ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

METRIC BASED MESH ADAPTATION METHODOLOGY APPLIED TO HEMLAB ALGORITHM

Ph.D. THESIS

Hülya SUKAS

Department of Aeronautical and Astronautical Engineering

Aeronautical and Astronautical Engineering Programme

JUNE 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

METRIC BASED MESH ADAPTATION METHODOLOGY APPLIED TO HEMLAB ALGORITHM

Ph.D. THESIS

Hülya SUKAS
(511192110)

Department of Aeronautical and Astronautical Engineering

Aeronautical and Astronautical Engineering Programme

Thesis Advisor: Prof. Dr. Mehmet ŞAHİN

JUNE 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

METRİK TABANLI AĞ UYARLAMA YÖNTEMİNİN HEMLAB ÇÖZÜCÜSÜNE UYGULANMASI

DOKTORA TEZİ

Hülya SUKAS
(511192110)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mehmet ŞAHİN

HAZİRAN 2025

Hülya SUKAS, a Ph.D. student of ITU Graduate School 511192110 successfully defended the thesis entitled "METRIC BASED MESH ADAPTATION METHODOLOGY APPLIED TO HEMLAB ALGORITHM", which he 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. Bayram ÇELİK** Istanbul Technical University	.....
	Prof. Dr. Metin MURADOĞLU Koç University	.....
	Prof. Dr. Aydın MISIRLIOĞLU Istanbul Technical University	.....
	Asst. Prof. Dr. Altuğ Melik BAŞOL Özyeğin University	.....

Date of Submission : **28 April 2025**
Date of Defense : **11 June 2025**

To my husband,
Ömer Faruk

FOREWORD

Completing this PhD thesis has been a journey filled with challenges, learning, and support from many people. This work is the result of years of effort, and I could not have done it alone.

First, I would like to sincerely thank my thesis advisor, Prof. Dr. Mehmet ŞAHİN. His guidance, knowledge, and encouragement have been invaluable throughout my research. His support has played a significant role in my academic journey, and I am truly grateful for his patience and advice. I have learned so much from him, and I could not have completed this work without his guidance. It has been an honor to be his student.

I am also deeply thankful to my husband, Ömer Faruk, for always believing in me. His support and encouragement have given me strength, especially in difficult times. I want to thank my family as well, whose love and support have helped me every step of the way. Their belief in me has meant everything.

A very special thanks to my daughter, Sare Doğa, who has brought endless joy and love into my life. Her laughter and energy remind me to cherish every moment, and she has been my greatest inspiration.

I also appreciate the computing resources provided by the National Center for High Performance Computing of Turkey (UYBHM) under grant number 10752009, as well as the computing facilities at TÜBİTAK ULAKBİM, High Performance and Grid Computing Center.

June 2025

Hülya SUKAS
(Aerospace and Aeronautical Engineer, M.Sc.)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ADT	**:** Alternating Digital Tree
AIAA	**:** American Institute of Aeronautics and Astronautics
ANSYS	**:** Analysis System
AoA	**:** Angle of Attafck
ARA	**:** Aircraft Research Association
ATS	**:** Adaptive Time Stepping
AUSM	**:** Advection Upstream Splitting Method
AV	**:** Air Vehicles
BSC	**:** Barcelona Supercomputing Center
BSL	**:** Baseline
CARDC	**:** China Aerodynamics Research and Development Center
CD	**:** Drag Coefficient
CFD	**:** Computational Fluid Dynamics
CFL	**:** Courant-Friedrichs-Lewy
CFMS	**:** The Centre for Modelling and Simulation
CFX	**:** Computational Fluid Dynamics Software
CL	**:** Lift Coefficient
CM	**:** Moment Coefficient
CREATE	**:** Computational Research and Engineering Acquisition Tools and Environments
CRM	**:** Common Research Model
CRMHL	**:** Common Research Model High Lift
CRM-HL-WB	**:** Common Research Model High Lift Wing Body
CSC	**:** Center for Computational Science
DES	**:** Detached Eddy Simulation
DGM	**:** Discontinuous Galerkin Method
DIST	**:** Distance
DLR	**:** German Aerospace Center
DOF	**:** Degrees of Freedom
DOD	**:** Department of Defense
EARSM	**:** Explicit Algebraic Reynolds Stress Model
EB	**:** Elliptic Blending
EMBRAER	**:** Empresa Brasileira de Aeronáutica S.A
EPIC	**:** Edge Primitive Insertion and Collapse
FDM	**:** Finite Difference Method
FEA	**:** Finite Element Analysis
FEFLO	**:** Finite Element Flow Solver
FEM	**:** Finite Element Method
FGMRES	**:** Flexible Generalized Minimal Residual Method
FOI	**:** Swedish Defence Research Agency

FTLE : Finite-Time Lyapunov Exponent
FVHFC : Finite Volume High-Fidelity Computation
FVM : Finite Volume Method
GGNS : Generalized Grid Navier-Stokes Solver
GMGW : Geometry and Mesh Generation Workshop
GMRES : Generalized Minimum Residual
GPU : Graphics Processing Unit
HL : High-Lift
GRET : General Research in Experimental Turbulence
HAD : Hesaplamalı Akışkanlar Dinamiği
HANIM : Hierarchical Adaptive Nonlinear Iteration Method
HLLC : Harten-Lax-van Leer-Contact
HLPW : High-Lift Prediction Workshop
HPCMP : High Performance Computing Modernization Program
HV : Horizontal and Vertical Stabilizers
HYPRE : High Performance Preconditioners
IAE : Institute of Aerodynamics and Engineering
IC : Insertion and Collapse
ICS : Insertion Collapse and element-face connectivity Swapping
ICSM : Insertion Collapse Swapping, and node Movement
ICUBE : Engineering Science, Computer Science, and Imaging Laboratory
ILU : Incomplete Lower-Upper factorization
INRIA : French National Institute for Research in Computer Science and Automation
ISCFDC : International Symposium on Computational Fluid Dynamics
JAXA : Japan Aerospace Exploration Agency
JFNK : Jacobian-Free Newton-Krylov
JSM : JAXA Standard Model
JST : Jameson-Schmidt-Turkel
KSP : Krylov Subspace
KSPBCGS : Krylov Subspace Bi-Conjugate Gradient Squared method
KSPGCR : Krylov Subspace Generalized Conjugate Residual method
KTH : Royal Institute of Technology
LAVA : Large-scale Atomic/Molecular Massively Parallel Simulator
LBM : Lattice Boltzmann Method
LDG : Local Discontinuous Galerkin
LE : Leading Edge
LEA : Linear Eddy-viscosity Approach
LES : Large Eddy Simulation
LU : Lower-Upper
LWST : Low-Wind-Speed Turbine
MAI : Moscow Aviation Institute
MATBAIJ : Matrix Block AIJ
MIT : Massachusetts Institute of Technology
NASA : National Aeronautics and Space Administration
NSMB : Navier-Stokes Multi-Block
ONERA : Office National d'Études et de Recherches Aérospatiales

PTC : Pseudo Transient Continuation
QCR : Quadratic Constitutive Relation
RANS : Reynolds-Averaged Navier-Stokes
RMS : Root Mean Square
RSM : Reynolds Stress Model
SA : Spalart-Allmaras
SALSA : System for Algebraic Library for Static Analysis
SARC : Spalart-Allmaras with Rotation and Curvature correction
SEM : Spectral Element Method
SGS : Sub-Grid Scale
SM : Standard Model
SNES : Scalable Nonlinear Equations Solver
SSG : Speziale-Sarkar-Gatski
SST : Shear Stress Transport
SUPG : Streamline Upwind/Petrov-Galerkin
TAU : German Aerospace Center's CFD solver
TE : Trailing Edge
TFG : Technology Focus Group
TMR : Turbulence Model Resource
TNT : Turbulent Non-Turbulent
TU : Technical University
URANS : Unsteady Reynolds-Averaged Navier-Stokes
USA : United States of America
UT : University of Texas
VLES : Very Large Eddy Simulation
XLES : Extra-Large Eddy Simulation
WALE : Wall-Adapting Local Eddy-viscosity
WB : Wing Body
WT : Wind Tunnel

SYMBOLS

A_i : Coefficient matrix
CFL : CFL number
C_D : Drag coefficient
C_F : Skin friction coefficient
$C_{f,x}$: Skin friction coefficient in x direction
C_L : Lift coefficient
C_M : Moment coefficient
C_P : Specific heat at constant pressure
C_p : Pressure coefficient
C_{rot} : Rotational coefficient
D : Drag force
E : Total energy
F_i : Inviscid fluxes
F_v : Viscous fluxes
H : Enthalpy
$\hat{H}$: Roe averaged enthalpy
J : Jacobian matrix
K_v : Limiter constant
L : Lift force
l : Characteristic length
L_e : Edge length
L_p : Norm
M : Mach number
M_∞ : Free stream Mach number
$M(x)$: Metric tensor
Pr_∞ : Free stream Prandtl number
Q : State vector
P : Eigenvector matrix
R : Residual vector
Re : Reynolds number
Re_∞ : Free stream Reynolds number
S : Domain surface
S : Sutherland constant
T : Temperature
U : Total velocity
U_∞ : Free stream velocity
$\hat{\mathbf{U}}$: Roe averaged velocity
V : Domain volume
W : Weight coefficient matrix
$\mathbf{W}$: Primitive variables vector
a : Speed of sound
$\hat{a}$: Roe averaged speed of sound

$\mathbf{b}$: Result vector
h : Mesh spacing
k : Thermal conductivity coefficient
$l_m(\vec{ab})$: Approximate length of vector
$\mathbf{n}$: Normal vector
p : Pressure
q : Heat flux
q_x : Heat flux in x direction
q_y : Heat flux in y direction
q_z : Heat flux in z direction
t : Time
u : Velocity component in x direction
v : Velocity component in y direction
w : Velocity component in z direction
x : Cartesian coordinate x axis
$\mathbf{x}$: Unknown vector
y : Cartesian coordinate y axis
y^+ : Non-dimensional wall distance
z : Cartesian coordinate z axis
χ : Non dimensional viscosity
Δ : Gradient operator
Λ : Eigenvalue matrix
Ω : Vorticity
α : Angle of attack
β : Upwinding parameter
γ : Ratio of specific heat
ε : Harten coefficient
λ^* : Non dimensional thermal conductivity
μ : Kinematic viscosity
μ^* : Non dimensional kinematic viscosity
μ_t^* : Non dimensional turbulent viscosity
ν : Dynamic viscosity
ρ : Density
$\hat{\rho}$: Roe averaged denistiy
τ : Stress tensor
ϕ : Limiter function

LIST OF TABLES

Page

LIST OF FIGURES

Page

METRIC BASED MESH ADAPTATION METHODOLOGY APPLIED TO HEMLAB ALGORITHM

SUMMARY

This dissertation presents a detailed study on metric-based mesh adaptation methodologies applied to high-lift aerodynamic simulations within the HEMLAB framework. The research integrates advanced computational techniques to enhance the accuracy and efficiency of computational fluid dynamics (CFD) solvers, particularly for high-lift configurations analyzed in the AIAA High Lift Prediction Workshop Series (HLPW). The primary focus is on improving numerical predictions by refining computational meshes dynamically in response to flow characteristics.

A key contribution of this research is the development and integration of metric-based anisotropic mesh adaptation strategies. These techniques refine computational grids in regions of high flow gradients, such as boundary layers and wake structures, ensuring improved resolution without excessive computational cost. The numerical studies conducted on HLPW-3, HLPW-4, and HLPW-5 test cases illustrate the impact of these adaptation methods on aerodynamic loads, particularly in improving lift and pressure coefficient distributions.

In addition to mesh adaptation, this study incorporates solution methodologies through the integration of the Scalable Nonlinear Equations Solvers (SNES) framework. By utilizing nonlinear Newton-Krylov methods with efficient preconditioners the computational cost of solving nonlinear equations is significantly reduced. The accurate Jacobian evaluation as a preconditiner further enchance solver efficiency.

Turbulence modeling is another aspect of this research. The initial version of HEMLAB was limited to the standard Spalart-Allmaras (SA) model. This study extends its turbulence modeling capabilities by incorporating SA-neg, SA-neg-ft2, and SA-QCR-2000 models. These modifications improve the solver's accuracy and convergence in predicting separated flows, transitional effects, and streamline curvature influences. A second-order discretization scheme is also introduced to the SA model, enhancing numerical precision and reducing diffusion errors.

Furthermore, this thesis highlights the significance of machine precision in high-fidelity simulations. The methodologies developed in this work contribute to the broader field of CFD by providing improved numerical accuracy, efficiency, and adaptability. Future research directions include refining transition modeling techniques, extending adaptive meshing strategies such as goal based adaptation strategies and improving sensor function and further investigation of cross diffusion effects on highly anisotropic meshes for the aircraft configurations.

METRİK TABANLI AĞ UYARLAMA YÖNTEMİNİN HEMLAB ÇÖZÜCÜSÜNE UYGULANMASI

ÖZET

Bu doktora tezi, kenar tabanlı sonlu hacimler yöntemini kullanan hesaplamalı akışkanlar dinamiği (HAD) çözücüsü HEMLAB algoritmasının yüksek taşıma konfigürasyonlarında kullanılmak üzere geliştirilmesine yönelik kapsamlı bir araştırmayı sunmaktadır. Bu bağlamda, mevcut çözücünün aerodinamik simülasyonlardaki doğruluk ve verimliliğini artırmak amacıyla çeşitli yenilikçi yaklaşımlar geliştirilmiştir. Özellikle AIAA High Lift Prediction Workshop (HLPW) serilerinde ele alınan karmaşık geometriler üzerinde odaklanılarak, metrik tabanlı ağ uyarlama teknikleri, türbülans modeli iyileştirmeleri ve ileri sayısal çözüm yöntemleri detaylı bir şekilde incelenmiştir. Çalışmada geliştirilen yöntemlerin özgün katkıları üç ana başlıkta toplanabilir: (i) Metrik tabanlı ve hibrid sensör destekli adaptif ağ stratejileri, (ii) Gelişmiş sayısal çözüm yöntemleri ile hesaplama verimliliğinde artış, ve (iii) Türbülans modellemesinde sağlanan doğruluk iyileştirmeleri.

Günümüz havacılık endüstrisinde operasyonel maliyetlerin azaltılması, yakıt verimliliğinin artırılması ve aerodinamik performansın iyileştirilmesi gibi hedefler, yüksek taşıma sistemlerinin tasarımını kritik hâle getirmiştir. Bu sistemler, temel kanat yapısına ek olarak slat, flap, fairing gibi bağlantı elemanlarından oluşan oldukça karmaşık geometrilere sahiptir. Bu tasarımsal karmaşıklık, akış alanının doğru şekilde modellenmesini zorlaştırmakta ve geleneksel HAD yöntemlerinin sınırlarını zorlamaktadır. Özellikle sınır tabakası ayrılmaları ve kanat ucu girdaplarının bir arada bulunduğu bu sistemlerde, hesaplamalı ağların optimal dağılımı büyük önem taşımaktadır. Bu tez çalışması, söz konusu zorluklara yönelik HEMLAB algoritmasında gerçekleştirilen iyileştirmeleri HLPW'de sunulan geometriler üzerinden kapsamlı biçimde doğrulamayı amaçlamaktadır. Temel motivasyon, endüstriyel uygulamalarda karşılaşılan yüksek hesaplama maliyetlerini düşürürken, aynı zamanda sonuç doğruluğunu artırmaktır.

Tezin en önemli katkılarından biri, hesaplamalı ağların akış özelliklerine göre dinamik olarak yeniden düzenlenmesini sağlayan metrik tabanlı anizotropik ağ uyarlama stratejilerinin geliştirilmesi ve HEMLAB çözücüsüne entegre edilmesidir. Bu kapsamda INRIA tarafından geliştirilen pyAMG ve NASA'nın REFINE algoritmaları başarıyla entegre edilmiştir. Bu entegrasyon sayesinde çözücünün adaptif ağ yeteneklerinde önemli gelişmeler sağlanmıştır. Özellikle sınır tabakaları, şok dalgaları, ayrılma bölgeleri ve kanat ucu girdapları gibi kritik akış yapılarının bulunduğu bölgelerde ağ otomatik olarak iyileştirilirken, diğer bölgelerde daha az ağ elemanı kullanılarak verimli bir dağılım elde edilmiştir.

Ağ adaptasyon sürecinde, geometrik karmaşıklığın yüksek olduğu bölgelerde hücre boyutlarının otomatik olarak küçültülmesi ve akışın fiziksel özelliklerine göre ağın yeniden düzenlenmesi gibi işlevler kazandırılmıştır. Mach, entropi, basınç, mesafe

gibi çeşitli akış özelliklerine dayalı fonksiyonlarla ağ uyarlaması yapılabilmektedir. Bu süreçte klasik Mach sayısı sensörüne ek olarak geliştirilen hibrid sensör yaklaşımı uygulanmıştır. Bu yöntem, özellikle geometriye yakın bölgelerde daha yoğun bir ağ dağılımı sağlamış ve sınır tabakası çözünürlüğünde kayda değer iyileşmeler elde edilmesini mümkün kılmıştır. Geleneksel yöntemlerde yüksek hız gradyanlarının olduğu bölgelerde ağ yoğunluğu artırılırken, geometriye yakın bölgelerde yeterli çözünürlük sağlanamaması önemli bir sorundu. Hibrid sensör yaklaşımı bu sorunu çözerek hem sınır tabakası içinde hem de serbest akış bölgesinde optimal ağ dağılımı sağlamıştır. Bu bağlamda, geometrik yakınlık ile akış özelliklerini birleştiren yeni bir ağ adaptasyon metriği geliştirilmiştir. Bu metrik, yüzey eğrilikleriyle birlikte akış değişkenlerindeki gradyanları dikkate alarak çok yönlü bir optimizasyon sunmaktadır.

Ayrıca ağ kalitesinin adaptasyon sonrası korunması amacıyla hücre şekil bozulmaları minimize edilmiş ve yüzey geometrisine bağlı ağ bozulmalarını engelleyen özel projeksiyon algoritmaları kullanılmıştır. Bu algoritmalar, adaptif ağların fiziksel doğruluğunu korurken, sayısal çözümün kararlılığını da desteklemiştir. Tüm bu geliştirmeler, hem ağ oluşturma süresini azaltmış hem de manuel müdahale ihtiyacını minimuma indirerek kullanıcı bağımsızlığını artırmıştır. Bu sayede HEMLAB, tamamen otomatik çalışan, uçtan uca çözüm sağlayan bir HAD aracı hâline gelmiştir. Bu otomasyon, özellikle endüstriyel projelerde farklı konfigürasyonlara hızlıca adapte olunmasını kolaylaştırmıştır.

HEMLAB algoritmasının çözüm verimliliğini artırmak amacıyla PETSc kütüphanesinin Scalable Nonlinear Equations Solvers (SNES) çerçevesi entegre edilmiştir. Bu entegrasyon, doğrusal olmayan denklem sistemlerinin çözümünde önemli avantajlar sağlamıştır. GMRES (Generalized Minimal Residual) ve FGMRES (Flexible GMRES) gibi ileri iteratif çözücülerin kullanımı, yüksek Reynolds sayılı akışlarda kararlı ve hızlı yakınsama sağlamıştır. Bu çözücüler, karmaşık akış yapılarında geleneksel yöntemlere göre daha iyi performans göstermiştir. Çözücü parametrelerinin dinamik olarak ayarlanabilmesi sayesinde farklı akış rejimlerinde otomatik optimizasyon sağlanmıştır. ILU(k), blok Jacobi ve eklemeli Schwarz gibi çeşitli ön koşullandırıcılar karşılaştırılarak problem tipine göre en uygun yöntemler belirlenmiştir. Bu karşılaştırmalar sonucunda yakınsama hızında önemli iyileşmeler ve büyük ölçekli 3B simülasyonlarda ciddi hesaplama süresi kazanımları elde edilmiştir. Ayrıca, Jacobian matrisinin açıkça oluşturulmasından kaynaklanan hesaplama yükünü azaltmak için Jacobian-Free Newton-Krylov (JFNK) yöntemi uygulanmıştır. Bu yöntemde sonlu fark yaklaşımlarıyla bellek tüketimi optimize edilmiştir. Yüksek doğruluk gerektiren durumlar için ise el ile hesaplanan analitik Jacobian matrisleri geliştirilmiş ve bu matrisler sonlu fark yaklaşımı ile karşılaştırılarak doğrulukları test edilmiştir. JFNK yönteminde sonlu fark adım boyutunun dinamik optimizasyonu, yakınsama karakteristiğini önemli ölçüde iyileştirmiştir.

HEMLAB'ın ilk versiyonu yalnızca standart Spalart-Allmaras (SA) türbülans modeliyle sınırlıyken, bu tezde önemli modelleme iyileştirmeleri gerçekleştirilmiştir. Ayrılmış akış bölgelerinde negatif türbülans viskozitesini işleyebilen SA-neg modeli, Newton metodunun quadratic yakınsama özelliğini yakalamamıza olanak verdiği için stall bölgesindeki akışlar daha doğru tahmin edilebilmiştir. Geçiş akışlarının modellenmesinde kullanılan f_{t2} fonksiyonu üzerinde yapılan iyileştirmeler, özellikle slat ve flap gibi elemanlar üzerindeki laminer-türbülans geçiş bölgelerindeki çözüm doğruluğunu artırmıştır. Ayrıca, dönel akışlar ve akış çizgisi eğriliklerini dikkate

alan SA-QCR-2000 modeli, kanat ucu girdapları gibi karmaşık akış yapılarının simülasyonunda başarılı sonuçlar vermiştir. SA modeline uygulanan ikinci mertebeden ayrıklaştırma şeması, sayısal difüzyonu azaltarak ince sınır tabakalarının çözüm kalitesini artırmıştır.

HLPW-3, HLPW-4 ve HLPW-5 test vakaları üzerinde yapılan kapsamlı simülasyonlar, geliştirilen adaptif yöntemlerin aerodinamik yükler üzerindeki olumlu etkisini açıkça ortaya koymuştur. Taşıma katsayısı (C_L), sürükleme katsayısı (C_D) ve basınç dağılımlarında önemli iyileştirmeler elde edilmiştir. CRM-HL geometrisi için yapılan simülasyonlarda, adaptif ağlar sayesinde geleneksel sabit ağlara kıyasla daha az ağ noktasıyla daha doğru sonuçlara ulaşılmış ve hesaplama maliyetlerinde önemli tasarruf sağlanmıştır. Farklı hücum açılarıyla yapılan testler, özellikle stall bölgelerinde geliştirilen yöntemlerin etkinliğini ortaya koymuştur. Rüzgar tüneli verileriyle yapılan karşılaştırmalar, elde edilen sonuçların deneysel verilerle uyum içinde olduğunu göstermiştir. Ayrıca, JAXA Standard Model (JSM) ve NASA Common Research Model (CRM) geometrileri üzerinde yapılan 2B ve 3B simülasyonlar, adaptif ağ yöntemlerinin farklı konfigürasyonlarda sağladığı başarıyı kanıtlamıştır. Bu testler sayesinde ağ adaptasyonunun ağ eleman sayısından bağımsız çözüme katkısı da gösterilmiş; farklı başlangıç ağlarıyla yapılan analizler sonrasında sonuçların ağdan bağımsız hâle geldiği ortaya konmuştur. CRM-HL Wing-Body konfigürasyonunda farklı türbülans modellerinin karşılaştırılması, önerilen metodolojinin çeşitli geometrilerdeki başarısını desteklemiştir.

Bu tez kapsamında geliştirilen metrik tabanlı ağ uyarlama metodolojilerinin HEMLAB çözücüsüne entegre edilmesi, sayısal yöntemlerin iyileştirilmesi ve türbülans modeli seçeneklerinin artırılması sayesinde yüksek taşıma aerodinamiği simülasyonlarında önemli gelişmeler elde edilmiştir. Elde edilen bulgular, hem akademik araştırmalar hem de endüstriyel uygulamalar açısından değerli katkılar sunmaktadır. Çalışmanın sonuçları, yüksek taşıma sistemlerinin tasarımı ve optimizasyonu süreçlerinde önemli zaman ve maliyet tasarrufu sağlama potansiyeline sahiptir. Gelecek çalışmalar kapsamında, geçiş modelleme tekniklerinin daha da geliştirilmesi, laminer-türbülans geçişinin daha doğru tahmin edilmesini sağlayabilir. Ayrıca paralel hesaplama ve GPU tabanlı çözücü entegrasyonu, büyük ölçekli simülasyonların hızını artırabilir. Yüksek mertebeden ayrıklaştırma şemalarının uygulanması, sayısal difüzyonu azaltarak çözüm doğruluğunu iyileştirebilir. Makine öğrenmesi tabanlı yöntemlerin ağ adaptasyon süreçlerine entegrasyonu ise hesaplama verimliliğini daha da artırma potansiyeline sahiptir.

Sonuç olarak, bu çalışma yüksek taşıma sistemlerinin tasarım ve analiz süreçlerinde HAD yöntemlerinin güvenilirliğini önemli ölçüde artırmaktadır. Tezde geliştirilen metodolojiler, yalnızca havacılık değil; otomotiv, enerji ve rüzgar türbini tasarımı gibi alanlarda da uyarlanabilir niteliktedir.

1. INTRODUCTION

Many developments and research efforts are still ongoing to achieve efficient civil transportation systems in terms of operational costs, payloads, fuel consumption, weight, and aerodynamic noise emissions. Therefore, high-lift configurations are still being developed to meet these requirements. These configurations as shown in Figure 1.1 can provide a higher lift coefficient before an aircraft enters the stall region. However, the challenge lies in the fact that the flow physics are quite complex due to the complexity of the geometry. This complexity arises from additional equipment such as slat, slat bracket, flap, fairing, nacelle, leading edge strake, pylon as shown in Figure 1.2. These multi-element systems have multiple boundary layers, as indicated in Figure 1.3, and the interactions between these regions need to be studied in detail to achieve the most realistic results. The same figure also highlights that flow separations and the tip vortexes which are crucial and must be accurately simulated using an appropriate numerical model.

Figure 1.1 : Wing tip vortex occured on the High Lift Configuration [1]

Several studies on high-lift configurations have been conducted so far, as higher lift coefficients offer numerous benefits. For instance, findings in the literature show that a 1% increase in the maximum lift coefficient results in an additional payload capacity of

22 passengers or 4,400 lbs under landing conditions [16]. The same study also analyzed the results in terms of the lift-to-drag ratio (L/D). It was found that a 1% increase in L/D leads to an additional payload capacity of 14 passengers or 2,800 lbs under take-off conditions.

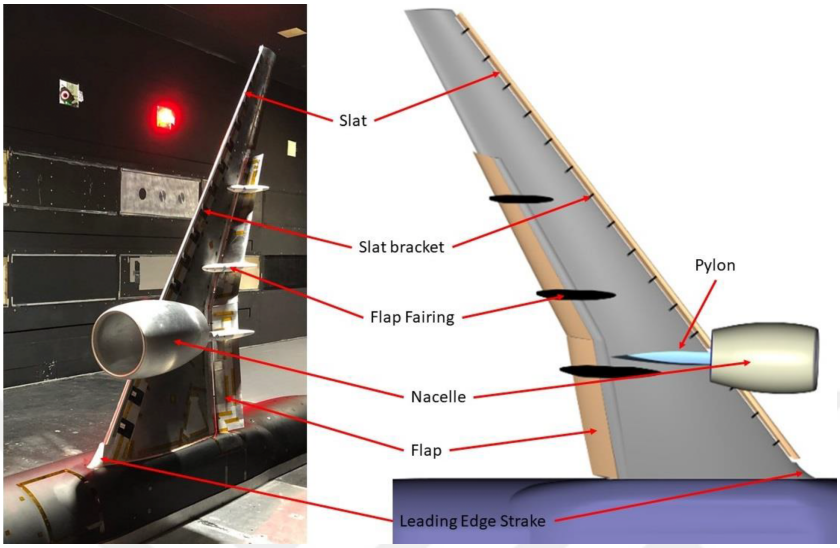


Figure 1.2 : Main parts for the high-lift configurations [2]

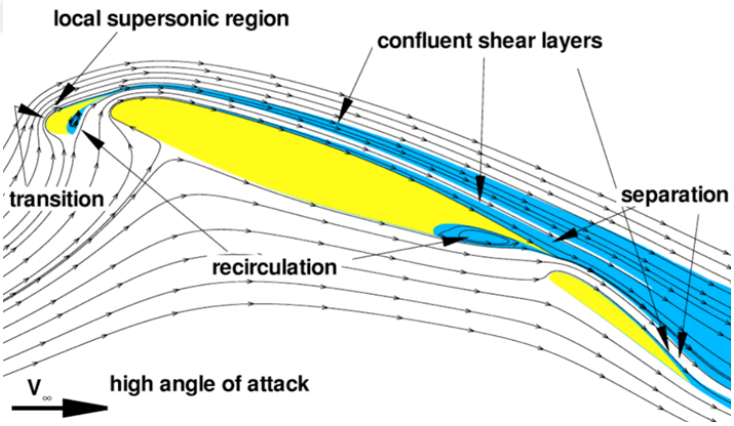


Figure 1.3 : Flow field around the wing section of a three element wing [3]

Apart from Figure 1.3, multi-element airfoils can have various configurations, such as only a slat, only a flap, or both. However, the observed behaviors will differ since their efficiencies are not the same. This phenomenon can be examined in Figure 1.4, which compares regular airfoils without any attachments, airfoils with only a slat, only a flap, and both slat and flap configurations in terms of lift coefficient for various angles of attack. It can be seen that the classical airfoil without any attachments has the minimum lift coefficient, while the maximum is achieved with the slat-flap combination.

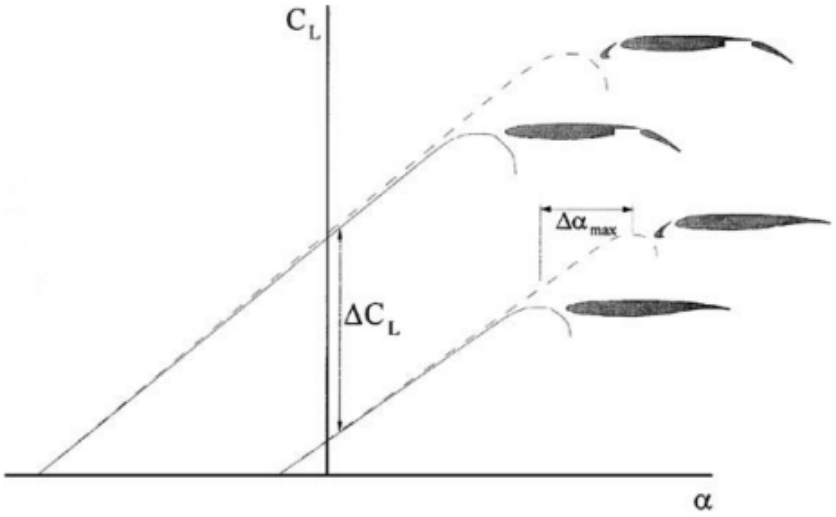


Figure 1.4 : Comparision of Multi Element Airfoils [4]

It should also be noted that the slat-only configuration increases the stall angle. The curve's trend moves horizontally with the addition of a slat, but it shifts vertically with the addition of a flap. As shown in Figure 1.3, there are gaps between the airfoil elements which can be increased or decreased depending on the flight condition. Therefore the lift coefficient can vary for each condition. The possible effects of these geometric variations are briefly explained by [17] in five major points:

• *Slat Effect*

Using a slat decreases the pressure peaks on the leading edge of the main body due to the circulation occurring behind the trailing edge of the slat.

• *Circulation Effect*

Each part of the airfoil creates a high-velocity region for the element located behind it. For instance, in the main body and flap area, a higher inclined velocity region forms at the trailing edge of the main body, strongly influencing the leading edge of the flap. Briefly, the trailing edge of the forward element induces larger circulation due to the inclined velocity affecting the leading edge of the downstream element.

• *Dumping Effect*

The high-speed flow exiting the trailing edge of an upstream element (such as the main airfoil) enters the gap between elements and effectively "dumps" into the region ahead of the downstream element (such as a flap). This process increases the pressure at the trailing edge of the upstream element and helps to decrease the adverse pressure

gradient, thereby delaying or preventing flow separation. By allowing for more efficient pressure recovery, the dumping effect enhances overall lift performance and contributes to the cooperative aerodynamic behavior between multiple elements.

• *Off-the-Surface Pressure Recovery*

The velocity generated by the forward elements reduces the backward velocity to near free-stream velocity levels. This occurs due to the absence of wall effects on the downstream element. The reduction in velocity creates a wake region, but the absence of the downstream element prevents a sudden drop in velocity. Thus, this is an efficient way to reduce the strength of wakes formed behind the elements.

• *Fresh Boundary Layer Effect*

Each airfoil element is independent, this is allowing the creation of new boundary layers for each element. The boundary layers of these elements are thinner compared to regular airfoils, it keeps them stable against to the adverse pressure gradient better than thick boundary layers. Consequently, the gap distances must be optimized to minimize flow separation.

Another critical point that requires detailed examination is the occurrence of the transition region. Accurate simulation of the boundary layer is essential to capture correct flow properties. Generally, Tollmien-Schlichting inflectional instability, laminar separation, and turbulence contamination effects are the primary causes of transition in 2D high-lift configurations. However, it has been emphasized that attachment-line instability must be considered for transition modeling in 3D high-lift configurations [18]. There is no single flow regime for attachment-line transition—it could be laminar, transitional, or turbulent. The flow field is influenced by the attachment-line transition.

The same study also identified factors contributing to attachment-line transition, including pressure distribution, leading-edge sweep angle, Reynolds number, surface roughness, and flow contamination. Some of these effects will be briefly discussed in the following sections.

1.1 Turbulence Effects on High-Lift Simulations

Among the various turbulence models, the literature suggests that the Spalart-Allmaras (SA) turbulence model provides more accurate results. Therefore, several modifications of the SA model have been investigated. Since these models are quite complex and non-linear, there are still aspects that need improvement. One significant finding in the study [19] shows that the SA and SST models show good agreement at low angles of attack (AoAs), but the SA model performs much better at higher AoAs .

For the 2D multi-element airfoil case, the effect of turbulence models on the results was also investigated. Three different turbulence models were selected (SARC, SST k-ω, and k-ε), and all of them predicted similar results before stall. However, the maximum lift coefficient was over-predicted by all of the models. A detailed examination of the results revealed that the SST k-ω turbulence model provided the most accurate predictions in terms of lift, drag, and moment coefficients, as well as the maximum lift coefficient and stall angles.

The SA model over-predicted the results in this 2D case, while the k-ε model estimated higher friction drag. Similar results were obtained for several 3D high-lift geometries in [20]. The SA and Menter's SST model were applied to an unstructured grid system, and similar outcomes to the previous 2D case were achieved for the 3D simulations as well. It was stated that the prediction of the maximum lift coefficient and stall angle is highly sensitive to the selected turbulence model. Additionally, it was found that the SST model provides more accurate results, especially at higher AoAs.

Another interesting outcome regarding turbulence model verification was obtained using several turbulence models in [21]. It was shown that one of the modified models (k-ω Wilcox) gave the most accurate results. Several turbulence model results were also compared for high-lift cases in another paper in [22]. SST, TNT, and linear and non-linear EARSM variations of the k-ω turbulence models were investigated. It was observed that there was no significant effect on the results in terms of using linear or non-linear EARSM models. However, lift was still over-predicted, especially at lower AoAs. These turbulence models were also utilized in [23], and the pressure

distributions were in good agreement with each other at higher AoAs. It was stated that the k-ω LEA model predicts the lift coefficient more accurately than the others.

To reduce total simulation time, some turbulence models were improved, and unsteady effects were investigated in detail using wall function approaches studied in [24]. The results obtained with this updated turbulence model showed that computational costs could be reduced by approximately 50%. However, the maximum lift was predicted to be higher than the experimental data, although there was good agreement in the linear range. Effects of the variation of k-ω turbulence models were also investigated by [25]. Edwards and Chandra's modification was used for take-off cases, and the standard Wilcox k-ω two-equation model was used for landing cases. The lift coefficient results were in good agreement with experiments, while there were some discrepancies for the drag coefficient. These results prove that the turbulence model selected has an obvious effect on aerodynamic coefficients, so this issue should be carefully considered in high-lift simulations. It would be beneficial to improve current turbulence models in terms of computational cost and accuracy, as they may fail to predict aerodynamic coefficients, stall angles, and pressure distributions accurately, especially for separated flows.

1.2 Transition Effects on High-Lift Simulations

Accurately capturing transition effects is particularly critical for high-lift configurations due to their strong influence on aerodynamic performance, including lift, drag, and separation characteristics. A boundary layer transition model is highly recommended for high-lift cases because the flow on slat and flap surfaces has a smaller Reynolds number and can be considered laminar in these regions [26]. Therefore, flow solvers that use fully turbulent conditions do not predict the transition effects as accurately.

There are also some additional implementations performed in [25] to predict the flow field in transition regions more accurately. The RANS-based laminar boundary layer model and e^N database method were integrated to predict the transition. This system also automatically predicts whether the flow regions are laminar, turbulent, or transitional. The results obtained with this model show excellent agreement with experimental data. However, in some cases, the model needs to be improved to capture

the transition point of laminar separations and wake regions. Also, all transition lengths were predicted in the correct order.

1.3 Reynolds Number Effects on High-Lift Simulations

The Reynolds number effect is one of the topics investigated by researchers for high-lift geometries. For instance, the results provided by [5] show that the Reynolds number has two important impacts on the linear and stall regions. The first is the increase in lift in the linear region as the free-stream velocity increases. The second impact is that stall behavior depends on the free-stream velocity. For example, stall occurs at 12° for a $60m/s$ free-stream velocity, but C_L does not drop in this region. It stays at the same level for a while and then begins to decrease. This was explained by the separation that dominates the stall. Additionally, there are two peaks in Figure 1.5 for $40m/s$ and $50m/s$ free-stream velocities [5]. It was shown that the higher peak occurs at a lower AoA, while the lower peak occurs at a higher AoA. The second-highest peak occurs where the initial velocity has the lowest value $(30m/s)$.

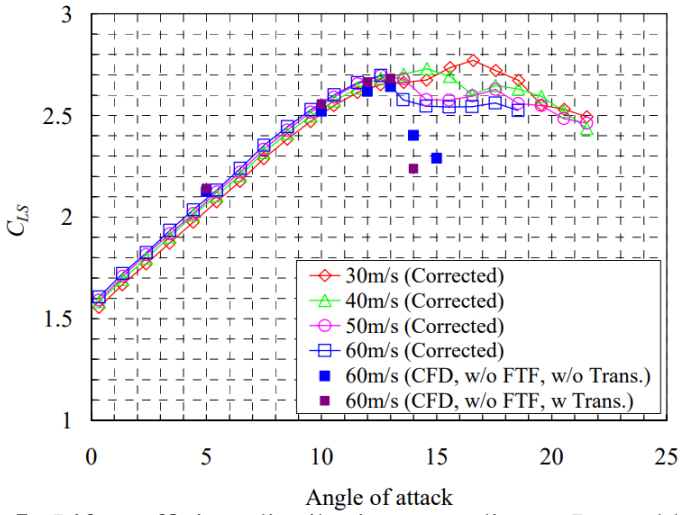


Figure 1.5 : Lift coefficient distribution according to Reynolds number [5]

1.4 Grid Effects on High-Lift Simulations

In terms of grid systems, numerical simulations are carried out with both structured and unstructured grid designs. For instance, high-lift geometries are investigated using multi-block structured and unstructured meshes in [27]. Multi-element airfoils are used to investigate grid dependency. To implement this, a mesh refinement approach is applied to the unstructured mesh cases. The results obtained in [27] prove that the

accuracy of drag prediction is strongly related to the mesh density away from the body, where large circulations and wake regions occur. C_L is less sensitive to grid density, so the results obtained are similar for both structured and unstructured grids, unlike C_D. The objective function is selected as entropy increment for the mesh refinement process. This is an effective method since it increases the grid elements where the number of points is crucial for accuracy.

A three-element trapezoidal wing attached to a fuselage with a full-span flap and part-span flaps was used for 3D simulations in [27]. These were performed with both structured and unstructured grid systems. The results were in good agreement even for the unstructured mesh. It is also noted that unstructured grid systems require more refinement steps to capture the wake near the trailing edge, wingtip vortices, and flow separations, compared to structured grid systems.

In [20], 3D high-lift models were analyzed with multi-block structured and unstructured grid systems using the SA and Menter's Shear Stress Transport models. Two geometries (three-element trapezoidal wing) were used, with full-span and part-span flaps. A comparison of structured and unstructured grid systems using the same turbulence model (SA) was provided. As an important outcome of this study, the importance of mesh density was emphasized, especially where flow starts to separate around the wing-fuselage junction. Another study on this topic was performed using hybrid unstructured and block-structured grids [21]. The results showed that the predicted lift coefficient increases as the grid is refined. It was also stated that the accuracy of results can be improved with the adaptive grid method when using an unstructured mesh design.

One other study emphasized the importance of the provided initial solution [28]. A different approach for simulation was proposed for the mesh adaptation process. In this approach, the solution starts with a certain angle of attack. Each flow solution uses the previous one's information through linear interpolation. It was recommended to use a lower angle for the initial solution since it helps prevent flow separations.

The AIAA Draft Prediction Workshop series also highlights the fact that the accuracy of numerical results is highly dependent on grid quality and resolution issues to obtain more accurate results for aerodynamic loads. Therefore, many mesh adaptation studies

have been performed for high-lift configurations to improve grid resolution in regions with high flow gradients.

Considering all the complex flow modeling of high-lift configurations explained so far, it is quite challenging to accurately mimic the real flow around the wing to obtain precise results. Hence, several high-lift workshops are organized to understand the reliability and capabilities of CFD solvers for high-lift cases.

1.5 AIAA High Lift Workshops

The AIAA High Lift Prediction Workshop (HLPW) series is a collection of organized events aimed at improving the understanding and accuracy of computational fluid dynamics (CFD) simulations for high-lift configurations. These workshops provide a structured platform for researchers and industry experts to evaluate and compare existing CFD models against high-quality experimental data, offering valuable insights into the strengths and limitations of different turbulence models, numerical techniques, and meshing strategies for high-lift prediction. Each workshop in the series focuses on specific test cases, ranging from two-dimensional high-lift airfoils to full-scale three-dimensional aircraft configurations, often incorporating realistic industrial geometries. Participants employ various computational approaches, including Reynolds Averaged Navier-Stokes (RANS), large-eddy simulation (LES) and hybrid RANS-LES methods, to predict aerodynamic performance and flow structures. The results are then systematically analyzed to identify key sources of discrepancies between CFD predictions and experimental measurements. Additionally, these workshops encourage discussions on best practices, mesh adaptation techniques, grid convergence studies, and the role of transition modeling in high-lift aerodynamics.

By encouraging collaboration between academia, industry, and government organizations, the HLPW series helps earning progress in computational methodologies, ultimately contributing to more reliable and efficient aircraft designs. The insights gained from these workshops have led to continuous improvements in CFD solvers, turbulence models, and meshing strategies applied for high-lift applications. In the following sections, each workshop in the series will be briefly explained, highlighting

the key test cases, methodologies, and findings that have shaped the field of high-lift aerodynamics.

1.5.1 High-lift prediction workshop -1

The first workshop was organized by AIAA in June 2010. The objective of the workshop was to assess the capability of CFD solvers for predicting the performance of swept, medium, and high-aspect-ratio wings in landing and takeoff configurations. The NASA Trapezoidal Wing that is shown in Figure 1.6 was selected as the benchmark geometry, and a total of 39 different results were submitted for comparison.

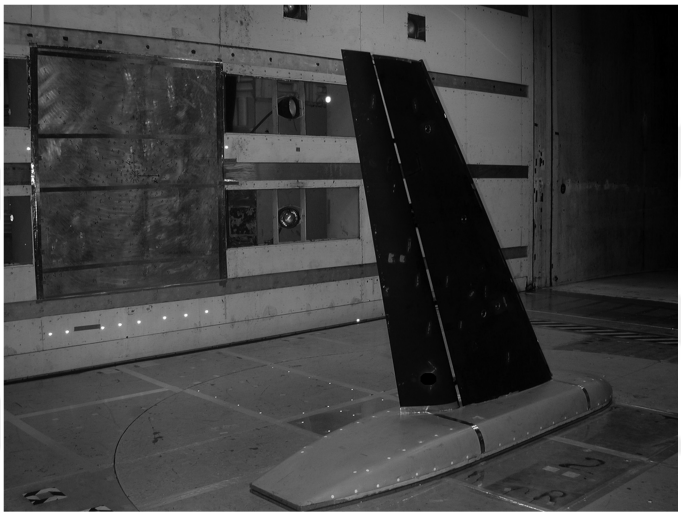


Figure 1.6 : NASA Langley Subsonic Wind Tunnel, AIAA 1st High Lift Prediction Workshop [6]

To understand the effects of grid design on the results, both structured and unstructured grid systems were used. Grid convergence studies were also performed for both grid systems with different turbulence models at an angle of attack (AoA) of 13°. The outcomes show that the structured grid results converge to lower values than the experimental data, providing more accurate results with the SA turbulence model. On the other hand, the unstructured grid results do not yield a clear outcome in terms of the turbulence model used. Both grid systems align more closely with the experimental data as the grid resolution increases. However, as shown in Figure 1.7, the unstructured grid results are more accurate than those of the structured grid at the coarse level. The grid convergence study was also conducted for an AoA of 28°. As illustrated in Figure 1.8, the results were similar to those at 13°. However, some participants predicted lower lift coefficients compared to others. It was observed that these participants did not

restart the converged solution obtained at lower angles of attack as the initial solution for higher angle-of-attack simulations. This highlights the sensitivity of CFD solvers to the initial solutions provided in simulations.

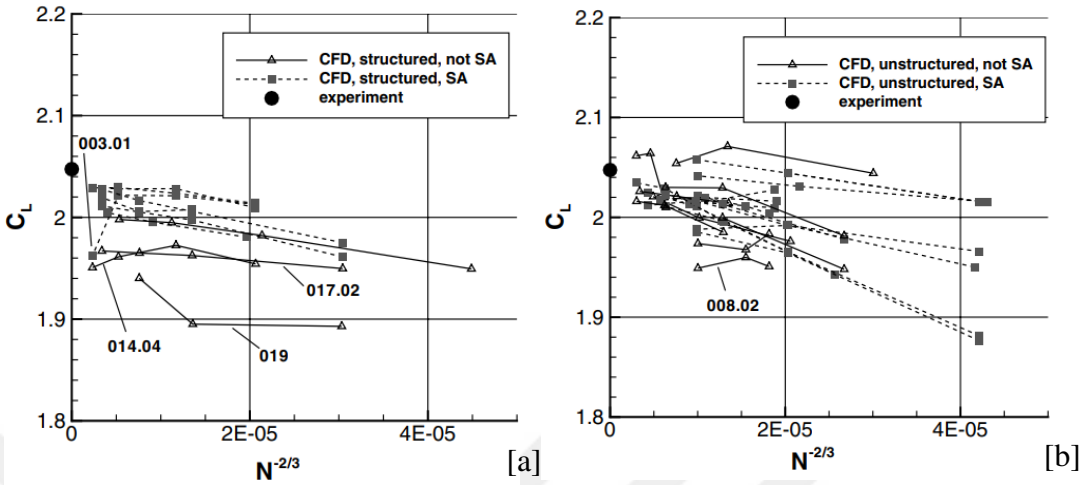


Figure 1.7 : Convergence of the grid in terms of C_L with structured [a] and unstructured [b] grid systems for α= 13°, AIAA 1st High Lift Prediction Workshop [6].

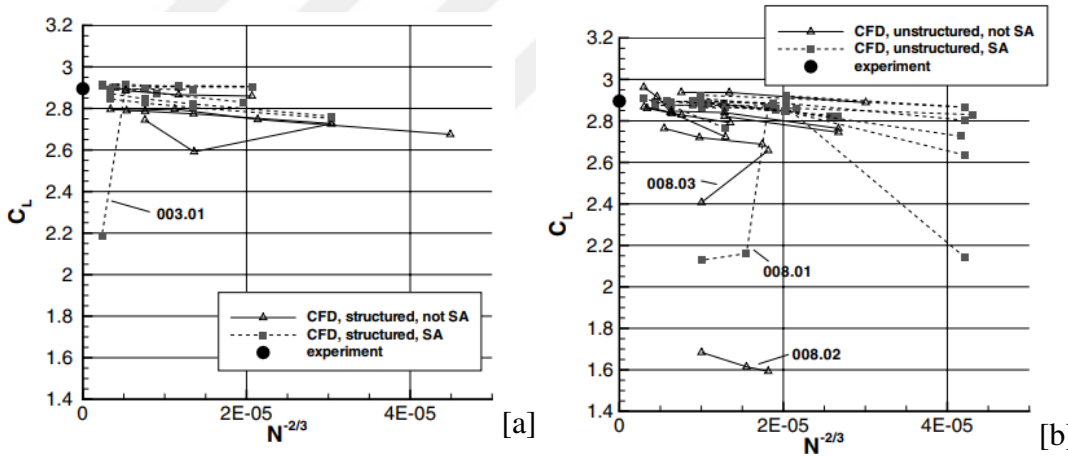
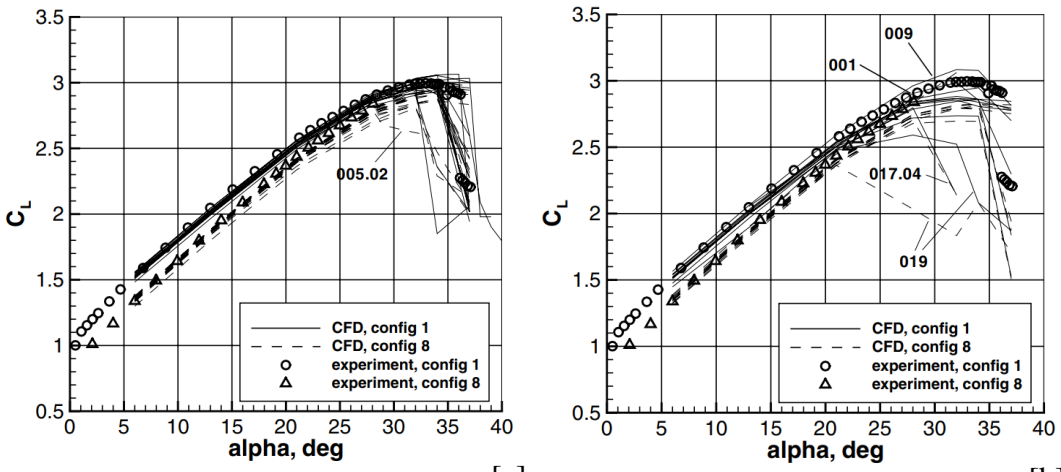


Figure 1.8 : Convergence of the grid in terms of C_L with structured and unstructured grid systems α= 28°, AIAA 1st High Lift Prediction Workshop [6]

To assess the prediction capabilities of CFD solvers for stall angles, results were presented at various AoAs using different turbulence models for two distinct wing configurations. “Configuration-1” and “Configuration-8” correspond to "slat 30°/ flap 25°" and "slat 30°/ flap 20°", respectively. The experimental results for Configuration-8 show lower lift compared to Configuration-1. Numerically predicted results were more accurate for Configuration-8 before the stall region. As shown in Figure 1.9, the SA turbulence model provides more accurate predictions around the stall region compared to the other turbulence models used.



[a] [b]

Figure 1.9 : Comparision of the results for several angle of attacks, using (a) SA and (b) Other turbulence models for two configurations, AIAA 1st High Lift Prediction Workshop [6]

Another important aerodynamic parameter that needs to be examined in detail is the surface pressure coefficient. It is calculated for selected sections for each element shown in Figure 1.10 separately. Obtained results were calculated by participants for a selected location at two angles of attack using the finest structured and unstructured grids. The results shown in Figure 1.11 were obtained using the SA turbulence model and were captured at the 85% span position for AoA 28°. It can be clearly seen that all results are in excellent agreement except for the flap element, particularly around the trailing edge.

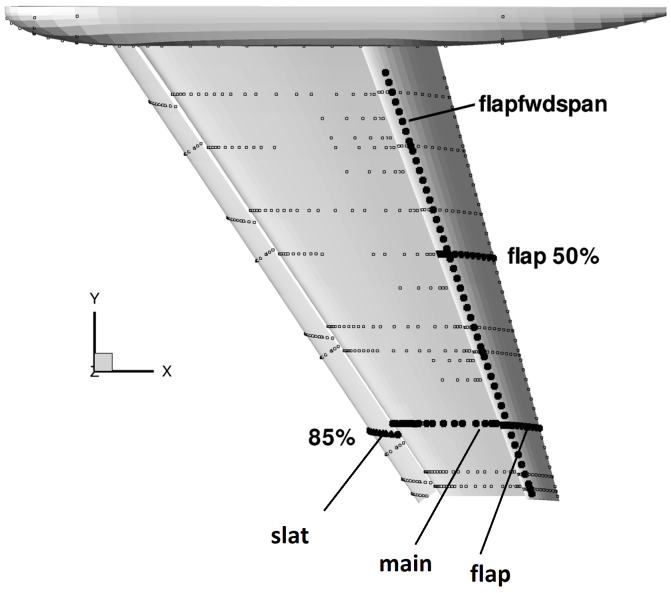


Figure 1.10 : Locations of the pressure stations, AIAA 1st High Lift Prediction Workshop [6]

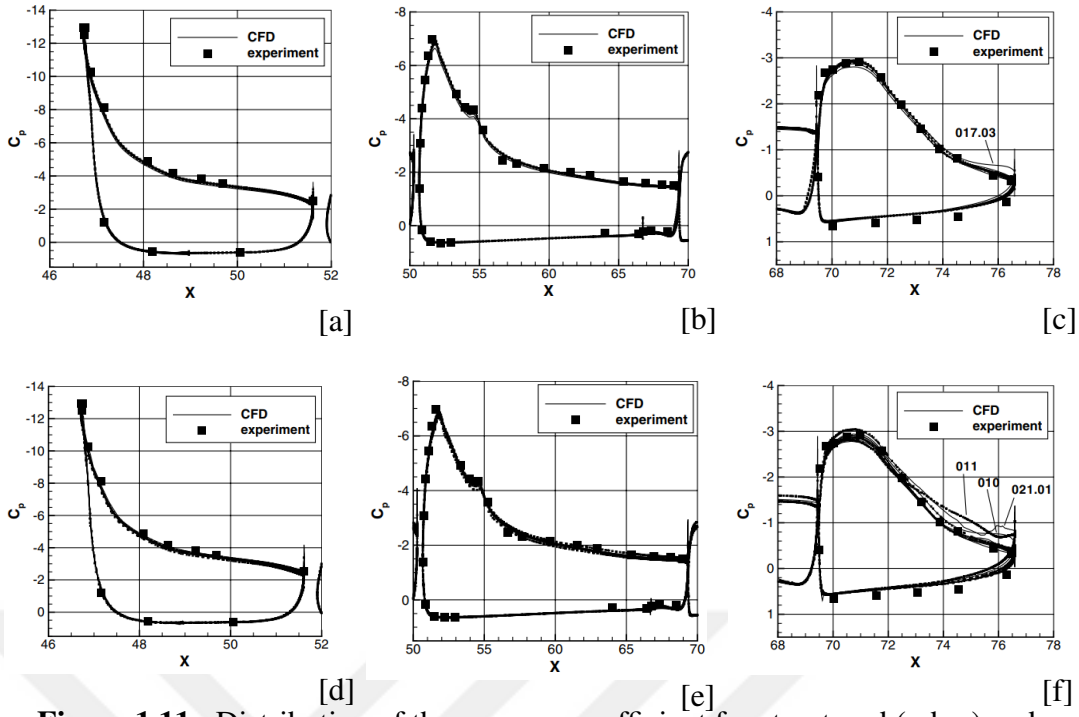


Figure 1.11 : Distribution of the pressure coefficient for structured (a-b-c) and unstructured (d-e-f) grid systems in slat (a & d), main (b & e), and flap (c & f) locations at the 85% span station for Configuration 1 with $\alpha = 28°$, AIAA 1st High Lift Prediction Workshop [6].

In Figure 1.12 [a] and [b], results are compared for two different turbulence models at the 85% flap position for AoA 13° using several structured grid densities. The results in Figure 1.12 [a] were obtained with the SST model, while Figure 1.12 [b] shows results using the SA model. The obtained results reveal that the SST turbulence model predicts flow separation more accurately around the trailing edge of the flap, whereas the SA turbulence model fails to capture it. It can also be observed that the results are independent of grid density for structured grid systems, suggesting that grid convergence is achieved for structured meshes. However, for unstructured grids, shown in Figure 1.12 [c] and [d], accuracy improves as grid resolution increases, highlighting the importance of finer grids in capturing flow behavior. Figure 1.12 further shows that the results obtained using tetrahedral elements [c] and hybrid elements [d] are nearly identical on the finest grid. However, on the coarse grid, tetrahedral elements provide better agreement with experimental data, suggesting that tetrahedral grids may offer a more reliable solution when resolution is limited. Unlike the structured grid results, the pressure coefficient around the trailing edge of the flap does not match the experimental data for unstructured grid systems. This discrepancy indicates that

the unstructured grids struggle to capture the separation and wake effects accurately, particularly at lower resolutions. The comparison of turbulence models highlights the superior performance of the SST model in predicting flow separation, as seen in Figure 1.12 [a]. The SA model, shown in Figure 1.12 [b], underpredicts the extent of separation, leading to discrepancies with experimental data. This suggests that while the SA model is widely used for its computational efficiency, it may not be the most suitable choice for accurately capturing high-lift flow physics, especially in regions of strong adverse pressure gradients and separation.

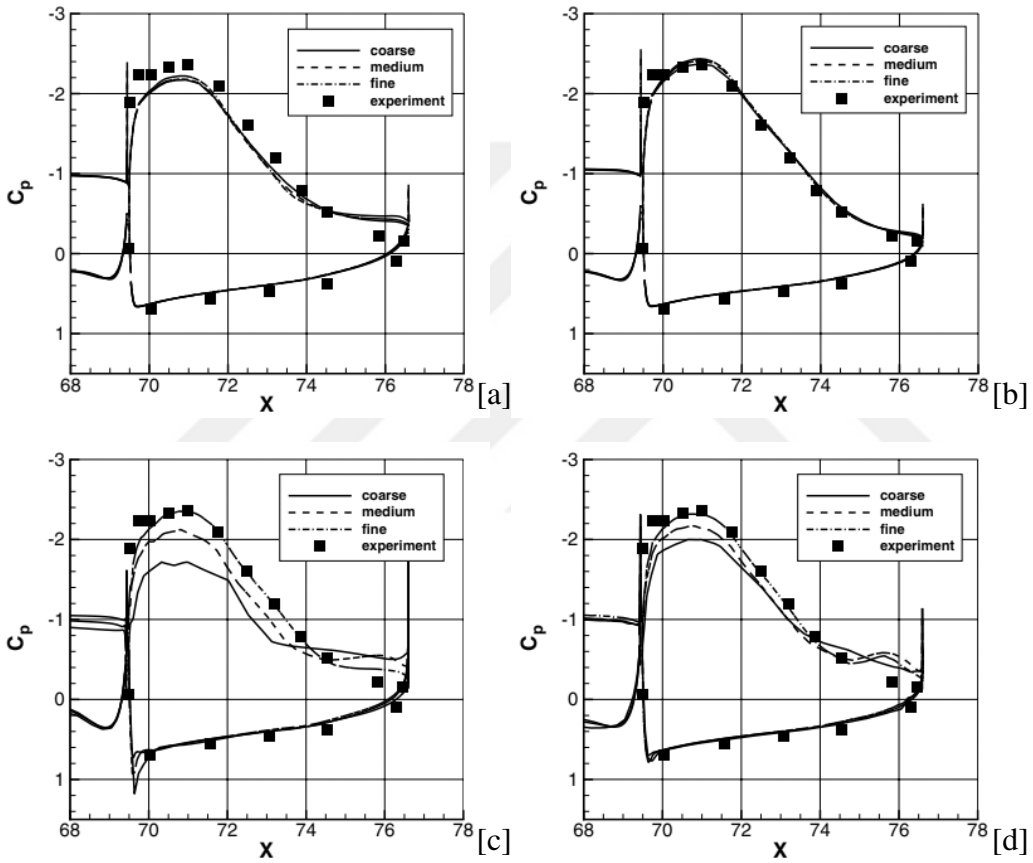
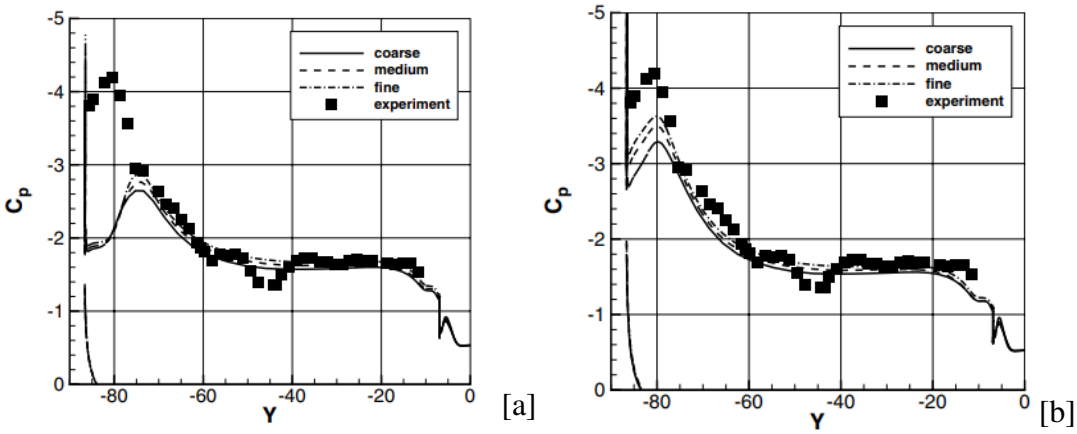


Figure 1.12 : Distribution of the pressure coefficient for structured grids with SST and SA turbulence models, respectively (a-b), and unstructured full tet and hybrid grid systems with the SA turbulence model (c-d) at the 85% flap station for Configuration 1 with $\alpha = 13^{\circ}$, AIAA 1st High Lift Prediction Workshop [6].

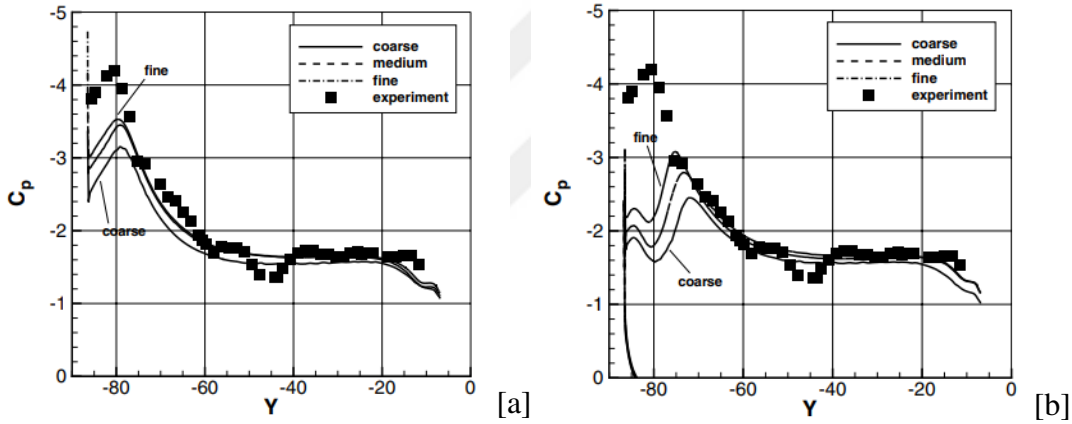
Another important factor in improving the accuracy of predictions around the wingtip is investigating the effect of viscous cross-derivative terms. This is one of the most significant findings from the workshops for future studies. The results obtained while neglecting these terms are shown in Figure 1.13 [a], whereas Figure 1.13 [b] presents the solution of the full Navier-Stokes equations. Note that the same turbulence model and grid structures are used in both cases. It can be clearly seen that the pressure

coefficients on the wingtip show better agreement with experimental data when the viscous cross-derivative terms are taken into consideration.



[a] [b]

Figure 1.13 : Effects of the cross derivative terms on the wing tip at $\alpha = 28°$, using SA turbulence model, AIAA 1st High Lift Prediction Workshop [6].



[a] [b]

Figure 1.14 : Effects of the turbulence models on the wing tip at $\alpha = 28°$, using unstructured tetrahedral grids, AIAA 1st High Lift Prediction Workshop [6].

The full Navier-Stokes solution was applied using both the SA and k-ω turbulence models to understand the effect of turbulence models on the wingtip. As shown in Figure 1.14, the results obtained using the SA turbulence model are closer to the experimental data on a fine grid. Even though the SA model provides better predictions, some discrepancies with the experimental results remain. This issue is attributed to the presence of brackets on the wing geometry, as shown in Figure 1.15. It was stated that these brackets were not included in the numerical results shared so far. Only one study accounted for the bracket effects, and its results captured the flap behavior more accurately. However, differences were still observed at the forward span stations of the flap.

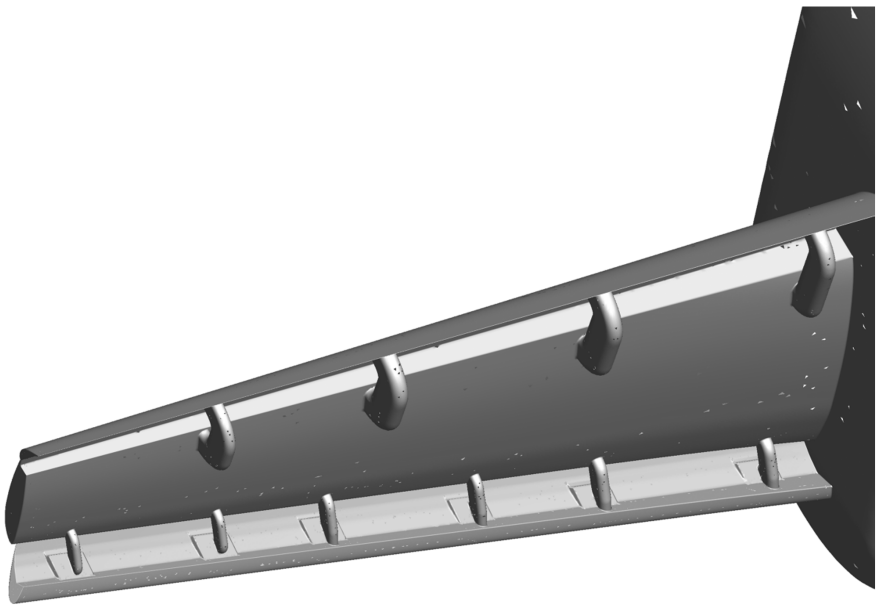


Figure 1.15 : Brackets on the underside of the wing that are not included in the numerical calculations, AIAA 1st High Lift Prediction Workshop [6]

1.5.2 High-lift prediction workshop -2

The second High-Lift Workshop continued with similar test cases, but different aircraft were selected as benchmark geometries. The DLR-F11 model that is shown in the Figure 1.16 is used for the test cases. This geometry is a more realistic aircraft geometry than the previous NASA trapezoidal model. It is studied in landing conditions with two different Reynolds numbers and the participants attended to the workshop are listed in Table 1.1. In addition, the 2D bumps case was used for turbulence model verification.

Figure 1.16 : DLR-F11 in the B-LWST Wind Tunnel, AIAA 2nd High Lift Prediction Workshop [7]

Table 1.1 : AIAA HLPW-2 Commitee List

Entry Number	Code Name	Turbulence Model
002.1	FUN3D	SA
002.2	CFL3D	SA
003.1+	OVERFLOW	SA, SA+AFT
003.2	OVERFLOW	SA-RC
003.3+	OVERFLOW	SA-QCR+AFT
004.1	CFD++	SA-RC
004.2*	CFD++	SA-RC
005.1	HiFUN	SA
005.2+	HiFUN	SA
006	FUN3D	SA-RC
007.1+	CFLOW	SA
007.2+	UG3	SA
007.3*	CFLOW	SA
008	CRUNCH CFD	SST
009.1+	OVERFLOW	SA
009.2*	OVERFLOW	SST
009.3*	OVERFLOW	SST-GRET
010.1+	CFD++	SA
010.2*	CFD++	SST
010.3+	CFD++	K-e-Rt
011.1+	NSMB	SA
011.2+	NSMB	SA-Edwards
011.3*	NSMB	SA-salsa
012.1+	PowerFLOW	LBM-VLES
012.2*	PowerFLOW	LBM-VLES+trans
013.1+	UPACS	SA-noft2-R
013.2*	UPACS	SA-noft2-R-QCR
013.3*	UPACS	SST-V
013.4*	UPACS	SST-V-QCR
013.5*	TAS	SA-noft2-R
013.6*	TAS	SA-noft2-R-QCR
014	UNICORN	SGS
015.1+	CFX	SST
015.2+	Fluent	SST
016.1+	elsa	SA
016.2*	elsa	SA
017	COBRA	SA
018	VULCAN	Wilcox1988

Similar to the first workshop, the effects of grid density, Reynolds number, and bracket effects on the results were investigated.

The SA and other turbulence models were used to validate the turbulence model. The results denoted as 002.1 (SA) and 002.2 (SA) are the validated benchmark results for this case, as shown in Figure 1.17 [a]. The only difference between these two results is the grid structure: 002.1 uses an unstructured grid, while 002.2 uses a structured grid. The dataset 013.1 (SA-noft2-R) captured the most accurate results compared to the benchmark data. Datasets 007.1, 007.2, and 011.1 used the classical SA turbulence model with different flow solvers. However, it is evident that the datasets 011.2 (SA-Edwards) and 011.3 (SA-SALSA) require correction. On the other hand, Figure 1.17 [b] provides insight into the grid convergence results for various test cases. The expected behavior is for the results to converge to those of datasets 002.1 and 002.2 as the grid is refined. The figure shows that only dataset 013.1 correctly predicts the results among the three models.

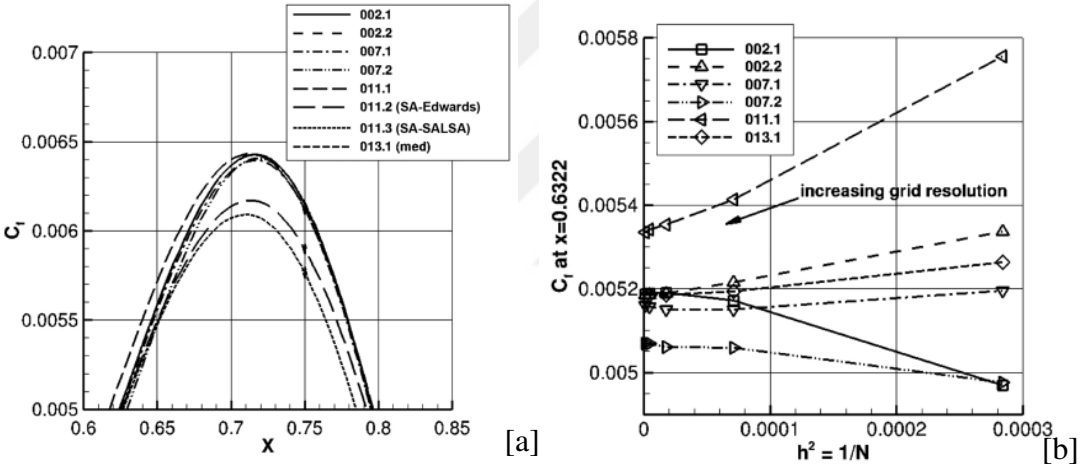


Figure 1.17 : Validation of the turbulence model with 2D bump case, AIAA 2nd High Lift Prediction Workshop [7].

In Figure 1.18, the geometric configuration without brackets was used to compare the results in terms of C_L, C_D, and C_M. It can be seen that most of the results are consistent with the experimental data for lower AoAs. However, the lift coefficient is over-predicted as the AoA increases. Some of the results fail to capture the stall even at AoA 22.4°. A possible reason for this issue is the exclusion of the brackets. Subsequently, all test cases were performed under the same conditions with the brackets. Most of the participants predicted a lower lift coefficient compared to the experimental results. As shown in Figure 1.19, the results obtained spread out at higher AoAs.

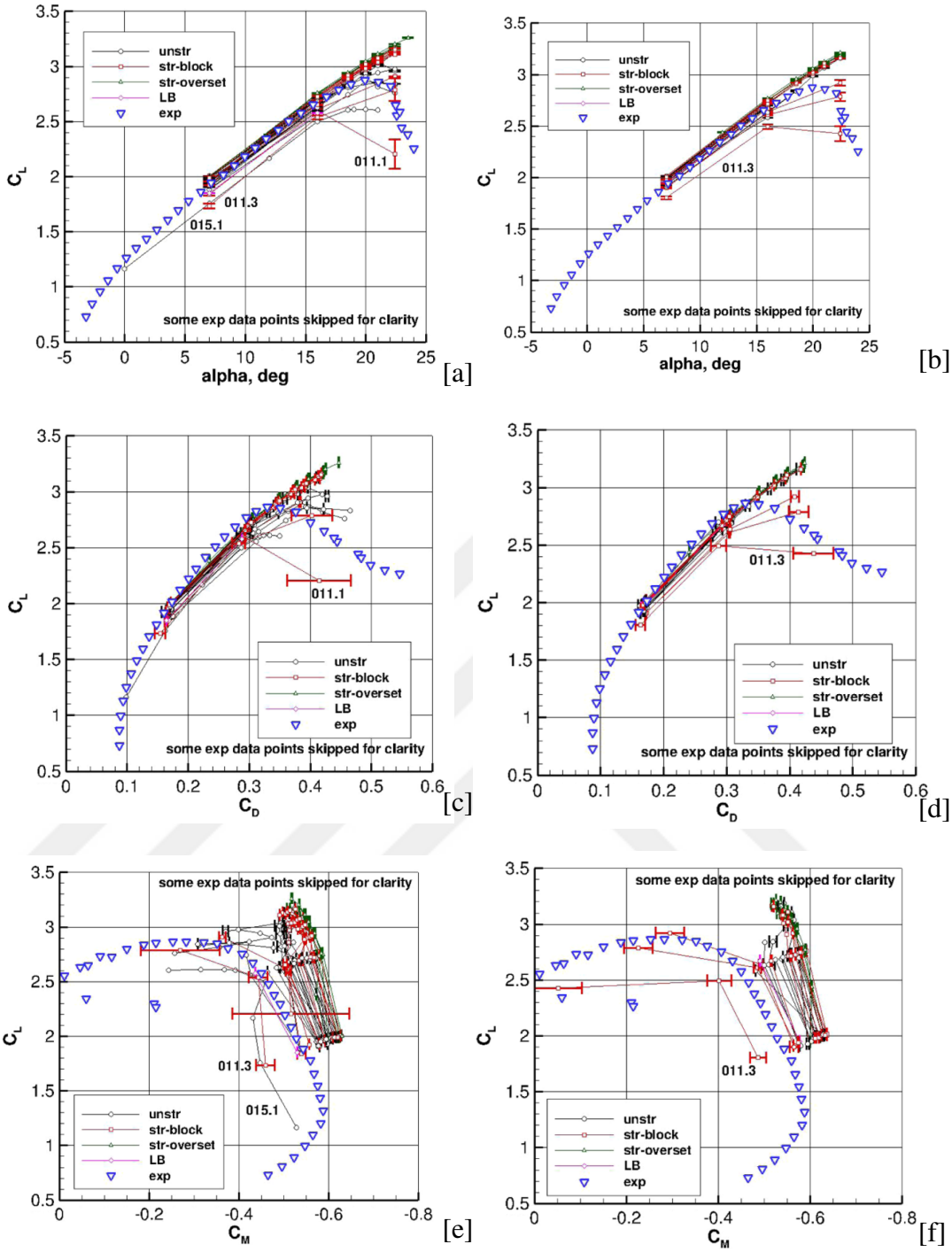


Figure 1.18 : Force and moment calculations for the case bracket effects excluded $Re = 15.1 \times 10^6$, AIAA 2nd High Lift Prediction Workshop [7]

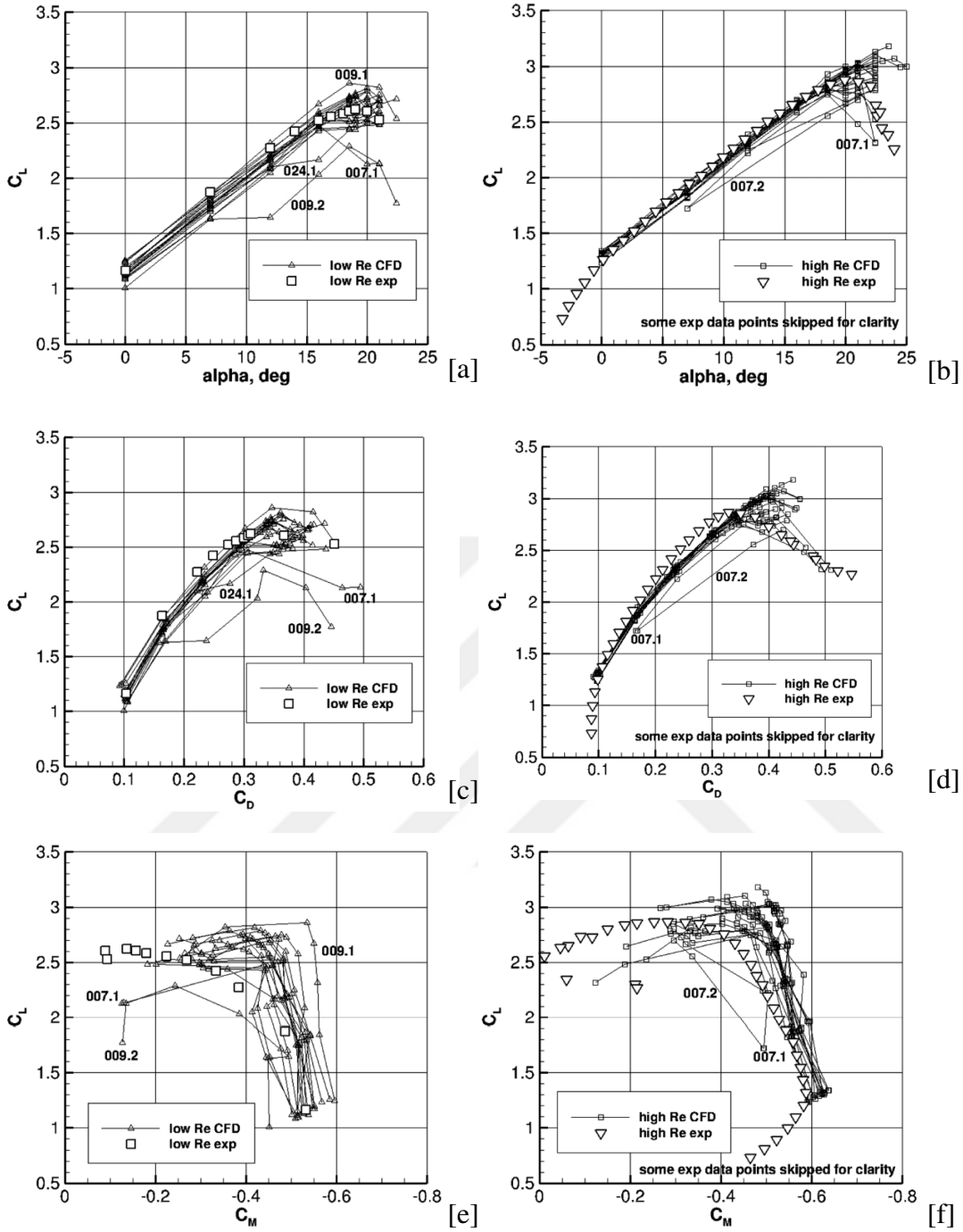


Figure 1.19 : Force and moment calculations for the case bracket effects included for $Re = 1.35 \times 10^6$ and $Re = 15.1 \times 10^6$, AIAA 2nd High Lift Prediction Workshop [7]

Transition effects on the results submitted were investigated. In Figure 1.20, a significant increase in lift is observed due to the transition effects. The lift coefficients were compared, and it was found that they are consistent with those of the NASA trapezoidal wing. Another finding was that including pressure tube bundles had a small effect, except in the stall region [7]. To observe the convergence, some participants

analyzed the residuals. It was emphasized that high-lift cases are generally difficult to converge for most CFD codes due to the complex flow physics around the wing.

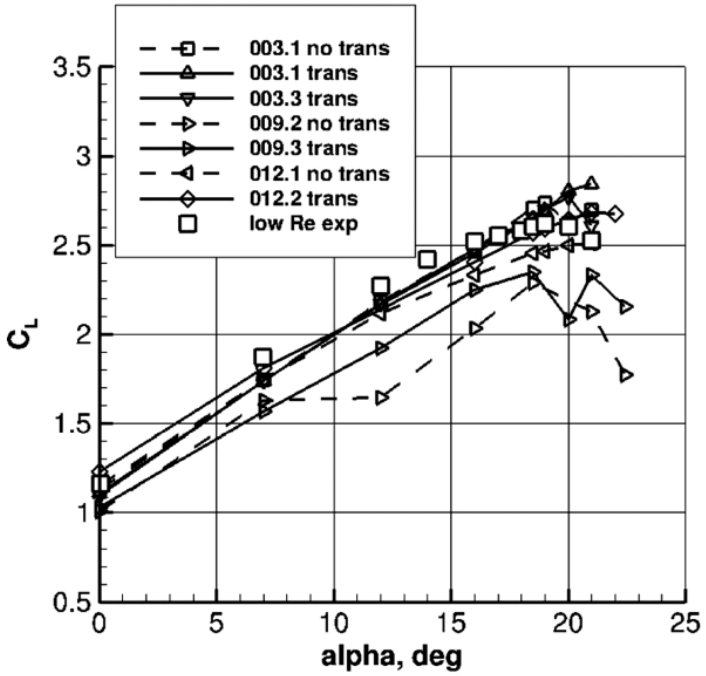


Figure 1.20 : Stall behavior including the transition for $Re = 1.35 \times 10^6$ with the brackets, AIAA 2nd High Lift Prediction Workshop [7]

Velocity profiles were examined at certain locations and compared with the experimental data in Figure 1.21. It is observed that the CFD results are in good agreement with each other at an angle of attack of 7°. However, there are some differences at higher angles of attack, such as 18.5° degrees. It was observed that the results were under-predicted compared to the experimental data. As shown in the Figure 1.22, CFD results for the pressure coefficients are consistent with each other and with the experimental data around the slat and main elements near the mid-span. However, the results on the flap are spread out. Again, the experimental results could not be captured, especially around the flap region.

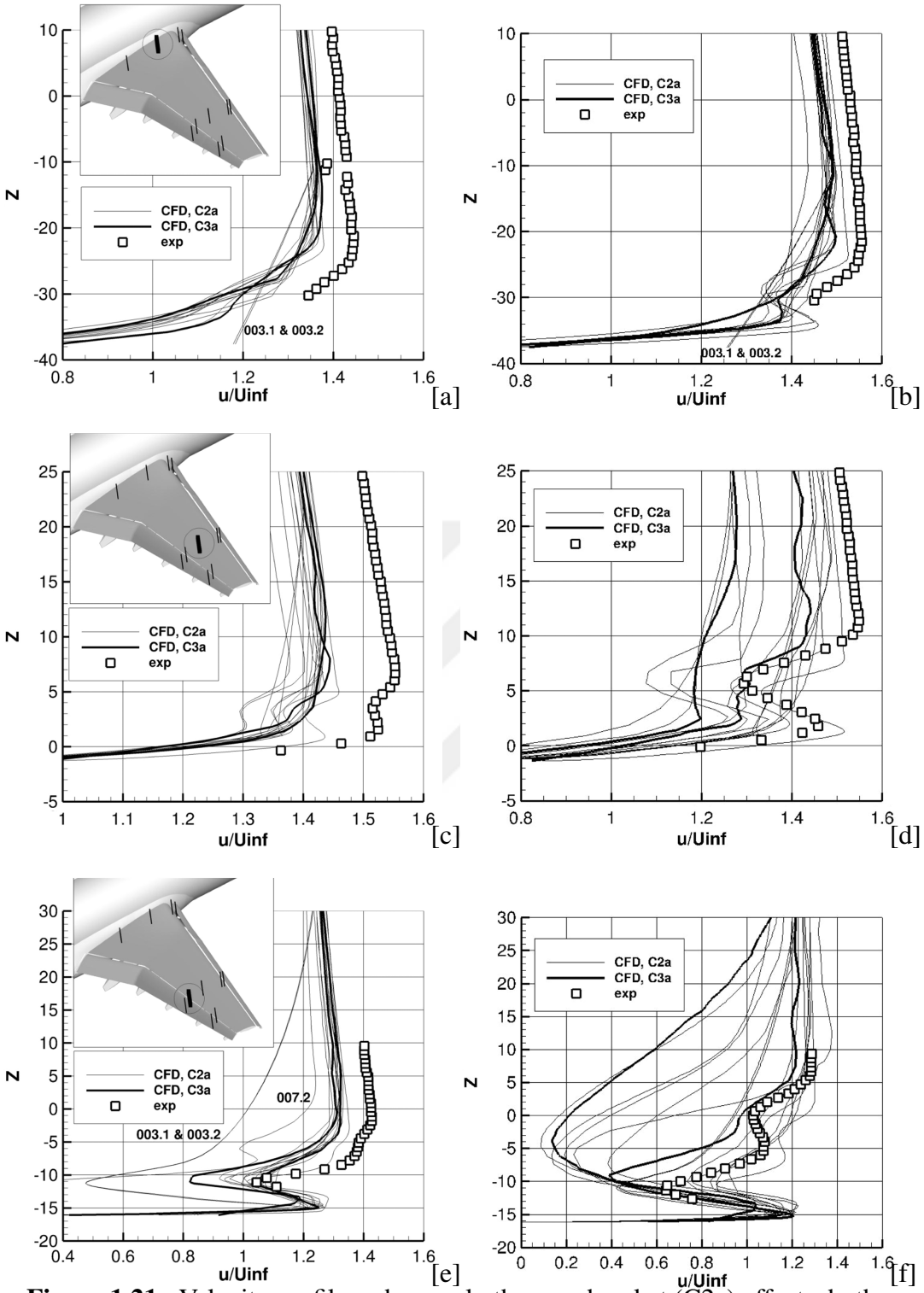


Figure 1.21 : Velocity profiles where only the case bracket (C2a) effects, both brackets and bundles effects included (C3a) for $Re = 1.35 \times 10^6$ with medium grids, AIAA 2nd High Lift Prediction Workshop [7]

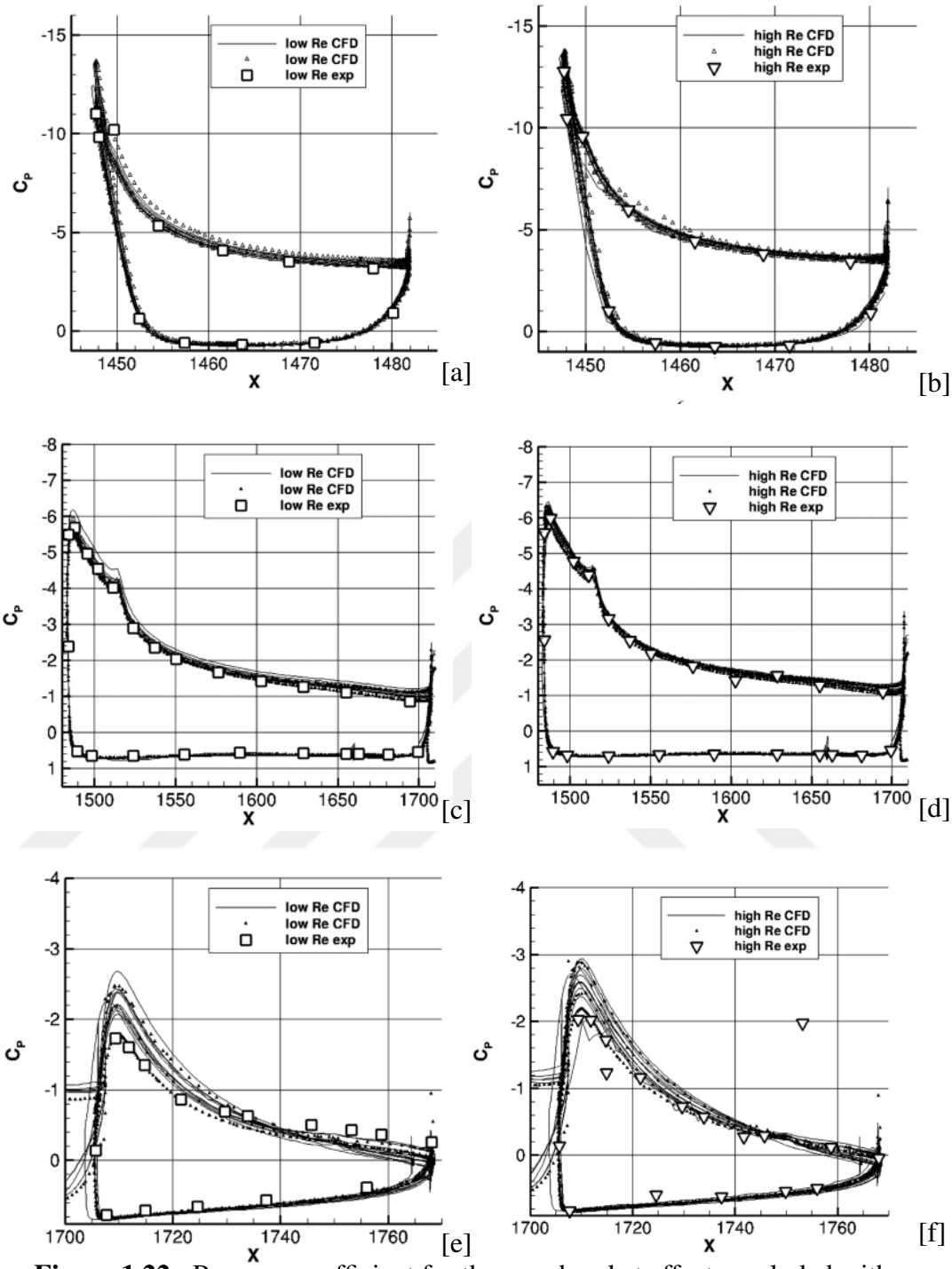


Figure 1.22 : Pressure coefficient for the case bracket effects excluded with $Re = 1.35 \times 10^6$ and $Re = 15.1 \times 10^6$ for slat (a,b), main (c,d), and flap (e,f) regions where α=20°, AIAA 2nd High Lift Prediction Workshop [7]

1.5.3 High-lift prediction workshop -3

In the 3rd High-Lift Workshop, the JAXA Standard Model and NASA-CRM (Common Research Model) were used as benchmark geometries, and nacelle/pylon effects were also taken into account when comparing the CFD results with the experimental data. Additionally, 2D verification was performed for the DSMA66 airfoil. The effects of structured and unstructured grid systems on the results were also investigated. The organizations attended to the workshop and their codes and turbulence models were listed in the Table 1.2. *a* and *b* notation in the entry number field represents the new submission added after workshop and modified after workshop, respectively. There were three main test cases during the workshop, and the details are explained below.

Table 1.2 : HLPW3 Committee Data

Entry number	Primary organization	Code name	Turbulence model
001.1	CARDC	Mflow	SA
002.1^b	U. Oxford	OpenFOAM	SA
002.2^b	U. Oxford	Star-CCM+	SA
003.1	Siemens	Star-CCM+	SST
003.2	Siemens	Star-CCM+	SA
003.3	Siemens	Star-CCM+	Lag-EB-ke
003.4	Siemens	Star-CCM+	SST-gamma
004.1^b	U. Tenn.	COFFE	SA-neg
004.2^b	U. Tenn.	COFFE	SA-neg
004.3^b	U. Tenn.	Kestrel	SA
004.4^b	U. Tenn.	Kestrel	SA
004.5^b	U. Tenn.	KCFD	BSL
004.6^b	U. Tenn.	KCFD	SA
005.1^b	U. Tenn.	OVERFLOW	SA-RC-QCR-
005.2^a	U. Tenn.	OVERFLOW	SA-noft2-RC-QCR
006.1	Metacomp	CFD++	SA-RC-QCR
006.2	Metacomp	CFD++	SA
007.1^b	Boeing	GGNS	SA-QCR
008.1	Kawasaki	Cflow	SA-noft2
009.1	ISCFDC	Arion	SST-2003
009.2	ISCFDC	Arion	SST-2003
010.1^b	ANSYS	Fluent	SA
010.2^b	ANSYS	Fluent	SST[$t = 1$]
011.1^a	ANSYS	Fluent	BSL
012.1	JAXA	TAS	SA-noft2-RC
012.2	JAXA	TAS	SA-noft2-RC-QCR
012.3	Boeing	GGNS	SA-QCR

Table 1.2 (continued) : HLPW-3 Commitee Data

013.1	Exa	PowerFLOW	LBM VLES
016.1	MAI	LOGOS	SSG/LRR-RSM-w2012
016.2	MAI	LOGOS	SA
016.3	MAI	LOGOS	SST
017.1[b]	ARA	TAU	SA
017.2[b]	ARA	TAU	SA-neg
018.1	ONERA	elsa	SA
019.1[b]	EMBRAER	SU2	SA
019.2[a]	EMBRAER	CFD++	SA-RC-QCR
019.3[a]	EMBRAER	CFD++	SA-RC-QCR
019.4[a]	EMBRAER	CFD++	SA-RC-QCR
021.1[b]	IAE	BRU3D	SA
022.1[b]	NASA Ames	OVERFLOW	SA-noft2-RC-QCR
022.2[b]	NASA Ames	OVERFLOW	SA-noft2
022.3[a]	NASA Ames	OVERFLOW	SA-noft2-RC-QCR
023.1	Boeing	BCFD	SA-RC-QCR
023.2	Boeing	BCFD	SA-RC
023.3	Boeing	BCFD	SA
023.4	Boeing	BCFD	SST
024.1	U. Tokyo	UTCart	SA-noft2-R
025.1[b]	CFMS	zCFD	SST-V-sust
026.1[b]	DLR	TAU	SA-neg
026.2[b]	DLR	TAU	SA-neg
028.1	NASA LaRC	PowerFLOW	LBM-VLES
030.1[b]	Bombardier	Dragon	Wilcox1988CC
030.2[b]	Bombardier	Dragon	Wilcox1988CC
030.3[b]	Bombardier	Dragon	Wilcox1988CC
030.4[b]	Bombardier	Dragon	Wilcox1988CC
030.5[b]	Bombardier	Dragon	SST
030.6[b]	Bombardier	Dragon	Wilcox1998
031.1	Dassault	XFlow	WALE
032.1	CSC	Unicorn	Implicit SGS
033.1[b]	NASA Ames	LAVA	SA-noft2
033.2[b]	NASA Ames	LAVA	SA-noft2
034.1	U. Los Andes	SU2	SA
035.1[a]	U. Colorado	PHASTA	SA-noft2
035.2[a]	U. Colorado	PHASTA	SA-noft2-QCR
035.3[a]	U. Colorado	PHASTA	SA-noft2
036.1[b]	TotalSim	FUN3D	SST
036.2[b]	TotalSim	FUN3D	SA-neg
036.3[b]	TotalSim	OpenFOAM	γ-Ret-SST
036.4[b]	TotalSim	OpenFOAM	SA
036.5[b]	TotalSim	OpenFOAM	SST[mod]
038.1[a]	TotalSim	FUN3D	SA
039.2[a]	Gulfstream	FUN3D	SA
040.1	ILight	OVERFLOW	SA-noft2

• *Case 1: Grid Convergence Study*

In this test case, the committee provided fixed grids, and the capability of the CFD solvers was measured for at least one family of coarse, medium, and fine workshop-provided meshes. The NASA-CRM wing-body system which is shown in Figure 1.23 was used as the geometry for a nominal landing configuration without the nacelle, pylon, tail, or support brackets.

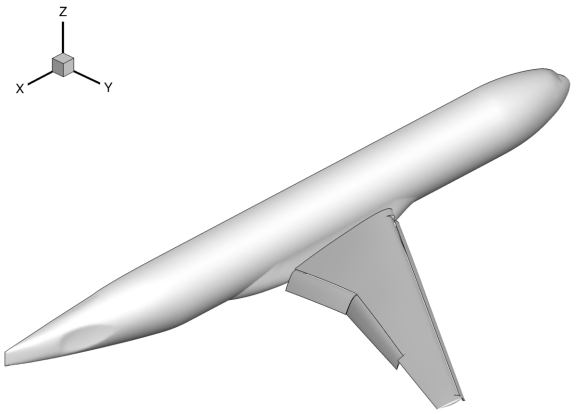


Figure 1.23 : NASA CRM, AIAA 3rd High Lift Prediction Workshop [8]

Case 1a: Full Chord Flap Gap

The test case involves a configuration with a high-lift system that features a full-chord flap on a wing with a gap between the flap and the wing surface. The flow conditions for this case are detailed in Table 1.3. The purpose of this test case is to evaluate the ability of aerodynamic prediction methods to accurately simulate the flow characteristics and performance of a high-lift system with a significant gap between the flap and the wing. This gap affects the airflow around the wing, particularly the lift, drag, and moment coefficients, and plays a role in the overall aerodynamic efficiency of the high-lift configuration.

Case 1b: Full Chord Flap Gap with Adaptation

It is requested to conduct the necessary grid convergence study by employing grid refinement through automatic solution adaptation and/or solution-guided grid regeneration, using the parameters specified in Case 1a.

Case 1c: Partially-sealed Chord Flap Gap

It is requested to obtain numerical solutions specifically for the medium grid, using the flow conditions outlined in Case 1a. The solution should include a partial chord seal

Table 1.3 : Flow conditions for Case 1a

Mach Number	0.2
Re Number	3.26×10^6
Angle of Attack	8.0°, 16.0°
Mean Aerodynamic Chord	275.8*inch*
Reference Static Pressure	14.7*psi*
Reference Static Temperature	518.67°*R*

between the inboard and outboard flaps, as well as between the inboard flap and the side of the body.

Case 1d: Partially-sealed Chord Flap Gap with Adaptation

It is requested to conduct the necessary grid convergence study by implementing grid refinement through automatic solution adaptation and/or solution-guided grid regeneration, using the parameters specified in Case 1c.

• *Case 2: Nacelle Installation Study*

As shown in the Figure 1.24, the JAXA Standard Model (JSM), which represents a wing-body high-lift system, is examined in a nominal landing configuration, featuring a single-segment baseline slat and a single-segment 30° flap, with support brackets and the option for nacelle/pylon inclusion or exclusion. The experiment utilized a semi-span model with a *60 mm* peniche standoff, but the requested computations are to be conducted in "free air."

Figure 1.24 : JAXA, AIAA 3rd High Lift Prediction Workshop [8]

Case 2a: Nacelle/Pylon OFF

This test case examines a wing-body high-lift configuration without the nacelle and pylon and the flow conditions are detailed in Table 1.4. The configuration is modeled with the wing-body system as the baseline, and the effects of the nacelle and pylon on the aerodynamic performance are excluded. This case is designed to evaluate the performance of the high-lift system under ideal conditions, where the additional drag and flow interference caused by the nacelle/pylon assembly are not present. The primary goal of this case is to assess the baseline aerodynamic characteristics of the wing-body system in its clean form and provide a reference for comparison with configurations that include nacelle/pylon components.

Table 1.4 : Flow conditions for Case 2a

Mach Number	0.172
Re Number	1.93×10^6
Angle of Attack	4.36°, 10.47°, 14.54°, 18.58°, 20.59°, 21.57°
Mean Aerodynamic Chord	529.2 mm
Reference Static Pressure	14.458 psi
Reference Static Temperature	551.79°R

Case 2b: Nacelle/Pylon OFF with Adaptation

It is requested to implement grid refinement through automatic solution adaptation and/or solution-guided grid regeneration to obtain the necessary flow solutions, using the parameters specified in Case 2a.

Case 2c: Nacelle/Pylon ON

It is requested to obtain flow solutions for a high-lift configuration with the nacelle/pylon assembly activated, using the parameters outlined in Case 2a.

Case 2d: Nacelle/Pylon ON with Adaptation

It is requested to generate flow solutions for a high-lift configuration with the nacelle/pylon assembly activated, using the parameters specified in Case 2a. Grid refinement will be performed through automatic solution adaptation and/or solution-guided grid regeneration for this purpose.

• *Case 3: Turbulence Model Verification Study*

Figure 1.25 : DSMA661, AIAA 3rd High Lift Prediction Workshop

Turbulence model verification was performed for the DSMA661 airfoil which is given in the Figure 1.25. The main focus was on measuring the flow field in the near wake region, as well as the forces acting on the airfoil. This airfoil was selected for the verification process because it has multiple elements, providing insight into the flow physics of high-lift configurations. The wakes from upstream elements affect the downstream elements in high-lift geometries, which is useful for investigating a 2D case. On the other hand, the verification of 3D cases is more difficult due to the fact that a very high number of grids are required, which increases simulation time and computational cost.

Most of the participants in the third workshop used a computational domain with a 20c length. Domains with a 500c length were also used, but only a small increase in lift was observed (around 0.0006).

The results obtained with the solvers CFL3D and FUN3D, both of which use the SA turbulence model, were validated with the manufactured solutions. Therefore, these results were considered as the reference benchmark results. It was noted that the turbulence models SA-noft2 and SA-neg had no impact on the results compared to the standard SA solution. Six of the 19 results were successfully verified in terms of the SA turbulence model, as shown in Figure 1.26. Figure 1.27 shows that eight of the solvers (CFL3D, FUN3D, Kestrel/COFFE, CFD++, OVERFLOW, BCFD, TAU, and LAVA) have excellent agreement in terms of the velocity profiles. These are calculated using the SA, SA-noft2, and SA-neg turbulence models.

As can be seen in Figure 1.28, the CFD results for the HL-CRM did not converge to an exact solution as the grid was refined. The trend of the curve showed an increase in lift, but a decrease was observed for the moment coefficient. For the drag coefficient, there was no clear trend, unlike the lift curve. On the other hand, the adaptive grid results,

shown with the red lines in Figure 1.28, indicate that the grid starts to adapt in regions where a finer mesh is needed. Therefore, fewer grid points are sufficient to obtain more accurate results.

Numerical solvers and turbulence models were verified with the 2D case, and the results obtained appear to be more consistent than those of the 3D HL-CRM case. The JSM (JAXA Standard Model), both with and without nacelle/pylons, was also selected as the benchmark geometry for validation. It was concluded that the grid has a significant impact on the results, especially in regions where the lift is maximum. Transition effects were also found to be important for the JSM in terms of the lift coefficient. Figure 1.29 shows oil flow visualization for the JSM. The oil flow patterns reveal surface streamlines, boundary layer separation, and vortex structures on the high-lift wing configuration with deployed slats and flaps.

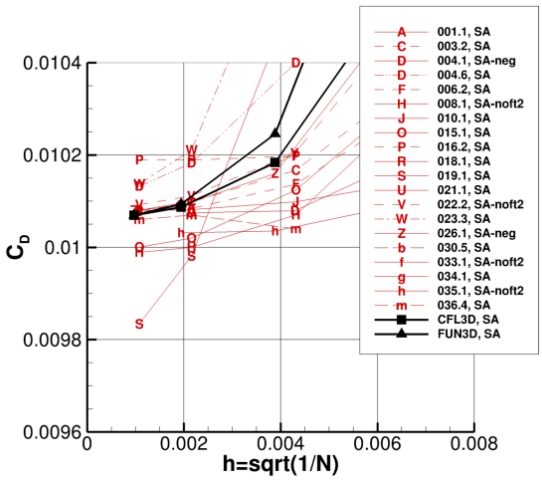


Figure 1.26 : Verification of the drag coefficient for several grid resolutions, AIAA 3rd High Lift Prediction Workshop [8]

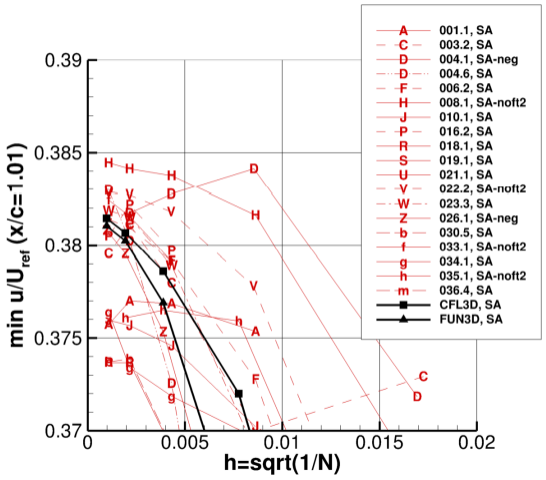


Figure 1.27 : Verification of the velocity profiles for several grid resolutions, AIAA 3rd High Lift Prediction Workshop [8]

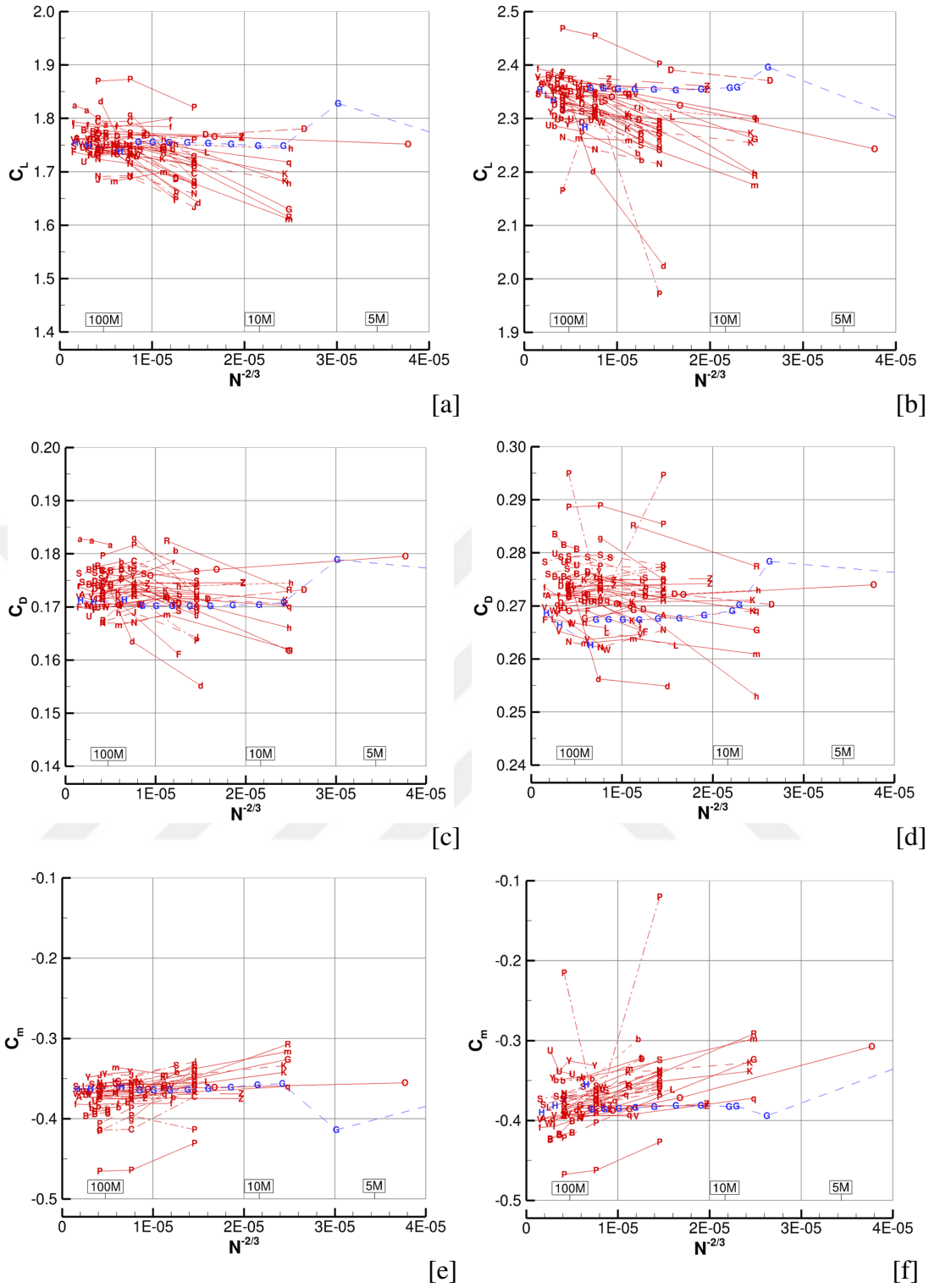


Figure 1.28 : Grid convergence results for HL-CRM where α=8° and α=16°, AIAA 3rd High Lift Prediction Workshop [8]

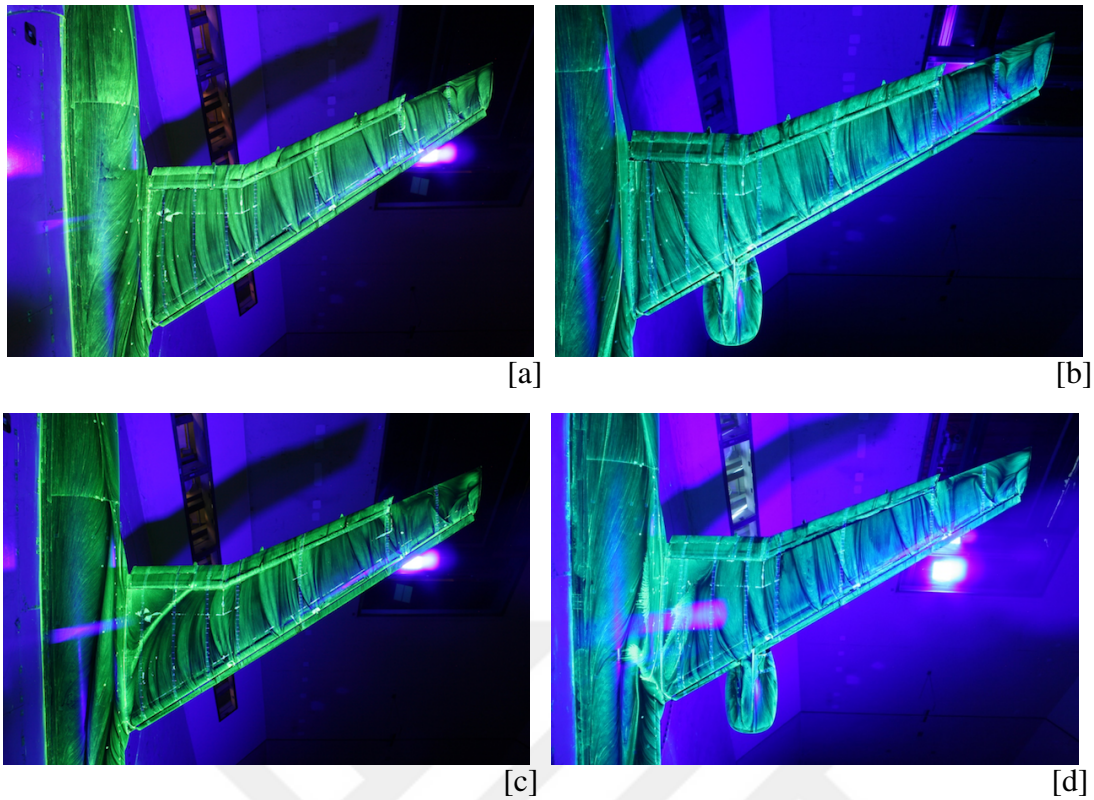


Figure 1.29 : Oil flow pictures for JSM, AIAA 3rd High Lift Prediction Workshop [8]

1.5.4 High-lift prediction workshop -4

The High-Lift Workshop-4 was organized in 2021. The NASA High Lift Common Research Model (CRM-HL), shown in Figure 1.30, was investigated by participants to calculate the maximum lift coefficients and high-lift characteristics at pre- and post-stall angles of attack in landing/take-off configurations.

Another 2D geometry, shown in Figure 1.39, derived from the flap, main, and slat sections, was used for validation cases. The studies conducted during the workshop will be explained in the results section, including our studies, as we also participated in the workshop as committee members. All the test cases studied by the committee members are listed below and compared with the wind tunnel data shown in the Figure 1.31. The participants attended to this workshop and the code used by them are listed in the Table 1.5.

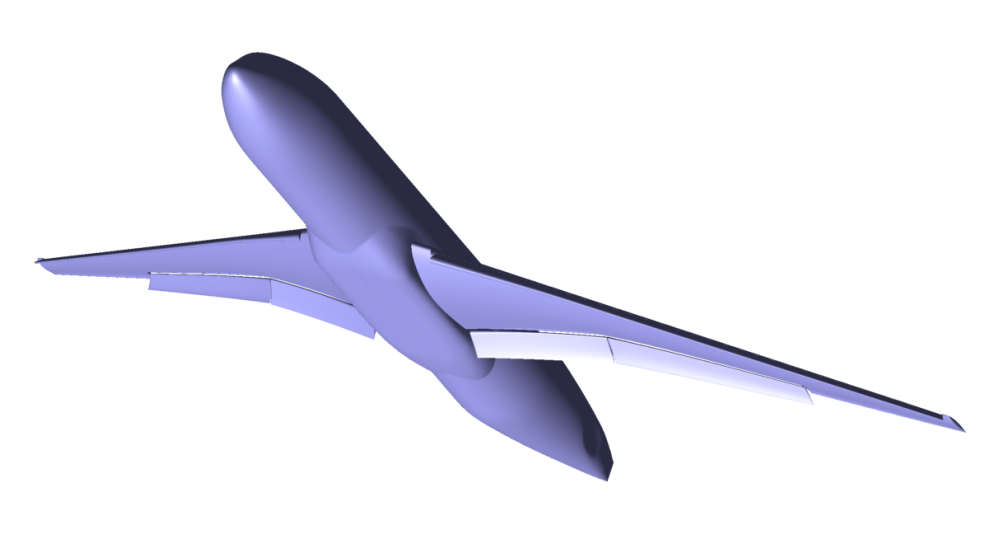


Figure 1.30 : High lift NASA Common Research Model (CRM).

Table 1.5 : HLPW-4 Commitee Members

Entry Number	Primary Organization	Code Name
H-004	MIT	SANS
H-005	U. Tennessee	Kestrel/COFFE
H-012	ONERA	NXO CC-CV
H-013	Princeton	maDG
H-023	Boeing	GGNS-T1
A-002	INRIA	Wolf
A-004	NASA LaRC	FUN3D
A-013	MIT	SANS
R-004	JAXA	TAS
R-008	NASA LaRC	FUN3D
R-009	Hexagon	scFLOW
R-011	Indian Inst. Sci.	HiFUN
R-015	NASA LaRC	USM3D
R-019	Embraer	CFD++
R-021	Zenotech	zCFD
R-025	NASA Ames	LAVA
R-028	ICUBE	NSMB
R-032	Seoul Nat. U.	ACTFlow
R-034	Poly. Montreal	CHAMPS
R-037	Boeing	OVERFLOW
R-043	ARA	TAU
R-050	TU Braunschweig	TAU
R-054	QinetiQ	Fluent
R-057	Kawasaki Heavy	Cflow
R-059	Siemens	STAR-CCM+
R-060	Flexcompute	Flow360

Table 1.5 (continued) : HLPW-4 Commitee Members

Entry Number	Primary Organization	Code Name
A-025	Boeing	GGNS-T1+EPIC
A-026	Istanbul Tech. U.	HEMLAB+PyAMG
A-031	U. Tennessee	Kestrel/COFFE
L-001	Amazon	CFD++
L-004	Virgin Galactic	FUN3D
L-005	U. Tennessee	Kestrel/KCFD
L-016	NASA Ames	LAVA
L-038	DLR	TAU
L-053	Kawasaki Heavy	Cflow
W-020	NASA Ames	LAVA
W-021	Stanford	charLES
W-030	KTH	Euler Real Flight Sim
W-031	Boeing	BCFD
W-032	Dassault Sys.	PowerFLOW
W-034	BSC	Alya
W-047	U. Kansas	hpMusic
W-049	Tohoku U.	FVHFC-ACE
W-050	NASA LaRC	FUN3D

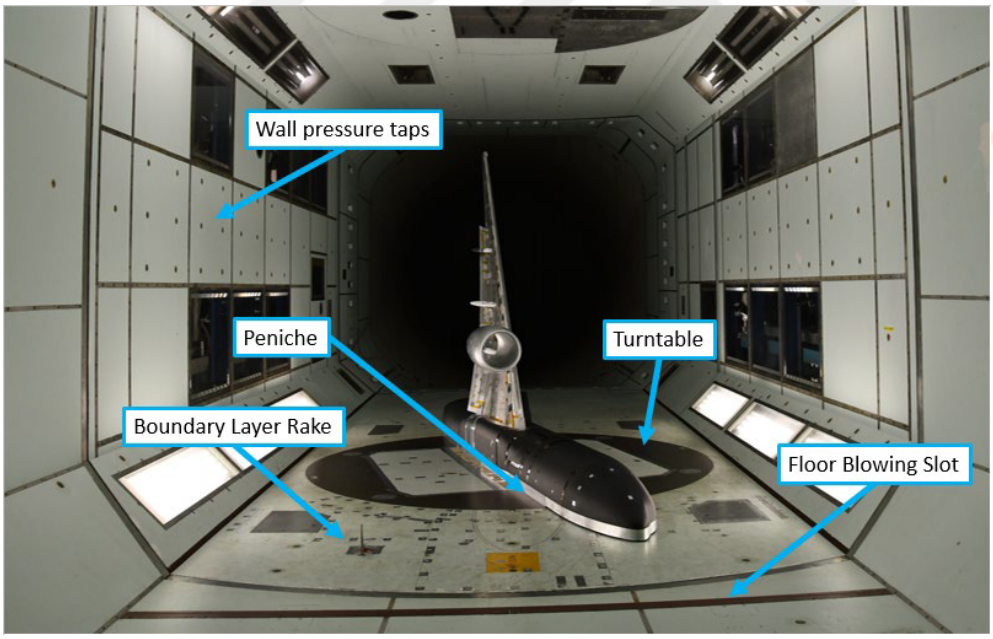


Figure 1.31 : CRM-HL in QinetiQ 5m Wind Tunnel [9]

• *Case 1 - Flap Deflection Study*

In order to compare and assess the capability of CFD methods around the trailing edge, flap deflection increments are studied. All simulations are requested from the committee members at three flap deflections with a constant angle of attack and

Reynolds number.

Case 1a: Comparison with QinetiQ WT Data

In this sub-case the results obtained with the flow conditions given in the Table 1.6 are compared with the wind tunnel data.

Table 1.6 : Flow conditions for Case 1a

Mach Number	0.2
Re Number	5.4910^6
Angle of Attack	7.05° (wall corrected)
Mean Aerodynamic Chord	$275.8 inch$
Reference Static Pressure	$24.67 psi$
Reference Static Temperature	$521.0^\circ R$
Semi-Span Reference Area	$29736.0 in^2$
Moment Reference Center	$x = 1325.9$ inches, $y = 0.0$ inches, $z = 177.95$ inches
Flap Deflection	3 diffrerent geometries: $40^\circ/37^\circ$ inboard/outboard (nominal) $37^\circ/34^\circ$ inboard/outboard $43^\circ/40^\circ$ inboard/outboard

Case 1b: Grid Convergence for Nominal Landing Configuration

Flow solutions are requested for the CRM-HL landing configuration with a nominal $40^\circ/37^\circ$ inboard/outboard TE flap setting, using refined fixed grids to evaluate grid convergence. At a minimum, three mesh-family files provided by the workshop should be used by committee members.

The results obtained for this case during the workshop are compared in terms of lift, drag, and moment coefficients, as shown in Figure 1.32. The $x-$ axis represents the characteristic length, calculated as $h = N^{-1/3}$ since it is a 3D geometry. As the mesh spacing decreases, the differences between submissions become smaller, and the general trend is towards higher lift. Although the variation in the drag coefficient, shown in Figure 1.32 [b], decreases with mesh refinement, the trends of each submission appear more distinct compared to those for lift and pitching moment. Figure 1.32 [c] presents the pitching moment coefficient, where the differences

between submissions decrease with smaller h , and the trend indicates a more negative pitching moment with mesh refinement.

In Figure 1.33, total skin friction contours for four submissions are presented. Each subfigure caption includes the submission identifier, mesh size, and metric formulation. On the wing's upper surface, slat bracket wakes form stripes, while the wakes of flap track fairings disrupt the flow through the gap between the main element and flaps, creating a separated flow region with low skin friction, represented by a black contour on the flap. The steps in the leading edge generate vortices that scrape the upper surface boundary layer. Nacelle chine corner vortices leave an imprint on the nacelle. These features are consistent across submissions, with A-025.1 in Figure 1.33 [c] and A-026 in Figure 1.33 [e] showing less distinct slat bracket wakes.

The pressure coefficient for each section is given in Figure 1.35, 1.36 and 1.37. The locations of the sections can also be seen in Figure 1.34.

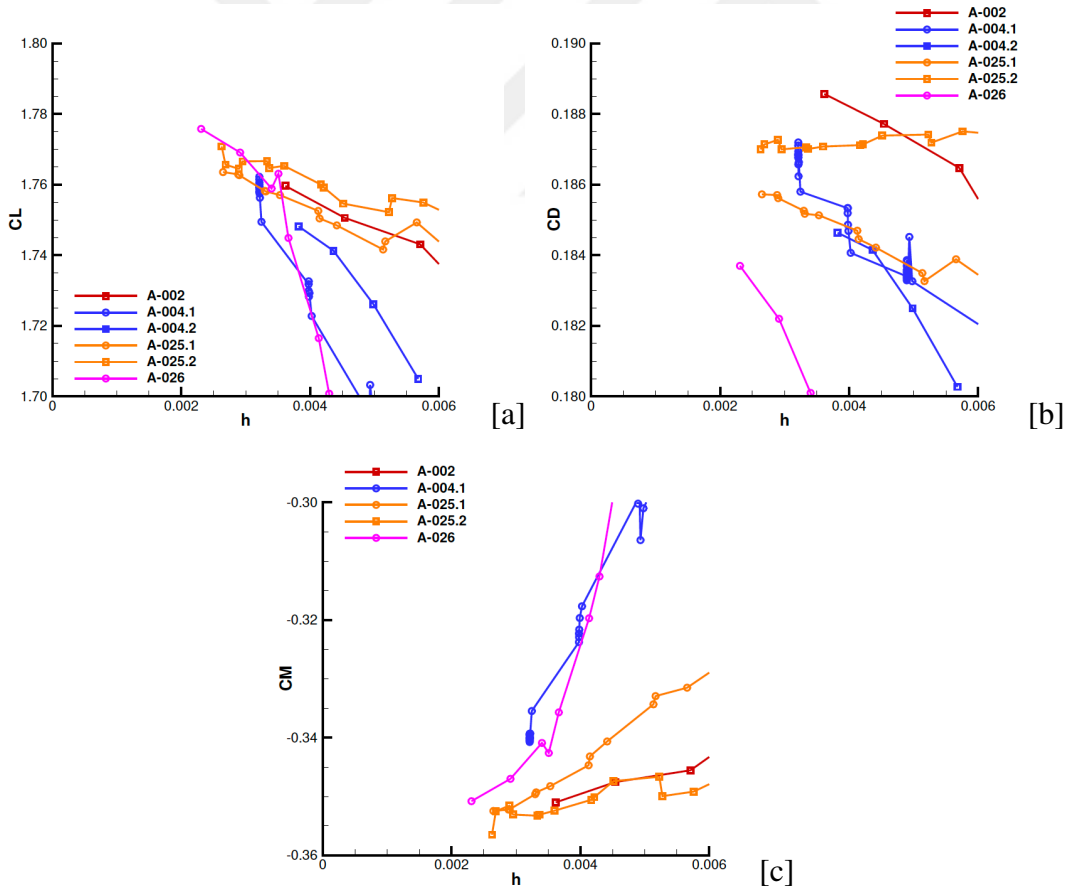


Figure 1.32 : Mesh convergence in terms of lift, drag, and moment coefficient for Case 1 [10]

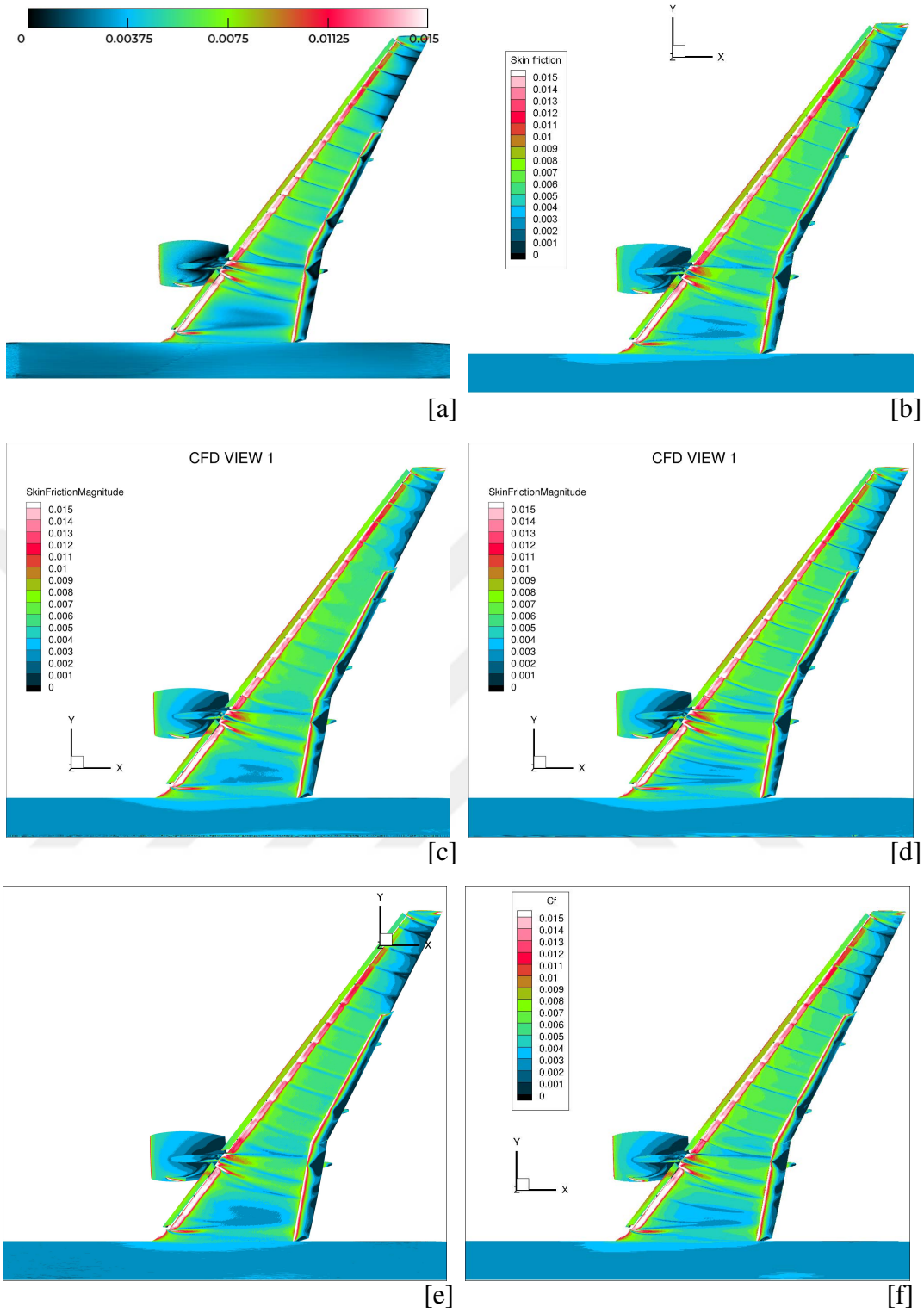


Figure 1.33 : CRM-HL skin friction contours at 7.05° angle of attack where [a] uses the lift metric by A-002, [b] is the multiscale metric by A-004, [c] is the multiscale metric by A-025.1, [d] is the drag metric by A-025.2, [e] is the multiscale metric by A-026, and [f] is the multiscale metric by A-031 [10].

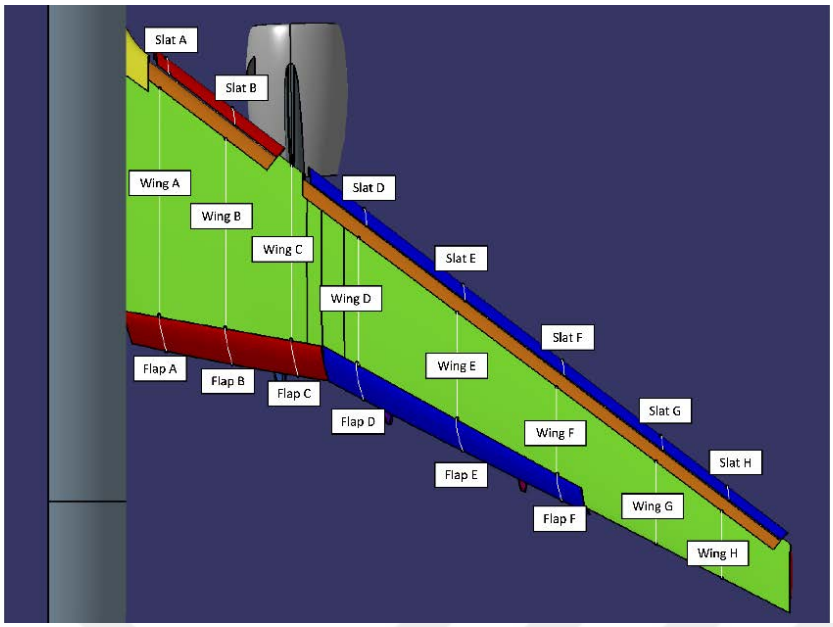
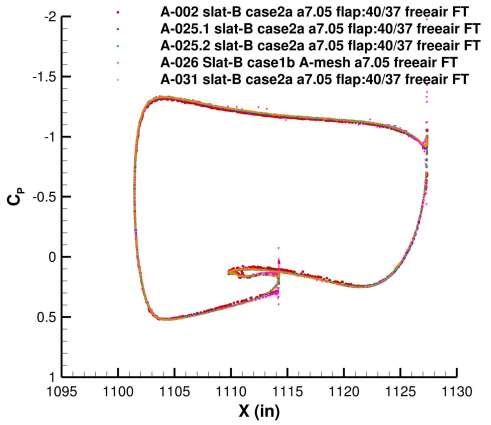
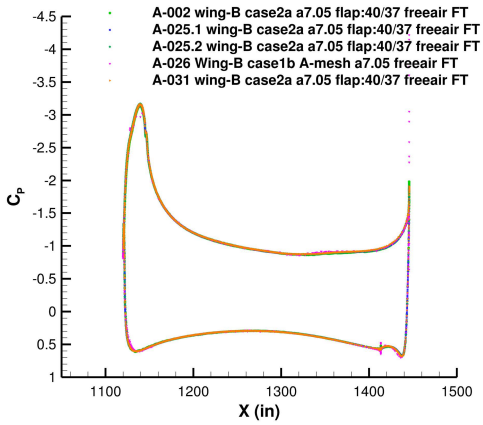


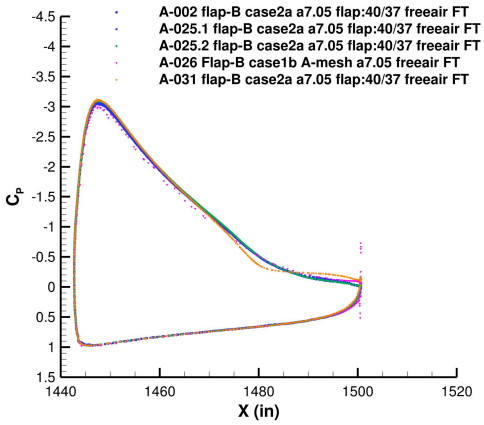
Figure 1.34 : CRM-HL pressure coefficients measurement locations [10].



[a]

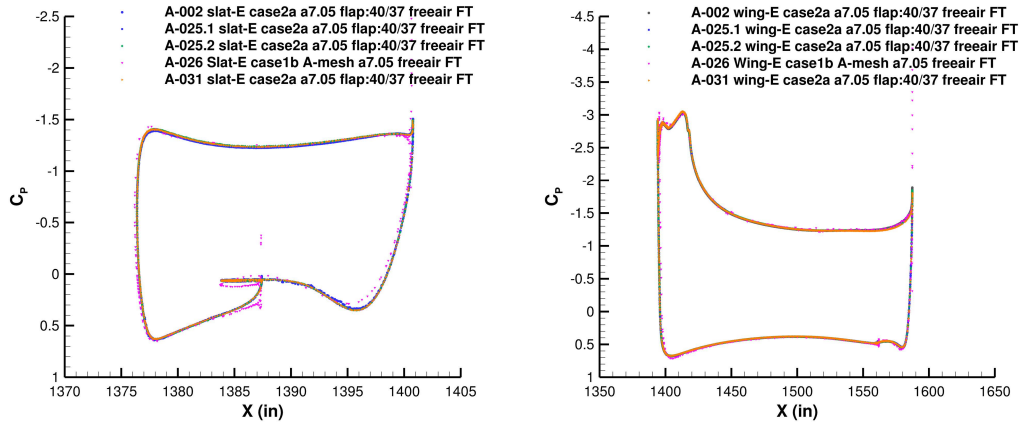


[b]

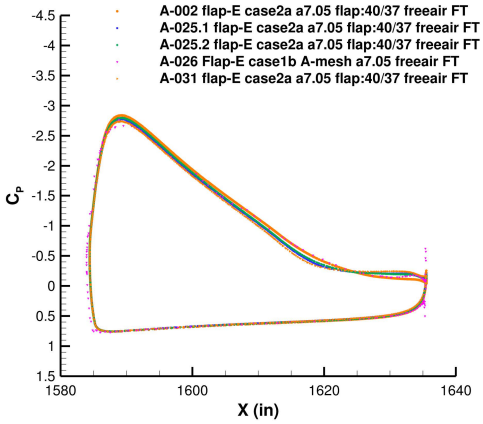


[c]

Figure 1.35 : CRM-HL pressure coefficients for the slat [a], wing [b] , and flap [c] at Section B [10].

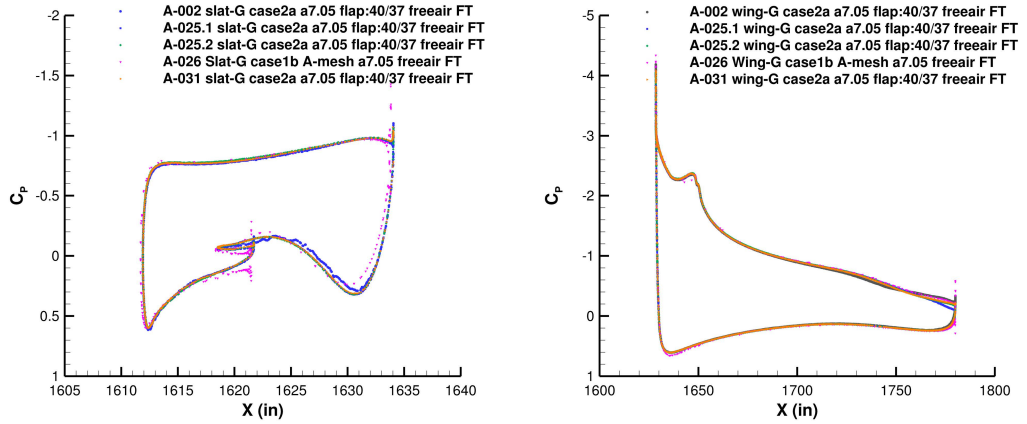


[a] [b]



[c]

Figure 1.36 : CRM-HL pressure coefficients for the slat [a], wing [b], and flap [c] at Section E [10].



[a] [b]

Figure 1.37 : CRM-HL pressure coefficients for the slat [a] and wing [b] at Section G [10].

• *Case 2 -$C_{L,max}$ Study*

In order to measure the maximum lift coefficient using CFD methods, numerical solutions are requested. Simulations should be performed in free air with the CRM-HL flow physics, and the wind tunnel walls should be modeled to compare the results with wind tunnel data.

Case 2a: Flow Physics Exploration for Future WT Testing

This case focuses on investigating key flow physics relevant to future wind tunnel (WT) testing of high-lift configurations. The objective is to analyze critical aerodynamic phenomena, such as flow separation, wake interactions, and boundary layer behavior, to guide the design of future experimental campaigns. Numerical simulations are performed in a free-air setting to assess potential challenges and refine WT testing strategies. The flow conditions are detailed in Table 1.7.

Table 1.7 : Flow conditions for Case 2a.

Mach Number	0.2
Re Number	5.4910^6
Angle of Attacks	2.78°, 7.05°, 11.29°, 17.05°, 19.57°, 20.55°, 21.47° (wall corrected)
Mean Aerodynamic Chord	275.8 inch
Reference Static Pressure	24.67 psi
Reference Static Temperature	$521.0°R$
Semi-Span Reference Area	29736.0 in^2
Moment Reference Center	$x = 1325.9$ inches, $y = 0.0$ inches, $z = 177.95$ inches
Flap Deflection	$40°/37°$ inboard/outboard (nominal)

Case 2b: Comparison with QinetiQ WT Data

This case involves comparing CFD predictions with experimental data obtained from QinetiQ wind tunnel tests. The primary goal is to evaluate the accuracy of different numerical methods in replicating measured aerodynamic forces, moments, and surface pressure distributions. By benchmarking CFD results against QinetiQ data, this case aims to assess model fidelity and identify areas for improvement in high-lift flow simulations. The details of the simulations are listed in Table 1.8.

The results obtained for Case 2 are compared in terms of angle of attack, lift, and drag

Table 1.8 : Flow conditions for Case 2b

Mach Number	0.2
Re Number	5.4910^6
Angles of incidence in tunnel	1.99°, 5.98°, 9.98°, 15.48°, 17.98°, 18.97°, 19.98° (uncorrected)
Mean Aerodynamic Chord	275.8 inch
Reference Static Pressure	24.67 psi
Reference Static Temperature	$521.0°R$
Semi-Span Reference Area	29736.0 in^2
Moment Reference Center	$x = 1325.9$ inches, $y = 0.0$ inches, $z = 177.95$ inches
Flap Deflection	$40°/37°$ inboard/outboard (nominal)

coefficients. Figure 1.38 [a] shows the lift coefficient results for each angle of attack. A-004.1 experienced incomplete iterative convergence, especially for the SA equation, which might be preventing the sudden loss of lift and more negative pitching moment observed post-stall in other submissions with complete iterative convergence. In Figure 1.38 [b], the drag polar is shown, and the scatter at 2.78° narrows the difference with the experiment at 11.29° for some participants. A-004.1 exhibits higher lift/drag and a more negative pitching moment in Figure 1.38 [c], approaching the maximum lift coefficient compared to submissions with machine-level iterative convergence. The results obtained with the adaptation committee submissions predicted lower lift than the experiment at high angles of attack, both before and at maximum lift, possibly due to variations in flow separation patterns.

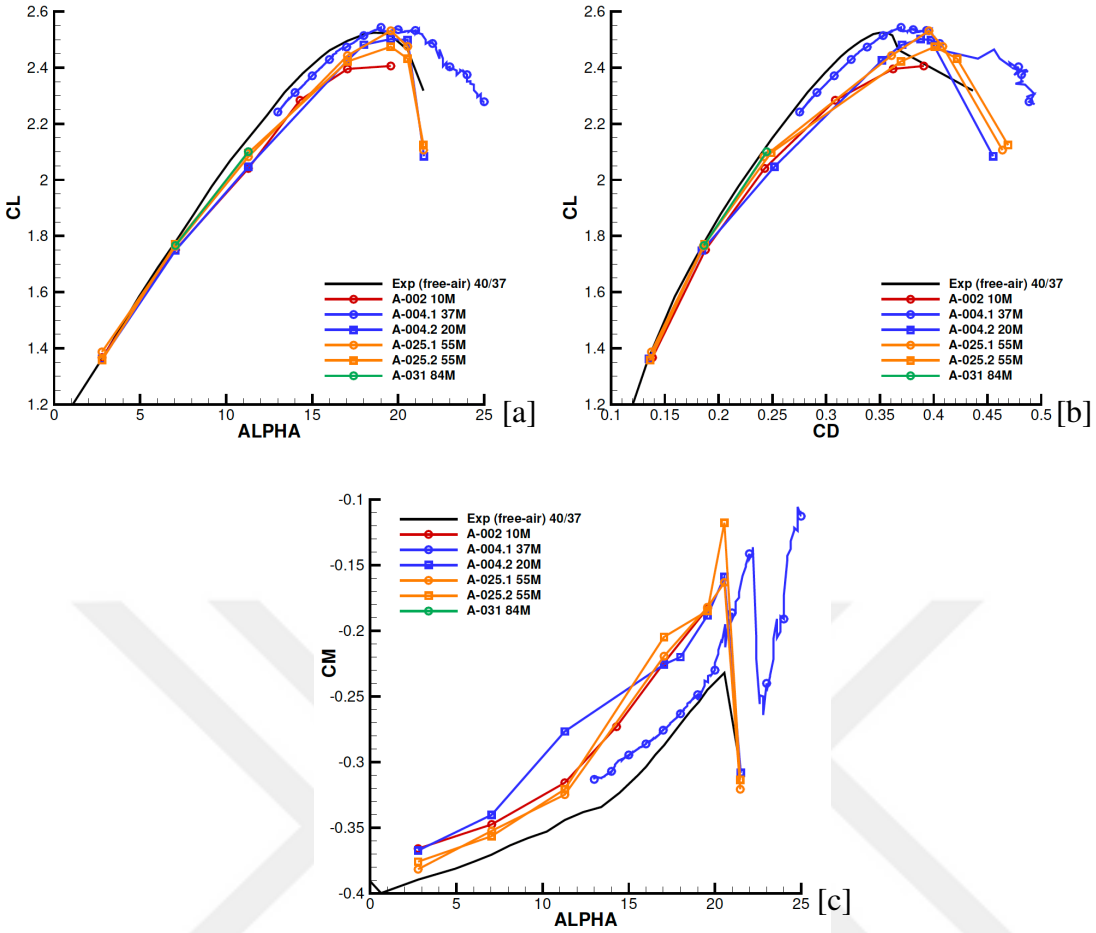


Figure 1.38 : Distribution of the lift, drag, and moment coefficients in terms of angle of attack for Case 2 [10].

• *Case 3 -Turbulence Model Verification Study*

The test case features a simplified two-dimensional (2-D) multi-element airfoil configuration, previously employed in the 2020 AIAA Aviation GMGW Special Session. Flow conditions for this case are listed in Table 1.9. The primary objective of this exercise is to demonstrate the accurate and consistent implementation of the standard Spalart-Allmaras (SA) turbulence model within the context of Reynolds-Averaged Navier-Stokes (RANS), as outlined on the NASA TMR website.

Case 3a: Turbulence Model Verification

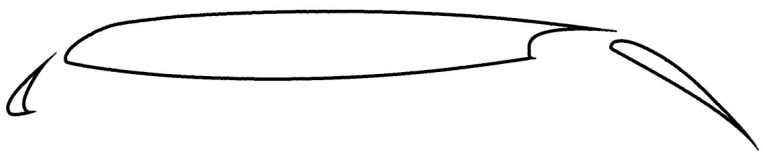


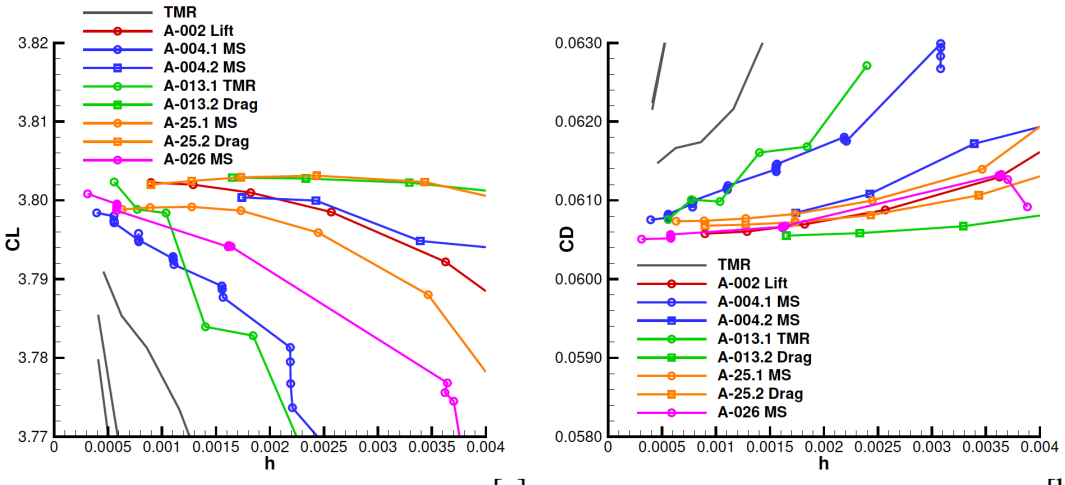
Figure 1.39 : 2D-CRM Multi Element Airfoil.

Table 1.9 : Flow conditions for Case3a

Mach Number	0.2
Re Number	5.0×10^6
Angle of Attacks	16°
Mean Aerodynamic Chord	1.0 m
Ratio of Specific Heats γ	1.4
Reference Static Temperature	$272.1°K$
Prandtl Number	0.72
Turbulent Prandtl Number	0.9
Reference Static Pressure	14.7 psi
$\hat{\nu}_{freestream}/\nu_\infty$	3.0

The results obtained by the mesh adaptation committee are presented in the figures below. Figure 1.40 [a] illustrates the lift coefficient, while Figure 1.40 [b] shows the drag coefficient. The x-axis represents the characteristic length, which is calculated as $h = N^{-1/2}$, since it is a 2D geometry. The gray lines without symbols, identified as Turbulence Modeling Resource (TMR), represent results obtained using uniformly refined, expert-crafted meshes. The adapted-mesh results demonstrate convergence within the range $C_L = [3.799 - 3.802]$ and $C_D = [0.0605 - 0.0607]$.

Notably, the TMR results have not converged. On the other hand, the results obtained with goal-oriented adaptation methods exhibit more rapid convergence to fine-mesh forces compared to multi-scale methods.



[a] [b]

Figure 1.40 : Convergence of the lift and drag coefficients for 2D-CRM case [10].

1.5.5 High-lift prediction workshop -5

The last High-Lift Prediction Workshop (HLPW-5) is organized in August 2024. This workshop is the final installment in a series of workshops focused on advancing the accuracy and reliability of high-lift prediction methods. It aims to assess the performance of computational fluid dynamics (CFD) models and turbulence models in predicting aerodynamic forces, moments, and flow characteristics for high-lift configurations, such as multi-element airfoils and complex geometries. The attendees listed in the Table 1.10 for adaptation focus group.

Table 1.10 : Attendee List for the HLPW-5 Adaptation Focus Group

Submission	Turbulence Model	Solver	Mesher
A-002	SA	HEMLAB	pyAMG
A-003.1	SA	T1	EPIC
A-003.2	SA-R-QCR2000	T1	EPIC
A-003.3	SA	BCFD	EPIC
A-004.1	SA	WOLF	FEFLO.A
A-004.2	SA-R-QCR2000	WOLF	FEFLO.A
A-006.1	SA	Fluent	PUMA
A-006.2	SST	Fluent	PUMA
W-001	None	Adaptive Euler	-

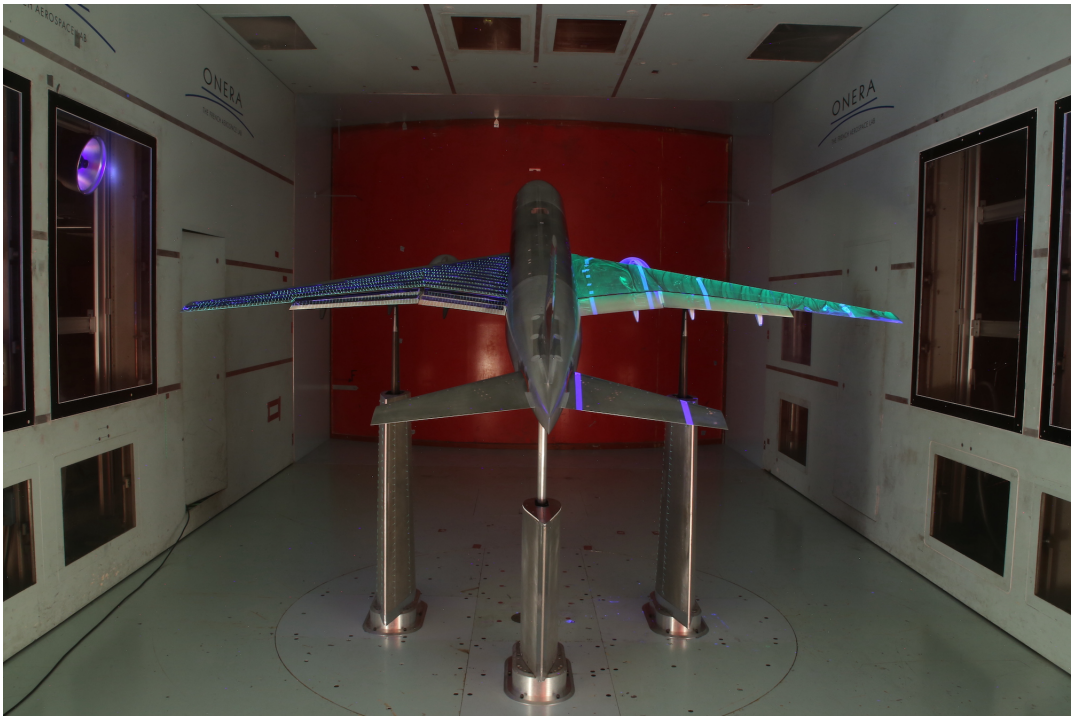


Figure 1.41 : CRM-HL Wind Tunnel

Three main test cases have been requested from the committee members, which are explained below. As with previous workshops, participants will provide simulation results for a set of predefined test cases. These results will be compared with experimental data which is studied in the wind tunnel shown in Figure 1.41 to evaluate the effectiveness of different computational approaches. The goal of HLPW-5 is to push the boundaries of high-lift prediction, offering valuable insights for the design of future aircraft configurations.

• *Case 1 - CRM-HL Wing-Body Verification*

This test case uses the CRM-HL Wing Body (CRM-HL-WB) geometry, as given in Figure 1.42. This configuration closely resembles the simplest version of Test Case 2, excluding the empennage and flap fairings. Details of the flow conditions can be found in Table 1.11. The results obtained for Case 1 are analyzed separately for SA and SA-QCR-2000 models in terms of mesh convergence.

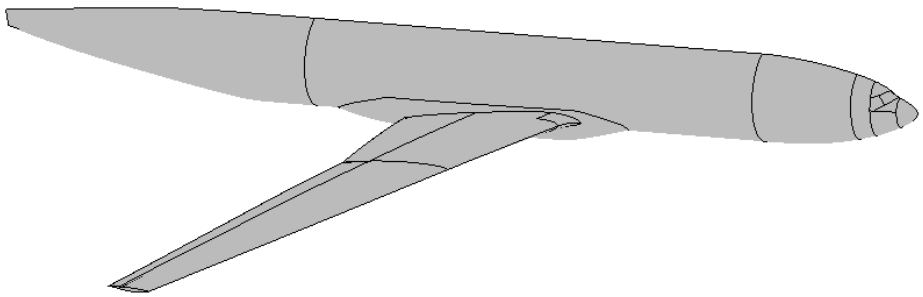


Figure 1.42 : CRM-HL Wing Body Configuration

Figure 1.43 presents the mesh convergence trends for Test Case 1 using the SA model. The x-axis represents a characteristic mesh length, defined as $h = N^{-1/3}$, where N corresponds to the number of control volumes or degrees of freedom. As h approaches zero, the characteristic mesh length diminishes, representing an infinitely refined mesh with an extremely large N. Submissions A-002, A-003.1, and A-004.1 exhibit a mesh convergence pattern similar to most of the selected Fixed Grid RANS TFG submissions. However, A-003.1 and A-004.1, which employ a goal-based adaptation strategy, achieve fine-mesh accuracy on relatively coarser meshes than A-002, which follows a multiscale adaptation approach. The convergence of A-006 stalls, likely due to the subdivision of interior mesh polyhedra while the surface and boundary layer

Table 1.11 : Flow conditions for Test Case 1 (CRM-HL-Wing-Body)

Mach Number	0.2
Re Number	5.6×10^6
Angle of Attacks	11.0°
Mean Aerodynamic Chord	275.8 inch
Reference Static Temperature	$521.0^\circ R$
Semi-Span Reference Area	29736.0 in^2
Moment Reference Center	$x = 1325.9$ inches, $y = 0.0$ inches, $z = 177.95$ inches
Prandtl Number	0.72
Turbulent Prandtl Number	0.9
$\hat{\nu}_{freestream}/\nu_\infty$	3.0
Ratio of Specific Heats γ	1.4

mesh remain unchanged. A-002, A-003.1, and A-004.1 refine tetrahedral meshes to ensure metric conformity across the interior, boundary layers, and surfaces.

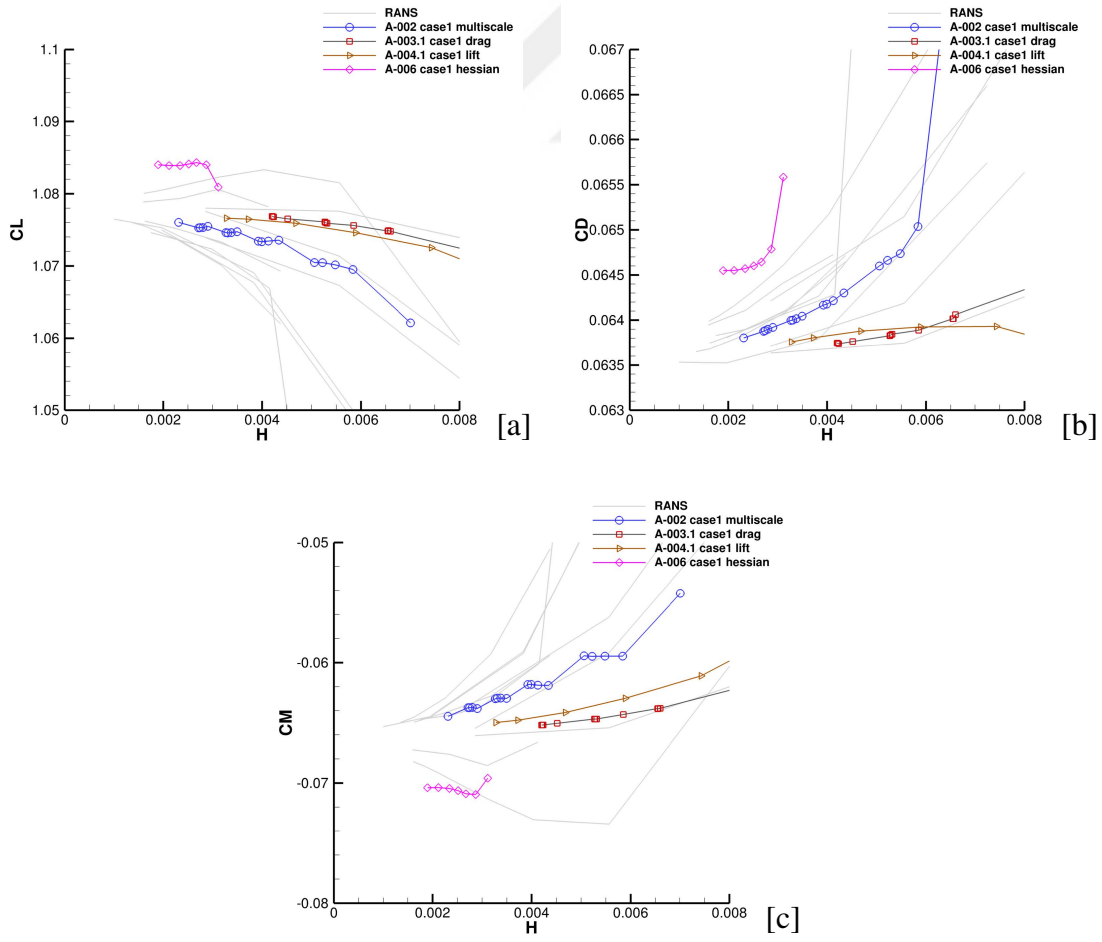


Figure 1.43 : Distribution of the lift, drag, and moment coefficients with SA model for Case 1 [11].

Additionally, Figure 1.44 illustrates the same case using the SA-R-QCR2000 ($C_{rot} = 1.0$) model. Submission A-003.2 shows a trend toward fine-mesh values of the selected Fixed Grid RANS TFG submissions, even at larger h or smaller N. The differences in fine-mesh lift, drag, and pitching moment between SA-R-QCR2000 ($C_{rot} = 1.0$) and SA highlight the sensitivity of this test case to variations in SA turbulence models. This sensitivity suggests that Test Case 1 serves as a valuable verification case for assessing the impact of different SA variants.

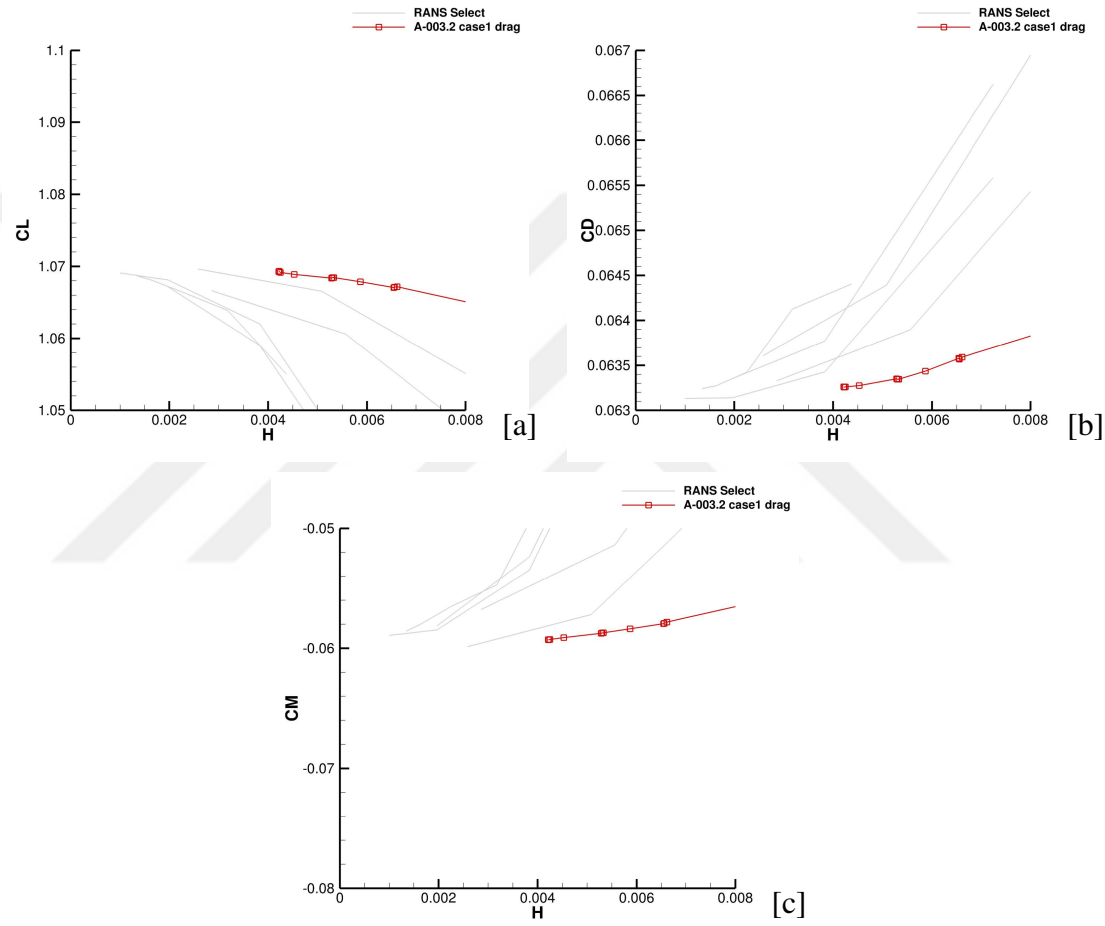


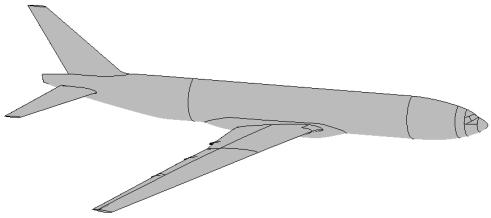
Figure 1.44 : Distribution of the lift, drag, and moment coefficients with QCR model for Case 1 [11].

• *Case 2 - Configuration Build-up*

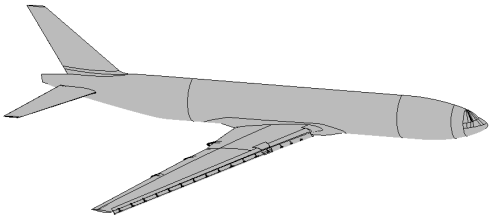
It was requested to perform numerical simulations using four different configurations of the classical CRM-HL Wing-Body configuration. Each configuration used in this case is shown in Figure 1.45 [a], [b], [c], and [d], and their flow conditions are listed in Table 1.12.

Table 1.12 : Flow conditions for Test Case 2

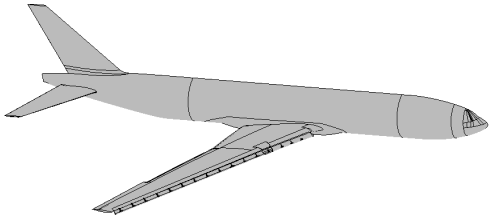
Mach Number	0.2
Re Number	5.4×10^6 (subcase 2.1), 5.6×10^6 (subcases 2.2 - 2.4)
Angle of Attacks	6.0°,10.0°,12.0°,13.0°,14.0° (subcase 2.1) 6.0°,10.0°,17.7°,20.0°,21.5°,23.0°,23.8° (subcase 2.2) 6.0°,10.0°,14.0°,16.0°,17.7°,20.7°,23.5° (subcase 2.3) 7.6°,10.0°,14.0°,16.0°,17.7°,19.7°,23.6° (subcase 2.4)
Reference Static Temperature	$518.67°R$
Reference Static Pressure	14.696 psi



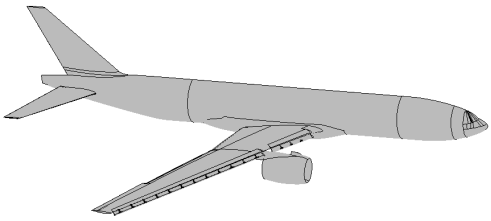
Case 2.1 - Wing-Body with HV (CRM-HL-WBHV) [a]



Case 2.2 - Wing-Body-Slat with HV (ONERA_LRM–WBSHV) [b]



Case 2.3 - Wing-Body-Slat-Flaps with HV (ONERA_LRM–WBSFHV) [c]



Case 2.4 - Wing-Body-Slat-Flaps-Nacelle with HV (ONERA_LRM–LDG) [d]

Figure 1.45 : Various configurations for CRM-HL Case 2, each labeled with their respective case numbers and descriptions.

Figure 1.46 presents the ADAPT TFG submissions for Case 2.1 alongside for selected submissions. Due to the clean wing and body configuration in Case 2.1, the differences among the ADAPT TFG submissions remain minimal. Below a 14° angle of attack, the variation between methods is relatively low and aligns with the trends observed in Case 1. This suggests that incorporating the vertical and horizontal tail surfaces does not substantially increase the level of variation in the results.

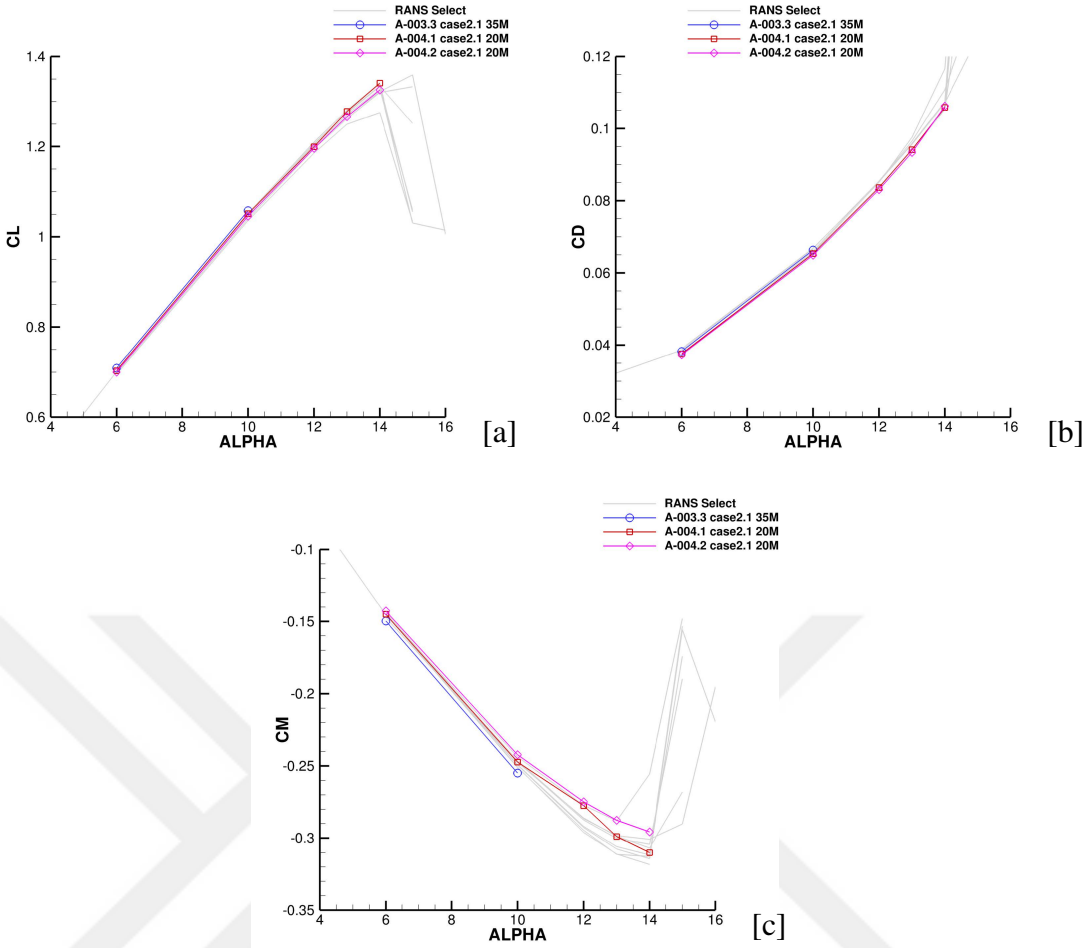


Figure 1.46 : Distribution of the lift, drag, and moment coefficients for Case 2.1 [11].

Figure 1.47 shows the submissions for Case 2.2. Among these results, W-001 utilizes an adapted unsteady inviscid approach, while the remaining entries employ adapted steady RANS methods. While W-001 exhibits similar lift and drag characteristics to the RANS methods, its pitching moment trends differ.

Generally, the ADAPT TFG submissions demonstrate higher lift, lower drag, and a more negative pitching moment compared to the Fixed Grid RANS submissions. Additionally, the variation among ADAPT TFG submissions is lower than that observed in the Fixed Grid RANS cases.

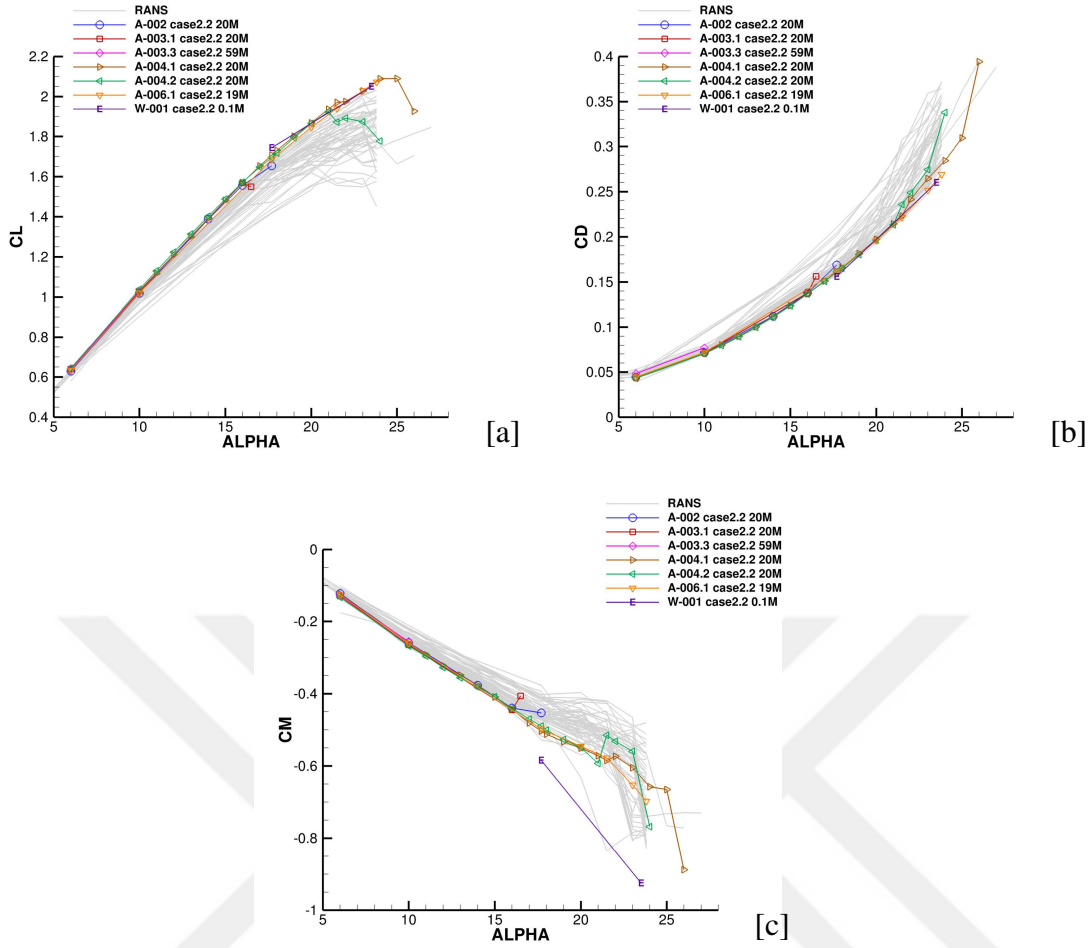


Figure 1.47 : Distribution of the lift, drag, and moment coefficients for Case 2.2 [11].

Figure 1.48 gives information for the Case 2.3 alongside Fixed Grid RANS TFG submissions. Compared to the simpler Case 2 geometries, the number of submissions is lower, while the differences between them have increased. The presence of trailing edge flaps in Case 2.3 leads to higher lift and circulation, which contributes to the increased variation among methods. Submission A-004.1 produces lower lift and drag compared to submission A-006.1. Additionally, A-004.1 exhibits an increase in drag and pitching moment slopes at a lower angle of attack than A-006.1, indicating different aerodynamic behavior. The adapted inviscid method, W-001, shows distinct lift and pitching moment trends compared to the RANS-based methods, further emphasizing the effect of different modeling approaches in this case.

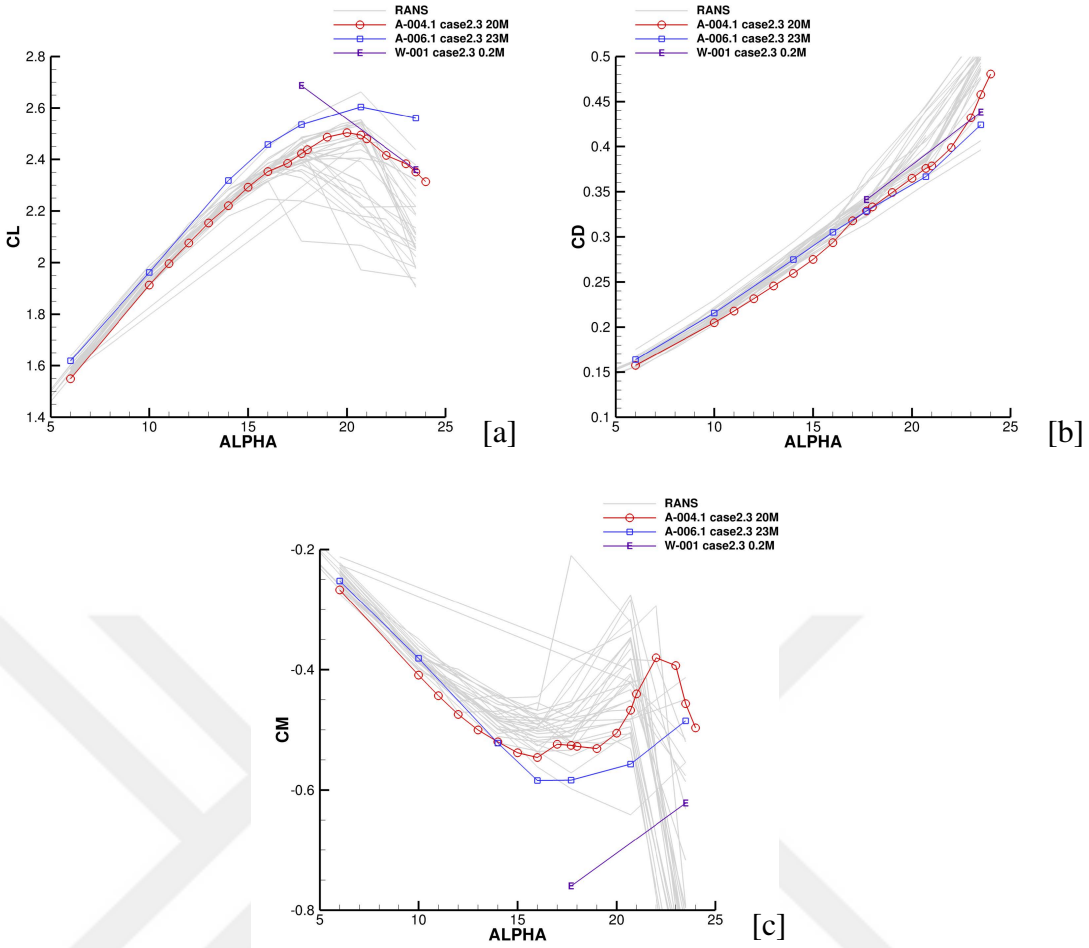


Figure 1.48 : Distribution of the lift, drag, and moment coefficients for Case 2.3 [11].

The results for the Case 2.4 can be found in the Figure 1.49. Compared to other simpler Case 2 geometries, the number of submissions is lower, and the differences between them are similar to those observed in Case 2.3.

The introduction of the nacelle and pylon in Case 2.4 results in a smaller increase in variation compared to the addition of trailing edge flaps in Case 2.3. Similar to Case 2.3, submission A-004.1 produces lower lift and drag than submission A-006.1. Additionally, A-004.1 shows an increase in pitching moment slope at a lower angle of attack than A-006.1, indicating a shift in aerodynamic behavior. While the adapted inviscid submission W-001 captures overall trends, its drag and pitching moment deviate from the RANS submissions at higher angles of attack. W-001 aligns with some lift and drag characteristics of the RANS submissions, but its pitching moment remains more negative at higher angles of attack.

The addition of the nacelle and chine in Case 2.4 may enhance the influence of large vortical structures, which are more prominently captured by the inviscid method. This

could explain the differences in angle-of-attack trends between W-001 and the RANS submissions, distinguishing Case 2.4 from Case 2.3, where such trends were more aligned.

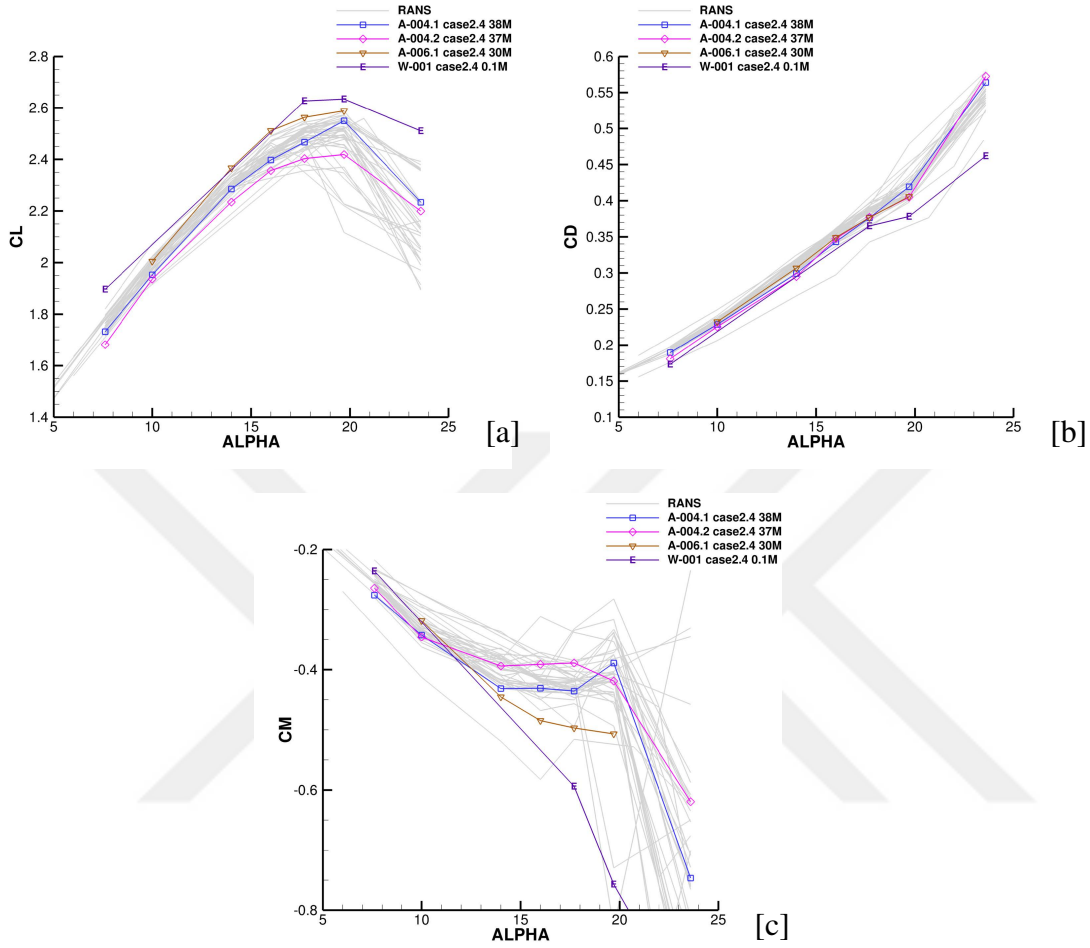


Figure 1.49 : Distribution of the lift, drag, and moment coefficients for Case 2.4 [11].

• *Case 3 - Reynolds Number Study*

Test Case 3 employs the NASA 5.2% model geometry in the standard LDG configuration, as shown in Figure 1.50. This setup includes nominal inboard/outboard trailing edge (TE) flap deflections of $40°/37°$, nominal $30°/30°$ inboard/outboard leading-edge (LE) slat settings, along with the inclusion of the nacelle, pylon, nacelle chine, LE brackets, and TE support fairings, while omitting the landing gear, vertical tail, and horizontal tail. Further details about the flow conditions can be found in Table 1.13. No results were obtained for this case by the committee members during the workshop.

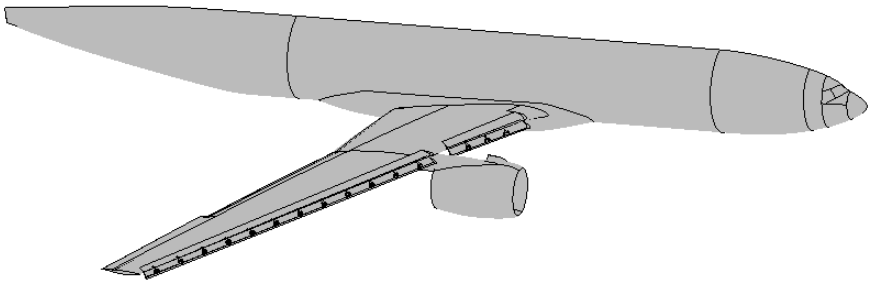


Figure 1.50 : NASA 5.2% model geometry in the standard LDG configuration

Table 1.13 : Flow conditions for Test Case 3

Mach Number	0.2
Re Number	1.05×10^6 (subcase 3.1) 5.49×10^6 (subcase 3.2) 16.0×10^6 (subcase 3.3) 30.0×10^6 (subcase 3.4)
Angle of Attacks	$6 - 10^\circ$
Reference Static Temperature	$518.67^\circ R$
Reference Static Pressure	14.696 psi

2. REVIEW OF FLOW SOLVERS AND ANISOTROPIC MESH ADAPTATION LIBRARIES

In this study, a Computational Fluid Dynamics (CFD) method based on the Reynolds-Averaged Navier-Stokes (RANS) equations is used to capture a realistic flow field around high-lift bodies. Several CFD methods can be applied to convert governing equations, which are in partial differential form, into an algebraic format. These methods include the Finite Difference Method (FDM), Finite Element Method (FEM), Finite Volume Method (FVM), Spectral Element Method (SEM), and Discontinuous Galerkin Method (DGM), among others. Figure 2.1 illustrates the most common models based on Molecular and Continuum Models.

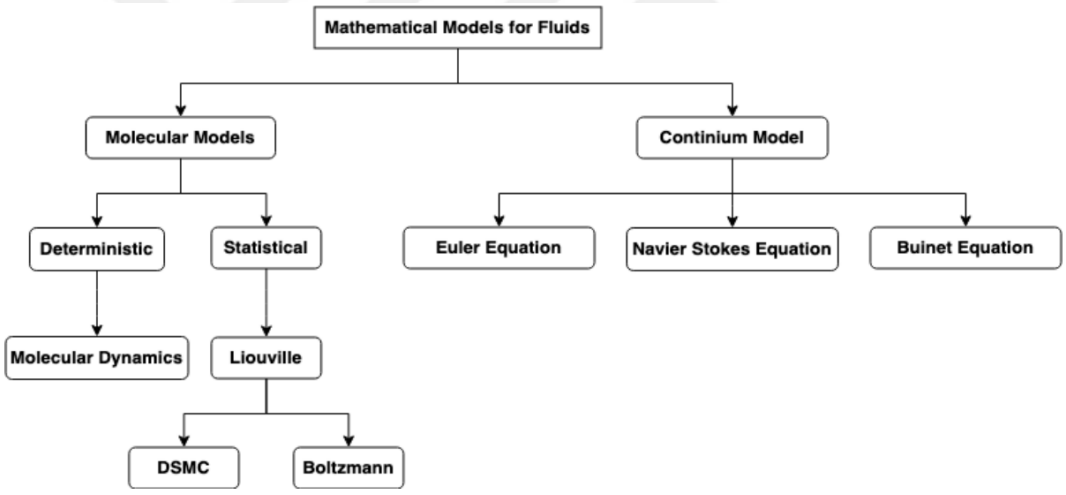


Figure 2.1 : Mathematical Model for Fluid [12]

The Knudsen number, which is the ratio of the mean free path length to the representative physical length scale, determines whether the flow should be modeled using a Molecular or Continuum Model. Figure 2.2 explains that if the Knudsen number is greater than 1, the molecular model should be applied. When the Knudsen number approaches zero, the Euler equation is used. This equation, based on inviscid theory, can be considered a governing equation but does not account for viscous effects, making it unsuitable for complex flow simulations. On the other hand, when the Knudsen number is lower than 0.1, the Navier-Stokes equations under the Continuum

Model can be used. These equations govern viscous flows and generally provide more accurate results.

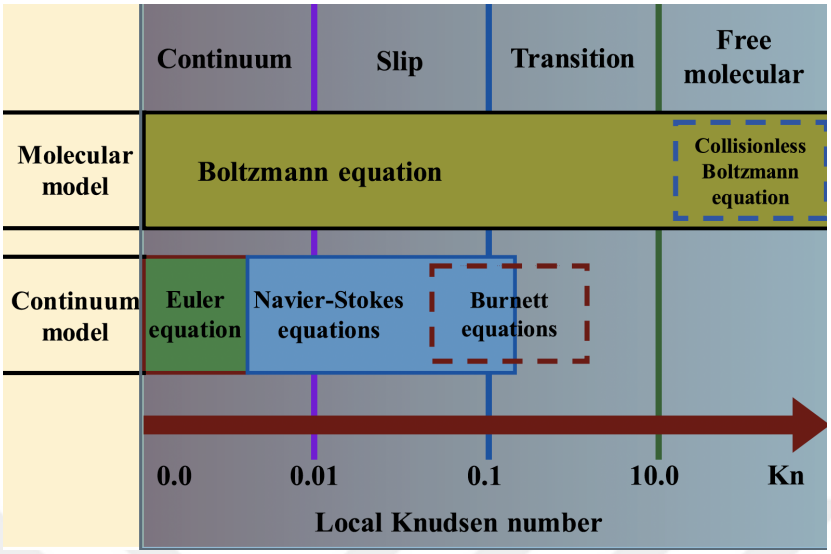


Figure 2.2 : Flow Regimes according to Kn number [13]

This study aims to obtain realistic results that align with experimental data for high-lift geometries. Therefore, the physical modeling is based on the Navier-Stokes equations within the Continuum Model, which will be explained in detail in the next chapter.

2.1 Literature Review on CFD Algorithms

In this section, several effective CFD solver algorithms available in the literature and their methodologies will be examined. The algorithms used in high-lift simulations have been selected for comparison purposes. First, the solvers FUN3D, SU2, Kestrel/COFFE, DLR TAU, and EDGE will be discussed.

2.1.1 FUN3D

FUN3D is a finite volume solver developed by researchers at NASA, USA. It is capable of solving the governing equations for various grid structures, including mixed elements, tetrahedral, pyramidal, prismatic, hexahedral, and triangular/quadrilateral grid systems [19]. The unknowns are located at the vertices or nodes of each grid element. The inviscid flux calculations are performed using a Riemann solver, and several flux schemes are available in the latest version, including Roe flux-difference

splitting, Van Leer flux-vector splitting, AUSM, and HLLC. For second-order accuracy, the unweighted least squares method is used to calculate interface values.

The algorithm includes various limiter options to improve numerical stability and accuracy. These include the Venkatakrishnan limiter, Bath limiter, and stencil-based limiters such as min-mod, van Leer, and van Albada, each augmented with a heuristic pressure limiter. Additional options include the standard min-mod, van Leer, and van Albada limiters, with an option that enables Green-Gauss gradients for inviscid reconstruction.

Viscous fluxes are computed using a finite volume approach, where velocity gradients on the dual faces are calculated with the Green-Gauss method for tetrahedral grids. Due to odd-even decoupling issues in the code, the Green-Gauss method has been updated for non-tetrahedral elements. The solver employs the backward Euler time-integration approach to update results at each time step.

For turbulence modeling, FUN3D provides multiple options, including the Spalart-Allmaras model, Menter k-omega SST model, Wilcox k-omega model, and Detached Eddy Simulation (DES). Additional turbulence models are also available, including specified or predicted transition models, allowing users flexibility in their simulations.

2.1.2 SU2

SU2 [29] is an open-source CFD solver developed at Stanford University. It consists of several packages designed for different purposes, including multi-physics simulations, design optimization, and fluid-structure interaction. This review focuses on its unstructured finite volume solver. SU2 is a vertex-based finite volume algorithm that supports mixed-element meshes. It incorporates both explicit and implicit time integration methods, along with central or upwind spatial integration schemes. Inviscid and viscous fluxes are computed at the midpoints of each edge. Various upwind and central schemes, such as Jameson-Schmidt-Turkel (JST), Roe, AUSM, HLLC, and Roe-Turkel, are available for inviscid flux calculations, with multiple limiter options for upwind schemes. Gradient calculations for viscous fluxes can be performed using either the Green-Gauss method or the weighted least-squares method. SU2 provides

turbulence modeling capabilities with both the Spalart-Allmaras (SA) and Menter's Shear-Stress Transport (SST) models.

2.1.3 Kestrel/COFFE

COFFE (Conservative Field Finite Element) is a CFD solver that is a component of Kestrel, developed under the Department of Defense (DoD) High-Performance Computing Modernization Program (HPCMP) CREATE-AV initiative [30]. The solver employs the Streamline Upwind/Petrov-Galerkin (SUPG) finite-element method to discretize partial differential equations and is suitable for both steady and unsteady simulations. It utilizes the negative variant of the Spalart-Allmaras (SA-neg) turbulence model. Several linear solvers are included, such as Generalized Minimum Residual (GMRES), Gauss-Seidel, and Point Jacobi. As a preconditioner, incomplete LU decomposition is available for GMRES, along with an unstructured linear Gauss-Seidel solver applicable to both preconditioning and solving. To enhance convergence efficiency for implicit solutions, a quasi-Newton method is employed for pseudo-transient residual calculations.

2.1.4 DLR TAU

The German Aerospace Center (DLR) TAU [31] algorithm is a comprehensive system designed to analyze flow physics around complex geometries. It is suitable for both viscous and inviscid flow simulations across all flow regimes, from subsonic to hypersonic. The solver employs a vertex-based scheme with an edge-based data structure and supports both hybrid unstructured and block-structured grid systems. For time integration, it uses an explicit Runge-Kutta algorithm, with an additional LU-SGS implicit approximate factorization option. A dual time-stepping approach is available for time-sensitive cases. Viscous fluxes are computed using a second-order central scheme, and the code offers a low Mach number preconditioning option for solving the Navier-Stokes equations.

DLR TAU supports grid adaptation on hybrid meshes using local grid refinement and wall-normal mesh movement within the semi-structured near-wall layers. The solver can also re-refine previously refined elements. The turbulence modeling options include Spalart-Allmaras with Edwards modification and several two-equation

turbulence models, such as $k-\omega$, Menter SST, Wilcox, and Kok-TNT. Recently, a model-specific "universal" wall function was integrated into the solver, improving accuracy while reducing simulation time by approximately 75% and memory usage by 40%. Additionally, the algorithm offers Detached Eddy Simulation (DES) and Extra-Large Eddy Simulation (XLES) capabilities. Given the importance of transition modeling, especially at high Mach numbers, DLR TAU includes multiple transition prediction approaches.

2.1.5 EDGE

EDGE [32] is a vertex-based finite volume solver developed by the Swedish Defence Research Agency (FOI). It operates on unstructured grid systems and supports various grid types, including hexahedral, triangular prism, and tetrahedral elements for 3D simulations, as well as quadrilateral and triangular grids for 2D cases. The solver employs second-order spatial discretization, and time integration is performed using the fourth-order Runge-Kutta method.

2.2 Review of Anisotropic Mesh Adaptation Libraries

Anisotropic mesh adaptation plays a crucial role in computational fluid dynamics (CFD) and finite element analysis (FEA) by dynamically refining and coarsening mesh elements to optimize computational accuracy and efficiency. Unlike isotropic approaches, which adjust mesh resolution uniformly in all directions, anisotropic adaptation tailors element sizes and shapes according to local flow features, such as shocks, boundary layers, or vortices. This results in improved solution fidelity while reducing the overall computational cost. The adaptation process is typically guided by a metric field that encodes local anisotropy and stretching ratios, ensuring that the mesh aligns with the dominant flow structures while maintaining numerical stability.

Various anisotropic mesh adaptation libraries have been developed to address different computational challenges, ranging from industrial aerodynamics to academic research. These libraries employ distinct algorithms, strategies, and numerical techniques to achieve optimal mesh refinement. Some focus on edge-based operations such as insertion, collapse, and swapping, while others incorporate higher-order methods to reconstruct gradients and Hessians for improved adaptation accuracy. Additionally,

many modern tools integrate parallel processing capabilities, enabling them to efficiently handle large-scale simulations. The following sections provide an overview of three widely used anisotropic mesh adaptation libraries.

2.2.1 EPIC

EPIC (Edge Primitive Insertion and Collapse) is an algorithm developed at Boeing [33] for anisotropic mesh adaptation that can be integrated with external flow solvers. The algorithm updates mesh elements through a sequence of operations including edge break, edge collapse, element reconnection, and vertex movement to progressively refine the grid. EPIC implements three types of mesh operations: EPIC-IC (insertion and collapse), EPIC-ICS (insertion, collapse, and element-face connectivity swapping), and EPIC-ICSM (insertion, collapse, swapping, and node movement). Studies have shown that connectivity swapping significantly improves grid quality at a relatively low computational cost, while node movement, although beneficial, is the most computationally expensive option. The edge length is calculated through numerical integration of the metric field along an edge [34] for instance, using an expression such as $L_e = \int_0^1 \sqrt{\mathbf{v}^T \mathbf{M}(\mathbf{x}(s))\mathbf{v}}\, ds$, where $\mathbf{v}$ represents the edge direction and $\mathbf{M}(\mathbf{x}(s))$ is the metric tensor along the edge ensuring higher accuracy in the measurement of mesh anisotropy. The metric field from the initial mesh is used to interpolate that of the adapted mesh, which can be prepared using methods that limit minimum and maximum local metric sizes, control stretching rates and anisotropy, and ensure smooth transitions in the metric distribution; surface curvature information from the original geometry or the initial mesh can also be incorporated to further refine this distribution. EPIC preserves the original model through geometric projection and local remeshing techniques, and a linear-elastic mesh deformation procedure is available to project the adapted mesh onto the geometry surface, maintaining the fidelity of the original design.

2.2.2 REFINE

REFINE is an algorithm developed at NASA [35] that serves as an open-source, metric field-based mesh adaptation tool, primarily designed for triangular and tetrahedral meshes. The algorithm employs edge splitting and collapse methods to iteratively

generate adapted meshes, focusing computational effort where it is most needed. In this process, the location of each node is determined based on its adjacent elements, and these individual contributions are combined to update the new node positions, thereby improving the shape and quality of elements in accordance with the anisotropic metric. REFINE computes mesh adaptation metrics by reconstructing gradients and Hessians from a given field, enabling the algorithm to accurately capture solution behavior and local curvature, which drives the anisotropic refinement process. Additionally, the tool has the capability to interpolate solutions between meshes, ensuring continuity and accuracy during adaptation. It supports interfaces for working with multiple geometry sources or surrogate geometry representations, enhancing its versatility in handling complex geometries. Furthermore, REFINE is designed for parallel execution, a feature that significantly enhances computational efficiency, making it well-suited for large-scale simulations. Its open-source nature encourages collaboration and ongoing development within the scientific community, ensuring that the tool remains adaptable to evolving computational needs and simulation frameworks.

In addition to the metric based adaptation, the REFINE algorithm also supports the goal oriented anisotropic refinement. This is primarily achieved through adjoint-based error estimation which is a process that calculates the sensitivity of a quantity of interest (QoI), such as lift or drag, to numerical errors in the simulation. This adjoint solution provides error indicators that highlight regions in the domain where errors have the most significant impact on the QoI, enabling the refinement of these areas to improve overall accuracy. For instance, in a supersonic flow over a blunt body, regions around the shock wave and boundary layer are identified and refined by considering the sharp gradients. NASA-REFINE employs metric tensor-based adaptation, which is another key aspect of its methodology. The metric tensor specifies the desired size, shape, and orientation of mesh elements throughout the domain. This allows for anisotropic mesh refinement, where elements are stretched or compressed to align with directional flow gradients. This technique is particularly useful in capturing vortices or wakes. In order to get accurate solutions the large and uniform meshes will require excessive computational cost, however the adaptation technique sets the mesh density dynamically so this reduces the computational cost.

2.2.3 pyAMG

pyAMG is a metric-based anisotropic mesh adaptation algorithm developed at INRIA [36] that operates within a Riemannian metric space using a multiscale approach to compute edge lengths and element volumes. By directly incorporating the anisotropic metric into the adaptation process, pyAMG supports standard mesh operations such as point insertion and deletion, edge swapping and collapse, and other local mesh adjustments as shown in the Figure 2.3. All these are implemented using a single-cavity-based operator for efficiency and consistency. This approach ensures that the adaptation process aligns the mesh elements with the underlying flow physics while maintaining the desired anisotropic resolution.

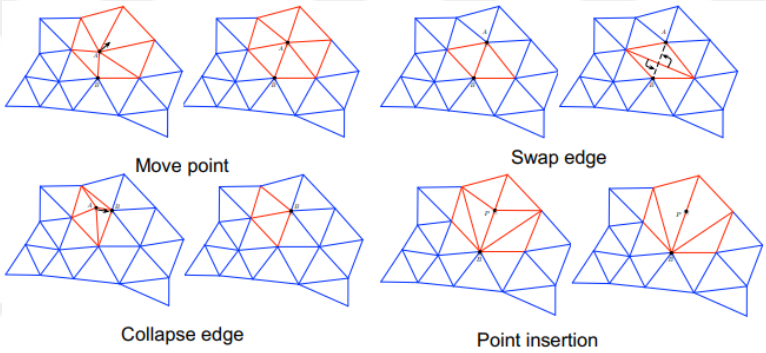


Figure 2.3 : Standard Mesh Operations [14]

The mesh adaptation process composed of in two main steps. In the first step, the algorithm generates a unit mesh by evaluating the edge lengths in the metric space. If the edges are too long they are being splited into smaller segments. On the otherhand if these are too short then they are collapsed to form longer edges, thereby aligning the mesh with the desired metric specifications.This step ensures that the mesh adapts to the anisotropic properties defined by the metric field. The approximate edge length "$\overrightarrow{ab}$" of an element in the metric space is determined using the following integral:

$$l_M(\overrightarrow{ab}) = \int_0^1 \sqrt{\overrightarrow{ab}^\top M(\overrightarrow{a} + t\overrightarrow{ab}) \quad \overrightarrow{ab}} \quad dt \tag{2.1}$$

where $M(\overrightarrow{x})$ is a metric tensor, which is a symmetric positive definite 3×3 matrix in three dimensions. This formulation allows pyAMG to transform an unstructured uniform mesh into an anisotropic one by ensuring that all edges have an average unit length in the metric space. It also enforces the alignment of edges with the local eigenvectors of the metric tensor, generating highly structured anisotropic meshes.

In the second step, the unit mesh is further optimized by inserting additional points to smooth the mesh and performing edge swaps to enhance element quality. This optimization helps prevent numerical errors due to poorly shaped elements and improves overall solution accuracy. The metric tensor formulation plays a crucial role in guiding these operations, enforcing mesh alignment with flow gradients and anisotropic structures in the domain.

This comprehensive, metric-driven approach enables pyAMG to effectively handle complex geometries and capture fine-scale features, making it a robust tool for various simulation applications. The adaptation method minimizes interpolation errors in the L_p norm, ensuring that solution accuracy is preserved while reducing computational overhead.

One of the defining aspects of pyAMG is its feature-based adaptation strategy, where the refinement process is guided by a sensor function derived from key flow variables. This sensor function, which can be based on Mach number, pressure, entropy, Finite-Time Lyapunov Exponent (FTLE), or other flow properties, identifies how mesh elements are redistributed to resolve critical flow features.

Moreover, pyAMG's capability to operate within a Riemannian metric space allows it to maintain a consistent level of accuracy across highly anisotropic flow regions. The metric field can be computed from the Hessian of a scalar flow variable, scaled by the determinant of the Hessian to control adaptation in regions with rapid changes. This method enables automatic refinement of areas requiring higher resolution while coarsening the mesh in less critical regions, optimizing the computational effort.

By enforcing local edge alignment with the principal directions of the metric tensor, pyAMG ensures that the mesh elements are naturally oriented along the most significant flow gradients. This leads to improved stability and accuracy in numerical simulations, particularly for high Reynolds number flows and complex aerodynamic configurations. The anisotropic cavity-based operators used in pyAMG further refine mesh quality by preserving local smoothness and preventing excessive element distribution during adaptation cycles. However, it should be noted that it is not possible to run publicly available version of the pyAMG in parallel.

3. MATHEMATICAL MODELING

In order to explain the mathematical approach used in the HEMLAB code, the fundamental equations must first be presented. The set of equations provided below represents the conservative form of the governing equations for fluid dynamics, combines continuity, momentum conservation in the x, y, and z directions, and energy conservation. The state vector Q contains the unknown variables, including the density ρ, the velocity components ρu, ρv, and ρw, and the total energy E. These governing equations are expressed in three-dimensional integral form, where the time derivative of the volume integral of the state vector is balanced by the surface integrals of the inviscid flux $\mathbf{F}_i$ and the viscous flux $\mathbf{F}_v$. The inviscid flux $\mathbf{F}_i$ is presented in component form, with each element corresponding to the flux in the x, y, and z directions. The terms involve the density ρ, the velocity components u, v, w, the pressure p, and the total energy E, while the dot notation $(\dot{x},\dot{y},\dot{z})$ represents the grid velocity. Similarly, the viscous flux $\mathbf{F}_v$ is provided in component form, incorporating the components of the stress tensor τ and the heat flux q.

$$\frac{\partial}{\partial t}\iiint_{\Omega}\mathbf{Q}\,dV+\oiint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F_i}\,dS-\oiint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F_v}\,dS=0 \tag{3.1}$$

$$\mathbf{Q}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{bmatrix} \tag{3.2}$$

$$\mathbf{F}_i=\begin{Bmatrix}\rho(u-\dot{x})\\ \rho u(u-\dot{x})+p\\ \rho v(u-\dot{x})\\ \rho w(u-\dot{x})\\ (E(u-\dot{x})+pu\end{Bmatrix}\mathbf{i}+\begin{Bmatrix}\rho(v-\dot{y})\\ \rho u(v-\dot{y})\\ \rho v(v-\dot{y})+p\\ \rho w(v-\dot{y})\\ (E(v-\dot{y})+pv\end{Bmatrix}\mathbf{j}+\begin{Bmatrix}\rho(w-\dot{z})\\ \rho u(w-\dot{z})\\ \rho v(w-\dot{z})\\ \rho w(w-\dot{z})+p\\ (E(w-\dot{z})+pw\end{Bmatrix}\mathbf{k} \tag{3.3}$$

$$\mathbf{F}_v = \left\{ \begin{array}{c} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{array} \right\} \mathbf{i} + \left\{ \begin{array}{c} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{array} \right\} \mathbf{j} + \left\{ \begin{array}{c} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{array} \right\} \mathbf{k} \tag{3.4}$$

The presented set of equations in 3.5 defines the components of the stress tensor (τ_{xx}, τ_{yy}, τ_{zz}, τ_{xy}, τ_{xz}, and τ_{yz}) and the heat flux (q_x, q_y, and q_z) in the context of compressible fluid dynamics. These quantities are derived from the Navier-Stokes equations and energy conservation principles. The parameters involved include the free-stream Mach number (M_∞), the Reynolds number at infinity (Re_∞), the dynamic viscosity (μ^*), the thermal conductivity (λ^*), and the Prandtl number at infinity (Pr_∞). The stress tensor components express the shear and normal stresses in the x, y, and z directions by incorporating velocity gradients and material properties. Similarly, the heat flux components represent the rate of heat transfer in the x, y, and z directions, influenced by temperature gradients and the thermophysical properties of the fluid.

$$\tau_{xx} = \frac{M_\infty}{Re_\infty}\left[2\mu^*\frac{\partial u}{\partial x} + \lambda^*\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right] \tag{3.5}$$

$$\tau_{yy} = \frac{M_\infty}{Re_\infty}\left[2\mu^*\frac{\partial v}{\partial y} + \lambda^*\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right] \tag{3.6}$$

$$\tau_{zz} = \frac{M_\infty}{Re_\infty}\left[2\mu^*\frac{\partial w}{\partial z} + \lambda^*\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)\right] \tag{3.7}$$

$$\tau_{xy} = \frac{M_\infty}{Re_\infty}\left[\mu^*\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] \tag{3.8}$$

$$\tau_{xz} = \frac{M_\infty}{Re_\infty}\left[\mu^*\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\right] \tag{3.9}$$

$$\tau_{yz} = \frac{M_\infty}{Re_\infty}\left[\mu^*\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right] \tag{3.10}$$

$$q_x = -\frac{M_\infty}{(\gamma-1)Re_\infty Pr_\infty}\left[\mu^*\frac{\partial T}{\partial x}\right] \tag{3.11}$$

$$q_y = -\frac{M_\infty}{(\gamma-1)Re_\infty Pr_\infty}\left[\mu^*\frac{\partial T}{\partial y}\right] \tag{3.12}$$

$$q_z = -\frac{M_\infty}{(\gamma-1)Re_\infty Pr_\infty}\left[\mu^*\frac{\partial T}{\partial z}\right] \tag{3.13}$$

where

$$M_\infty = \frac{U_\infty}{a} = \frac{U_\infty}{\sqrt{\gamma RT}} \tag{3.14}$$

$$Re_\infty = \frac{\rho U_\infty l}{\mu} = \frac{U_\infty l}{\nu} \tag{3.15}$$

$$Pr_\infty = \frac{C_p \mu}{k} = \frac{\mu}{(\gamma-1)k} \tag{3.16}$$

$$p = (\gamma-1)\left[E - \rho\frac{u^2+v^2+w^2}{2}\right] \tag{3.17}$$

$$T = \frac{p}{R\rho} = \frac{(\gamma-1)}{R\rho}\left[E - \rho\frac{u^2+v^2+w^2}{2}\right] \tag{3.18}$$

$$\mu^* = \frac{\mu}{\mu_\infty} = T\sqrt{T}\frac{1+S/T_\infty}{T+S/T_\infty} \tag{3.19}$$

$$\lambda^* = -\frac{2\mu^*}{3} \tag{3.20}$$

The equations given above represent a fundamental set of aerodynamic and thermodynamic relations for compressible flows. In Equation 3.14, M_∞ represents the free-stream Mach number, defined as the ratio of the free-stream velocity U_∞ to the local speed of sound a, where a is given by the square root of the product of the specific heat ratio (γ, the gas constant (R), and the temperature (T). Equation 3.15 introduces the Reynolds number at infinity (Re_∞), which measures the relative significance of inertial forces compared to viscous forces. Here, ρ is the density, U_∞ is the free-stream velocity, l is a characteristic length, μ is the dynamic viscosity, and ν is the kinematic viscosity. Moving on, Equation 3.16 defines the Prandtl number at infinity (Pr_∞), a dimensionless parameter characterizing the ratio of momentum diffusivity to thermal diffusivity. In this context, C_p denotes the specific heat at constant pressure, μ is the dynamic viscosity, and k represents the thermal conductivity. Equations 3.17 and 3.18 describe the relationships between pressure (p), temperature (T), and specific internal energy (E). The pressure is expressed in terms of the specific gas constant (R), density (ρ), and the velocity components u, v, and w. Equation 3.19 defines the non-dimensional temperature dependent laminar viscosity which can be calculated with the Sutherland's law. The parameter given with S defines the Sutherland constant, $S = 110.4^\circ K$. The temperature is then determined as a function of pressure, density, and internal energy. These equations provide a comprehensive framework for understanding the thermodynamic and aerodynamic aspects of compressible flows,

where the involved parameters play crucial roles in characterizing the behavior of the fluid.

3.1 HEMLAB Algorithm

The HEMLAB algorithm [37, 38] is a vertex-based unstructured finite volume approach designed for hybrid meshes, supporting both triangular and quadrilateral elements in 2D as well as hexahedral, prismatic, tetrahedral, and pyramidal elements in 3D. The fluxes required for this method are evaluated over the edges of these hybrid unstructured meshes. This algorithm uses the first-order Euler implicit method in time for the implicit calculations. For a vertex-based finite volume formulation, the residual vector can be written as:

$$\mathbf{R}(\mathbf{Q}) = \frac{\partial}{\partial t}\iiint_{\Omega} \mathbf{Q}\, dV + \oiint_{\partial\Omega} \mathbf{n}\cdot\mathbf{F_i}\, dS - \oiint_{\partial\Omega} \mathbf{n}\cdot\mathbf{F_v}\, dS = 0, \tag{3.21}$$

and the residual vector is required to be zero at time level $n+1$. Using the backward Euler for time discretization, the first term becomes:

$$\frac{\partial Q}{\partial t} \approx \frac{Q^{n+1} - Q^n}{\Delta t} \tag{3.22}$$

In order to resolve the Q^{n+1}, Newton's method is used. Here m denotes the m^{th} Newton iteration.

$$\mathbf{Q}^{m+1} = \mathbf{Q}^m + \Delta\mathbf{Q}^{m+1} \tag{3.23}$$

Expanding $\mathbf{R}(\mathbf{Q}^{m+1})$ using a first-order Taylor series:

$$\mathbf{R}(\mathbf{Q}^{m+1}) \approx \mathbf{R}(\mathbf{Q}^m) + \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\Delta\mathbf{Q}^{m+1} \tag{3.24}$$

Since R is the residual vector and it should go to zero at the mth iteration $\mathbf{R}(\mathbf{Q}^{m+1}) = 0$.

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\Delta\mathbf{Q}^{m+1} = -\mathbf{R}(\mathbf{Q}^m) \tag{3.25}$$

The Jacobian matrix $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}$ is obtained by differentiating $\mathbf{R}(\mathbf{Q})$. The time derivative contribution comes from the first term, which is $\frac{1}{\Delta t}\iiint_{\Omega} Q\, dV$. The following expression can be obtained by taking its derivative with respect to $\mathbf{Q}$:

$$\frac{\partial}{\partial Q}\left(\frac{M}{\Delta t}Q\right) = \frac{M}{\Delta t} \tag{3.26}$$

where M is the mass matrix. The remaining terms involve fluxes:

$$\int_{\partial\Omega} \mathbf{n}\cdot\mathbf{F}i(Q)\,dS - \int_{\partial\Omega} \mathbf{n}\cdot\mathbf{F}_v(Q)\,dS \tag{3.27}$$

Since $\mathbf{F}_i(Q)$ and $\mathbf{F}_v(Q)$ are nonlinear functions of Q and their Jacobian is:

$$J = \frac{\partial}{\partial Q}\left(\sum \mathbf{n}\cdot\mathbf{F}_i(Q)S - \sum \mathbf{n}\cdot\mathbf{F}_v(Q)S\right) \tag{3.28}$$

After combining both contributions, the full Jacobian matrix becomes:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} = \frac{M}{\Delta t} + J \tag{3.29}$$

where $\frac{M}{\Delta t}$ comes from the time derivative discretization, and J accounts for the Jacobian of the fluxes.

$$\begin{aligned} \frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\Delta\mathbf{Q}^{m+1} &= \left[\frac{M}{\Delta t} + J\right]\Delta\mathbf{Q}^{m+1} = -\mathbf{R}(\mathbf{Q}^m) \\ &= -\left[M\frac{\mathbf{Q}^m - \mathbf{Q}^n}{\Delta t}V + \sum \mathbf{n}\cdot\mathbf{F_i}(\mathbf{Q}^m)S - \sum \mathbf{n}\cdot\mathbf{F_v}(\mathbf{Q}^m)S\right] \end{aligned} \tag{3.30}$$

The convergence of the Newton method is highly sensitive to the accuracy of the Jacobian matrix evaluation, so special care is taken to compute these Jacobian matrices exactly.

Next, the algorithm addresses the computation of both inviscid and viscous fluxes. For the inviscid fluxes, the Roe scheme [39], HLLC [40], and AUSM$^+$-up [41] are implemented. The required left and right state vectors are determined through unweighted first-, second-, or third-order upwind least squares interpolations [42]. The inviscid flux Jacobian matrices are computed precisely using source code transformations provided by the Tepaneda library [43, 44].

For viscous fluxes, gradients of the primitive variables are computed using the Green-Gauss theorem at edge midpoints, rather than using simple averages, to prevent odd-even decoupling.

Additionally, various versions of the Spalart-Allmaras turbulence model [45] are available within the algorithm, with wall distance calculations implemented using the Alternating Digital Tree (ADT) algorithm [46] for anisotropic meshes. The resulting system, which comprises the compressible Navier-Stokes equations coupled with a one-equation Spalart-Allmaras turbulence model, is solved using a monolithic

approach with the restricted additive Schwarz method and the flexible generalized conjugate residual algorithm within the PETSc library [47].

Further, the construction of the Jacobian matrix can be performed using either first- or second-order upwind least squares, although the latter requires more memory. The Jacobian matrix-vector multiplication is executed using a Jacobian-free approach with Fréchet derivatives, due to the unavailability of analytical contributions related to limiters and the CFL number.

The nonlinear Newton method is implemented using PETSc's Scalable Nonlinear Equations Solvers (SNES) with a line search technique, supporting both Jacobian-free finite differences and direct approaches for the Jacobian-vector product. Although hand-coded Jacobian programming is time-consuming, it ensures computational efficiency, and any implementation errors are minimized by comparing the hand-coded Jacobian with the finite difference approach available in PETSc. With this way simulations can be done with high CFL numbers and achieving machine precision in pseudo-transient continuation (PTC) is possible since the Jacobians are calculated more accurately. In addition, the algorithm explores adaptive time-stepping (ATS). Although the ATS algorithm was attempted, it did not significantly increase time steps in 3D—despite yielding better convergence in 2D as the solution approached a steady state. However, combining ATS with the PTC algorithm may enhance robustness in 3D, as noted in [48].

The computational domain is decomposed into sub-domains using the METIS library [49] for balanced domain decomposition, with an ILU(k) preconditioner employed for the restricted additive Schwarz method. The default Venkatakrishnan-Wang limiter [50] is deactivated due to adaptive mesh refinement.

Finally, the numerical method relies on a quad-edge data structure [51] and a half-edge data structure [52] to enhance cache efficiency. Although the method is capable of handling arbitrary polyhedra in 3D, it is currently applied only to tetrahedra, hexahedra, wedges, and pyramids.

In Sections 3.1.1 and 3.1.2, the Least Squares method used for the inviscid fluxes and the Green-Gauss approach used for the viscous fluxes are explained, while the inviscid and viscous Jacobians are discussed in Sections 3.1.3 and 3.1.4.

3.1.1 Least squares method

In order to achieve second-order spatial accuracy for the inviscid flux, it is necessary to obtain accurate gradients of the state variables. One recommended approach is to employ a Taylor series approximation to evaluate the inviscid flux. Using this method, accurate approximations of the state variables at the faces of the control volume are obtained—even if the gradients at the nodes are not perfectly computed. Since the convective flux evaluation relies solely on the state variables at the control volume faces, obtaining a better approximation for these face values is crucial for minimizing errors [53].

An accurate approximation of the face-center values can be achieved by using the gradient information at a node to obtain second-order spatial accuracy. The values at neighboring nodes are approximated using a Taylor series expansion. In order to solve this system of equations with the least squares method, the number of equations must exceed the number of unknowns. Generally, there are more neighboring points than unknowns for both 2D and 3D mesh designs. Additionally, it is recommended to have at least three neighbors for a point in 2D to attain a second-order accurate reconstruction [54].

Although the resulting overdetermined system can be solved by multiplying both sides by the transpose of the matrix, this approach may lead to errors due to round-off. An analytical solution for this problem has been provided in [53], [55], and the procedure is outlined below.

The system of equations, represented in matrix form, is designed to express the spatial gradient terms in terms of the differences between the values at a reference node x_i and the values at its neighboring nodes $x_1, x_2, \ldots, x_n$ for a given variable q. The matrix $\mathbf{A}_i$ is constructed from the differences between the coordinates of the neighboring nodes and the reference node in each dimension. This system can be expressed as

$$\mathbf{A}_i \mathbf{x} = \mathbf{b}_i,$$

where $\mathbf{x}$ is a column vector containing the partial derivatives of q with respect to x, y, and z, and $\mathbf{b}_i$ is a column vector representing the differences in the values of q between the neighboring nodes and the reference node.

The matrix system is explicitly given by

$$\begin{bmatrix} \mathbf{x}_1 - \mathbf{x}_i \\ \mathbf{x}_2 - \mathbf{x}_i \\ \mathbf{x}_3 - \mathbf{x}_i \\ \mathbf{x}_4 - \mathbf{x}_i \\ \vdots \\ \mathbf{x}_n - \mathbf{x}_i \end{bmatrix} \begin{bmatrix} \frac{\partial Q}{\partial x} \\ \frac{\partial Q}{\partial y} \\ \frac{\partial Q}{\partial z} \end{bmatrix} = \begin{bmatrix} Q_1 - Q_i \\ Q_2 - Q_i \\ Q_3 - Q_i \\ Q_4 - Q_i \\ \vdots \\ Q_n - Q_i \end{bmatrix}. \tag{3.31}$$

The gradient terms at the reference node can then be calculated using the following expressions:

$$\left.\frac{\partial Q}{\partial x}\right|_i = \sum_{k=1}^{N_i} W_{ik}^x (Q_k - Q_i), \tag{3.32}$$

$$\left.\frac{\partial Q}{\partial y}\right|_i = \sum_{k=1}^{N_i} W_{ik}^y (Q_k - Q_i), \tag{3.33}$$

$$\left.\frac{\partial Q}{\partial z}\right|_i = \sum_{k=1}^{N_i} W_{ik}^z (Q_k - Q_i), \tag{3.34}$$

where N_i is the number of first-level neighbors around node i, and Q_i and Q_k are the state variable values at the reference node and its neighbors, respectively. Because the geometric coefficients remain constant as long as the mesh remains unchanged, the corresponding weights are constant. Consequently, they are computed once at the outset and stored for each outgoing edges. However, since the state variables at the nodes change at each timestep, the gradient values must be updated before each flux calculation.

The weighting coefficients are initially stored on the outgoing edges and then used for the fast evaluation of nodal gradients, which is exact for linear functions. With these gradients available, the left and right state vectors can be evaluated as

$$\mathbf{Q}_L = \mathbf{Q}_i + \phi_i \sum_{k=1}^{N_i} W_{ik} (\mathbf{Q}_k - \mathbf{Q}_j), \tag{3.35}$$

$$\mathbf{Q}_R = \mathbf{Q}_j + \phi_j \sum_{k=1}^{N_j} W_{jk} (\mathbf{Q}_k - \mathbf{Q}_j) \tag{3.36}$$

where

$$W_{ik} = \frac{1}{2}\left[W_{ik}^x (x_j - x_i) + W_{ik}^y (y_j - y_i) + W_{ik}^z (z_j - z_i)\right] \tag{3.37}$$

$$W_{jk} = \frac{1}{2}\left[W_{jk}^x (x_i - x_j) + W_{jk}^y (y_i - y_j) + W_{jk}^z (z_i - z_j)\right] \tag{3.38}$$

An analytic formulation for the least squares problem in 2D geometry is given in [53]:

$$W_{ij}^{x} = \frac{\Delta x_{ij}}{r_{11}^2} - \frac{r_{12}}{r_{11}\, r_{22}^2}\left[\Delta y_{ij} - \Delta x_{ij}\frac{r_{12}}{r_{11}}\right], \tag{3.39}$$

$$W_{ij}^{y} = \frac{1}{r_{22}^2}\left[\Delta y_{ij} - \Delta x_{ij}\frac{r_{12}}{r_{11}}\right], \tag{3.40}$$

$$r_{11} = \sqrt{\sum_{j=1}^{N_A} (\Delta x_{ij})^2}, \tag{3.41}$$

$$r_{12} = \frac{1}{r_{11}} \sum_{j=1}^{N_A} \Delta x_{ij} \Delta y_{ij} \tag{3.42}$$

$$r_{22} = \sqrt{\sum_{j=1}^{N_A} \left[\Delta y_{ij} - \Delta x_{ij}\frac{r_{12}}{r_{11}}\right]^2}. \tag{3.43}$$

These equations define the weight coefficients W_{ij}^{x} and W_{ij}^{y} for calculating spatial gradients. The weights are determined by the distances and relative positions between a reference node and its neighboring nodes. In W_{ij}^{x}, the x coordinate differences Δx_{ij} are incorporated, while W_{ij}^{y} involves both Δx_{ij} and Δy_{ij}. The parameters r_{11}, r_{12}, and r_{22} represent, respectively, the Euclidean norm of the x coordinate differences, the dot product of Δx_{ij} and Δy_{ij} normalized by r_{11}, and the adjusted Euclidean norm of the y coordinate differences.

An analytic formulation for the least squares problem in 3D geometry is presented in [55]. This set of equations describes the calculation of the weight coefficients W_{ij}^{x}, W_{ij}^{y}, and W_{ij}^{z} for spatial gradient calculations in three dimensions. The expressions for $\alpha_{ij,1}$, $\alpha_{ij,2}$, and $\alpha_{ij,3}$ involve combinations of the coordinate differences Δx_{ij}, Δy_{ij}, and Δz_{ij} along with the Euclidean norms r_{11}, r_{22}, and r_{33}. The parameter β is introduced to capture additional geometric relationships between the nodes.

Similar to the 2D case, the weight coefficients in 3D are determined as linear combinations of $\alpha_{ij,1}$, $\alpha_{ij,2}$, and $\alpha_{ij,3}$:

$$W_{ij}^{x} = \alpha_{ij,1} - \frac{r_{12}}{r_{11}}\,\alpha_{ij,2} + \beta\,\alpha_{ij,3}, \tag{3.44}$$

$$W_{ij}^{y} = \alpha_{ij,2} - \frac{r_{33}}{r_{22}}\,\alpha_{ij,3}, \tag{3.45}$$

$$W_{ij}^{z} = \alpha_{ij,3}, \tag{3.46}$$

where

$$\alpha_{ij,1} = \frac{\Delta x_{ij}}{r_{11}^2}, \tag{3.47}$$

$$\alpha_{ij,2} = \frac{1}{r_{22}^2}\left(\Delta y_{ij} - \frac{r_{12}}{r_{11}}\Delta x_{ij}\right), \tag{3.48}$$

$$\alpha_{ij,3} = \frac{1}{r_{33}^2}\left(\Delta z_{ij} - \frac{r_{23}}{r_{22}}\Delta y_{ij} + \beta\,\Delta x_{ij}\right), \tag{3.49}$$

$$\beta = \frac{r_{12}\,r_{23} - r_{13}\,r_{22}}{r_{11}\,r_{22}}, \tag{3.50}$$

$$r_{11} = \sqrt{\sum_{j=1}^{N_A}(\Delta x_{ij})^2}, \tag{3.51}$$

$$r_{12} = \frac{1}{r_{11}}\sum_{j=1}^{N_A}\Delta x_{ij}\,\Delta y_{ij}, \tag{3.52}$$

$$r_{22} = \sqrt{\sum_{j=1}^{N_A}(\Delta y_{ij})^2 - r_{12}^2}, \tag{3.53}$$

$$r_{13} = \frac{1}{r_{11}}\sum_{j=1}^{N_A}\Delta x_{ij}\,\Delta z_{ij}, \tag{3.54}$$

$$r_{23} = \frac{1}{r_{22}}\left(\sum_{j=1}^{N_A}\Delta y_{ij}\,\Delta z_{ij} - \frac{r_{12}}{r_{11}}\sum_{j=1}^{N_A}\Delta x_{ij}\,\Delta z_{ij}\right), \tag{3.55}$$

$$r_{33} = \sqrt{\sum_{j=1}^{N_A}(\Delta z_{ij})^2 - \left(r_{13}^2 + r_{23}^2\right)}. \tag{3.56}$$

The provided equations define the weight coefficients W_{ij}^x, W_{ij}^y, and W_{ij}^z used in the calculation of spatial gradients in 3D. These weights are determined by the distances and relative positions between a reference node and its neighboring nodes, ensuring that the geometric relationships between nodes are appropriately captured.

The limiter function ϕ_i is computed using the Venkatakrishnan-Wang limiter [50] as follows:

$$\Phi_i = \min(\Phi_{ik}), \tag{3.57}$$

where

$$\Phi_{ik} = \frac{1}{\Delta_-}\left[\frac{(\Delta_+^2 + \varepsilon^2)\,\Delta_- + 2\Delta_-^2\,\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_+\Delta_- + \varepsilon^2}\right], \tag{3.58}$$

and

$$\Delta_- = Q_k - Q_i, \tag{3.59}$$

$$\Delta_+ = \begin{cases} Q_{i,\max} - Q_i, & \text{if } \Delta_- \geq 0, \\ Q_{i,\min} - Q_i, & \text{if } \Delta_- < 0. \end{cases} \tag{3.60}$$

Here, $\varepsilon = K_v(Q^{\max} - Q^{\min})$, where K_v is a limiter constant, and $Q^{\min}$ and $Q^{\max}$ are the global minimum and maximum values, respectively.

The second-order inviscid Jacobian construction leads to

$$\begin{bmatrix} J_L\phi_i W_{i1} & J_L\phi_i W_{i2} & \dots & J_L(1-\phi_i\sum_{k=1}^{N_i} W_{ik}) & J_R\phi_j W_{j1} & J_R\phi_j W_{j2} & \dots & J_R(1-\phi_j\sum_{k=1}^{N_j} W_{jk}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -J_L\phi_i W_{i1} & -J_L\phi_i W_{i2} & \dots & -J_L(1-\phi_i\sum_{k=1}^{N_i} W_{ik}) & -J_R\phi_j W_{j1} & -J_R\phi_j W_{j2} & \dots & -J_R(1-\phi_j\sum_{k=1}^{N_j} W_{jk}) \end{bmatrix} \begin{bmatrix} \Delta Q_{i1} \\ \Delta Q_{i2} \\ \vdots \\ \Delta Q_i \\ \vdots \\ \Delta Q_{j1} \\ \Delta Q_{j2} \\ \vdots \\ \Delta Q_j \end{bmatrix} = - \begin{bmatrix} R_i \\ \vdots \\ -R_j \end{bmatrix} \tag{3.61}$$

where the local inviscid Jacobian matrices are defined as

$$\mathbf{J_L} = \frac{\partial\left(\mathbf{n}\cdot\mathbf{F_i}(\mathbf{Q_L},\mathbf{Q_R})\right)}{\partial \mathbf{Q}_L}, \tag{3.62}$$

$$\mathbf{J_R} = \frac{\partial\left(\mathbf{n}\cdot\mathbf{F_i}(\mathbf{Q_L},\mathbf{Q_R})\right)}{\partial \mathbf{Q}_R}. \tag{3.63}$$

3.1.2 Green-Gauss approach

Due to the presence of viscous forces, the calculation of diffusive fluxes is necessary. For RANS-based solvers, the diffusion term originates from the right-hand side of the Navier-Stokes equations. Since the diffusive fluxes are elliptic in nature, a central numerical scheme must be used. However, accurate evaluation of the velocity and temperature gradients at the control volume faces is required. Therefore, the Green-Gauss approach is employed for gradient evaluation.

To compute the viscous flux, the derivatives of the primitive variables at the edge centers must first be determined. The accuracy of the viscous fluxes is directly related to the precision of these derivative approximations. Because the Green-Gauss approach yields more accurate gradients than a Taylor series expansion, it is preferred in the HEMLAB algorithm. In this approach, the derivative calculations at the edge centers require a linear interpolation of the state variables at the control volume faces adjacent to each edge. Figure 3.1 illustrates the dual volume construction in 2D, which is created

using the midpoint of each edge of the control volume. The Green-Gauss theorem provides an alternative method for integrating over the shaded area by converting an area integration into a line integration. Consequently, to calculate the gradient at the center of this area, only linear interpolation to the edge centers of the dual volume is necessary.

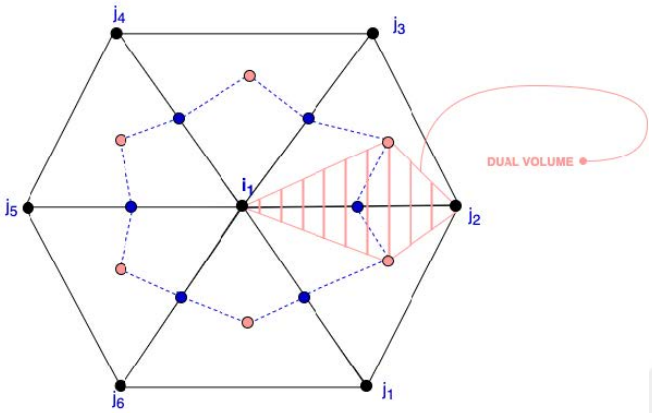


Figure 3.1 : Dual Volume Schema for 2D

Similarly, in 3D, integration is performed over the faces surrounding the volume element, reducing the overall numerical cost. Figure 3.2 shows the dual volume construction for 3D cases. In this figure, the midpoint of the purple edge is the location where the gradients must be computed. The dual control area is constructed around the purple edge using the centers of cells and faces from adjacent elements. The green edges indicate the boundaries between nodal control volumes where the code integrates the stresses; this is the same region used for calculating the viscous fluxes at those edges. Importantly, the gradient evaluation method does not cause any odd-even oscillations.

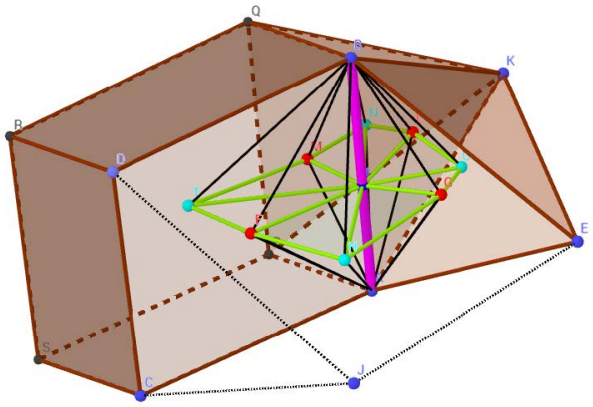


Figure 3.2 : Dual Volume Schema for 3D [15]

In a manner similar to the determination of weights in the least squares method, the Green-Gauss approach can compute and store constant weights for non-deforming meshes. This approach is particularly applied in the 3D solver, while in 2D the situation is relatively straightforward and such weights are not necessary. In cases where the mesh remains unchanged (as in many 2D scenarios), the face areas of the dual control volume surrounding the purple edge in Figure 3.2 remain constant. Initially, the computation involves linearly interpolating between the enclosing vertices to determine the face areas; these values are then stored for each edge. For the purple edge in Figure 3.2, 11 surrounding nodes contribute, involving 33 double-precision variables. Despite the increased memory usage, this method significantly accelerates the overall computation.

The gradient at the midpoint of an edge is evaluated as

$$\left.\frac{\partial \phi}{\partial x}\right|_{AB} = \sum_{i=1}^{N} W_i^x \phi_i, \tag{3.64}$$

$$\left.\frac{\partial \phi}{\partial y}\right|_{AB} = \sum_{i=1}^{N} W_i^y \phi_i, \tag{3.65}$$

$$\left.\frac{\partial \phi}{\partial z}\right|_{AB} = \sum_{i=1}^{N} W_i^z \phi_i, \tag{3.66}$$

where N represents the total number of nodes belonging to the elements sharing the active edge, ϕ is a scalar quantity at each node, and V^d denotes the dual volume constructed around the edge. The gradient at the edge midpoint is influenced by all nodes in the surrounding elements; each node contributes three components, and these contributions are computed once and stored for every edge. During gradient evaluation, the stored coefficients are multiplied by the corresponding scalar values, eliminating the need to recompute the dual volume areas at every time step, which significantly improves performance.

3.1.3 Inviscid jacobian calculation

The convective term in the Navier-Stokes equations represents the inviscid component of the flow. Because convective equations transfer information in only one direction, any mathematical discretization must account for this unidirectional nature; otherwise, the applied model may not accurately capture the physical behavior. To ensure consistency between the mathematical formulation and the underlying physics, an

upwinding approach is used. Upwinding involves applying forward or backward differentiation based on the direction of information flow. Numerous schemes exist in the literature for computing the convective part of the Navier-Stokes equations. When evaluating convective fluxes, neglecting the viscous component yields the Euler equations. The inviscid Jacobian, denoted as A_i, is then defined as follows. The mathematical expressions describe the inviscid terms in the context of fluid dynamics, particularly within the Navier-Stokes framework. Equation 3.67 presents a matrix formulation that involves various parameters related to the flow, including the flow variables and the speed of sound (a). The variable K is defined as $\gamma - 1$, where γ is the ratio of specific heats. Equation 3.69 introduces the eigenvalues (Λ) of the inviscid Jacobian matrix, which are essential for analyzing the fluid system's behavior. The matrix Λ is diagonal, with its components corresponding to different flow directions and the speed of sound. Equations 3.71 and 3.70 provide the left eigenvector matrix (P) and its inverse (P^{-1}), respectively. These matrices are instrumental in diagonalizing the Jacobian matrix and simplifying the analysis of fluid dynamics by defining the wave properties and interactions within the flow.

$$[A] = \begin{bmatrix} -V_t & n_x & n_y & n_z & 0 \\ n_x\phi - uV & V - V_t - a_3 n_x u & n_x v - a_2 n_y u & n_x w - a_2 n_z u & n_x a_1 - a_2 uV \\ n_y\phi & n_y u - a_2 n_x v & V - V_t - a_3 n_y v & n_y w - a_2 n_z v & n_y a_1 - a_2 vV \\ n_z\phi & n_z u - a_2 n_x w & n_z v - a_2 n_y w & V - V_t - a_3 n_z w & n_z a_1 - a_2 wV \\ V(\phi - a_1) & n_x a_1 - a_2 uV & n_y a_1 - a_2 vV & n_z a_1 - a_2 wV & \gamma V - V_t \end{bmatrix} \quad (3.67)$$

$$[A] = [P]\Lambda[P]^{-1} \quad (3.68)$$

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 & 0 & 0 & 0 \\ 0 & \Lambda_2 & 0 & 0 & 0 \\ 0 & 0 & \Lambda_3 & 0 & 0 \\ 0 & 0 & 0 & \Lambda_4 & 0 \\ 0 & 0 & 0 & 0 & \Lambda_5 \end{bmatrix} \quad (3.69)$$

$$\hat{P} = \begin{bmatrix} n_x & n_y & n_z & a_3 & a_3 \\ n_x u & n_y u - n_z\rho & n_z u + n_y\rho & a_3(u + n_x c) & a_3(u - n_x c) \\ n_x v + n_z\rho & n_y v & n_z v - n_x\rho & a_3(v + n_y c) & a_3(v - n_y c) \\ n_x w - n_y\rho & n_y w + n_x\rho & n_z w & a_3(w + n_z c) & a_3(w - n_z c) \\ n_x a_6 + \rho(n_z v - n_y w) & n_y a_6 + \rho(n_x w - n_z u) & n_z a_6 + \rho(n_y u - n_x v) & a_3(a_4 + cV) & a_3(a_4 - cV) \end{bmatrix} \quad (3.70)$$

$$\hat{P}^{-1} = \begin{bmatrix} n_x a_5 - \frac{(n_z v - n_y w)}{\rho} & n_y a_5 - \frac{(n_x w - n_z u)}{\rho} & n_z a_5 - \frac{(n_y u - n_x v)}{\rho} & a_2(\phi - cV) & a_2(\phi + cV) \\ \frac{n_x a_1 u}{c^2} & \frac{n_y a_1 u}{c^2} - \frac{n_z}{\rho} & \frac{n_z a_1 u}{c^2} + \frac{n_y}{\rho} & -a_2(a_1 u - n_x c) & -a_2(a_1 u + n_x c) \\ \frac{n_x a_1 v}{c^2} + \frac{n_z}{\rho} & \frac{n_y a_1 v}{c^2} & \frac{n_z a_1 v}{c^2} - \frac{n_x}{\rho} & -a_2(a_1 v - n_y c) & -a_2(a_1 v + n_y c) \\ \frac{n_x a_1 w}{c^2} - \frac{n_y}{\rho} & \frac{n_y a_1 w}{c^2} + \frac{n_x}{\rho} & \frac{n_z a_1 w}{c^2} & -a_2(a_1 w - n_z c) & -a_2(a_1 w + n_z c) \\ -\frac{n_x a_1}{c^2} & -\frac{n_y a_1}{c^2} & -\frac{n_z a_1}{c^2} & a_1 a_2 & a_1 a_2 \end{bmatrix} \quad (3.71)$$

The following auxiliary variables are defined:

$$
\begin{aligned}
a_1 &= \gamma - 1, && (3.72)\\
a_2 &= \frac{1}{\rho\, c\sqrt{2}}, && (3.73)\\
a_3 &= \frac{\rho}{c\sqrt{2}}, && (3.74)\\
a_4 &= \frac{\phi + c^2}{\gamma - 1}, && (3.75)\\
a_5 &= 1 - \frac{\phi}{c^2}, && (3.76)\\
a_6 &= \frac{\phi}{\gamma - 1}, && (3.77)\\
V &= n_x u + n_y v + n_z w, && (3.78)\\
V_t &= n_x \frac{\partial x}{\partial t} + n_y \frac{\partial y}{\partial t} + n_z \frac{\partial z}{\partial t}, && (3.79)\\
\phi &= \frac{1}{2}(\gamma - 1)(u^2 + v^2 + w^2), && (3.80)\\
\Lambda_1 &= \Lambda_2 = \Lambda_3 = V - V_t, && (3.81)\\
\Lambda_4 &= V - V_t + c, && (3.82)\\
\Lambda_5 &= V - V_t - c. && (3.83)
\end{aligned}
$$

In this thesis, the Roe scheme [39, 55] is selected as the flux scheme for solving fluid dynamics problems. The inviscid flux is expressed as a function of the left ($\mathbf{Q}_L$) and right ($\mathbf{Q}_R$) state vectors:

$$
\mathbf{F}(\mathbf{Q}_L, \mathbf{Q}_R) = \frac{1}{2}\Big[F(\mathbf{Q}_L) + F(\mathbf{Q}_R) - \hat{P}\,|\hat{\Lambda}|\,\hat{P}^{-1}(\mathbf{Q}_R - \mathbf{Q}_L)\Big]. \tag{3.84}
$$

Roe-averaged values, such as density ($\hat{\rho}$), velocity ($\hat{\mathbf{U}}$), total enthalpy ($\hat{H}$), and speed of sound ($\hat{a}$), are computed based on the left and right states:

$$
\begin{aligned}
\hat{\rho} &= \sqrt{\rho_L \rho_R}, && (3.85)\\
\hat{\mathbf{U}} &= \frac{\sqrt{\rho_L}\,\mathbf{U}_L + \sqrt{\rho_R}\,\mathbf{U}_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, && (3.86)\\
\hat{H} &= \frac{\sqrt{\rho_L}\,H_L + \sqrt{\rho_R}\,H_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, && (3.87)\\
\hat{a} &= \sqrt{(\gamma - 1)\left[\hat{H} - 0.5\,\hat{u}^2\right]}. && (3.88)
\end{aligned}
$$

During computations using the Roe flux method [39], the Harten correction [56] (Equation 3.89) is applied to modify eigenvalues below a threshold $h = \varepsilon \max(|\lambda_i|)$:

$$
\lambda_i = \frac{h^2 + \lambda_i^2}{2h}. \tag{3.89}
$$

3.1.4 Viscous jacobian calculation

The viscous fluxes in Equation 3.4 are functions of the gradients of both velocity and temperature. However, the solution vector contains only conservative variables. In the HEMLAB algorithm, the viscous Jacobian is computed by first assuming that the solution vector is based on primitive variables. The resulting Jacobian is then multiplied by the primitive-to-conservative Jacobian to obtain the correct global matrix.

For example, consider the x-momentum equation:

$$\begin{aligned}\sum_{i=1}^{N} \tau_{xx} A_x^c + \tau_{xy} A_y^c + \tau_{xz} A_z^c &= \sum_{i=1}^{N} \mu \Big[(2+\lambda) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y} + \lambda \frac{\partial w}{\partial z} \Big] A_x^c \\ &\quad + \mu \Big[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \Big] A_y^c \\ &\quad + \mu \Big[\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \Big] A_z^c. \end{aligned} \tag{3.90}$$

Similarly, the y- and z-momentum equations and the energy equation are formulated using appropriate combinations of stress components, gradients, and face areas.

In the explicit algorithm, values for u, v, w, and T are taken from the current time step, and the corresponding weights are simply multiplied by these values. In the implicit solution, where u, v, w, and T at the next time step are computed, a matrix is constructed using only the weights associated with neighboring nodes. This matrix is then incorporated into the implicit global matrix. In the implicit case, the nodes of the elements surrounding an edge contribute to the coupled system (denoted N_i, $i = 1, 2, \ldots, N$), resulting in a wider Jacobian for the viscous flux compared to the inviscid flux. The contribution of the viscous flux for each edge is calculated by solving the following matrix system:

$$\begin{bmatrix} \ldots & J_L & \ldots & J_{N1} & \ldots & J_R & \ldots & J_{N2} & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & -J_L & \ldots & -J_{N1} & \ldots & -J_R & \ldots & -J_{N2} & \ldots \end{bmatrix} \times \begin{bmatrix} \Delta Q_L \\ \vdots \\ \Delta Q_{N_1} \\ \vdots \\ \Delta Q_R \\ \vdots \\ \Delta Q_{N_2} \\ \vdots \end{bmatrix} = \begin{bmatrix} R_L \\ \vdots \\ R_R \end{bmatrix}. \tag{3.91}$$

The transformation from conservative variables $\mathbf{Q}$ to primitive variables $\mathbf{W}$ is performed via:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} = \frac{\partial \mathbf{R}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{Q}}, \tag{3.92}$$

where

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} \rho \\ u \\ v \\ w \\ T \end{bmatrix}. \tag{3.93}$$

The method increases memory usage and the time required for matrix-vector multiplication but yields a more precise evaluation of the Jacobian matrix. This precision allows for the use of larger CFL numbers and facilitates convergence in fewer iterations, in line with current trends toward achieving machine-zero convergence [57, 58]. It is important to note that while the Green-Gauss approximation yields first-order accuracy on boundaries, it is not used for constructing the Jacobian at these locations because of the imposed physical no-slip and adiabatic/isothermal boundary conditions. For aerodynamic loads and moments, a conservative approach is preferred—obtaining forces and moments by summing the control volume fluxes adjacent to solid walls [59]. The accuracy of boundary-edge gradients can also be enhanced by methods described in [60]. Strongly enforced boundary conditions impose no-slip conditions at boundary nodes, while far-field boundary conditions can be weakly imposed using either inviscid flux values or Riemann variables [61].

3.2 Turbulence Model

Turbulence modeling plays a crucial role in computational fluid dynamics (CFD), especially in aerospace applications, where accurate predictions of aerodynamic performance are essential. Among the various turbulence models, the Spalart-Allmaras (SA) model is widely used due to its robustness and efficiency in handling external aerodynamic flows, particularly around airfoils, wings, and full aircraft configurations. Originally developed for aerospace applications, the SA model is a one-equation turbulence model that provides a good balance between accuracy and computational cost. Over time, variations and enhancements of the SA model, such as the Negative

Spalart-Allmaras (SA-neg) model, have been introduced to improve stability and accuracy, particularly in regions of separated flow and adverse pressure gradients.

3.2.1 Negative Spalart-Allmaras model

This model divides the turbulence equations based on the value of the turbulent viscosity. The most common form of the SA-neg model is given below. The right-hand side of the turbulence equation consists of three parts: production, destruction, and diffusion.

$$\frac{\partial \hat{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \hat{\nu} = \text{Production} + \text{Destruction} + \text{Diffusion} \tag{3.94}$$

For the case when $\hat{\nu} \geq 0$:

$$\begin{aligned} \frac{\partial \hat{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \hat{\nu} &= c_{b1} \hat{S} \hat{\nu} \\ &- \frac{M_\infty}{Re} \left[c_{w1} f_w \right] \left(\frac{\hat{\nu}}{d} \right)^2 \\ &+ \frac{M_\infty}{Re\, \sigma} \left[\nabla \cdot \left[(\hat{\nu} f_n + \nu) \nabla \hat{\nu} \right] + c_{b2} \nabla \hat{\nu} \cdot \nabla \hat{\nu} \right] \end{aligned} \tag{3.95}$$

For $\hat{\nu} < 0$:

$$\begin{aligned} \frac{\partial \hat{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \hat{\nu} &= c_{b1} |\omega| \hat{\nu} \\ &+ \frac{M_\infty}{Re} c_{w1} \left(\frac{\hat{\nu}}{d} \right)^2 \alpha \\ &+ \frac{M_\infty}{Re\, \sigma} \left[\nabla \cdot \left[(\hat{\nu} f_n + \nu) \nabla \hat{\nu} \right] + c_{b2} \nabla \hat{\nu} \cdot \nabla \hat{\nu} \right] \end{aligned} \tag{3.96}$$

Unlike the classical Spalart-Allmaras model, the SA-neg model does not include a wall function calculation for f_w in the destruction term when $\hat{\nu}$ is negative. Additionally, the production term in the negative model does not use the $\hat{S}$ parameter; it only requires the vorticity term. The final difference between the two sets of equations is the α parameter, which is proposed to be $\alpha = 10$ as suggested in [62]. There is no difference in the diffusion term between the positive and negative models.

The turbulent viscosity is defined as

$$\mu_t^* = \rho \, \hat{\nu} \, f_{v1}. \tag{3.97}$$

https://turbmodels.larc.nasa.gov/

The $\hat{S}$ parameter, which is essential for computing the production term, can be determined based on a vorticity limit:

$$\hat{S} = \begin{cases} |\omega| + \bar{S}, & \text{if } \bar{S} \geq -c_{v2}\,|\omega|, \\ |\omega|\left(1 + \frac{c_{v2}^2\,|\omega| + c_{v3}\,\bar{S}}{(c_{v3} - 2c_{v2})\,|\omega| - \bar{S}}\right), & \text{otherwise.} \end{cases} \tag{3.98}$$

Other auxiliary parameters are defined as follows:

$$\chi = \frac{\hat{\nu}}{\nu^*} = \frac{\hat{\nu}\,\rho}{\mu_t^*}, \tag{3.99}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \tag{3.100}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi\, f_{v1}}, \tag{3.101}$$

$$\bar{S} = \frac{M_\infty}{Re}\,\frac{\hat{\nu}}{\kappa^2\, d^2}\, f_{v2}. \tag{3.102}$$

The wall function parameter f_w in the destruction term is calculated by:

$$r = \min\left[10, \frac{M_\infty}{Re_\infty}\,\frac{\hat{\nu}}{\hat{S}\,\kappa^2\, d^2}\right], \tag{3.103}$$

$$g = r + c_{w2}\,(r^6 - r), \tag{3.104}$$

$$f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}. \tag{3.105}$$

The parameter f_n, which modifies the diffusion coefficient, is given by:

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}, \tag{3.106}$$

where $f_n = 1$ in regions where the turbulent viscosity is positive, and c_{n1} is set to 16.

The remaining coefficients are specified as:

$$\begin{aligned} \sigma &= 2/3, \quad c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad \kappa = 0.41, \\ c_{w1} &= \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{v1} = 7.1, \\ c_{t3} &= 1.2, \quad c_{t4} = 0.5, \quad c_{n1} = 16, \quad c_{v2} = 0.7, \quad c_{v3} = 0.9. \end{aligned} \tag{3.107}$$

The previous version of the HEMLAB code assumed the turbulent viscosity was set to zero for negative values. This has been corrected by implementing the SA-negative model, which improves convergence of the Newton method. The basic turbulence equations are formulated in a finite volume format and solved using Newton's

method. This requires adding an infinitesimal value for each unknown—denoted by Δ—integrated over the control volume. The implicit Euler method combined with Newton's method is applied for both positive and negative $\hat{\nu}$. After computing all coefficients and terms, the production, destruction, and diffusion parts are calculated separately according to the unknowns (ν, ρ, ω) and then inserted into the appropriate locations in the Jacobian matrix.

For example, if $\hat{\nu} \geq 0$, the finite volume discretization yields:

$$
\begin{aligned}
&\iiint_{\Omega} \frac{\Delta\hat{\nu}}{\Delta t}\,dV + \oiint_{\partial\Omega} \left[(\mathbf{u}\cdot\mathbf{n})\,\Delta\hat{\nu} + \hat{\nu}\,(\mathbf{n}\cdot\Delta\mathbf{u})\right] dS - \iiint_{\Omega} \left[(\nabla\cdot\mathbf{u})\,\Delta\hat{\nu} + (\nabla\cdot\Delta\mathbf{u})\,\hat{\nu}\right] dV \\
&- \iiint_{\Omega} c_{b1}\,\Delta\hat{S}\,\hat{\nu}\,dV - \iiint_{\Omega} c_{b1}\,\hat{S}\,\Delta\hat{\nu}\,dV + \iiint_{\Omega} \frac{M_\infty}{Re\,d^2}\,c_{w1}\left[2 f_w\,\hat{\nu}\,\Delta\hat{\nu} + \hat{\nu}^2\,\Delta f_w\right] dV \\
&- \oiint_{\partial\Omega} \frac{M_\infty}{Re\,\sigma}\,\mathbf{n}\cdot\nabla\hat{\nu}\left[f_n\,\Delta\hat{\nu} + \hat{\nu}\,\Delta f_n + \Delta\nu\right] dS - \oiint_{\partial\Omega} \frac{M_\infty}{Re\,\sigma}\left[\mathbf{n}\cdot\nabla(\Delta\hat{\nu})\,(f_n\,\hat{\nu} + \nu)\right] dS \\
&- \iiint_{\Omega} \frac{M_\infty}{Re\,\sigma}\,c_{b2}\left[2\,\nabla(\Delta\hat{\nu})\cdot\nabla\hat{\nu}\right] dV \qquad (3.108) \\
&= -\oiint_{\partial\Omega} \mathbf{n}\cdot\mathbf{u}\,\hat{\nu}\,dS + \iiint_{\Omega} (\nabla\cdot\mathbf{u})\,\hat{\nu}\,dV + \iiint_{\Omega} c_{b1}\,\hat{S}\,\hat{\nu}\,dV - \iiint_{\Omega} \frac{M_\infty}{Re\,d^2}\,c_{w1}\,f_w\,\hat{\nu}^2\,dV \\
&+ \oiint_{\partial\Omega} \frac{M_\infty}{Re\,\sigma}\left[\mathbf{n}\cdot\nabla\hat{\nu}\,(f_n\,\hat{\nu} + \nu)\right] dS + \iiint_{\Omega} \frac{M_\infty}{Re\,\sigma}\,c_{b2}\,\nabla\hat{\nu}\cdot\nabla\hat{\nu}\,dV
\end{aligned}
$$

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

$$
\begin{aligned}
&\iiint_{\Omega} \frac{\Delta\hat{\nu}}{\Delta t}dV + \oiint_{\partial\Omega} \Big[(\mathbf{u}\cdot\mathbf{n})\Delta\hat{\nu}) + \hat{\nu}(\mathbf{n}\cdot\Delta\mathbf{u})\Big] dS - \iiint_{\Omega} (\nabla\cdot\mathbf{u})\Delta\hat{\nu} + (\nabla\cdot\Delta\mathbf{u})\hat{\nu} dV \\
&- \iiint_{\Omega} c_{b1}|\Delta\omega|\hat{\nu} dV - \iiint_{\Omega} c_{b1}|\omega|\Delta\hat{\nu} dV - \iiint_{\Omega} \frac{M_\infty}{Re d^2} c_{w1}[2\hat{\nu}\Delta\hat{\nu}]\alpha dV \\
&- \oiint_{\partial\Omega} \frac{M_\infty}{Re\sigma}\mathbf{n}\cdot\nabla\hat{\nu}\Big[f_n\Delta\hat{\nu} + \hat{\nu}\Delta f_n + \Delta\nu\Big] dS - \oiint_{\partial\Omega} \frac{M_\infty}{Re\sigma}\Big[\mathbf{n}\cdot\nabla(\Delta\hat{\nu})\Big(f_n\hat{\nu} + \nu\Big)\Big] dS \\
&- \iiint_{\Omega} \frac{M_\infty}{Re\sigma} c_{b2}\Big[2\nabla(\Delta\hat{\nu})\cdot\nabla\hat{\nu}\Big] dV = -\oiint_{\partial\Omega} \mathbf{n}\cdot\mathbf{u}\hat{\nu} dS \qquad (3.109) \\
&+ \iiint_{\Omega} (\nabla\cdot\mathbf{u})\hat{\nu} dV + \iiint_{\Omega} c_{b1}|\omega|\hat{\nu} dV + \iiint_{\Omega} \frac{M_\infty}{Re d^2} c_{w1}\hat{\nu}^2\alpha dV \\
&+ \oiint_{\partial\Omega} \frac{M_\infty}{Re\sigma}\Big[\mathbf{n}\cdot\nabla\hat{\nu}\Big(f_n\hat{\nu} + \nu\Big)\Big] dS + \iiint_{\Omega} \frac{M_\infty}{Re\sigma} c_{b2}\nabla\hat{\nu}\cdot\nabla\hat{\nu} dV
\end{aligned}
$$

The residual on the right-hand side represents the contributions from all Δ terms and derivatives, as expressed by:

$$\Delta\chi = \frac{\partial\chi}{\partial\rho}\Delta\rho + \frac{\partial\chi}{\partial\hat{\nu}}\Delta\hat{\nu} + \frac{\partial\chi}{\partial T}\Delta T, \tag{3.110}$$

$$\Delta f_{v1} = \frac{df_{v1}}{d\chi}\Delta\chi, \tag{3.111}$$

$$\Delta f_{v2} = \frac{df_{v2}}{d\chi}\Delta\chi, \tag{3.112}$$

$$\Delta f_n = \frac{df_n}{d\chi}\Delta\chi, \tag{3.113}$$

$$\Delta f_w = \frac{df_w}{dg}\frac{dg}{dr}\Delta r, \tag{3.114}$$

$$\Delta\bar{S} = \frac{M_\infty}{Re\,\kappa^2 d^2}\left[\hat{\nu}\frac{df_{v2}}{d\chi}\Delta\chi + \Delta\hat{\nu}\,f_{v2}\right], \tag{3.115}$$

$$\Delta\hat{S} = \frac{\partial\hat{S}}{\partial|\omega|}|\Delta\omega| + \frac{\partial\hat{S}}{\partial\bar{S}}\Delta\bar{S}, \tag{3.116}$$

$$\Delta r = \frac{\partial r}{\partial\hat{\nu}}\Delta\hat{\nu} + \frac{\partial r}{\partial\hat{S}}\Delta\hat{S} = \frac{M_\infty}{Re\,\kappa^2 d^2}\left[\frac{1}{\hat{S}}\Delta\hat{\nu} - \frac{\hat{\nu}}{\hat{S}^2}\Delta\hat{S}\right], \tag{3.117}$$

$$\begin{aligned}|\Delta\omega| = \frac{1}{|\omega|}\Bigg[&\omega_x\left(\frac{\partial(\Delta w)}{\partial y} - \frac{\partial(\Delta v)}{\partial z}\right) + \omega_y\left(\frac{\partial(\Delta u)}{\partial z} - \frac{\partial(\Delta w)}{\partial x}\right)\\ &+ \omega_z\left(\frac{\partial(\Delta v)}{\partial x} - \frac{\partial(\Delta u)}{\partial y}\right)\Bigg],\end{aligned} \tag{3.118}$$

$$|\Delta\omega| = \mathrm{sgn}(\omega)\left[\frac{\partial(\Delta v)}{\partial x} - \frac{\partial(\Delta u)}{\partial y}\right] \quad \text{in 2D.} \tag{3.119}$$

The partial derivatives are provided as follows:

$$\frac{\partial\chi}{\partial\rho} = \frac{\hat{\nu}}{\mu}, \tag{3.120}$$

$$\frac{\partial\chi}{\partial\hat{\nu}} = \frac{\rho}{\mu}, \tag{3.121}$$

$$\frac{df_{v1}}{d\chi} = \frac{3c_{v1}^3\chi^2}{\left(\chi^3 + c_{v1}^3\right)^2}, \tag{3.122}$$

$$\frac{df_{v2}}{d\chi} = -\frac{1 - \chi^2\frac{df_{v1}}{d\chi}}{\left(1 + f_{v1}\chi\right)^2}, \tag{3.123}$$

$$\frac{dg}{dr} = 1 + c_{w2}\left(6r^5 - 1\right), \tag{3.124}$$

$$\frac{df_n}{d\chi} = \frac{6c_{n1}\chi^2}{\left(c_{n1} - \chi^3\right)^2}, \tag{3.125}$$

$$\frac{df_w}{dg} = \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6}\frac{c_{w3}^6}{g^6 + c_{w3}^6}, \tag{3.126}$$

$$\frac{\partial r}{\partial \hat{\nu}} = \frac{M_\infty}{Re}\frac{1}{\kappa^2 d^2 \hat{S}}, \tag{3.127}$$

$$\frac{\partial r}{\partial \hat{S}} = \frac{M_\infty}{Re}\frac{\hat{\nu}}{\kappa^2 d^2 \hat{S}^2}. \tag{3.128}$$

The partial derivatives of $\hat{S}$ are computed as follows:

$$\frac{\partial \bar{S}}{\partial \rho} = \frac{M_\infty}{Re}\frac{\hat{\nu}}{(\kappa d)^2}\left(\frac{\partial f_{v2}}{\partial \chi}\frac{\partial \chi}{\partial \rho}\right), \tag{3.129}$$

$$\frac{\partial \bar{S}}{\partial \hat{\nu}} = \frac{M_\infty}{Re}\frac{\hat{\nu}}{(\kappa d)^2}\left(\frac{\partial f_{v2}}{\partial \chi}\frac{\partial \chi}{\partial \hat{\nu}} + f_{v2}\right). \tag{3.130}$$

$$\frac{\partial \hat{S}}{\partial \rho} = \begin{cases} \frac{\partial \bar{S}}{\partial \rho}, & \text{if } \bar{S} \geq -c_{v2}\,|\omega|, \\ \frac{\partial \bar{S}}{\partial \rho}\,|\omega|\,\frac{c_{v3}\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]+\left[c_{v2}^2\,|\omega|+c_{v3}\,\bar{S}\right]}{\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]^2}, & \text{otherwise,} \end{cases} \tag{3.131}$$

$$\frac{\partial \hat{S}}{\partial \hat{\nu}} = \begin{cases} \frac{\partial \bar{S}}{\partial \hat{\nu}}, & \text{if } \bar{S} \geq -c_{v2}\,|\omega|, \\ \frac{\partial \bar{S}}{\partial \hat{\nu}}\,|\omega|\,\frac{c_{v3}\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]+\left[c_{v2}^2\,|\omega|+c_{v3}\,\bar{S}\right]}{\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]^2}, & \text{otherwise,} \end{cases} \tag{3.132}$$

$$\frac{\partial \hat{S}}{\partial |\omega|} = \begin{cases} 1, & \text{if } \bar{S} \geq -c_{v2}\,|\omega|, \\ 1+\frac{\left(2c_{v2}^2\,|\omega|+c_{v3}\,\bar{S}\right)\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]-(c_{v3}-2c_{v2})\,|\omega|\left[c_{v2}^2\,|\omega|+c_{v3}\,\bar{S}\right]}{\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]^2}, & \text{otherwise.} \end{cases} \tag{3.133}$$

$$\frac{\partial \hat{S}}{\partial \bar{S}} = \begin{cases} 1, & \text{if } \bar{S} \geq -c_{v2}\,|\omega|, \\ \frac{c_{v3}\,|\omega|\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]+|\omega|\left[c_{v2}^2\,|\omega|+c_{v3}\,\bar{S}\right]}{\left[(c_{v3}-2c_{v2})\,|\omega|-\bar{S}\right]^2}, & \text{otherwise.} \end{cases} \tag{3.134}$$

As mentioned earlier, the turbulence equation is divided into three parts—production, destruction, and diffusion—which are given below.

For $\hat{\nu} \geq 0$:

$$\text{Production} = c_{b1}\,\hat{S}\,\hat{\nu}, \tag{3.135}$$

$$\text{Destruction} = \frac{M_\infty}{Re}\,c_{w1}\,f_w\left(\frac{\hat{\nu}}{d}\right)^2, \tag{3.136}$$

$$\text{Diffusion} = \frac{M_\infty}{Re\,\sigma}\left[\nabla\cdot\left(\left(\hat{\nu}\,f_n+\nu\right)\nabla\hat{\nu}\right)+c_{b2}\,\nabla\hat{\nu}\cdot\nabla\hat{\nu}\right]. \tag{3.137}$$

For $\hat{\nu} < 0$:

$$\text{Production} = c_{b1}\,|\omega|\,\hat{\nu}, \tag{3.138}$$

$$\text{Destruction} = \frac{M_\infty}{Re} c_{w1} \left(\frac{\hat{\nu}}{d}\right)^2 \alpha, \tag{3.139}$$

$$\text{Diffusion} = \frac{M_\infty}{Re\,\sigma} \left[\nabla \cdot \left((\hat{\nu} f_n + \nu) \nabla \hat{\nu}\right) + c_{b2} \nabla \hat{\nu} \cdot \nabla \hat{\nu}\right]. \tag{3.140}$$

The implementation of the production term is explained as follows. Since the production term depends on three unknowns (ν, ρ, ω), three production derivatives are defined based on the sign of $\hat{\nu}$.

For $\hat{\nu} \geq 0$:

$$\frac{\partial P}{\partial \rho} = c_{b1} \frac{\partial \hat{S}}{\partial \rho} \hat{\nu}, \tag{3.141}$$

$$\frac{\partial P}{\partial \hat{\nu}} = c_{b1} \left(\frac{\partial \hat{S}}{\partial \hat{\nu}} \hat{\nu} + \hat{S}\right), \tag{3.142}$$

$$\frac{\partial P}{\partial \omega} = c_{b1} \frac{\partial \hat{S}}{\partial \omega} \hat{\nu}. \tag{3.143}$$

For $\hat{\nu} < 0$:

$$\frac{\partial P}{\partial \rho} = 0, \tag{3.144}$$

$$\frac{\partial P}{\partial \hat{\nu}} = c_{b1} |\omega|, \tag{3.145}$$

$$\frac{\partial P}{\partial \omega} = c_{b1} \hat{\nu}. \tag{3.146}$$

A similar approach is applied to the destruction term.

For $\hat{\nu} \geq 0$:

$$\frac{\partial D}{\partial \rho} = \frac{M_\infty}{Re} \frac{c_{w1}}{d^2} \hat{\nu}^2 \frac{\partial f_w}{\partial g} \frac{\partial g}{\partial r} \left(\frac{\partial r}{\partial \hat{S}} \frac{\partial \hat{S}}{\partial \rho}\right), \tag{3.147}$$

$$\frac{\partial D}{\partial \hat{\nu}} = \frac{M_\infty}{Re} \frac{c_{w1}}{d^2} \left(\hat{\nu}^2 \frac{\partial f_w}{\partial g} \frac{\partial g}{\partial r} \frac{\partial r}{\partial \hat{S}} \frac{\partial \hat{S}}{\partial \hat{\nu}} + 2\hat{\nu} f_w\right), \tag{3.148}$$

$$\frac{\partial D}{\partial \omega} = \frac{M_\infty}{Re} \frac{c_{w1}}{d^2} \hat{\nu}^2 \frac{\partial f_w}{\partial g} \frac{\partial g}{\partial r} \left(\frac{\partial r}{\partial \hat{S}} \frac{\partial \hat{S}}{\partial \omega}\right). \tag{3.149}$$

For $\hat{\nu} < 0$:

$$\frac{\partial D}{\partial \rho} = 0, \tag{3.150}$$

$$\frac{\partial D}{\partial \hat{\nu}} = -\frac{M_\infty}{Re} \frac{c_{w1}}{d^2} (2\hat{\nu}), \tag{3.151}$$

$$\frac{\partial D}{\partial \omega} = 0. \tag{3.152}$$

Finally, the production and destruction terms are merged into the source term, which is then incorporated into the Jacobian matrix for the left and right nodes.

In addition to the improvements in the SA turbulence model, updates are required in the Navier-Stokes equations to solve the system in a coupled manner. The method used in this work is two-way coupled, allowing simulations at higher CFL numbers. All calculations are performed with a second-order discretization. When updating the turbulence model, the definition of the turbulent viscosity is revised as follows:

$$\mu_t^* = \rho\,\hat{\nu}\,f_{v1}. \tag{3.153}$$

All terms using the new turbulent viscosity definition must be updated in the Navier-Stokes equations to provide a coupled solution method. Therefore, the viscous Jacobian is corrected for each viscous flux according to the changes in the turbulent viscosity. A new vector, $\frac{\partial \mu_t^*}{\partial W}$ (see Equation 3.154), is defined to store the derivatives of the turbulent viscosity with respect to the primitive variables. Recall that $\mu_t^* = \rho\,\hat{\nu}\,f_{v1}$ and that f_{v1} is a function of temperature, density, and viscosity. Using this vector, the Jacobian corrections are applied for both the left and right nodes. The derivatives of temperature with respect to the conservative variables are given in Equation 3.155. This term is calculated for ensuring that temperature-dependent effects are accurately accounted for in the viscous Jacobian corrections.

$$\frac{\partial \mu_t^*}{\partial W} = \begin{Bmatrix} f_{v1}\hat{\nu} + \hat{\nu}\rho\frac{\partial f_{v1}}{\partial \chi}\frac{\partial \chi}{\partial \rho} \\ 0 \\ 0 \\ 0 \\ \hat{\nu}\rho\frac{\partial f_{v1}}{\partial \chi}\frac{\partial \chi}{\partial \mu}\frac{\partial \mu}{\partial T} \\ \rho f_{v1} + \rho\hat{\nu}\frac{df_{v1}}{d\chi}\frac{d\chi}{d\hat{\nu}} \end{Bmatrix} \tag{3.154}$$

$$\frac{\partial T}{\partial Q} = \begin{Bmatrix} \frac{\gamma-1}{\rho R}\left(u^2+v^2+w^2-\frac{1}{2\rho}(E_L+E_R)\right) \\ \frac{\gamma-1}{\rho R}u \\ \frac{\gamma-1}{\rho R}v \\ \frac{\gamma-1}{\rho R}w \\ \frac{\gamma-1}{\rho R} \end{Bmatrix} \tag{3.155}$$

3.2.2 f_{t2} modification

In turbulence modeling, the f_{t2} term which is called as laminar suppression term [63] refers to a component in the Spalart-Allmaras turbulence model that helps suppress turbulence in regions of strong flow rotation and streamline curvature. The SA model is particularly effective at predicting both attached and separated flows. It is based on a single transport equation for the eddy viscosity, which represents the effect of turbulence on the mean flow. The transition term is added to this transport equation to incorporate the effects of flow transition.

During flow transition, the flow evolves from laminar (smooth and ordered) to turbulent (chaotic and irregular). This transition is influenced by local flow conditions, pressure gradients, and surface roughness. In the SA model, the transition term acts as a source or sink in the eddy viscosity transport equation, modifying the turbulence production and damping rates. The specific form of the transition term depends on the model variant; various modifications have been proposed based on empirical correlations and experimental data.

Overall, the transition term plays a crucial role in accurately simulating flow transition and capturing the effects of laminar-to-turbulent change. It enables the model to handle a wide range of flow conditions, making it suitable for diverse engineering applications. Consequently, this feature has been added to the current version of the HEMLAB algorithm. As described in the first chapter, the SA-negative model consists of two different equations based on the sign of the turbulent viscosity. Similarly, the equations for the Spalart-Allmaras turbulence model with the transition term are given below. For

$\hat{\nu} \geq 0$:

$$\begin{aligned}\frac{\partial \hat{\nu}}{\partial t} + \mathbf{u}\cdot\nabla\hat{\nu} &= c_{b1}(1-f_{t2})\hat{S}\hat{\nu} \\ &- \frac{M_\infty}{Re_\infty}\left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\hat{\nu}}{d}\right)^2 \\ &+ \frac{M_\infty}{Re_\infty \sigma}\left[\nabla\cdot((\hat{\nu}f_n+\nu)\nabla\hat{\nu}) + c_{b2}\nabla\hat{\nu}\cdot\nabla\hat{\nu}\right] \end{aligned} \tag{3.156}$$

For $\hat{\nu} < 0$:

$$\begin{aligned}\frac{\partial \hat{\nu}}{\partial t} + \mathbf{u}\cdot\nabla\hat{\nu} &= c_{b1}(1-c_{t3})|\omega|\hat{\nu} \\ &+ \frac{M_\infty}{Re_\infty}c_{w1}\left(\frac{\hat{\nu}}{d}\right)^2\alpha \\ &+ \frac{M_\infty}{Re_\infty \sigma}\left[\nabla\cdot((\hat{\nu}f_n+\nu)\nabla\hat{\nu}) + c_{b2}\nabla\hat{\nu}\cdot\nabla\hat{\nu}\right] \end{aligned} \tag{3.157}$$

Here,

$$f_{t2} = c_{t3}\,e^{-c_{t4}\chi^2} \tag{3.158}$$

$$c_{t3} = 1.2 \tag{3.159}$$

$$c_{t4} = 0.5 \tag{3.160}$$

The production term and its implementation are explained below. Since the production term depends on three unknowns (ν, ρ, ω), separate production derivatives are defined based on the sign of $\hat{\nu}$.

For $\hat{\nu} \geq 0$:

$$\frac{\partial P}{\partial \rho} = c_{b1}(1-f_{t2})\frac{\partial \hat{S}}{\partial \rho}\hat{\nu} - c_{b1}\hat{S}\hat{\nu}\frac{df_{t2}}{d\chi}\frac{\partial \chi}{\partial \rho} \tag{3.161}$$

$$\frac{\partial P}{\partial \hat{\nu}} = c_{b1}(1-f_{t2})\left(\frac{\partial \hat{S}}{\partial \hat{\nu}}\hat{\nu} + \hat{S}\right) - c_{b1}\hat{S}\hat{\nu}\frac{df_{t2}}{d\chi}\frac{\partial \chi}{\partial \hat{\nu}} \tag{3.162}$$

$$\frac{\partial P}{\partial \omega} = c_{b1}(1-f_{t2})\frac{\partial \hat{S}}{\partial \omega}\hat{\nu} \tag{3.163}$$

For $\hat{\nu} < 0$:

$$\frac{\partial P}{\partial \rho} = 0 \tag{3.164}$$

$$\frac{\partial P}{\partial \hat{\nu}} = c_{b1}(1-c_{t3})|\omega| \tag{3.165}$$

$$\frac{\partial P}{\partial \omega} = c_{b1}\hat{\nu} \tag{3.166}$$

where

$$\frac{df_{t2}}{d\chi} = 2\,c_{t3}\,c_{t4}\,\chi\,e^{-c_{t4}\chi^2} \tag{3.167}$$

A similar approach is applied to the destruction term. The modified production and destruction terms are then merged into the source term and incorporated into the Jacobian matrix.

For $\hat{\nu} \geq 0$:

$$\begin{aligned}
\frac{\partial D}{\partial \rho} &= \frac{M_\infty}{Re}\frac{c_{w1}}{d^2}\hat{\nu}^2\frac{\partial f_w}{\partial g}\frac{\partial g}{\partial r}\left(\frac{\partial r}{\partial \hat{S}}\frac{\partial \hat{S}}{\partial \rho}\right) && (3.168)\\
&- \frac{M_\infty}{Re}\frac{c_{b1}}{\kappa^2 d^2}\hat{\nu}^2\frac{df_{t2}}{d\chi}\frac{d\chi}{d\rho}\\
\frac{\partial D}{\partial \hat{\nu}} &= \frac{M_\infty}{Re}\frac{c_{w1}}{d^2}\left(\hat{\nu}^2\frac{\partial f_w}{\partial g}\frac{\partial g}{\partial r}\left(\frac{\partial r}{\partial \hat{S}}\frac{\partial \hat{S}}{\partial \hat{\nu}}+\frac{\partial r}{\partial \hat{\nu}}\right)+2\hat{\nu}\,f_w\right) && (3.169)\\
&- \frac{M_\infty}{Re}\frac{c_{b1}}{\kappa^2 d^2}\hat{\nu}^2\frac{df_{t2}}{d\chi}\frac{d\chi}{d\hat{\nu}}\\
\frac{\partial D}{\partial \omega} &= \frac{M_\infty}{Re}\frac{c_{w1}}{d^2}\hat{\nu}^2\frac{\partial f_w}{\partial g}\frac{\partial g}{\partial r}\left(\frac{\partial r}{\partial \hat{S}}\frac{\partial \hat{S}}{\partial \omega}\right) && (3.170)
\end{aligned}$$

For $\hat{\nu} < 0$:

$$\begin{aligned}
\frac{\partial D}{\partial \rho} &= 0 && (3.171)\\
\frac{\partial D}{\partial \hat{\nu}} &= -\frac{M_\infty}{Re}\frac{c_{w1}}{d^2}(2\hat{\nu})\,\alpha && (3.172)\\
\frac{\partial D}{\partial \omega} &= 0. && (3.173)
\end{aligned}$$

3.2.3 Rotation correction (SA-R)

The SA-R model [64] was developed to improve the prediction of flows with significant rotation and curvature, such as swirling flows or flows around curved surfaces. These flows exhibit additional complexities compared to the original SA model, which assumes predominantly irrotational flow. In the SA-R model, two primary modifications are introduced: the curvature correction term and the rotation correction term. These terms adjust the turbulent viscosity and production/dissipation rates to capture the anisotropic behavior induced by rotation and curvature effects.

The curvature correction term accounts for the influence of flow curvature on the turbulence field. It is designed to enhance the representation of secondary flows,

vortex generation, and the redistribution of turbulence in curved flows. The curvature correction term modifies the production and dissipation of turbulent eddy viscosity by considering the effects of flow curvature.

The rotation correction term aims to capture the effects of flow rotation on the turbulence field. It improves the prediction of the anisotropic behavior induced by rotation, such as the elongation of turbulent structures in the direction of rotation. This term modifies the production and dissipation of turbulent eddy viscosity by incorporating the rotational effects. Both the curvature correction term and the rotation correction term are typically formulated based on empirical correlations or experimental data. The specific functional forms of these terms may vary depending on the version or variant of the SA-R model used. By including these additional corrections, the SA-R model offers improved accuracy in predicting turbulent flows with rotation and curvature. It enables better simulation of complex flow phenomena, such as swirling flows, flows around curved surfaces, or flows with significant rotational effects, where the original SA model may not capture the flow characteristics accurately. So, the rotational correction has been added to the current version of the HEMLAB code in order to capture the flow characteristics such reagions. The equation form of the rotational correction is given below.

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

$$\begin{aligned} \frac{\partial \hat{\nu}}{\partial t} + \mathbf{u} \cdot \nabla\, \hat{\nu} &= c_{b1}(1-f_{t2})(\hat{S} + c_{rot} min[0, S-|\omega|])\hat{\nu} \\ &- \frac{M_\infty}{Re_\infty}[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}](\frac{\hat{\nu}}{d})^2 \\ &+ \frac{M_\infty}{Re_\infty \sigma}[\nabla \cdot [(\hat{\nu} f_n + \nu)\nabla \hat{\nu}\,] + c_{b2} \nabla \hat{\nu} \cdot \nabla \hat{\nu}\,] \end{aligned} \tag{3.174}$$

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

$$\begin{aligned} \frac{\partial \hat{\nu}}{\partial t} + \mathbf{u} \cdot \nabla\, \hat{\nu} &= c_{b1}\,(1-c_{t3})\,|\omega + c_{rot} min[0, S-|\omega|]\hat{\nu} \\ &- \frac{M_\infty}{Re_\infty} c_{w1}(\frac{\hat{\nu}}{d})^2 \alpha \\ &+ \frac{M_\infty}{Re_\infty \sigma}[\nabla \cdot [(\hat{\nu} f_n + \nu)\nabla \hat{\nu}\,] + c_{b2} \nabla \hat{\nu} \cdot \nabla \hat{\nu}\,] \end{aligned} \tag{3.175}$$

Equations below are obtained by doing very similar updates on the code for the production, destruction and diffusion parts for the rotation correction modification.

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

$$\frac{\partial P}{\partial \rho} = c_{b1}(1-f_{t2})(\frac{\partial \hat{S}}{\partial \bar{S}}\frac{\partial \bar{S}}{\partial \rho}\hat{\nu}) \tag{3.176}$$

$$\frac{\partial P}{\partial \hat{\nu}} = c_{b1}\,(1-f_{t2})(\frac{\partial \hat{S}}{\partial \bar{S}}\frac{\partial \bar{S}}{\partial \hat{\nu}}\hat{\nu} + \hat{S} + c_{rot}min[0, S-|\omega|]) \tag{3.177}$$

$$\frac{\partial P}{\partial \omega} = c_{b1}\,(1-f_{t2})(\frac{\partial \hat{S}}{\partial \omega} + \frac{\partial (c_{rot}min[0, S-|\omega|])}{\partial \omega})\hat{\nu} \tag{3.178}$$

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

$$\frac{\partial P}{\partial \rho} = 0 \tag{3.179}$$

$$\frac{\partial P}{\partial \hat{\nu}} = c_{b1}(1-c_{t3})(|\omega| + c_{rot}min[0, S-|\omega|]) \tag{3.180}$$

$$\frac{\partial P}{\partial \omega} = c_{b1}(1-c_{t3})(1 + \frac{\partial (c_{rot}min[0, S-|\omega|])}{\partial \omega})\,\hat{\nu} \tag{3.181}$$

Nothing is changed for the destruction terms.

3.2.4 QCR-2000 modification

The Spalart-Allmaras One-Equation Model with Quadratic Constitutive Relation, 2000 version (SA-QCR2000), is an enhanced version of the original SA turbulence model detailed in [65]. It incorporates a quadratic constitutive relation to improve the representation of turbulence production and dissipation rates. This model was developed to address limitations in predicting flows with adverse pressure gradients and separation, providing better accuracy in regions where the flow transitions from attached to separated or recirculating.

The SA-QCR2000 model introduces a quadratic term in the constitutive relation for the turbulent eddy viscosity, which relates the Reynolds stresses to the mean strain rate. Unlike the original SA model's linear constitutive relation, the quadratic relation captures turbulence anisotropy more effectively by considering nonlinear interactions between Reynolds stresses and strain rates. Since this model has been recommended for the AIAA High Lift Prediction Workshop Series V, it has been implemented in the current version of the HEMLAB algorithm. The equations for the SA-QCR2000 model are given below.

For $\hat{\nu} \geq 0$:

$$\frac{\partial \hat{\nu}}{\partial t} + \mathbf{u}\cdot\nabla\hat{\nu} = c_{b1}\,(1-f_{t2})\,\hat{S}\hat{\nu}$$

$$
\begin{aligned}
&- \frac{M_\infty}{Re_\infty}\left[c_{w1}\, f_w - \frac{c_{b1}}{\kappa^2}\, f_{t2}\right]\left(\frac{\hat{\nu}}{d}\right)^2 \\
&+ \frac{M_\infty}{Re_\infty\, \sigma}\left[\nabla\cdot((\hat{\nu}\, f_n + \nu)\nabla\hat{\nu}) + c_{b2}\,\nabla\hat{\nu}\cdot\nabla\hat{\nu}\right]. \qquad (3.182)
\end{aligned}
$$

For $\hat{\nu} < 0$:

$$
\begin{aligned}
\frac{\partial\hat{\nu}}{\partial t} + \mathbf{u}\cdot\nabla\hat{\nu} &= c_{b1}\,(1-c_{t3})\,|\omega|\hat{\nu} \\
&- \frac{M_\infty}{Re_\infty}\, c_{w1}\left(\frac{\hat{\nu}}{d}\right)^2 \alpha \\
&+ \frac{M_\infty}{Re_\infty\, \sigma}\left[\nabla\cdot((\hat{\nu}\, f_n + \nu)\nabla\hat{\nu}) + c_{b2}\,\nabla\hat{\nu}\cdot\nabla\hat{\nu}\right]. \qquad (3.183)
\end{aligned}
$$

In these equations, the SA-QCR2000 model does not only modify the transport equation but also alters the turbulent stress tensor. The QCR turbulent stress tensor is defined differently from the classical model. For example:

$$
\begin{aligned}
\tau_{xx,qcr} &= \tau_{xx} - 2c_{r1}\,(O_{xy}\,\tau_{xy} + O_{xz}\,\tau_{xz}), && (3.184)\\
\tau_{yy,qcr} &= \tau_{yy} + 2c_{r1}\,(O_{xy}\,\tau_{yx} - O_{yz}\,\tau_{yz}), && (3.185)\\
\tau_{zz,qcr} &= \tau_{zz} + 2c_{r1}\,(O_{xz}\,\tau_{zx} + O_{yz}\,\tau_{zy}), && (3.186)\\
\tau_{xy,qcr} &= \tau_{xy} - c_{r1}\,(O_{xy}(\tau_{yy} - \tau_{xx}) + O_{yz}\,\tau_{xz} + O_{xz}\,\tau_{yz}), && (3.187)\\
\tau_{xz,qcr} &= \tau_{xz} - c_{r1}\,(O_{xz}(\tau_{zz} - \tau_{xx}) + O_{xy}\,\tau_{zy} - O_{yz}\,\tau_{xy}), && (3.188)\\
\tau_{yz,qcr} &= \tau_{yz} - c_{r1}\,(O_{yz}(\tau_{zz} - \tau_{yy}) - O_{xy}\,\tau_{zx} - O_{xz}\,\tau_{yx}), && (3.189)
\end{aligned}
$$

with

$$
\begin{aligned}
c_{r1} &= 0.3, && (3.190)\\
O_{xx} &= O_{yy} = O_{zz} = 0, && (3.191)\\
O_{xy} &= -O_{yx} = \frac{\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}}{b}, && (3.192)\\
O_{xz} &= -O_{zx} = \frac{\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}}{b}, && (3.193)\\
O_{yz} &= -O_{zy} = \frac{\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}}{b}, && (3.194)
\end{aligned}
$$

$$
b = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2} \qquad (3.195)
$$

3.3 Integration of Scalable Nonlinear Equations Solvers (SNES)

The present nonlinear Newton method is implemented using PETSc's Scalable Nonlinear Equations Solvers (SNES) with a line search technique [47], integrated into the implicit vertex-based unstructured finite volume algorithm on hybrid meshes. PETSc-SNES provides a user-friendly interface for Newton methods to solve nonlinear equations [66], offering various Newton-like solvers based on line search and trust region methods. For the current fully coupled system of nonlinear equations, we use the line search Newton method (SNESNEWTONLS).

The computational domain is divided into sub-domains using the METIS library [49] for balanced domain decomposition. After decomposition, vertices are renumbered using the reverse Cuthill-McKee algorithm from PARSEPACK [67]. To enhance cache efficiency, edges are constructed in increasing order based on the node number at one end of each edge [68]. In addition, sub-domain boundary vertices required for parallel communication are ordered by their destination processor ID, which allows for more efficient data packaging during parallel communication.

The application of the fully coupled Newton method with pseudo-transient continuation (PTC) leads to the following system:

$$\begin{aligned}\left[\frac{M}{\Delta t}+\frac{M}{\Delta t_{CFL}}+J\right]\Delta\mathbf{Q}^{m+1} &= -\mathbf{R}(\mathbf{Q}^m)\\ &= -\left[M\frac{\mathbf{Q}^m-\mathbf{Q}^n}{\Delta t}V+M\frac{\mathbf{Q}^m-\mathbf{Q}^n}{\Delta t_{CFL}}V\right.\\ &\quad\left.+\sum\mathbf{n}\cdot\mathbf{F_i}(\mathbf{Q}^m)S-\sum\mathbf{n}\cdot\mathbf{F_v}(\mathbf{Q}^m)S\right].\end{aligned} \tag{3.196}$$

The pseudo time step computed from the CFL number is defined as

$$\Delta t_{CFL}=\min\left(\frac{V_i}{\sum_{k=1}^{N}\lambda_{c,k}S_k},\frac{V_i}{2\sum_{k=1}^{N}\lambda_{v,k}S_k^2}\right)CFL, \tag{3.197}$$

where $\lambda_c=\max(|\lambda_i|)$ and

$$\lambda_v=\frac{\mu}{\rho V_i}\max\left(\frac{4}{3},\frac{\gamma}{Pr}\right)\frac{M_\infty}{Re}.$$

The solver also offers CFL options based on the minimum outgoing edge length and a distance function, in which the minimum outgoing edge length is replaced by a

slightly modified wall distance function. This option can be especially useful for highly irregular anisotropic boundary layer meshes.

As Δt_{CFL} is gradually increased to infinity, the above approximation converges to the classical Newton method. This approach leads to a consistent Newton method similar to that of Thompson and O'Connell [69], where the derivation of the right-hand side exactly matches the left-hand side. The advantage is that the residual can be significantly reduced in just a few Newton sub-iterations, and the block ILU(0) preconditioner is usually robust enough for three-dimensional problems. Additionally, it allows for larger CFL numbers at startup due to the damping effect of the time derivative on the right-hand side. However, as the solution approaches the final steady state, the convergence of the Newton method may slow if the ramped CFL number is not sufficiently high. The transient term (i.e., the second term with Δt_{CFL} on the right-hand side) is generally not included in pseudo-transient continuation algorithms, which require more robust preconditioners. This term can compromise time accuracy when the final CFL number is not infinite and must be removed for unsteady calculations. Although the current algorithm supports both consistent and non-consistent approaches, we prefer the consistent approach in our numerical simulations.

PETSc-SNES is implemented using both the Jacobian-free approach and the direct approach, with either first- or second-order Jacobian matrix construction for the matrix-vector product. The following FORTRAN90 subroutine which is given in Figure 3.3 demonstrates the implementation.

In this implementation, the right-hand side residual vector is computed in the `Form_Residual` function, while the Jacobian matrix is evaluated in the `Form_Jacobian` function. The current PETSc-SNES setup typically uses the first-order Jacobian matrix as a preconditioner in the matrix-free approach (via the Fréchet derivative). The method for computing the differencing parameters for the finite difference Jacobian-free formulation follows the default approach proposed by Walker and Pernice [70] with $e_{rel} = 1 \times 10^{-6}$. The matrix-free approach can be sensitive to the e_{rel} parameter, and we observed convergence difficulties at high CFL numbers with the default value $e_{rel} = 1 \times 10^{-8}$. While the first-order Jacobian matrix is usually employed as a preconditioner in the Jacobian-free approach, constructing

the second-order Jacobian is also possible, though it significantly increases memory requirements.

```
subroutine Implicit_Euler( A00, mesh_info, solver_vrbls, rms,
dt_min, dt_max )
#include <petsc/finclude/petsc.h>
use petsc
CALL SNESCreate(PETSC_COMM_WORLD,snes,ierr)
CALL SNESSetFunction(snes,RH0,Form_Residual,mesh_info,ierr)
IF ( solver_vrbls%Matrix_Free_Jacob ) THEN
CALL MatCreateSNESMF(snes,AA,ierr)
CALL MatMFFDSetFunction(AA,Form_MFFDFunction,mesh_info,ierr)
CALL SNESSetUpdate(snes,Form_Update,ierr)
CALL SNESSetJacobian(snes,AA,A00,Form_Jacobian,mesh_info,ierr)
CALL MatMFFDSetType(AA,MATMFFD_WP,ierr) !MATMFFD_DS
CALL MatMFFDSetFunctionError(AA,1e-6,ierr)
CALL MatMFFDDSSetUmin(AA,1e-4,ierr)
ELSE
CALL SNESSetJacobian(snes,A00,A00,Form_Jacobian,mesh_info,ierr)
END IF
end subroutine solve_implicit_euler
```

Figure 3.3 : Pseudo-code of the Implicit_Euler subroutine using PETSc SNES.

Additionally, the first- or second-order Jacobian matrix can be used directly for the Jacobian-vector product, which is beneficial when finite difference round-off errors become significant in the Jacobian-free method. To implement this, replace the symbolic Jacobian-free matrix AA with the Jacobian matrix A00 in the `SNESSetJacobian` function, as illustrated above.

The Jacobian matrix is constructed using the PETSc sparse parallel matrix in block compressed row (MATBAIJ) format, which significantly reduces memory usage and speeds up matrix-vector multiplications. The impact of different PETSc matrix formats on computational efficiency is detailed in [52]. The Jacobian matrix can be computed at each SNES iteration or frozen for several iterations to accelerate the calculations.

Although the Jacobian matrix can be computed using the finite difference approximation within PETSc, this approach is computationally expensive and is recommended only for debugging purposes. The current algorithm has been compiled in both double- and quad-precision, with quad-precision (–with-precision=__float128) used to validate the Jacobian matrix implementation for the compressible RANS equations. Tests on a small three-dimensional mesh showed that the relative error

in the Frobenius norm ($||J - J_{fd}||_F/||J||_F$) is 7.252×10^{-14}, where J_{fd} is the Jacobian evaluated using the finite difference approach provided by PETSc. For the laminar Navier-Stokes equations, this error is reduced to 1.20512×10^{-16}. Running the same test case with double precision results in larger truncation errors, making it difficult to assess accuracy due to very small or large off-diagonal block entries between the Navier-Stokes equations and the turbulence model. The preconditioning method is based on the restricted additive Schwarz method with block incomplete LU factorization with k levels of fill (ILU(k)) within each sub-block. The block ILU(k) preconditioner can be obtained directly from the PETSc library or from the HYPRE [71] Euclid ILU(k) preconditioner, accessed through the PETSc interface.

The linear solver can use various Krylov subspace methods, such as the flexible generalized minimal residual method (FGMRES) [72], the flexible generalized conjugate residual method (KSPGCR) [73], or the stabilized biconjugate gradient method (KSPBCGS) [74]. The Newton method employs the default backtracking line search algorithm (SNESLINESEARCHBT), which is essential for preventing divergence during startup caused by aggressive CFL ramping.

If a consistent Newton approach is used with a constant CFL number at each iteration, the PETSc-SNES algorithm requires only these two functions. However, updating the CFL within Newton sub-iterations and implementing the dual-time method necessitates additional modifications.

For the unsteady dual-time approach, the `Form_Residual` function handles the right-hand side evaluation, while the `Form_MFFDFunction` subroutine performs the matrix-free Jacobian-vector multiplications. This same method is applied for the non-consistent implementation of the PTC algorithm. Although this approach can converge faster to a steady state, the restricted additive Schwarz method with the ILU(0) preconditioner is typically not robust enough for three-dimensional cases. Moreover, it requires evaluating an additional vector from the `Form_MFFDFunction` subroutine since the right-hand side vector cannot be used in finite differencing for the Jacobian-vector multiplication.

While agglomeration multigrid approaches with line smoothers are known to be robust for anisotropic meshes [75], their overall effectiveness remains uncertain. However,

successive anisotropic mesh refinement, with increasing complexity, significantly reduces the burden on the linear solver by interpolating the converged solution from the previous level, leading to better startup conditions. This is crucial for the convergence of the Newton method, as the initial guess must be sufficiently close to the final steady state to achieve quadratic convergence.

Another challenge with the PETSc-SNES solver is achieving proper scaling to ensure convergence for the coupled algebraic equations. We tested several scaling methods, including scaling by the Jacobian diagonal, $1/\text{volume}$, $(1/\text{volume})^{1/3}$, minimum outgoing edge length/volume, $1/$maximum Jacobian row entry absolute value, and $1/$sum of the absolute values in each Jacobian row. We found that scaling by the maximum Jacobian row absolute value sum and $(1/\text{volume})^{1/3}$ provided good convergence. The latter is also used in the Hierarchical Adaptive Nonlinear Iteration Method (HANIM) framework [76]. However, PETSc does not permit scaling changes between SNES iterations because this disrupts the convergence of the line search algorithm; therefore, scaling must be applied externally.

Complex CFL update and ramping strategies based on line search algorithms are discussed in the literature [69, 77, 78], but our current approach employs a simpler strategy. Here, the CFL number is multiplied by a factor (between 1 and 1.5) if the initial residual is reduced by an order of magnitude. If the reduction is less than 0.8, the CFL number is reduced by a factor of 0.1; otherwise, no modification is made during that Newton iteration. This CFL ramping strategy is implemented within the `Implicit_Euler` subroutine for both the consistent Newton approach and the dual approach, thereby keeping the CFL number fixed within the Newton sub-iterations.

Another approach to modifying the CFL number is to implement it within the `Form_Update` function, which would allow for CFL ramping within Newton sub-iterations. However, this method requires careful consideration of the line search algorithm, as increasing the CFL number during sub-iterations can initially increase the residual, which the line search algorithm may interpret as divergence and potentially halt the updates. This issue is particularly challenging for the turbulence model, where the residual initially increases before eventually decreasing. Some authors have explored excluding the turbulence residual from the line search algorithm [79] to

address this problem. Additionally, scaling the Jacobian matrix presents challenges, as scaling cannot be altered within the SNES solver.

Alternatively, PETSc-SNES can be used as a one-step classical Newton method without a line search algorithm (via the SNESKSPONLY option), as described in [38]. This setup is suitable for solving relatively simple problems at low angles of attack. Moreover, the algorithm has been integrated with the SLEPc library [80] to explore leading eigenvalues, typically using the Jacobi-Davidson method to compute the smallest eigenvalue.

3.4 Improvements on the Mesh Adaptation Process

The integration of pyAMG and NASA-REFINE into the HEMLAB algorithm has significantly enhanced the mesh adaptation process, improving both accuracy and efficiency in computational fluid dynamics (CFD) simulations. By using the strengths of these advanced mesh adaptation tools, the process now incorporates multiscale anisotropic refinement techniques, better geometric preservation, and optimized boundary layer generation. This integration ensures that the computational resources are allocated more effectively to regions of interest, such as shock waves, boundary layers, and separation zones, leading to improved solution quality while maintaining manageable computational costs. Additionally, the combination of these tools provides a more adaptive and flexible refinement framework, allowing automated adjustment of mesh resolution based on flow physics, thereby reducing the reliance on manual meshing strategies.

One of the key advantages of this integration is the ability to leverage different adaptation methodologies for various flow regimes. pyAMG's multiscale anisotropic adaptation excels at preserving fine-scale flow structures by efficiently redistributing mesh density, whereas NASA-REFINE's metric tensor-based approach ensures targeted refinement in critical flow regions without excessive computational overhead. The synergy between these two methods enables a comprehensive refinement strategy, balancing localized resolution enhancement with global mesh quality control.

Furthermore, this integration enhances boundary layer handling, a crucial aspect in high-fidelity simulations involving high Reynolds number flows. By utilizing

metric-based refinement and structured adaptation techniques, the computational mesh is better aligned with wall-bounded flow characteristics, reducing numerical diffusion and improving accuracy in turbulence modeling. The automated adaptation workflow reduces preprocessing time and ensures consistent mesh quality across multiple adaptation cycles.

The adaptation methodology also incorporates recent advancements in anisotropic mesh refinement algorithms. The feature-based adaptation technique relies on a Riemannian metric space formulation, allowing edge lengths and element volumes to be determined based on the metric field [81]. This ensures alignment of mesh elements with local eigenvectors of the metric tensor, generating highly optimized anisotropic meshes. Additionally, the adaptation process utilizes Hessian-based multiscale metric computation, where the metric field is scaled using the Hessian determinant to minimize interpolation errors in the L_p norm.

In practical applications, the sensor function plays a crucial role in guiding mesh adaptation. The sensor function can utilize flow variables such as Mach number, pressure, entropy, and FTLE. The current implementation employs an exponential distance-based sensor function:

$$Sensor \quad = \quad \sum_{i=1}^{4} 0.25 \; e^{-100 \times 10^{i} d} \tag{3.198}$$

where d represents the minimum wall distance. This enables the refinement process to begin even without an initial boundary layer, progressively forming a structured boundary layer mesh. Once established, further adaptation incorporates the Mach number as a refinement criterion:

$$Sensor \quad = \quad Mach \tag{3.199}$$

This multistage approach ensures an efficient and accurate refinement process, focusing computational effort on critical flow regions while preserving geometric fidelity. Additional sensor functions, including entropy-based and entropy-Mach combined metrics, are available for capturing wake shear layers and complex flow structures [38, 82].

Moreover, special attention is given to the challenges of anisotropic adaptation in high-lift configurations and complex geometries. The adaptation process prevents

numerical issues such as tetrahedral elements with all nodes on solid walls, which can cause singularities in the flux computations. The NASA-REFINE library enforces conditions ensuring proper edge connectivity to non-boundary nodes, mitigating potential numerical instabilities [83].

The following sections provide a detailed discussion of the HEMLAB-pyAMG and HEMLAB-REFINE integrations, outlining the methodologies, implementation details, and their impact on the adaptation process.

3.4.1 HEMLAB-pyAMG integration

To further enhance numerical accuracy, the anisotropic mesh adaptation library pyAMG from INRIA [36] is integrated into the HEMLAB solver. The edge lengths and element volumes inside the mesh generator are computed in the Riemannian metric space defined by the given metric field using a multiscale approach. It employs anisotropic cavity-based operators, where standard operations such as collapse, insertion, and swaps are recast within this cavity framework to perform anisotropic mesh adaptation.

This feature-based adaptation technique can utilize one variable of the flow fields, such as Mach number, pressure, entropy, FTLE, etc. In the current multiscale approach, the Mach number is modified by adding exponential functions based on the minimum wall distance, which is then used as a sensor function, as shown below:

$$Mach\ Sensor = \beta Mach + \sum_{i=1}^{5} 0.25\ e^{-100x10^{i}d} \tag{3.200}$$

In this equation, if β is set to zero, only the distance function is considered during adaptation. If β is set to 1, both the distance function and the Mach number are taken into account. With this approach, mesh refinement can begin even without an initial boundary layer. In this case, several successive refinements based solely on the distance function are applied to form the boundary layer mesh. Once a well-refined boundary layer mesh is obtained, the refinement process can continue using the sensor function, incorporating both the Mach number and wall distance. Additionally, a $p-$norm, which controls the interpolation error in the sensor function in the L norm, is used to manage error control.

The classical mesh adaptation process described above tends to lose some parts of the original geometry during interpolation, leading to inaccuracies in the adapted mesh. To overcome this issue, a background mesh finer than the initial mesh has been used as a reference during the adaptation process. This feature is referred to as "adap_back" in the original pyAMG library. Using this feature, the results obtained protect the original geometry during the refinement.

The adapted surface mesh is projected onto the background mesh in Gamma mesh format which is given in the Figure 3.5. The MeshLink library [84], which provides mesh-geometry associativity in an open framework, is utilized to obtain the edge and corner data required for the Gamma mesh format. The current multiscale mesh adaptation is performed using the Gamma mesh format instead of the SU2 format, where such data is lost after mesh refinement. Therefore, additional code has been written to convert and build the Gamma mesh format. Furthermore, the NASA Refine anisotropic mesh adaptation library [83] is used to facilitate the conversion of mesh file formats. The updated mesh adaptation code parameters have been added to the refinement part of the input file of HEMLAB, as shown below in Figure 3.4.

```
&ADAP_OPTS
adap_sizes = (400000 ; 800000 ; 2000000 ; 4000000) ;
adap_norm = 4,
adap_subite = (9 ; 1 ; 3 ; 1 ),
adap_ext_ite = (0 ; 400 ; 200 ; 200),
adap_residual_reduction = (3 ; 3 ; 3 ; 3),
adap_sensor = (DIST; DIST; MACH_DIST; MACH_DIST),
adap_hmin = 1e-6,
adap_hmax = 4.0,
adap_hgrad = 1.2,
adap_ridge= false,
adap_back = back_reoriented.mesh
```

Figure 3.4 : Adaptation part of the input file of HEMLAB used for pyAMG-based mesh adaptation.

```
MeshVersionFormatted 3
Dimension 3

Vertices
1534333
-22.83464     0.00000         22.83464          0
-17.50000     0.00000         22.53564          0
-15.50000     0.00000         12.32464          0
----
Triangles
578006
2380         2381         39348            1
2381         2382         39349            1
2382         23823        39350            1
----
Tetrahedra
8668795
22193        22119        22118      1349447            0
698945       698944       698947     1407227            0
398945       398944       398947     1207227            0
----
Edges
21581
1           2            0
2           3            0
3           4            0
----
Corners
289
1
2
----
```

Figure 3.5 : Sample Gamma Mesh format showing vertices, triangles, tetrahedra, edges, and corners.

3.4.2 HEMLAB-REFINE integration

NASA-REFINE is a widely used adaptation tool in the aerospace industry that enhances the quality and efficiency of computational fluid dynamics (CFD) simulations through dynamic and adaptive mesh refinements. These tools are essential for addressing challenges associated with high Reynolds numbers, compressible flows, and complex geometries. The primary goal of mesh adaptation is to optimize computational resources, focusing refinement efforts on critical flow features such as shock waves, boundary layers, and separation zones, which are crucial for accurately capturing aerodynamic forces and heat transfer [85].

NASA-REFINE employs a metric tensor-based adaptation approach, which is central to its methodology. The metric tensor defines the desired size, shape, and orientation of mesh elements throughout the computational domain, enabling anisotropic mesh refinement. This allows elements to be stretched or compressed in alignment with directional flow gradients, significantly improving accuracy in resolving vortices, wakes, and other complex flow structures. Large, uniform meshes would otherwise require excessive computational cost, but adaptation dynamically sets mesh density, optimizing computational efficiency.

A particularly challenging aspect of CFD is resolving viscous layers near solid boundaries, where high velocity and temperature gradients exist. NASA-REFINE addresses this challenge through viscous layer refinement, ensuring proper resolution and alignment of mesh elements within boundary layers. The refinement process incorporates models such as the “Spalding” formula for boundary-layer thickness to guide mesh refinement, ensuring accurate computation of aerodynamic forces and heat transfer at walls [86].

Unlike its traditional implementation, where adjoint-based error estimation is used to identify refinement regions based on sensitivities to a quantity of interest (QoI), the current framework does not utilize this approach. Instead, our framework integrates NASA-REFINE as an adaptation tool by directly leveraging metric tensors computed for a specific sensor function. The sensor function captures critical flow characteristics, such as Mach number gradients, and guides the refinement process accordingly. This approach eliminates the need for adjoint solutions while still ensuring that mesh adaptation efficiently captures important flow features. The adaptation workflow in our framework consists of several steps, beginning with the conversion of mesh and solution files into formats compatible with NASA-REFINE. Mesh files are transformed between .mesh and .meshb formats, while solution files are converted to .sol and .solb. These conversions ensure seamless integration into the adaptation process.

In order to run the HEMLAB with NASA-REFINE following parameters needs to be entered in to the adaptation part of the config file as shown in Figure 3.6.

```
&ADAP_OPTS
adap_sizes = (2000 ;  4000; 40000 ; 200000),
adap_norm = 4,
adap_subite = (3 ; 1 ; 3 ; 1),
adap_ext_ite = (200 ; 2000 ; 2000; 2000),
adap_residual_reduction = (3 ; 3 ; 3 ; 3),
adap_sensor = (MACH_DIST; MACH_DIST; MACH; MACH),
adap_hmin = 1e-6,
adap_hmax = 100.,
adap_hgrad = (1.2; 1.2; 1.2; 1.2),
adap_aspect_ratio=  1000,
adap_refine_ite_max = 1,
adap_exit_iteration =3,
adap_spalding = 0.0001,
adap_bl_exit_iteration = 2,
adap_viscous_tags = 1,
bl_creation= true,
delete_files= true,
additional_refine_metric_options= "" ,
additional_refine_adap_options= ""    ,
additional_refine_adap_bl_options= "",
```

Figure 3.6 : Adaptation part of the input file of HEMLAB for REFINE-based mesh adaptation.

The first step in the framework involves generating a metric file, which provides a mathematical description of the desired mesh refinement. The following command was used for this step:

```
mpirun -np 24 refmpi multiscale  mesh_file.meshb
Mach_sensor.solb 8000 output-metric.solb
--norm-power 4 --gradation 1.2   --aspect-ratio 1000
```

Figure 3.7 : Command-line execution of multiscale metric generation using REFINE with 24 MPI processes.

Here, the $--norm-power$ parameter controls the sensitivity of the metric to solution gradients, $--gradation$ specifies the smoothness of mesh transitions, and $--aspect-ratio$ defines the maximum allowable stretching of mesh elements. These parameters are being used to control the generation of a metric tensor, which ensures efficient and accurate refinement of critical flow regions, such as shocks or vortices.

With the metric file generated, the adaptation process begins using the command:

```
mpirun -np 24 refmpi adap mesh_file.meshb -m
output-metric.solb -x mesh_adap.meshb -s 1
```

Figure 3.8 : Execution of REFINE adaptation step using 24 MPI processes with input metric file and output adapted mesh.

In this step, NASA-REFINE modifies the mesh according to the specified metric, creating an adapted mesh (*mesh_adap.meshb*). However, if the original mesh lacks a boundary layer, this step does not create one. Boundary layers are crucial for resolving sharp gradients near walls, such as velocity and temperature changes.

To incorporate a boundary layer, the following command is executed:

```
mpirun -np 24 refmpi adap mesh_adap.meshb  -x
mesh_adap_bl.meshb -s 15.0  --viscous-tags 1
--spalding 0.00001  400000
```

Figure 3.9 : Execution of REFINE boundary layer adaptation.

This step creates a boundary layer using viscous tags and the Spalding boundary-layer thickness model. The $--spalding$ parameter determines the thickness of the layer, and the $--viscous-tags$ option ensures the mesh elements align properly within the layer. The adapted mesh with the boundary layer is output as *mesh_adap_bl.meshb*.

Finally, the adapted mesh must be interpolated to the solution file. This is achieved using the following command:

```
mpirun -np 24 refmpi interpolate
mesh_adap_bl.meshb initial_solution.solb mesh_final.meshb
receptor-field.solb
```

Figure 3.10 : REFINE command for interpolating the solution from adapted mesh to final mesh.

Interpolation transfers the solution from the original mesh to the refined mesh, ensuring continuity of simulation data. The resulting $mesh\_final.meshb$ serves as the starting point for the next adaptation step or the subsequent CFD simulation. These steps

complete the mesh adaptation process. Once the adapted mesh and solution are ready, the HEMLAB solver can be executed to simulate the flow on the refined mesh. This iterative process of adaptation and simulation ensures that computational resources are relatively dense in regions of high interest, accurate and efficient CFD solutions.

4. NUMERICAL RESULTS FOR HLPW-3

As described in Section 1.5.3, there are three main test cases proposed for this workshop. Due to limited computational resources, only two of them were studied during this thesis. Validation of HEMLAB algorithm was performed using the two-dimensional L1T2 (NHLP) geometry which is given in Figure 4.1 and the three-dimensional JAXA Standard Model high-lift configuration cases [87]. Details for each test case are provided in Tables 4.1 and 4.2. In this study, the SU2 solver integrated with the pyAMG library is used as the RANS solver. The final adapted mesh results were also validated using the HEMLAB algorithm.

Figure 4.1 : 2D Multi Element Airfoil L1T2 (NHLP).

Table 4.1 : Flow conditions for 2D Test Case (L1T2/NHLP)

Mach Number	0.197
Re Number	3.52×10^6
Angle of Attacks	$4.01^\circ, 20.18^\circ$
Mean Aerodynamic Chord	$1.0m$
Moment Location	$x = 0.25$ m, $y = 0$ m, $z = 0$ m
Reference Area	$1.0m^2$
Prandtl Number	0.78
Turbulent Prandtl Number	1.0

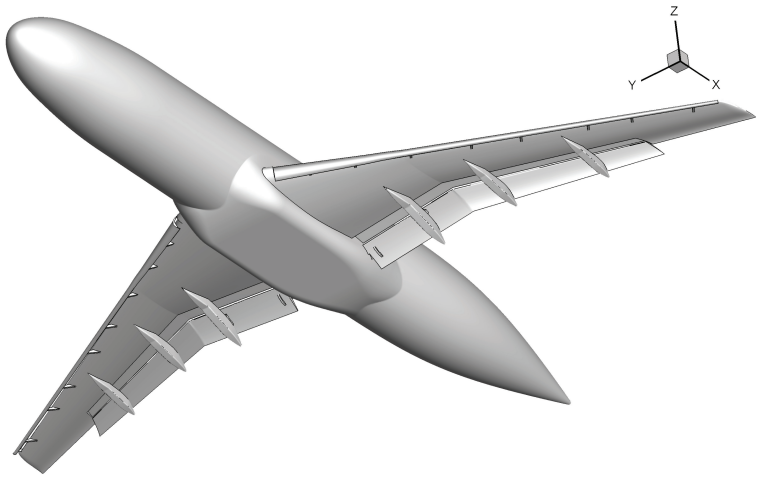


Figure 4.2 : High lift JAXA Standard Model (JSM) (Nacelle/Pylon-Off model, Case 2b).

Table 4.2 : Flow conditions for 3D Test Case (JAXA-SM)

Mach Number	0.172
Re Number	1.93×10^6
Angle of Attacks	$4.36°, 10.47°, 18.58°$
Mean Aerodynamic Chord	$0.5292\,m$
Moment Location	$x = 2.3757$ m, $y = 0$ m, $z = 0$ m
Reference Static Pressure	$747.70\,mmHg$
Reference Static Temperature	$551.79°R$
Reference Area	$2.2466\,m^2$
Prandtl Number	0.78
Turbulent Prandtl Number	1.0

The L1T2 is a three-element airfoil consisting of a main element, a slat (located forward of the main element), and a flap (located aft of the main element). Calculations for the 2D cases were performed at a free-stream Mach number of $M_\infty = 0.197$ and a Reynolds number of $Re = 3.52 \times 10^6$. To validate the numerical setup, simulations were conducted using a well-designed unstructured mesh with a boundary layer. Both the SU2 and HEMLAB solutions agreed well with the experimental data and with each other.

After validating the numerical setup, the adaptation process was studied. It should be noted that the initial mesh used in the simulations did not include a boundary layer. After four levels of refinement using only the distance function, a boundary layer mesh was generated with the pyAMG algorithm. Then, eight additional refinement

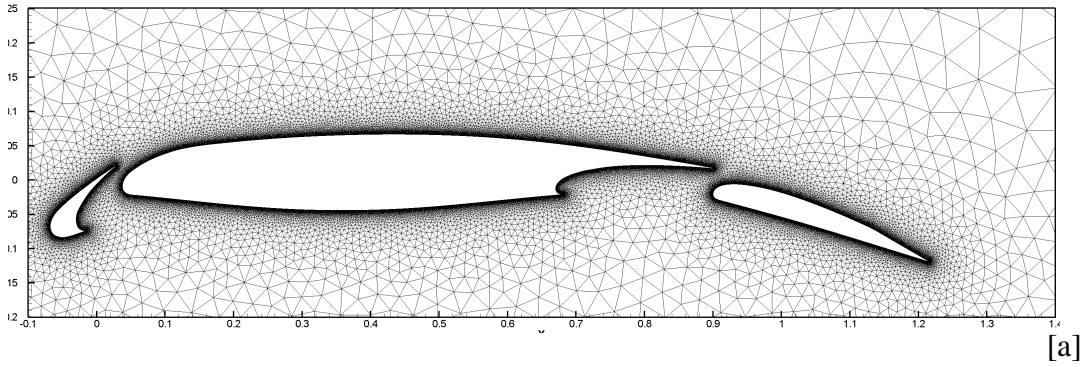
levels were performed using both the distance function and Mach number to achieve an adapted mesh. The development of the wake region is illustrated in Figure 4.3. Results were obtained for two angles of attack, $\alpha = 4.01^\circ$ and $\alpha = 20.18^\circ$. Details of the each refinement level are given in the Table 4.3.

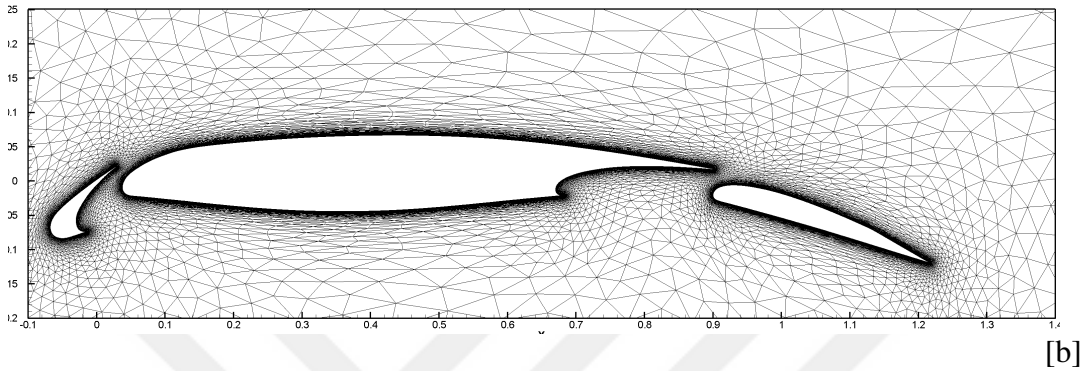
The high mesh resolution on the solid walls of the initial mesh was employed to maintain geometric accuracy. Computed pressure coefficient contours for both solvers are provided in Figures 4.4 and 4.5. A comparison of the computed pressure coefficients on the L1T2 is also presented in Figure 4.6, and the results are in good agreement with each other.

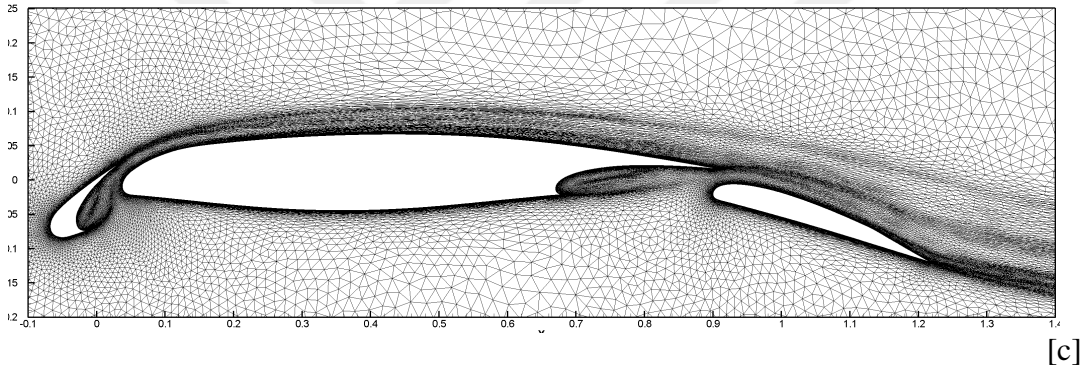
Although there is a slight deviation between the two solvers around the leading-edge suction peak at $\alpha = 4.01^\circ$, the use of rotational correction [64] in the Spalart-Allmaras model (SA-R) eliminates this difference, and the modified model significantly reduces the turbulence viscosity within high-lift cavities. The lift and drag coefficients obtained with SU2 and HEMLAB are tabulated in Table 4.4. It can be seen that the numerical results are compatible for both angles of attack.

Table 4.3 : Refinement data for $\alpha = 4.01^\circ$

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	Number of Surface Elements
ite0 (Initial Mesh)	59,657	113,600	5,662
ite1 (Distance)	107,742	208,409	7,045
ite2 (Distance)	108,909	210,537	7,249
ite3 (Distance)	100,267	193,581	6,922
ite4 (Distance)	87,955	169,531	6,346
ite5 (Distance + Mach)	87,621	170,505	4,533
ite6 (Distance + Mach)	93,404	182,264	4,274
ite7 (Distance + Mach)	97,564	190,575	4,225
ite8 (Distance + Mach)	98,467	192,343	4,235
ite9 (Distance + Mach)	293,080	577,376	8,131
ite10 (Distance + Mach)	295,892	582,730	8,384
ite11 (Distance + Mach)	296,018	582,883	8,481
ite12 (Distance + Mach)	296,247	583,303	8,510


[a]


[b]


[c]

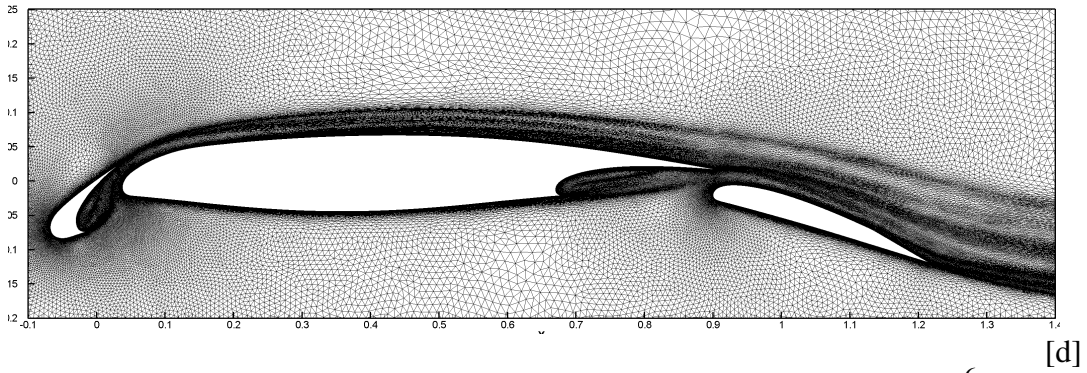

[d]

Figure 4.3 : Progress of mesh adaptation at $M_\infty = 0.197$, $Re = 3.52 \times 10^6$ and $\alpha = 4.01^\circ$ around L1T2 (NHLP) high lift system: [a] Initial mesh with no boundary layer, [b] 4th iteration with distance only, [c] 8th iteration with distance and Mach, [d] 12th iteration with distance and Mach.

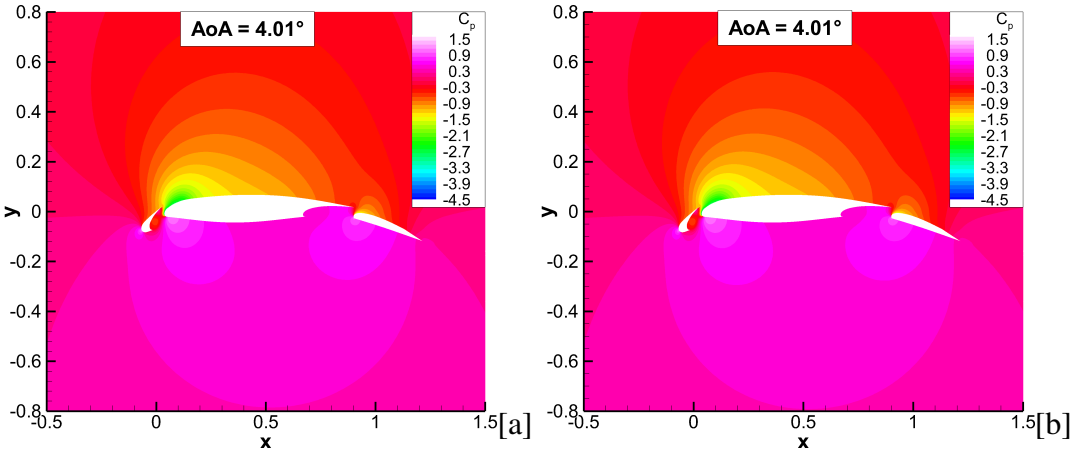


Figure 4.4 : Pressure coefficient contours of adapted mesh with SU2 [a] and HEMLAB [b] at Mach=0.197, Re=3.52 $\times$ 10^6, and $\alpha = 4.01°$.

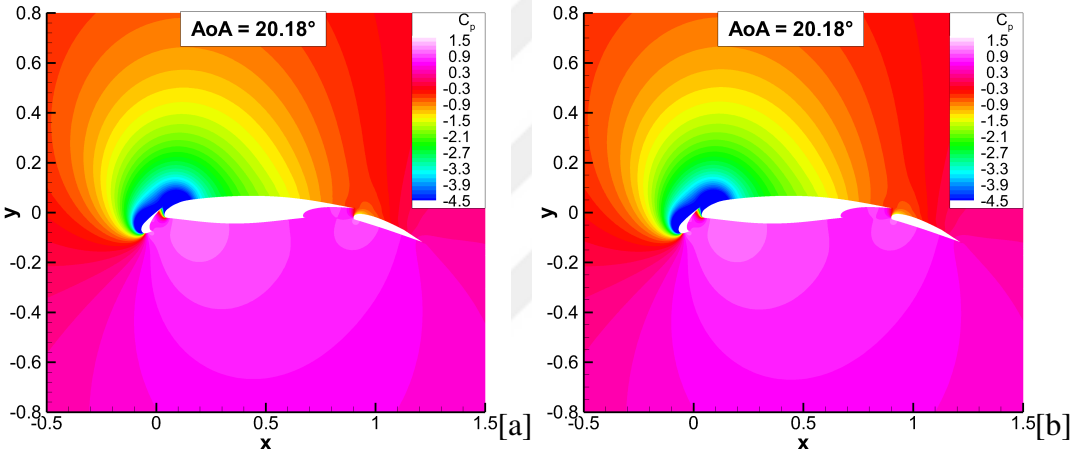


Figure 4.5 : Pressure coefficient contours of adapted mesh with SU2 [a] and HEMLAB [b] at Mach=0.197, Re=3.52 $\times$ 10^6 and $\alpha = 20.18°$.

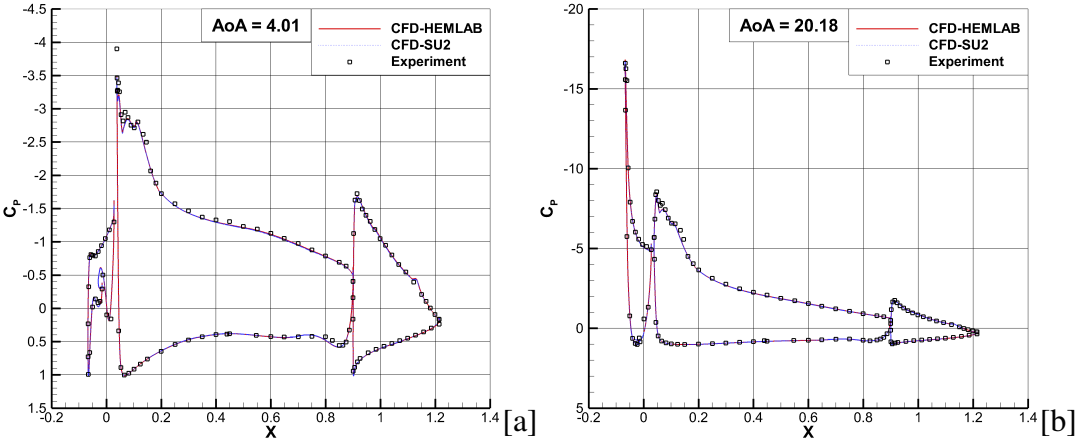


Figure 4.6 : Comparison of computed pressure coefficients with wind tunnel data for $\alpha = 4.01°$ [a] and $\alpha = 20.18°$ [b].

Table 4.4 : CL and CD results for given AoAs

	CL		CD	
AoA	SU2	HEMLAB	SU2	HEMLAB
4.01°	2.0944	2.1053	0.0325	0.0326
20.18°	4.0300	4.0448	0.0851	0.0782

On the other hand, the same calculations were applied to the JAXA Standard Model, which is the geometry proposed for the 3rd AIAA CFD High Lift Prediction Workshop [88]. Figure 4.2 shows the high-lift JAXA Standard Model in the nacelle/pylon-off configuration (Case 2b). The top view of the model, along with approximate wing sections, is presented in Figure 4.7 to facilitate comparison with the experimental data of Yokokawa et al. [89].

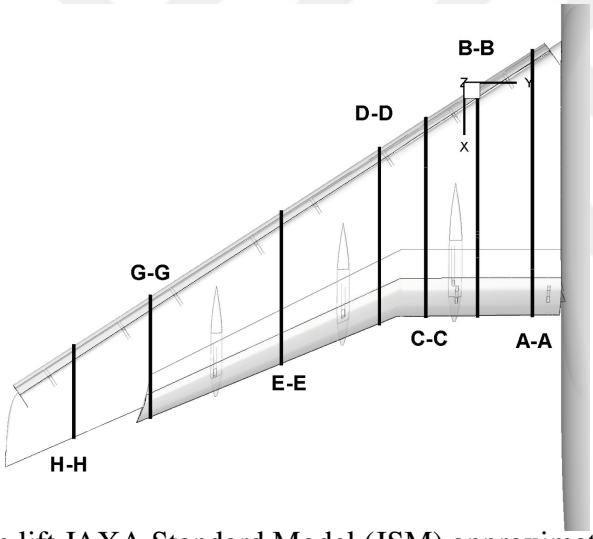


Figure 4.7 : High lift JAXA Standard Model (JSM) approximate wing sections.

The RANS equations can be efficiently solved using Newton's method combined with an accurate Jacobian evaluation, reducing the residual to machine precision [58]. However, this approach has a notable drawback: multiple solutions to the RANS equations may exist—even when the residual is reduced to machine zero [58,90,91]—making it difficult to identify non-physical solutions. These non-physical steady solutions are generally unstable to infinitesimal temporal disturbances and may not occur in URANS simulations. Furthermore, this method is relatively expensive; for example, the convergence of moment data is slow and requires a large number of time steps [58].

In the literature, several authors [91–93] use an alpha continuation approach, wherein the solution is restarted from a previously converged solution at a slightly lower angle of attack rather than starting impulsively with uniform free-stream conditions. Others [91] use time-accurate transient simulations. Both approaches delay premature flow separation and result in flow predictions and lift computations that better agree with experimental data.

In the present study, the SU2 calculations are carried out using the Euler implicit method in time with a constant CFL of 2 and $\Delta t = 1$; larger CFL numbers lead to divergence. Additionally, the gradient terms required for the inviscid/viscous fluxes and turbulence source terms are evaluated with the Green-Gauss approach, since the weighted least squares approximation for the turbulence source terms is highly inaccurate and leads to divergence on anisotropic tetrahedral meshes.

Similarly, the HEMLAB calculations are performed using the Euler implicit method with $\Delta t = 1$; however, the CFL number is exponentially ramped to 100 to accelerate the numerical calculations. The larger CFL number with initial free-stream values also results in premature separation, so the solutions are restarted from a previously converged solution at a lower angle of attack. The sensor function for anisotropic mesh refinement is similar to that used for two-dimensional cases and is arranged as follows:

$$Sensor = \beta\, Mach + \sum_{i=1}^{4} 0.25\, e^{-100\times 10^{i} d} \tag{4.1}$$

where d is the minimum wall distance function. The value of β is zero for the initial four adaptation levels and is set to 1 thereafter. Using several exponential functions allows us to account for different initial mesh sizes on solid walls. The minimum edge length is set to 1×10^{-6} with an exponential size variation factor of 1.2.

The initial calculation was carried out at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$, and $\alpha = 4.36^\circ$. In the present simulations, the free-stream turbulent viscosity for the classical Spalart-Allmaras turbulence model was set to 0.05, as in the two-dimensional cases. The progress of the mesh adaptation is shown in Figure 4.8 on the $y = -0.66$ plane. Although the geometry is relatively complex, the boundary layers and wakes are captured well. Mesh resolutions for each refinement level are provided in Table 4.5. The final adapted mesh consists of 29,174,966 vertices and 169,747,550 tetrahedral elements (175,049,796 DOF). In these simulations, the full JSM model was used

instead of the half-span JSM model, and there are 2,805,424 triangular elements on the solid walls.

The computed pressure coefficients are presented in Figure 4.9. The results show relatively good agreement on the main wing section; however, there is some discrepancy in the outboard H-H section for the HEMLAB solver. This discrepancy is attributed to the separation that occurs behind the second-most outboard slat bracket. It should also be noted that the pressure coefficients along the stagnation lines are lower than one compared to the two-dimensional cases, and the leading part of the main wing experiences a less favorable pressure gradient.

In Figure 4.9, the experimental wing sections do not exactly correspond to constant y-planes. Consequently, there is some difference on the slats. Although it is possible to interpolate the pressure values at the experimental points using NASA's open source *adt_utilities.f90* module, the number of experimental points is limited. We observed better agreement with this approximation since some sections are very close to the brackets (see section D-D, for example).

Subsequently, the angle of attack was increased to $\alpha = 10.47^\circ$ and $\alpha = 18.58^\circ$, and the computed pressure coefficients for these cases are provided in Figures 4.10 and 4.11, respectively. The agreement is relatively good for $\alpha = 10.47^\circ$, but for $\alpha = 18.58^\circ$ very large separation is observed in the outboard portion of the main wing.

There is no certain explanation about the source of this discrepancy. One possibility is poor mesh resolution within the boundary layers; another possibility is inappropriate wake behavior in the turbulence model.

The computed skin friction lines for both the HEMLAB and SU2 solutions are shown in Figures 4.12 and 4.13 which are also compared with experimental data. The skin friction lines indicate small separation at the flap tips, where the flow tends to move toward the outboard wing tips at low angles of attack. At higher angles of attack, significant separation is observed in the outboard portion of the main wing.

The complete skin friction lines around the JSM model are presented in Figure 4.14, where separation lines can be observed on the JAXA main body at high angles of attack. The constant entropy isosurface is shown in Figure 4.15 for $\alpha = 4.36^\circ$, along

with a mesh slice at the $x = 5$ plane, to demonstrate the wake resolution. The roll-up behavior of the wake is clearly visible.

At high angles of attack, the wing-body junction vortex merges with the vortex from the inboard slat tip to form a relatively stronger vortex. At the outboard wing tips, the vortex from the outboard slat tip rolls around the main wing tip vortex.

The present calculations indicate very good accuracy near stall angles for the inboard sections, including the separation bubble at the wing-body junction; however, further improvement is required for the outboard sections.

To assess the required normal mesh resolution, the y^+ values for $\alpha = 4.36^\circ$ are presented in Figure 4.16. The maximum y^+ value is computed to be less than 5, with the highest values occurring near the main wing tips.

It is also observed that using the SA model without the transition model (SA-noft2) does not result in a fully turbulent simulation at the current free-stream turbulence level [94]. Even when the free-stream turbulent viscosity is increased to 3, no significant difference in the separation lines on the flap surfaces is observed. Nevertheless, the computed lift, drag, and moment coefficients — calculated with respect to the moment location as defined in 4.2 — are 1.6860, 0.1657, and -0.5260, respectively. The lift coefficient shows better agreement with the experimental value.

Table 4.5 : Refinement data for AoA=4.36°

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	Number of Surface Elements
ite0 (Initial Mesh)	16,591,604	98,181,146	896,704
ite1 (Distance)	13,691,577	79,581,271	1,435,956
ite2 (Distance)	15,157,646	88,126,699	1,548,066
ite3 (Distance)	16,345,926	95,104,769	1,621,898
ite4 (Distance)	12,600,773	73,338,581	1,284,372
ite5 (Distance + Mach)	13,657,242	79,376,223	1,436,600
ite6 (Distance + Mach)	15,133,152	87,975,397	1,551,246
ite7 (Distance + Mach)	16,309,510	94,876,430	1,626,188
ite8 (Distance + Mach)	17,164,661	99,897,529	1,679,616
ite9 (Distance + Mach)	22,844,559	132,861,764	2,196,918
ite10 (Distance + Mach)	25,833,893	150,171,226	2,525,996
ite11 (Distance + Mach)	27,820,608	161,770,118	2,705,578
ite12 (Distance + Mach)	29,174,966	169,747,550	2,805,424

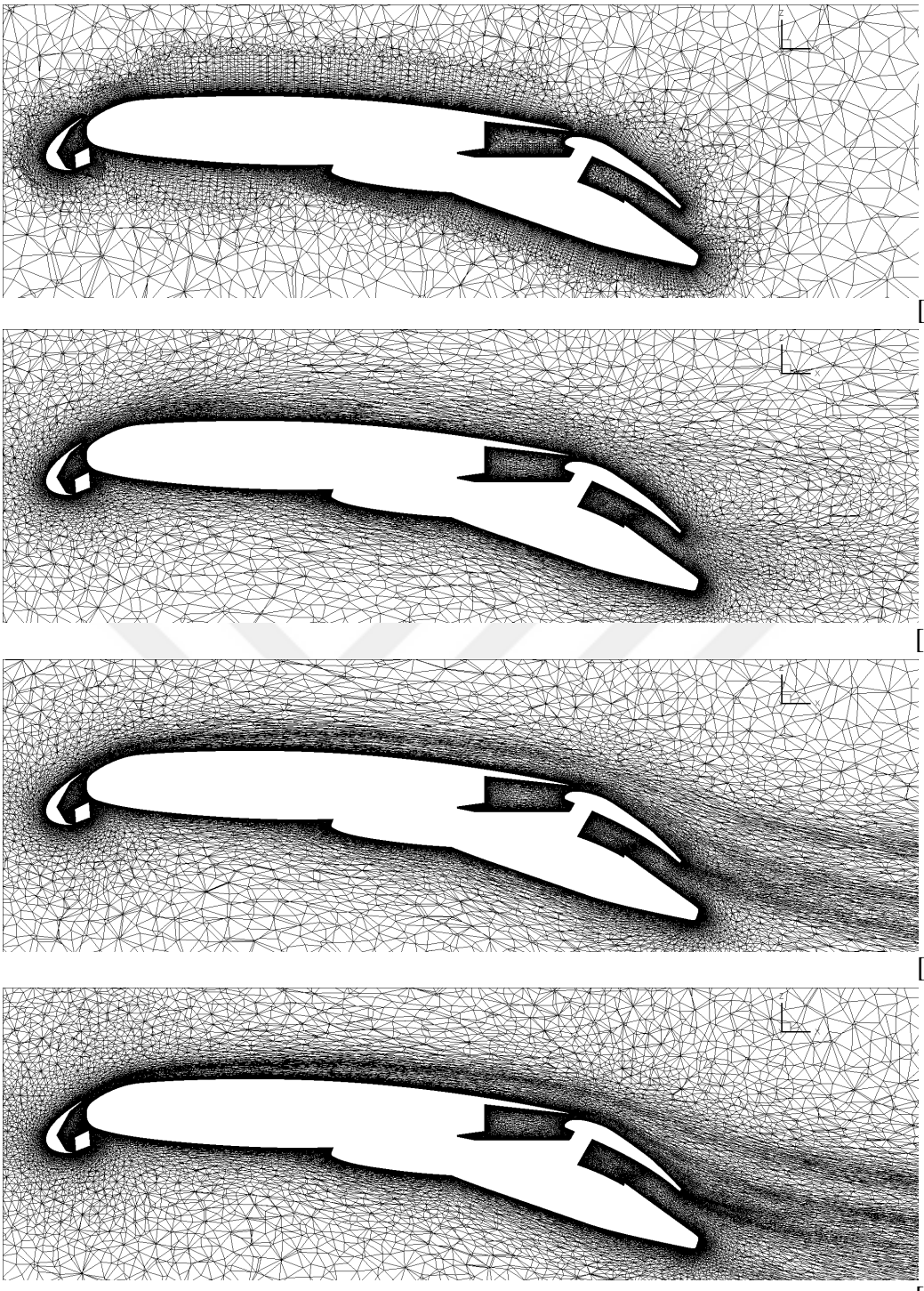


Figure 4.8 : Progress of mesh adaptation at $y = -0.66$ plane at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$ and $\alpha = 4.36^\circ$ around the JAXA high lift system: Initial mesh with boundary layer [a], 4th iteration with distance only [b], 8th iteration with distance and Mach [c], and 12th iteration with distance and Mach [d].

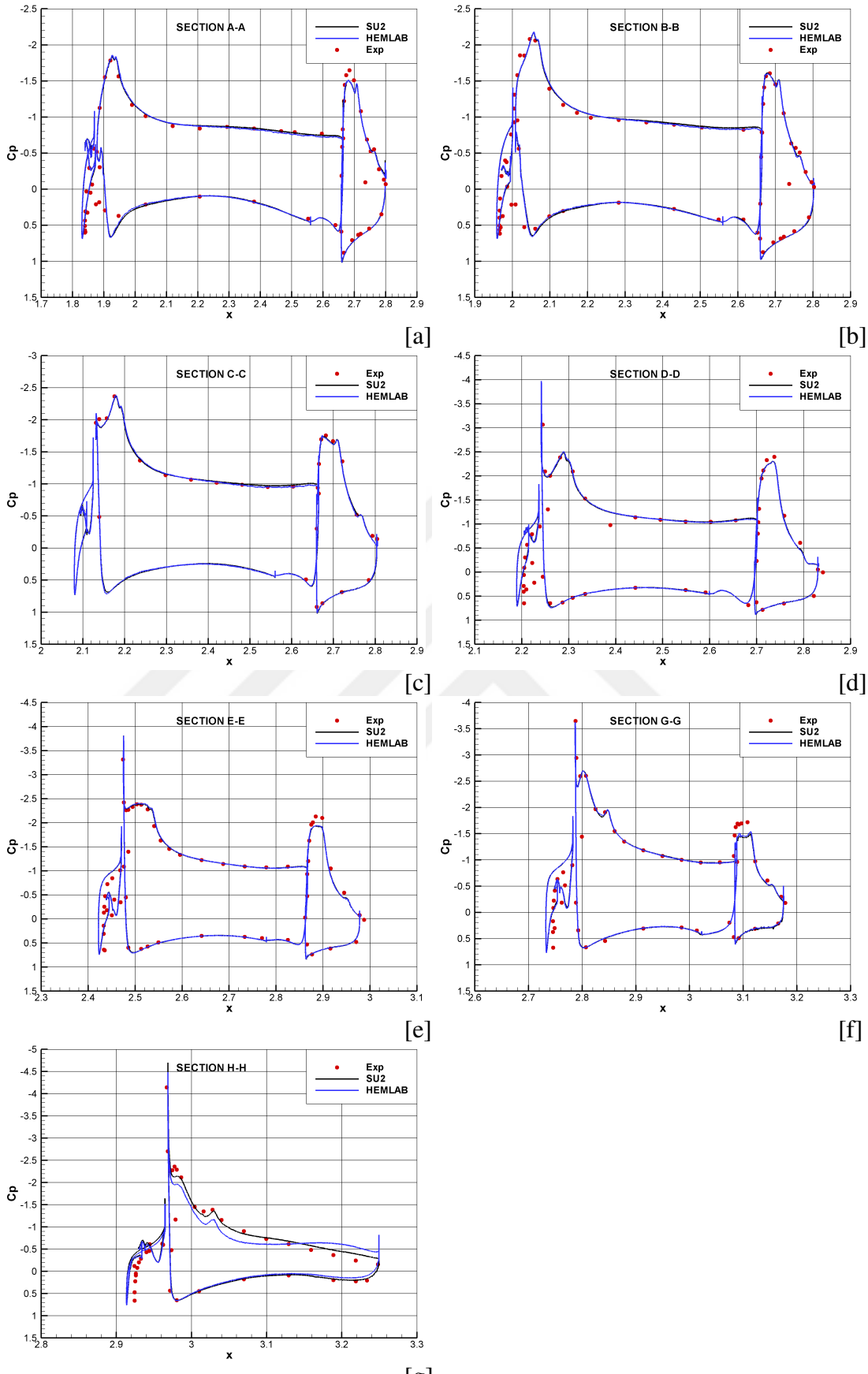


Figure 4.9 : Comparison of pressure coefficients at sections AA [a], BB [b], CC [c], DD [d], EE [e], GG [f] and HH [g] at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$ and $\alpha = 4.36°$ around the JAXA high lift system.

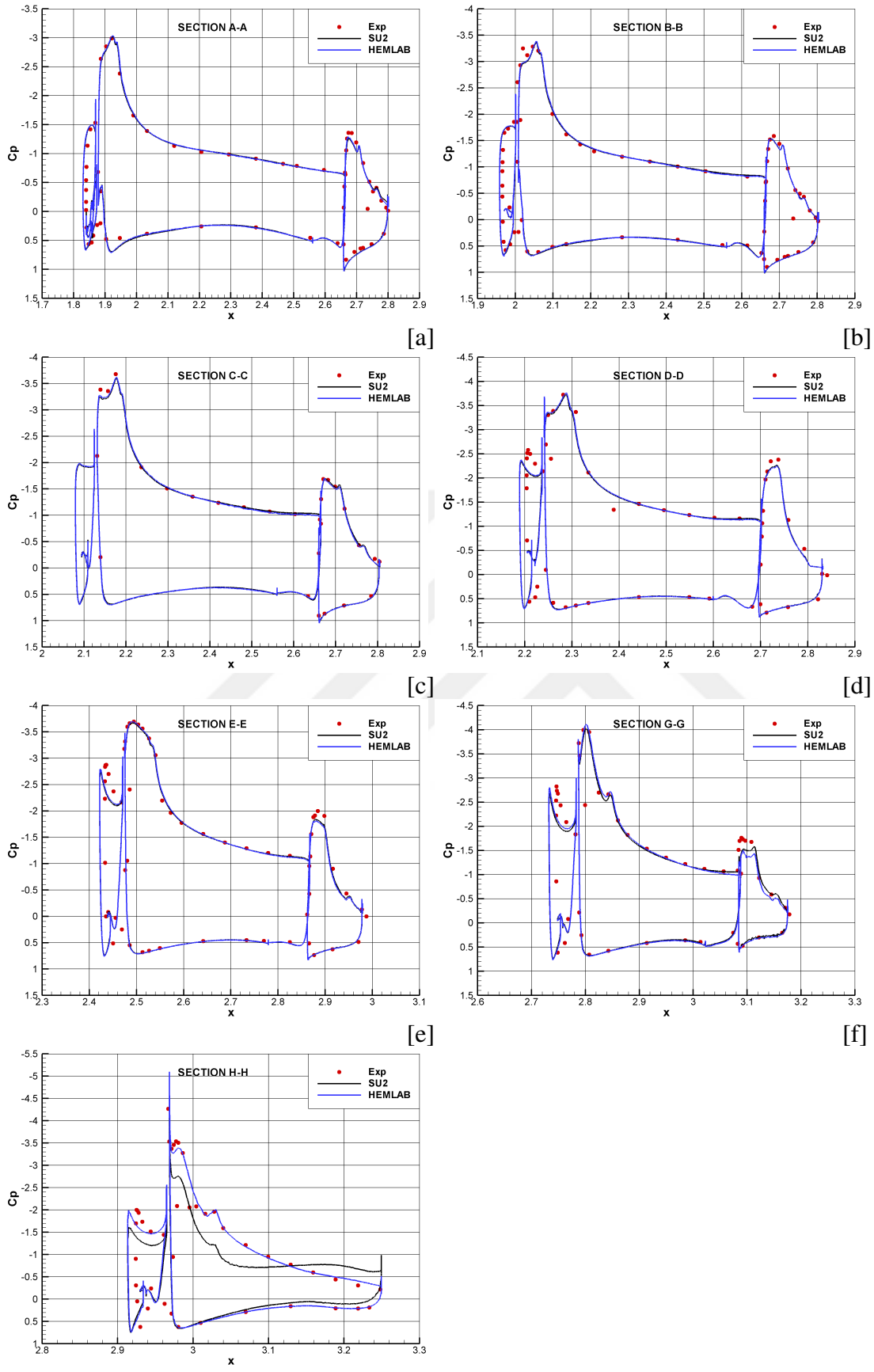


Figure 4.10 : Comparison of pressure coefficients at sections AA [a], BB [b], CC [c], DD [d], EE [e], GG [f] and HH [g] at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$ and $\alpha = 10.47°$ around the JAXA high lift system.

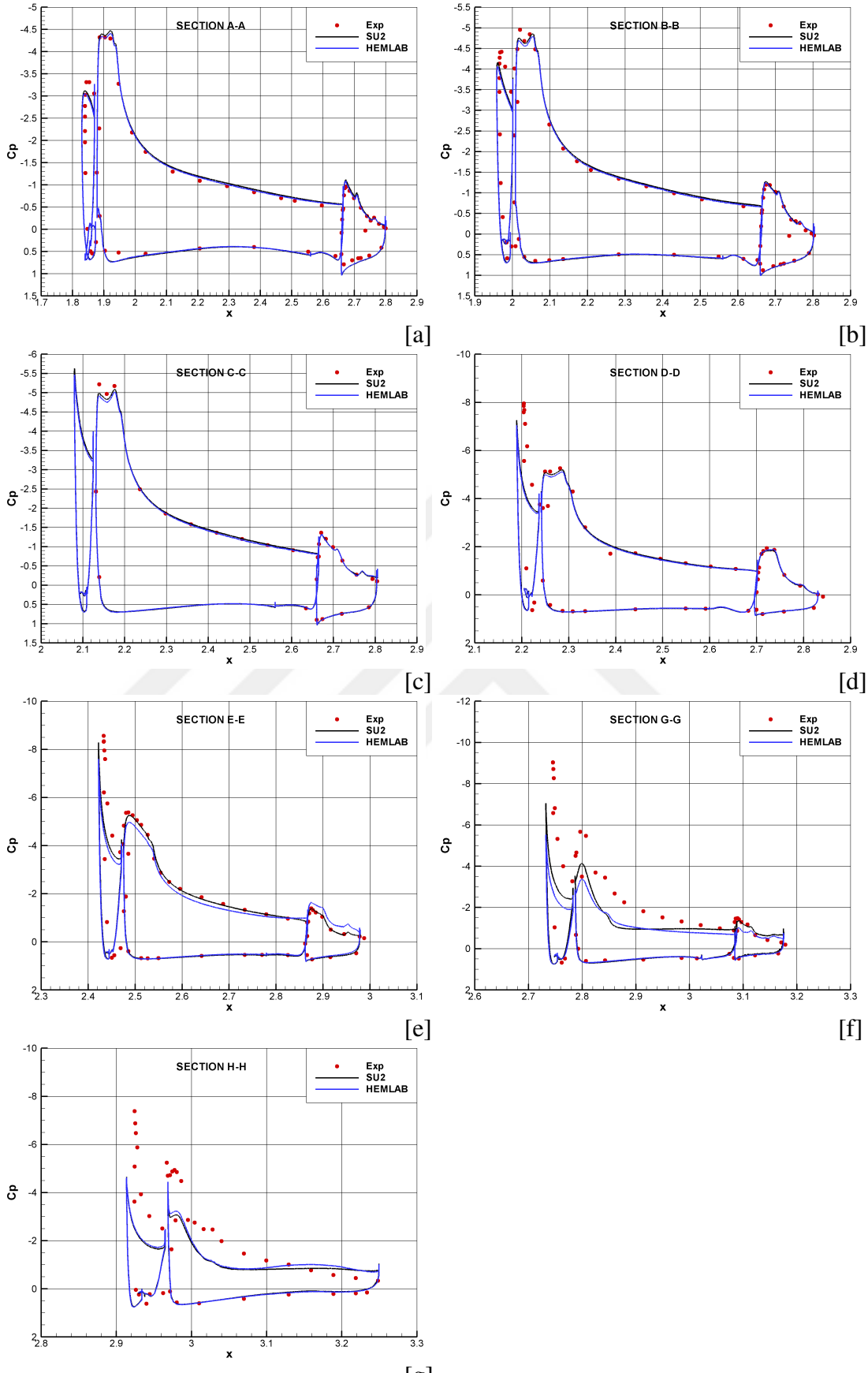


Figure 4.11 : Comparison of pressure coefficients at sections AA [a], BB [b], CC [c], DD [d], EE [e], GG [f] and HH [g] at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$ and $\alpha = 18.58°$ around the JAXA high lift system.

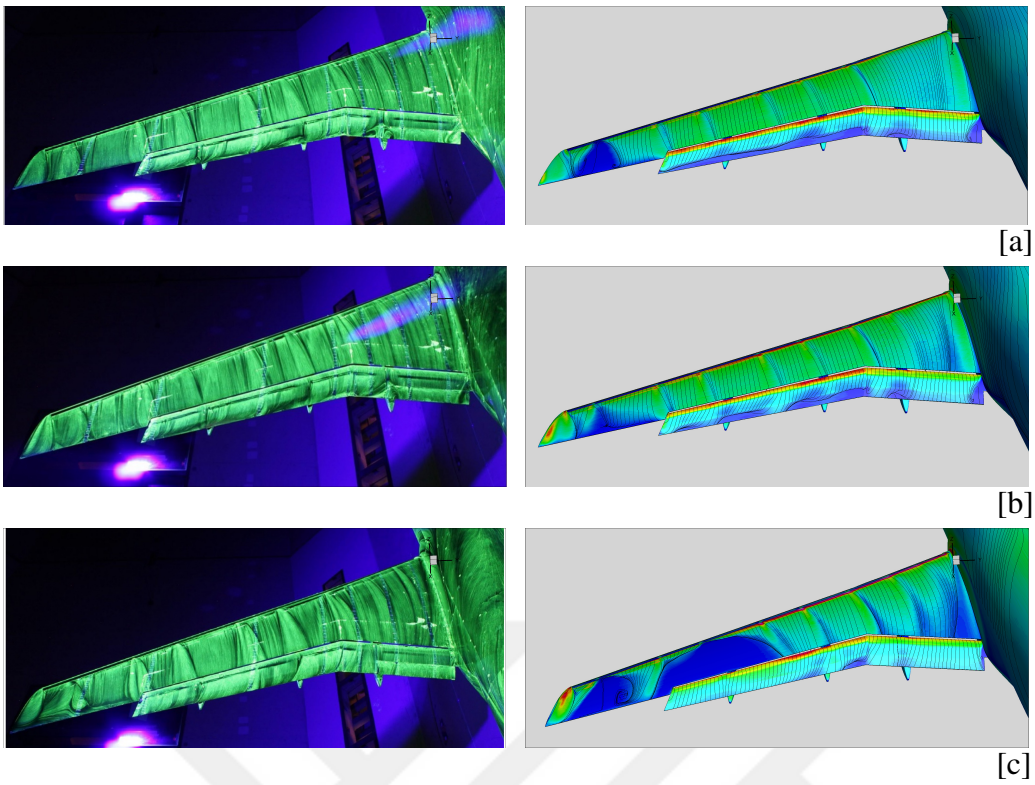

[a]
[b]
[c]

Figure 4.12 : Comparison of computed HEMLAB streamtraces with experiments at $M_\infty = 0.172$ and $Re = 1.93 \times 10^6$ for $\alpha = 4.36°$ [a], $\alpha = 10.47°$ [b] and $\alpha = 18.58°$ [c]. The contours correspond to C_{fx} values ($0 < C_{fx} < 0.020281$).

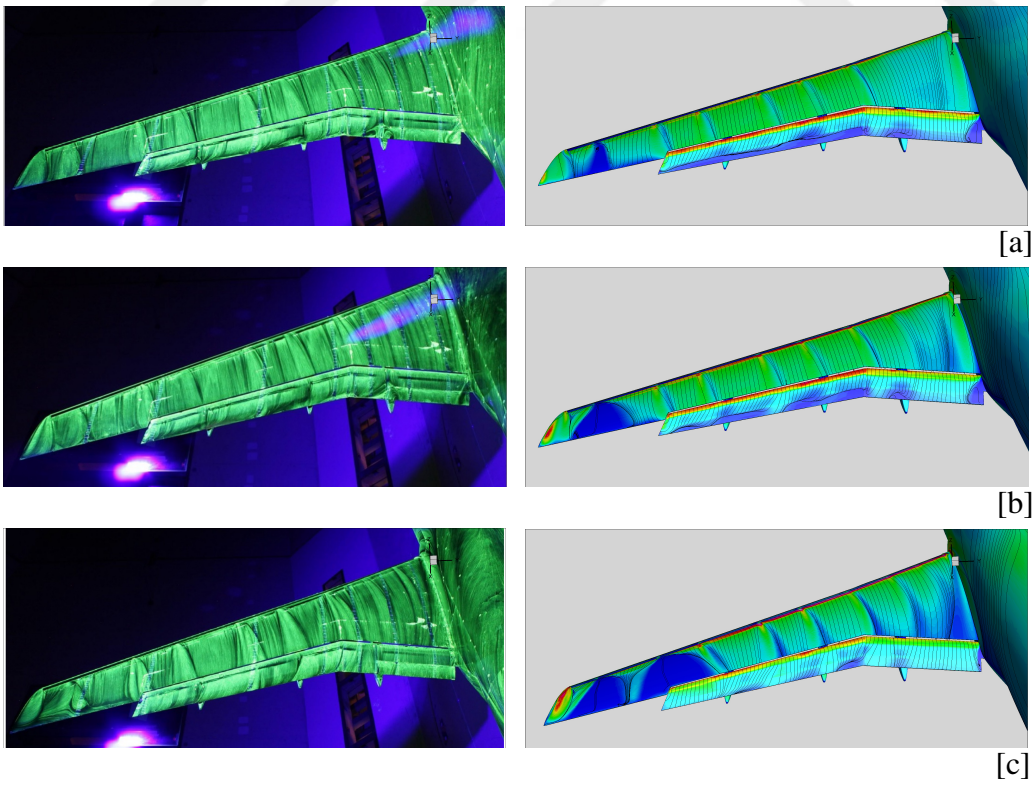
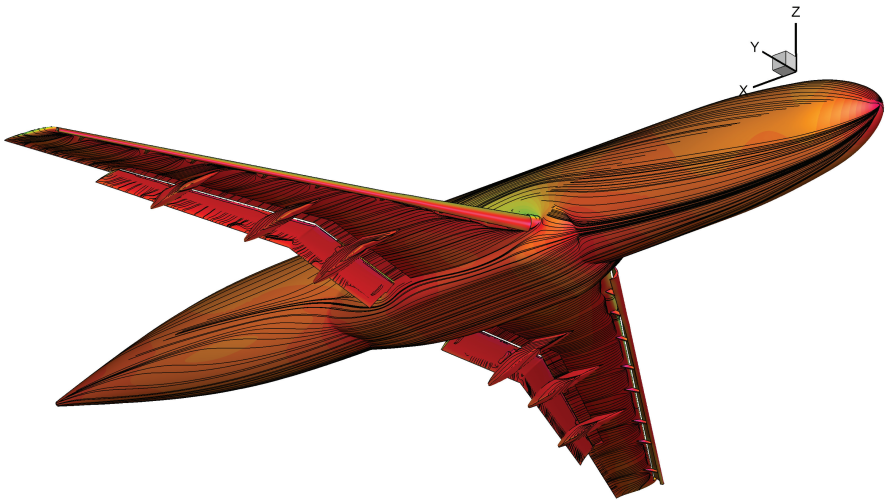
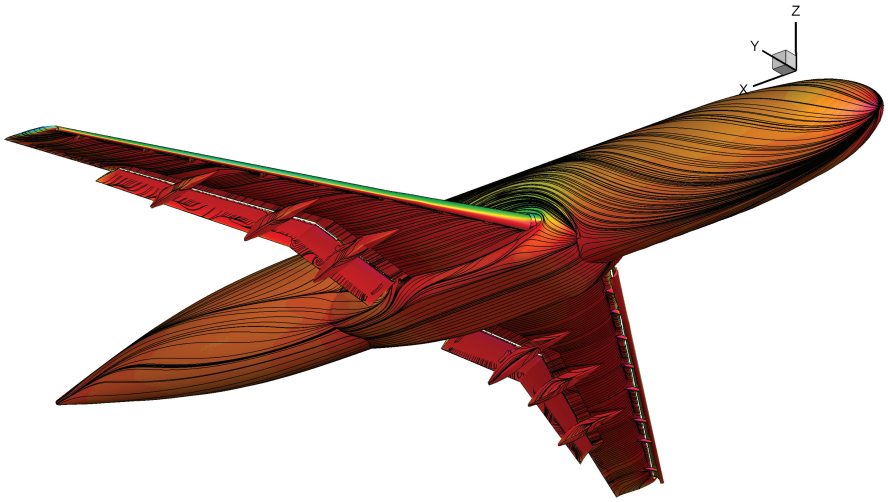

[a]
[b]
[c]

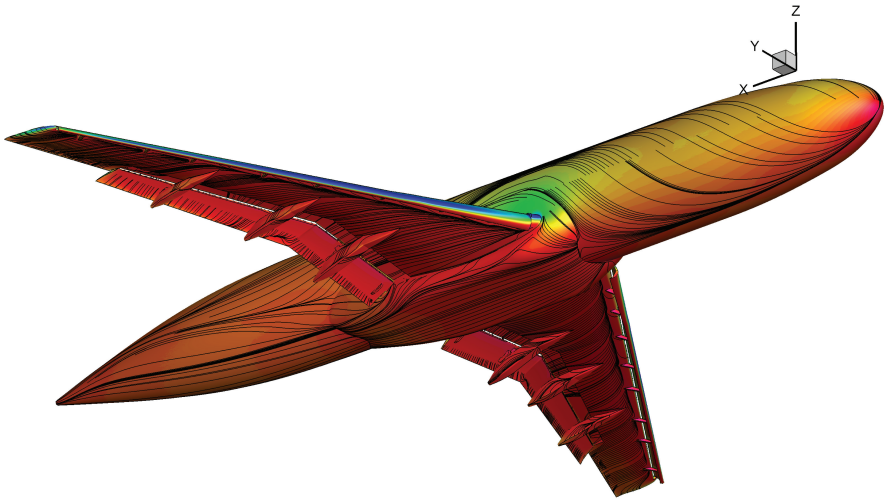
Figure 4.13 : Comparison of computed SU2 streamtraces with experiments at $M_\infty = 0.172$ and $Re = 1.93 \times 10^6$ for $\alpha = 4.36°$ [a], $\alpha = 10.47°$ [b], and $\alpha = 18.58°$ [c]. The contours correspond to C_{fx} values ($0 < C_{fx} < 0.020281$).



[a]



[b]



[c]

Figure 4.14 : Effect of AoA on computed streamtraces at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$ around the JAXA high lift system (HEMLAB) for $\alpha = 4.36^\circ$ [a], $\alpha = 10.47^\circ$ [b] and $\alpha = 18.58^\circ$ [c]. The contours correspond to C_p values $(-4.3 < C_p < 2.9)$.

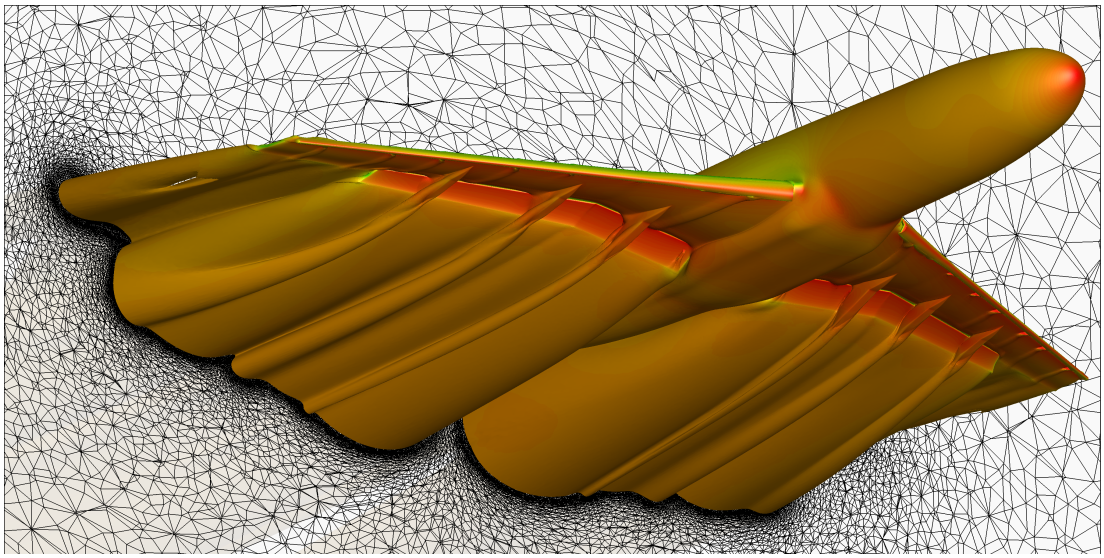


Figure 4.15 : Computed entropy isosurface ($\Delta s = 0.001$) with a slice at $x = 5$ plane at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$ and $\alpha = 4.36°$ around the JAXA high lift system (HEMLAB).

Table 4.6 : CL, CD and CM results for given AoAs

	CL			CD			CM		
AoA	Exp	SU2	HEMLAB	Exp	SU2	HEMLAB	Exp	SU2	HEMLAB
4.36°	1.6818	1.6899	1.6860	0.1563	0.1691	0.1657	-0.5496	-0.5343	-0.5260
10.47°	2.23268	2.19666	2.2301	0.22592	0.25607	0.2417	-0.46323	-0.42042	-0.4464
18.58°	2.74305	2.62894	2.5088	0.38710	0.40823	0.4029	-0.23318	-0.17097	-0.1321

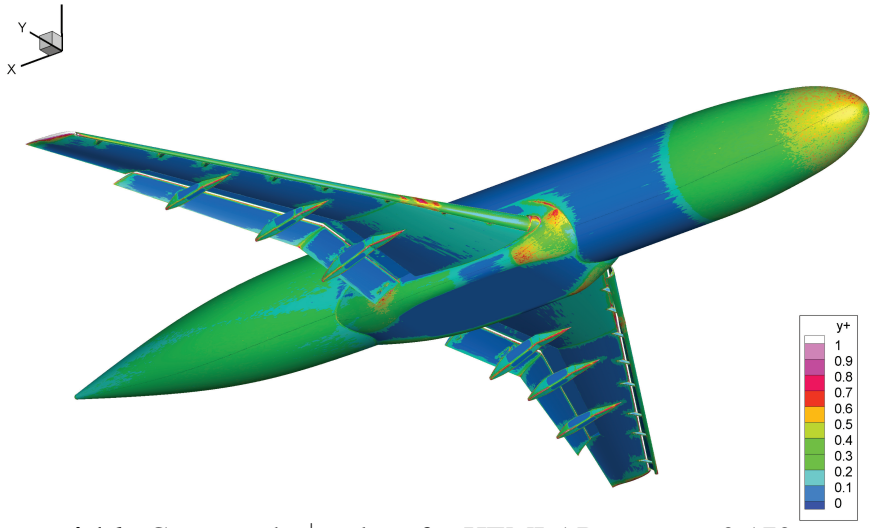


Figure 4.16 : Computed y^+ values for HEMLAB at $M_\infty = 0.172$, $Re = 1.93 \times 10^6$ and $\alpha = 4.36°$ around the JAXA high lift system.

5. NUMERICAL RESULTS FOR HLPW-4

In this workshop, a turbulence verification study was conducted for the 2D-CRM model as shown in 5.1. Figure 5.2 shows the 3D-CRM model was used for the other test cases as detailed Section in 1.5.4. The parameters used in these test cases are provided in Tables 5.1 and 5.2. In performing the simulations, HEMLAB was integrated with pyAMG which is an anisotropic mesh adaptation library so SU2 was not used as the flow solver as in the HLPW-3 studies. Another change from the HLPW-3 studies is in the turbulence modeling. The updated code uses the SA-negative model with second-order discretization. All calculations were performed by solving the turbulence model equations coupled with the Navier-Stokes equations.

Figure 5.1 : 2D-CRM Multi Element Airfoil.

Table 5.1 : Flow conditions for 2D Test Case (2D-CRM)

Mach Number	0.2
Re Number	5.0×10^6
Angle of Attacks	16°
Mean Aerodynamic Chord	$1.0\,m$
Moment Location	$x = 0.25$ m, $y = 0$ m, $z = 0$ m
Ratio of Specific Heats γ	1.4
Reference Static Temperature	$272.1^\circ K$
Prandtl Number	0.72
Turbulent Prandtl Number	0.9
Reference Static Pressure	$14.7\,psi$
$\hat{\nu}_{freestream}/\nu_\infty$	3.0

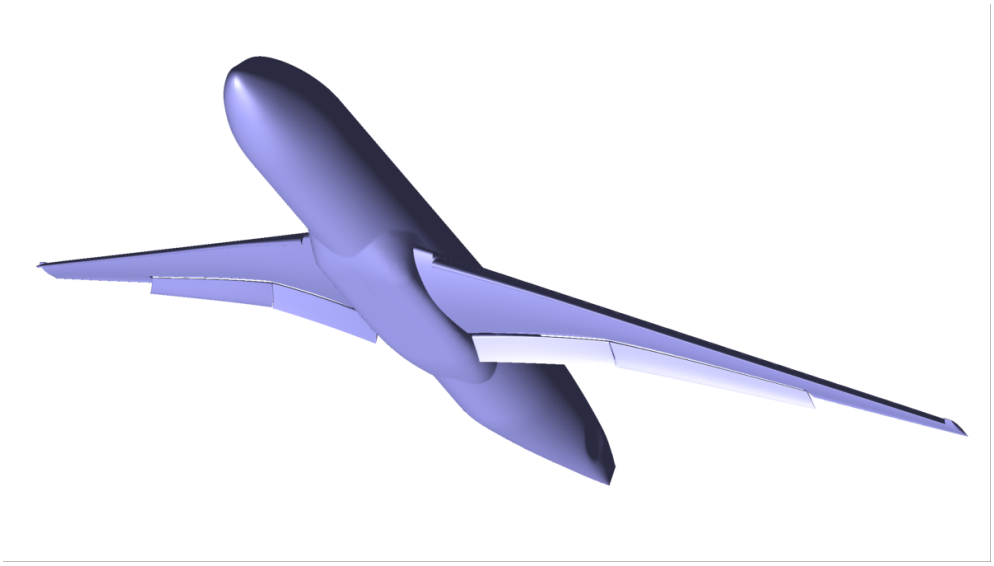


Figure 5.2 : High lift NASA Common Research Model (CRM).

Table 5.2 : Flow conditions for 3D Test Case (3DCRM)

Mach Number	0.2
Re Number	5.49×10^6
Angle of Attacks	7.05°
Mean Aerodynamic Chord	275.8 inch
Reference Static Pressure	$24.67\,psi$
Reference Static Temperature	$521.0^\circ R$
Semi-Span Reference Area	$29736.0\,in^2$
Moment Reference Center	$x = 1325.9\,inches,\ y = 0.0\,inches,\ z = 177.95\,inches$

The initial mesh has no boundary layer, and the flow conditions are $M_\infty = 0.20$, $Re = 5 \times 10^6$, $Pr = 0.72$, $\alpha = 16^\circ$, and $T_\infty = 272.1\,\mathrm{K}$ was the 2D-CRM calculations. The boundary layer is generated during the mesh adaptation process. The computational domain is set to 1000 chord lengths of the airfoil. Time-dependent simulations were run with a maximum CFL number of 1000.

$$Mach\ Sensor = \beta Mach + \sum_{i=1}^{4} 0.25\ e^{-100x10^i d} \tag{5.1}$$

For the mesh adaptation, a Mach sensor was used. In the first four iterations, only the distance function was employed ($\beta = 0$) to obtain the boundary layer; subsequently, the remaining steps were carried out using the full Mach sensor ($\beta = 1$). In the end, the results captured the wake region and the pressure distribution correctly, as shown in Figure 5.3.

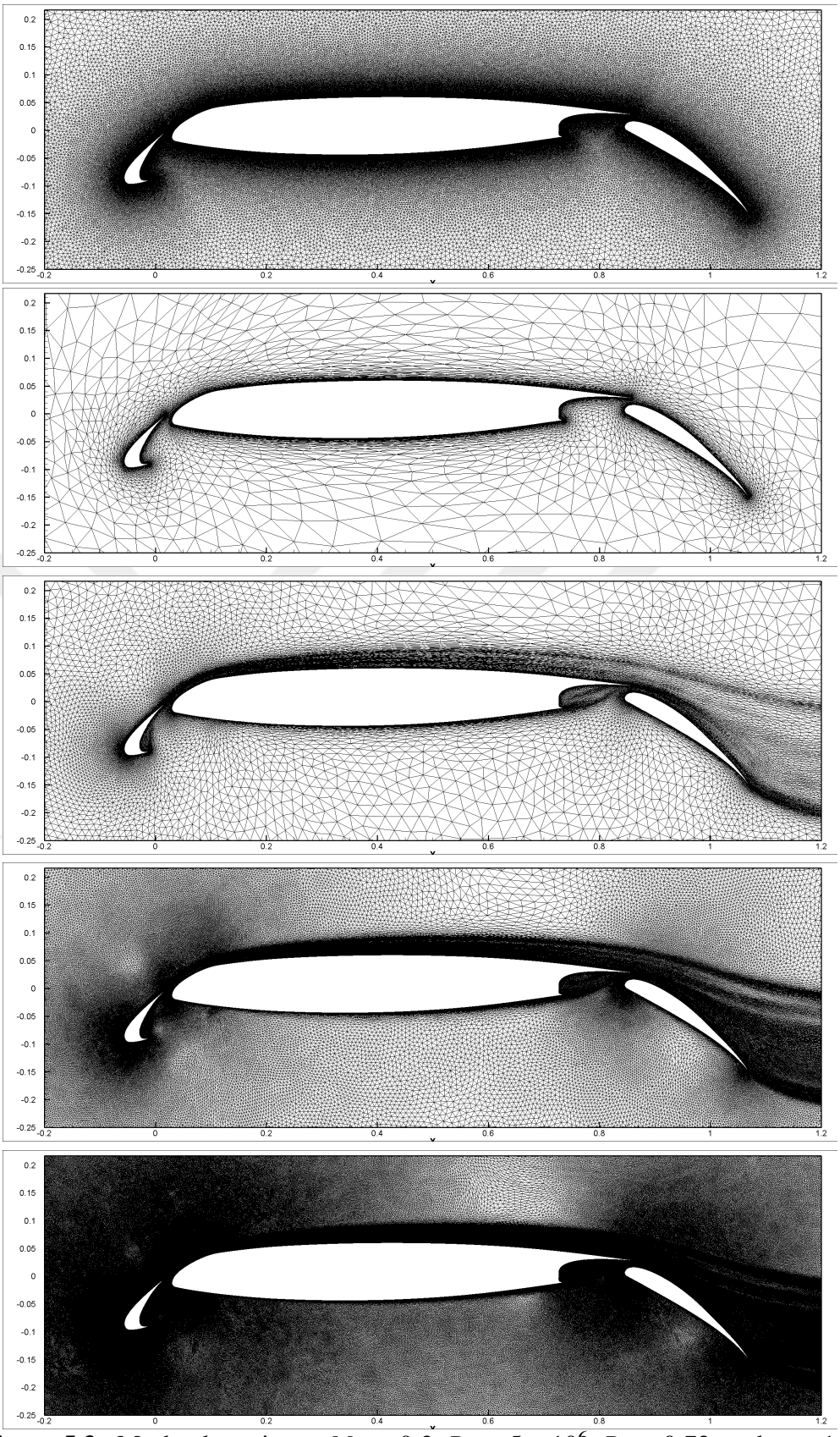
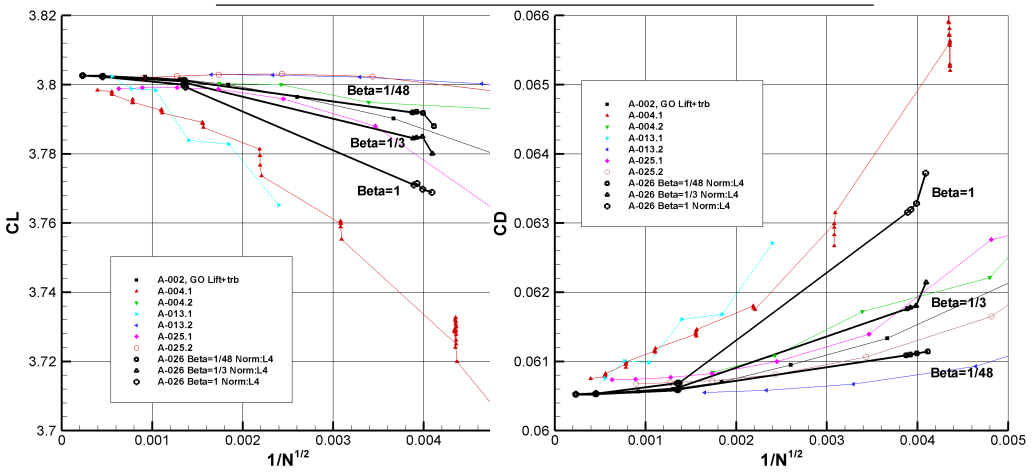


Figure 5.3 : Mesh adaptation at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16$[e] around 2D-CRM: Initial mesh with no boundary layer [a], 4th [b], 8th [c], 12th [d], and 16th [e] iterations with distance and Mach.

The lift and drag coefficients obtained are compared with the results presented in the AIAA 4th High Lift Prediction Workshop. The label A-026 represents the results obtained with the new version of the HEMLAB algorithm, while the other labels correspond to A-002 (INRIA), A-004 (NASA), A-013 (MIT), A-025 (Boeing), and A-031 (UT). The β value, which serves as a blending parameter between upwind and central interpolations ($\beta = 0$ representing a central approximation and $\beta = 1$ corresponding to upwind least-squares interpolation), is shown in Figure 5.4. This parameter governs the degree of upwinding and has a significant impact on convergence. Details of the grid elements for each mesh adaptation level, along with the corresponding lift, drag, and moment coefficients — calculated with respect to the moment location given in Table 5.1 — are provided in Table 5.3.

Table 5.3 : Refinement data for L4, $\alpha = 16^\circ$ and $\beta = 1/48$.

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	Number of Surface Elements	C_L	C_D	C_M
ite0 (Initial Mesh)	325,794	136,785,487	9,156	-	-	-
ite1 (Distance)	80,517	153,908	7,051	-	-	-
ite2 (Distance)	75,179	144,383	5,900	-	-	-
ite3 (Distance)	69,850	134,398	5,227	-	-	-
ite4 (Distance)	66,769	127,951	5,512	-	-	-
ite5 (Mach)	59,075	114,701	3,374	3.787990	0.061144	-0.380541
ite6 (Mach)	62,788	122,495	3,001	3.791839	0.061114	-0.381260
ite7 (Mach)	65,084	127,163	2,918	3.792202	0.061097	-0.381325
ite8 (Mach)	66,556	130,135	2,879	3.791986	0.061087	-0.381277
ite9 (Mach)	535,817	1,063,137	8,272	3.801085	0.060595	-0.382779
ite10 (Mach)	545,380	1,081,172	9,347	3.801261	0.060581	-0.382805
ite11 (Mach)	546,086	1,082,540	9,394	3.801343	0.060581	-0.382823
ite12 (Mach)	545,482	1,081,333	9,398	3.801392	0.060581	-0.382831
ite13 (Mach)	4,931,609	9,834,558	28,020	3.802472	0.060527	-0.383015
ite14 (Mach)	4,902,898	9,776,620	28,523	3.802529	0.060524	-0.383026
ite15 (Mach)	4,880,518	9,731,785	28,587	3.802536	0.060524	-0.383027
ite16 (Mach)	4,865,985	9,702,708	28,601	3.802536	0.060524	-0.383027
ite17 (Mach)	18,734,411	37,404,255	62,930	3.802598	0.060521	-0.383037
Ref [95]				3.802952	0.060529	-



[a] [b]

Figure 5.4 : Convergence of lift coefficient [a] and drag coefficient [b] at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ around a two-dimensional high-lift system compared with several other workshop benchmark data, N corresponds to the total number of vertices.

Table 5.4 : Effect of numerical methods on aerodynamic loads for ite8 mesh (66,556 nodes).

Inviscid fluxes	β	(K_v)	Harten (ε)	C_L	C_D	C_M
Roe second-order	1	$K_v = 0.01$	$\varepsilon = 0.001$	3.740486	0.065880	-0.373570
Roe second-order	1	$K_v = 0.1$	$\varepsilon = 0.001$	3.769412	0.063391	-0.377939
Roe second-order	1	$K_v = 1$	$\varepsilon = 0.001$	3.771079	0.063262	-0.378203
Roe second-order	1	no limiter	$\varepsilon = 0.1$	3.749120	0.064479	-0.374499
Roe second-order	1	no limiter	$\varepsilon = 0.01$	3.769094	0.063367	-0.377864
Roe second-order	1	no limiter	$\varepsilon = 0.001$	3.771098	0.063260	-0.378206
Roe second-order	1	no limiter	$\varepsilon = 0.0$	3.771222	0.063254	-0.378227
Roe second-order	2/3	no limiter	$\varepsilon = 0.001$	3.777978	0.062528	-0.379212
Roe second-order	1/2	no limiter	$\varepsilon = 0.001$	3.781486	0.062159	-0.379728
Roe second-order	1/3	no limiter	$\varepsilon = 0.001$	3.785052	0.061789	-0.380254
Roe second-order	1/6	no limiter	$\varepsilon = 0.001$	3.788678	0.061415	-0.380790
Roe second-order	1/12	no limiter	$\varepsilon = 0.001$	3.790539	0.061226	-0.381066
Roe second-order	1/48	no limiter	$\varepsilon = 0.001$	3.791986	0.061087	-0.381277
HLLC second-order	1	no limiter	-	3.771212	0.063255	-0.378225
HLLC second-order	1/48	no limiter	-	3.791989	0.061087	-0.381277
AUSM$^+$-up second-order	1	no limiter	-	3.773094	0.063263	-0.378618
AUSM$^+$-up second-order	1/48	no limiter	-	3.791942	0.061260	-0.381356

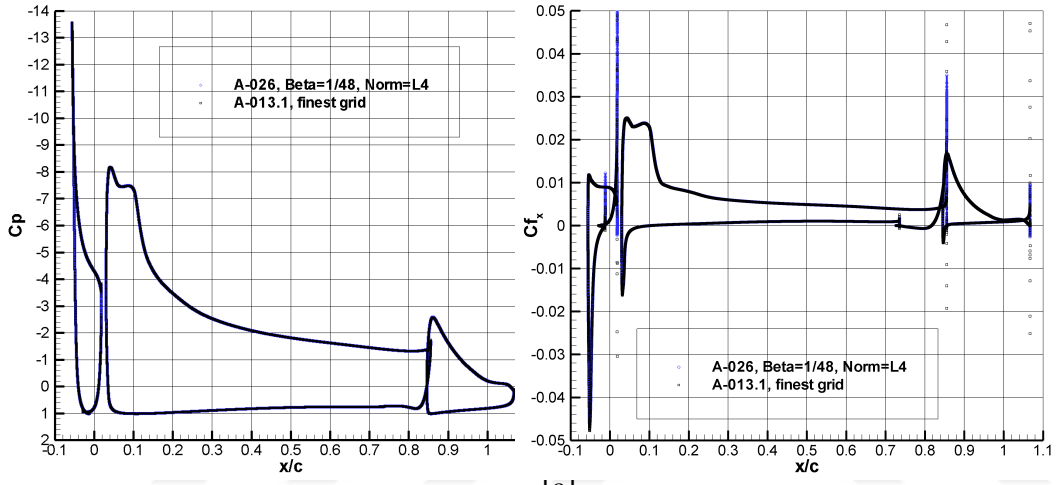
The other sources of numerical dissipation such as the use of the Harten correction coefficient ε, the limiter coefficient used in the Venkatakrishnan limiter K_V are also investigated on a relatively coarse mesh and the results are provided in Table 5.4. The results clearly indicate that the amount of upwinding strongly affects the resulting aerodynamic forces. The large value of the Harten correction coefficient also adversely affects the aerodynamic loads. The third-order upwind least squares interpolations for the convective fluxes is also used and it shows significant improvement on the drag coefficient. During the simulations, it has been noted that our refinement is relatively aggressive compared to the other benchmark results and the number of levels is increased in three-dimensions. The use of several more iterative refinement levels can help to achieve a better mesh convergence since mesh adaptation convergence is relatively slow. In addition, it can be improved the convergence of Newton iterations since the initial solution will be closer to final converged solution.

The computed surface pressure coefficient and x-component of skin friction coefficient are provided in Figure 5.5 and the computed results are compared with the results provided by MIT.

The computed Mach contours along with streamlines are shown in Figure 5.6 multi-element airfoil and the flow is attached over the main wing and the flap. The entropy contours with streamlines are also provided to illustrate viscous dissipation

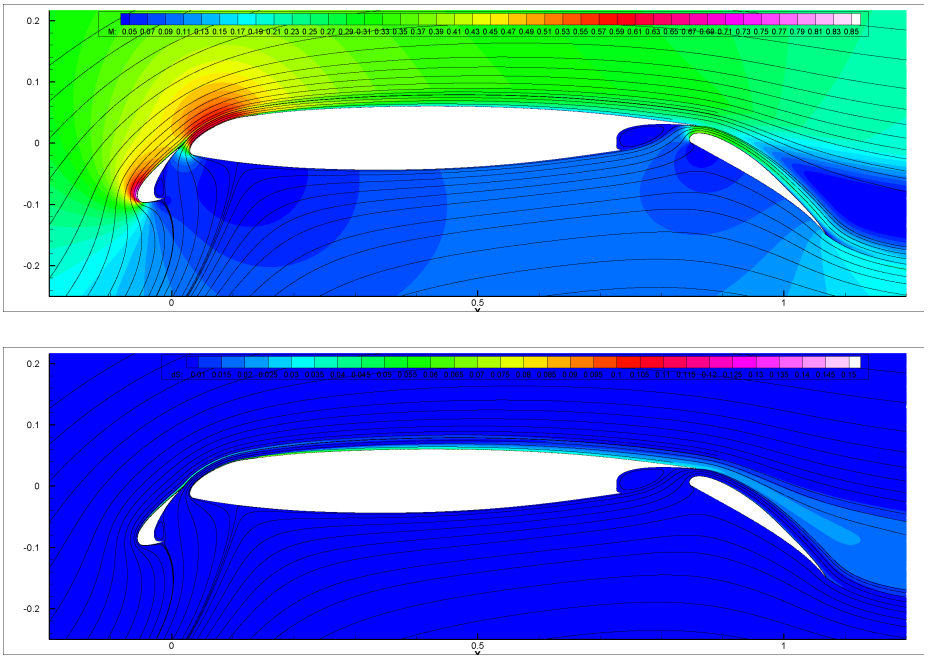
effects. The entropy function is also used as a sensor function during the adaptation, but it leads to excessive coarsening in inviscid regions, hence the computed aerodynamic loads are not better than those of Mach number.

For the present two-dimensional calculations, the conservation equation residuals are reduced to 1×10^{-8} or 1×10^{-9}. The residuals for the governing equations are computed and they are given in Figure 5.7, also the same figure provides information about the convergence of the lift, drag and moment coefficients for each mesh adaptation level.



[a] [b]

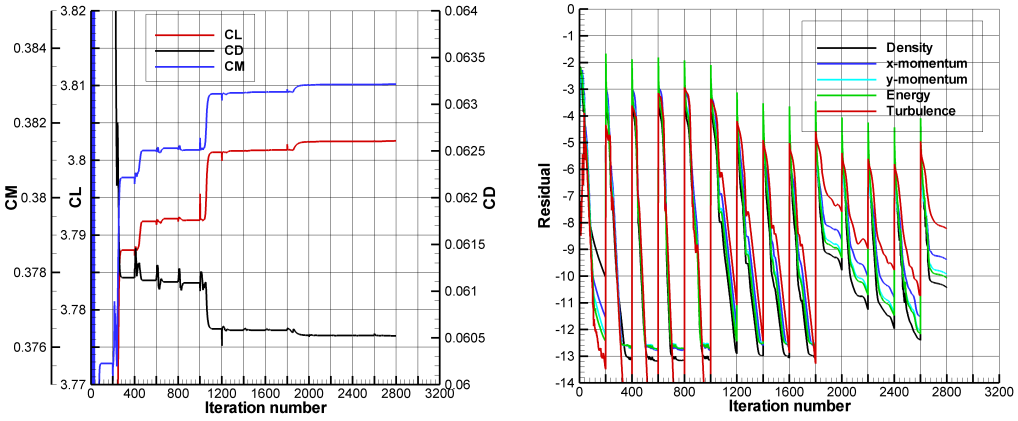
Figure 5.5 : Computed pressure [a] and x-component of skin friction [b] coefficients.



[a]

[b]

Figure 5.6 : Computed Mach [a] and entropy [b] contours at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ around a two-dimensional high-lift system.



[a] [b]

Figure 5.7 : Convergence of forces and residuals for each adaptation level at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ with L4 norm, $\beta = 1/48$, and $\varepsilon = 0.001$.

To evaluate the performance of the monolithic nonlinear Newton method under different configurations, a series of calculations were conducted using various combinations. These include: a first-order preconditioner with first-order Jacobian-vector multiplication, a first-order preconditioner with second-order Jacobian-vector multiplication, and a second-order preconditioner with second-order Jacobian-vector multiplication, as illustrated in Figure 5.8. The assessments were carried out using both the finite difference method and the direct approach on the ite8 mesh described in Table 5.3. The preconditioner is implemented using the restricted additive Schwarz method, with ILU(5) applied within each sub-block, and a maximum Krylov subspace dimension set to 80. When constructing a first-order accurate Jacobian matrix, the calculations exhibit poor convergence, as illustrated in Figure 5.8-[a-b]. These cases often require thousands of iterations, and it emphasizes the importance of the accurate Jacobian evaluation. At higher angles of attack, this approach may even fail to converge to machine precision.

In contrast, using a second-order accurate Jacobian matrix with a first-order preconditioner achieves the best residual convergence, as shown in Figure 5.8-[c-d]. The results indicate minimal differences in convergence between the finite difference and direct approaches, with the direct method performing slightly better when errors drop below the truncation threshold of the finite difference method. However, solver time is significantly influenced by specified tolerances. While the finite difference method requires higher computational cost for matrix-vector multiplication, the direct

approach is more expensive due to constructing the second-order Jacobian matrix. Despite these costs, the direct method is approximately 24% faster, though it demands significantly more memory.

On the other hand, coupling a second-order preconditioner matrix with the ILU(5) preconditioner significantly slows down the residual convergence, as shown in Figure 5.8-[e-f]. This issue caused from limitations in achieving higher CFL numbers. While the second-order preconditioner matrix offers a more accurate approximation of the Jacobian, convergence can theoretically occur within two iterations if an ideal preconditioner is used. This approach reduces the cost of Jacobian-vector multiplications during Krylov subspace iterations in the finite difference method, but such efficiency is not possible with a first-order preconditioner.

Surprisingly, the Krylov subspace iterations often stall when using the ILU(k) preconditioner in two-dimensional cases, it causing a significant reduction in the maximum achievable CFL number. To better understand the iterative convergence issues with the second-order Jacobian as a preconditioner, leading minimum and maximum eigenvalues were computed using the SLEPc library [80] without matrix scaling. The eigenvalue spectrum for both approaches is summarized in Table 5.5 and Table 5.6. The second-order Jacobian exhibits a large complex component in its maximum eigenvalue, which likely contributes to the poor convergence [96]. These results were validated through the Arnoldi process explained in PETSc [47], which further confirmed that preconditioned eigenvalues from the second-order Jacobian demonstrate greater scatter in the complex plane. Increasing the ILU(k) preconditioner level improves this convergence issue. This observation aligns with the findings of Nejat and Ollivier-Gooch [97], who reported that higher-order preconditioners require greater ILU(k) fill levels for optimal convergence in two-dimensional Euler equations. They also noted that first-order Jacobians tend to be better conditioned than their higher-order calculations.

Therefore, employing a second-order accurate Jacobian with a first-order preconditioner matrix seems the most effective approach for residual convergence. Although the direct method is slightly faster—by approximately 24%—the computational time difference between direct and finite difference methods is relatively minor.

The influence of the blending parameter β on convergence is shown in Figure 5.9, where the upwind method consistently outperforms others. Adjustments to parameters such as $\beta = 1$, larger Harten coefficients, and smaller CFL_{max} values [78] in the preconditioner matrix construction enhance ILU(k) stability and convergence. However, consistent preconditioner construction remains critical during the simulations.

These calculations employ the default backtracking line search algorithm, yet a full-step Newton method, which avoids line search, may have better convergence for similar cases. Notably, the turbulence residual dominates the total residual on finely adapted meshes, even with a large scaling factor of 1×10^6 for the turbulence working variable. This negatively impacts overall convergence and becomes more clear at higher angles of attack. Future studies may investigate removing turbulence residuals from the line search algorithm, as suggested in [79].

Table 5.5 : Leading minimum and maximum eigenvalues at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$ and $\alpha = 16^\circ$ with L4 norm, $\beta = 1/48$ and $\varepsilon = 0.001$ using $CFL = 10,000$ for ite8 mesh for the First-order Jacobian matrix.

Index	Smallest Eigenvalue	Largest Eigenvalue
1	4.394522×10^{-8}	2.591630×10^{2}
2	4.790529×10^{-8}	2.554619×10^{2}
3	4.831292×10^{-8}	2.507811×10^{2}
4	5.483217×10^{-8}	2.501795×10^{2}
5	6.003424×10^{-8}	2.491021×10^{2}

Table 5.6 : Leading minimum and maximum eigenvalues at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$ and $\alpha = 16^\circ$ with L4 norm, $\beta = 1/48$ and $\varepsilon = 0.001$ using $CFL = 10,000$ for ite8 mesh for the Second-order Jacobian matrix.

Index	Smallest Eigenvalue	Largest Eigenvalue
1	3.311019×10^{-8}	$1.796677 + 1.192024 \times 10^{2}i$
2	3.318570×10^{-8}	$1.796677 - 1.192024 \times 10^{2}i$
3	3.537033×10^{-8}	$2.047810 + 1.183090 \times 10^{2}i$
4	3.697612×10^{-8}	$2.047810 - 1.183090 \times 10^{2}i$
5	3.990653×10^{-8}	$1.852557 + 1.164716 \times 10^{2}i$

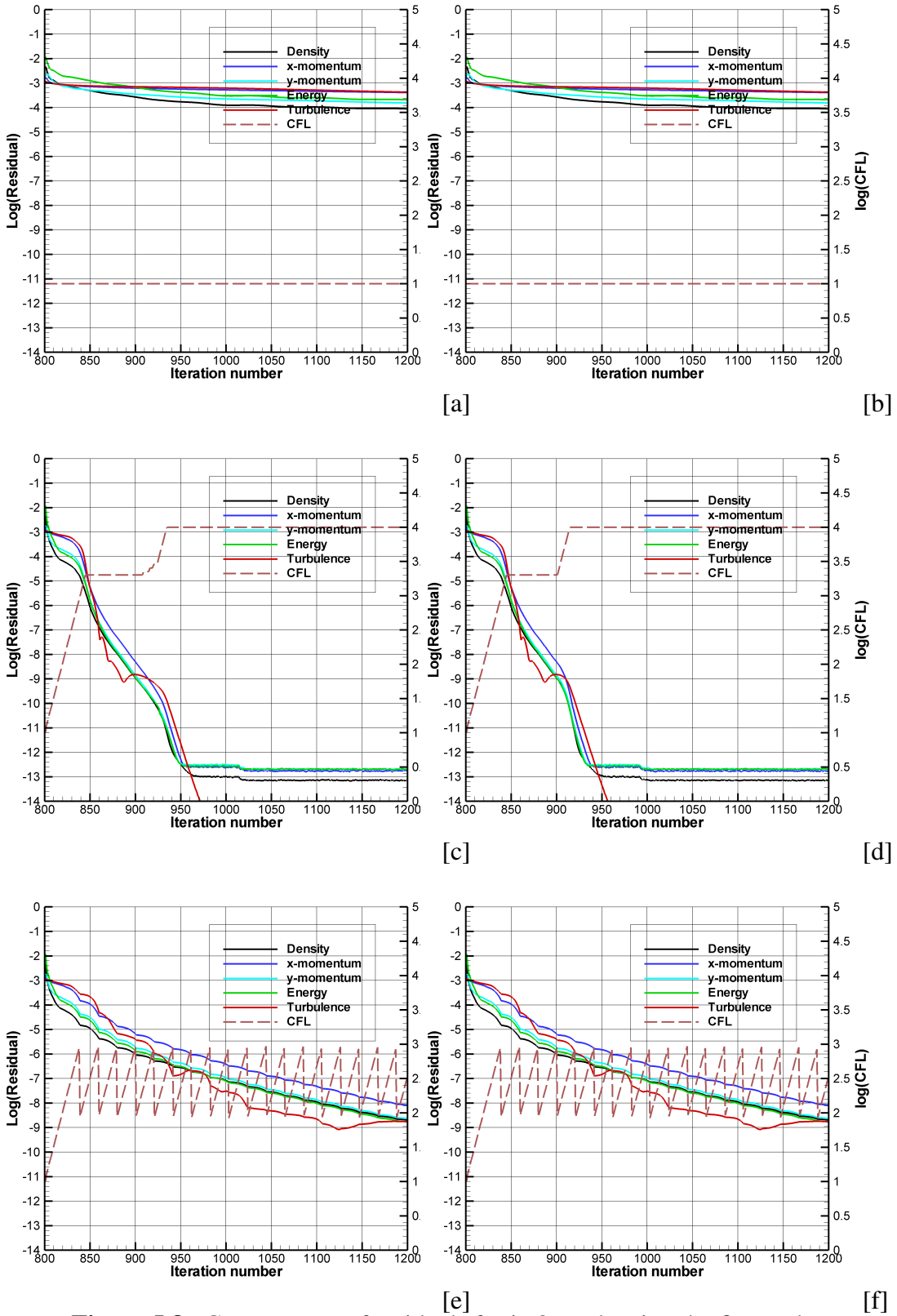
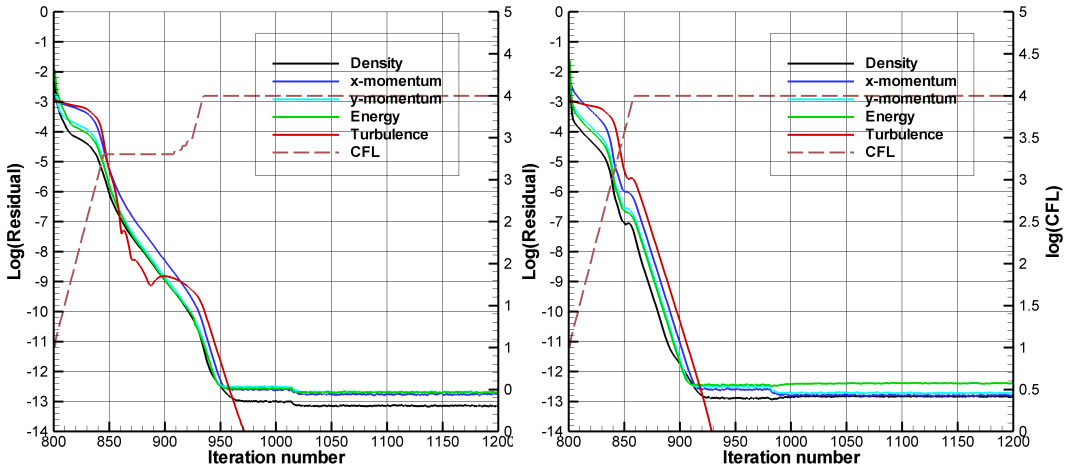


Figure 5.8 : Convergence of residuals for ite8 mesh using the first-order preconditioner and the first-order Jacobian vector multiplication [a-b], the first-order preconditioner and the second-order Jacobian vector multiplication [c-d], and the second-order preconditioner and the second-order Jacobian vector multiplication [e-f] with finite difference (left) and direct approach (right) at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ with L4 norm, $\beta = 1/48$, and $\varepsilon = 0.001$.



[a] [b]

Figure 5.9 : Convergence of residuals for ite8 mesh using the first-order preconditioner and the second-order Jacobian vector multiplication with finite difference using blending parameter $\beta = 1/48$ [a] and $\beta = 1$ (upwind) [b] at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ with L4 norm, and $\varepsilon = 0.001$.

The second test case, designated as Case 1b, involves conducting mesh convergence studies on the three-dimensional NASA High Lift Common Research Model (CRM-HL) in a landing configuration as detailed in the Section 1.5.4.

The physical parameters for this simulation are $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$. Similar to the two-dimensional cases, the mesh adaptation process gradually increases in complexity, with refinement levels increasing by a factor of two across four successive iterations. This approach ensures stability for the Newton method in three-dimensional calculations. The minimum edge length in the adapted mesh is set to 1×10^{-6}, and the pyAMG stretching factor is maintained at 1.2 to ensure enough resolution in critical regions.

Boundary layer formation, which is relatively slow using the classical Mach sensor, is accelerated in the first ten adaptation levels by employing an exponential distance-based sensor function. Subsequent calculations continued with the Mach sensor function to refine the mesh further. The computational domain at the farfield, scaled by the mean chord length, spans $[-82.794198, 110.630203] \times [0, 82.794198] \times [82.794198, 82.794198]$, maintaining a similar body-length-to-far-field ratio as used in the JAXA case for the third AIAA CFD High-Lift Prediction Workshop [87]. A smaller computational domain is used compared to the one proposed on the workshop to reduce

the large number of vertices in the far-field wake, which can reduce the quality of mesh resolution near the body.

The compressible Navier-Stokes equations are solved using the Roe scheme [39], with an upwinding parameter β initially set to $1/3$ to minimize cross-diffusion effects. The turbulence model employs second-order upwind interpolation with $\beta = 1$. Simulations start from free-stream conditions (cold start) with a large physical time step ($\Delta t = 1 \times 10^{20}$) and an initial CFL number of 10, ramped up using a multiplier of 1.25 to a maximum CFL of $10,000$. To address issues of negative density values arising from excessively small volumes on fine meshes, the Harten correction coefficient $\varepsilon = 0.001$ is used, along with a modification proposed by Harten and Hyman [98].

The preconditioner is the PETSc restricted additive Schwarz method with ILU(0) applied to each sub-block. The Krylov subspace solver utilizes a restarted flexible generalized conjugate residual method, with a maximum of 80 stored vectors. This choice is informed by studies such as those by Yildirim et al. [99], which emphasize the importance of appropriate solver tolerances which is set as $rtol = 10^{-8}$ and $atol = 10^{-20}$. These values are stricter than typically seen in the literature [78], but they helped to address convergence challenges for this complex geometry.

Several adapted meshes using the L4 norm are presented at the $y/\bar{c} = 1.417693$ plane in Figure 5.10, with additional details provided in Table 5.8. The progress of surface mesh refinement is also illustrated in Figure 5.11, highlighting the extensive refinement within boundary layers, sharp corners, and wakes. Notably, mesh refinement is clear in the wake region of flap brackets on the upper surface of the main wing, as shown in Figure 5.11. Predicting these regions in advance using only theoretical knowledge is particularly difficult. In earlier work [100], these refinement regions in the flap bracket wakes were entirely absent due to the use of an exponential distance sensor combined with the Mach number, in contrast to the current Mach-only sensor. These small yet critical refinement regions are essential for accurately capturing aerodynamic loads, demonstrating the importance of sensor selection in mesh adaptation processes.

The final adapted mesh consists of 92,993,470 vertices and 543,825,694 tetrahedral elements.The calculation for the finest mesh is repeated using a lower β value of 1/48, and the corresponding lift, drag, and moment coefficients — calculated with respect

to the moment reference center given in Table 5.2 — are detailed in Table 5.7. The convergence behavior of lift and drag coefficients with successive mesh refinement is depicted in Figure 5.12, with numerical values provided in Table 5.8.

Table 5.7 : Refinement data at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$ and $\alpha = 7.05^\circ$ with L4 norm, $\beta = 1/48$ and $\varepsilon = 0.001$.

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	Number of Surface Elements	Cl	Cd	Cm
ite27 (Mach)	92,993,470	543,825,694	10,203,212	1.769122	0.186155	-0.353909
Exp.				1.778620	0.186706	-0.370599

Table 5.8 : Refinement data for L2, $\alpha = 7.05^\circ$ and $\beta = 1/3$.

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	Number of Surface Elements	Cl	Cd	Cm
ite0 (Initial Mesh)	2,618,827	14,509,928	843,643			
ite1 (Initial Mesh)	5,660,885	13,196,847	1,003,329			
ite2 (Distance)	6,021,693	14,987,885	830,244			
ite3 (Distance)	6,643,490	18,463,634	811,506			
ite4 (Distance)	7,496,239	22,941,616	880,928			
ite5 (Distance)	8,572,758	28,193,846	998,814			
ite6 (Distance)	9,713,500	33,388,392	1,132,785			
ite7 (Distance)	10,712,948	37,694,004	1,252,267			
ite8 (Distance)	11,490,737	40,712,295	1,349,091			
ite9 (Distance)	12,034,853	42,756,610	1,424,385			
ite10 (Mach)	12,449,392	72,478,755	1,482,386	1.704234	0.180114	-0.311449
ite11 (Mach)	12,721,876	73,923,462	1,640,066	1.708239	0.178125	-0.315657
ite12 (Mach)	13,286,148	77,446,487	1,552,871	1.733436	0.181784	-0.324712
ite13 (Mach)	13,876,245	80,978,130	1,555,917	1.744635	0.183299	-0.330147
ite14 (Mach)	14,336,919	83,725,144	1,564,116	1.737834	0.182961	-0.328728
ite15 (Mach)	19,901,979	116,649,283	1,955,555	1.744465	0.183693	-0.334081
ite16 (Mach)	22,008,855	128,949,870	2,117,485	1.745356	0.184149	-0.335724
ite17 (Mach)	23,339,823	136,785,487	2,210,317	1.741864	0.183549	-0.336332
ite18 (Mach)	24,069,842	141,072,890	2,269,402	1.740719	0.183466	-0.336205
ite19 (Mach)	32,106,767	188,019,314	3,223,728	1.745157	0.183651	-0.340011
ite20 (Mach)	34,808,438	203,794,362	3,445,091	1.746952	0.183873	-0.341205
ite21 (Mach)	36,330,510	212,756,204	3,553,626	1.747490	0.183924	-0.341603
ite22 (Mach)	37,103,098	217,289,371	3,621,617	1.747814	0.183989	-0.341658
ite23 (Mach)	50,399,305	294,965,511	5,156,653	1.752651	0.184323	-0.344492
ite24 (Mach)	55,001,010	321,856,921	5,536,462	1.754142	0.184579	-0.345350
ite25 (Mach)	57,335,167	335,582,845	5,721,636	1.758472	0.185291	-0.346369
ite26 (Mach)	58,528,316	342,562,218	5,839,797	1.759224	0.185333	-0.346673
ite27 (Mach)	92,993,470	543,825,694	10,203,212	1.763347	0.185624	-0.349474
Exp.				1.778620	0.186706	-0.370599

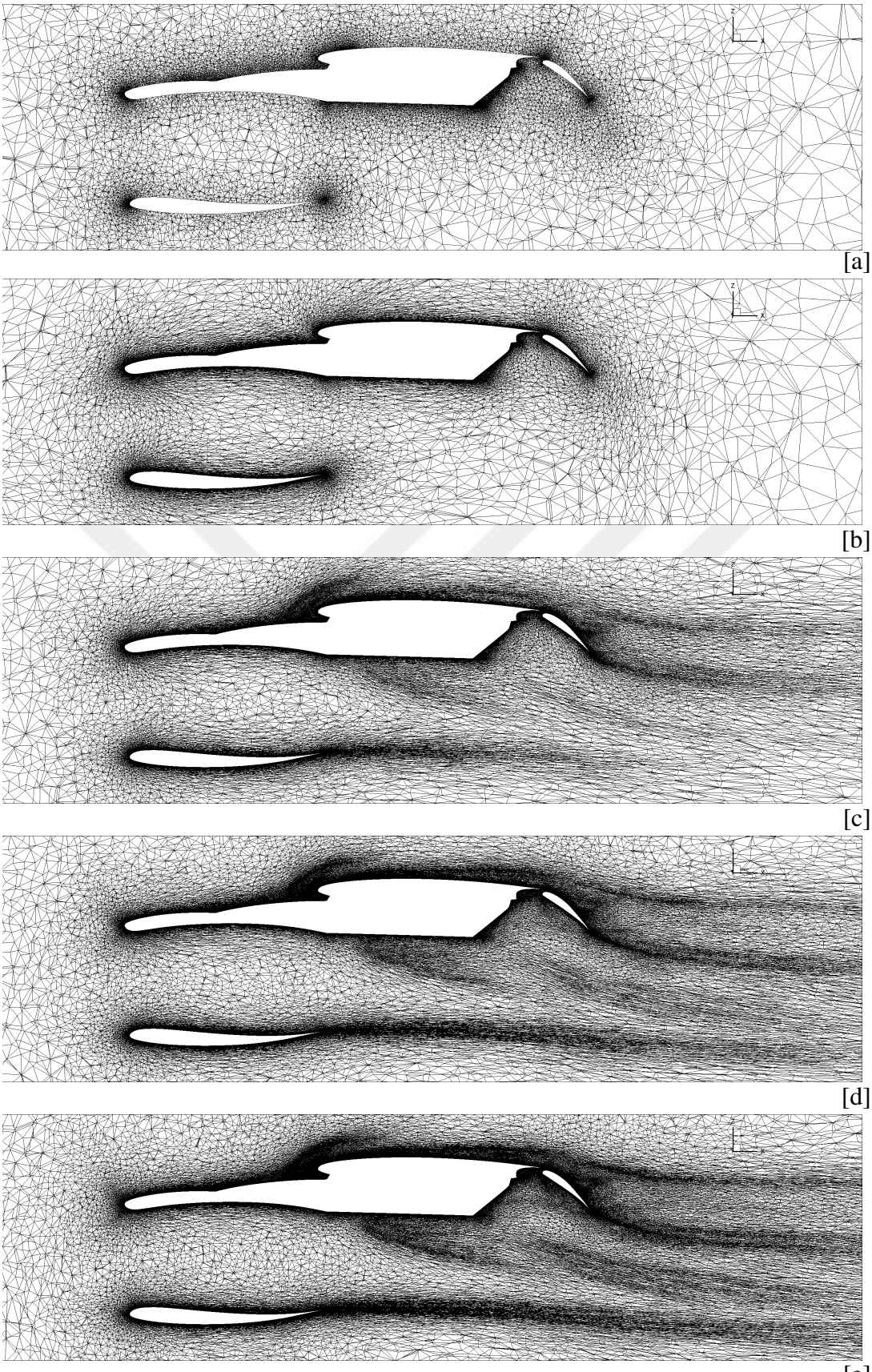
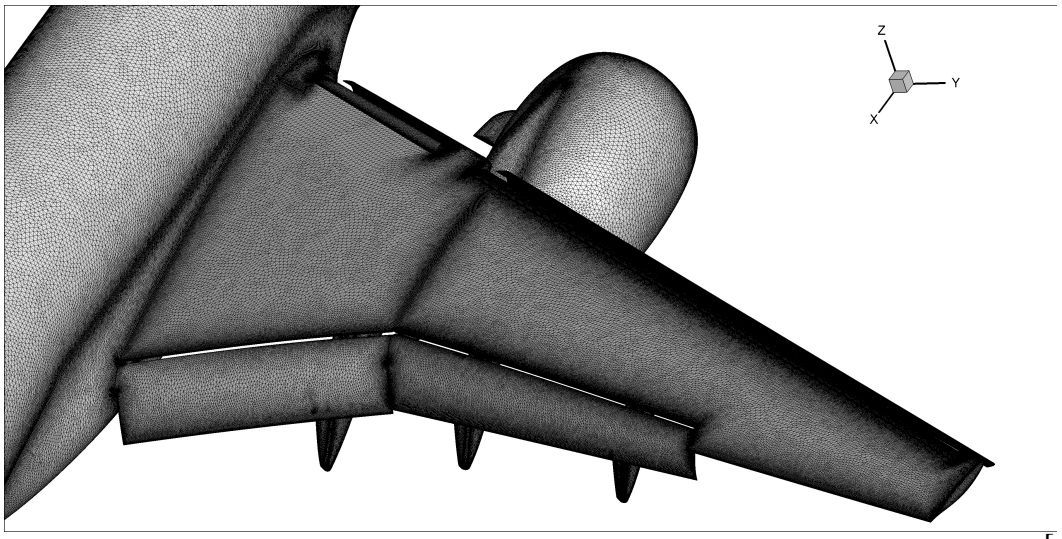
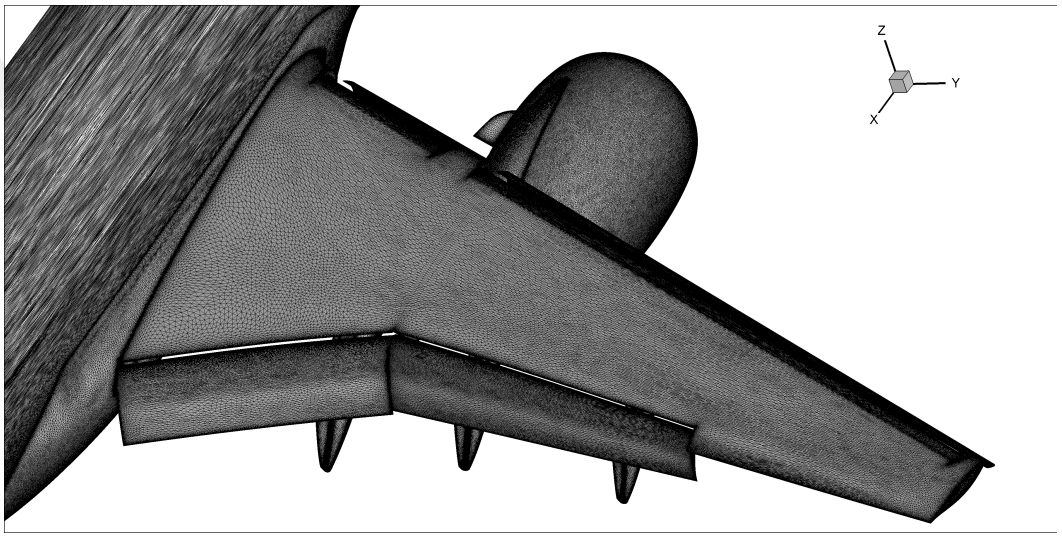


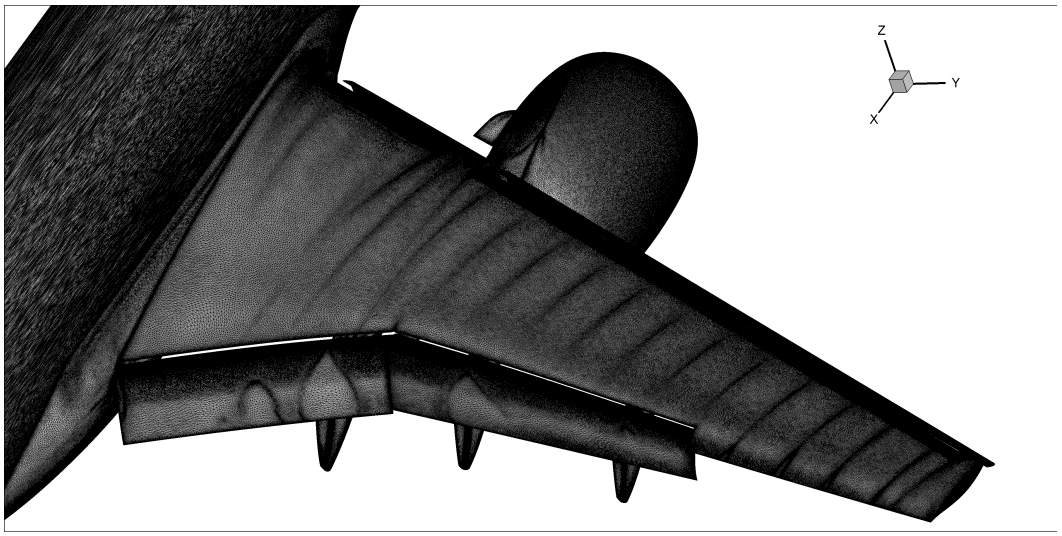
Figure 5.10 : Progress of mesh adaptation at $y = 393.70$ plane for $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$ and $\alpha = 7.05^\circ$ around 3D-CRM high-lift system: [a] Initial mesh with no boundary layer, [b] 10th iteration with distance only, [c] 14th [d], 18th [e] and 22nd iteration with distance and Mach.



[a]



[b]



[c]

Figure 5.11 : Progress of surface mesh adaptation at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$ and $\alpha = 7.05^\circ$ with L4 norm, $\beta = 1/3$ and $\varepsilon = 0.001$ around three-dimensional NASA CRM-HL system: [a] Initial mesh with no boundary layer, [b] 10th iteration with exponential distance sensor, and [c] 22nd iteration with Mach sensor.

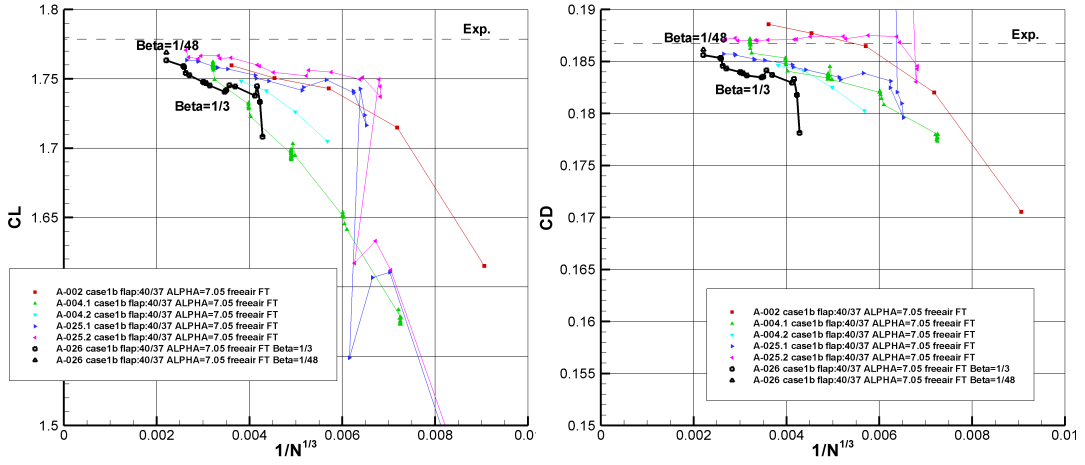
For multiscale mesh adaptation, the L4 norm is employed in both two- and three-dimensional analyses. Typically, the L4 norm yields a higher mesh resolution within the boundary layer, resulting in improved aerodynamic load calculations, particularly for the drag coefficient [83, 101]. However, the L2 norm tends to deliver better mesh resolution in the wake regions. In the case of the NASA CRM high-lift system, the nacelle strake generates significant vortex lift over the main wing, making accurate capture of its wake region critically important. As a result, the L2 norm may offer advantages in achieving a more precise lift coefficient, particularly for resolving the wakes of slat brackets located close to the main wing. However, the L2 norm often requires a substantial number of vertices to resolve the wake region adequately, particularly in cases involving a relatively large far-field domain.

Alauzet and Frazza [101] observed that starting with coarser meshes is highly beneficial since the convergence process involves extensive iterations, which are significantly less computationally expensive compared to those required for finer meshes. Additionally, mesh adaptation tends to converge more slowly on finer meshes. However, in this study, starting with an extremely coarse mesh was not feasible due to the potential for meshing errors, particularly around the sharp leading edge of the main wing.

To address inaccuracies associated with Green-Gauss gradient values near boundaries, aerodynamic loads in this work are computed using a conservative approach. This method involves summing the control volume momentum fluxes at vertices adjacent to boundaries, enabling accurate computation of aerodynamic forces on solid walls [59]. The computed aerodynamic loads are also compared against the results submitted by the AIAA ADAPT Technology Focus Group (TFG) during the workshop, as shown in Figure 5.12. The details of the codes and participants are given in the Table 1.5 in section 1.5.4. Boeing's computed drag coefficient (A-0025.2) on coarse meshes demonstrates relatively good agreement with the experimental data, largely due to the use of objective-based mesh adaptation combined with boundary layer insertion and anisotropic refinement in the EPIC software [33]. The current adaptive results shows the aerodynamic loads that are closely aligned with one another, demonstrating improved consistency when compared to the RANS TFG results. However, the fine mesh solutions exhibit slightly lower values than those observed in the experimental

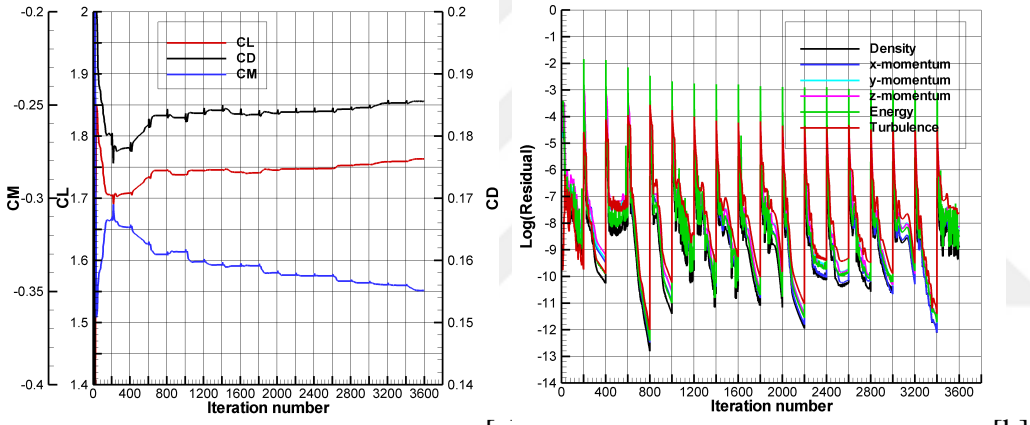
data. A similar trend was reported during the 7th AIAA CFD Drag Prediction Workshop (DPW7) for adaptive results [102]. This discrepancy may be attributed to the large turbulence variables within vortex cores, which can cause excessive dissipation and adversely impact the effective angle of attack.

The convergence of aerodynamic loads and residuals is depicted in Figure 5.13, as a function of the Newton iteration number. For the complex three-dimensional NASA High Lift Common Research Model (case 1b), residual values approaching machine precision can be achieved, provided that the CFL number is sufficiently large (on the order of 10,000). During the 4th AIAA CFD High Lift Prediction Workshop, achieving machine-precision residual convergence presented significant challenges. Even the HANIM solver, used in conjunction with the NASA FUN3D solver, could only achieve residual convergence to approximately 1×10^{-9} for case 1b [103]. This underscores the complexity of ensuring convergence for intricate high-lift configurations. Machine-precision convergence was achieved for the three-dimensional wing-body configuration of the CRM-HL (case 2.1) at the 5th AIAA CFD High Lift Prediction Workshop with a maximum CFL number of 1,000. However, more complex configurations, such as those incorporating slats and slat brackets (case 2.2), failed to reach machine precision under the same conditions, exhibiting residual oscillations around 1×10^{-7}. Notably, both cases successfully converged to machine precision when the maximum CFL number was increased to 10,000 [104]. To emphasize the critical role of higher CFL values, a calculation for the ite15 mesh in Table 5.8 was repeated with $\text{CFL}_{max} = 1,000$ and $\text{CFL}_{max} = 10,000$. As shown in Figure 5.14, the lower $\text{CFL}_{max} = 1,000$ value caused the residuals to stall at approximately 1×10^{-7}. This shows the importance of utilizing high CFL values to enforce steady-state and achieve machine-precision convergence.



[a] [b]

Figure 5.12 : Mesh convergence of conservative lift coefficient [a] and drag coefficient [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ with L4 norm, $\beta = 1/3$, and $\varepsilon = 0.001$ around the three-dimensional NASA CRM-HL system.



[a] [b]

Figure 5.13 : Convergence of forces and residuals for each adaptation level at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ with L4 norm, $\beta = 1/3$, and $\varepsilon = 0.001$.

The convergence behavior of high-order preconditioner construction in three dimensions was also examined using the direct approach with first- and second-order preconditioner reconstructions, as illustrated in Figure 5.15. Interestingly, the behavior of the second-order preconditioner in three dimensions differed significantly from that in two dimensions. It did not require a high level of fill in the block incomplete Lower Upper factorization (ILU(k)) preconditioner to achieve optimal convergence. Moreover, the second-order preconditioner allowed for slightly higher maximum CFL values, resulting in improved convergence.

The computation time analysis for the direct and finite difference approaches, employing the first-order preconditioner reconstruction, reveals that the direct

approach is approximately 14% more computationally expensive. It is worth noting that at high CFL numbers, the linear solver struggles to achieve the specified relative tolerance within the maximum limit of 80 Krylov subspace iterations. However, lower relative tolerances are generally sufficient to ensure the convergence of the Newton method [99].

Both first- and second-order preconditioning approaches exhibit a consistent trend toward achieving machine-precision convergence, even for geometrically complex three-dimensional cases. The finite difference-based approach holds a distinct advantage in its ability to capture solution variations during Krylov subspace iterations, therefore making it less likely to stall during SNES iterations.

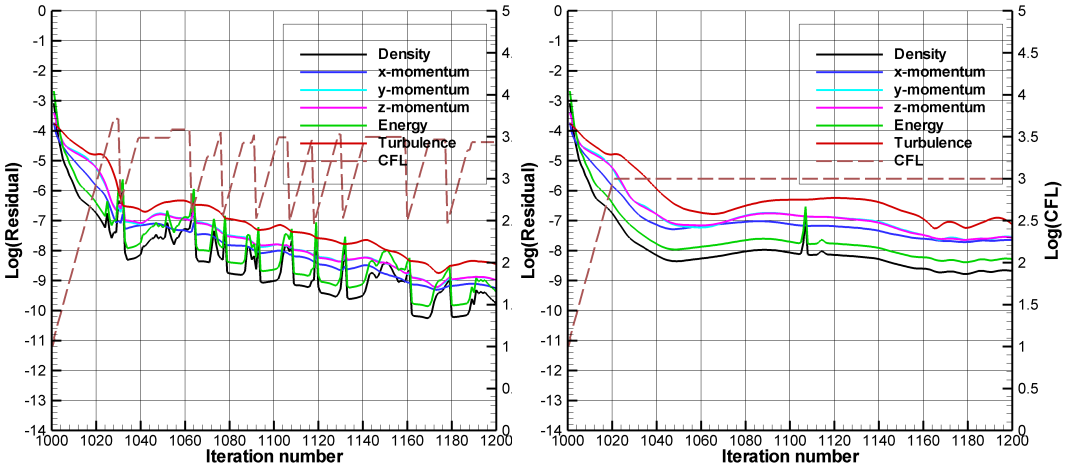
In contrast, the direct approach delivers faster computation times by performing more efficient Jacobian-vector multiplications. This eliminates the need for costly residual vector evaluations and becomes particularly advantageous as the number of multiplications increases, such as under higher CFL conditions. However, this benefit is counterbalanced by the significantly higher memory requirements of the direct method.

The computed pressure coefficients at various sections are compared with experimental data, as shown in Figure 5.16. While the results indicate a slight underprediction of the pressure coefficients on the upper surface of the main wing, particularly near the trailing edges, the computed total lift coefficient on the finest mesh aligns closely with the experimental value. The exact cause of this inconsistency remains unclear.

Figure 5.17 presents the computed skin friction coefficient contours using several workshop view formats. The flow over the outboard flap shows separation, and a significant separation bubble is observed on the inner flap surface. Figure 5.18 shows the comparison of these skin friction coefficient contours obtained with HEMLAB at Figure 5.18 [f] and the other committee members.

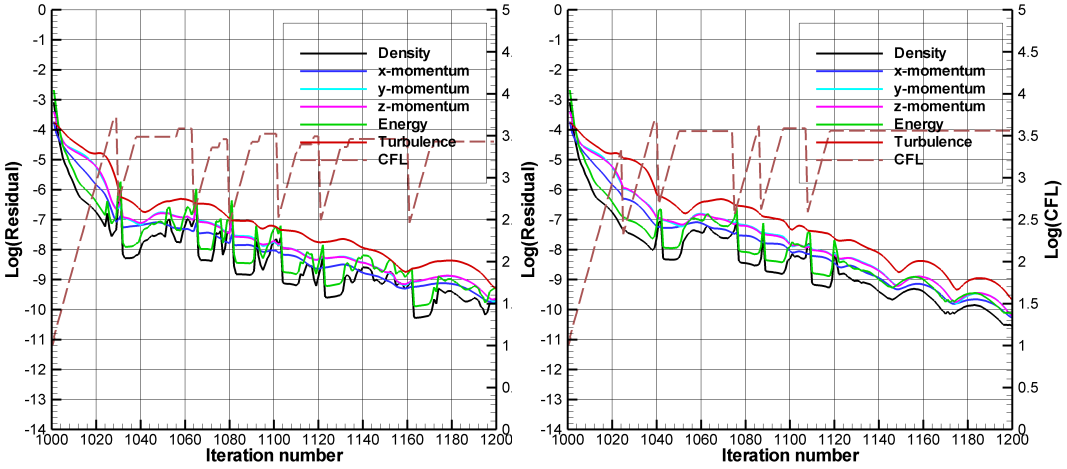
Additionally, Figure 5.17 and Figure 5.20 illustrates the skin friction contours with streamlines and the wake structure represented by entropy isosurfaces around the NASA CRM-HL model. These visualizations highlight a prominent vortical structure generated by the nacelle-strake, along with another strong vortical structure originating from the nacelle-pylon.

The vorticity contours with the adapted meshes are displayed at multiple planes in Figure 5.19, following the workshop's postprocessing guidelines. Despite the inherent complexity of the three-dimensional wake structure, the adapted mesh effectively captures these regions, demonstrating its capability to resolve intricate flow features accurately.



[a] [b]

Figure 5.14 : Convergence of residuals for ite15 mesh using the first-order preconditioner and the second-order Jacobian vector multiplication with finite difference with $\text{CFL}_{max} = 10,000$ [a] and $\text{CFL}_{max} = 1,000$ [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ with L4 norm, $\beta = 1/3$, and $\varepsilon = 0.001$.



[a] [b]

Figure 5.15 : Convergence of residuals for ite15 mesh using the first-order preconditioner [a] and the second-order preconditioner [b] using the second-order Jacobian vector multiplication with direct approach at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$ and $\alpha = 7.05^\circ$ with L4 norm, $\beta = 1/3$ and $\varepsilon = 0.001$.

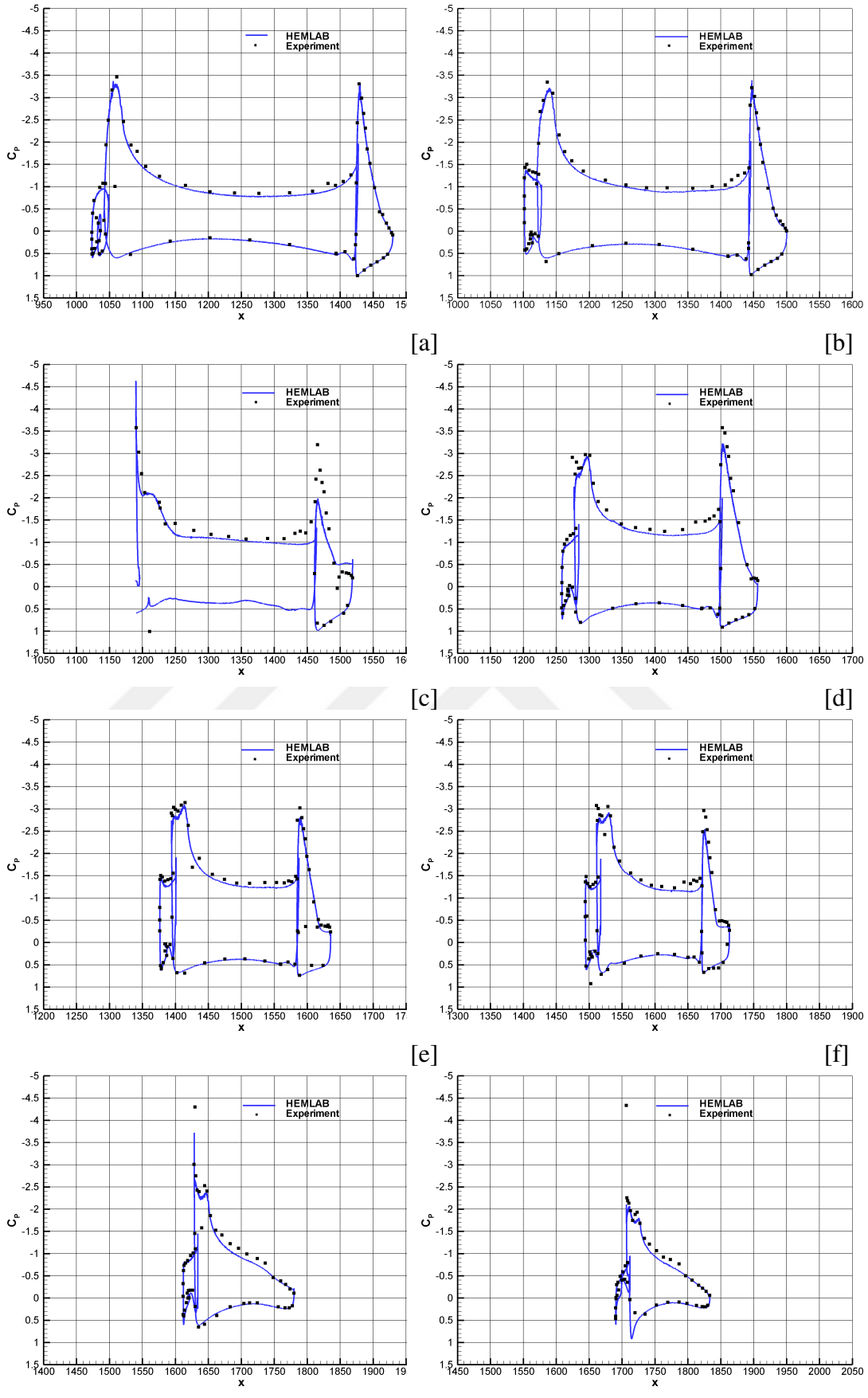
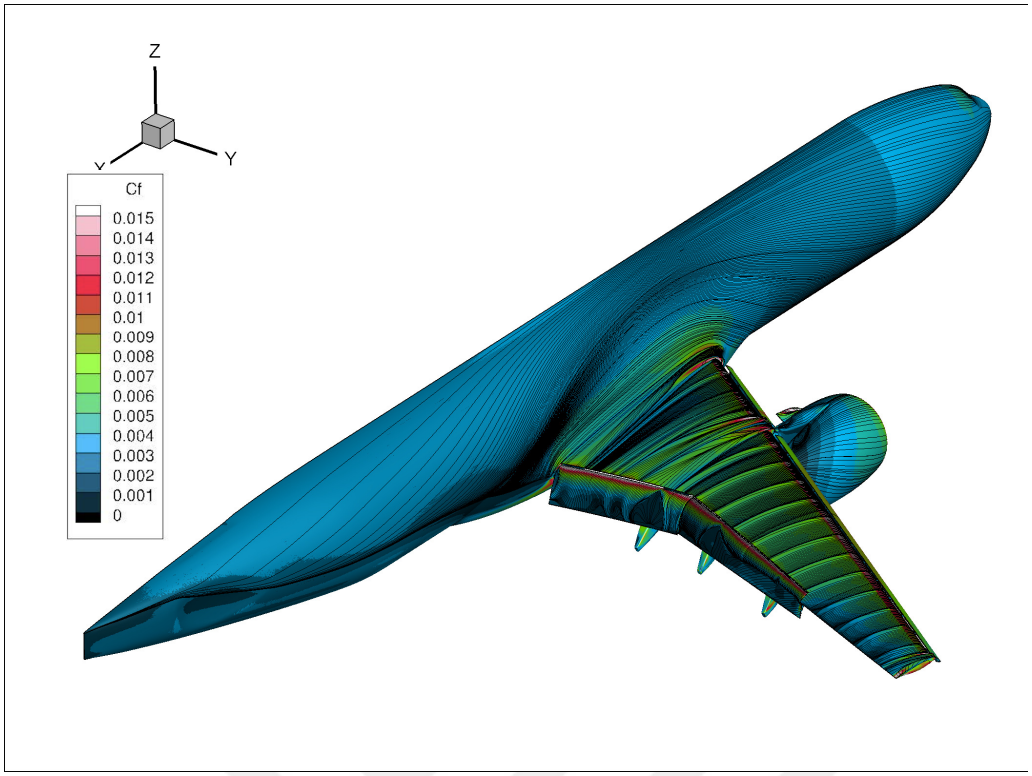
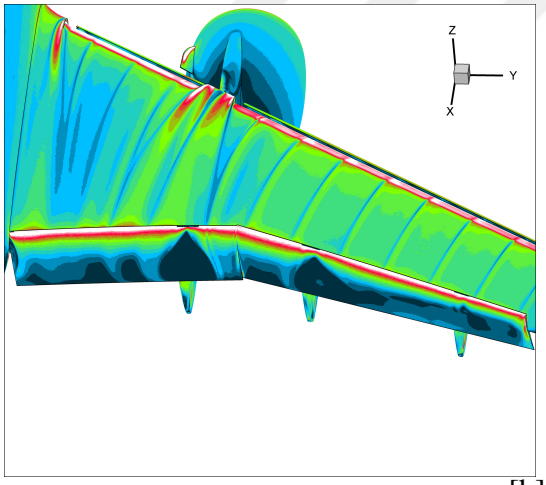


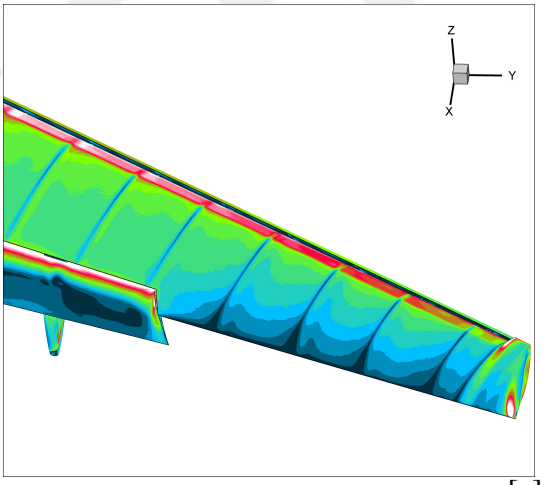
Figure 5.16 : Computed pressure coefficients at sections A to H [a-h] for $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05°$ around the NASA CRM high-lift system.



[a]



[b]



[c]

Figure 5.17 : Computed skin friction contours with streamtraces around the NASA CRM high-lift system: [a] overall view, [b] wing root (view5), and [c] wing tip (view6) at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05°$.

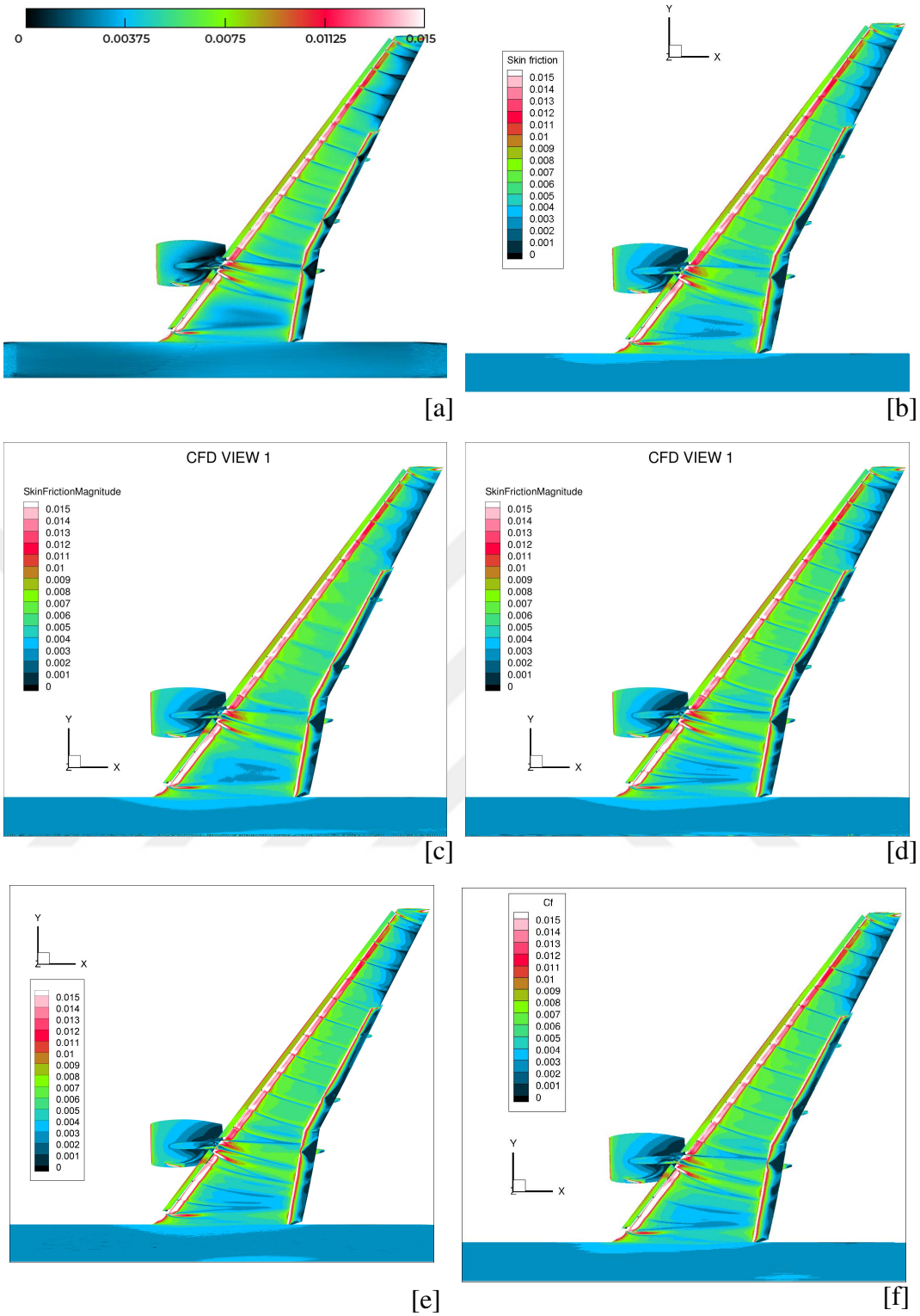
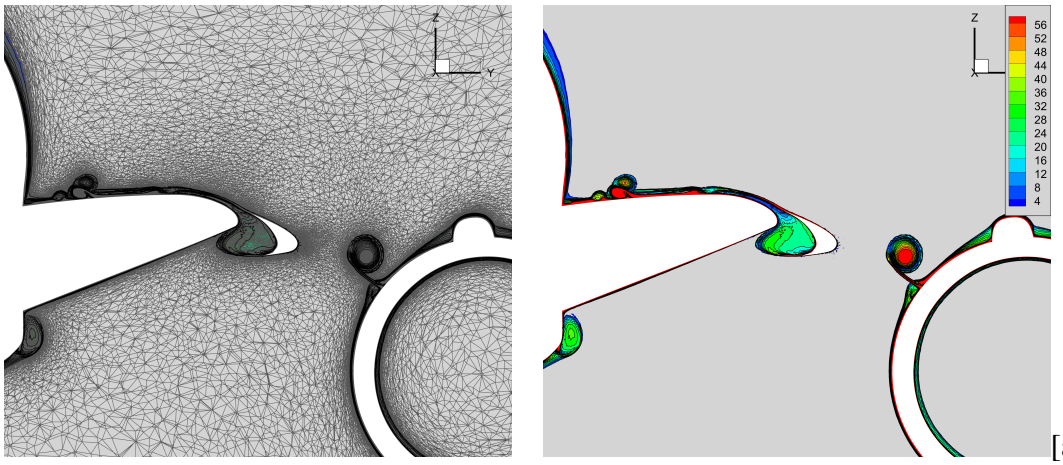
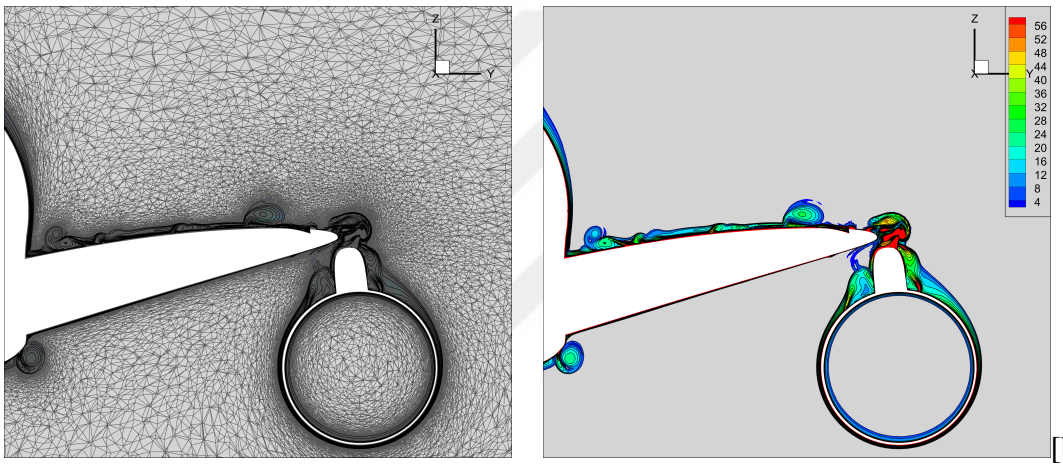


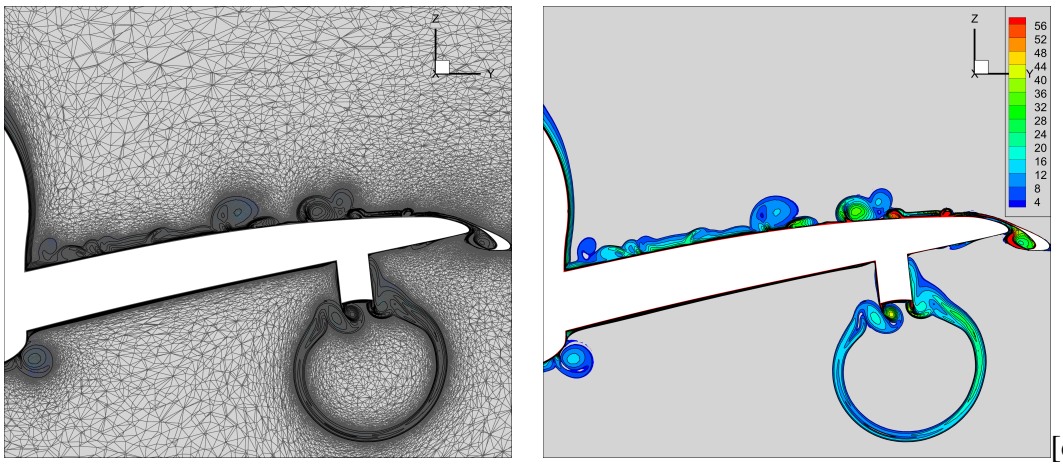
Figure 5.18 : CRM-HL skin friction contours at 7.05° angle of attack where [a] uses the lift metric by A-002, [b] is the multiscale metric by A-004, [c] is the multiscale metric by A-025.1, [d] is the drag metric by A-025.2, [e] is the multiscale metric by A-026 (HEMLAB), and [f] is the multiscale metric by A-031 [10].



[a]



[b]



[c]

Figure 5.19 : Adapted meshes (left) and computed non-dimensional vorticity magnitude ($|\omega|\bar{c}/U_\infty$) contours (right) at $x/\bar{c} = 3.970268$ (view 11) [a], $x/\bar{c} = 4.296591$ (view 12) [b] and $x/\bar{c} = 4.622915$ (view 13) [c] planes for $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$ and $\alpha = 7.05°$ around NASA CRM-HL system.

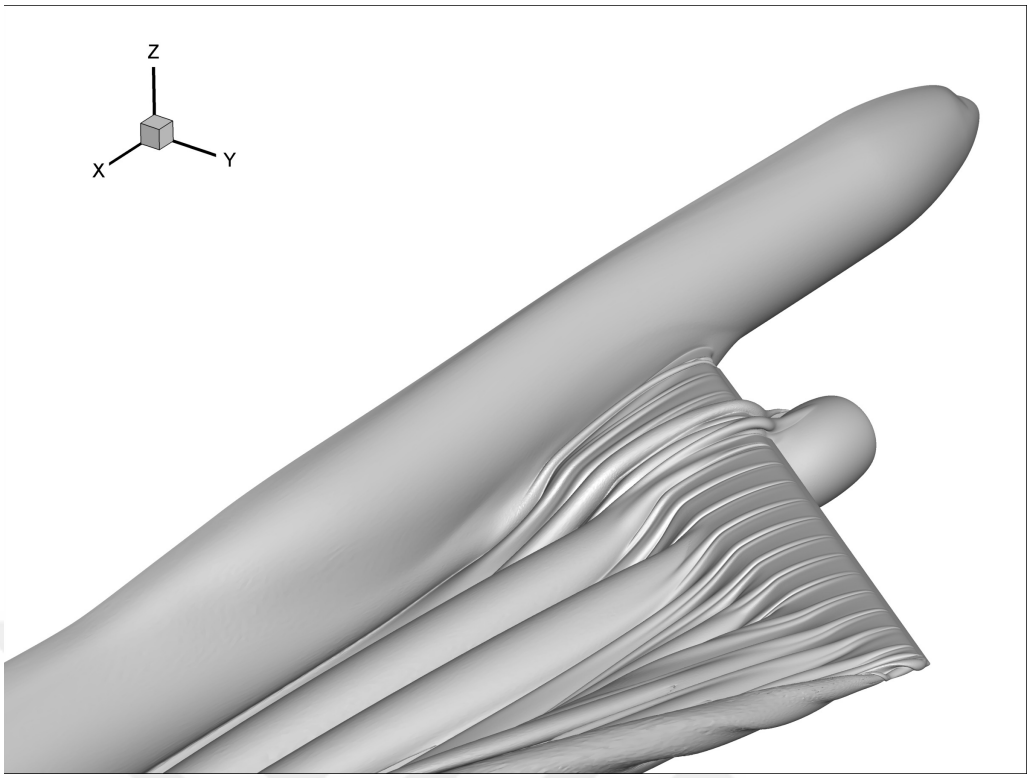


Figure 5.20 : Computed entropy isosurface ($\Delta s = 0.001$) at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ around the NASA CRM high-lift system.

6. NUMERICAL RESULTS FOR HLPW-5

As explained in Section 1.5.5, three main test cases will be investigated by the committee members in the High Lift Prediction Workshop-5. The first test case was announced in February 2023, with results collected during a mini-workshop in July 2023 and subsequently shared by the committee members, including those obtained with HEMLAB. This test case investigates the numerical results for the NASA-CRM-Wing-Body (CRM-HL-WB) configuration at $Re = 5.6 \times 10^6$, $M = 0.2$, $\alpha = 11^\circ$, and $T_\infty = 521^\circ R$. Although the Spalart-Allmaras-QCR-2000 turbulence model, which is available in the latest version of HEMLAB, is highly recommended, convergence problems necessitated that all calculations presented below were obtained using the Negative Spalart-Allmaras model. A similar mesh adaptation process was performed for this configuration as was done in HLPW-4. The results obtained with the HEMLAB framework also use the adap_back feature of HEMLAB, as described in Section 3.4. As detailed in Section 3.3, the numerical solution was obtained using the new version of HEMLAB, which employs the SNES solver. All parameters used in the first test case are summarized in Table 6.1.

Table 6.1 : Flow conditions for Test Case 1 (CRM-HL-Wing-Body)

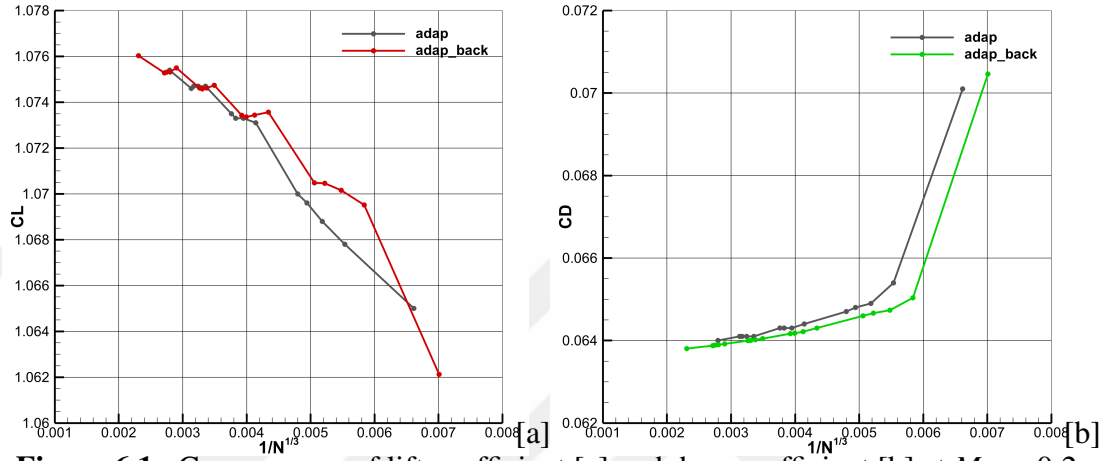
Mach Number	0.2
Re Number	5.6×10^6
Angle of Attacks	11.0°
Mean Aerodynamic Chord	275.8 inch
Reference Static Temperature	$521.0^\circ R$
Semi-Span Reference Area	$29736.0\,in^2$
Moment Reference Center	$x = 1325.9$ inches, $y = 0.0$ inches, $z = 177.95$ inches
Prandtl Number	0.72
Turbulent Prandtl Number	0.9
$\hat{\nu}_{freestream}/\nu_\infty$	3.0
Ratio of Specific Heats γ	1.4

Table 6.2 : Refinement data for L2, $\alpha = 7.05^\circ$ and $\beta = 1/3$.

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	C_L	C_D	C_M
ite0 (Initial Mesh)	10,630,135	62,254,785	-	-	
ite1 (Distance)	3,170,048	18,313,665	-	-	-
ite2 (Distance)	2,119,011	12,146,828	-	-	-
...	...	...	...	...	...
ite9 (Distance)	2,861,878	16,455,322	-	-	-
ite10 (Distance + Mach)	2,904,264	16,696,572	1.06213	0.07046	-0.05423
ite11 (Distance + Mach)	5,024,686	28,889,937	1.06951	0.06504	-0.05946
ite12 (Distance + Mach)	6,076,128	34,912,258	1.07016	0.06474	-0.05947
ite13 (Distance + Mach)	7,023,176	40,402,638	1.07046	0.06466	-0.05948
ite14 (Distance + Mach)	7,729,280	44,485,066	1.07048	0.06460	-0.05943
ite15 (Distance + Mach)	12,243,163	70,559,034	1.07357	0.06430	-0.06189
ite16 (Distance + Mach)	14,246,766	81,848,987	1.07344	0.06422	-0.06188
ite17 (Distance + Mach)	15,704,381	90,133,089	1.07336	0.06418	-0.06181
ite18 (Distance + Mach)	16,572,243	95,071,016	1.07342	0.06417	-0.06182
ite19 (Distance + Mach)	23,462,103	134,864,961	1.07473	0.06404	-0.06297
ite20 (Distance + Mach)	25,963,210	149,000,809	1.07462	0.06402	-0.06294
ite21 (Distance + Mach)	27,687,592	158,861,434	1.07457	0.06400	-0.06296
ite22 (Distance + Mach)	28,702,161	164,628,045	1.07462	0.06400	-0.06300
ite23 (Distance + Mach)	40,904,253	235,017,809	1.07549	0.06392	-0.06381
ite24 (Distance + Mach)	45,391,256	260,457,584	1.07532	0.06390	-0.06372
ite25 (Distance + Mach)	48,284,598	276,999,225	1.07530	0.06389	-0.06374
ite26 (Distance + Mach)	49,936,010	286,411,899	1.07528	0.06388	-0.06374
ite27 (Distance + Mach)	81,121,147	465,956,049	1.07603	0.06380	-0.06446

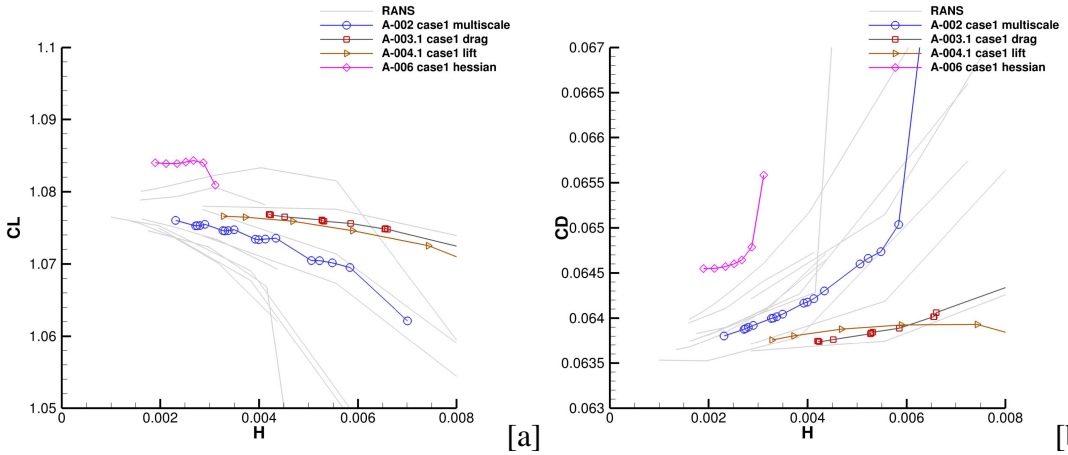
The numerical results — in terms of lift, drag, and moment coefficients calculated with respect to the moment reference center given in Table 6.1 — converged as the mesh was refined. The corresponding lift and drag coefficients, calculated using the L2 norm, are shown in Figure 6.1 with and without adap_back module. The black lines show calculations with an approach that uses the previous mesh level instead of projecting onto a finer surface mesh. This method can introduce small errors on the surface, especially near the outboard section of the leading edge. However, the overall aerodynamic forces remain largely unchanged, and the drag coefficient clearly improves as the mesh is refined. The comparison of the results with the other committee members can be found in Figure 6.2. The results indicated with A-002 are the ones obtained with HEMLAB with adap_back feature. Details of the skin friction coefficient and the corresponding surface skin friction lines are displayed in Figure 6.3 and 6.4. The flow exhibits a small separation bubble at the wing–body junction and a minor

reverse flow region at the trailing edge. Vorticity contours at several x–locations, along with the final anisotropic adapted mesh, are presented in Figure 6.5 and 6.6. The adapted mesh clearly captures the boundary layer wake. As computed earlier, pressure coefficients and skin friction coefficients were obtained on several sections of the wing, as presented in Figure 6.7. Figure 6.8 shows the convergence of the residuals with iteration number, and the numerical method converges to machine precision.



[a] [b]

Figure 6.1 : Convergence of lift coefficient [a] and drag coefficient [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ around NASA CRM high-lift system compared with several other workshop benchmark data.



[a] [b]

Figure 6.2 : Convergence of lift coefficient [a] and drag coefficient [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ around NASA CRM high-lift system compared with several other workshop benchmark data and HEMLAB (A-002) results.

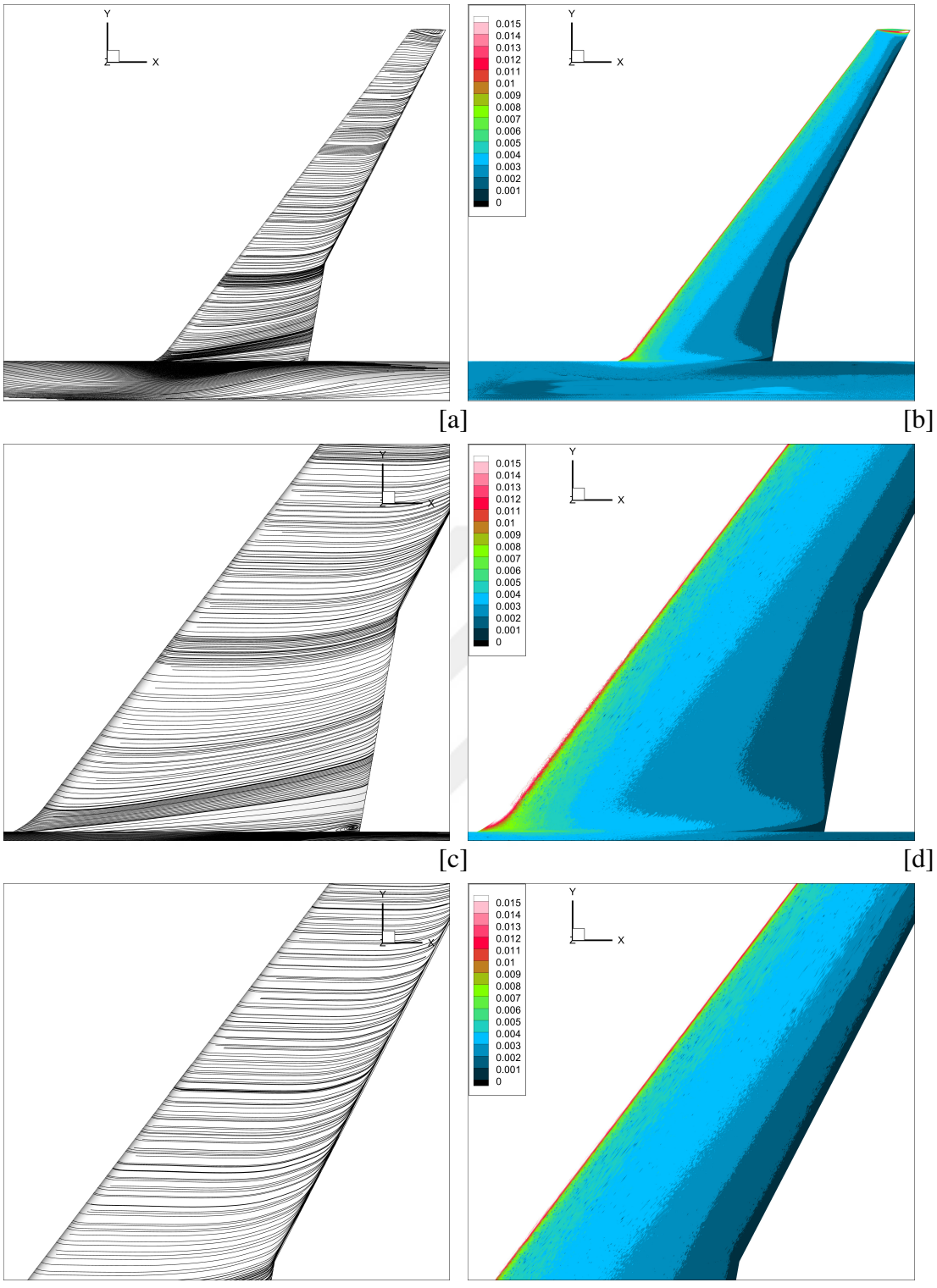
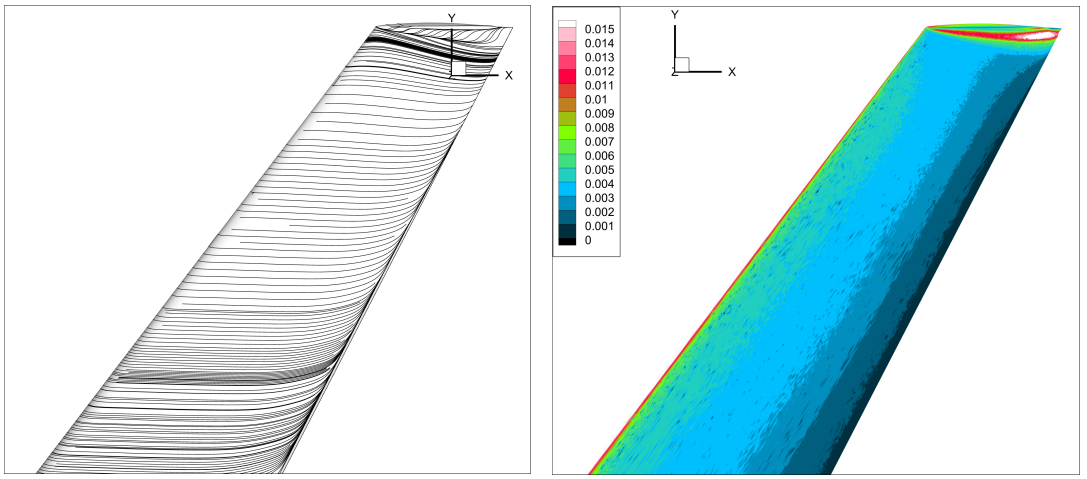
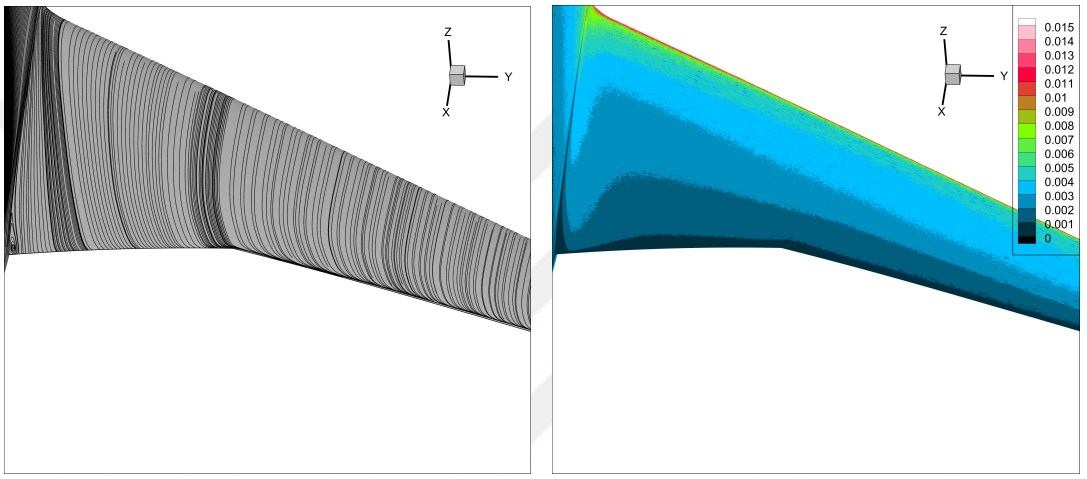


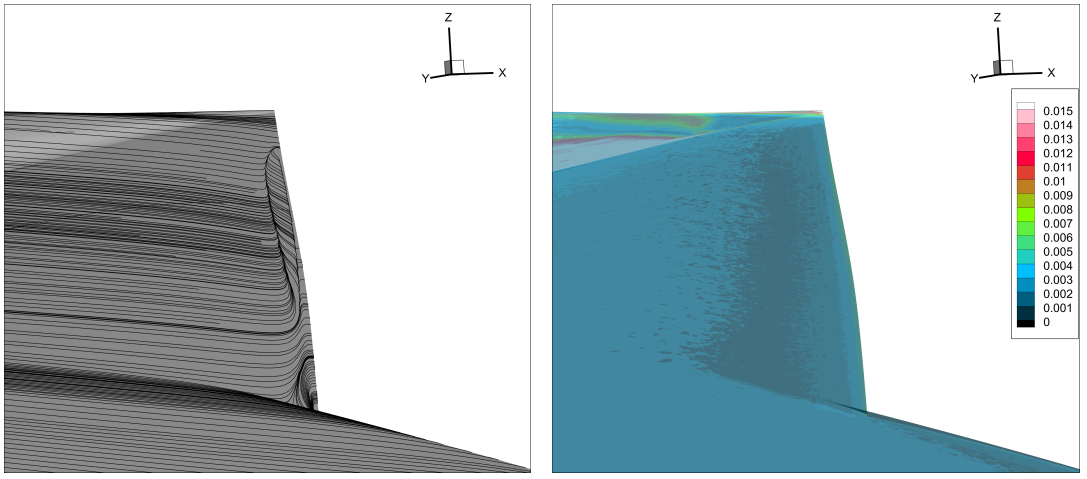
Figure 6.3 : Skin friction lines and contours around CRM-HL Wing Body configuration for View1 [a]&[b], View2 [c]&[d], and View3 [e]&[f].



[g] [h]



[i] [j]



[k] [l]

Figure 6.4 : Skin friction lines and contours around CRM-HL Wing Body configuration for View4 [g]&[h], View5 [i]&[j], and View SP1.1 [k]&[l]. (continued).

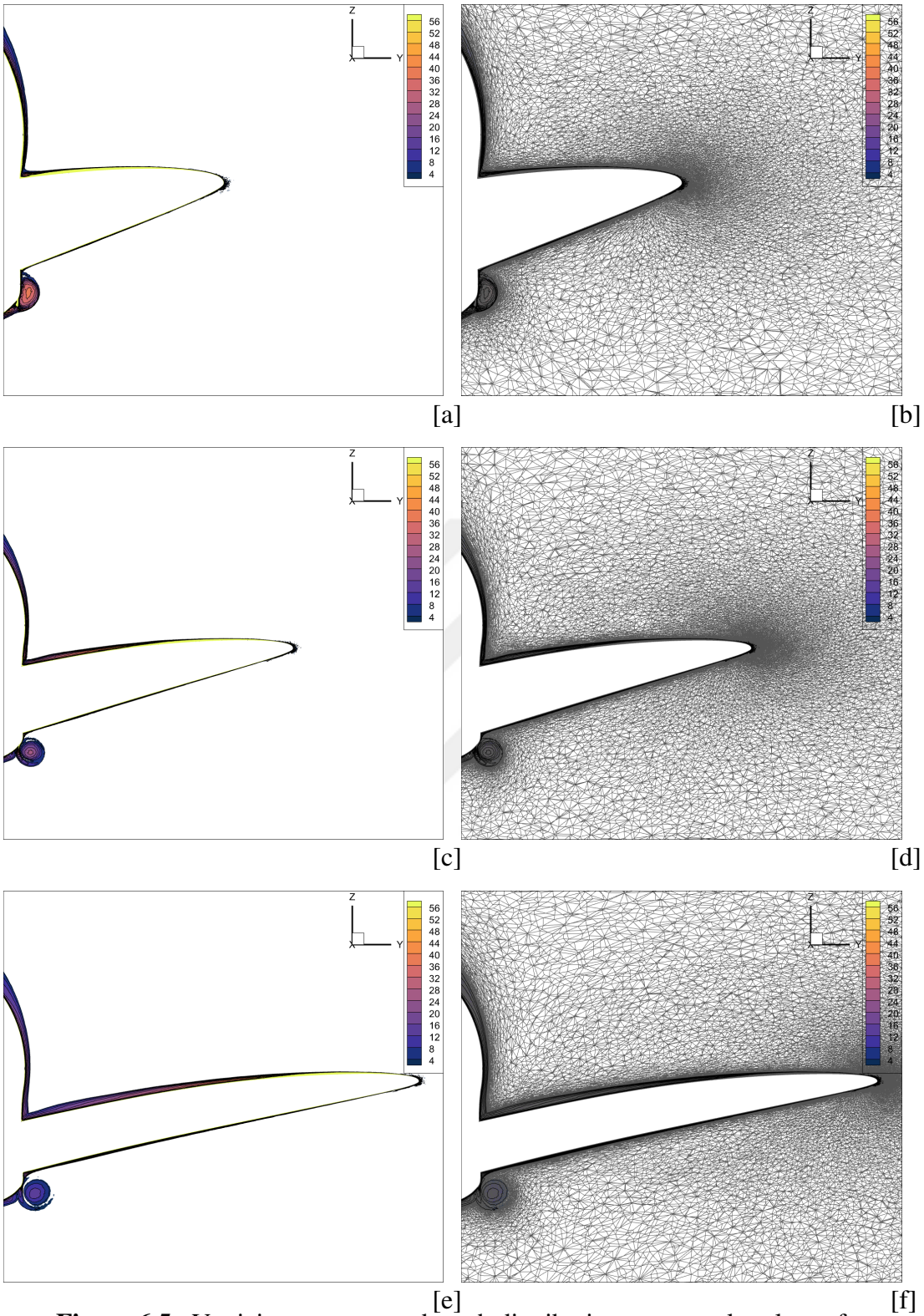
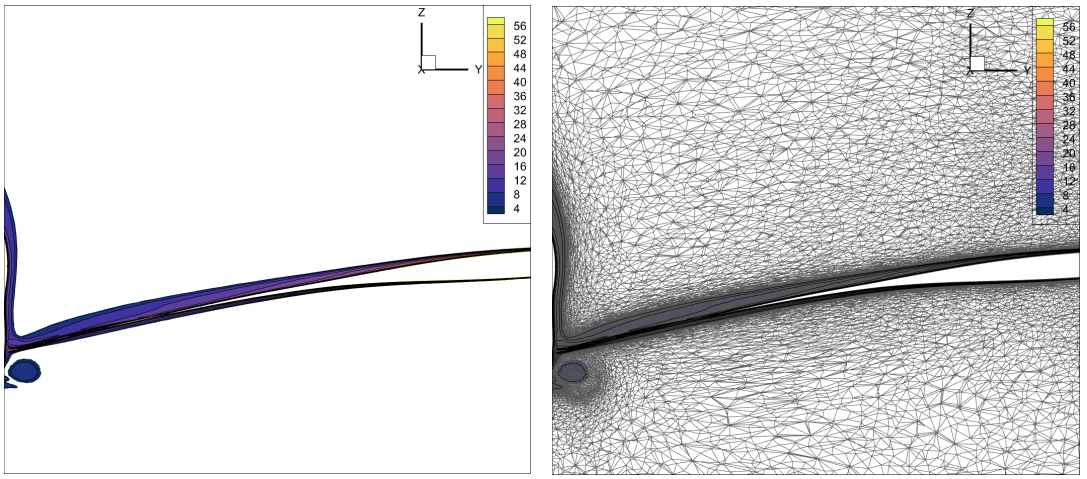
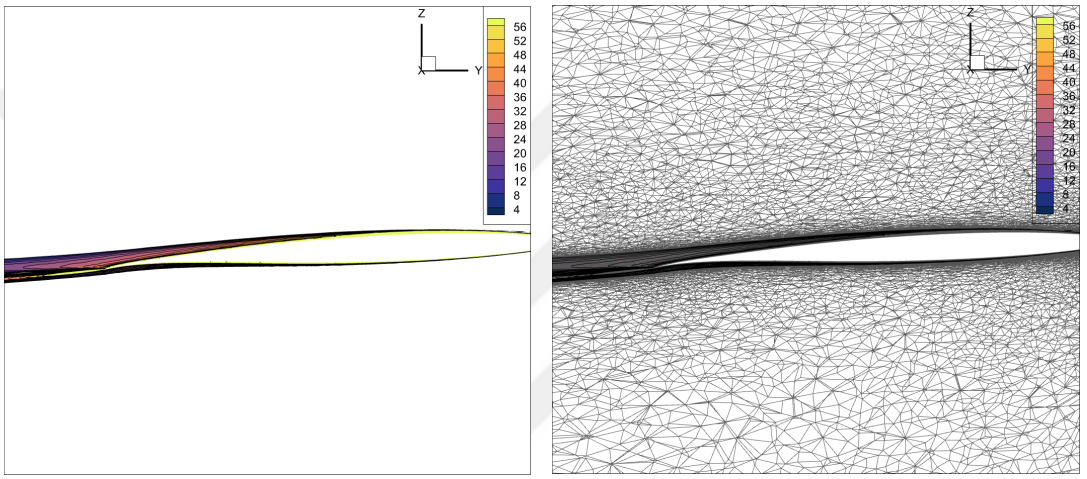


Figure 6.5 : Vorticity contours and mesh distribution on several *x*-planes for CRM-HL Wing Body configuration for View11 (x=1095 inch) [a]&[b], View12 (x=1185 inch) [c]&[d], and View13 (x=1275 inch) [e]&[f].

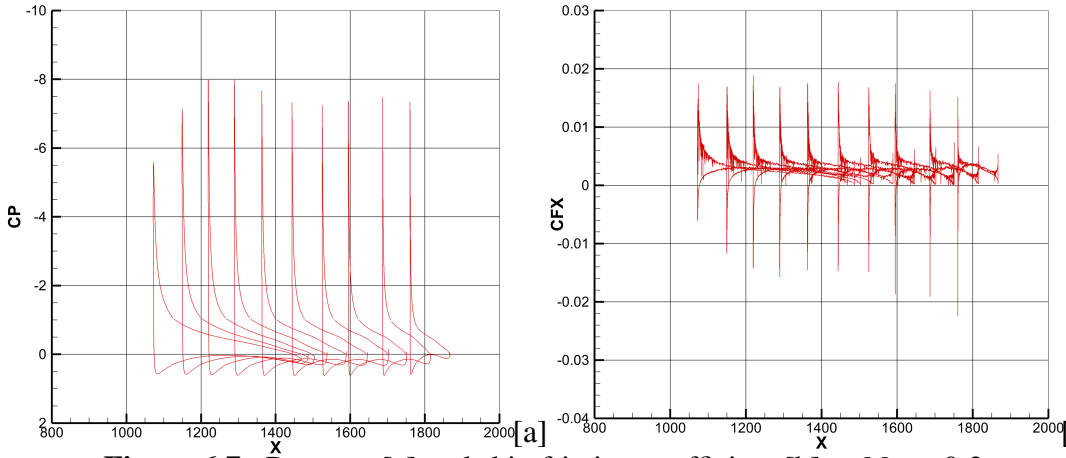


[g] [h]



[i] [j]

Figure 6.6 : Vorticity contours and mesh distribution on several x-planes for CRM-HL Wing Body configuration (continued) for View14 (x=1485 inch) [g]&[h] and View15 (x=1735 inch) [i]&[j].



[a] [b]

Figure 6.7 : Pressure [a] and skin friction coefficient [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ around NASA CRM high-lift system compared with several other workshop benchmark data.

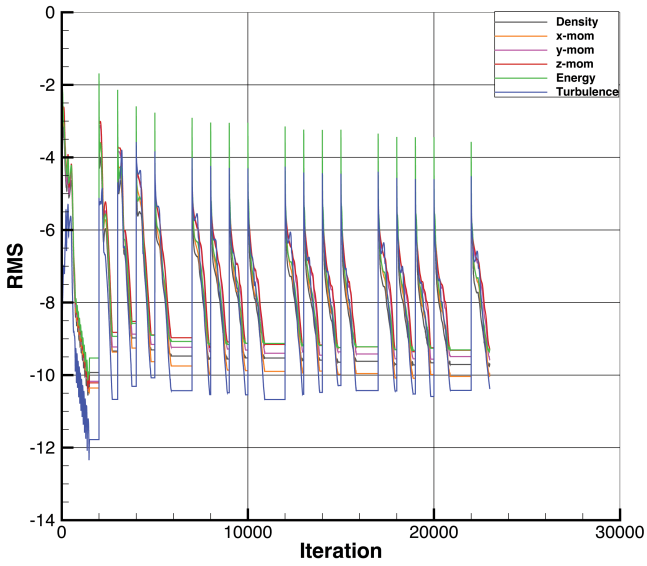


Figure 6.8 : Convergence of residual values with iteration number for each refinement level for CRM-HL Wing Body configuration.

After the turbulence verification model (Case-1), geometric effects were investigated during the workshop at $Re = 5.9 \times 10^6$, $Pr = 0.72$, $M_\infty = 0.2$, and $T_\infty = 518.67^\circ R$. The calculations were performed similarly to the previous case, using a 10-level refinement to construct the boundary layer mesh. However, this time the calculations employed a Mach-only sensor instead of a Mach+distance sensor. This adjustment allowed to capture better results on the wake region behind the flap brackets and to minimize cross-diffusion effects within highly anisotropic boundary layer meshes. In the solver, PETSc's Scalable Nonlinear Equations Solvers (SNES) with a line search algorithm is used, rather than the previous PETSc Krylov Subspace Methods (KSP). This robust nonlinear solver is crucial for achieving machine-precision convergence at high angles of attack in under 100 Newton iterations. The initial CFL number was set to 10 and increased by a factor of 1.125 each time the residual vector was reduced by an order of magnitude.

The initial calculation was conducted at a 6° angle of attack. Subsequent calculations used the alpha continuation method, starting from the previously converged angle. Calculations were performed at 6°, 10°, 14°, 16°, and 17.7°, with the computed aerodynamic loads presented in Figure 6.9 and Table 6.3.

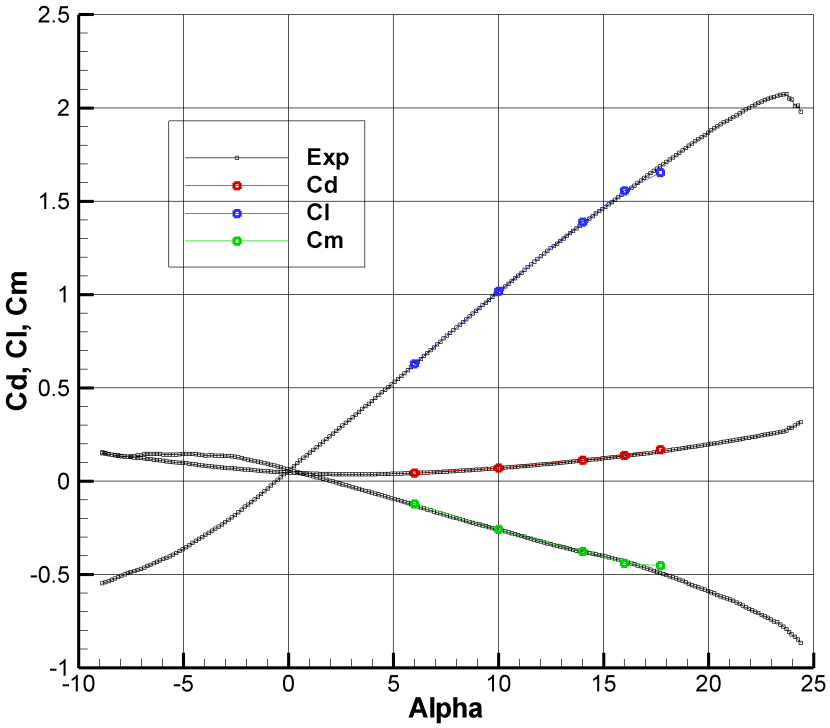


Figure 6.9 : Comparison of computed aerodynamic loads for case 2.2 CRM-HL configuration.

Table 6.3 : Aerodynamic loads for Case 2.2.

Alpha	Nodes	C_L	C_D	C_M
6.0°	19,620,617	0.629437	0.044356	-0.121557
10.0°	20,259,448	1.018601	0.071470	-0.258877
14.0°	20,537,562	1.388209	0.111988	-0.376624
16.0°	21,084,912	1.555762	0.137893	-0.439587
17.7°	21,179,493	1.653319	0.168740	-0.453178

The results showed good agreement up to 16°. However, at 17.7°, early leading-edge separation caused deviations in the computed aerodynamic loads, preventing further calculations at higher angles of attack. The computed skin friction coefficients and streamlines are shown in Figure 6.10, where the pizza slice–shaped separation pattern is clearly visible in Figures 6.10-[e-f]. A comparison of pressure coefficients with ONERA experimental data are provided in Figure 6.11, 6.12 and 6.13 for sections A–I, showing good agreement. Strong pressure and skin friction coefficient peaks were observed near the leading edge at the outboard section of the main wing, and the separation pattern is also apparent on the adapted surface mesh (Figure 6.14). The residual convergence of the PETSc SNES solver is shown in Figure 6.15, indicating

some convergence difficulty at 10° (attached flow) for the fine mesh, despite the final density residual dropping to 1×10^{-8}. At high angles of attack, there was a significant increase in the turbulence model residual, even with a constant scaling factor of 1×10^{6} for the turbulence variable.

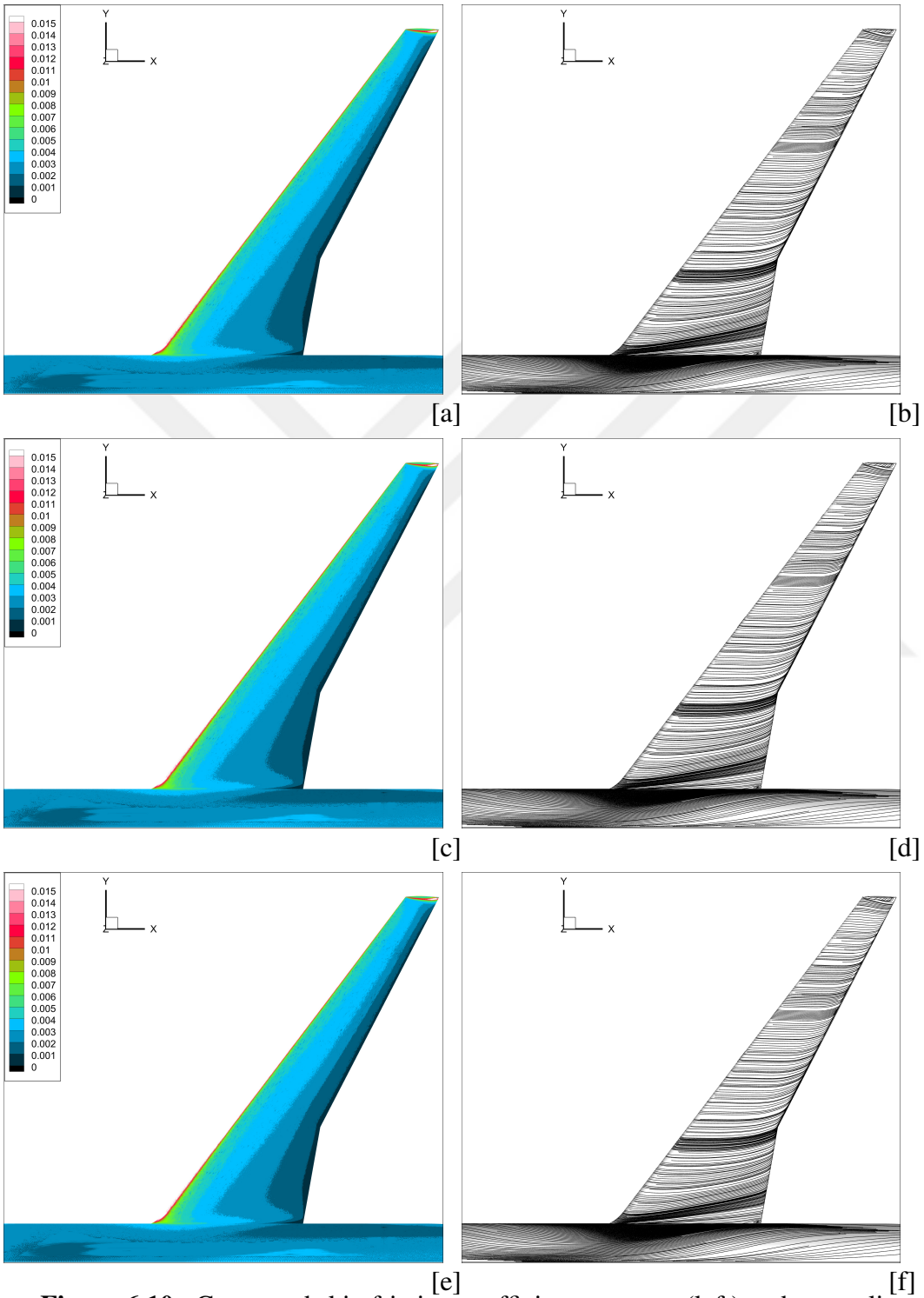


Figure 6.10 : Computed skin friction coefficient contours (left) and streamlines (right) at $\alpha = 6°$ [a-b], 10° [c-d], and 17.7° [e-f] for case2.2 configuration.

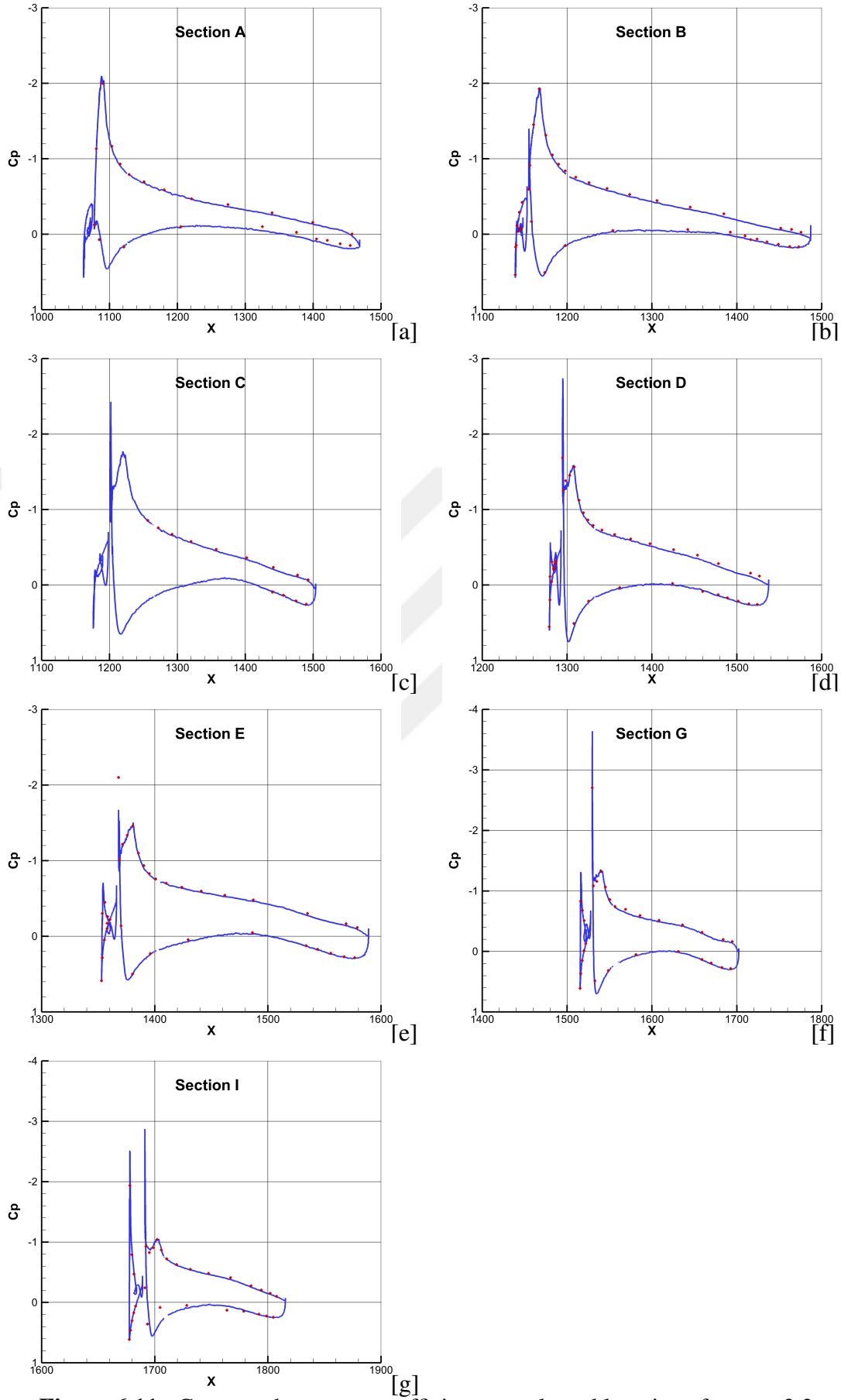


Figure 6.11 : Computed pressure coefficients at selected locations for case 2.2 configuration for $\alpha = 6°$.

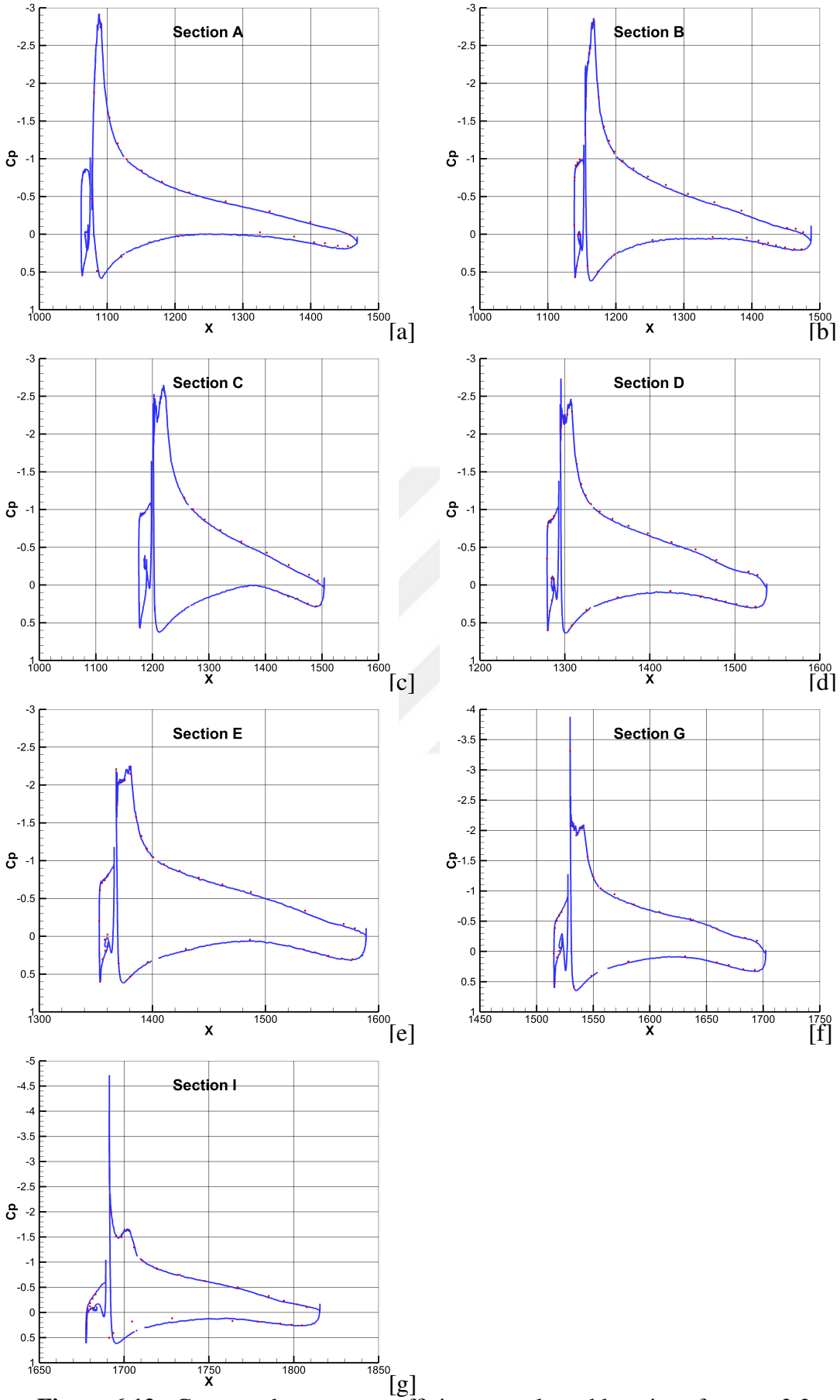


Figure 6.12 : Computed pressure coefficients at selected locations for case 2.2 configuration for $\alpha = 10°$.

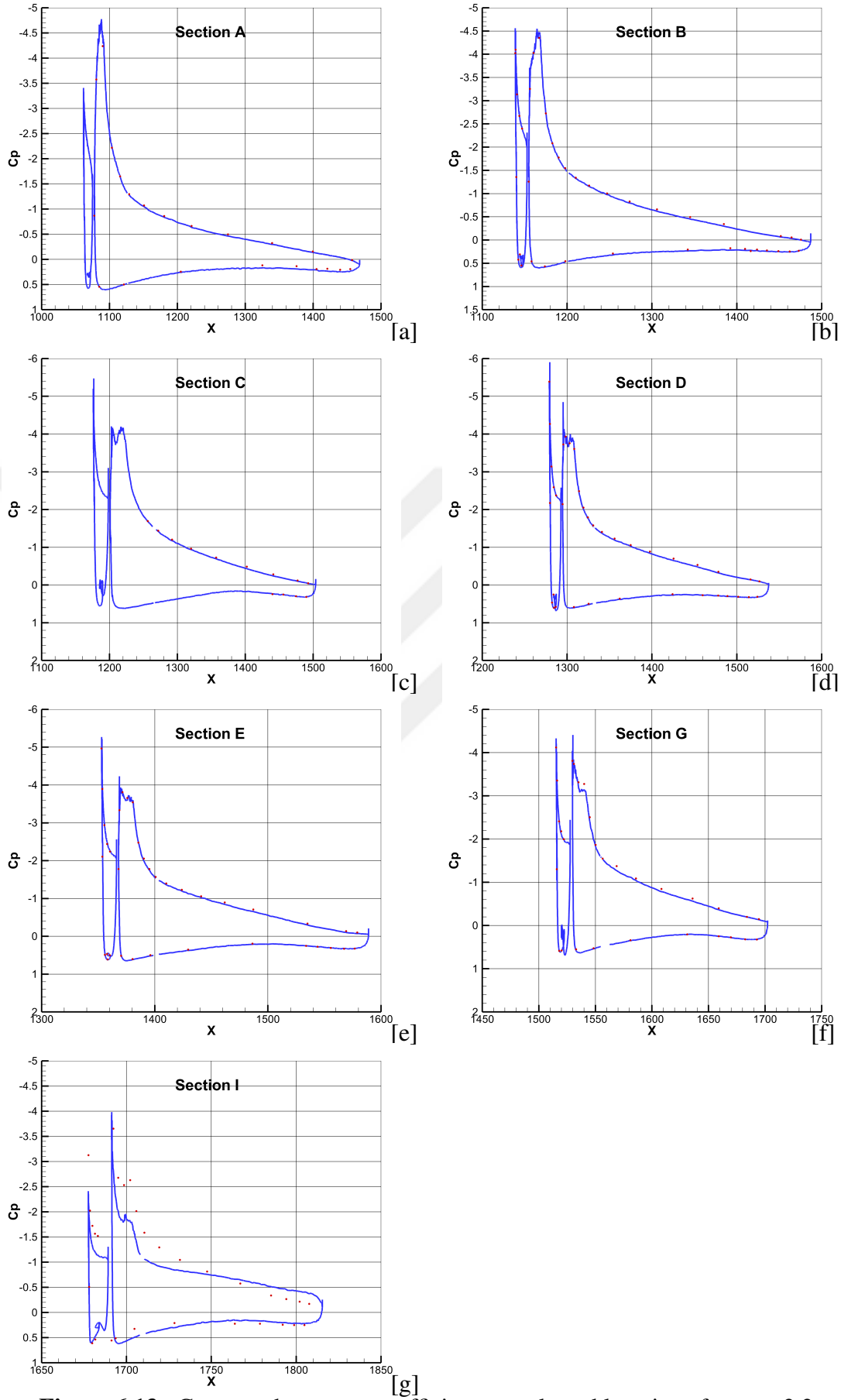
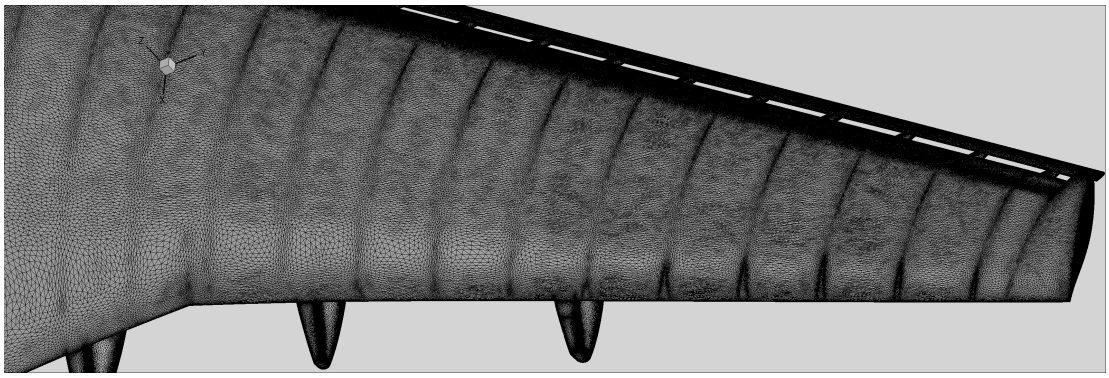
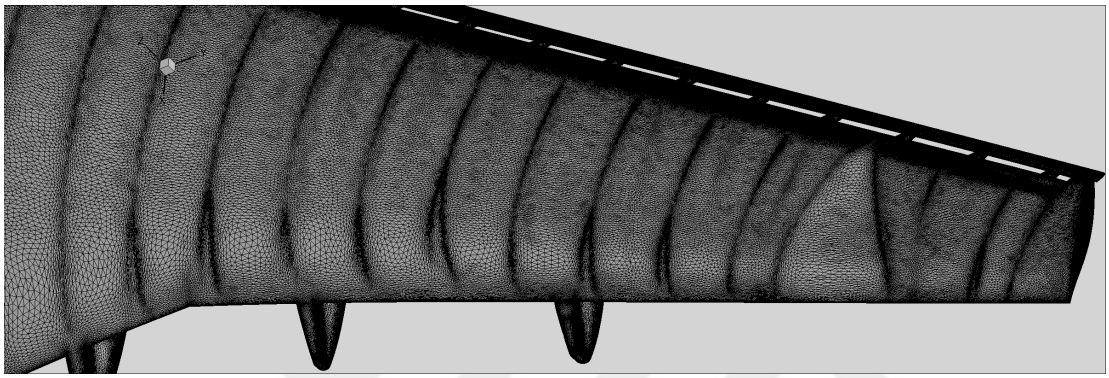


Figure 6.13 : Computed pressure coefficients at selected locations for case 2.2 configuration for $\alpha = 17.7°$.

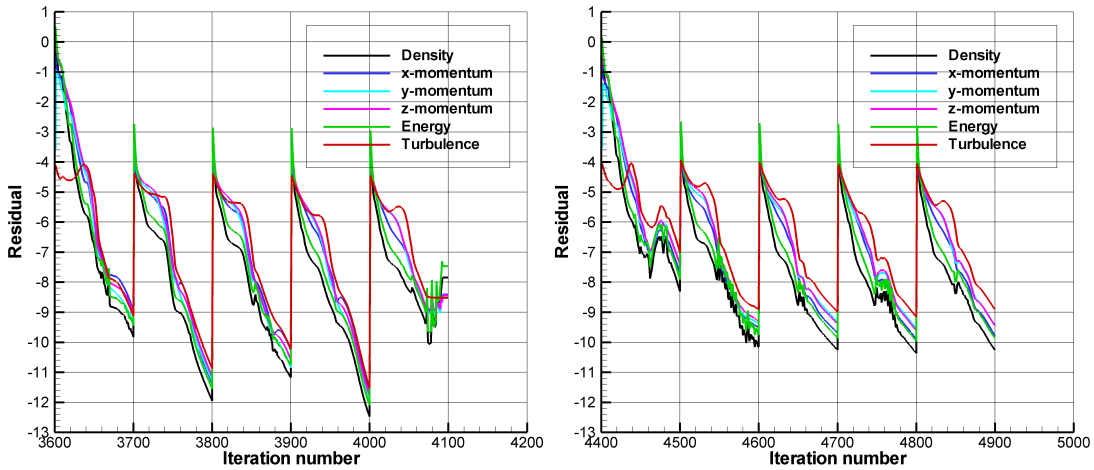


[a]



[b]

Figure 6.14 : Adapted surface meshes at $\alpha = 10^\circ$ [a] and $\alpha = 17.7^\circ$ [b] for case2.2 configuration.



[a] [b]

Figure 6.15 : Residual convergence at $\alpha = 10^\circ$ [a] and $\alpha = 17.7^\circ$ [b] for case2.2 configuration (view13).

The computed vorticity contours at the $x = 1275$ plane, shown in Figure 6.16, along with the adapted meshes, demonstrate how mesh adaptation effectively captures the boundary layer and wake regions.

In this study, it has been aimed to understand the early separation pattern observed in the numerical results. A similar separation pattern was observed in Boeing results obtained with the GGNS solver [58] and the EPIC anisotropic mesh adaptation library [33]. However, this pattern was absent in the impulsively started CFX converged results with isentropic refinement, where initial separation patterns reattached with increased refinement. In the ONERA experiments, the separation pattern was associated with a lambda-type trailing edge separation, which was also seen in the present adapted surface meshes (Figure 6.14). At 17.7°, the separation pattern began with a double wake structure (Figure 6.17), with wake shedding from both the inboard and outboard sides of the flap bracket, leading to a wider wake region. The outboard wake was either absent or very weak for attached flap bracket wakes, suggesting potential issues with turbulence modeling affecting the prediction of double wake behavior. Another possibility involves the negative Spalart-Allmaras turbulence model, where numerical inaccuracies might result in excessive negative $\hat{\nu}$ values, causing locally very low turbulence viscosity (laminarization) and early leading-edge flow separation. Similar trends were also observed in results from other committee members, as compared in Figure 6.18. This figure shows results for $\alpha = 17.7^\circ$ obtained using HEMLAB [a], ANSYS [b], and INRIA [c]. Here, the early separation pattern is only visible in HEMLAB's results. On the other hand, Figures 6.19 [a] and [b] present results obtained at $\alpha = 16^\circ$ using HEMLAB and Boeing, respectively. While both results show similar flow characteristics at this angle of attack, early separation appears in Boeing's results at $\alpha = 16.5^\circ$, as shown in Figure 6.18 [d].

On the solver side, it has been examined both the numerical discretization and the PETSc SNES nonlinear solver. For PETSc SNES, the second-order Jacobian is constructed exactly and used it directly for matrix-vector multiplication instead of the matrix-free approach; however, it did not alter the final flow pattern. In the HEMLAB solver, it has been used the Green-Gauss method rather than the unweighted least squares method and reduced the upwinding parameter β for both the Navier-Stokes equation and the negative Spalart-Allmaras turbulence model. While this resulted in a slight delay in the formation of the separation pattern, further reduction in the β parameter led to negative density values. At this point it is not possible to address mesh quality issues due to a lack of access to the INRIA pyAMG library source code.

Another possibility is that mesh adaptation should be performed after several Newton iterations rather than on the final converged solution, as the flow pattern may change significantly and the adapted mesh might not properly represent the solution.

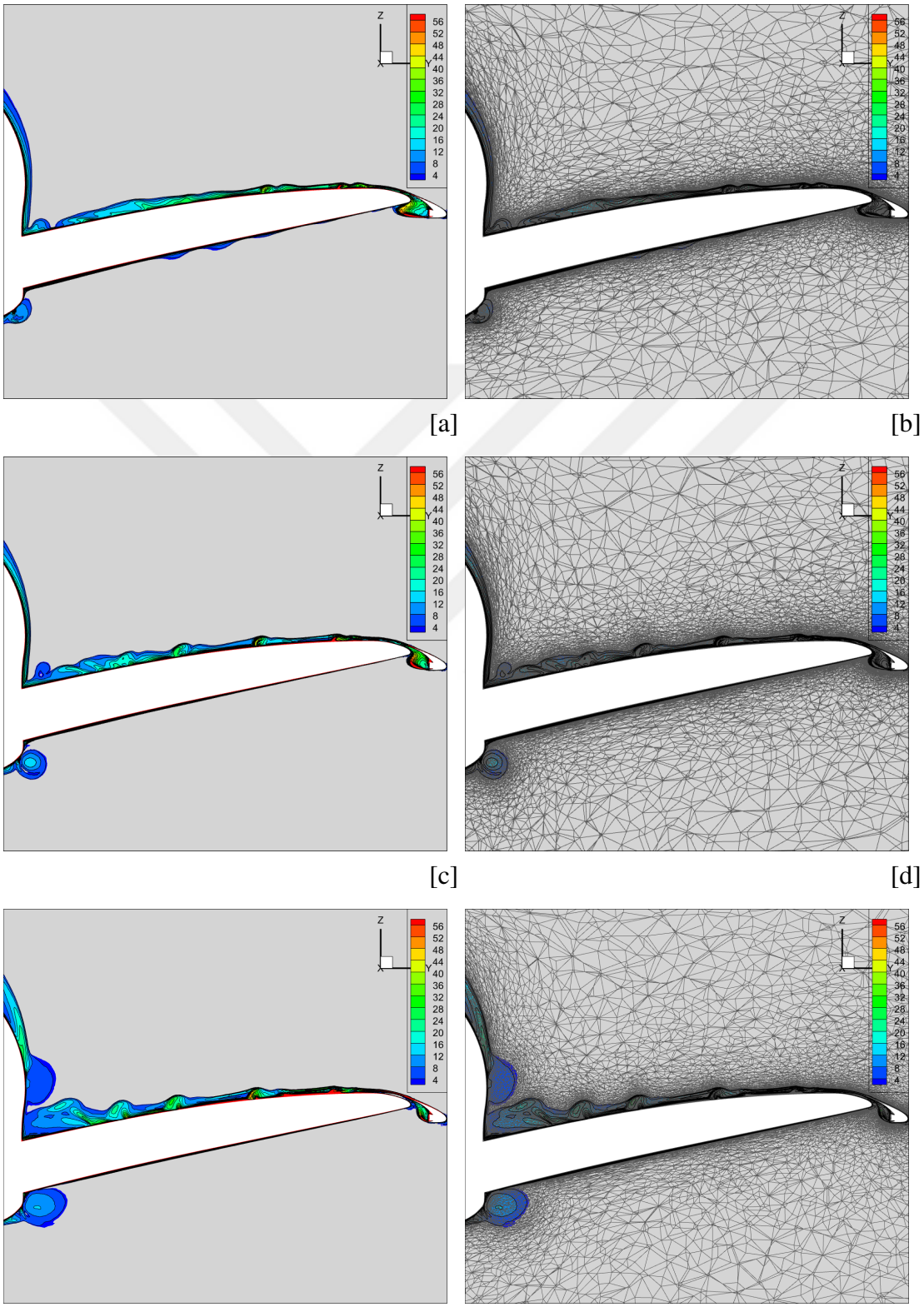
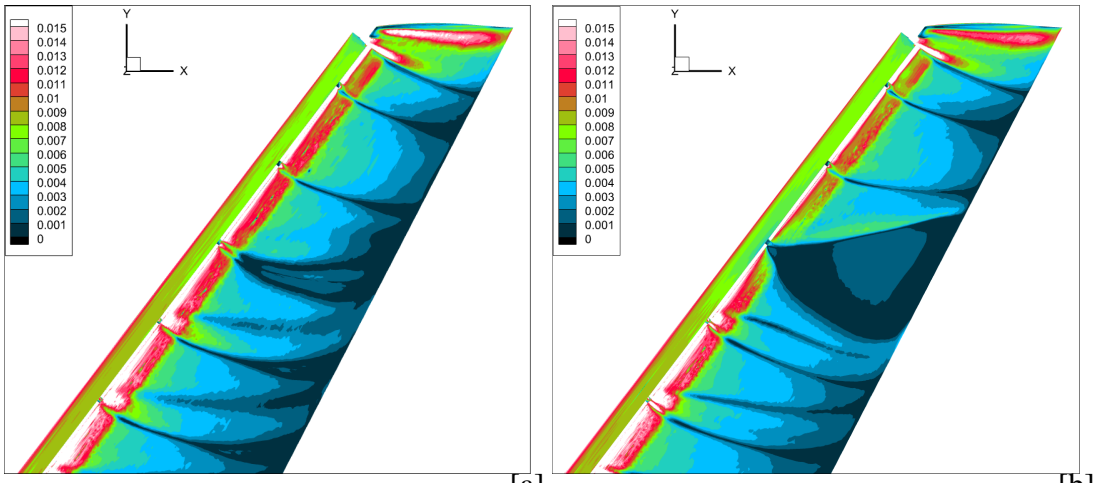
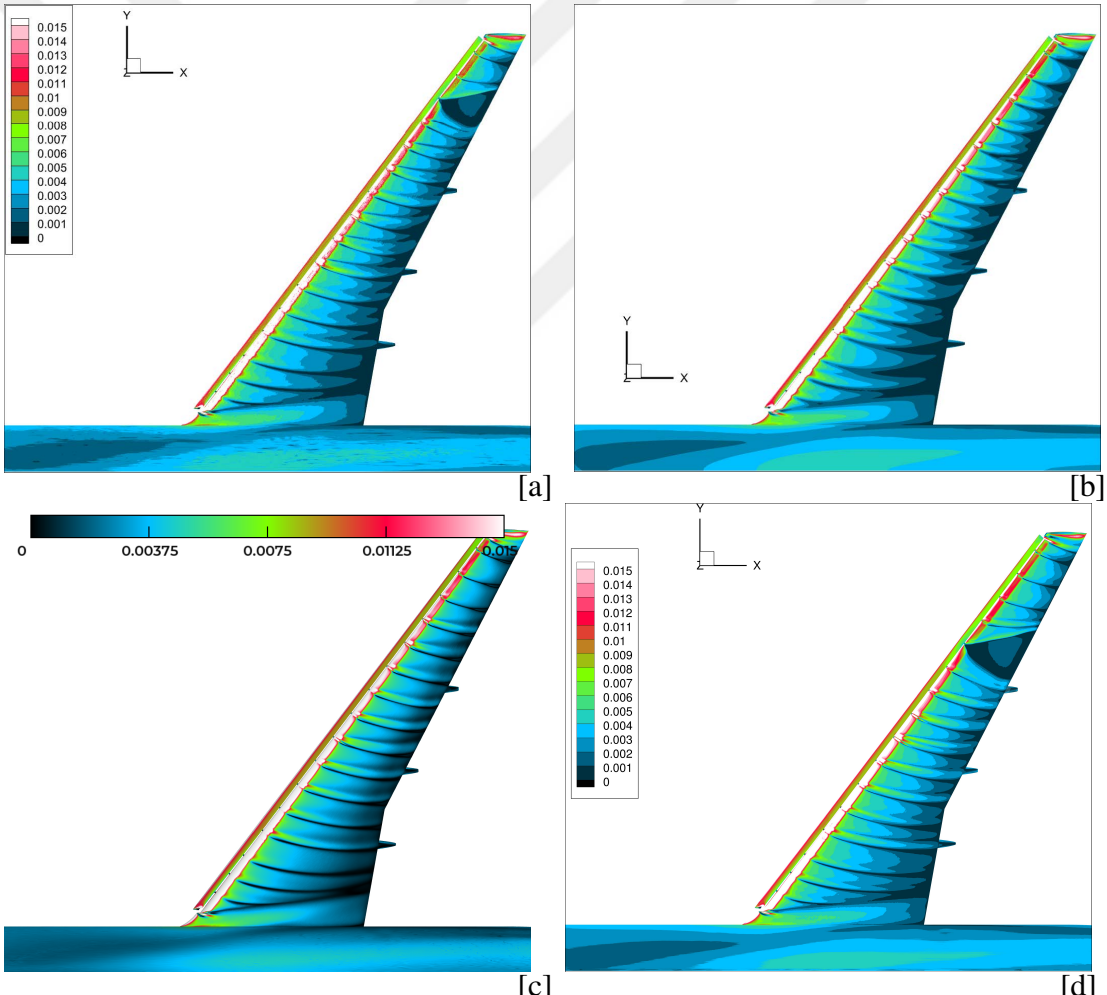


Figure 6.16 : Computed vorticity contours (left) and adapted mesh (right) at $\alpha = 6^\circ$ [a-b], $\alpha = 10^\circ$ [c-d], and $\alpha = 17.7^\circ$ [e-f] for case2.2 configuration (view13).



[a] [b]

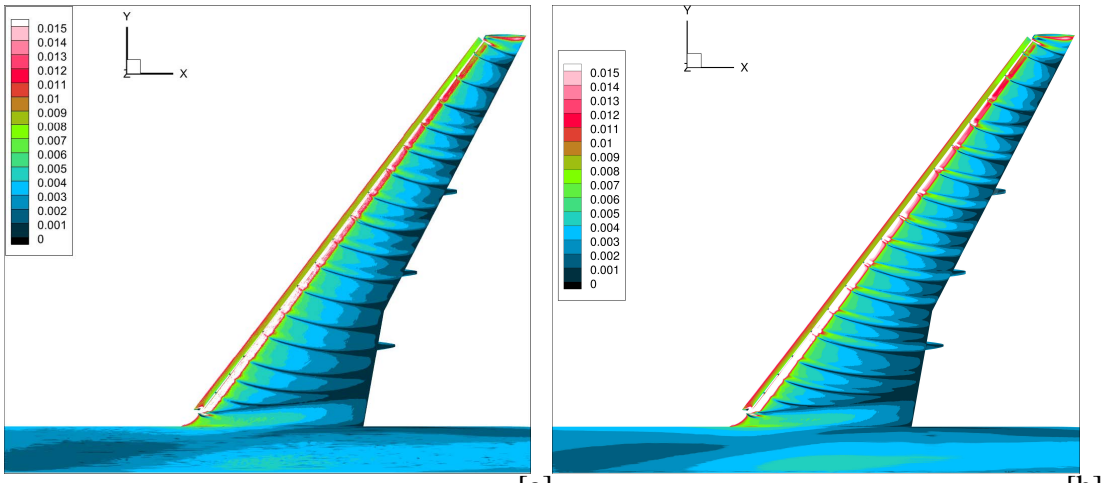
Figure 6.17 : Computed transient [a] and final converged [b] skin friction contours at $\alpha = 17.7°$ for case2.2 CRM-HL Wing Body configuration (view13).



[a] [b]

[c] [d]

Figure 6.18 : Computed skin friction contours at $\alpha = 17.7°$ with HEMLAB [a], A-006 (TFG Results with ANSYS) [b], A-004 (TFG Results with INRIA) [c] and $\alpha = 16.5°$ A-003 (TFG Results with BOEING) [d] case 2.2 CRM-HL Wing Body configuration [11].



[a] [b]

Figure 6.19 : Computed skin friction contours at $\alpha = 16^{\circ}$ with HEMLAB [a] , A-003 (TFG Results with BOEING) [b] case 2.2 CRM-HL Wing Body configuration [11].

7. NUMERICAL RESULTS FOR REFINE

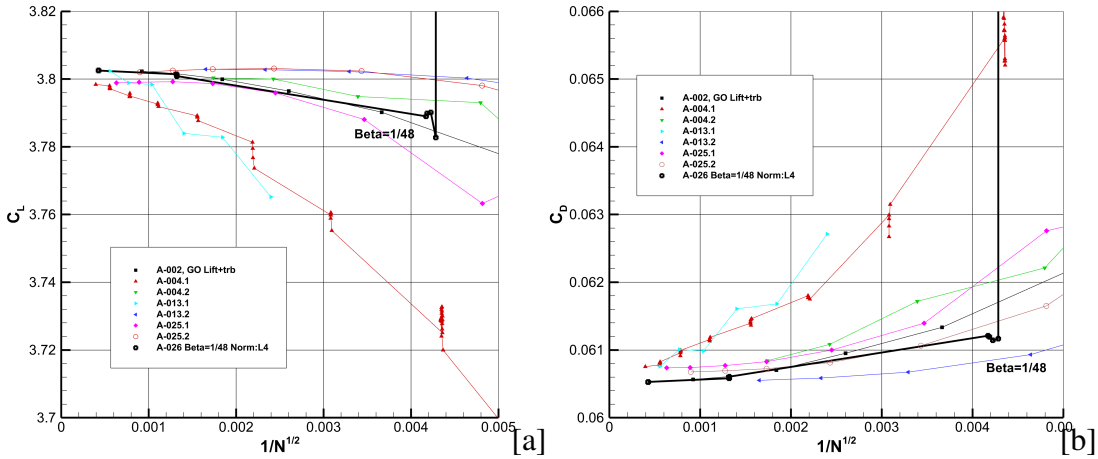
As discussed in Chapter 5, the 2D high-lift configuration was also examined using the NASA REFINE mesh adaptation framework. This tool provides a robust and highly parallelizable anisotropic adaptation methodology, capable of capturing complex flow features with fewer mesh elements than uniform refinement. The simulation was carried out at a freestream Mach number $M_\infty = 0.2$, Reynolds number $Re = 5 \times 10^6$, Prandtl number $Pr = 0.72$, and an angle of attack $\alpha = 16^\circ$. The adaptation procedure utilized a hybrid refinement strategy — starting with distance-based sensors for establishing wall resolution and progressing to Mach-based sensors. The adaptation statistics are provided in Table 7.1. A total of 17 refinement levels were executed, and the corresponding values for aerodynamic coefficients C_L, C_D, and C_M are presented. Notably, convergence in aerodynamic performance is achieved by the final iterations, where the lift coefficient reaches 3.8026 and the drag coefficient stabilizes at 0.06052, both closely matching the reference value $C_L = 3.802952$ from the reference results presented in [95]. As illustrated in Figure 7.1, the computed lift and drag coefficients demonstrate monotonic convergence with mesh refinement. Similar to the pyAMG-HEMLAB framework, early adaptations (from 0th iteration to the 4th iteration) focus on resolving the near-wall regions with the distance sensor. Subsequent iterations (from 5th iteration to the 17th iteration), using the Mach sensor, enhance the resolution in flow regions with strong gradients such as shock waves and wakes. The refinement levels beyond 13th level iteration show minimal change in the computed aerodynamic coefficients, confirming mesh-independent results. The deviation from the reference lift coefficient at the final refinement level is less than 0.01%, demonstrating the high fidelity achieved through REFINE. Figure 7.2 present pressure coefficients and skin friction distribution where Figure 7.4 shows the Mach number, and entropy contours. The high-resolution mesh captures boundary-layer behavior, separation zones, and wake dynamics with clarity. Especially, the entropy contours in Figure 7.4(b) show well-resolved entropy layers downstream of the trailing edge, which are indicative of accurate shock capturing and viscous interaction

modeling. In Figure 7.3, the residual convergence and force history across refinement iterations are plotted. The solver exhibits stable and monotonic residual reduction across all adaptation stages, highlighting the robustness of the SNES-based nonlinear solver when combined with meshes generated by REFINE. Residuals show rapid decay after each adaptation step, indicating good metric compatibility and preserved mesh quality. No oscillations or stagnation are observed in the convergence history.

Figure 7.5 illustrates the mesh structures at the 8th adaptation level obtained from the REFINE [a] and pyAMG [b] frameworks. Although both methods apply the same flow sensors without any explicit boundary-layer generation routines, generated mesh by REFINE exhibits more directional stretching near the walls and wake region. This difference likely arises from internal remeshing strategies and more aggressive anisotropy enforcement during metric projection. In contrast, the pyAMG mesh appears smoother and more isotropic, particularly around the trailing edge.

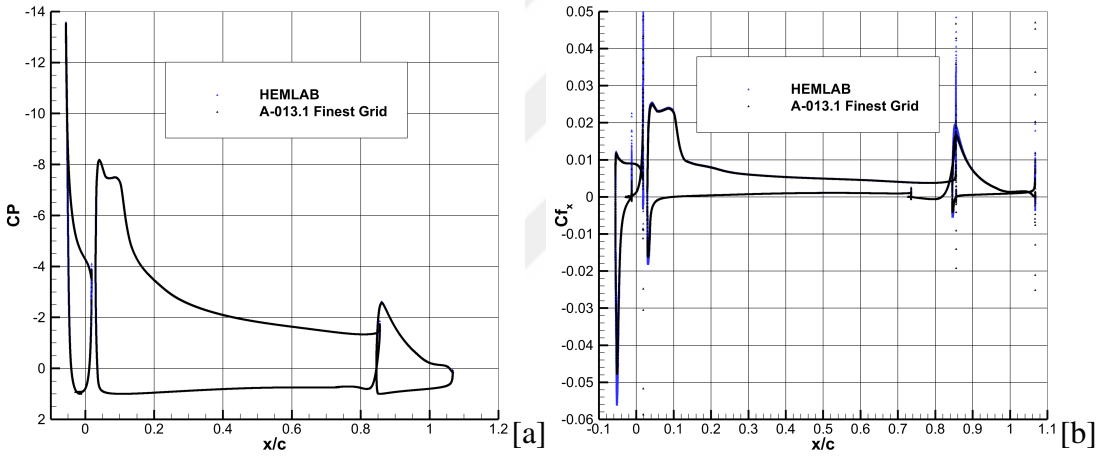
Table 7.1 : Refinement data for L4, $\alpha = 16^\circ$ and $\beta = 1/48$.

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	Number of Surface Elements	C_L	C_D	C_M
ite0 (Initial Mesh)	325,794	642,357	9,156	-	-	-
ite1 (Distance)	54,274	101,897	6,576	-	-	-
ite2 (Distance)	40,758	76,535	4,906	-	-	-
ite3 (Distance)	53,469	101,964	4,899	-	-	-
ite4 (Distance)	54,590	103,784	5,321	-	-	-
ite5 (Mach)	54,487	103,898	5,001	3.782751	0.061166	0.379569
ite6 (Mach)	56,020	107,248	4,717	3.790154	0.061144	0.380961
ite7 (Mach)	57,090	109,418	4,683	3.789870	0.061194	0.380922
ite8 (Mach)	57,458	110,180	4,646	3.788992	0.061210	0.380755
ite9 (Mach)	573,351	1,137,745	8,736	3.800873	0.060605	0.382748
ite10 (Mach)	574,940	1,138,977	10,677	3.801294	0.060583	0.382828
ite11 (Mach)	575,158	1,139,414	10,675	3.801299	0.060582	0.382821
ite12 (Mach)	574,558	1,138,231	10,653	3.801394	0.060582	0.382840
ite13 (Mach)	5,464,790	10,895,465	33,388	3.802502	0.060526	0.383025
ite14 (Mach)	5,477,291	10,915,256	38,466	3.802531	0.060525	0.383031
ite15 (Mach)	5,498,440	10,956,603	39,203	3.802529	0.060524	0.383032
ite16 (Mach)	5,506,133	10,971,522	39,523	3.802533	0.060524	0.383032
ite17 (Mach)	22,498,153	44,913,272	80,182	3.802596	0.060520	0.383042
Ref [95]				3.802952	0.060529	-



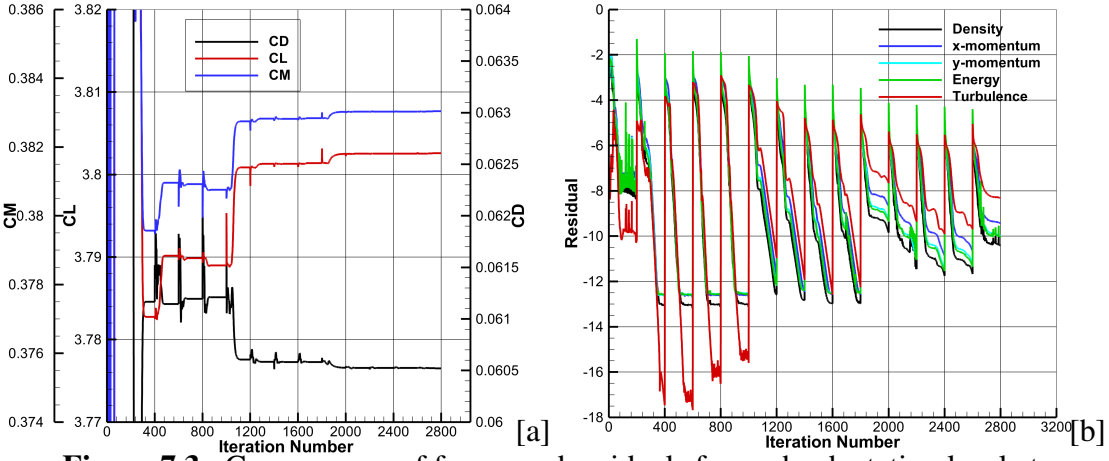
[a] [b]

Figure 7.1 : Convergence of lift coefficient [a] and drag coefficient [b] at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ around a two-dimensional high-lift system compared with several other workshop benchmark data, N corresponds to the total number of vertices.



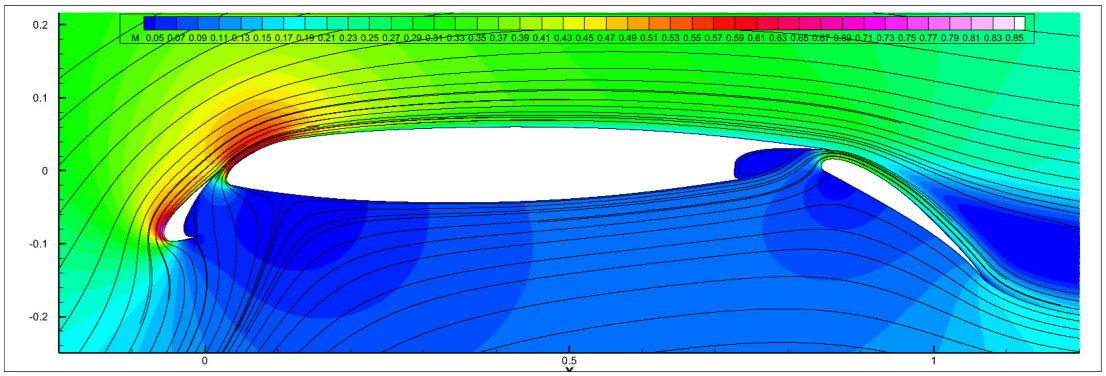
[a] [b]

Figure 7.2 : Computed pressure [a] and x-component of skin friction [b] coefficients.

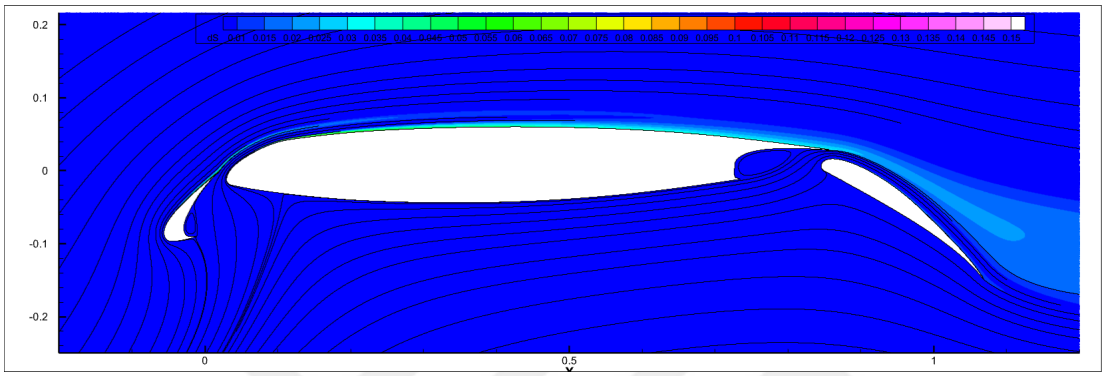


[a] [b]

Figure 7.3 : Convergence of forces and residuals for each adaptation level at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ with L4 norm, $\beta = 1/48$, and $\varepsilon = 0.001$.

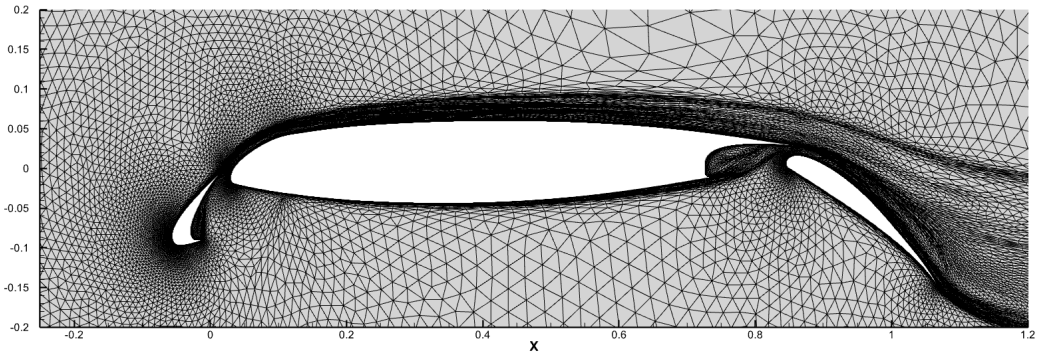


[a]



[b]

Figure 7.4 : Computed Mach [a] and entropy [b] contours at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16^\circ$ around a two-dimensional high-lift system.



[a]

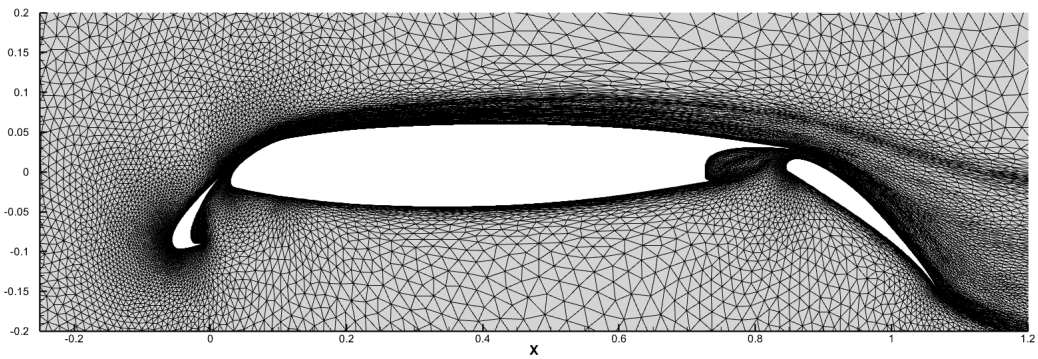
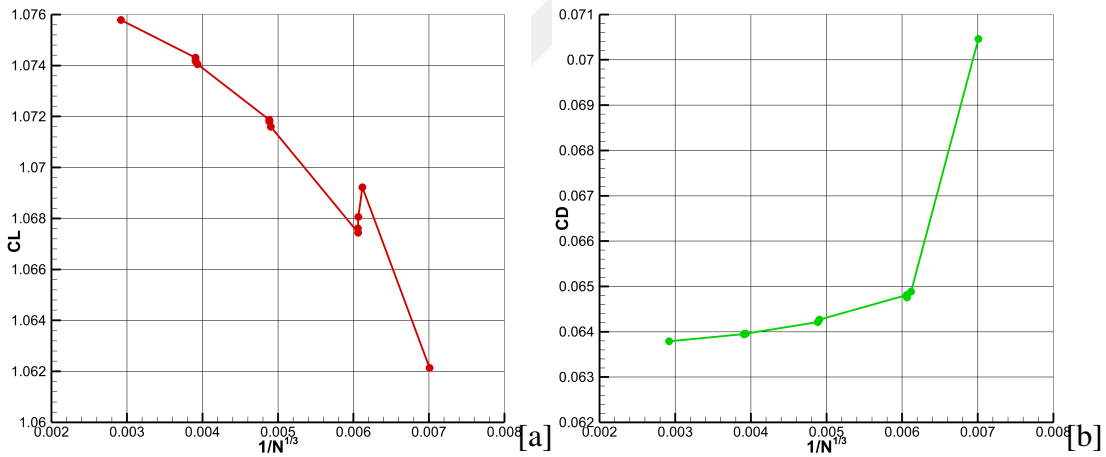


Figure 7.5 : Mesh adaptation at $M_\infty = 0.2$, $Re = 5 \times 10^6$, $Pr = 0.72$, and $\alpha = 16$¶[b] around 2D-CRM at 8th refinement level with REFINE [a] and pyAMG [b].

In addition to the 2D studies, three-dimensional (3D) simulations were performed using the NASA-REFINE framework for the CRM-HL Wing-Body (CRMHL-WB) configuration. The simulations were conducted at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$. Distinct from the pyAMG approach, the REFINE simulations employed the L_4 norm for error estimation, enhancing refinement focus on critical flow features. Furthermore, both the initial mesh and solution file were sourced from the 10th refinement level of the pyAMG framework. Subsequent refinement levels proceeded exclusively with the Mach sensor to effectively resolve wall-bounded layers and freestream phenomena. The adaptation statistics, detailed in Table 7.2, show a systematic increase in mesh resolution, achieves over 40 million vertices and 224 million elements by iteration 23. The lift and drag coefficients converged to $C_L = 1.07578$ and $C_D = 0.06379$, respectively, demonstrating close alignment with workshop benchmarks and confirming mesh independence, as illustrated in Figures 7.6 and 7.7. Residual histories (Figure 7.8) exhibit smooth and rapid convergence across all refinement levels, validating the efficiency and stability of the REFINE adaptation process when integrated with a SNES-based solver. To further analyze surface flow characteristics, Figures 7.10 and 7.11 provide skin-friction lines and near-wall contours from multiple viewpoints. These visualizations shows detailed flow separation, reattachment, and complex shear layer interactions, particularly around slat and flap cove regions, as well as at the wing-body junction. Cross-sectional vorticity contours, presented in Figures 7.12 and 7.13, highlight the development of wingtip vortices, wake roll-up, and trailing-edge shedding across several axial planes. The alignment of refined mesh regions with areas of high vorticity confirms effective metric-driven targeting of critical flow features. Grid overlays in selected subfigures demonstrate the anisotropic refinement strategy, ensuring high element density within vortical cores and wake structures. Finally, Figure 7.9 presents the computed pressure and skin friction coefficients over the CRM-HL configuration for several sections.

Table 7.2 : Refinement data for L2, $\alpha = 7.05^\circ$ and $\beta = 1/3$.

Refinement Level (Sensor Function)	Number of Vertices	Number of Elements	Number of Surface Elements	C_L	C_D	C_M
ite0 (Initial Mesh)	10,630,135	62,254,785	-	-		
ite1 (Distance)	3,170,048	18,313,665	-	-	-	
ite2 (Distance)	2,119,011	12,146,828	-	-	-	
...	...	...	...	...	...	
ite9 (Distance)	2,861,878	16,455,322	-	-	-	
ite10 (Distance)	2,904,264	16,696,572	442,766	1.06213	0.07046	-0.05423
ite11 (Mach)	4,364,826	23,557,552	1,268,264	1.06922	0.06489	-0.05752
ite12 (Mach)	4,486,723	24,129,314	1,349,978	1.06806	0.06475	-0.05628
ite13 (Mach)	4,494,562	24,191,567	1,361,317	1.06761	0.06478	-0.05569
ite14 (Mach)	4,492,884	24,200,267	1,334,163	1.06744	0.06481	-0.05549
ite15 (Mach)	8,466,110	46,130,018	2,220,800	1.07160	0.06427	-0.05947
ite16 (Mach)	8,569,112	46,677,188	2,255,172	1.07181	0.06422	-0.05981
ite17 (Mach)	8,579,941	46,692,011	2,291,436	1.07181	0.06421	-0.05972
ite18 (Mach)	8,579,795	46,669,793	2,304,239	1.07187	0.06421	-0.05982
ite19 (Mach)	16,412,920	90,489,126	3,670,843	1.07405	0.06396	-0.06191
ite20 (Mach)	16,704,761	91,814,017	3,931,316	1.07415	0.06395	-0.06206
ite21 (Mach)	16,744,639	91,959,488	3,986,137	1.07421	0.06394	-0.06202
ite22 (Mach)	16,767,908	92,024,225	4,029,809	1.07430	0.06395	-0.06205
ite23 (Mach)	40,206,567	224,244,705	7,638,331	1.07578	0.06379	-0.06364



[a] [b]

Figure 7.6 : Convergence of lift coefficient [a] and drag coefficient [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ around NASA CRM high-lift system compared with several other workshop benchmark data.

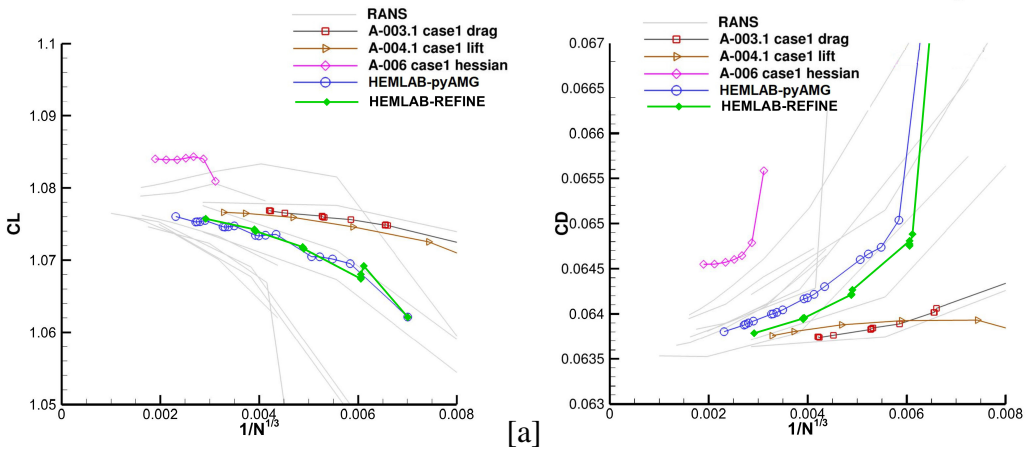


Figure 7.7 : Convergence of lift coefficient [a] and drag coefficient [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ around NASA CRM high-lift system compared with several other workshop benchmark data and HEMLAB (A-002) results.

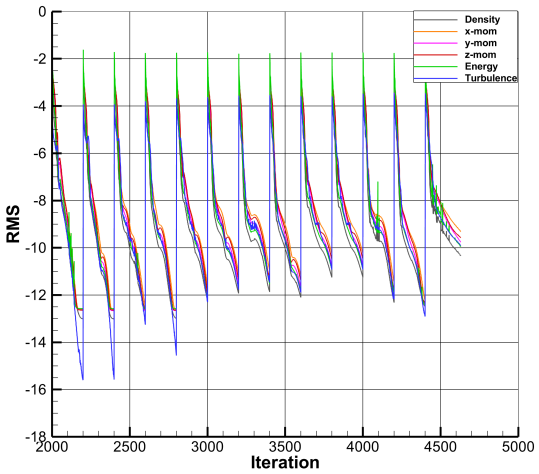


Figure 7.8 : Convergence of residual values with iteration number for each refinement level for CRM-HL Wing Body configuration.

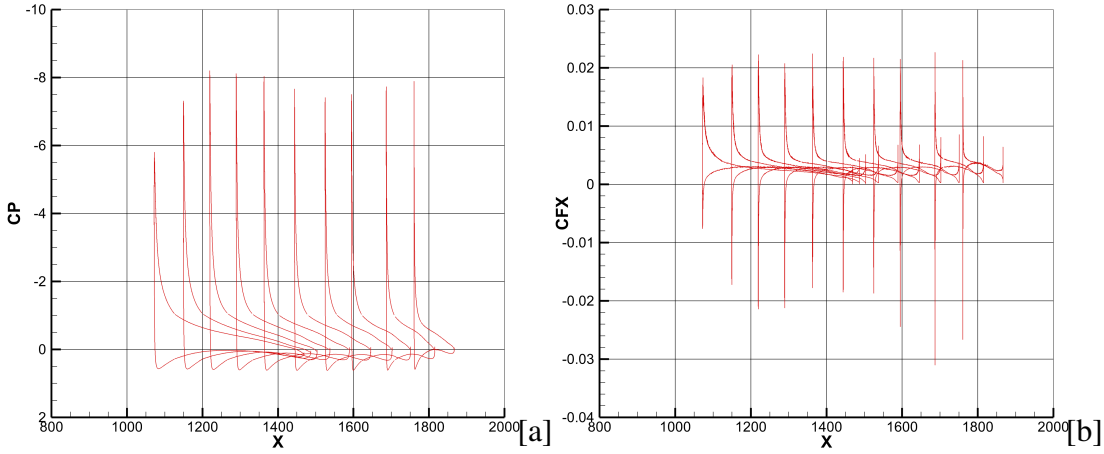


Figure 7.9 : Pressure [a] and skin friction coefficient [b] at $M_\infty = 0.2$, $Re = 5.49 \times 10^6$, $Pr = 0.72$, and $\alpha = 7.05^\circ$ around NASA CRM high-lift system compared with several other workshop benchmark data.

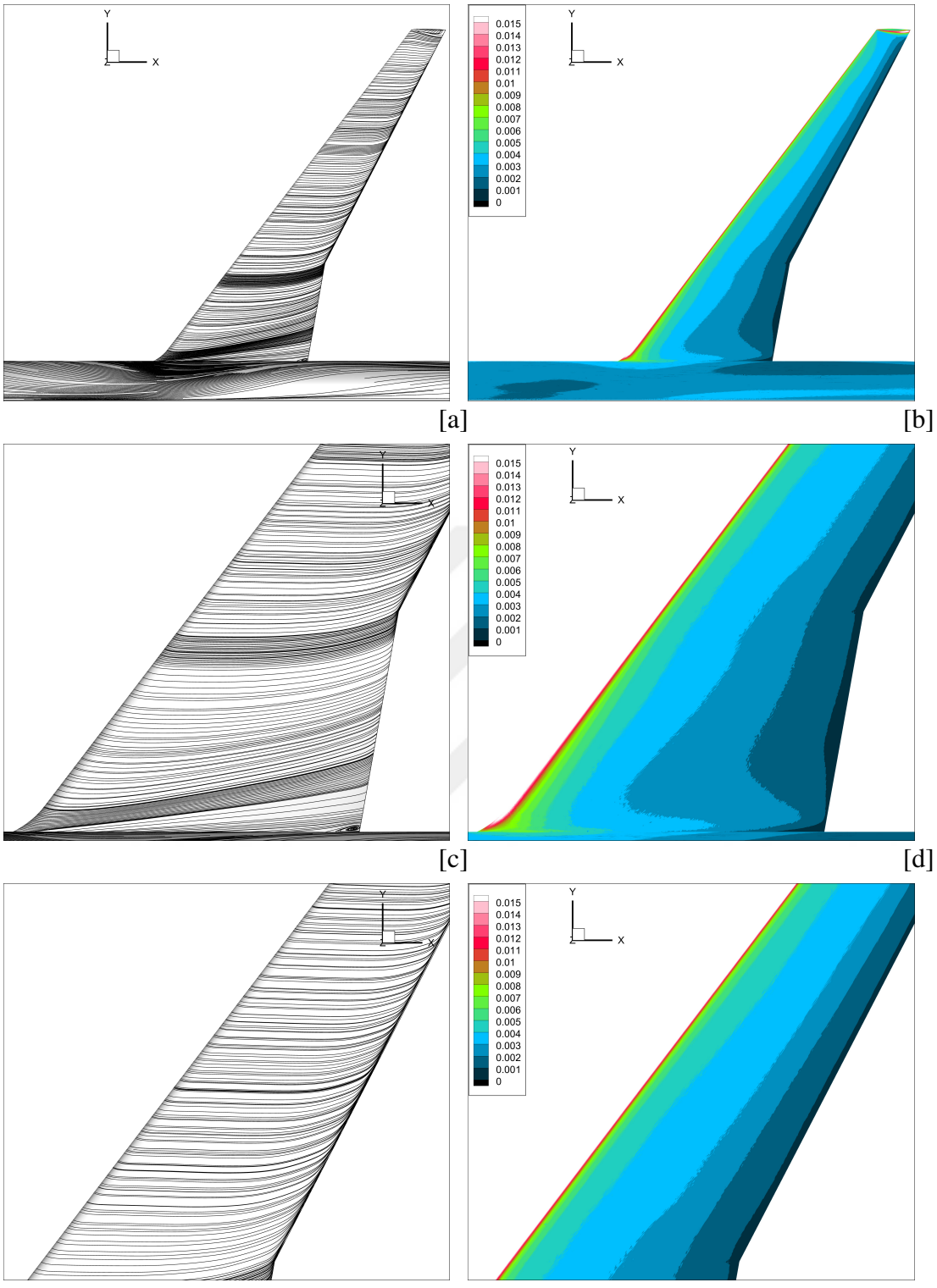


Figure 7.10 : Skin friction lines and contours around CRM-HL Wing Body configuration for View1 [a]&[b], View2 [c]&[d], and View3 [e]&[f].

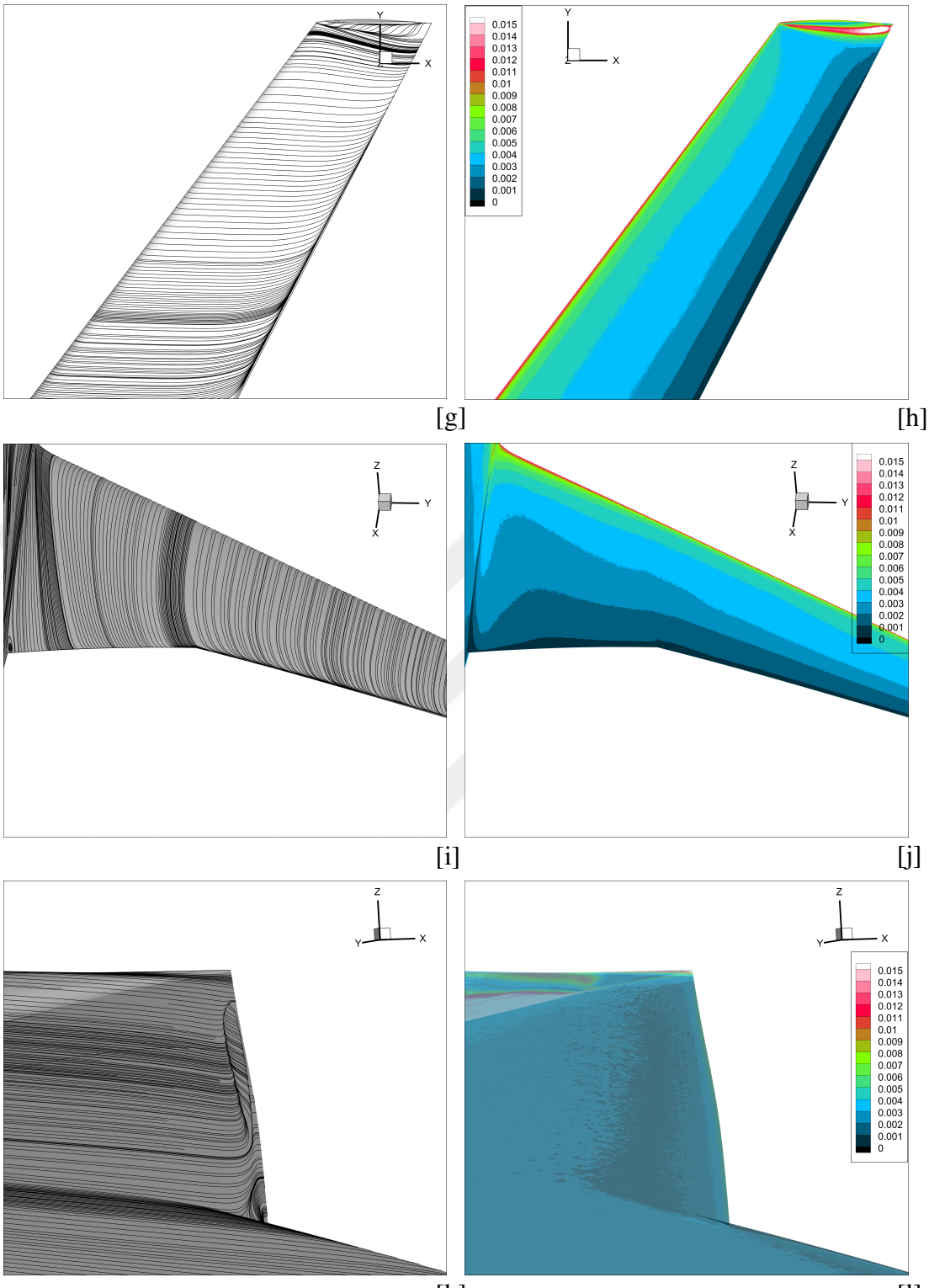
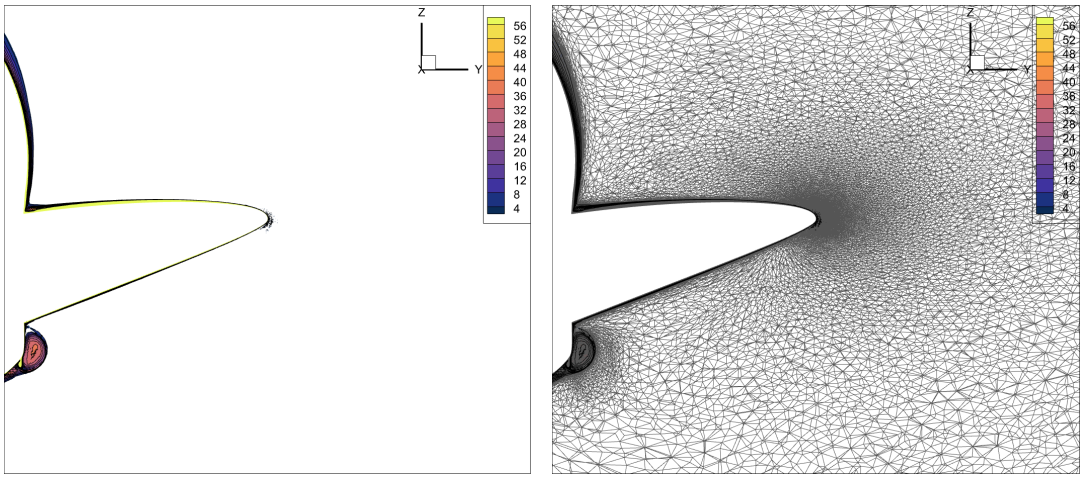
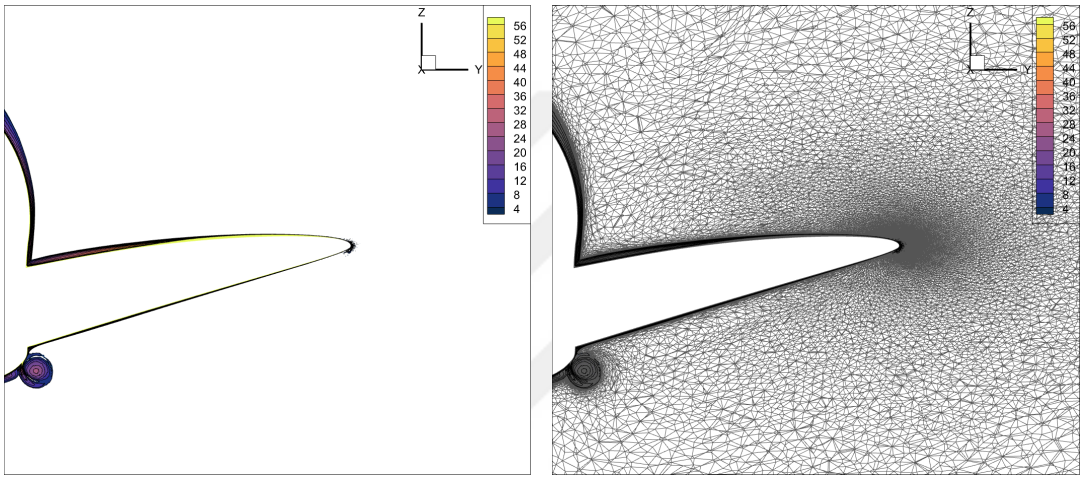


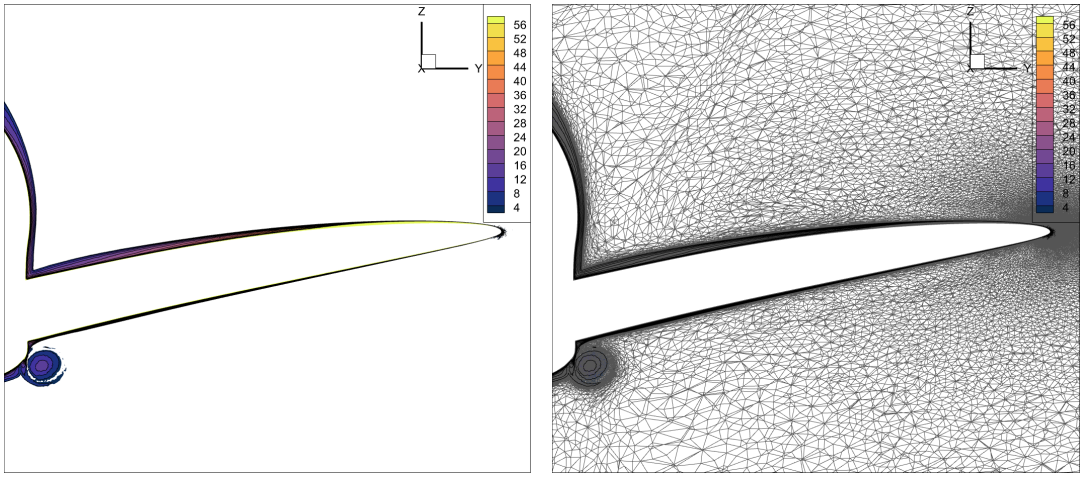
Figure 7.11 : Skin friction lines and contours around CRM-HL Wing Body configuration for View4 [g]&[h], View5 [i]&[j], and View SP1.1 [k]&[l]. (continued).



[a] [b]

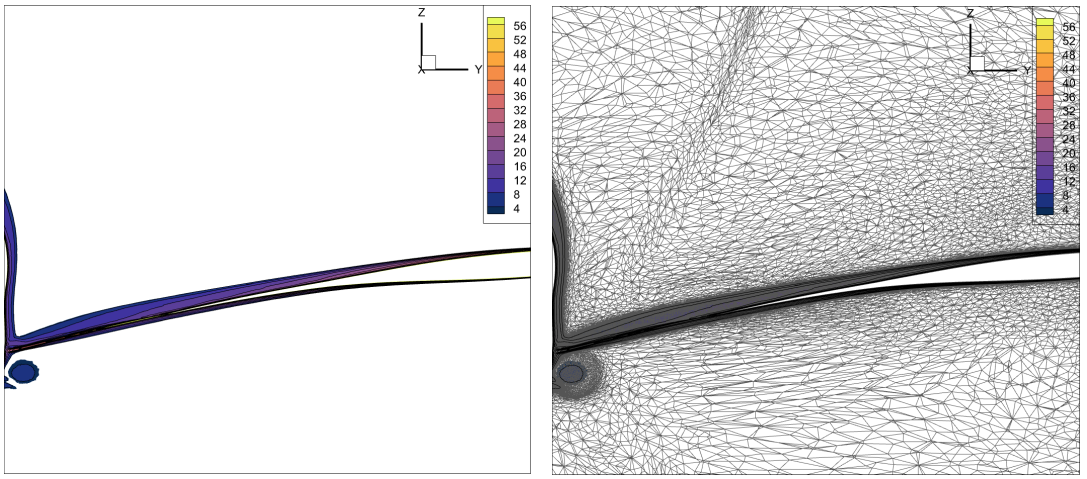


[c] [d]

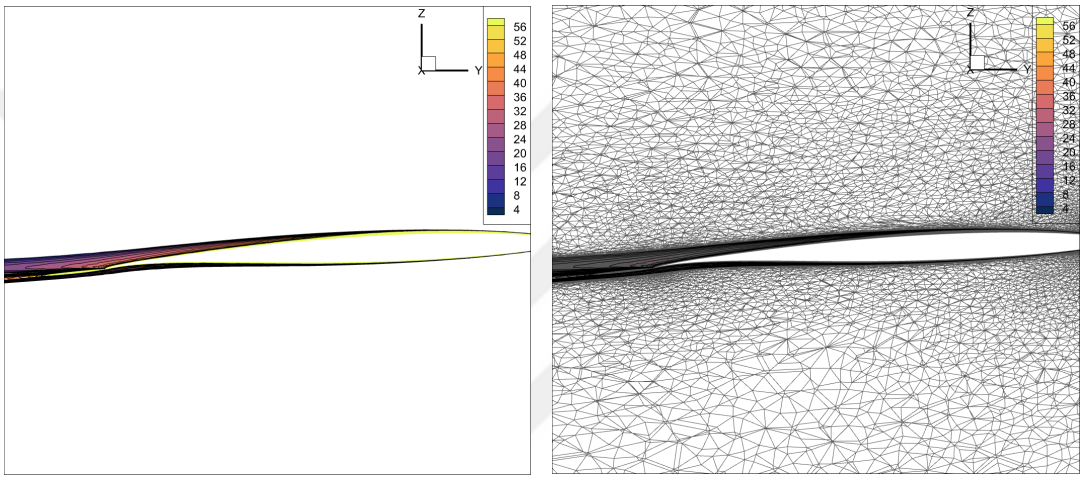


[e] [f]

Figure 7.12 : Vorticity contours and mesh distribution on several *x*-planes for CRM-HL Wing Body configuration for View11 (x=1095 inch) [a]&[b], View12 (x=1185 inch) [c]&[d], and View13 (x=1275 inch) [e]&[f].



[g] [h]



[i] [j]

Figure 7.13 : Vorticity contours and mesh distribution on several x-planes for CRM-HL Wing Body configuration (continued) for View14 (x=1485 inch) [g]&[h] and View15 (x=1735 inch) [i]&[j].

8. CONCLUSION

This study presents a detailed worked on metric-based mesh adaptation methodologies applied to the HEMLAB algorithm, with a focus on high-lift aerodynamic simulations. The research integrates advanced computational techniques to improve the accuracy and efficiency of computational fluid dynamics (CFD) solvers, particularly for complex high-lift configurations analyzed in the AIAA High Lift Prediction Workshops.

A major aspect of this research is the development and integration of metric-based anisotropic mesh adaptation strategies, which enable the refinement of computational grids in regions of high flow gradients while maintaining computational efficiency. The implementation of these techniques within the HEMLAB framework has demonstrated significant improvements in capturing flow physics, particularly in predicting lift coefficients, pressure distributions, and separation effects in high-lift aerodynamic configurations.

The numerical studies have been performed for HLPW-3, HLPW-4, and HLPW-5 test cases illustrate the impact of mesh adaptation techniques on the accuracy of aerodynamic predictions. Furthermore, the integration of HEMLAB with external solvers such as pyAMG and REFINE has enhanced the efficiency of numerical computations, enabling more precise turbulence modeling and transition capturing.

The results of this research emphasize the importance of accurately resolving complex flow phenomena, particularly in the context of high-lift aerodynamics where interactions between multi-element airfoils, boundary layer transitions, and wake dynamics play a crucial role. The findings confirm that metric-based mesh adaptation techniques lead to significant improvements in the prediction of aerodynamic coefficients, reducing errors associated with under-resolved flow features. The study also highlights the need for careful selection of turbulence models and solver configurations to optimize computational efficiency without compromising accuracy.

A crucial component of this research is the integration of the Scalable Nonlinear Equations Solvers (SNES) framework within the HEMLAB algorithm. SNES provides robust and efficient solution techniques for nonlinear systems, enabling the accurate resolution of governing equations in CFD applications. The implementation covers advanced iterative solvers, including Newton-Krylov methods, to enhance the stability and convergence of numerical computations. By utilizing SNES, this research has improved the handling of complex aerodynamic phenomena, particularly in cases involving strong nonlinearities and flow separation.

Preconditioners play a significant role in accelerating convergence and improving solver performance. The selection of the appropriate preconditioner was based on the problem characteristics and computational efficiency requirements. By coupling SNES with optimized preconditioning strategies, the computational cost of high-lift simulations was significantly reduced, making large-scale simulations more feasible.

A significant part of this research also involves the implementation and analysis of the Jacobian-Free Newton-Krylov (JFNK) method, which offers an efficient approach for solving large nonlinear systems without explicitly constructing the Jacobian matrix. JFNK leverages finite-difference approximations to estimate matrix-vector products, reducing memory consumption and computational effort compared to direct Jacobian construction methods. This approach is particularly beneficial for high-fidelity aerodynamic simulations where the Jacobian matrix can be prohibitively large and costly to compute.

In addition to JFNK, this research investigates direct approaches for solving nonlinear equations, where the Jacobian is explicitly constructed and factorized Schwarz method with block incomplete LU factorization with k levels of fill (ILU(k)) within each sub-block. While direct methods provide highly accurate solutions, they are often computationally expensive, particularly for large-scale problems. The trade-offs between direct and iterative methods are analyzed in detail, highlighting scenarios where each approach is most effective. In cases with well-conditioned systems, direct methods provide robust and reliable results, whereas JFNK proves advantageous in handling highly nonlinear and large-scale problems due to its lower memory footprint.

Furthermore, the research explores advanced Jacobian construction techniques, including analytical differentiation, automatic differentiation, and finite-difference approximations. Each method's impact on solver efficiency and accuracy is systematically evaluated. Analytical Jacobian construction offers the highest accuracy but requires substantial derivation effort, whereas automatic differentiation provides a balance between accuracy and ease of implementation. Finite-difference approximations, while straightforward to implement, may introduce numerical errors if not carefully tuned. The comparative analysis of these techniques informs best practices for choosing an appropriate Jacobian representation depending on the problem characteristics. After all these improvements it is possible to achieve quadratic convergence with the recent version of the code.

The initial version of HEMLAB was limited to the standard Spalart-Allmaras (SA) turbulence model. This research expanded its turbulence modeling capabilities by implementing several advanced SA-based variations. The SA-neg model was integrated to enhance the solver's ability to predict separated flows and recirculation regions more accurately. Additionally, the SA-neg-ft2 model was introduced to improve the robustness of turbulence modeling for transitional flows by incorporating ft2 modifications. Furthermore, the SA-QCR-2000 model was implemented to account for quadratic constitutive relations, which provide better predictions of secondary flows and streamline curvature effects.

A major contribution of this research is the development and implementation of a second-order discretization scheme for these turbulence models. The second-order formulation improves solution accuracy, particularly in capturing turbulence anisotropy and near-wall effects. These enhancements allow for better resolution of high-lift aerodynamic features, reducing numerical diffusion and improving the overall predictive capability of the HEMLAB solver. The integration of these turbulence models, coupled with metric-based mesh adaptation, has led to more reliable aerodynamic simulations, particularly in predicting stall point, flow separation, and wake interactions.

An essential aspect of high-fidelity computational modeling is the role of machine precision in ensuring numerical accuracy and stability. The sensitivity of numerical methods to floating-point precision can significantly impact the robustness of iterative

solvers, particularly for ill-conditioned systems. In this study, the effects of machine precision were also examined in the context of matrix-vector operations, residual convergence, and solution stability. The results show that using lower precision arithmetic can cause small rounding errors that build up over many iterations, potentially causing nonlinear solvers to fail. Conversely, higher precision arithmetic improves accuracy but at an increased computational cost. This research shows that balancing computational cost and precision is key for reliable and efficient CFD simulations. Future improvements in adaptive precision computing may further optimize solver performance, making large-scale aerodynamic simulations more practical.

In future, additional studies can be conducted to explore the impact of different adaptation criteria on the accuracy of high-lift aerodynamic predictions. Investigating alternative metric formulations, refining transition modeling techniques, and applying the framework to full aircraft configurations will contribute to the ongoing evolution of computational aerodynamics. Also, adjoint-based mesh adaptation techniques will be investigated and developed to enhance the sensitivity-driven refinement of the computational grid. These techniques will allow targeted resolution improvements in regions that most significantly influence key aerodynamic outputs, such as lift and drag. Furthermore, the integration of additional turbulence models, such as the k–ω SST model, will be considered to improve the robustness and accuracy of simulations, particularly for flows involving strong adverse pressure gradients and separation.

REFERENCES

[1] *High Lift Figure*, https://www.airplane-pictures.net/photo/904157/a6-epj-emirates-airlines-boeing-777-31h-er/.

[2] **Koklu, M.**, **Lin, J.**, **Hannon, J.**, **Melton, L.**, **Andino, M.**, **Paschal, K. and Vatsa, V.** (2021). Investigation of the nacelle/pylon vortex system on the high-lift common research model, *AIAA Journal*, *59*(9), 3748–3763.

[3] **Brezillion, J.**, **Dwight, R.P. and Wild, J.** (2008). Numerical Aerodynamic Optimisation of 3D High-lift Configurations, *26th International Congress of the Aeronautical Sciences*.

[4] **van DAM, C.P.** (2002). The aerodynamic design of multi-element high-lift systems for Transport Airplanes, *Progress in Aerospace Science*, *38*, 101–144.

[5] **Yokokawa, Y.**, **Murayama, M.**, **Ito, T. and Yamamoto, K.** (2006). Experiment and CFD of a High-lift Configuration Civil Transport Aircraft Model, *AIAA Paper 2006-3452*.

[6] **Rumsey, C.L.**, **Slotnick, J.P.**, **Long, M.**, **Stuever, R.A. and Wayman, T.R.** (2011). Summary of the First AIAA CFD High-Lift Prediction Workshop, *AIAA Journal of Aircraft*, *48*(6), 2068–2079.

[7] **Rumsey, C.L. and Slotnick, J.P.** (2015). Overview and Summary of the Second AIAA High Lift Prediction Workshop, *AIAA Journal of Aircraft*, *52*(4), 1006–1025.

[8] **Rumsey, C.L.**, **Slotnick, J.P. and Sclafani, A.J.** (2019). Overview and Summary of the Third AIAA High Lift Prediction Workshop, *AIAA Journal of Aircraft*, *56*(2), 621–644.

[9] **Evans, A.N.**, **Lacy, D.S.**, **Smith, I. and Rivers, B.** (2020). Test summary of the NASA high-lift common research model half-span at QinetiQ 5-metre pressurized low-speed wind tunnel, *AIAA Aviation 2020 Forum*, pp.AIAA Paper 2020–2770.

[10] **Park, M.A.**, **Alauzet, F. and Michal, T.** (2023). HLPW-4/GMGW-3: Mesh Adaptation for RANS Technology Focus Group Workshop Summary, *Journal of Aircraft*.

[11] **Park, M.A.** (2025). HLPW-5: Mesh Adaptation for RANS Technology Focus Group Workshop Summary, *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics, Orlando, FL, https://arc.aiaa.org/doi/pdfplus/10.2514/6.2025-0047.

[12] **Gad-el Hak, M.** (2003). Comments on critical view on new results in micro-fluid mechanics, *International Journal of Heat and Mass Transfer*, *46*(20), 3941–3945.

[13] **Yinghao, S.**, **Pang, Y.**, **Shen, Z.**, **Tian, Y. and Ge, H.** (2018). Multiparameter Analysis of Gas Transport Phenomena in Shale Gas Reservoirs: Apparent Permeability Characterization, *Scientific Reports*, *8*(1), 2601, `https://www.nature.com/articles/s41598-018-20949-2`.

[14] **Loseille, A.**, **Munguía, B. and Alonso, J.J.** (2019). Anisotropic Mesh Adaptation with the (Py)AMG Library, *3rd Annual SU2 Developers Meeting*.

[15] **Akkurt, S.** (2018). *A parallel unstructured finite volume method with an efficient edge based data structure for compressible flows*, Istanbul Technical University.

[16] **Meredith, P.** (1993). Viscous phenomena affecting high-lift systems and suggestions for future CFD development, *High-Lift System Aerodynamics, AGARD CP 515*, pp.19–1–19–8.

[17] **Smith, A.M.O.** (1975). High-lift aerodynamics, *Journal of Aircraft*, *12*(6), 501–530.

[18] **Woodward, D.S.**, **Hardy, B.C. and Ashill, P.R.** (1988). Some types of scale effect in low-speed high-lift flows, *ICAS Paper 4.9.3*.

[19] **Park, M.A.**, **Lee-Rausch, E.M. and Rumsey, C.L.** (2011). FUN3D and CFL3D Computations for the First High Lift Prediction Workshop, *AIAA Paper 2011-936*.

[20] **Murayama, M.**, **Lei, Z.**, **Mukai, J. and Yamamoto, K.** (2006). CFD Validation for High-Lift Devices: Three-Element Airfoil, *Trans. Jpn. Soc. Aeronant. Space Sci.*, *49*, 40–48.

[21] **Hansen, H.**, **Thiede, P.**, **Moens, F.**, **Rudnik, R. and Quest, J.** (2004). Overview about the European High Lift Research Programme EUROLIFT, *AIAA Paper 2004-0767*.

[22] **Rudnik, R.**, **Eliasson, P. and Perraud, J.** (2005). Evaluation of CFD methods for transport aircraft high lift systems, *The Aeronautical Journal*, *109*(1092), 53–64.

[23] **Rudnik, R.** (2003). CFD Assessment for 3D High Lift Flows in The European Project EUROLIFT, *AIAA 2003-3794*.

[24] **Eliasson, P.**, **Catalano, P.**, **Le Pape, M.**, **Ortmann, J.**, **Pelizzari, E. and Ponsin, J.** (2007). Improved CFD Predictions for High Lift Flows in the European Project EUROLIFT II, *25th AIAA Applied Aerodynamics Conference*, Miami, FL.

[25] **Krumbein, A.** (2004). Automatic transition prediction and application to high lift multi-element configurations, *AIAA Paper 2004-2543*.

[26] **Lei, Z.**, **Murayama, M.**, **Takenaka, K. and Yamamoto, K.** (2004). CFD Validation for High Lift Devices: Two-Element Airfoil, *Proceedings of 2004 KSAS-JSASS Joint Symposium on Aerospace Engineering*.

[27] **Murayama, M.**, **Yamamoto, K. and Kobayashi, K.** (2005). Validation of Flows on High-Lift Configurations by Structured- and Unstructured- Mesh Method, *AIAA Paper 2005-1226*.

[28] **Balan, A.**, **Park, M.A.**, **Wood, S.L.**, **Anderson, W.K. and Jacobson, K.** (2022). Angle-of-Attack Sweep with Mesh Adaptation for High-Lift Configurations, *AIAA SCITECH 2022 Forum*, p.0218.

[29] **Palacios, F.**, **Colonno, M.**, **Aranake, A.**, **Campos, A.**, **Copeland, S.**, **Economon, T.**, **Lonkar, A.**, **Lukaczyk, T.**, **Taylor, T. and Alonso, J.** (2013). Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design, *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, TX.

[30] **Glasby, R.S.**, **Erwin, T.**, **Stefanski, D.L.**, **Allmaras, S.R.**, **Galbraith, M.C.**, **Anderson, W.K. and Nichols, R.H.** (2016). Introduction to COFFE: The Next-Generation HPCMP CREATE-AV CFD Solver, *AIAA Journal*, *54*(3), March.

[31] **Schwamborn, D.**, **Gerhold, T. and Heinrich, R.** (2006). The DLR TAU-Code: Recent Applications in Research and Industry, Technical Report, FOI.

[32] **Eliasson, P.** (2001). EDGE: A Navier-Stokes Solver for Unstructured Grids, Technical Report, FOI, fOI Scientific Report, FOI-R-0298-SE.

[33] **Michal, T. and Krakos, J.** (2012). Anisotropic Mesh Adaptation Through Edge Primitive Operations, *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*.

[34] **Ibanez, D.**, **Barral, N.**, **Krakos, J.**, **Loseille, A.**, **Michal, T. and Park, M.** (2017). First benchmark of the Unstructured Grid Adaptation Working Group, *Procedia Engineering*, volume203, pp.154–166.

[35] **Michal, T.**, **Krakos, J.**, **Kamenetskiy, D.**, **Galbraith, M.**, **Ursachi, C.I.**, **Park, M.A.**, **Anderson, W.K.**, **Alauzet, F. and Loseille, A.** (2020). Comparing Unstructured Adaptive Mesh Solutions for the High Lift Common Research Model Airfoil, *AIAA AVIATION 2020 FORUM*, pp.AIAA 2020–3219.

[36] **Loseille, A.** (2014). Metric-orthogonal Anisotropic Mesh Generation, *Procedia Engineering*, *82*, 403–415.

[37] **Akkurt, S. and Sahin, M.** (2017). An efficient data structure for three-dimensional vertex based finite volume method, *APS 70th Annual Meeting Division of Fluid Dynamics*, Denver, CO, USA.

[38] **Akkurt, S. and Sahin, M.** (2022). 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*, *94*, 13–31.

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

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

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

[42] **Barth, T.J.** (1991). A 3-D upwind Euler solver for unstructured meshes, *10th Computational Fluid Dynamics Conference*, Honolulu.

[43] **Hascoet, L. and Pascual, V.** (2004). The Tapenade automatic differentiation tool: principles, model, and specification, *ACM Transactions on Mathematical Software*, *39*(3), 1–43.

[44] **Hascoet, L. and Pascual, V.** (2004). TAPENADE 2.1 user's guide, Technical Report, INRIA, technical Report RT-0300.

[45] **Spalart, P.R. and Allmaras, S.R.** (1994). A one-equation turbulence model for aerodynamic flows, *La Recherche Aerospatiale*, *1*, 5–21.

[46] **Bonet, J. and Peraire, J.** (1991). An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems, *International Journal for Numerical Methods in Engineering*, *31*(1), 1–17.

[47] **Balay, S.**, **Abhyankar, S.**, **Adams, M.**, **Brown, J.**, **Brune, P.**, **Buschelman, K.**, **Dalcin, L.**, **Dener, A.**, **Eijkhout, V.**, **Gropp, W.**, **Karpeyev, D.**, **Kaushik, D.**, **Knepley, M.**, **May, D.**, **LMcInnes, L.**, **Mills, R. andMunson, T.**, **Rupp, K.**, **Sanan, P.**, **Smith, B.**, **Zampini, S.**, **Zhang, H. and Zhang, H.**

[48] **Nived, M.R.**, **Kalkoe, N.N. and Eswaran, V.** (2023). Convergence acceleration of turbulent flow simulations using an implicit adaptive time-stepping (ATS) algorithm, *61st AIAA Aerospace Sciences Meeting*, National Harbor, MD, pp.AIAA 2023–2147.

[49] **Karypis, G. and Kumar, V.** (1999). A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM Journal of Scientific Computing*, *20*(1), 359–392.

[50] **Wang, Z.J.** (2000). A fast nested multi-grid viscous flow solver for adaptive Cartesian/quad grids, *International Journal for Numerical Methods in Fluids*, *33*, 657–680.

[51] **Guibas, L. and Stolfi, J.** (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Transactions on Graphics*, *4*(2), 75–123.

[52] **Botsch, M.**, **Steinberg, S.**, **Bischoff, S. and Kobbelt, L.** (2001). OpenMesh - a generic and efficient polygon mesh data structure, *1st OpenSG Symposium*.

[53] **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.

[54] **Ollivier-Gooch, C. and Van Altena, M.** (2002). A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, *Journal of Computational Physics*, *181*(2), 729–752.

[55] **Blazek, J.** (2015). *Computational Fluid Dynamics: Principles and Applications*, Butterworth-Heinemann.

[56] **Harten, A.** (1983). High resolution schemes for hyperbolic conservation laws, *Journal of Computational Physics*, *49*(3), 357–393.

[57] **Johnson, F.**, **Kamenetskiy, D.**, **Melvin, R.**, **Venkatakrishnan, V.**, **Wigton, L.**, **Young, D.**, **Allmaras, S.**, **Bussoletti, J. and Hilmes, C.** (2011). Observations regarding algorithm required for robust CFD codes, *Mathematical Modelling and Numerical Analysis*, *6*(3), 2–27.

[58] **Kamenetskiy, D.S.**, **Bussoletti, J.E.**, **Hilmes, C.L.**, **Venkatakrishnan, V.**, **Wigton, L.B. and Johnson, F.T.** (2014). Numerical evidence of multiple solutions for the Reynolds-Averaged Navier-Stokes equations, *AIAA Journal*, *52*(8), 1686–1698.

[59] **Frazza, L.** (2018). *3D anisotropic mesh adaptation for Reynolds Averaged Navier-Stokes simulations*, Sorbonne Université.

[60] **Luo, H.**, **Baum, J.D. and Löhner, R.** (1995). An improved finite volume scheme for compressible flows on unstructured grids, *33rd Aerospace Science Meeting and Exhibit*, Reno, NV, pp.AIAA Paper 1995–348.

[61] **Carlson, J.R.** (2011). Inflow/outflow boundary conditions with application to FUN3D, Technical ReportNASA/TM-2011-217181, NASA.

[62] **Anderson, W.K.**, **Wood, S.L. and Allmaras, S.R.** (2019). An Initial Exploration of Improved Numerics within the Guidelines of the Negative Spalart-Allmaras Turbulence Model, Technical ReportNASA/TM-2019-220429, NASA.

[63] **Allmaras, S.**, **Johnson, F. and Spalart, P.** (2012). Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model, *Proceedings of the Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, Big Island, Hawaii, `https://www.iccfd.org/iccfd7/assets/pdf/papers/ICCFD7-1902_paper.pdf`.

[64] **Dacles-Mariani, J.**, **Zilliac, G.G.**, **Chow, J.S. and Bradshaw, P.** (1995). Numerical/experimental study of a wingtip vortex in the near field, *AIAA Journal*, *33*(9), 1561–1568.

[65] **Diskin, B.**, **Ahmad, N.**, **Anderson, K.**, **Derlaga, J.**, **Pandya, M.**, **Rumsey, C.**, **Wang, L.**, **Wood, S.**, **Liu, Y.**, **Nishikawa, H. and Galbraith, M.** (2021). Verification Test Suite for Spalart-Allmaras QCR2000 Turbulence Model, *AIAA Scitech 2021 Forum*, `https://arc.aiaa.org/doi/10.2514/6.2021-1552`.

[66] **Brune, P.R.**, **Knepley, M.G.**, **Smith, B.F. and Tu, X.** (2015). Composing scalable nonlinear algebraic solvers, *SIAM Review*, *57*(4), 535–564.

[67] **George, A. and Liu, J.** (1981). *Computer solution of large sparse positive definite systems*, Prentice Hall.

[68] **Gropp, W.D.**, **Kaushik, D.K.**, **Keyes, D.E. and Smith, B.F.** (2001). High performance parallel implicit CFD, *Parallel Computing*, *27*(4), 337–362.

[69] **Thompson, K.B. and O'Connell, B.** (2019). Streamlined convergence acceleration for CFD codes, *AIAA Aviation 2019 Forum*, Dallas, TX, pp.AIAA Paper 2019–3709.

[70] **Pernice, M. and Walker, H.F.** (1998). NITSOL: a Newton iterative solver for nonlinear systems, *SIAM Journal on Scientific Computing*, *19*(1), 302–318.

[71] **Falgout, R.**, **Baker, A.**, **Chow, R.**, **Henson, V.E.**, **Hill, E.**, **Jones, J.**, **Kolev, T.**, **Lee, B.**, **Painter, J.**, **Tong, C.**, **Vassilevski, P. and Yang, U.M.** (1999). HYPRE high performance preconditioners (AUCRL-MA-137155 DR), Technical Report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Albuquerque, NM.

[72] **Saad, Y.** (1993). A flexible inner-outer preconditioned GMRES algorithm, *SIAM Journal on Scientific Computing*, *14*(2), 461–469.

[73] **Eisenstat, S.C.**, **Elman, H.C. and Schultz, M.H.** (1983). Variational iterative methods for nonsymmetric systems of linear equations, *SIAM Journal on Numerical Analysis*, *20*(2), 345–357.

[74] **van der Vorst, H.** (1992). Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM Journal on Scientific Computing*, *13*(2), 631–644.

[75] **Mavriplis, D.J.** (2002). An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers, *Journal of Computational Physics*, *175*(1), 302–325.

[76] **Wang, L.**, **Diskin, B.**, **Nielsen, E.J. and Liu, Y.** (2021). Improvements in iterative convergence of FUN3D solutions, *AIAA Scitech 2021 Forum*, Virtual Event, pp.AIAA Paper 2021–0857.

[77] **Ceze, M.A.** (2013). *A Robust hp-adaptation method for discontinuous Galerkin discretizations applied to aerodynamic flows*, University of Michigan.

[78] **Ahrabi, B.R. and Mavriplis, D.J.** (2020). An implicit ILU smoother for preconditioning of Newton-Krylov solvers with application in high-order stabilized finite-element methods, *Computers & Methods in Applied Mechanics and Engineering*, *358*, 112637.

[79] **Zhou, C. and Wang, Z.J.** (2015). CPR high-order discretization of the RANS equations with the SA model, *53rd AIAA Aerospace Sciences Meeting*, Kissimmee, FL, pp.AIAA Paper 2015–1286.

[80] **Hernandez, V.**, **Roman, J.E. and Vidal, V.** (2005). SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems, *ACM Transactions on Mathematical Software*, *31*(3), 351–362.

[81] **Alauzet, F. and Louseille, A.** (2016). A decade of progress on anisotropic mesh adaptation for computational fluid dynamics, *Computer-Aided Design*, *72*, 13–19.

[82] **Aksoy, E. and Sahin, M.** (2023). HEMLAB ALE: A new ALE algorithm for compressible flow problems, *ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems*, Lisbon, Portugal.

[83] **Park, M.A.**, **Balan, A.**, **Anderson, W.K.**, **Galbraith, M.C.**, **Caplan, P.C.**, **Carson, H.A.**, **Michal, T.**, **Krakos, J.A.**, **Kamenetskiy, D.S.**, **Loseille, A.**, **Alauzet, F.**, **Frazza, L. and Barral, N.** (2019). Verification of unstructured grid adaptation components, *AIAA Aerospace Science Meeting*, San Diego, CA, USA.

[84] **Wyman, N.J.**, **Park, M.A.**, **Baker, P. and Chawner, J.R.** (2021). A Framework for mesh-geometry associativity during mesh adaptation, *AIAA SciTech 2021 Forum*, AIAA 2021-1900, Virtual Event.

[85] **Documentation, N.R.** (2024). User's Guide and Technical Manual NASA/TM–20240006306, Technical Report.

[86] **Giles, M.B. and Pierce, N.A.** (2000). Adjoint Equations in CFD, *Annual Review of Fluid Mechanics*.

[87] **Sukas, H. and Sahin, M.** (2021). HEMLAB algorithm applied to the high-lift JAXA standard model, *AIAA SciTech Forum*.

[88] **Rumsey, C.L.**, **Jeffrey, S.S. and Anthony, J.S.** (2018). Overview and summary of the third AIAA high lift prediction workshop, *AIAA SciTech Forum*, Kissimmee, FL.

[89] **Yokokawa, Y.**, **Murayama, M.**, **Kanazaki, M.**, **Murota, K.**, **Ito, T. and Yamamoto, K.** (2008). Investigation and improvement of high-lift aerodynamic performances in low-speed wind tunnel testing, *46th AIAA Aerospace Sciences Meeting and Exhibit*, pp.AIAA–2008–0350.

[90] **Johnson, F.T.**, **Kamenetskiy, D.S.**, **Melvin, R.G.**, **Venkatakrishnan, V.**, **Wigton, L.B.**, **Young, D.P.**, **Allmaras, S.R.**, **Bussoletti, J.E. and Hilmes, C.L.** (2011). Observations regarding algorithms required for robust CFD codes, *Mathematical Modelling and Numerical Analysis*, *6*(3), 2–27.

[91] **Balin, J. and Jansen, K.E.** (2018). A comparison of RANS, URANS and DDES for high-lift systems from HiLiftPW-3, *AIAA Paper 2018-1254*.

[92] **Becker, G.G. and Granzoto, R.M.** (2018). DPW-6 and HiLiftPW-3 using the Stanford university unstructured (SU2), *AIAA Paper 18-2845*.

[93] **Jensen, J.C.**, **Stich, G.D.**, **Housmann, J.A.**, **Denison, M. and Kiris, C.C.** (2019). LAVA simulations for the 3rd AIAA CFD High Lift Prediction Workshop with body-fitted grids, *AIAA Aerospace Science Meeting*.

[94] **Rumsey, L.R.** (2007). Apparent transition behavior of widely-used turbulence models, *International Journal of Heat and Fluid Flow*, *28*, 1460–1471.

[95] **Galbraith, M.C.**, **Ursachi, C.I.**, **Chandel, D.**, **Allmaras, S.R.**, **Darmofal, D.L.**, **Hereth, E.A.**, **Mukhopadhaya, J. and Alonso, J.J.** (2022). Comparing multi-element airfoil flow solutions using multiple solvers with output-based adapted meshes, *AIAA Journal*, *60*(4), 1039–1054.

[96] **van der Vorst, H.A.** (2003). *Iterative Krylov methods for large linear systems*, Cambridge University Press.

[97] **Nejat, A. and Ollivier-Gooch, C.** (2008). Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations, *Journal of Computational Physics*, *227*, 2366–2386.

[98] **Harten, A. and Hyman, J.M.** (1983). Self adjusting grid methods for one-dimensional hyperbolic conservation laws, *Journal of Computational Physics*, *50*, 253–269.

[99] **Yildirim, A.**, **Kenway, G.K.W.**, **Mader, C.A. and Martins, J.R.R.A.** (2019). A Jacobian-free approximate Newton-Krylov startup strategy for RANS simulations, *Journal of Computational Physics*, *397*, 108741.

[100] **Sukas, H. and Sahin, M.** (2022). HEMLAB Algorithm Applied to 4th AIAA CFD High Lift Prediction Workshop, *AIAA Aviation 2022 Forum*, Chicago, IL & Virtual, pp.AIAA Paper 2022–3809.

[101] **Alauzet, F. and Frazza, L.** (2021). Feature-based and goal-oriented anisotropic mesh adaptation for RANS applications in aeronautics and aerospace, *Journal of Computational Physics*, *110340*, 1–77.

[102] **Pomeroy, B.** (2023). Kestrel results for DPW-VII including adaptive Mesh and IDDES computations, *AIAA Aviation 2023 Forum*, San Diego, CA.

[103] **Ahmad, N.**, **Wang, L.**, **Anderson, W.K.**, **Balakumar, P.**, **Iyer, P.S. and Nielsen, E.J.** (2022). FUN3D simulations for the 4th AIAA high-lift prediction workshop, *AIAA Aviation 2022 Forum (Virtual)*, Chicago, IL, pp.AIAA Paper 2022–3436.

[104] **Sukas, H. and Sahin, M.** (2024). HEMLAB Algorithm Applied to 5th AIAA CFD High Lift Prediction Workshop, *AIAA Aviation Forum and Ascend 2024*, Las Vegas, NV, pp.AIAA Paper 2024–3994.

[105] **Mavriplis, D.J.**, **Vassberg, J.C.**, **Tinoco, E.N.**, **Mani, M.**, **Brodersen, O.P.**, **Eisfeld, B.**, **Wahls, R.A.**, **Morisson, J.H.**, **Zickuhr, T.**, **Levy, D. and Murayama, M.** (2008). Grid quality and resolution issues from the drag prediction workshop series, *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada.

[106] **Carlson, J.**, **Gnoffo, P.**, **Jones, W.**, **Kleb, B.**, **Thomas, J.**, **Rumsey, C.**, **Biedron, R.**, **Derlaga, J.**, **Nielsen, E.**, **Hammond, D.**, **Lee-Rausch, E.**, **Thompson, K.**, **Walden, A.**, **Wang, L. and Wood, W.** (2020). FUN3D Manual 13.7, Technical Report, NASA Langley Research Center, `https://fun3d.larc.nasa.gov/papers/FUN3D_Manual-13.7_CORRECTED_COPY.pdf`, nASA/TM-20205010139.

[107] **Loseille, A. and Löhner, R.** (2013). Cavity-based operators for mesh adaptation, *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*.

[108] **Anderson, W.K.**, **Newman, J.C. and Karman, S.L.** (2018). Stabilized finite elements in FUN3D, *Journal of Aircraft*, *55*, 696–714.

[109] **Galbraith, M.C.**, **Allmaras, S.R. and Darmofal, D.L.** (2018). SANS RANS solutions for 3D benchmark configurations, *2018 AIAA Aerospace Sciences Meeting*.

[110] **Economon, T.D.**, **Mudigere, D.**, **Bansal, G.**, **Heinecke, A.**, **Palacios, F.**, **Park, J.**, **M., S.**, **Alonso, J.J. and Dubey, P.** (2016). Performance optimizations for scalable implicit RANS calculations with SU2, *Computers and Fluids*, *129*, 146–158.

[111] **Antoniadis, A.F.**, **Tsoutsanis, P. and Drikakis, D.** (2017). Assessment of high-order finite volume methods on unstructured meshes for RANS solutions of aeronautical configurations, *Computer and Fluids*, volume146, pp.86–104.

[112] **Thornber, B.**, **Mosedale, A.**, **Drikakis, D.**, **Youngs, D. and Williams, R.** (2008). An improved reconstruction method for compressible flows with low Mach number features, *Journal of Computational Physics*, *227*(10), 4873–94.

[113] **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 Forum*, San Diego, CA.

[114] **Dilek, E.**, **Erzincanli, B. and Sahin, M.** (2019). The numerical investigation of Lagrangian and Eulerian coherent structures for the near wake structure of a hovering Drosophile, *Theoretical and Computational Fluid Dynamics*, *33*, 255–279.

[115] **Ursachi, C.I.**, **Galbraith, M.C.**, **Allmaras, S.R. and Darmofal, D.L.** (2020). Output-based adaptive RANS solutions using higher-order FEM on a multi-element airfoil, *AIAA Aviation Forum*, Virtual event.

[116] **Park, M.A. and Anderson, W.K.** (2016). Spatial convergence of three dimensional turbulent flows, *54th AIAA Aerospace Sciences Meeting*.

[117] **Simmonds, N.**, **Tsoutsanis, P.**, **Antoniadis, A.**, **Jenkins, K. and Gaylard, A.** (2018). Low-Mach number treatment for finite-volume schemes on unstructured meshes, *Applied Mathematics and Computation*, *336*, 368–393.

[118] **Brown, P.N. and Saad, Y.** (1990). Hybrid Krylov methods for nonlinear systems of equations, *SIAM Journal on Scientific and Statistical Computing*, *11*, 450–481.

[119] **Rumsey, C.L. and Ying, S.X.** (2002). Prediction of High Lift: Review of Present CFD Capability, *Progress in Aerospace Sciences*, *38*, 145–180.

[120] **Slotnick, J.P.**, **An, M.Y.**, **Mysko, S.J.**, **Yeh, D.T.**, **Rogers, S.E.**, **Roth, K.**, **Baker, M.D. and Nash, S.M.** (2000). Navier-Stokes analysis of a high wing transport high-lift configuration with externally blown flaps, *AIAA Paper 2000-4219*.

[121] **Chaffin, S.M. and Pirzadeh, S.** (2005). Unstructured Navier-Stokes High-Lift Computations on a Trapezoidal Wing, *AIAA Paper 2005-5084*.

[122] **Loseille, A.** (2017). Unstructured Mesh Generation and Adaptation, *Handbook of Numerical Methods for Hyperbolic Problems - Applied and Modern Issues*, 263–302.

[123] **Brodersen, O. and Stuermer, A.** (2001). Drag prediction of engine-airframe interference effects using unstructured Navier-Stokes calculations, *19th AIAA Applied Aerodynamics Conference*, aIAA 2001-2414.

[124] **Cezze, M.** (2015). Steady turbulent transonic flow over an airfoil, *3rd International Workshop on High-Order Methods, 53rd AIAA Aerospace Science Meeting*.

[125] **Javaherchi, T.** (2010). Review of Spalart-Allmaras turbulence model and its modifications, Technical Report, University of Washington.

[126] **Moir, I.R.M.** (1994). Measurements on a Two-Dimensional Aerofoil with High Lift Devices, *AGARD AR 303*, pp.A2.1–A2.12.

[127] **Rumsey, C.L.**, **Nehart, D.H. and Kegerise, M.A.** (2016). The NASA juncture flow experiment: Goals, progress, and preliminary testing, *54th AIAA Aerospace Sciences Meeting*, San Diego, CA, USA.

[128] **Knoll, D.A. and Keyes, D.E.** (2004). Jacobian-free Newton-Krylov methods: a survey of approaches and applications, *Journal of Computational Physics*, *193*(2), 357–397.

[129] *ADIFOR - Automatic Differentiation of Fortran*, `https://www.mcs.anl.gov/research/projects/adifor/highlights.html`.

[130] **Chan, T.F. and Jackson, K.R.** (1984). Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms, *SIAM Journal on Scientific and Statistical Computing*, *5*, 533–542.

[131] **Kelley, C.T.** (1995). *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia.

[132] **Kamenetskiy, D.**, **Bussoletti, J.**, **Hilmes, C.**, **Johnson, F.**, **Venkatakrishnan, V. and Wigton, L.** (2013). Numerical Evidence of Multiple Solutions for the Reynolds-Averaged Navier-Stokes Equations for High-Lift Configurations, *51st AIAA Aerospace Sciences Meeting*.

[133] **Djeddi, R. and Ekici, K.** (2020). Memory Efficient Adjoint Sensitivity Analysis for Aerodynamic Shape Optimization, *AIAA Scitech 2020 Forum*.

[134] **Fleischmann, N.**, **Adami, S. and Adams, N.A.** (2020). A shock-stable modification of the HLLC Riemann solver with reduced numerical dissipation, *Journal of Computational Physics*, *423*, 109762.

[135] **Anderson, J.D.** (1995). *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill.

[136] **Venkatakrishnan, V.** (1996). Error Estimation and Adaptive Refinement for Aerodynamic Flows, *Journal of Computational Physics*.

[137] **Spalding, D.B.** (1961). A Single Formula for the Law of the Wall, *Journal of Applied Mechanics*.

CURRICULUM VITAE

Name Surname: Hülya SUKAS

EDUCATION:

- **Ph.D.:** 2019, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Aeronautics and Astronautics Engineering
- **M.Sc.:** 2017, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Aeronautics and Astronautics Engineering
- **B.Sc.:** 2014, Yildiz Technical University, Faculty of Mechanical Engineering, Mechanical Engineering
- **B.Sc.:** 2013, Yildiz Technical University, Faculty of Naval Architecture and Marine Engineering, Naval Architecture and Marine Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2017 The Best Thesis Award: Yildiz Technical University, Faculty of Naval Architecture and Marine Engineering

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Sukas H.** and Sahin M. (2025). A robust monolithic nonlinear Newton method for the compressible Reynolds averaged Navier-Stokes Equations, *Computers & Fluids*, 289(2):106549.
- **Sukas H.** and Sahin M. (2025). HEMLAB Algorithm Applied to the High-Lift Japan Aerospace Exploration Agency Standard Model, *Journal of Aircraft*, 62(1):1-12.
- **Sukas H.** and Sahin M. (2024). HEMLAB Algorithm Applied to 5th AIAA CFD High Lift Prediction Workshop, *AIAA Aviation 2024*, July 2024, Las Vegas, Nevada, USA.
- **Sukas H.** and Sahin M. (2022). HEMLAB Algorithm Applied to 4th AIAA CFD High Lift Prediction Workshop, *AIAA AVIATION 2022 FORUM*, June 2022, Chicago, IL & Virtual.
- **Sukas H.** and Sahin M. (2021). HEMLAB Algorithm Applied to the High-Lift JAXA Standard Model, *AIAA Scitech 2021 Forum*, January 2021, Virtual Event.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Sukas H.** and Sahin M. (2021). Prediction of the Nearfield Pressure Signature Using HEMLAB Algorithm, *11th ANKARA INTERNATIONAL AEROSPACE CONFERENCE*, September 2021, Ankara, TURKEY.

- **Sukas H.**, Pirlepeli, B., Ramesh, K., Konstantinos, K., Nikbay M. (2019). Aerodynamic Design Optimization of a Non-Planar C-Wing Configuration, *10th ANKARA INTERNATIONAL AEROSPACE CONFERENCE*, September 2019, Ankara, TURKEY.