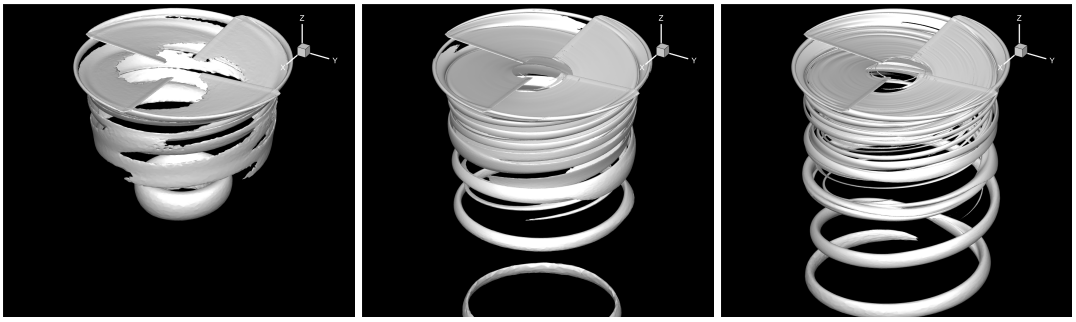

Figure 5.11 : Simulation 3 (entropy sensor and Roe/HLLC scheme) vortex structures (Revolution 11, 15 and 21 from left to right).

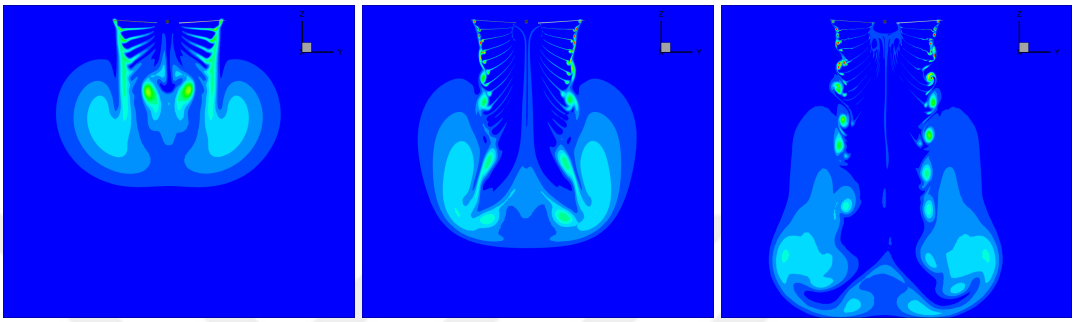

Figure 5.12 : Simulation 3 (entropy sensor and Roe/HLLC scheme) wake region (Revolution 11, 15 and 21 from left to right).

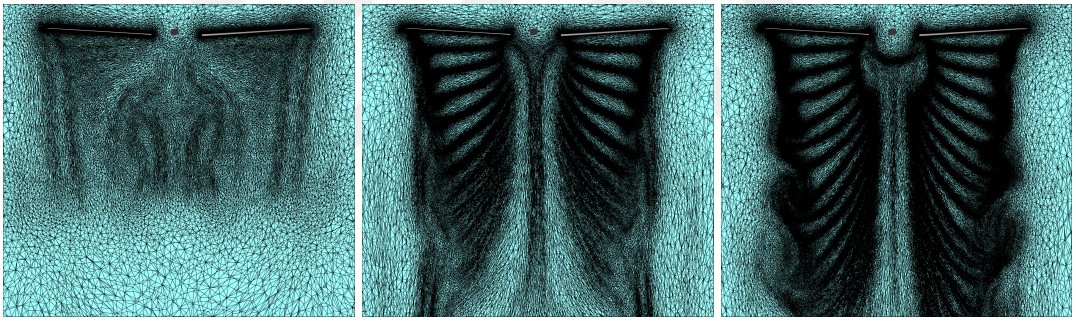

Figure 5.13 : Simulation 3 (entropy sensor and Roe/HLLC scheme) mesh structures (Revolution 11, 15 and 21 from left to right).

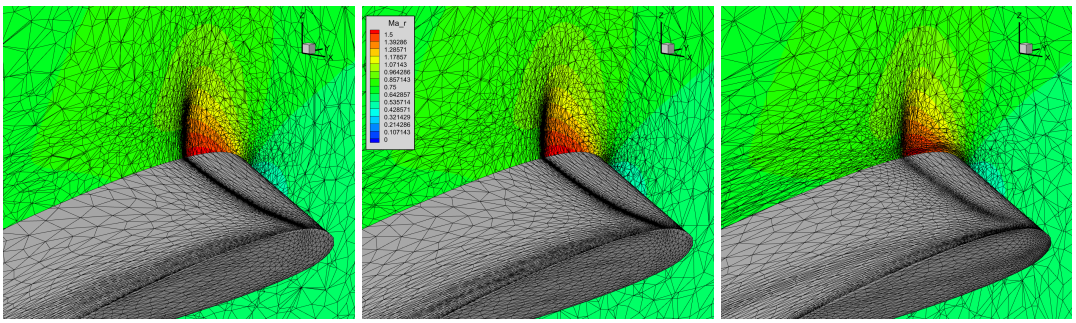

Figure 5.14 : Relative Mach contours of shock wave at the tip (Simulation 1, 2 and 3 from left to right).

results obtained from Balch and Lombardi's work [12]. Nevertheless, the improved agreement between the entropy sensor simulations and the experimental data does

not apply to the torque coefficient. It is important to mention that the lack of RANS simulations at this point makes it too early to conclusively conclude which sensor is the most optimal. An optimal strategy could potentially entail the integration of entropy and Mach sensors. The main benefit of the entropy sensor is its effective capture of the wake zone. However, it may face difficulties in capturing velocity gradients that are located further away from the body. As a result, this can lead to the creation of coarse meshes in potential flow regions. Conversely, the Mach sensor has superior performance in these areas.

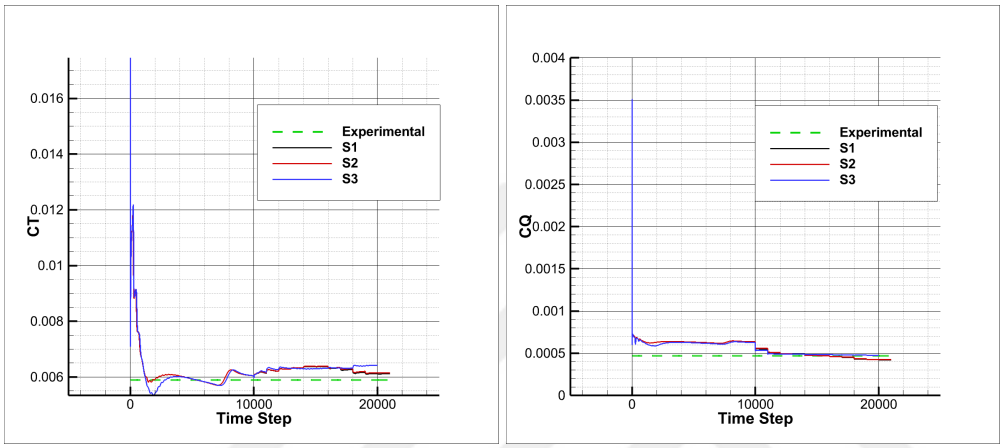

Figure 5.15 : Performance parameter comparison of simulations.

5.5.1 Effect of HLLC scheme

The Figure 5.16 shows the tip of the rotor on which strong rarefaction wave takes place. As seen in Figure 5.16, Roe scheme gives rise to a discontinuity in the expansion region whereas it is not observed with HLLC. Arguably, this might be the reason behind negative density and the divergence of the solution. However, since there is no sufficient evidence, we have to be careful to conclude. However, the difference between the two schemes are clear on the figure.

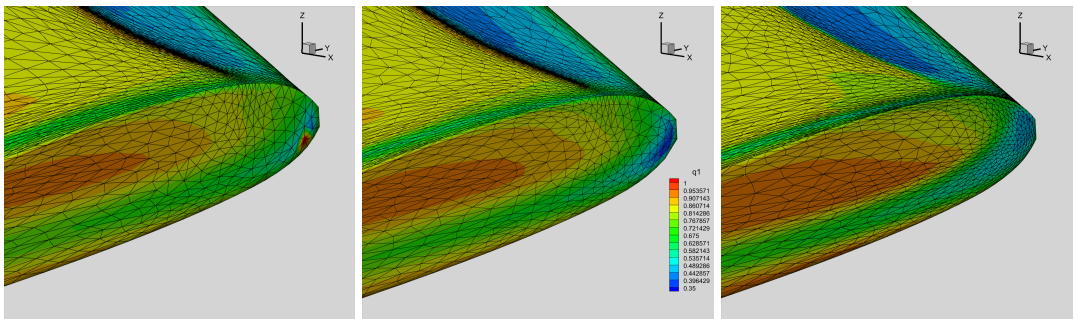

Figure 5.16 : Density contours of expansion region at the tip (Simulation 1, 2 and 3 from left to right).

6. SOLUTION OF THE FLOW FIELD AROUND S76 MAIN ROTOR USING RANS EQUATIONS

In this section, RANS equations are solved to simulate flow around S76 main rotor. The geometry of the rotor was explained in the Section 5.1. The RANS mesh is also generated in a similar fashion as in Section 5.2. However, the surface mesh is modified and the surface mesh for RANS simulatins are shown in Figure 6.1.

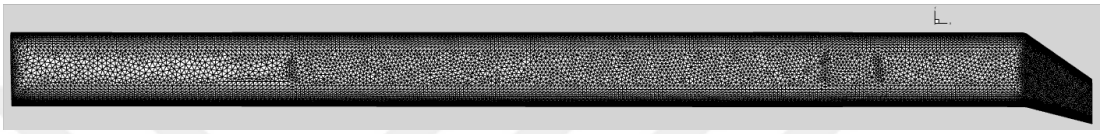

Figure 6.1 : S76 Blade surface mesh for baseline solution.

The near body mesh is shown in Figure 6.2 with boundary layer mesh. The first layer height in boundary layer is specified as 10^{-6} to achieve y^+ under unity. Including the boundary layer, all elements are tetrahedral in the domain.

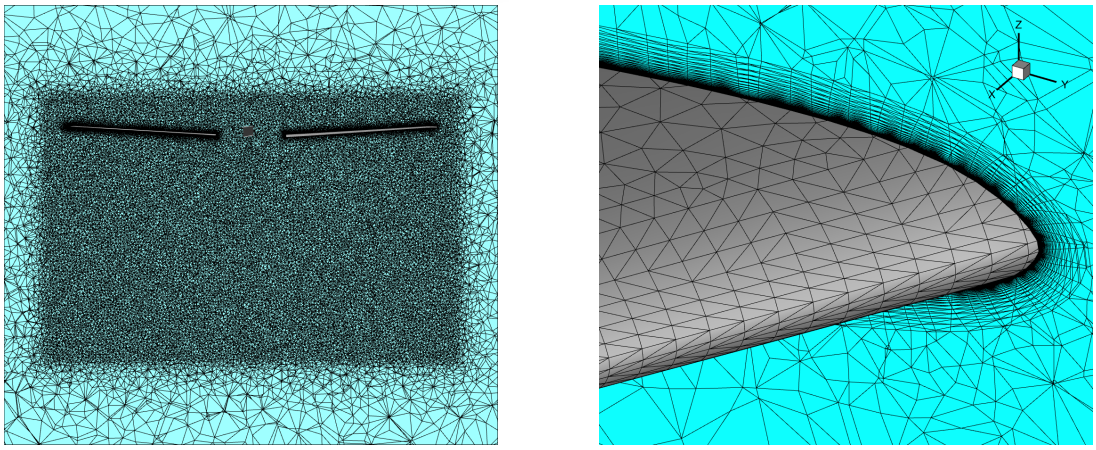

Figure 6.2 : S76 near body mesh (left) and boundary layer mesh (right) for RANS equations.

6.1 Analysis Conditions

The rotor is practised when Reynolds number is $1.18 Million$ and Mach number is 0.65 at the tip. Roe scheme is utilised in the simulation with Jacobians calculated by Tapenade. 10 Newton's iterations are performed to reach steady-state between each

physical time step. The CFL number is set to 10 during the simulations. Similar to the Euler analysis, one revolution is divided into 1000 physical time steps. In addition, we applied a new scaling in order to obtain fast convergence in lineer solver. Convergence of stationary preconditioners depends on the eigenvalues of the matrix. When eigenvalues are lower than unity, convergence of stationary methods are guaranteed. Using this idea, we scaled matrix in order to bound all eigenvalues with unity. This can be guaranteed by making sum of absolute values of elements in a row lower than 1.

6.2 Simulation Results

Tip vortices are shown in Figure 6.3. Since mesh resolution is not sufficient, vortices look like sheet. The same can be seen in Figure 6.4 for wake region.

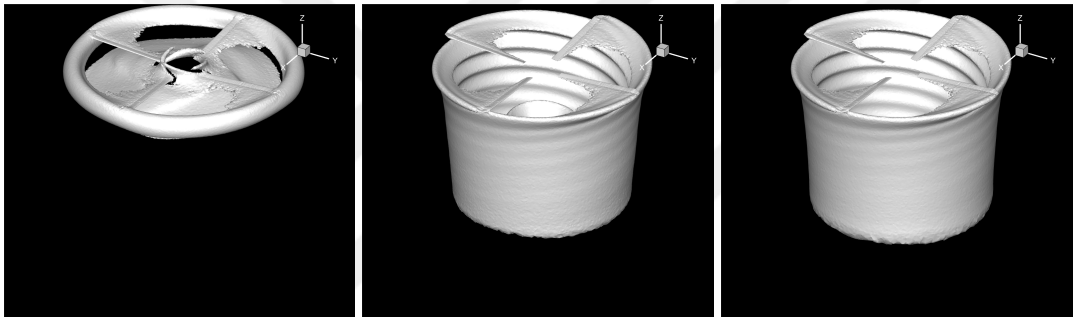

Figure 6.3 : RANS simulation vortex structures (Revolution 1, 10 and 21 from left to right).

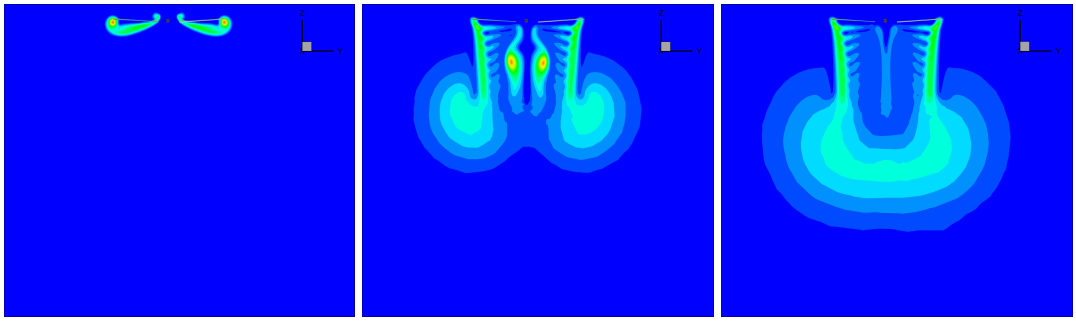

Figure 6.4 : RANS simulation wake region entropy contours (Revolution 1, 10 and 21 from left to right).

We experienced that solution depends highly on boundary layer mesh. Using prism elements in the boundary layer, the solutions can be obtained without any problem. However, using tetrahedral elements, solution was not obtained unless DGCL is satisfied. Even satisfying DGCL is not enough to obtain physical solutions. On highly anisotropic meshes, RANS solutions include very high densities (higher than 5 times

the freestream value). In addition, application of the new scaling explained in the previous section reduced the number of lineer solver iterations by half.

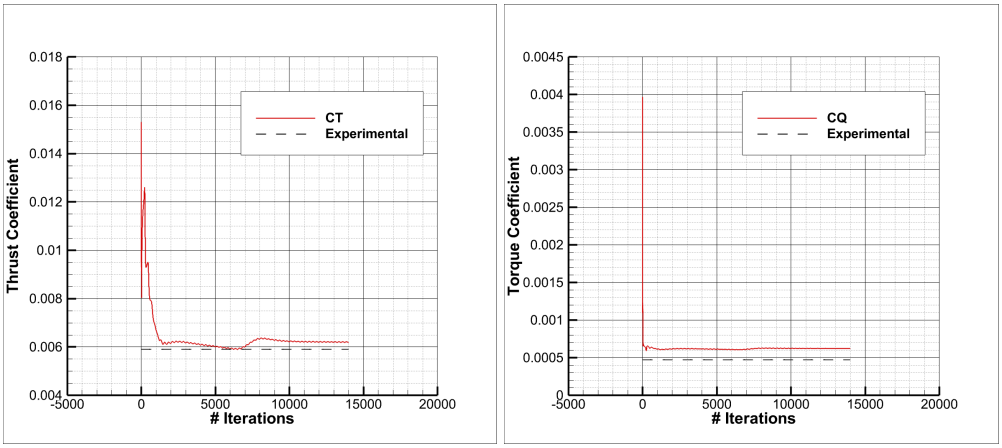

Figure 6.5 : Performance parameter comparison of RANS simulation and experimental values.

As seen in the Figure 6.5, mesh resolution is obviously not sufficient to correctly predict the performance parameters. Therefore, mesh adaptation or a refined fixed mesh is required.

7. CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

In the present study, an Arbitrary Lagrangian-Eulerian approach has been developed to simulate compressible flow problems. Capabilities of HEMLAB solver is extended to incorporate fluxes due to grid movement. New flux schemes were added with novel Jacobian implementations. To enhance the convergence of Newton's subiterations, CFL calculations, Jacobian constructions and the efficiency of preconditioners were revisited. Residual smoothing approach was applied. Due to preconditioner effectivity issues, a new scaling of the linear equation system was tried and approved on RANS solutions. The Jacobians of Roe and the Jacobian calculated using Tapenade automatic differentiation library was compared. It was observed during the study that accurate calculations of Jacobian is key to the solution of matrix-vector system.

To simulate complex physics with adequate accuracy, mesh adaptation have been said to be necessary. In order to do anisotropic mesh adaptation, HEMLAB solver was integrated with pyAMG library. To preserve the original surface shape, HEMLAB and interface between HEMLAB and pyAMG was modified to deal with *.mesh* format which includes edge and corner informations. Using a well refined mesh on surface, pyAMG library projects adapted mesh onto the original one.

The developed algorithm and techniques were validated on 2D airfoil case which is proposed under HiOCFD4 Workshop. The results were promising. Then, the main focus of the study was tried to be resolved. The main rotor of S76 helicopter is practised under hover conditions that is proposed under AIAA Hover Prediction Workshop. Firstly, Euler simulations were conducted usign anisotropic mesh adaptation. However, due to the limiters, the solutions were not converged to steady. However, the results were very close to the previous researchs. Then, RANS simulations were conducted on a fixed mesh.

7.2 Future Works

- The current formulations deals with only the rigid mesh motion. Mesh deformation algorithm should be extended to deal with arbitrary deformations.
- Since mesh adaptation creates highly anisotropic meshes, the solution can be very difficult on these meshes. And it is hard to adjust the regions that is needed to be refined using sensor functions explicitly. RANS solutions are even harder because the boundary layer formation can be very difficult usign adaptation. Therefore, optimal solution would be to develop our own mesh adaptation algorithm to appropriately adjust and generate suitable meshes.
- Euler simulations can be re-done using DGCL and weighted least square based interpolation as proposed in the literature. Since we deal with highly anisotropic meshes, flow is not aligned with the grid. Therefore, least square interpolation may lead erroneous results.
- RANS simulations will be continued using Adaptive Mesh Refinement.

REFERENCES

[1] **Abras, J. and Hariharan, N.S.** (2015). Comparison of CFD hover predictions on the S-76 rotor, *53rd AIAA Aerospace Sciences Meeting*, p.1711.

[2] **Aksoy, E. and Sahin, M.** (2022). An Application of Anisotropic Mesh Refinement to Solve Flow Around the S-76 Main Rotor with Swept-Tapered Tip, *AIAA SciTech 2022 Forum*, p.1319.

[3] **Donea, J.**, **Huerta, A.**, **Ponthot, J.P. and Rodríguez-Ferran, A.** (2004). Arbitrary Lagrangian-Eulerian Methods, *Encyclopedia of computational mechanics*.

[4] **Guventurk, C. and Sahin, M.** (2022). A mass conserving arbitrary Lagrangian–Eulerian formulation for three-dimensional multiphase fluid flows, *International Journal for Numerical Methods in Fluids*, *94*(4), 346–376.

[5] **Karali, H.**, **Inalhan, G.**, **Umut Demirezen, M. and Adil Yukselen, M.** (2021). A new nonlinear lifting line method for aerodynamic analysis and deep learning modeling of small unmanned aerial vehicles, *International Journal of Micro Air Vehicles*, *13*, 17568293211016817.

[6] **Hirsch, C.** (1988). Numerical computation of internal and external flows, Volume 1: Fundamentals of numerical discretization, *John Wiley and Sons*, *9*, 10.

[7] (2018), pyAMG, `https://pyamg.saclay.inria.fr/pyamg.html`.

[8] **Chaderjian, N.M.** (2012). Advances in rotor performance and turbulent wake simulation using DES and adaptive mesh refinement, *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*.

[9] **Jain, R.** (2016). A comparison of CFD hover predictions for the Sikorsky S-76 rotor, *54th AIAA Aerospace Sciences Meeting*, p.0032.

[10] **Jain, R.** (2015). Hover predictions for the S-76 rotor with tip shape variation using CREATE-AV Helios, *53rd AIAA Aerospace Sciences Meeting*, p.1244.

[11] **Tadghighi, H.** (2014). Helios simulation of rotors in hover: The Boeing company, *52nd Aerospace Sciences Meeting*, p.0209.

[12] **Balch, D. and Lombardi, J.** (1985). Experimental study of main rotor tip geometry and tail rotor interactions in hover. Volume 1. Text and figures.

[13] **Narducci, R.** (2015). Hover performance assessment of several tip shapes using OVERFLOW, *53rd AIAA Aerospace Sciences Meeting*, p.1243.

[14] **Johnson, W.** (1980). Performance and loads data from a wind tunnel test of a full-scale rotor with four blade tip planforms, **Technical Report**, NATIONAL AERONAUTICS AND SPACE ADMINISTRATION MOFFETT FIELD CA AMES RESEARCH CENTER.

[15] **SHINODA, P. and Johnson, W.** (1993). Performance results from a test of an S-76 rotor in the NASA Ames 80-by 120-foot wind tunnel, *11th Applied Aerodynamics Conference*, p.3414.

[16] **Berker, R.** (1982). An exact solution of the navier-stokes equation the vortex with curvilinear axis, *International Journal of Engineering Science*, *20*(2), 217–230, `https://www.sciencedirect.com/science/article/pii/0020722582900179`.

[17] **Grundy, R. and McLaughlin, R.** (1999). Three-dimensional blow-up solutions of the Navier-Stokes equations, *IMA Journal of Applied Mathematics*, *63*(3), 287–306.

[18] **Tao, T.**, (2020), Quantitative bounds for critically bounded solutions to the Navier-Stokes equations, `1908.04958`.

[19] **Buresti, G.** (2015). A note on Stokes' hypothesis, *Acta Mechanica*, *226*, 3555–3559.

[20] **Xu, J.**, **Ma, D.**, **Liu, P.**, **Bi, L.**, **Yuan, X. and Chen, L.**, (2022), Corrected Navier-Stokes equations for compressible flows, `2212.12642`.

[21] **Skinner, D.**, (2015), Lecture notes in Mathematical Methods, `https://www.damtp.cam.ac.uk/user/dbs26/1Bmethods.html`.

[22] **LeVeque, R.J.** (1992). *Numerical methods for conservation laws*, volume214, Springer.

[23] **Akkurt, S. and Sahin, M.** (2021). An efficient edge based data structure for the compressible Reynolds-averaged Navier–Stokes equations on hybrid unstructured meshes, *International Journal for Numerical Methods in Fluids*.

[24] **Spalart, P. and Allmaras, S.** (1992). A one-equation turbulence model for aerodynamic flows, *30th aerospace sciences meeting and exhibit*, p.439.

[25] **Balay, S.**, **Abhyankar, S.**, **Adams, M.**, **Brown, J.**, **Brune, P.**, **Buschelman, K.**, **Dalcin, L.**, **Dener, A.**, **Eijkhout, V.**, **Gropp, W.** *et al.* (2019). PETSc users manual.

[26] **Jameson, A. and Shankaran, S.**, (2009). An assessment of dual-time stepping, time spectral and artificial compressibility based numerical algorithms for unsteady flow with applications to flapping wings, 19th AIAA Computational Fluid Dynamics, p.4273.

[27] **Roe, P.L.** (1981). Approximate Riemann solvers, parameter vectors, and difference schemes, *Journal of computational physics*, *43*(2), 357–372.

[28] **Toro, E.F.**, **Spruce, M. and Speares, W.** (1994). Restoration of the contact surface in the HLL-Riemann solver, *Shock waves*, *4*, 25–34.

[29] **Liou, M.S.** (2006). A sequel to AUSM, Part II: AUSM+-up for all speeds, *Journal of computational physics*, *214*(1), 137–170.

[30] **Hascoët, L. and Pascual, V.** (2004). Tapenade 2.1 user's guide, *Ph.D. Dissertation*, INRIA.

[31] **Hascoet, L. and Pascual, V.** (2013). The Tapenade automatic differentiation tool: principles, model, and specification, *ACM Transactions on Mathematical Software (TOMS)*, *39*(3), 1–43.

[32] **Anderson, W.K. and Bonhaus, D.L.** (1994). An implicit upwind algorithm for computing turbulent flows on unstructured grids, *Computers & Fluids*, *23*(1), 1–21.

[33] **Blazek, J.** (2001). *Computational fluid dynamics: principles and applications*, Elsevier.

[34] **Batten, P.**, **Clarke, N.**, **Lambert, C. and Causon, D.M.** (1997). On the choice of wavespeeds for the HLLC Riemann solver, *SIAM Journal on Scientific Computing*, *18*(6), 1553–1570.

[35] **Thomas, P. and Lombard, C.** (1979). Geometric conservation law and its application to flow computations on moving grids, *AIAA journal*, *17*(10), 1030–1037.

[36] **Farhat, C.**, **Geuzaine, P. and Grandmont, C.** (2001). The discrete geometric conservation law and its effects on nonlinear stability and accuracy, *15th AIAA Computational Fluid Dynamics Conference*, p.2607.

[37] **Mavriplis, D.** (2003). Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, *16th AIAA computational fluid dynamics conference*, p.3986.

[38] **Ollivier-Gooch, C.**, **Nejat, A. and Michalak, K.** (2009). Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations, *AIAA journal*, *47*(9), 2105–2120.

[39] **Jalali, A. and Ollivier Gooch, C.F.** (2013). Higher-order finite volume solution reconstruction on highly anisotropic meshes, *21st AIAA Computational Fluid Dynamics Conference*, p.2565.

[40] **Carlson, J.R.** (2011). Inflow/outflow boundary conditions with application to FUN3D, **Technical Report**.

[41] **Mavriplis, D.J.** (2021). A residual smoothing strategy for accelerating Newton method continuation, *Computers & Fluids*, *220*, 104859.

[42] **Zandsalimy, M. and Ollivier-Gooch, C.** (2022). A novel approach to mesh optimization to stabilize unstructured finite volume simulations, *Journal of Computational Physics*, *453*, 110959.

[43] **Zangeneh, R. and Ollivier-Gooch, C.F.** (2017). Mesh optimization to improve the stability of finite-volume methods on unstructured meshes, *Computers & Fluids*, *156*, 590–601.

[44] **Habashi, W.G.**, **Dompierre, J.**, **Bourgault, Y.**, **Ait-Ali-Yahia, D.**, **Fortin, M. and Vallet, M.G.** (2000). Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles, *International Journal for Numerical Methods in Fluids*, *32*(6), 725–744.

[45] **Loseille, A.**, (2017). Unstructured mesh generation and adaptation, Handbook of Numerical Analysis, volume 18, Elsevier, pp.263–302.

[46] **Fidkowski, K.J. and Roe, P.L.** (2010). An entropy adjoint approach to mesh refinement, *SIAM Journal on Scientific Computing*, *32*(3), 1261–1287.

[47] **Wukie, N.A.** (2019). A no-slip, moving-wall boundary condition for the Navier-Stokes equations, *AIAA Aviation 2019 Forum*, p.3318.

[48] **Einfeldt, B.**, **Munz, C.D.**, **Roe, P.L. and Sjögreen, B.** (1991). On Godunov-type methods near low densities, *Journal of computational physics*, *92*(2), 273–295.

CURRICULUM VITAE

Name Surname : Erol AKSOY

EDUCATION :

- **B.Sc.** : 2021, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Department of Aeronautical Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2022-Current date Research Assistant at Istanbul Technical University
- 2021-2023 TUBITAK 2210 Domestic Graduate Scholarship

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Aksoy, E.,** & Sahin, M. (2023).*HEMLAB ALE: A New ALE Algorithm for Compressible Flow Problems*, II ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems, July 19-21, 2023, Lisbon, Portugal.

- **Aksoy, E.,** Sahin, M.(2023) . *HEMLAB ALE Algorithm Applied to Solve Flow Around S-76 Main Rotor*, XII. Ankara International Aerospace Conference, Sep 2023, Ankara/Turkey.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Aksoy, E.,** & Sahin, M. (2022).*An Application of Anisotropic Mesh Refinement to Solve Flow Around the S-76 Main Rotor with Swept-Tapered Tip*, AIAA SciTech 2022 Forum (p. 1319).

- **Aksoy, E.,** Sahin, M. (2021). *The Numerical Solution of Flow Field Around S76 Rotor*, XI. Ankara International Aerospace Conference, Sep 2021, Ankara.

- **Aksoy, E.,** Karali, H. & Yükselen, M. A. (2020). *Kanat Profilleri için Doğrusal Olmayan Aerodinamik Performans Görselleştirmesi ve İHA Analiz Programı ile Entegrasyonu*, VIII. Ulusal Havacılık ve Uzay Konferansı, Eylül 2020, Ankara.