ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

FULLY-COUPLED *k–ω* SST TURBULENCE MODEL IMPLEMENTATION FOR NONLINEAR NEWTON METHOD IN UNSTRUCTURED VERTEX-BASED HEMLAB ALGORITHM

M.Sc. THESIS

Atakan OKAN

Department of Aeronautical and Astronautical Engineering

Aeronautical and Astronautical Engineering Programme

JUNE 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

FULLY-COUPLED *k–ω* SST TURBULENCE MODEL IMPLEMENTATION FOR NONLINEAR NEWTON METHOD IN UNSTRUCTURED VERTEX-BASED HEMLAB ALGORITHM

M.Sc. THESIS

Atakan OKAN
(511221102)

Department of Aeronautical and Astronautical Engineering

Aeronautical and Astronautical Engineering Programme

Thesis Advisor: Prof. Dr. Mehmet ŞAHİN

JUNE 2025

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPISAL OLMAYAN DÜĞÜM TABANLI HEMLAB ALGORİTMASINDA DOĞRUSAL OLMAYAN NEWTON YÖNTEMİ İÇİN TAM BAĞLI *k–ω* SST TÜRBÜLANS MODELİ UYGULAMASI

YÜKSEK LİSANS TEZİ

Atakan OKAN
(511221102)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Mehmet ŞAHİN

HAZİRAN 2025

Atakan Okan, a M.Sc. student of ITU Graduate School student ID 511221102, successfully defended the thesis entitled "FULLY-COUPLED k–ω SST TURBULENCE MODEL IMPLEMENTATION FOR NONLINEAR NEWTON METHOD IN UNSTRUCTURED VERTEX-BASED 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** İstanbul Technical University	.....
Jury Members :	**Prof. Dr. Bayram ÇELİK** İstanbul Technical University	.....
	Prof. Dr. Metin MURADOĞLU Koç University	.....

Date of Submission : 29 May 2025
Date of Defense : 26 June 2025

To the scientists and engineers whose work advances our understanding of nature,

FOREWORD

I would like to express my gratitude to my thesis advisor Prof. Dr. Mehmet Şahin for his mentoring and selfless support during the code implementation and writing process of this study. I would also like to thank my coworkers in Computational Mechanics Department from Turkish Aerospace for their encouragement during my Masters degree studies. This acknowledgement extends to the National Center for High Performance Computing of Turkey (UHeM) and for the use of their computer resources. Last but not least; I would like to thank to my family, lovely girlfriend Ceren and bestfriend Buğra for their endless support throughout my life.

May 2025

Atakan OKAN
(Senior CFD Engineer)

TABLE OF CONTENTS

Page

FOREWORD ..... **ix**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xiii**
SYMBOLS ..... **xv**
LIST OF TABLES ..... **xvii**
LIST OF FIGURES ..... **xix**
SUMMARY ..... **xxiii**
ÖZET ..... **xxv**
1. INTRODUCTION **1**
1.1 Motivation and Aim 1
1.2 Literature Review 2
1.3 Thesis Outline 4
2. STANDARD *k-ω* SST TURBULENCE MODEL EQUATIONS **5**
2.1 Non-dimensionalization 8
2.2 Main Aspects of the *k-ω* SST Model 10
2.2.1 Blending functions 10
2.2.1.1 Numerical implications 10
2.2.2 Stress limiter 12
2.2.2.1 Numerical implications 14
2.2.3 Production limiter 14
2.2.3.1 Numerical implications 15
2.2.4 Other model variants 15
2.2.4.1 SST-V 15
2.2.4.2 SST-2003 16
2.2.4.3 SST-KL 16
2.2.4.4 SST-sust 17
2.2.4.5 SST-RC 17
2.2.4.6 SST-LM 18
2.2.4.7 *k-ω* TNT 18
2.2.5 Initial and boundary conditions 19
2.2.5.1 Analytical value of ω at the wall 20
2.2.5.2 Numerical implications 21
2.2.6 Positivity of *k* and ω 22
2.2.6.1 Variable transformations 24
3. *k-q* TRANSFORM FOR THE *k-ω* SST TURBULENCE MODEL **25**
3.1 Production and Stress Limiter Terms 26
3.2 Blending Functions and Intermediate Terms 26
3.3 Comments 27
4. NUMERICAL DISCRETIZATION **29**
4.1 Temporal Discretization 29
4.1.1 Deriving semi-discrete form 29

4.1.2 Backward-Euler implicit time integration 31
4.1.3 Newton method for steady-state solutions 33
4.1.4 Non-linear Newton solver 34
4.2 Spatial Discretization 35
4.2.1 Vertex-based finite volume method 35
4.2.2 Gradient evaluation 36
4.2.3 Flux reconstruction 38
4.2.4 Roe's flux difference splitting method 40
4.2.5 Numerical representation of k-q SST model 40
5. COUPLING OF SST TURBULENCE MODEL EQUATIONS 43
5.1 Segregated and Coupled Implementations 43
5.2 Flux Jacobian 44
5.3 Inviscid Jacobian 46
5.4 Viscous Jacobian 47
5.5 Contributions of Turbulence Model Source Terms 48
6. ADAPTIVE MESH REFINEMENT 51
6.1 Isotropic and Anisotropic Refinement 51
6.2 Implementation Challenges in AMR 52
6.3 Available AMR Tools 53
6.3.1 NASA-refine 53
6.3.2 INRIA-pyAMG 53
7. TEST CASES FOR MODEL VERIFICATION 55
7.1 Zero Pressure Gradient Flat Plate 55
7.1.1 Case conditions 55
7.1.2 Mesh 55
7.1.3 Numerics 57
7.1.4 Results 57
7.2 4th High Lift Prediction Workshop CRM-HL Multi-Element Airfoil 62
7.2.1 Case conditions 62
7.2.2 Mesh 63
7.2.3 Numerics 67
7.2.4 Results 68
7.3 Transonic ONERA OAT15A Supercritical Airfoil (DPW–8 Test Case 1) 73
7.3.1 Case conditions 73
7.3.2 Mesh 74
7.3.3 Numerics 76
7.3.4 Results 76
7.4 Subsonic NACA-0012 Airfoil 80
7.4.1 Case conditions 80
7.4.2 Mesh 81
7.4.3 Numerics 84
7.4.4 Results 85
8. CONCLUSIONS AND RECOMMENDATIONS 91
8.1 Conclusions 91
8.2 Future Work 92
REFERENCES 93
APPENDICES 97
APPENDIX A 99
APPENDIX B 99
CURRICULUM VITAE 105

ABBREVIATIONS

AMR : Adaptive mesh refinement
AUSM : Advection upstream splitting method
CFD : Computational fluid dynamics
CFL : Courant-Friedrichs-Lewy
CRM : Common Research Model
DPW : Drag Prediction Workshop
FDS : Flux difference scheme
FVM : Finite volume method
GPU : Graphics processor unit
HL : High-lift
HLLC : Harten-Lax-Van Leer-Contact
HLPW : High-lift Prediction Workshop
MPI : Message Parsing Interface
NACA : National Advisory Committee for Aeronautics
NASA : National Aeronautics and Space Administration
TMR : Turbulence Modeling Resource
SST : Shear stress transport

SYMBOLS

Re	: Reynolds number
Ma	: Mach number
Δt	: Physical timestep
$\Delta \tau$	: Pseudo-timestep
c	: Reference chord
L	: Reference length
d	: Wall-distance
μ	: Dynamic viscosity
v	: Kinematic viscosity
μ_t	: Turbulent viscosity
∞	: Freestream quantity
k	: Turbulent kinetic energy
ω	: Specific turbulent dissipation rate
α	: Angle of attack
J	: Flux Jacobian
C_D	: Drag coefficient
C_f	: Skin friction coefficient
C_L	: Lift coefficient
ρ	: Density
u,v,w	: Velocity vector components
E	: Total energy
Q	: State vector
R	: Residual vector
P	: Production term
D	: Destruction term
τ_{ij}	: Stress tensor
S_{ij}	: Strain rate tensor
W_{ij}	: Rotation rate tensor
Ω	: Vorticity magnitude

LIST OF TABLES

Page

Table 2.1 : Term classification for the standard *k–ω* SST-m turbulence model equations. 7
Table 2.2 : Term classification for the non-dimensional *k–ω* SST-m turbulence model equations. 9
Table 2.3 : Freestream insensitivity study of the *k–ω* SST model tested by NASA using CFL3D for the benchmark zero-pressure gradient flat plate. 20
Table 2.4 : Near-wall forms for the standard *k−ω* model (Kalitzin et al., 2005)..... 22
Table 4.1 : Finite-volume discretisation of the non-dimensional *k*-equation for HEMLAB *k–q* SST model using median-dual volume for vertex V. ... 41
Table 4.2 : Finite-volume discretisation of the non-dimensional *q*-equation for HEMLAB *k–q* SST model using median-dual volume for vertex V. ... 41
Table 7.1 : Physical conditions for the flat plate validation case 55
Table 7.2 : Mesh parameters for the flat plate validation case 56
Table 7.3 : Numerical parameters for the flat plate validation case 57
Table 7.4 : Comparison of drag coefficient C_d for different solvers and turbulence models for 545 × 385 grid. 61
Table 7.5 : Physical conditions for the HLPW4 multi-element airfoil case 62
Table 7.6 : Mesh parameters for the HLPW4 multi-element airfoil case 63
Table 7.7 : Numerical parameters for the high lift multi-element airfoil..... 67
Table 7.8 : HEMLAB refinement data at $M_\infty = 0.2, Re = 5 \times 10^6$ and $\alpha = 16^\circ$ with L_4 norm, $\beta = 1$ and $\varepsilon = 0.001$ 73
Table 7.9 : Physical conditions for DPW8 transonic ONERA OAT-15 airfoil case. 73
Table 7.10 : Mesh parameters for DPW8 transonic ONERA OAT-15 airfoil case. 74
Table 7.11 : Numerical parameters for the transonic ONERA OAT-15 airfoil case. 76
Table 7.12 : Grid-convergence of HEMLAB drag and lift coefficients at $Ma_\infty = 0.73$ and $\alpha = 1.5^\circ$ for the ONERA OAT15 transonic airfoil study..... 79
Table 7.13 : Physical conditions for the subsonic NACA-0012 airfoil 81
Table 7.14 : Mesh parameters for the subsonic NACA-0012 airfoil case 81
Table 7.15 : Numerical parameters for the subsonic NACA-0012 airfoil case 84
Table 7.16 : HEMLAB mesh independence study for NACA0012 airfoil using the SA-neg & *k-ω* SST models..... 89

LIST OF FIGURES

Page

Figure 2.1 : Behaviour of F_1 blending function in SST model **11**
Figure 2.2 : Rectangular grid illustrating wall-to-node distances **11**
Figure 2.3 : F_1 and F_2 vs wall distance (Adapted from FUN3D (NASA Langley Research Center (n.d.)) **12**
Figure 4.1 : Dual-volume constructed around the primal mesh vertex i **36**
Figure 7.1 : Structured finest level 545 × 385 flat plate mesh (NASA Langley Research Center (n.d.)). **56**
Figure 7.2 : *k*-contours along the boundary layer on the flat plate and profile at x=0.97 m for HEMLAB k-q SST-m model for 545 × 385 grid **58**
Figure 7.3 : *k*-contours along the boundary layer on the flat plate and profile at x=0.97 m for FUN3D SST-Vm model for 545 × 385 grid **59**
Figure 7.4 : ω-contours along the boundary layer on the flat plate and profile at x=0.97 m for HEMLAB for 545 × 385 grid **59**
Figure 7.5 : ω-contours along the boundary layer on the flat plate and profile at x=0.97 m for FUN3D for 545 × 385 grid **59**
Figure 7.6 : Comparison of *k*-profiles at x=0.97 m for HEMLAB and FUN3D for 545 × 385 grid …..... **60**
Figure 7.7 : Comparison of ω profiles at x=0.97 m for HEMLAB and FUN3D vs analytical solution at the wall for 545 × 385 grid **60**
Figure 7.8 : Skin friction variation along the flat plate for HEMLAB, FUN3D and theory for 545 × 385 grid **61**
Figure 7.9 : L_2-norm convergence plot for HEMLAB SST model using 545 × 385 flat plate grid. **61**
Figure 7.10 : Variation of CFL number vs iteration number for HEMLAB SST model using 545 × 385 flat plate grid **62**
Figure 7.11 : CRM-HL multi-element airfoil geometry denoting slat, main element and flap regions **63**
Figure 7.12 : Initial coarse triangular unstructured near-wall mesh including all three elements from 1st sub-iteration for 2D CRM-HL airfoil case **64**
Figure 7.13 : Initial coarse triangular unstructured domain mesh from 1st sub-iteration for 2D CRM-HL airfoil case **64**
Figure 7.14 : Adapted unstructured mesh from 9th sub-iteration for 2D CRM-HL airfoil case using HEMLAB *k*-ω SST model at AoA=16°..... **65**
Figure 7.15 : Initial coarse triangular unstructured slat region mesh from 1st sub-iteration for 2D CRM-HL airfoil case **65**
Figure 7.16 : Adapted slat-main element gap region mesh from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k*-ω SST model at AoA=16° **66**
Figure 7.17 : Initial coarse triangular unstructured flap region mesh from 1st sub-iteration for 2D CRM-HL airfoil case. **66**
Figure 7.18 : Adapted flap trailing edge region mesh from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k*-ω SST model at AoA=16° **....** **67**

Figure 7.19 : Pressure coefficient distribution plot of the adaptive solution from 9th sub-iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16◦ vs fixed-grid FUN3D SA-neg solution..... **68**
Figure 7.20 : Mach contours around all airfoil parts for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°. **69**
Figure 7.21 : Mach contours around airfoil slat region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16° **69**
Figure 7.22 : Mach contours around airfoil flap region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°. **70**
Figure 7.23 : k contours around airfoil slat region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°. **70**
Figure 7.24 : k contours around airfoil flap region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°. **71**
Figure 7.25 : *ω* contours around airfoil slat region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16° **71**
Figure 7.26 : ω contours around airfoil flap region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°. **72**
Figure 7.27 : L_2-norm convergence plot for HEMLAB SST model using consecutive anisotropic mesh adaptation for high-lift multi-element airfoil. **72**
Figure 7.28 : Variation of CFL number vs iteration number for HEMLAB SST model using consecutive anisotropic mesh adaptation for high-lift multi-element airfoil. **72**
Figure 7.29 : ONERA OAT-15 airfoil geometry **74**
Figure 7.30 : Level-2 structured near-wall mesh for ONERA OAT-15 airfoil case. **74**
Figure 7.31 : Zoomed views for Level-2 structured ONERA OAT15A mesh: (top) leading and (bottom) trailing edge **75**
Figure 7.32 : Mach contours around ONERA airfoil for the Level-2 grid solution using HEMLAB *k-ω* SST model at AoA=1.5° **77**
Figure 7.33 : Pressure coefficient distribution plots around the transonic ONERA OAT-15 airfoil using the Level-2 structured grid for HEMLAB *k-ω* SST model at AoA=2.5◦ vs experimental data and HEMLAB SA-neg model **78**
Figure 7.34 : *k* contours around ONERA airfoil for the Level-2 grid solution using HEMLAB *k-ω* SST model at AoA=1.5° **78**
Figure 7.35 : *ω* contours around ONERA airfoil for the Level-2 grid solution using HEMLAB *k-ω* SST model at AoA=1.5° **79**
Figure 7.36 : L2-norm convergence plot for HEMLAB SST model using the Level-2 structured grid for transonic ONERA OAT-15 airfoil..... **80**
Figure 7.37 : Variation of CFL number vs iteration number for HEMLAB SST model using the Level-2 structured grid for transonic ONERA OAT-15 airfoil..... **80**

Figure 7.38 : Mesh independence study for HEMLAB using 5 different structured o-grid type meshes at different angles of attack **82**
Figure 7.39 : Level-4 structured o-grid for NACA0012 airfoil near-wall region..... **83**
Figure 7.40 : Level-4 structured o-grid for NACA0012 airfoil leading-edge region **83**
Figure 7.41 : Level-4 structured o-grid for NACA0012 airfoil trailing-edge region **84**
Figure 7.42 : Drag polar for NACA0012 case for different CFD solvers and experimental data **85**
Figure 7.43 : Pressure coefficient plot for NACA0012 case using HEMLAB turbulence models and reference experimental data at AoA=10° (NASA Langley Research Center (n.d)) **86**
Figure 7.44 : Mach contour plot for NACA0012 case using HEMLAB *k*-ω SST model at AoA=10° **87**
Figure 7.45 : *k* contour plot for NACA0012 case using HEMLAB *k*-ω SST model at AoA=10° **87**
Figure 7.46 : ω contour plot for NACA0012 case using HEMLAB *k*-ω SST model at AoA=10° **88**
Figure 7.47 : L_2-norm convergence plot for HEMLAB SST model using NACA-0012 Level-4 structured grid **88**
Figure 7.48 : Variation of CFL number for HEMLAB SST model using NACA-0012 Level-4 structured grid. **89**

FULLY-COUPLED *k–ω* SST TURBULENCE MODEL IMPLEMENTATION FOR NONLINEAR NEWTON METHOD IN UNSTRUCTURED VERTEX-BASED HEMLAB ALGORITHM

SUMMARY

This thesis presents the theory, implementation and verification details of a fully-coupled *k-ω* Shear Stress Transport (SST) turbulence model in the density-based, unstructured and vertex-based CFD solver HEMLAB for robust and efficient solution of external-aerodynamics cases with a non-linear Newton method.

The standard *k-ω* SST equations are first non-dimensionalized using the compressible, conservative form of original equations with production limiter included for *k*-equation. Then, an "inverse-square-root" transformation is applied to the *ω* variable and the "*k-q* SST model" is obtained so that the new *ω* variable remains positive near solid walls.

Spatial discretisation is built on unstructured meshes and the vertex-based finite-volume method with median-dual cells around primal mesh vertices. For inviscid flux reconstruction, blending of upwinding with central approach is utilized. At cell interfaces, Roe flux difference splitting scheme is used. Left/right state vectors are obtained with unweighted least-squares approach. In addition, gradients of primitive variables in viscous fluxes are obtained at edge midpoints with Green-Gauss theorem. For convective terms of SST model, first order upwinding is used. Non-dimensional, fixed and Dirichlet-type BCs are applied at walls and farfield boundaries for *k* and *q* (transformed *ω*) variables.

Solver is based on implicit dual-time stepping with a pseudo-time continuation (PTC) strategy in steady-state. For inviscid flux Jacobians, automatic differentiation (AD)-based source code transformation is applied. Exact, hand-derived source-term contributions are added to the Jacobians so that Newton method can achieve quadratic convergence. PETSc non-linear SNES solver is applied with a line-search technique which controls each Newton update during the solution. pyAMG anisotropic adaptive mesh refinement (AMR) library is integrated to the solution algorithm for automatic capturing of high-gradient solution regions with Mach and distance-based refinement sensors. For the *k-q* SST model, the positive constant parameter "*A*" is included in the transformation for controlling the convergence and residual behaviour of *q* equation.

Verification is tested on four two-dimensional external-aerodynamics test cases. For the zero-pressure-gradient flat plate at $Re = 5\times10^6$ and $M_\infty = 0.20$, the skin-friction curve follows analytical solution, the drag coefficient differs from FUN3D by around half drag count and *k-ω* profiles match FUN3D in the boundary layer.

For the NACA-0012 airfoil at $M_\infty = 0.15$ and $Re = 6\times10^6$, mesh-independent lift and drag coefficients are obtained at 0°, 10° and 15° angle of attack. Surface pressure coefficient at 10° matches experimental results and the drag value is within five counts of HEMLAB's SA-neg model.

In the CRM-HL multi-element airfoil at $Re = 5\times10^6$, $M_\infty = 0.20$ and $\alpha = 16°$, an anisotropically adapted, unstructured triangular grid automatically and properly refines the slat-gap shear layer and flap wake based on Mach sensor. Pressure coefficients on all airfoil elements closely follow FUN3D results, and convergence is very good on all successive adaptation cycles.

Finally, the transonic ONERA OAT15A airfoil at $Re = 3\times10^6$, $M_\infty = 0.73$ and $\alpha = 1.5°$ demonstrates that the implementation still stays stable in the presence of a shock system. The surface pressure coefficient trend follows the experimental distribution.

For all these test cases, the fully-coupled *k-q* SST model implementation in HEMLAB preserved positivity of the q - and therefore ω - variable. In addition, it achieved pseudo-CFL numbers around 10000, converged rapidly while the L_2-norm of the residual fell to 10^{-10} and 10^{-12} levels. Results were in good accordance with other reference numerical and experimental solutions.

Therefore, this study shows that exact-Jacobian-based fully-coupled implementation of *k-ω* SST model in HEMLAB can provide both robustness and accuracy on unstructured grids and that it works well even with anisotropic refinement. Recommended next steps are to extend the implementation to three-dimensional problems and conduct further verification and validation studies, add second-order upwinding for the turbulence equations, and optionally: implement transition and rotation-curvature corrections, and other SST model variants.

YAPISAL OLMAYAN DÜĞÜM TABANLI HEMLAB ALGORİTMASINDA DOĞRUSAL OLMAYAN NEWTON YÖNTEMİ İÇİN TAM BAĞLI k–ω SST TÜRBÜLANS MODELİ UYGULAMASI

ÖZET

Hesaplamalı akışkanlar dinamiğinde (HAD) karmaşık dış akışlara ait aerodinamik problemleri yüksek doğrulukla çözebilmek, türbülans modellerinin hem fiziksel tutarlılığına hem de ilgili çözücünün nümerik sağlamlığına bağlıdır. Özellikle çözücünün denklemleri birbirine tam bağlı bir biçimde kullanması, analizin yakınsama sürecinde çözüm kararlılığını oldukça artırmaktadır. Ancak geleneksel uygulamalarda, türbülans denklemlerinin momentum denklemlerine gevşek bağlı uygulanması, bu potansiyeli tam anlamıyla kullanmayı engeller.

Bu tez, söz konusu boşluğu doldurarak, *k–ω* SST modelini düğüm tabanlı bir çözücüde doğrusal olmayan Newton yönteminde tam bağlı şekilde kullanmanın beraberinde getireceği avantajları ortaya koymayı amaçlamaktadır. Çalışma aynı zamanda, yapısal olmayan adaptif ağ iyileştirme yaklaşımını kullanan bir algoritmaya da entegre olduğundan, gelecekteki daha kompleks aerodinamik koşullar ve geometriler içeren uygulamalara da güçlü bir zemin hazırlamaktadır.

CFD Vision 2030 Raporu, hesaplamalı akışkanlar dinamiği (HAD) alanında önemli bir dönüm noktasıdır. Rapor, yüksek Reynolds sayılı endüstriyel akışların kararlı ve verimli RANS çözücülerine ihtiyaç duyduğunu, ancak karmaşık geometriler etrafında ağ üretiminin hâlen yüksek hassasiyetli simülasyonların ortalama bir mühendisin rutin bir şekilde uygulayabilecek kadar yaygınlaşmasını engelleyen başlıca "darboğaz" olduğunu açıkça belirtir. Bunu aşmak için adaptif ağ iyileştirme teknikleri (AMR) etkili bir seçenek olsa da, yöntem ancak karmaşık geometrilerde kararlı çalışan çözücülerle birlikte anlam kazanmaktadır.

Açık zaman adımlı çözücüler programlamasının görece kolay olmasına karşın, zor problemlerde yavaş yakınsama gösterir ve zaman adımı kararlılık açısından daha sert kısıtlamalara tabidir; dolayısıyla pratik mühendislik uygulamalarında genellikle örtük zaman adımlı yöntemler gereklidir.

Örtük zaman adımıyla çalışan HAD çözücüleri verimlilik gerekçesiyle çoğunlukla tek-denklemli Spalart–Allmaras modelini içerir; iki denklemli *k–ω* SST modeli de sıkça kullanılmakla birlikte, çoğu zaman Newton yöntemiyle tam kuadratik yakınsamayı engelleyen, gevşek-bağlı bir biçimde programlanır.

Buna karşılık, denklemlerin tam olarak birbirine bağlanması ve Jacobian matrislerinin analitik olarak oluşturulması, çözücünün artık değerlerini makine hassasiyetine kadar düşürebilmesine ve çok daha sağlam bir yakınsama davranışı göstermesine olanak tanır.

Bu kapsamda, bu tez çalışması dış akış aerodinamiği problemlerinin doğrusal olmayan Newton yöntemiyle kararlı ve verimli şekilde çözülmesi amacıyla, yoğunluk-tabanlı, yapısız ve düğüm-tabanlı HAD çözücüsü HEMLAB'a, tam bağlı *k-ω* Kayma

Gerilmesi Taşınımı (SST) türbülans modelinin uygulamasını, modelin kuramsal temellerini ve doğrulama çalışmalarına dair ayrıntıları sunmaktadır.

Standart k-ω SST denklemleri, önce sıkıştırılabilir, korunumlu biçim temel alınarak, k denklemi için üretim sınırlayıcısını da içerecek şekilde boyutsuzlaştırılmıştır. Ardından ω değişkenine "ters karekök" dönüşümü uygulanmış ve yeni ω değişkeninin katı duvarlara yakın bölgede pozitif kalmasını sağlayan k-q SST modeli elde edilmiştir. Yakınsama hızı ile q denkleminin artık davranışını ayarlamak amacıyla model dönüşümünde pozitif bir A sabiti kullanılmıştır.

Uzaysal ayrıklaştırma, asal ağ düğümleri çevresinde medyan-ikili hücreler kullanan düğüm-tabanlı sonlu hacim yöntemiyle yapısız ağlar üzerinde gerçekleştirilmiştir. Viskoz olmayan yüzey akılarının rekonstrüksiyonu için yukarıya dönük ve merkezli şemaların harmanlanması kullanılmış; hücre arayüzlerindeki sayısal akı hesaplarında Roe şeması uygulanmıştır.

Akı hesaplarında ihtiyaç duyulan sol-sağ çözüm vektörleri ağırlıksız en küçük kareler yaklaşımıyla elde edilmiş, viskoz akılardaki temel değişken gradyanları ise kenar orta noktalarında Green–Gauss teoremi ile hesaplanmıştır.

SST modelinin taşınım terimleri için birinci mertebe yukarıya dönük şema tercih edilmiştir. k ve q (dönüştürülmüş ω) değişkenleri için duvarlarda ve uzak bölgede boyutsuz, sabit Dirichlet sınır koşulları uygulanmıştır.

Çözücü, zamandan bağımsız çözümlerde sözde-zaman sürekliliği (PTC) stratejisiyle birleştirilmiş, örtük çift-zaman adımlaması yöntemine dayanmaktadır. Viskoz olmayan akı Jacobian matrisleri otomatik türevleme (AD) tabanlı kaynak-kod dönüşümüyle oluşturulmuş, tam türetilmiş kaynak-terim katkıları bu Jacobian matrislerine eklenerek Newton yönteminin kuadratik yakınsama özelliği korunmuştur.

Her Newton adımını denetleyen "çizgi-arama tekniği", PETSc doğrusal olmayan SNES çözücüsü aracılığıyla uygulanmıştır. Mach ve mesafe tabanlı sensörler kullanan anizotropik adaptif ağ iyileştirme (AMR) kütüphanesi pyAMG, yüksek gradyanlı bölgelerin otomatik olarak iyileştirilmesi için algoritmaya entegre edilmiştir.

Model doğrulaması dört iki-boyutlu dış aerodinamik test vakası üzerinde yapılmıştır. Sıfır basınç gradyenli düz plaka ($Re = 5 \times 10^6$, $M_\infty = 0.20$) için yüzey sürtünme eğrisi analitik çözümü izlemiş, sürükleme katsayısı FUN3D değerinden sadece yarım sürükleme katsayı birimi kadar sapmış ve sınır tabakasındaki k-ω profilleri FUN3D sonuçlarıyla uyuşmuştur.

NACA 0012 profilinde ($M_\infty = 0.15$, $Re = 6 \times 10^6$) 0°, 10° ve 15° hücum açıları için ağdan bağımsız olacak şekilde taşıma ve sürükleme katsayıları elde edilmiştir. Yüzey basınç katsayısı 10° hücum açısında deneyle örtüşmüş, sürükleme değeri HEMLAB'ın SA-neg modeliyle karşılaştırıldığında 5 sürükleme katsayısı biriminden küçük kalmıştır.

CRM-HL çok elemanlı kanat profilinde ($Re = 5 \times 10^6$, $M_\infty = 0.20$, $\alpha = 16°$) anizotropik olarak adapte edilmiş yapısız üçgen ağ, Mach sensörüne bağlı olarak slat aralığı kayma tabakasını ve flap arkasında yer alan iz bölgesine ait ağ yapılarını doğru biçimde, otomatik olarak iyileştirmiştir. Tüm elemanlar üzerindeki basınç katsayıları FUN3D sonuçlarını yakından takip etmiş, ardışık adaptif ağ iyileştirmesi döngülerinin tümünde iyi bir yakınsama sağlanmıştır.

Transonik ONERA OAT15A profili ($Re = 3 \times 10^6$, $M_\infty = 0.73$, $\alpha = 1.5°$) mevcut şok sistemine rağmen çözümün kararlı kaldığını göstermiş, yüzey basınç katsayısı deneysel dağılımla aynı eğilimi sergilemiştir.

Tüm vakalarda, HEMLAB'daki tam bağlı *k-q* SST modeli q ve dolayısıyla ω değişkeninin pozitifliğini korumuş; CFL sayıları yaklaşık 10 000'e ulaşmış, L_2 "artık normu" 10^{-10}–10^{-12} seviyelerine düşmüş ve hızlı yakınsama elde edilmiştir. Sonuçlar, referans sayısal ve deneysel verilerle iyi uyum göstermektedir.

Çalışma, tam Jacobian kullanan, tam bağlı *k-ω* SST implementasyonunun yapısız ağlarda hem kararlı hem de yüksek doğruluklu sonuçlar verdiğini ve anizotropik ağ iyileştirmesi ile de uyumlu çalıştığını kanıtlamaktadır.

Gelecekteki çalışmalar üç boyutlu problemler için de implementasyonun genişletilip doğrulama çalışmalarının yapılmasını ve türbülans denklemleri için ikinci mertebe konvektif şema implementasyonunu içermelidir. Ayrıca isteğe bağlı olarak: laminer-türbülanslı akış geçiş modellemesi kabiliyetleri, dönme-eğrilik düzeltmeleri ve diğer SST varyantları da HEMLAB çözücüsüne eklenebilir.

1. INTRODUCTION

1.1 Motivation and Aim

The “CFD Vision 2030 Study” which is one of the most critical milestones in the field to this date states that “industrial high-Reynolds-number flows demand robust, efficient RANS solvers—where mesh generation is still considered a bottleneck that prevents high-fidelity simulations of around complex geometries from becoming a routine practice for an average CFD engineer” (Slotnick et. al., 2014).

Adaptive mesh refinement can help solve this issue but only when stable solvers are available for complex geometries (Park et. al, 2016). Although explicit solvers are easier to program, they suffer from slow convergence for stiff problems and have more strict time-step and stability limits, requiring implicit methods to be used (Blazek, 2001).

Most industrial implicit solvers implement the one-equation Spalart–Allmaras RANS model for efficiency, two-equation k-ω SST is also widely adopted (ANSYS Inc, 2020) —although typically in a loosely coupled form that prevents quadratic convergence when a Newton method is used. If full coupling is enabled and exact Jacobians are constructed, solvers can reach machine-precision residuals with more robust convergence characteristics (Sukas & Şahin, 2025; Thompson & O’Connell, 2019).

Accordingly, the present thesis aims to implement Menter’s two-equation k-ω SST turbulence model for the RANS equations in a fully coupled (monolithic) and implicit manner within the density-based, unstructured, vertex-centered HEMLAB code, while targeting robust and efficient external-aerodynamics CFD simulations where machine-precision residual tolerance is desired. The main objectives are:

1. Implement the k-ω SST model in HEMLAB in a monolithic fashion.
2. Use exact (hand-derived) source term contributions to the Jacobians instead of approximate factorization methods commonly found in the literature.

3. Apply a consistent non-linear Newton method with line-search using PETSc and push the CFL number to very large values for rapid convergence with pseudo-transient continuation (PTC).
4. Provide solver stability for the ω-equation using appropriate variable transformations for handling challenging near-wall behaviour.
5. Verify implementation accuracy on the 2-D, subsonic, zero-pressure-gradient flat-plate benchmark case, and test convergence characteristics on more challenging aerodynamic problems; a subsonic airfoil at 0°, 10° and 15° angle of attack, another subsonic and high-lift, multi-element airfoil configuration at 16° angle of attack and a transonic supercritical airfoil.

1.2 Literature Review

Based on the objectives of this study, the literature survey was conducted on the topics of wall boundary condition implementation, variable transformations, mean flow-RANS equation coupling and Jacobian construction methods using the source terms with regards to the two-equation RANS model implementations in the literature.

In regards to wall boundary condition implementations, Hellsten (1998) presented an improved solid-wall ω boundary condition based on wall-roughness in order to handle the drastic analytical behaviour of ω variable near walls and adjusted the SST blending function to strengthen numerical stability. This resulted in better skin friction predictions without near-wall grid sensitivity.

In addition, Eça & Hoekstra (2008) tested three ω wall-boundary options -finite analytic value, rough-wall correction, and setting ω not at the wall but at the nearest node -on a flat-plate case with systematic grid refinements. They concluded that k-ω model is far more sensitive to near-wall grid spacing than to the BC variant itself - where no extra accuracy was obtained by changing BCs compared to simply reducing the $y+$ values.

For positivity preservation, Ilinca and Pelletier (1999) transformed the k–ε and k–ω equations in terms of natural logarithms (i.e. $ln\ k$-$ln\ \omega$), which guarantees positivity of turbulence variables and eddy viscosity without using variable clipping. For k variable, they had to use a wall function due to vanishing k at the wall. They applied mesh adaptation inside a segregated solver for turbulence equations. The slower

variation of logarithmic variables resulted in more stable iterations and higher accuracy in shear-layer and backward-facing-step cases where it was possible to use coarser meshes.

As an alternative for maintaining solver stability, Moryossef and Levy (2006) designed an implicit scheme in which the k–ω Jacobian is constructed as an "M-matrix". This made sure that k, ω, and the eddy viscosity remain positive for any time step without clipping or source-term bounding. The M-matrix made the system diagonally dominant, resulting in stable solutions with good convergence in both steady and unsteady 2-D/3-D test cases.

In terms of model equations' coupling, Sinha and Candler (1998) joined the Navier–Stokes and k–ε equations in a fully-implicit, block-coupled matrix and linearised every turbulence-model source term within the Jacobian . They found that this was able to raise the maximum CFL numbers by two orders of magnitude, and cut runtime to roughly 10% of the original decoupled and approximate Jacobian methods.

Merci et al. (2000) replaced the commonly preferred sign-based splitting —implicit sinks, explicit sources— with a fully implicit discretisation that uses the exact Jacobian of the k–ω source terms within a multigrid solver. This helped with accelerating convergence and reaching steady-state roughly ten times faster on high-aspect-ratio boundary-layer and separated-flow tests while k and ω was also kept positive.

Lee and Choi (2005) compared a strongly coupled, ADI-based solver with a loosely coupled system that handle the two-variable turbulence block separately for transformed q–ω turbulence equations of Coakley (1992). Approximate factorization was applied for implicit handling of turbulence source terms. Transonic wing simulations showed very similar convergence histories and they concluded that total execution time favoured the loosely coupled approach.

Lian, Xia, and Merkle (2010) point that solvers break due to stiff production-and-dissipation terms in the k–ω equations and show that stability is achieved when all sink terms are treated implicitly while positive sources are handled explicitly. They observed that this eliminates the need for tight timestep limits and they were able to obtain steady-state convergence to machine accuracy across a range of turbulent internal flow test cases.

Finally, Content, Outtier, and Cinnella (2013) test the k-ω SST model in both fully coupled and decoupled fashion, both using a Jacobian-free Newton–Krylov (JFNK) solver which skips forming the full Jacobian. They only kept destruction terms of the turbulence sources in the implicit part to preserve diagonal dominance. They found that fully coupled JFNK variant reduced nonlinear iterations by roughly 25% compared with the decoupled mode in subsonic flat plate. However, while the coupled system converged for a transonic airfoil case, decoupled solution failed.

1.3 Thesis Outline

This thesis is structured as follows. Chapter 2 presents the standard k-ω SST turbulence model equations and their numerical aspects. Chapter 3 introduces the k-q transformation for improving stability k-ω SST model and enforcing positivity. Chapter 4 outlines the temporal and spatial discretization methods used in the HEMLAB solver. Chapter 5 discusses the coupling strategy between turbulence and mean-flow equations, including flux Jacobians. Chapter 6 describes the adaptive mesh refinement techniques and available tools in HEMLAB. Chapter 7 validates the model using benchmark subsonic & transonic 2D aerodynamic test cases. Chapter 8 concludes the study with key findings and future research directions regarding SST model implementation in HEMLAB.

2. STANDARD *k-ω* SST TURBULENCE MODEL EQUATIONS

NASA Turbulence Modeling Resource (TMR) website (NASA Langley Research Center, n.d.) provides the following format for the two-equation eddy-viscosity-based *k-ω* SST-m RANS turbulence model by Menter in a dimensional and conservative form as shown in equation 2.1, where the first is the turbulent kinetic energy (*k*) equation and the latter is the equation for specific turbulent dissipation rate (*ω*):

$$\begin{aligned}\frac{\partial(\rho k)}{\partial t} + \nabla\cdot(\rho \boldsymbol{u} k) &= P_{lim} \\ &\quad -\beta^*\rho\omega k \\ &\quad +\nabla\cdot[(\mu+\sigma_k\mu_t)\nabla k] \\ \frac{\partial(\rho\omega)}{\partial t} + \nabla\cdot(\rho \boldsymbol{u}\omega) &= \gamma\rho S^2 \\ &\quad -\beta\rho\omega^2 \\ &\quad +\nabla\cdot[(\mu+\sigma_\omega\mu_t)\nabla\omega] \\ &\quad +2(1-F_1)\frac{\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega\end{aligned} \tag{2.1}$$

where ρ is density, u is the velocity vector, μ is the dynamic viscosity, v is the kinematic viscosity, d is the nearest wall-normal distance to a wall node, Ω is the vorticity magnitude, W is the anti-symmetric rotation-rate tensor, $CD_{k\omega}$ is the positive part of cross-diffusion term. In addition, intermediate terms based on blending functions F_1 and F_2, and the model calibration constants are shown in equations 2.2 and 2.3:

$$\begin{aligned}&\sigma_k = F_1\sigma_{k1} + (1-F_1)\sigma_{k2} \\ &\sigma_w = F_1\sigma_{w1} + (1-F_1)\sigma_{w2} \\ &\beta = F_1\beta_1 + (1-F_1)\beta_2 \\ &\gamma = F_1\gamma_1 + (1-F_1)\gamma_2 \\ &a_1 = 0.31 \quad \beta^* = 0.09 \quad \kappa = 0.41 \\ &\gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} \quad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}} \\ &\sigma_{k1} = 0.85\ \sigma_{k2} = 1.0\ \sigma_{\omega 1} = 0.5\ \sigma_{\omega 2} = 0.856\ \beta_1 = 0.075\ \beta_2 = 0.0828\end{aligned} \tag{2.2}$$

$$
\begin{aligned}
P_{lim} &= min(P, 20\beta^* \rho\omega k) \\
P &= \mu_t S^2 \\
\mu_t &= \frac{\rho a_1 k}{max(a_1\omega, \Omega F_2)} \\
S &= \sqrt{2(\boldsymbol{S}\cdot\boldsymbol{S})} \\
\boldsymbol{S} &= \frac{1}{2}(\nabla\cdot\boldsymbol{u} + (\nabla\cdot\boldsymbol{u})^T) \\
\Omega &= \sqrt{2(\boldsymbol{W}\cdot\boldsymbol{W})} \\
\boldsymbol{W} &= \frac{1}{2}(\nabla\cdot\boldsymbol{u} - (\nabla\cdot\boldsymbol{u})^T) \\
F_2 &= tanh(arg_2^2) \\
arg_2 &= max(2\Gamma_1, \Gamma_2) \\
\Gamma_1 &= \frac{\sqrt{k}}{\beta^*\omega d} \\
\Gamma_2 &= \frac{500\nu}{d^2\omega} \\
F_1 &= tanh(arg_1^4) \\
arg_1 &= min(\Gamma_3, \Gamma_4) \\
\Gamma_3 &= max(\Gamma_1, \Gamma_2) \\
\Gamma_4 &= \frac{4\rho\sigma_{w2}k}{CD_{k\omega}d^2} \\
CD_{k\omega} &= max\left(2\rho\sigma_{w2}\frac{1}{\omega}\nabla k\cdot\nabla\omega, 10^{-20}\right)
\end{aligned}
\tag{2.3}
$$

In the standard *k-ω* SST turbulence model by Menter (1994), using Einstein's summation notation, the production term *P* in the turbulent kinetic energy equation is originally defined as in equation 2.4:

$$P = \tau_{ij}\frac{\partial u_i}{\partial x_j} \tag{2.4}$$

where τ_{ij} is the Reynolds stress tensor is given in equation 2.5:

$$\tau_{ij} = \mu_t\left(2S_{ij} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k\delta_{ij} \tag{2.5}$$

and δ_{ij} is the Kronecker-delta, μ_t is the turbulent viscosity which provides the eddy-viscosity based RANS closure, and S_{ij} is the mean strain rate tensor which is the symmetric part of velocity-gradient tensor given by equation 2.6:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.6}$$

However, production term is often approximated as $P = \mu_t S^2$ where the new model is tagged as ”-m” family and S is the magnitude of the strain rate tensor from equation 2.7:

$$S = \sqrt{2S_{ij}S_{ij}} \tag{2.7}$$

This approximation reduces the number of required operations in addition to being exact for incompressible flows and a very good approximation except for very high Mach number flows (NASA Langley Research Center, n.d.). For both k and ω equations, the left-hand side of the equality denotes the time derivative and convective terms while the right-hand side shows production, destruction, and diffusion terms, correspondingly. Only extra term is the ”cross-diffusion” term in ω equation which reduces the freestream sensitivity of the original k-ω model (Wilcox, 1988). As a summary, Table 2.1 shows each term classified for their contributions to model equations where it can be seen that model is in compressible and conservative form:

Table 2.1 : Term classification for the standard k-ω SST-m turbulence model equations.

Term	k-equation	ω-equation
Time derivative	$\frac{\partial(\rho k)}{\partial t}$	$\frac{\partial(\rho \omega)}{\partial t}$
Convection	$\nabla \cdot (\rho \mathbf{u} k)$	$\nabla \cdot (\rho \mathbf{u} \omega)$
Production	$\min(\mu_t S^2, 20\beta^* \rho \omega k)$	$\gamma \rho S^2$
Destruction	$-\beta^* \rho \omega k$	$-\beta \rho \omega^2$
Diffusion	$\nabla \cdot [(\mu + \sigma_k \mu_t)\nabla k]$	$\nabla \cdot [(\mu + \sigma_\omega \mu_t)\nabla \omega]$
Cross-Diffusion	-	$2(1 - F_1)\frac{\rho \sigma_{\omega 2}}{\omega} \nabla k \cdot \nabla \omega$

F_1 and F_2 terms in the model equations are termed as ”blending functions” which help recover either the k-ω behaviour towards the no-slip wall or k-ϵ formulation near the boundary layer edge towards freestream. Menter (1994) recommends the production limiter for cases involving stagnation points and to deal with extreme turbulent kinetic energy production in some adverse pressure gradient flows where in the standard model, the production term is typically limited to at most twenty times the destruction term.

2.1 Non-dimensionalization

The same reference quantities applied to the compressible Navier-Stokes equations in HEMLAB are also used for the compressible form of *k-ω* SST turbulence model equations. Following set of non-dimensional variables are obtained where ”tilde” is used for clarity on the dimensional quantities and ”∞” denotes freestream reference quantities as shown in equation 2.8:

$$\begin{aligned} k &= \frac{\tilde{k}}{\tilde{a}_\infty^2} \quad & \omega &= \frac{\tilde{\mu}_\infty \tilde{\omega}}{\tilde{\rho}_\infty \tilde{a}_\infty^2} \quad & \rho &= \frac{\tilde{\rho}}{\tilde{\rho}_\infty} \\ \boldsymbol{u} &= \frac{\tilde{\boldsymbol{u}}}{\tilde{a}_\infty} \quad & \mu &= \frac{\tilde{\mu}}{\tilde{\mu}_\infty} \quad & x &= \frac{\tilde{x}}{\tilde{L}_\infty} \\ t &= \frac{\tilde{a}_\infty \tilde{t}}{\tilde{L}_\infty} \quad & \Omega &= \frac{\tilde{\Omega} \tilde{L}_\infty}{\tilde{a}_\infty} \quad & S &= \frac{\tilde{S} \tilde{L}_\infty}{\tilde{a}_\infty} \end{aligned} \tag{2.8}$$

When applied to original set of equations, following non-dimensional and conservative forms in equation 2.9 are obtained:

$$\begin{aligned} \frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} k) &= P_{lim} \\ &\quad - \frac{Re_\infty}{Ma_\infty} \beta^* \rho \omega k \\ &\quad + \frac{M_\infty}{Re_\infty} \nabla \cdot [(\mu + \sigma_k \mu_t) \nabla k] \\ \frac{\partial(\rho \omega)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \omega) &= \frac{M_\infty}{Re_\infty} \gamma \rho S^2 \\ &\quad - \frac{Re_\infty}{Ma_\infty} \beta \rho \omega^2 \\ &\quad + \frac{M_\infty}{Re_\infty} \nabla \cdot [(\mu + \sigma_\omega \mu_t) \nabla \omega] \\ &\quad + \frac{M_\infty}{Re_\infty} 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \nabla k \cdot \nabla \omega \end{aligned} \tag{2.9}$$

where intermediate terms also involve non-dimensional terms as shown in equation 2.10 and previous categorization is repeated in Table 2.2 with non-dimensional terms:

$$
\begin{aligned}
\frac{Re_\infty}{M_\infty} &= \frac{\rho_\infty a_\infty L_\infty}{\mu_\infty} \\
P_{lim} &= min\left(\frac{M_\infty}{Re_\infty}\mu_t S^2, \frac{Re_\infty}{Ma_\infty} 20\beta^* \rho\omega k\right) \\
\mu_t &= \frac{\rho a_1 k}{max\left(a_1\omega, \left(\frac{M_\infty}{Re_\infty}\right)\Omega F_2\right)} \\
S &= \sqrt{2(\boldsymbol{S}\cdot\boldsymbol{S})} \\
\boldsymbol{S} &= \frac{1}{2}(\nabla\cdot\boldsymbol{u} + (\nabla\cdot\boldsymbol{u})^\top) \\
\Omega &= \sqrt{2(\boldsymbol{W}\cdot\boldsymbol{W})} \\
\boldsymbol{W} &= \frac{1}{2}(\nabla\cdot\boldsymbol{u} - (\nabla\cdot\boldsymbol{u})^\top) \\
F_2 &= tanh(arg_2^2) \\
arg_2 &= max(2\Gamma_1, \Gamma_2) \\
\Gamma_1 &= \frac{M_\infty}{Ree_\infty}\frac{\sqrt{k}}{\beta^*\omega d} \\
\Gamma_2 &= \left(\frac{M_\infty}{Ree_\infty}\right)^2 \frac{500\nu}{d^2\omega} \\
F_1 &= tanh(arg_1^4) \\
arg_1 &= min(\Gamma_3, \Gamma_4) \\
\Gamma_3 &= max(\Gamma_1, \Gamma_2) \\
\Gamma_4 &= \frac{4\rho\sigma_{w2}k}{CD_{k\omega}d^2} \\
CD_{k\omega} &= max\left(2\rho\sigma_{w2}\frac{1}{\omega}\nabla k\cdot\nabla\omega, 10^{-20}\right)
\end{aligned}
\tag{2.10}
$$

Table 2.2 : Term classification for the non-dimensional *k*-ω SST-m turbulence model equations.

Term	*k*-Equation	ω-Equation
Time derivative	$\frac{\partial(\rho k)}{\partial t}$	$\frac{\partial(\rho\omega)}{\partial t}$
Convection	$\nabla\cdot(\rho\mathbf{u}k)$	$\nabla\cdot(\rho\mathbf{u}\omega)$
Production	$\min\left(\frac{M_\infty}{Re_\infty}\mu_t S^2, \frac{Re_\infty}{Ma_\infty}20\beta^*\rho\omega k\right)$	$\frac{M_\infty}{Re_\infty}\gamma\rho S^2$
Destruction	$-\frac{Re_\infty}{Ma_\infty}\beta^*\rho\omega k$	$-\frac{Re_\infty}{Ma_\infty}\beta\rho\omega^2$
Diffusion	$\frac{M_\infty}{Re_\infty}\nabla\cdot[(\mu+\sigma_k\mu_t)\nabla k]$	$\frac{M_\infty}{Re_\infty}\nabla\cdot[(\mu+\sigma_\omega\mu_t)\nabla\omega]$
Cross-Diffusion	-	$\frac{M_\infty^\infty}{Re_\infty}2(1-F_1)\frac{\rho\sigma_{\omega2}}{\omega}\nabla k\cdot\nabla\omega$

2.2 Main Aspects of the *k-ω* SST Model

2.2.1 Blending functions

The two-equation *k-ω* SST model can be derived by first transforming variables of the high-Re *k-ϵ* model (without damping functions) into a *k-ω* formulation. Then, combining the transformed model multiplied by 1 – F_1 while the original *k-ω* model by Wilcox (1988) is multiplied by F_1, the *k-ω* SST model is obtained (Menter, 1994). Here, the F_1 acts as a ”blending function” where it goes to 1 in the logarithmic and viscous sublayer portions of the boundary layer, while as boundary layer edges are approached, it gradually becomes 0. This means that *k-ω* SST model can automatically switch between *k-ϵ* and *k-ω* models depending on the cell in which the calculations take place. This function is utilized on obtaining model constants as in equation 2.11:

$$\phi_{SST} = F_1\sigma_{\phi_1} + (1 - F_1)\sigma_{\phi_2} \tag{2.11}$$

where $\sigma_{\phi 1}$ is the constant related to *k-ω* and $\sigma_{\phi 2}$ comes from the $k-\epsilon$ model. In addition, the cross-diffusion term of *k-ω* SST model also arises inside of the *ω* equation due to blending operation as in equation 2.12:

$$\frac{M_\infty}{Re_\infty} 2(1 - F_1) \frac{\rho\sigma_{\omega 2}}{\omega} \nabla k \cdot \nabla\omega \tag{2.12}$$

2.2.1.1 Numerical implications

Here, for a smooth switch between different models based on boundary layer regions, a ”*tanh*” function is preferred. Since arg_1 will attain a high value a due to inverse-wall distance relationships inside its definition, $\arg_1$ is raised to the fourth power inside the ”tanh” function to make sure that F_1 reaches 1 very quickly as the wall is approached and *k-ω* model is recovered where equation 2.13 shows the blending function and its parameter $\arg_1$.

$$\begin{aligned} F_1 &= \tanh(\arg_1^4) \\ \arg_1 &= \min\left\{\max\left(\frac{M_\infty}{Re_\infty}\frac{\sqrt{k}}{\beta^*\omega d}, \left(\frac{M_\infty}{Re_\infty}\right)^2 \frac{500\nu}{d^2\omega}\right), \frac{4\rho\sigma_{\omega 2}k}{C_{Dk\omega}d^2}\right\} \end{aligned} \tag{2.13}$$

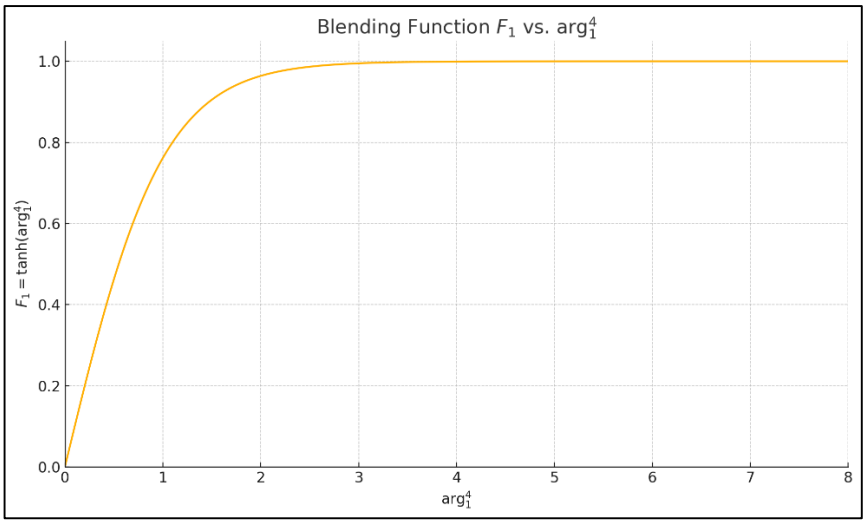

Figure 2.1 : Behaviour of F_1 blending function in SST model.

Similarly, as freestream is approached inside the boundary layer, F_1 rapidly becomes 0 as shown on Figure 2.1 above. Biggest disadvantage comes from the fact that ”closest wall-distance” parameter d shown below for a simple hex grid on Figure 2.2 is required for blending function calculations at each iteration. Although for fixed meshes this is computed only once in the geometry pre-processing step before the start of the simulation and SST model is on par with Wilcox’s original model in terms of computations per timestep with a slightly increased programming workload. However, it can be a computational burden for moving meshes in transient simulations (Menter, 1994). Still, the main advantage is that blending operations help reduce the freestream sensitivity associated with the k-ω model and increases modeling accuracy since k-ϵ does not suffer from this sensitivity.

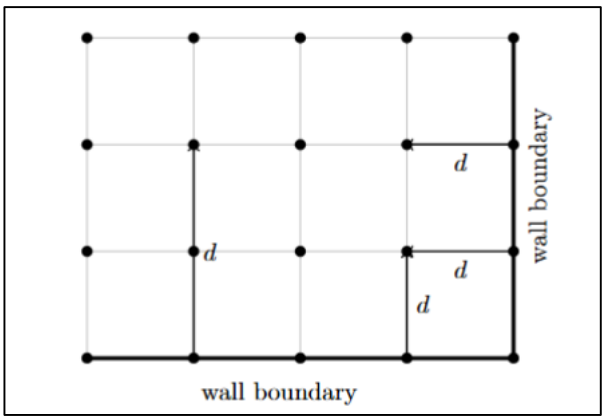

Figure 2.2 : Rectangular grid illustrating wall-to-node distances.

Although F_1 and F_2 behave similarly, F_2 extends further into upper regions of boundary layer as shown below on Figure 2.3 for a flat plate case (NASA Langley

Research Center (n.d.)), while F_1 switches earlier to avoid freestream dependence (Menter, 1994):

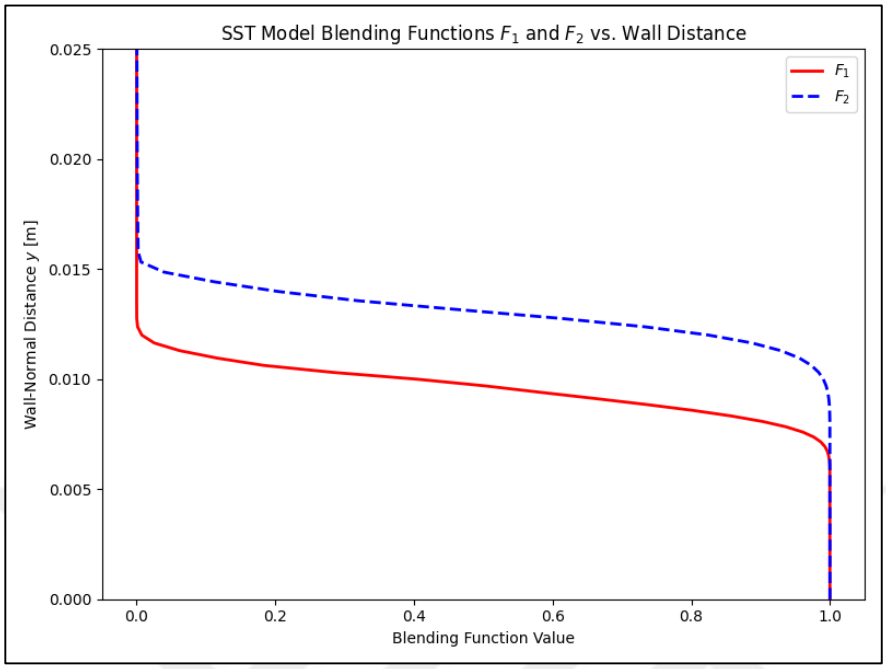

Figure 2.3 : F_1 and F_2 vs wall distance (Adapted from FUN3D NASA Langley Research Center (n.d.)).

2.2.2 Stress limiter

When the blending between k-ϵ and k-ω models are handled via only F_1 and the eddy viscosity is simply defined as in equation 2.14:

$$\mu_t = \frac{\rho k}{\omega} \tag{2.14}$$

this is called the ”BSL model” (Menter, 1994). However, when a second blending function F_2 is used to modify the definition of μ_t so that eddy-viscosity (non-dimensionalized form) is limited with a maximum function as in equation 2.15:

$$\mu_t = \frac{\rho a_1 k}{max\left(a_1\omega, \left(\frac{M_\infty}{Re_\infty}\right)\Omega F_2\right)} \tag{2.15}$$

where F_2 is again another smooth ”*tanh*” function and its argument arg_2 is defined by equation 2.16:

$$F_2 = tanh(arg_2^2)$$
$$arg_2 = max\left(\frac{M_\infty}{Re_\infty}\frac{2\sqrt{k}}{\beta^*\omega d}, \left(\frac{M_\infty}{Re_\infty}\right)^2 \frac{500\nu}{d^2\omega}\right) \quad (2.16)$$

This version is called the "*k-ω* Shear Stress Transport (SST) model" (Menter, 1994). Naming is due to the fact that Reynolds-Stress-Models succesfully accounts for shear-stress transport with additional equations for RANS closure which improves adverse pressure gradient flow predictions. SST variant also tries to involve the effects of shear-stress transport by incorporating the Bradshaw's assumption shown in equation 2.17 that shear-stress and the turbulent kinetic energy are proportional in the boundary layer (Menter, 1994):

$$\tau_{\text{turb}} = \rho a_1 k \quad (2.17)$$

where a_1 is constant. In addition, turbulent shear stress in eddy-viscosity based RANS models is usually defined by equation 2.18:

$$\tau_{turb} = \mu_t \frac{\partial u}{\partial y} = \mu_t \Omega \quad (2.18)$$

Therefore, using these relations the new definition of eddy-viscosity is shown equation 2.19:

$$\mu_t = \frac{\rho a_1 k}{\Omega} \quad (2.19)$$

For adverse pressure gradient flows, production becomes much larger than dissipation, where production can be associated with Ω and dissipation is with ω. If the above relation is modified again as in equation 2.20:

$$\mu_t = \frac{\rho a_1 k}{max(a_1\omega, \Omega)} \quad (2.20)$$

This results in the model selecting the unlimited form automatically for adverse pressure gradient regions where production, therefore Ω is high, while the very first μ_t definition is used for the rest of the boundary layer. Finally, Bradshaw's assumption is not satisfied in free-shear layer flows (Menter, 1994). Therefore, the blending function F_2 is utilized in the limiter as shown in equation 2.21 in order to switch to

original μ_t formulation in these regions where F_2 behaves similar to the blending function F_1 by reaching 0 towards boundary layer edge.

$$\mu_t = \frac{\rho a_1 k}{max(a_1\omega, \Omega F_2)} \tag{2.21}$$

2.2.2.1 Numerical implications

Just like the dynamic viscosity, eddy viscosity is a scalar positive quantity as well. Therefore, care should be taken to make sure either μ_t or its constituting components in the above formulae are limited appropriately and a positive number is outputted. A mistake in this regard will propagate into other terms involving μ_t such as the production and diffusion terms. Since all eddy-viscosity models provides a RANS closure through μ_t, this will also affect the Navier-Stokes equations and the exact Jacobian construction and full-coupling of the turbulence model. In addition to positivity, upper-bound limits can be applied to μ_t as well (NASA Langley Research Center (n.d.)). In HEMLAB, instead of this, limiting the vorticity magnitude (Ω) by a lower-bound of 10^{-10} was preferred to prevent excessive values of μ_t.

2.2.3 Production limiter

Instead of the original formulation where production term in equation 2.22 is linear in velocity gradients:

$$P = \tau_{ij}\frac{\partial u_i}{\partial x_j} \tag{2.22}$$

SST-m model changes the *k*-equation production term to be quadratic as in equation 2.23:

$$P = \mu_t S^2 \tag{2.23}$$

The non-linear strain rate may result in problems where strain rate is not caused by actual shearing effects but instead due to non-zero velocity gradients of the inviscid flow such as encountered near the stagnation regions of leading edges of airfoils (ANSYS Inc., 2020). In such cases, excessive eddy-viscosity build-up can be observed. Menter (1994) suggests applying a limiter to the production term in relation

to the dissipation term of *k*-equation as follows (in non-dimensional form shown by equation 2.24):

$$P_{lim} = min\left(\frac{M_\infty}{Re_\infty}\mu_t S^2, \frac{Re_\infty}{Ma_\infty} 20\beta^* \rho\omega k\right) \tag{2.24}$$

2.2.3.1 Numerical implications

The given limiter factor of 20 is considered to be rather conservative, since usually most calibration flows do not result in P_k / D_k ratios greater than 2 and it is purely intended to avoid numerical spikes (Menter, 1994). In addition, due to the minimum function involved, programming gets more complex in exact Jacobian construction with derivatives of source terms. This also makes convergence more problematic since any ”limiter” where the true value has to be temporarily replaced with an unphysical value effectively stalls the convergence of the Newton method.

2.2.4 Other model variants

Beyond the baseline BSL and SST formulations, several other variants of the *k–ω* SST model have been suggested to address specific turbulence-related physical phenomena or numerical limitations (ANSYS Inc., 2020). These modifications are usually made to the production terms, eddy-viscosity definition, or by the adding extra transport equations. Following subsections briefly introduce commonly used SST variants as listed in NASA TMR (NASA Langley Research Center, (n.d.)).

2.2.4.1 SST-V

SST-V variant replaces the standard strain-rate based turbulent kinetic energy production term with a vorticity-based form in equation 2.25:

$$P = \mu_t \Omega^2 \tag{2.25}$$

where vorticity magnitude $\Omega = \sqrt{\boldsymbol{2W_{ij}W_{ij}}}$ and W_{ij} is the antisymmetric rotation-rate tensor. Here, the turbulence production is directly connected to local vorticity rather than strain. Still, the definition of μ_t is still based on strain rate. Without modifying the diffusion or dissipation mechanisms, SST-V is shown to improve predictions for hypersonic flow simulations, especially for strong bow shock cases (NASA Langley Research Center, (n.d.)).

2.2.4.2 SST-2003

The main change is that instead of vorticity as was shown in 1994 model, eddy viscosity definition is now based on strain rate as shown in equation 2.26 (Menter et al., 2003):

$$\mu_t = \frac{\rho a_1 k}{max(a_1 \omega, SF_2)} \tag{2.26}$$

In addition, this revision recalibrates model coefficients from 1994 version. Production limiter coefficient is reduced from 20 to 10 and applied to not only *k* but also to the ω equation. Therefore, the production terms of the *k* and ω equations become as shown in equation 2.27:

$$\begin{aligned} P_k &= min(\mu_t S^2, 10\beta^* \rho \omega k) \\ P_\omega &= \frac{\gamma}{\nu_t} min(\mu_t S^2, 10\beta^* \rho \omega k) \end{aligned} \tag{2.27}$$

Cross-diffusion lower bound is set to 10^{-10} instead of 10^{-20}. Values of γ_1 and γ_2 are updated to 5/9 and 0.44, respectively.

2.2.4.3 SST-KL

Similar to SST-V, Kato–Launder (KL) variant modifies the production term as shown in equation. 2.28 (Kato and Launder, 1993):

$$P = \mu_t S \Omega \tag{2.28}$$

Since most turbulence models are calibrated based on shear flows where the strain rate and vorticity magnitudes are nearly equal ($S = \Omega$), applying the Kato–Launder correction does not alter the original model calibration for these cases. However, in inviscid regions where the vorticity vanishes ($\Omega = 0$), the modification suppresses turbulence production (ANSYS, Inc., 2020). This modification blends the strain rate and vorticity magnitudes in the production limiter and it is shown to reduce unphysical turbulence buildup near the stagnation region of airfoils during transitional flow simulations. However, its applicatio may interfere with other regions involving rotational flow or curvature effects, and should be used with care (ANSYS, Inc., 2020).

2.2.4.4 SST-sust

This model adds the following ”ambient source terms” in equation 2.29 correspondingly to both *k* and ω equations of *k*-ω SST model in order to sustain turbulence outside the boundary layer (Spalart and Rumsey, 2007):

$$+\beta^{*}\rho\omega_{\infty}k_{\infty}, \qquad +\beta\rho\omega_{\infty}^{2} \tag{2.29}$$

where typical freestream values are $\omega_{\infty} = 5U_{\infty}/L$ and $k_{\infty} = 10^{-6}\,U_{\infty}^{2}$ with *L* being a ”reference geometric length” (NASA Langley Research Center, (n.d.)). Here, everything except the sustaining source terms are identical to the standard SST model. These new terms help eliminate the non-physical decay of *k* and ω from the freestream for external aerodynamics cases, balance the destruction terms in the freestream when turbulence levels are equal to the given ambient levels, and maintain a realistic background turbulence level (Spalart and Rumsey, 2007; NASA Langley Research Center, (n.d.)).

2.2.4.5 SST-RC

Turbulence is enhanced on highly concave surfaces and damped for the convex surfaces. Standard eddy-viscosity models do not account for this and need to be sensitized (ANSYS Inc., 2020). The rotation-curvature (RC) correction introduces a multiplication factor fr_1 as shown in equation 2.30 in front of the production terms of both *k* and ω equations to account for high curvature and system rotation (Smirnov & Menter, 2009):

$$\begin{aligned} f_{r1} &= max[min(f_{rot}(r^{*}), 1.25), 0] \\ f_{rot} &= \frac{4r^{*}}{1+r^{*}}(1 - tan^{-1}(2\hat{r})) - 1, r^{*} = \frac{S}{\Omega} \\ \hat{r} &= \frac{2W_{ik}S_{jk}}{WD^{3}}\left(\frac{DS_{ij}}{Dt} + \left(\varepsilon_{imn}S_{jn} + \varepsilon_{jmn}S_{in}\right)\Omega_{m}^{rot}\right) \\ D^{2} &= max(S^{2}, 0.09\omega^{2}) \end{aligned} \tag{2.30}$$

In equation 2.30, the term DS_{ij}/Dt is used for the components of the Lagrangian derivative of the strain rate tensor. SST-RC model improves modeling accuracy in cases of highly curved or rotating flows, in rotating frame simulations and especially for vortical flows seen in cases of high angle of attack fighter jet simulations (Smirnov & Menter, 2009; ANSYS Inc., 2020). It is shown that models without curvature correction produce too high eddy-viscosity levels inside vortices (ANSYS, Inc., 2020).

2.2.4.6 SST-LM

Langtry & Menter (2009) introduced the *k-ω* SST Langtry–Menter (LM) four-equation transition model which adds two convection-diffusion type transport equations to the SST-2003 model: first for the intermittency variable (γ) whose production increase controls the turbulence levels in the boundary layer (NASA Langley Research Center (n.d.)). Here, freestream BC is set to $\gamma_\infty = 1$, since the original model behavior (i.e. fully-turbulent) needs to be recovered at the freestream (Langtry & Menter, 2009). Second equation is for the transition momentum-thickness Reynolds number variable (Re_θ), in order to determine when laminar flows switch to turbulent (transition onset location in terms of freestream turbulence intensity levels). Both variables have zero-gradient wall BCs.

It should be noted that this model is ”correlation-based” and actually does not try to model the transition physics directly Langtry & Menter (2009). The modified *k*-equation production term becomes $\gamma\mu_t S^2$, so that until γ grows from 0 to 1, turbulence is suppressed (NASA Langley Research Center, (n.d.)). Calibration constants together with empirical correlations for Re_θ enable more accurate predictions for transition phenomena, which is of critical value for efficient (high *L/D* ratio) aircraft design, turbomachinery blade simulations and improved heat transfer predictions. However, the model is not suitable for fully-developed channel and pipe flows since SST γ-Re_θ model requires a freestream region (ANSYS Inc, 2020).

2.2.4.7 *k-ω* TNT

The turbulent/nonturbulent (TNT) formulation is a direct modification of Wilcox’s original *k–ω* model, not of Menter’s SST. Starting from Wilcox’s model equations and using a set of 1-D diffusion equations which is a model of the freestream edges of turbulent regions, Kok (2000) showed that a weak solution is a moving front in $+y$ direction with positive velocity and derived the following ”diffusion coefficients (TNT set)” in equation 2.31:

$$\sigma_k = \frac{2}{3}, \sigma_\omega = 0.5, \sigma_d = 0.5 \tag{2.31}$$

in addition to a permanent cross-diffusion term shown in equation 2.32 which is always active and never needs blending:

$$CD = \sigma_d \rho \frac{1}{\omega} \max(\nabla k \cdot \nabla \omega, 0) \tag{2.32}$$

In addition, no additional wall-distance parameter or low-Re damping is required (Kok, 2000). These coefficients satisfy the interface constraints in equation 2.33:

$$\begin{aligned} \sigma_\omega - \sigma_k + \sigma_d &> 0, \\ \sigma_k - \sigma_d &> 0, \\ \sigma_\omega - \sigma_k + \sigma_d &\leq \sigma_k \sigma_\omega \\ \sigma_\omega &> 0, \\ \sigma_k &> 0.5. \end{aligned} \tag{2.33}$$

which guarantee that a turbulent front always moves into the nonturbulent region and the solution becomes independent of the freestream value of ω where the base Wilcox model cannot get rid of this dependency since the model coefficients do not satisfy the constraints (Kok, 2000). Solvers that already implement Menter's SST equations can activate TNT by forcing the blending function $F_1 = 1$ (thus disabling the k–ε and k-ω blending), and substituting the diffusion coefficients above. The TNT variant can be used in external aerodynamics cases with poorly known freestream turbulence in order to make results such as skin-friction distributions and shock predictions insensitive to ω_∞. In addition, it can be used on unstructured grids where wall-distance evaluation and SST blending would add complexity. If transition modeling is later required, TNT variant can be coupled without extra changes because its production and destruction terms are identical to the baseline Wilcox model (Kok, 2000).

2.2.5 Initial and boundary conditions

For standard k-ω SST model, non-dimensional strong-boundary conditions for k and ω are as shown in equation 2.34 (Menter ,1994; NASA Langley Research Center, (n.d.)):

$$\begin{aligned} k|_{wall} &= 0 \\ k|_{farfield} &= 9 \times 10^{-9} \\ \omega|_{wall} &= \left(\frac{M_\infty}{Re_\infty}\right)^2 \frac{60 \nu_w}{0.075(\Delta y_1)^2} \\ \omega|_{farfield} &= 1 \times 10^{-6} \end{aligned} \tag{2.34}$$

where ν_w is the wall kinematic viscosity, and Δy_1 is the wall normal distance of the nearest node away from the wall. These values correspond to a turbulent viscosity ratio (μ_t/μ_∞) level of 0.009 in the freestream. No wall functions were implemented for

current implementation with a $y+ < 3$ requirement for the model (Menter, 1994). All nodal values in the domain are initialized with fixed freestream values k_∞ and ω_∞. The original reference suggests the following dimensional freestream boundary condition ranges in equation 2.35 for k and ω where Re_L is based on a reference domain length L and U_∞ is the freestream velocity:

$$\begin{aligned} \frac{U_\infty}{L} < \omega_{farfield} < 10\frac{U_\infty}{L} \\ \frac{10^{-5}U_\infty^2}{Re_L} < k_{farfield} < \frac{0.1U_\infty^2}{Re_L}. \end{aligned} \tag{2.35}$$

The study involving the Table 2.3 below by NASA Langley Research Center (n.d.) shows that as long as the freestream turbulence levels are in a reasonable range, their impact on the solution is minimal - less than 0.1 percent variation for all cases is observed. This also verifies the "freestream independence" of the BSL & SST models compared to the Wilcox's model thanks to the blending functions (Menter ,1994). Still, it is stated that for fully turbulent flows, the turbulence levels must be high enough to activate turbulence at the leading edge, but not very unrealistic such that they distort the turbulence fields (NASA Langley Research Center (n.d.)).

Table 2.3 : Freestream insensitivity study of k-ω SST model tested by NASA using CFL3D NASA Langley Research Center (n.d.) for the benchmark zero-pressure gradient flat plate case.

μ_t/μ_∞	C_d	C_f at $x = 0.97$
0.009 (default)	0.284557	0.00268299
0.001	0.284553	0.00268295
0.1	0.284569	0.00268311
1.0	0.284575	0.00268317
10.0	0.284576	0.00268318

2.2.5.1 Analytical value of ω at the wall

Exact solution of the ω transport equation in the wall-normal direction of the viscous sublayer when diffusion and dissipation balance each other, convective terms are dropped and the μ_t is equal to 0 is given by Wilcox (1993) as shown in equation 2.36:

$$\omega = \frac{6\nu}{\beta y^2} \tag{2.36}$$

where y is the wall-normal distance, $\beta = 0.075$ and ν is kinematic viscosity. It is clear that at the wall where $y = 0$, analytically $\omega \rightarrow \infty$.

2.2.5.2 Numerical implications

Above analytical expression causes numerical implementation problems regarding wall BC of ω since a ”finite” number has to be provided in the solution. To represent this behaviour numerically, Wilcox (1993) suggested putting 7 to 10 grid nodes in the viscous sublayer and apply the above relation on all these nodes. However, it is stated that this brings an arbitrariness in the number of nodes with it which causes a grid-dependence as well as the difficulty of making sure all nodes are actually inside the viscous sublayer (Eça & Hoekstra, 2004). As an even simpler approximation, Menter et. al. (2003) proposed to apply the following BC only once at the wall, as long as $y+$ is less than 3 as shown in equation 2.37:

$$\omega|_{wall} = \frac{60\nu_w}{0.075(\Delta y_1)^2} \tag{2.37}$$

where Δy_1 is either the first cell-center or node’s wall-normal distance. However, although computationally less intensive than applying on multiple layer of nodes as in Wilcox’s approach, this is still a relation which explicitly depends on the wall-normal distance Δy, which again makes the model grid-independent since no two BC values will be the same numerically when the boundary layer mesh is refined or coarsened at the first cell (Eça & Hoekstra, 2004). Another suggested alternative is to apply the analytical formula not at the wall as Menter did, but on the first available node/cell-center adjacent to the wall (y_2) as shown in equation 2.38 – which still requires making sure that this node is in the viscous sublayer (Eça & Hoekstra, 2004).

$$\omega_2 = \frac{6\nu_2}{0.075(y_2)^2} \tag{2.38}$$

Finally, Wilcox (1993) states that since on a rough surface, ω has a finite value given by $\omega_w = \frac{u_\tau^2}{\nu} S_R$, where $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity and S_R is a non-dimensional function defined in equation 2.39:

$$S_R = \begin{cases} \left(\frac{50}{k_s^+}\right)^2, & k_s^+ < 25 \\ \frac{100}{k_s^+}, & k_s^+ \geq 25 \end{cases} \tag{2.39}$$

where k_s+ is the non-dimensional sand-grain roughness height showin in equation 2.40 (Hellsten, 1998):

$$k_s^+ = \frac{u_\tau k_s}{\nu} \tag{2.40}$$

This "rough-wall" boundary condition was considered as well to get rid of grid-independence as long as $k_s+ < 5$ for simulating smooth-surfaces (Wilcox, 1993; Eça & Hoekstra, 2004; Hellsten, 1998). However, this was shown to be unable to produce smooth wall solutions for turbulence fields (Eça & Hoekstra, 2004).

2.2.6 Positivity of *k* and *ω*

Kalitzin et. al. (2005) shows that the near-wall behaviour for *k* and *ω* variables has the analytical forms which are summarized in Table 2.4 below:

Table 2.4: Near-wall forms for the standard $k-\omega$ model (Kalitzin et al., 2005).

Boundary layer region	Non-dimensional quantity	Analytical form
Viscous sublayer	$\omega+$	$\frac{6}{\beta_1 (y_+)^2}$
Viscous sublayer	$k+$	$C_k (y_+)^{3.23}$
Log layer	$\omega+$	$\frac{1}{\kappa\sqrt{C_\mu}}\left(\frac{1}{y_+}\right)$
Log layer	$k+$	$\frac{1}{\sqrt{C_\mu}}$

The table contrasts the behaviour of the turbulence variables in the near-wall and logarithmic boundary layer regions where $y+$ is the non-dimensional wall distance unit and C_μ and κ are positive constants. Near the wall, the specific-dissipation rate (ω) rises quickly, while the turbulent kinetic energy (k) vanishes. This makes the specific-dissipation rate the main source of numerical stiffness and positivity issues. In the logarithmic region, the rate of increase of the specific-dissipation rate slows down and the turbulent kinetic energy is actually constant. As can be clearly seen, both variables stay positive everywhere.

Numerical schemes for the mean-flow and turbulence equations must also keep all variables positive. For the Navier-Stokes system, this means that density, pressure, and temperature must stay positive at all times. When higher-order flux reconstructions are used for the discretization of mean-flow equations (density, momentum, energy) and the numerical scheme overshoots due to shocks or strong/unlimited gradients, spikes might occur in strain rate. This can still affect the production and destruction terms of k and ω model and make the variables negative, even though SST model is only implemented with first order convection terms. Aggressive CFL ramping and production-destruction imbalance during solution initialization/start-up are other causes of overshoots that result in negative quantities (Swanson, 2021). Meanwhile, the quadratic, rapidly increasing nature of near-wall ω behaviour shown in Table 2.4 above is also prone to make ω negative if the numerical implementation is not robust. Since both turbulent kinetic energy k and specific dissipation rate ω are also physically positive scalar quantities in turbulent flows, k-ω SST model breaks down during flux calculations if any of these two variables becomes negative and are used without any countermeasures, because the eddy viscosity $\mu_t = \rho k/\omega$ has to change its sign and this would make the solver unstable since this must be also positive by definition in the RANS closures of Reynolds shear stresses (ANSYS Inc., 2020). In addition, mathematical operations inside the standard SST model formulation requires taking square-root of the k variable - which would again cause fatal errors for negative quantities. Furthermore, blending function F_1 depends on k and ω where an incorrect sign means improper activation and switch which then would lead to unwanted model behaviour. Finally, productions as well as destruction terms would change sign which results in unphysical flow evolution.

Available solutions from the literature are listed as follows (Swanson, 2021): (i) direct/forced variable clipping where nodal k and ω values are limited using a maximum function as $max(k,0)$ and $max(\omega,0)$ whenever k or ω becomes negative; (ii) variable transformations which mathematically keep the state of original k or ω positive; and (iii) forming an M–matrix for the Jacobian so that implicit solution of each timestep/iteration results in a non-negative variable state (Moryossef & Levy, 2006). Although clipping of nodal values is simple, it can impact the Newton method convergence negatively just like the slope limiter (Swanson, 2021). In HEMLAB, instead of nodal clipping, reconstructed k and ω values at the dual-volume face

midpoints are limited using the above direct clipping during flux and source term calculations. M-matrix methods are costly yet robust; however the limit maximum CFL number can reach may still be limited (Swanson, 2021). Langer & Swanson (2024) shows an alternative method which repeatedly reduces/damps the Newton update by half until new values of k and ω are both positive although the convergence may slow down.

2.2.6.1 Variable transformations

Variable transformations are the preferred methods for positivity requirements where transformations such as $\sqrt{k}$-$\sqrt{\omega}$, $ln(k)$-$ln(\omega)$, $log(k)$-$log(\omega)$ exist. Inverse versions such as $1/\sqrt{\omega}$ or single-variable variants such as $k-ln(\omega)$ are implemented as well (Swanson, 2021). Logarithmic forms guarantee positivity during solution and they are favorable since logarithm of variables vary more smoothly than variables themselves (Ilinca & Pelletier, 1999); however they require a wall function since $k = 0$ at the wall (Swanson, 2021). A very critical matter is the fact that implementing typical ω wall boundary condition on unstructured anisotropic adaptive meshes becomes numerically prone to strong instabilities due to Δy term involved in the standard wall boundary condition. In anisotropic meshes near wall, wall spacing can greatly vary between stretched neighbour elements as well as near vertices.

Therefore, to maintain robustness on such grids, this study implements the $k-q$ formulation with "inverse-square-root form" for ω variable in HEMLAB. It avoids the issues regarding quadratic ω wall behaviour where the transformed wall boundary condition for now simply becomes $q = 0$ (Dirichlet-type just like BC for k variable). In addition to the grid-dependence issues, quadratic nature of the above formulae makes the linear interpolations associated with the second-order finite volume method unable to represent the near-wall ω behaviour properly. This transformation helps obtain a linear near wall behaviour.

It should be noted that that any transformation must still result in a positive eddy viscosity and must satisfy the original untransformed equations' boundary conditions (Langer & Swanson, 2024).

3. *k-q* TRANSFORM FOR THE NON-DIMENSIONAL *k-ω* SST MODEL

In HEMLAB SST implementation, a transformation is applied to ω variable while the k is not transformed where a new variable q is defined in equation 2.40 such that transformed *k-ω* SST-m equations (now called as *k-q* SST-m) are shown in equations 2.41 and 2.42 where A is a positive constant:

$$q = \frac{\sqrt{A}}{\sqrt{\omega}} \Rightarrow \omega = \frac{A}{q^2} \tag{2.40}$$

$$\begin{aligned} \frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} k) = P_{lim} \\ - \frac{Re_\infty}{Ma_\infty} \beta^* \rho \frac{A}{q^2} k \\ + \frac{M_\infty}{Re_\infty} \nabla \cdot [(\mu + \sigma_k \mu_t) \nabla k] \end{aligned} \tag{2.41}$$

$$\begin{aligned} \frac{\partial(\rho q)}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} q) = -\frac{q^3}{2A} \frac{M_\infty}{Re_\infty} \gamma \rho S^2 \\ + \frac{Re_\infty}{Ma_\infty} \beta \rho \frac{A}{2q} \\ + \frac{M_\infty}{Re_\infty} \nabla \cdot [(\mu + \sigma_\omega \mu_t) \nabla q] - \frac{3}{q} \frac{M_\infty}{Re_\infty} \Gamma |\nabla q|^2 \\ + \frac{M_\infty}{Re_\infty} \frac{2}{A} q^2 (1 - F_1) \rho \sigma_{\omega 2} \nabla k \cdot \nabla q \end{aligned} \tag{2.42}$$

Any Dirichlet condition previously specified for ω (e.g. ω_{wall} or ω_∞ in freestream) must be transformed into a boundary condition q using the relation $\boldsymbol{\omega} = \frac{\boldsymbol{A}}{\boldsymbol{q^2}}$ as shown in

equation 2.43. In addition, equations 2.44 to 2.47 show the production term, stress limiter, blending functions, cross diffusion term, intermediate terms and constants:

$$q_{wall} = \sqrt{\frac{A}{\omega_{wall}}} = 0, \text{ where analytically } \omega_{wall} = \infty$$
$$q_{\infty} = \sqrt{\frac{A}{\omega_{\infty}}}, \text{ where previously } \omega_{\infty} = 1 \times 10^{-6} \tag{2.43}$$

3.1 Production and Stress Limiter Terms

$$P_{lim} = min\left(\frac{M_{\infty}}{Re_{\infty}}\mu_t S^2, \frac{Re_{\infty}}{Ma_{\infty}} 20\beta^* \rho \frac{A}{q^2} k\right)$$
$$\mu_t = \frac{\rho a_1 k}{max\left(a_1 \frac{A}{q^2}, \frac{M_{\infty}}{Re_{\infty}} \Omega F_2\right)} \tag{2.44}$$

3.2 Blending Functions and Intermediate Terms

$$F_2 = tanh(arg_2^2)$$
$$arg_2 = max(2\Gamma_1, \Gamma_2)$$
$$\Gamma_1 = \frac{M_{\infty}}{Re_{\infty}} \frac{\sqrt{k}}{\beta^* \omega d} = \frac{M_{\infty}}{Re_{\infty}} \frac{\sqrt{k}}{\beta^* (A/q^2) d} = \frac{M_{\infty}}{Re_{\infty}} \frac{\sqrt{k} q^2}{\beta^* A d}$$
$$\Gamma_2 = \left(\frac{M_{\infty}}{Re_{\infty}}\right)^2 \frac{500\nu}{d^2 \omega} = \left(\frac{M_{\infty}}{Re_{\infty}}\right)^2 \frac{500\nu q^2}{A d^2} \tag{2.45}$$
$$F_1 = tanh(arg_1^4)$$
$$arg_1 = min(\Gamma_3, \Gamma_4)$$
$$\Gamma_3 = max(\Gamma_1, \Gamma_2)$$
$$\Gamma_4 = \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega} d^2}$$

$$\frac{1}{\omega}\nabla\omega = \frac{q^2}{A}\left(-\frac{2A}{q^3}\nabla q\right) = -\frac{2}{q}\nabla q$$
$$CD_{k\omega} = max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\nabla k \cdot \nabla\omega, 10^{-20}\right)$$
$$= max\left(-\frac{4\rho\sigma_{\omega 2}}{q}\nabla k \cdot \nabla q, 10^{-20}\right) \tag{2.46}$$

$$
\begin{aligned}
&\sigma_k = F_1\sigma_{k1} + (1 - F_1)\sigma_{k2} \\
&\sigma_w = F_1\sigma_{w1} + (1 - F_1)\sigma_{w2} \\
&\beta = F_1\beta_1 + (1 - F_1)\beta_2 \\
&\gamma = F_1\gamma_1 + (1 - F_1)\gamma_2 \\
&a_1 = 0.31\beta^* = 0.09\ \kappa = 0.41 \\
&\gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}}\ \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}} \\
&\sigma_{k1} = 0.85\ \sigma_{k2} = 1.0\ \sigma_{\omega 1} = 0.5\ \sigma_{\omega 2} = 0.856\ \beta_1 = 0.075\ \beta_2 = 0.0828
\end{aligned}
\tag{2.47}
$$

3.3 Comments

It can be seen that given *k*-*q* transformation is purely mathematical and the new transport equation for *q* variable does not have physically meaningful signs for production, destruction and diffusion terms of the original ω equation. Therefore, a further post-processing step is needed to recover the original definition of ω in order to verify the physical behaviour of the turbulence field for each saved solution data. In addition, the new diffusion operator based on *q* results in a non-linear part that acts like a source term.

Furthermore, the positive constant ”*A*” is used as a parameter for having control on residual of *q* equation, thereby also on convergence characteristics of the CFD solution. It should be noted that the selected *A* constant changes the freestream boundary condition of ω_∞ as well, although the ω wall BC now attains a simple Dirichlet value of 0. Boundary conditions for *k* is not affected by the transformation. Finally, although the current transformation is more stable and well-behaved due to the new Dirichlet wall BC of 0 and is able to recover from negative values of *q* to positive more easily than *k*-ω version, the *q* values are still limited by the lower bound of $q_{min} = 10^{-10}$ for positivity requirements in the flux calculations for cases where q becomes negative.

4. NUMERICAL DISCRETIZATION

4.1 Temporal Discretization

Due to the inherent stiffness of high Reynolds number flow CFD simulations - which require very small time steps to accurately resolve sharp boundary-layer gradients near no-slip walls - explicit time-integration methods in the context of traditional finite-volume methods (FVM) are generally infeasible for industrial-scale aerodynamic applications even though they are favorable when it comes to code implementation (Blazek, 2001). This requires dealing with matrix-based implicit methods where large-scale, three-dimensional (3D), high Reynolds number problems can be tackled more effectively, since implicit methods have fewer timestep restrictions compared to explicit methods, therefore enabling faster overall convergence. Although they are shown to be computationally more intensive per timestep with increased memory requirements and are considered more challenging to program, the benefits owing to a proper linear matrix solver and effective preconditioner selection still outweigh the cons in the long run. Following section outlines the derivation of the steps that lead to recovering Newton's method in steady-state solutions of RANS equations within the dual timestepping framework using backward-Euler 1st order implicit scheme.

4.1.1 Deriving semi-discrete form

The general form for a system of conservation laws can be written in conservative differential form as shown in equation 2.48:

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{Q}) = 0 \tag{2.48}$$

where Q is the state/solution vector of conserved variables, t is the physical time, F vector can be decomposed into inviscid (F_i) and viscous flux (F_v) parts as $F = F_i - F_v$, and ∇ is the divergence operator. Here, conservative form guarantees that the numerical schemes based on this form conserves mass, momentum, and energy across discontinuities (Blazek, 2001). In the vertex-based finite volume method approach,

control volumes which are either called ”dual volumes” or ”median-dual cells” are constructed around the vertices of the primary mesh. Integrating the governing equation over a ”fixed” control volume Ω_i and applying the Gauss divergence theorem to the flux term in equation 2.49:

$$\frac{d}{dt}\int_{\Omega_i} \boldsymbol{Q} dV + \int_{\partial\Omega_i} \boldsymbol{F}(\boldsymbol{Q}) \cdot \boldsymbol{n} dS = 0 \tag{2.49}$$

Here, Ω_i is the volume of the i-th dual cell, $\partial\Omega_i$ is the boundary of the i-th dual cell, n is the outward-pointing unit normal vector to the surface element dS. This conservation equation states that the rate of change of the average Q in the control volume ”i” plus the net flux out of the control volume is zero. Then, the residual vector of the entire coupled equations for the node-based FVM system is given by $\boldsymbol{R}$ in equation 2.50 as:

$$\boldsymbol{R}(\boldsymbol{Q}) = \frac{d}{dt}\iiint_{\Omega} \boldsymbol{Q} dV + \iint_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{F}_i dS - \iint_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{F}_v dS = 0 \tag{2.50}$$

Defining the cell-averaged state as shown in equation 2.51 where M_i is the scalar volume (equivalent to a lumped mass matrix from the finite element literature) for vertex-based dual-volume (Blazek, 2001):

$$Q_i = \frac{1}{|\Omega_i|}\int_{\Omega_i} \boldsymbol{Q} dV \text{ where } M_i \equiv |\Omega_i| \tag{2.51}$$

Substituting gives the following ordinary differential form (in time) shown in equation 2.52 for each dual volume:

$$M_i \frac{dQ_i}{dt} + \boldsymbol{R}_i^{flux}(\boldsymbol{Q}) = 0 \tag{2.52}$$

Based on ”method-of-lines philosophy” described by Blazek (2001), applying spatial discretization to the residual for flux integrals $R_i^{flux}(Q)$ first, each control volume i now satisfies the following semi-discrete form in equation 2.53:

$$M_i \frac{d\boldsymbol{Q}_i}{dt} + \sum_{f\in\partial\Omega_i} (\boldsymbol{F}_f \cdot \boldsymbol{n}_f)\Delta S_f = 0 \tag{2.53}$$

where f denotes each face of the dual volume boundaries, ΔS_f is the face area, F_f is the numerical face flux (e.g. Roe flux) and n_f is the face normal. This system is called "semi-discrete" because space is discretized, but time is still continuous.

4.1.2 Backward-Euler implicit time integration

1st order backward-Euler implicit time integration scheme is a common choice for its stability properties which is beneficial for stiff problems often encountered in CFD (Blazek, 2001). While marching in physical time, $t^{n+1} = t^n + \Delta t$ and the backward Euler approximation for the time derivative is shown in equation 2.54:

$$\left.\frac{dQ_i}{dt}\right|_{t^{n+1}} \approx \frac{Q_i^{n+1} - Q_i^n}{\Delta t} \tag{2.54}$$

where Q_i^n is the current/known solution at time t^n, and Q^{n+1} is the unknown solution at t^{n+1}. Since the flux $R_i^{flux}(Q)$ is evaluated at t^{n+1} for an implicit scheme, the semi-discrete equation above becomes equation 2.55:

$$M_i \frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \boldsymbol{R}_i^{flux}(\boldsymbol{Q}^{n+1}) = 0 \tag{2.55}$$

Defining the physical residual of each dual volume at the $n+1$ time level as in equation 2.56:

$$\boldsymbol{R}_i(\boldsymbol{Q}^{n+1}) \equiv M_i \frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \boldsymbol{R}_i^{flux}(\boldsymbol{Q}^{n+1}) \tag{2.56}$$

The goal becomes finding a Q^{n+1} such that $R_i(Q^{n+1}) = 0$ for all dual volumes. It should be noted that $R_i^{flux}(Q^{n+1})$ is generally a non-linear function of Q^{n+1} because the physical fluxes $F(Q)$ are non-linear and the numerical fluxes F_f at the dual volume faces are also typically non-linear (Blazek, 2001). So, equation above represents a large system of coupled non-linear algebraic equations for Q^{n+1}.

To solve the non-linear system $R_i(Q^{n+1}) = 0$, an artificial time term, often called pseudo-time τ is introduced. The non-linear residual for Q^{n+1} is inserted into a new, artificial evolution equation in pseudo-time τ as shown in equation 2.57:

$$\boldsymbol{R}_i(\boldsymbol{Q}^{n+1}) \equiv M_i \frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \boldsymbol{R}_i^{flux}(\boldsymbol{Q}^{n+1}) \tag{2.57}$$

Here, Q^{n+1} is the variable which is trying to be solved for the next physical time step $n+1$. The new artifical time derivative is with respect to τ - not physical time t. This is called the ”dual-timestepping” or ”pseudo-transient continuation” (PTC). It should be noted that when the steady-state solution is reached in the ”pseudo-time”, any errors from the non–time-accurate continuation should disappear (Thompson & O'Connell, 2019). Substituting the definition of $R_i(Q^{n+1})$ from above, the following expanded form is obtained in equation 2.58:

$$M_i \frac{\partial Q_i^{n+1}}{\partial \tau} + M_i \frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \boldsymbol{R}_i^{flux}(\boldsymbol{Q}^{n+1}) = 0 \tag{2.58}$$

The idea is to ”evolve/march” this new form of the equation in pseudo-time τ until the pseudo-time derivative $\partial Q_i^{n+1}/\partial \tau$ becomes zero (within a specified tolerance level). Then, the remaining terms will satisfy $R_i(Q^{n+1}) \approx 0$, which is actually the solution to the new physical time step. However, for steady-state solutions - which is the case in this thesis - the physical time derivative vanishes, so the non-linear residual is just based on spatial fluxes shown in equation 2.59:

$$\boldsymbol{R}_i^{flux}(\boldsymbol{Q}) = 0 \tag{2.59}$$

where the pseudo-time equation turns into equation 2.60:

$$M_i \frac{\partial Q_i}{\partial \tau} + \boldsymbol{R}_i^{flux}(\boldsymbol{Q}) = 0 \tag{2.60}$$

For physical time levels, n and $n+1$ were used. For pseudo-iterations $k \rightarrow k+1$, using the 1st order backward-Euler scheme again for implicit pseudo-timestepping gives equation 2.61:

$$M_i \frac{\mathrm{Q}_i^{k+1} - \mathrm{Q}_i^k}{\Delta \tau} + \mathbf{R}_i^{\mathrm{flux}}(\mathbf{Q}^{k+1}) = \mathbf{0} \tag{2.61}$$

Defining $\Delta Q_i^k = Q_i^{k+1} - Q_i^k$ and linearizing the non-linear flux residual using a Taylor series expansion (keeping only first-order terms) results in equation 2.62:

$$\boldsymbol{R}_i^{flux}(\boldsymbol{Q}^{k+1}) \approx \boldsymbol{R}_i^{flux}(\boldsymbol{Q}^k) + \boldsymbol{J}_i^{(k)} \Delta Q_i^k \text{ where } \boldsymbol{J}_i^{(k)} = \left.\frac{\partial \boldsymbol{R}_i^{flux}}{\partial \boldsymbol{Q}}\right|_{\boldsymbol{Q}^k} \tag{2.62}$$

where $J_i^{(k)}$ is the flux Jacobian of residuals. Then, after inserting the linearized residual into pseudo-time equation, linear system at each pseudo-time becomes equation 2.63:

$$\left(\frac{M_i}{\Delta\tau}\mathbf{I} + \mathbf{J}_i^k\right)\Delta\mathrm{Q}_i^k = -\mathbf{R}_i^{\mathrm{flux}}(\mathbf{Q}^k) \tag{2.63}$$

For applying local pseudo-timestepping on each dual volume i, the most restrictive convective or viscous eigenwave speed is used with the pseudo-CFL number (Sukas & Sahin, 2025) as shown in equation 2.64:

$$\Delta\tau_i = min\left(\frac{V_i}{\sum_{k=1}^{N_f}\ \lambda_{c,k}S_k}, \frac{V_i}{2\sum_{k=1}^{N_f}\ \lambda_{v,k}S_k^2}\right)CFL \tag{2.64}$$

where V_i is the control-volume (area in 2-D), S_k is the area (edge length in 2-D) of face k, N_f is the number of faces of the dual volume, $\lambda_{c,k} = max_j\ |\lambda_j\ |$ is the convective eigenwave speed on face k, $\boldsymbol{\lambda_{v,k}} = \frac{\frac{\mu}{\rho V_i}\boldsymbol{max}\left(\frac{4}{3},\frac{\gamma}{Pr}\right)\boldsymbol{Ma_\infty}}{\boldsymbol{Re_\infty}}$ is the viscous eigenwave speed, and *CFL* is the user-specified pseudo-CFL number. Local pseudo-timestepping allows each control volume to behave independently, accelerating overall convergence on meshes with variable cell sizes (Blazek, 2001).

4.1.3 Newton method for steady-state solutions

By ramping-up the pseudo-CFL number and driving the pseudo-timestep to infinity $(\Delta\tau \rightarrow \infty)$ during the steady-state solutions, the PTC approach recovers the classical Newton's root finding method in equation 2.65 for solving the non-linear system $R_i(Q^{k+1}) = 0$:

$$\boldsymbol{J}_i^k \Delta Q_i^k = -\boldsymbol{R}_i^{flux}(\boldsymbol{Q}^k) \tag{2.65}$$

Although Newton's method shows rapid quadratic convergence near the solution, it also requires an accurate initial guess to avoid divergence (Blazek, 2001). This is why the $M_i/\Delta\tau$ term in the PTC formulation can be seen as a "globalization" term for the Newton method (Thompson & O'Connell, 2019). This helps stabilize the iterations by damping unphysical transients - especially during the start-up phase of the solution thanks to the finite $\Delta\tau$ values. In addition, if the CFL number -therefore the pseudotime - is not high enough, the Newton iterations can slow down as the solution approaches

steady state. In HEMLAB, a consistent Newton method is implemented where the time derivative term is included in the flux residual on the right-hand side as well as shown in equation 2.66 (Thompson & O'Connell, 2019; Sukas & Sahin, 2025):

$$\left(\frac{M_i}{\Delta\tau}\boldsymbol{I} + \boldsymbol{J}_i^k\right)\Delta Q_i^k = -\boldsymbol{R}_i^{flux}(\boldsymbol{Q}^k)$$
$$= -\left[\frac{\boldsymbol{Q}^m - \boldsymbol{Q}^n}{\Delta\tau}M_i + \sum_{f\in\partial\Omega_i}\left(\boldsymbol{F_i}(\boldsymbol{Q^k})\cdot\boldsymbol{n}_f\right)\Delta S_f \right. \tag{2.66}$$
$$\left. - \sum_{f\in\partial\Omega_i}\left(\boldsymbol{F_v}(\boldsymbol{Q^k})\cdot\boldsymbol{n}_f\right)\Delta S_f\right]$$

where *m* is the m[th] Newton sub-iteration. One key advantage of this method is that the residual drops by several orders of magnitude in just a few Newton sub-iterations. Furthermore, the damping provided by the time-derivative term on the RHS allows the use of larger initial CFL numbers without causing instability (Sukas & Sahin, 2025).

4.1.4 Non-linear Newton solver

The solution can pass through highly non-linear regions where even PTC approach may not help with convergence. In such cases, a line-search can be combined with the the Newton update step to guarantee a reduction in the non-linear residual (Thompson & O'Connell, 2019; Sukas & Sahin, 2025). After computing the Newton update Δu, a scalar $\eta \in (0,1]$ is searched as in equation 2.67:

$$\|f(\boldsymbol{u})\|_2 > \|f(\boldsymbol{u} + \eta\Delta\boldsymbol{u})\|_2 \tag{2.67}$$

where $f(\boldsymbol{u})$ is the nonlinear residual function and $\boldsymbol{u}$ is the current solution. The line-search tries to find the largest η that satisfies the equation above, to guarantee that each Newton update actually decreases the non-linear residual norm (Thompson & O'Connell, 2019; Sukas & Sahin, 2025). PETSc-SNES (Scalable Nonlinear Equations Solvers) framework is utilized in HEMLAB in order to implement the non-linear Newton method with a line search technique to enhance convergence stability in the iterative process. PETSc (Portable, Extensible Toolkit for Scientific Computation) is a parallel software library for solving scientific problems modeled by partial differential equations (PDEs) (Balay et al., 2018). It is ideal for large CFD runs

because it is scalable and handles complex models and big datasets thanks to the MPI and GPU support.

4.2 Spatial Discretization

For inviscid flux reconstruction within vertex-based finite volume method framework in HEMLAB, a blending of upwinding with central approach is utilized (Akkurt & Sahin, 2022). At the cell interfaces, inviscid fluxes are calculated using the flux-difference splitting method (Roe, 1982).

Several other inviscid flux schemes such as HLLC, AUSM+up and AUSM+M are also available. Left and right state vectors are obtained via an unweighted least-squares approach.

In addition, gradients of the primitive variables in viscous flux calculations are obtained at edge midpoints with the Green–Gauss theorem. Currently, for the convective terms of k-ω SST model, only a 1st order upwinding is implemented. In addition, although model equations are written in compressible form, k and ω are stored/solved in the state vector, not the ρk or $\rho\omega$.

Following sections briefly introduce the important concepts related to application of the vertex-based finite volume method, gradient schemes, upwinding and flux schemes in HEMLAB algorithm. Final part shows how the spatial terms in the k-q SST model are handled numerically.

4.2.1 Vertex-based finite volume method

Goal of the finite volume method is to obtain numerical inviscid/viscous face fluxes F_f in the semi-discrete form of coupled system in equation 2.68:

$$M_i \frac{dQ_i}{dt} + \sum_{f \in \partial\Omega_i} \left(\boldsymbol{F}_f \cdot \boldsymbol{n}_f\right) \Delta S_f = 0 \tag{2.68}$$

HEMLAB uses median-dual control volumes Ω_i around each mesh vertex V_i, obtained by connecting the edge midpoints with centroids of all elements sharing that mesh

vertex. As an example, the resulting polyhedral median dual-volume that surrounds the vertex *V* is shown in Figure 4.1 for a primal triangular background mesh:

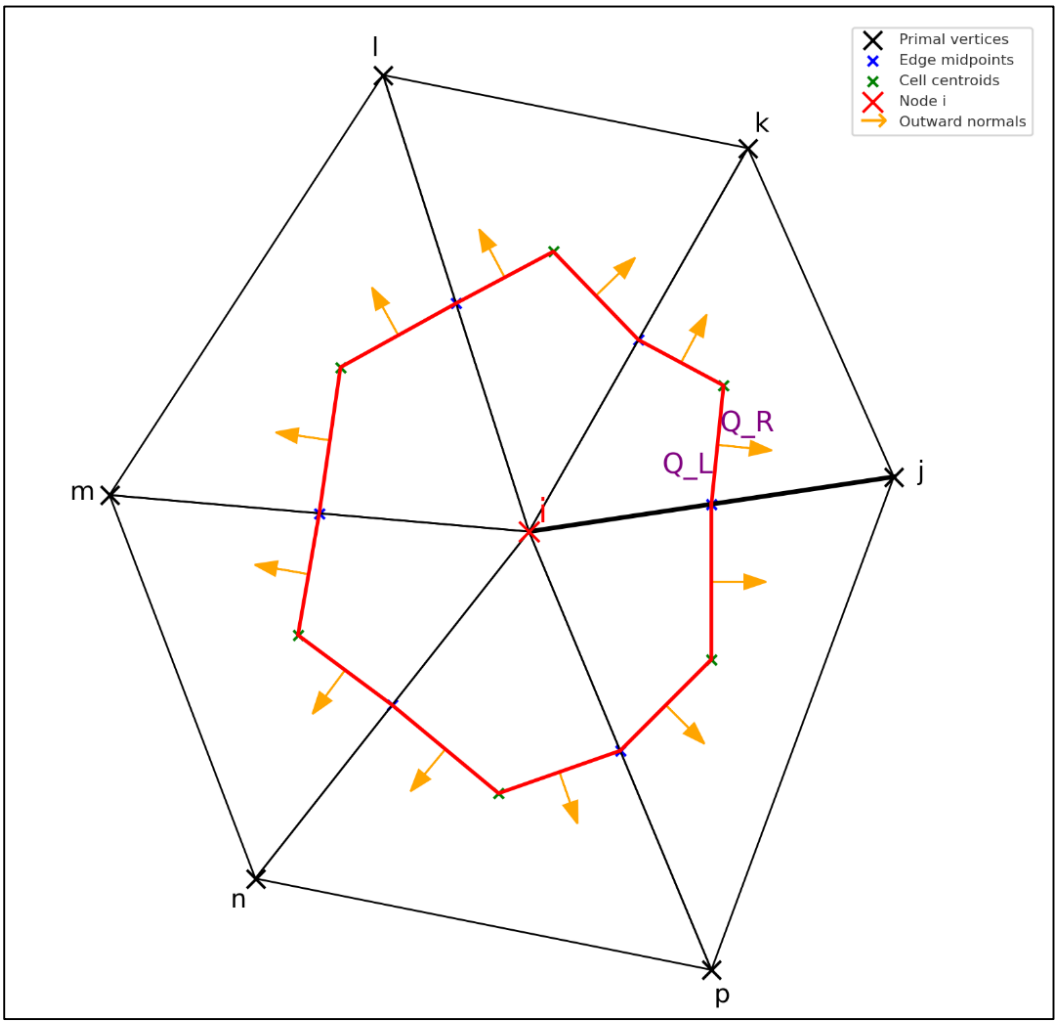

Figure 4.1 : Dual-volume constructed around the primal mesh vertex i.

In contrast, it would be called ”centroid-dual” if the edge midpoints were to be directly joined (Blazek, 2001). For non-moving grids, all the geometric constructs and dual-volume related data only need to be calculated once before the solution iterations.

4.2.2 Gradient evaluation

For second-order accuracy in inviscid fluxes, nodal gradients of the state vector *Q* variables are calculated with the unweighted upwind least-squares interpolation (Sukas

& Sahin, 2025). Taylor expansion around the node i with n outgoing-edges towards its closest neighbours at $\{x_k\}$ gives the over-constrained linear system in equation 2.69:

$$\begin{bmatrix} \boldsymbol{x_1 - x_i} \\ \boldsymbol{x_2 - x_i} \\ \vdots \\ \boldsymbol{x_n - x_i} \end{bmatrix} \begin{bmatrix} \dfrac{\partial Q}{\partial x} \\ \dfrac{\partial Q}{\partial y} \\ \dfrac{\partial Q}{\partial z} \end{bmatrix} = \boldsymbol{Ax} = \boldsymbol{b} = \begin{bmatrix} \boldsymbol{Q_1 - Q_i} \\ \boldsymbol{Q_2 - Q_i} \\ \vdots \\ \boldsymbol{Q_n - Q_i} \end{bmatrix} \tag{2.69}$$

where Q_i and Q_k are state vector variables at corresponding nodes. Although it is possible to solve above $\boldsymbol{Ax} = \boldsymbol{b}$ system for the nodal gradients ∇Q_i with normal equations approach as shown in equation 2.70 (Blazek, 2001):

$$A^T Ax = A^T b \implies x = (A^T A)^{-1} A^T \tag{2.70}$$

the analytical least-squares solution based on Gram-Schmidt orthogonalization of matrix A is used in HEMLAB as shown in equation 2.71 (Sukas & Sahin, 2025):

$$\begin{aligned} A = QR \qquad & Q^T Q = I \\ Rx = Q^T b \implies \quad & x = R^{-1} Q^T b \end{aligned} \tag{2.71}$$

where Q is an orthogonal matrix (not to be confused with state vector Q), R is an upper-triangular matrix, I is identity matrix. This prevents ill-conditioning issues with matrix inversion in cases of highly stretched grids encountered during anisotropic refinement. In addition, computed gradients are exact for linear functions. Resulting gradient vector components can be written as a ”weighted-sum” of edge lengths in equations 2.72 (Blazek, 2001):

$$\begin{aligned} \left.\frac{\partial Q}{\partial x}\right|_i &= \sum_{k=1}^{N_i} W_{ik}^x (Q_k - Q_i), \\ \left.\frac{\partial Q}{\partial y}\right|_i &= \sum_{k=1}^{N_i} W_{ik}^y (Q_k - Q_i), \\ \left.\frac{\partial Q}{\partial z}\right|_i &= \sum_{k=1}^{N_i} W_{ik}^z (Q_k - Q_i) \end{aligned} \tag{2.72}$$

Although a slope limiter ϕ propsoed by Wang (2000) can be applied in HEMLAB to the nodal gradients in order to prevent non-physical oscillations, it was not used for the numerical applications in this thesis. Especially, the min/max functions in the limiter can degrade the coupled system convergence (Sukas & Sahin, 2025). Finally, the least-square gradients do not cause inaccuracies since they are only included in inviscid fluxes (Sukas & Sahin, 2025).

For viscous fluxes, gradients of the primitive variables $W = [\rho, u, v, w, T]^T$ are evaluated using the Green–Gauss theorem at the midpoint of the median-dual volume edges. For this, a new control volume V_d is constructed around each edge of the median-dual volume, using element centers and face midpoints connected to that edge (Sukas & Sahin, 2025) As an example, gradient of *x*-velocity components are shown in equations 2.73:

$$\frac{\partial u}{\partial x} = \frac{\int_{\partial V_d} u n_x dS}{\int_{V_d} dV}, \quad \frac{\partial u}{\partial y} = \frac{\int_{\partial V_d} u n_y dS}{\int_{V_d} dV}, \quad \frac{\partial u}{\partial z} = \frac{\int_{\partial V_d} u n_z dS}{\int_{V_d} dV} \tag{2.73}$$

where dS is the area of each face of the new control volume, n_x is the face normals, u is the *x*-velocity component at those faces and $\int_{V_d} \boldsymbol{dV}$ is the new control volume size. Surface integrals at new control volume faces are evaluated with the mid-point rule (Sukas & Sahin, 2025).

4.2.3 Flux reconstruction

Using the general slope-limited form of nodal gradients from the primal mesh vertices, the left and right state vectors at the midpoint of the each edge connecting the median dual-volume vertices are shown in equation 2.74:

$$Q_L = Q_i + \phi_i \sum_{k=1}^{N_i} W_{ik}(Q_k - Q_i)$$
$$Q_R = Q_j + \phi_j \sum_{k=1}^{N_j} W_{jk}(Q_k - Q_j) \tag{2.74}$$

where weighting coefficients are fiven in equation 2.75:

$$W_{ik} = \frac{W_{ik}^x(x_j - x_i) + W_{ik}^y(y_j - y_i) + W_{ik}^z(z_j - z_i)}{2},$$
$$W_{jk} = \frac{W_{jk}^x(x_j - x_i) + W_{jk}^y(y_j - y_i) + W_{jk}^z(z_j - z_i)}{2} \tag{2.75}$$

It should be noted that for median-dual approach, dual-volume faces/edges do not necessarily intersect the primal mesh as shown in Figure 4.1 in contrast to centroid-dual approach, except for orthogonal meshes where element centers align with face/edge midpoints. Here, each dual-volume edge that is connected to the primal mesh edge midpoint use separate reconstructed states for flux calculations. In addition, for controlling numerical diffusion, upwind and central contributions are blended at the state interfaces in HEMLAB as shown in equation 2.76 (Sukas & Sahin, 2025):

$$\hat{Q}_L = \beta Q_L + (1-\beta)\frac{Q_i + Q_j}{2}$$
$$\hat{Q}_R = \beta Q_R + (1-\beta)\frac{Q_i + Q_j}{2} \tag{2.76}$$

where $\beta = 1$ with full upwinding was used in this thesis. However, only 1st order upwind for k and ω equations were implemented. As a summary, based on Figure 4.1, the list of spatial numerical discretization operations that lead to inviscid flux calculation for the vertex-based FVM with median-dual control volumes are as follows: (i) for each primal-mesh vertex V_i, assign and store the state vector Q, (ii) compute the nodal gradient ∇Q_V via the unweighted least-squares interpolation, (iii) construct the median-dual volume around primal mesh vertex V_i, (iv) compute each median-dual volume face's normal n_f, centroid x_f and area ΔS_f, (v) using the least-squares-based nodal gradients at each V_i (with optional slope limiter) and the nodal state vector Q, apply an upwind interpolation to left/right states of the median-dual volume's each edge midpoint x_f , (vi) evaluate the numerical flux scheme (e.g. Roe

FDS) at the median-dual volume face where $F_f = F(Q_L, Q_R, n_f, \Delta S_f)$, and finally (vii) add F_f on median-dual volume face as "positive" contribution into the flux residual sum of node i and subtract from node j.

4.2.4 Roe's flux difference splitting method

Inviscid fluxes are computed at cell interfaces using approximate Riemann solver of Roe (1982), based on the idea of splitting the flux difference on the face of the control volume into a sum of wave contributions Blazek (2001). If the face-normal unit vector is shown with n_f , Roe-averaged state by $\tilde{Q}$, and Roe-matrix with $\tilde{A} = \partial F/\partial \tilde{Q}$, the numerical flux F_f at the faces of the median-dual volume is given as in equation 2.77:

$$\boldsymbol{F_f} = \frac{1}{2}\left[F(\hat{Q}_L) + F(\hat{Q}_R)\right] \cdot \boldsymbol{n}_f - \frac{1}{2}|\tilde{A}|(\hat{Q}_R - \hat{Q}_L) \tag{2.77}$$

In order to deal with unphysical expansion shocks in stationary expansions and carbuncle phenomenon in bow shock cases due to original scheme not being able to detect sonic points in the flow (Blazek, 2001), Harten's (1983) entropy fix/correction is applied in HEMLAB which modifies eigenvalues (λ) below the threshold $h = \epsilon\, max_m |\lambda_m|$ as shown in equation 2.78:

$$|\lambda| = \begin{cases} \dfrac{\lambda^2 + h^2}{2h}, & |\lambda| < h \\ |\lambda|, & \text{otherwise} \end{cases} \tag{2.78}$$

4.2.5 Numerical representation of k-q SST model

Tables 4.1 & 4.2 below summarize the finite volume based numerical counterparts of analytical terms in both non-dimensional transformed model equations where ΔV is the median-dual cell volume, F_f^{mass} is the mass flux ($\boldsymbol{\rho_f\, u_f \cdot n_f}$) on face, ∂V is the median-dual boundary faces and V denotes the nodal value of given variable.

Table 4.1 : Finite-volume discretisation of the non-dimensional k-equation for HEMLAB $k-q$ SST model using median-dual volume for vertex V.

Equation term	Analytical form	FVM discrete form
Temporal term	$\frac{\partial(\rho k)}{\partial t}$	$\left(\frac{(\rho k)_V^{n+1}-(\rho k)_V^n}{\Delta t}\right)\Delta V_V$
Convective term	$\nabla\cdot(\rho \boldsymbol{u} k)$	$\sum_{f\in\partial V} F_f^{mass}\, k_f$
Diffusion term	$\frac{M_\infty}{Re_\infty}\nabla\cdot[(\mu+\sigma_k\mu_t)\nabla k]$	$\frac{M_\infty}{Re_\infty}\sum_{f\in\partial V}(\mu+\sigma_k\mu_t)_f(\nabla k)_f\cdot \boldsymbol{n}_f A_f$
Production	$min\left(\frac{M_\infty}{Re_\infty}\mu_{tV}S_V^2, \frac{Re_\infty}{Ma_\infty}20\beta^*\rho\frac{A}{q^2}k\right)$	$min\left(\frac{M_\infty}{Re_\infty}\mu_{tV}S_V^2, \frac{Re_\infty}{Ma_\infty}20\beta^*\rho_V\frac{A}{q_V^2}k_V\right)\Delta V_V$
Destruction	$-\frac{Re_\infty}{Ma_\infty}\beta^*\rho\frac{A}{q^2}k$	$\left(-\frac{Re_\infty}{Ma_\infty}\beta^*\rho_V\frac{A}{q_V^2}k_V\right)\Delta V_V$

Table 4.2 : Finite-volume discretisation of the non-dimensional q-equation for HEMLAB $k-q$ SST model using a median-dual volume for vertex V.

Equation term	Analytical form	FVM discrete form
Temporal term	$\frac{\partial(\rho q)}{\partial t}$	$\left(\frac{(\rho q)_V^{n+1}-(\rho q)_V^n}{\Delta t}\right)\Delta V_V$
Convective term	$\nabla\cdot(\rho \boldsymbol{u} q)$	$\sum_{f\in\partial V} F_f^{mass}\, q_f$
Diffusion term	$\frac{M_\infty}{Re_\infty}\nabla\cdot[(\mu+\sigma_\omega\mu_t)\nabla q]$	$\frac{M_\infty}{Re_\infty}\sum_{f\in\partial V}(\mu+\sigma_\omega\mu_t)_f(\nabla q)_f\cdot \boldsymbol{n}_f A_f$
Production	$-\frac{q^3}{2A}\frac{M_\infty}{Re_\infty}\gamma\rho S^2$	$\left(-\frac{q_V^3}{2A}\frac{M_\infty}{Re_\infty}\gamma\rho_V S_V^2\right)\Delta V_V$
Destruction	$\frac{Re_\infty}{Ma_\infty}\beta\rho\frac{A}{2q}$	$\left(\frac{Re_\infty}{Ma_\infty}\beta\rho_V\frac{A}{2q_V}\right)\Delta V_V$
Non-linear part of diffusion term	$-\frac{3}{q}\frac{M_\infty}{Re_\infty}\Gamma\lvert\nabla q\rvert^2$	$\left(-\frac{3}{q_V}\frac{M_\infty}{Re_\infty}\Gamma_V\lvert\nabla q_V\rvert^2\right)\Delta V_V$
Cross-diffusion term	$\frac{M_\infty}{Re_\infty}\frac{2}{A}q^2(1-F_1)\rho\sigma_{\omega 2}\nabla k\cdot\nabla q$	$\frac{M_\infty}{Re_\infty}\left[\frac{2}{A}q_V^2(1-F_{1,V})\rho_V\sigma_{\omega 2}(\nabla k_V\cdot\nabla q_V)\right]\Delta V_V$

5. COUPLING OF SST TURBULENCE MODEL EQUATIONS

5.1 Segregated and Coupled Implementations

High CFL numbers are of extreme importance in Newton-type methods for reaching quadratic convergence, where highly accurate Jacobian evaluations are needed. Automatic differentiation based source code transformation library Tapenade is integrated with HEMLAB for exact Jacobian calculations of inviscid fluxes (Inria (n.d.)). For turbulence equations' source term contributions to Jacobians, analytical derivatives of each production and destruction term in *k-ω* SST model is obtained by-hand. Therefore, fully-coupling of Navier-Stokes system is achieved in pursuit of higher CFL numbers. It should be noted that this is not the preferred method in most CFD solvers due to implementation difficulty in implicit approach and computational cost considerations associated with Jacobian matrix assignments where usu ally approximate factorization-type methods are utilized instead of exact Jacobians (Content et al., 2013). Finally, most *k-ω* SST model implementations in the literature uses a loosely-coupled method of coupling the Navier-Stokes equations together, while solving the turbulence equations in segregated manner (Langer & Swanson, 2024). In addition to the numerical advantages, a fully-coupled treatment is also physically meaningful (Suarez, 2020). In the standard, dimensional form of *k-ω* SST model the eddy (turbulent) viscosity and vorticity terms are defined as in equation 2.79:

$$\mu_t = \frac{\rho a_1 k}{max(a_1\omega, \Omega F_2)}, \qquad \Omega = \sqrt{2\Omega_{ij}\Omega_{ij}}, \qquad \Omega_{ij} = \frac{1}{2}\left(\partial_j u_i - \partial_i u_j\right) \tag{2.79}$$

The production term in the *k*-equation is $P_k = \mu_t S^2$, so any perturbation in the mean velocity field that changes Ω_{ij} changes the denominator of μ_t, therefore P_k and the turbulence content in the flowfield. μ_t also appears in the viscous stress tensor as part of the "effective viscosity" definition in equation 2.80:

$$\tau_{ij} = 2(\mu + \mu_t)S_{ij} - \frac{2}{3}\rho k \delta_{ij} \tag{2.80}$$

Therefore, a ”disturbance” in the turbulence variables immediately affects the momentum and energy equations as well (Suarez, 2020). A fully-coupled Newton method includes this natural physical coupling inside each nonlinear iteration, while a segregated approach lags the velocity field behind (several pseudo-iterations are required) which causes the maximum stable CFL number to be limited.

5.2 Flux Jacobian

In a Newton-type iterative solution, the nodal state vector variables and the flux residual work together like a closed-loop cycle until convergence. First, the current state vector Q is used for flux reconstruction on every dual face, which then gets combined into the flux residual $R_V(Q)$ around the vertex V where residual is the local imbalance (how far each vertex is from satisfying conservation laws) of the mass, momentum and energy conservation (Blazek, 2001). For the fully-implicit, density-based formulation, the semi-discrete form in steady-state is based on the spatial FVM flux residual for the dual-volume around a primal mesh vertex V as shown in equation 2.81:

$$\boldsymbol{R_v}(\boldsymbol{Q}) = \sum_{f \in \partial V} \left[\boldsymbol{F}_f^{inv}(\boldsymbol{Q}) - \boldsymbol{F}_f^{vis}(\boldsymbol{Q}, \nabla \boldsymbol{Q})\right] \cdot \boldsymbol{n_f} \Delta S_f = 0 \tag{2.81}$$

where the solution/state vector $Q = [\rho, \rho u, \rho v, \rho w, E]^T$ is stored at every vertex V of the primal mesh, F_{inv} and F_{vis} show the inviscid and viscous fluxes respectively which are calculated at the dual face f, n_f is the dual-face outward normal vector and $\Delta S_f = /\!/ \vec{n}_f /\!/$ is the area of the each dual face of the median-dual control volume. The nonlinear flux residual R_V on each dual-face includes contributions to the 2 neighbour vertices (e.g. i and j from Figure 4.1), so the Newton linearization $\boldsymbol{J}\Delta\boldsymbol{Q} = -\boldsymbol{R}$ requires that a global Jacobian matrix involving the effects of the 2 connecting nodes per each primal mesh edge is constructed as in equation 2.82:

$$\boldsymbol{J} = \frac{\boldsymbol{\partial R}}{\boldsymbol{\partial Q}} \tag{2.82}$$

The Jacobian matrix describes how a very small change in any state vector variable would affect the flux residual. Solving the linear system $\boldsymbol{J}\,\Delta\boldsymbol{Q} = -\boldsymbol{R}$ results in a Newton update/correction that - if the Jacobian is accurate - tries to cancel the current flux imbalance. This correction updates the nodal state vectors, which in turn changes the fluxes and produces a new, usually smaller residual which closes the loop. Iterations continue until the residual (imbalance) is within a desired tolerance (hopefully around machine tolerance levels) where the influence between *Q* and *R* is at an ”equilibrium” and the solution can be considered converged. Commonly used Jacobian construction methods can be grouped into four classes:

Exact analytic Jacobian: For a Newton-type implicit solver, rapid (quadratic) convergence during the Newton correction is achieved only when Jacobian matrices are assembled using exact/analytical derivatives. However, this has increased memory requirements and larger matrix build time (Blazek, 2001), since every flux’s derivative needs to be obtained with respect to the state vector variables stored at the vertices that influence the dual face.

Approximate analytic Jacobian: If approximate Jacobians are utilized (e.g. by keeping only the most important physical couplings via diagonal contributions), the rapid Newton convergence is lost in exchange for a cheaper solution because many cross-derivative (off-diagonal) terms are omitted (Blazek, 2001).

Numerical Jacobian: Each state vector variable is ”perturbed” with a small finite-difference step which requires re-evaluation of residuals for every vertex. Although it is easy to implement, assembling and storing the resulting matrix is very expensive in large-scale 3-D problems (Blazek, 2001).

Jacobian-free Newton–Krylov (JFNK): Alternatively, JFNK methods skip building the Jacobian matrix to lower computational cost (Knoll & Keyes, 2004). Instead, they rely on using a finite difference representation of flux residuals to perform matrix–vector products; however these methods often require stronger preconditioning, and selecting the step-size “ϵ” is a challenging aspect (Blazek, 2001). A strong preconditioner is required for robustness. Equation 2.83 that describes numerical Jacobian construction and equation 2.84 for the JFNK method look similar as shown below:

$$J_{ij} \approx \frac{\mathcal{R}_i\left(\boldsymbol{Q} + h\boldsymbol{e}_j\right) - \mathcal{R}_i(\boldsymbol{Q})}{h} \tag{2.83}$$

$$[\boldsymbol{J}(\boldsymbol{Q})]\boldsymbol{v} \approx \frac{\mathcal{R}(\boldsymbol{Q} + \varepsilon\boldsymbol{v}) - \mathcal{R}(\boldsymbol{Q})}{\varepsilon} \tag{2.84}$$

Here R is the flux residual vector, Q the current state vector, h and ε small step sizes for finite difference representations, e_j is the jth unit coordinate vector, and v is an arbitrary Krylov search direction. In HEMLAB, first and second-order Jacobian matrices are evaluated with a mixed strategy which is made of hand–coded analytic derivatives for most terms and automatic differentiation (AD) for the inviscid fluxes (Sukas & Sahin, 2025). Inviscid Jacobians based on the right and left-state vectors are evaluated exactly using INRIA Tapenade source-code transformation library (Inria, (n.d.)), and then scaled by the least–squares coefficients to obtain the second-order Jacobian. Analytical derivatives are used for turbulence source terms' contributions. Although derivations are time consuming and error-prone, it is desirable for computational efficiency. For code verification, HEMLAB also allows using a Jacobian-free finite-difference evaluation of the matrix–vector product with PETSc; comparing this to the hand-coded Jacobian is helpful to avoid code implementation errors (Sukas & Sahin, 2025).

5.3 Inviscid Jacobian

For every dual-volume face, inviscid flux residual contribution is differentiated with respect to its left and right states, resulting in two 5 × 5 local inviscid Jacobian blocks in equation 2.85:

$$J_L = \frac{\partial\left(\boldsymbol{n} \cdot \boldsymbol{F}_i(\boldsymbol{Q}_L, \boldsymbol{Q}_R)\right)}{\partial \boldsymbol{Q}_L} \quad \text{and} \quad J_R = \frac{\partial\left(\boldsymbol{n} \cdot \boldsymbol{F}_i(\boldsymbol{Q}_L, \boldsymbol{Q}_R)\right)}{\partial \boldsymbol{Q}_R} \tag{2.85}$$

where n is the unit face normal and F_i is the numerical face flux (Roe scheme with Harten correction in this study); slope limiter contributions are left out in the inviscid Jacobian because they were not used and assembly cost is reduced. Exact evaluations for J_L and J_R are generated with the Tapenade source-code transformation library. The local 5 × 5 blocks J_L and J_R are then assembled in the second-order Jacobian matrix with the scalar weights W_{ij} obtained during the least–squares reconstruction step with a stencil made up of N_i vertices. The least-squares weights also play a role since Jacobian by definition must account for all possible influences on the flux residual. For an interior primal mesh edge connecting the vertex i to its neighbour j, second-

order inviscid Jacobian assembly is carried out as follows (Sukas & Sahin, 2025) in equation 2.86:

$$\begin{bmatrix} J_L\boldsymbol{\Phi}_i & J_R\boldsymbol{\Phi}_j \\ -J_L\boldsymbol{\Phi}_i & -J_R\boldsymbol{\Phi}_j \end{bmatrix}\begin{bmatrix} \Delta\boldsymbol{Q}_i \\ \Delta\boldsymbol{Q}_j \end{bmatrix}, \boldsymbol{\Phi}_i = diag\left(\phi_i W_{i1}, \dots, 1-\phi_i \sum_{k=1}^{N_i} W_{ik}\right) \quad (2.86)$$
$$\Delta\boldsymbol{Q}_i = \left[\Delta\boldsymbol{Q}_{i1}, \Delta\boldsymbol{Q}_{i2}, \dots, \Delta\boldsymbol{Q}_{iN_i}\right]^T$$

where Φ_i and Φ_j includes all least–squares weights for vertex i and j, respectively. The second index 1, 2, . . . , N_i in ΔQ_i shows which vertex in the stencil the update belongs to. This block matrix is the only contribution of edge (i, j) to the global Jacobian where the top row updates the residual at vertex i and the bottom row is for the opposite contribution at vertex j.

5.4 Viscous Jacobian

Unlike the inviscid Jacobian which connects only a single vertex pair through a median-dual volume's face, viscous Jacobian involves the full edge stencil where every node { i, j, N_1, . . . , N_N } of the elements connected to the same primal edge has a contribution to the momentum and energy equations; this results in a wider Jacobian matrix (Sukas & Sahin, 2025). Since the viscous stresses and heat fluxes depend on the temperature–dependent properties (i.e. $\mu(T)$ and $\kappa(T)$) (Blazek, 2001), the viscous Jacobian obtained from the Green–Gauss-based fluxes are evaluated at the edge midpoints with the primitive variable vector $W = [\rho, u, v, w, T]^T$ as well. The resulting primitive Jacobian matrix is then converted back to Jacobian using the conservative state vector variables with the following transformation (Sukas & Sahin, 2025) in equation 2.87:

$$\boldsymbol{J}_Q = \frac{\partial \mathcal{R}}{\partial \boldsymbol{Q}} = \frac{\partial \mathcal{R}}{\partial \boldsymbol{W}}\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{Q}} \quad (2.87)$$

This results in the "exact" local viscous Jacobian in equation 2.88:

$$\begin{bmatrix} J_i & J_j & J_{N_1} & \cdots & J_{N_N} \\ -J_i & -J_j & -J_{N_1} & \cdots & -J_{N_N} \end{bmatrix}\begin{bmatrix} \Delta\boldsymbol{Q}_i \\ \Delta\boldsymbol{Q}_j \\ \Delta\boldsymbol{Q}_{N_1} \\ \vdots \\ \Delta\boldsymbol{Q}_{N_N} \end{bmatrix} = -\begin{bmatrix} R_i \\ -R_j \end{bmatrix} \quad (2.88)$$

Although this exact evaluation increases memory consumption and the cost of matrix–vector products, it enables the Newton method to achieve very high CFL numbers. Finally, because the boundaries in the present study use no-slip adiabatic conditions at walls, the first-order Green–Gauss approximation on boundaries is excluded from the viscous Jacobian and the assembled matrix still has second-order accuracy in the interior (Sukas & Sahin, 2025).

5.5 Contributions of Turbulence Model Source Terms

Just like the SA-neg-noft2 turbulence model implementation in HEMLAB (Sukas & Sahin, 2025), the *k–ω* SST source terms enter the Newton method through exact analytic derivatives that are evaluated at the edge midpoints rather than at the nodes. Each source term is integrated over half of the dual control volume (for smoother production/destruction terms); and Jacobian terms are assembled in the global matrix in the same way as the viscous Jacobian (Sukas & Sahin, 2025). Because these derivatives involve more independent variables than the inviscid fluxes, they were coded by hand rather than using the Tapenade source-code transformation library (which is preferred for the inviscid Jacobian). Although it requires large manual programming effort, the fully analytic turbulence model Jacobian contributions allow significantly high maximum CFL numbers to be achieved during pseudo-transient continuation with the Newton method (Sukas & Sahin, 2025). Detailed source term derivatives that go into the exact Jacobian for the *k*-equation are shown step-by-step in the Appendix-A.

Approximate methods are widely used shortcuts in implicit one or two-equations RANS model implementations is to linearize only the negative (destruction) part of each source term and move it to the left–hand side, while the positive (production) part is kept as explicit (Merci et al., 2000). For the *k-ω* SST model, this "source-term splitting" results in the following Newton update shown in equations 2.89 and 2.90:

$$\left(1 + \Delta\tau \frac{\partial D_k}{\partial k}\right) \Delta k = -(R_k + P_k - D_k) \tag{2.89}$$

$$\left(1 + \Delta\tau \frac{\partial D_\omega}{\partial \omega}\right) \Delta \omega = -(R_\omega + P_\omega - D_\omega) \tag{2.90}$$

Here $\partial D_k/\partial k$ and $\partial D_\omega/\partial \omega$ are the negative Jacobian contributions of the destruction terms in D_k for k and D_ω for ω equations, respectively. Here, R on the right hand side is the explicit inviscid and viscous flux residual for each equation and P denotes the production term. This approximation makes the diagonal contributions always positive and increases diagonal dominance and stability while avoiding assembling the full, exact turbulence Jacobian with cross derivatives Merci et al. (2000); however this ignores physical coupling between variables and results in slower convergence for Newton method approach and therefore not preferred in HEMLAB.

6. ADAPTIVE MESH REFINEMENT

In CFD, the accuracy of results is often limited more by the quality of the mesh, than by the solver itself. The NASA CFD Vision 2030 study considered mesh generation as the ”primary bottleneck” that prevents high-fidelity simulations of around complex geometries from becoming a routine practice for an average CFD engineer (Park et al., 2016; Slotnick et al., 2014). Adaptive Mesh Refinement (AMR) attempts to solve this issue by optimizing the mesh during the solution process automatically, increasing cell density where local error indicators predict large discretization error while coarsening the cells in other areas (Mavriplis, 1995; Roy, 2010).

A generic AMR cycle consists of (i) selecting a refinement sensor, (ii) tagging/filtering elements or edges for refinement, (iii) modifying the mesh (point insertion/removal, local remeshing, smoothing methods), and (iv) interpolating the solution and continuing the simulation until a stopping criterion is reached (Powell & Roe, 1993; Baker, 1997).

Almost every AMR strategy works with the “error equidistribution principle” where it looks for obtaining a mesh on which a chosen error indicator is nearly uniform. For isotropic refinement the target is uniform elemental error; whereas for anisotropic algorithms the goal becomes uniform ”edge-based” error (Habashi et al., 2000).

6.1 Isotropic and Anisotropic Refinement

Because many turbulent compressible-flow features (shocks, boundary layers, vortices) are highly directional phenomena, uniformly refining all three spatial directions (isotropic refinement) is a waste of resources in both memory and CPU time, especially in 3-D simulations.

Anisotropic AMR methods aim to stretch or collapse elements along flow-aligned directions defined by local solution gradients or Hessians (Alauzet & Loseille, 2016).

For any type of refinement method, choosing the refinement sensor is of utmost importance. Widely used sensors are listed as follows:

Feature-based sensors: Classical criteria look at first and second-derivative jumps in solution variables (e.g. $|\nabla p|$, $\nabla^2 \rho$) to detect shocks, shear layers and vortices as well as directly at the solution variables themselves such as Mach number or vorticity magnitude (Löhner, 1995).

Prior knowledge of the expected flow physics guides the choice of variables—for instance, pressure gradients for shocks,vorticity magnitude for vortical structures. Experience shows that feature detectors are most effective when the initial mesh already resolves the predicted location of the target phenomena; otherwise the algorithm may refine entirely wrong regions and stall the convergence (Mavriplis, 1995; Warren, 1991).

Metric-based sensors: Hessian-driven metrics aim to equidistribute an interpolation error estimate. A possible unwanted isotropic refinement near discontinuities can be mitigated using multi-scale metrics (Habashi et al., 2000; Loseille, 2014).

Goal-oriented (output-based) sensors: Requires an adjoint solution. Here, refinement is based on integral quantities of engineering interest such as lift, drag and total pressure instead of physical flow features/solution variables (Fidkowski & Darmofal, 2011; Alauzet & Frazza, 2021).

6.2 Implementation Challenges in AMR

Initial mesh dependence: A coarse starting grid can misidentify flow features and lead the algorithm to refine in the wrong place and waste CPU time (Warren, 1991).

Sensor sensitivity: Discontinuities in one variable may overlap smooth regions for another variable where an inappropriate sensor selection can trigger ”refine–and-coarsen” cycles (Powell & Roe, 1993).

Uncontrolled growth: Without size constraints AMR may lead to generation of excessive number of nodes. Therefore, minimum edge lengths or error-tolerance

bounds need to be imposed, or adaptation may be stopped after a target grid density is reached (Baker, 1997).

Element quality: Repeating refinement around shocks can generate low quality triangles which can disrupt solver stability such that smoothing or remeshing is often required (Habashi et al., 2000).

6.3 Available AMR Tools

6.3.1 NASA-refine

”refine” is an open-source anisotropic grid adaptation library which was developed by NASA (2025). Its main capabilities are: mesh adaptation mechanics can handle 2D/3D simplex (triangle and tetrahedra) meshes; mesh adaptation metrics can be computed by using gradients and Hessians from a solution field; solutions can be interpolated between meshes; interfaced with HEMLAB (Sukas & Sahin, 2025).

6.3.2 INRIA-pyAMG

”pyAMG” is an anisotropic mesh adaptation library developed by the GAMMA team at Inria Lab (Loseille, 2014). Its main capabilities are: unstructured anisotropic volume and surface remeshing through local modifications; metric computation from a solution field using a multiscale metric; solution field interpolation from one mesh onto another; interfaced with CFD solvers such as SU2 (Economon et al., 2016) and HEMLAB (Akkurt & Sahin, 2022).

In this thesis, pyAMG was used for the adaptive solution of high-lift multi-element airfoil case by NASA Langley Research Center (n.d.) to test k-ω SST model implementation and assess its convergence in case of highly-streched, anisotropic triangular grid elements.

The mesh that was originally generated in the Euclidean space is re-interpreted by the pyAMG anisotropic adaptation algorithm in a ”Riemannian metric space”, with a symmetric, positive-definite 3x3 metric tensor $M(x)$. Metric field is generated from the Hessian of a scalar sensor such as Mach, pressure or entropy. Hence, by redefining “length” concept through $M(x)$, any mesher that can produce a uniform grid can be re-interpreted as an anisotropic mesher. Within this new metric space an “ideal” mesh

has ”unit” edge lengths (ℓ_M) for each element (Alauzet & Loseille, 2016). Therefore, an unstructured grid is iteratively transformed as shown in equation 2.91 until:

$$\ell_M(\overrightarrow{ab}) = \int_0^1 \sqrt{\overrightarrow{ab}^\top \boldsymbol{M}(\boldsymbol{a} + t\overrightarrow{ab})\overrightarrow{ab}}\, dt = 1 \tag{2.91}$$

for every straight edge *ab*. However, because the metric tensor *M(x)* stretches or compresses lengths differently in different directions, a uniform triangle in metric space becomes an anisotropic element in physical Euclidean space.

In addition, adaptation procedure tries to minimize the global spatial interpolation error in the L_p norm ($p = 4$ is most accurate for compressible RANS with Mach sensor (Park et al., 2016)), while applying the error equidistribution principle (Alauzet & Frazza, 2021). Transfer/interpolation of solution data between adaptive iterations and grids is carried out by a Python interface similar to SU2 (Economon et al., 2016), while the mesh is stored in Gamma (.mesh) format to preserve sharp edges and corners during successive adaptations (Alauzet & Frazza, 2021).

In HEMLAB, initial mesh adaptation sub-iterations use an exponential, wall-distance based sensor shown in equation 2.92 that builds resolution near wall boundaries even if no boundary-layer mesh is present:

$$Sensor(d) = \sum_{i=1}^{4} 0.25e^{-100x10^i d} \tag{2.92}$$

where *d* denotes the minimum wall distance. Once a rough boundary layer has formed, refinement is continued with a pure Mach-number sensor until the end. Non-dimensional entropy, pressure and combined entropy–Mach sensors are also available in HEMLAB (Sukas & Sahin, 2025).

7. TEST CASES FOR MODEL VERIFICATION

7.1 Zero Pressure Gradient Flat Plate

7.1.1 Case conditions

A validation study was conducted using the zero-pressure-gradient flat plate benchmark by NASA Langley Research Center (n.d.) at $Re_L = 5 \times 10^6$ and $M_\infty = 0.2$, focusing on the behavior of turbulent quantities (k and ω) in the boundary layer to verify the fully-coupled k-ω SST model implementation. Table 7.1 below provides a summary for the physical case conditions below.

Table 7.1 : Physical conditions for the flat plate validation case.

Physical Condition	Value
Reynolds number (Re_L)	5×10^6
Freestream Mach number (M_∞)	0.2
Angle of attack (α)	0°
Angle of sideslip (β)	0°
Freestream non-dim. turbulent kinetic energy (k_∞)	9×10^{-9}
Freestream non-dim. specific dissipation rate (ω_∞)	1×10^{-6}
Reference non-dim. dynamic viscosity (μ)	1.0
Reference non-dim. static pressure (p_i)	0.7143
Reference non-dim. static temperature (T_i)	1.0
Turbulent Prandtl number (Pr_t)	0.9
Specific heat ratio (γ)	1.4

7.1.2 Mesh

Structured meshes provided by NASA were used, with even the coarsest mesh ensuring $y+ \approx 1.7$ NASA Langley Research Center (n.d.). No mesh adaptation was applied for this case. Table 7.2 below provides a summary for the grid parameters. Figure 7.1 below shows the finest mesh available.

Table 7.2 : Mesh parameters for the flat plate validation case.

Mesh Parameter	Value
Grid type	Structured
Element type	Quadrilateral
Fixed/Adaptive Grid	Fixed
Domain extent in x	[−0.33, 2.0]
Domain extent in y	[0, 1.0]
Reference length for Re_L (L_{ref})	1.0
Reference area (A_{ref})	2.0
Finest mesh resolution	545×385
Coarsest mesh resolution	35x25

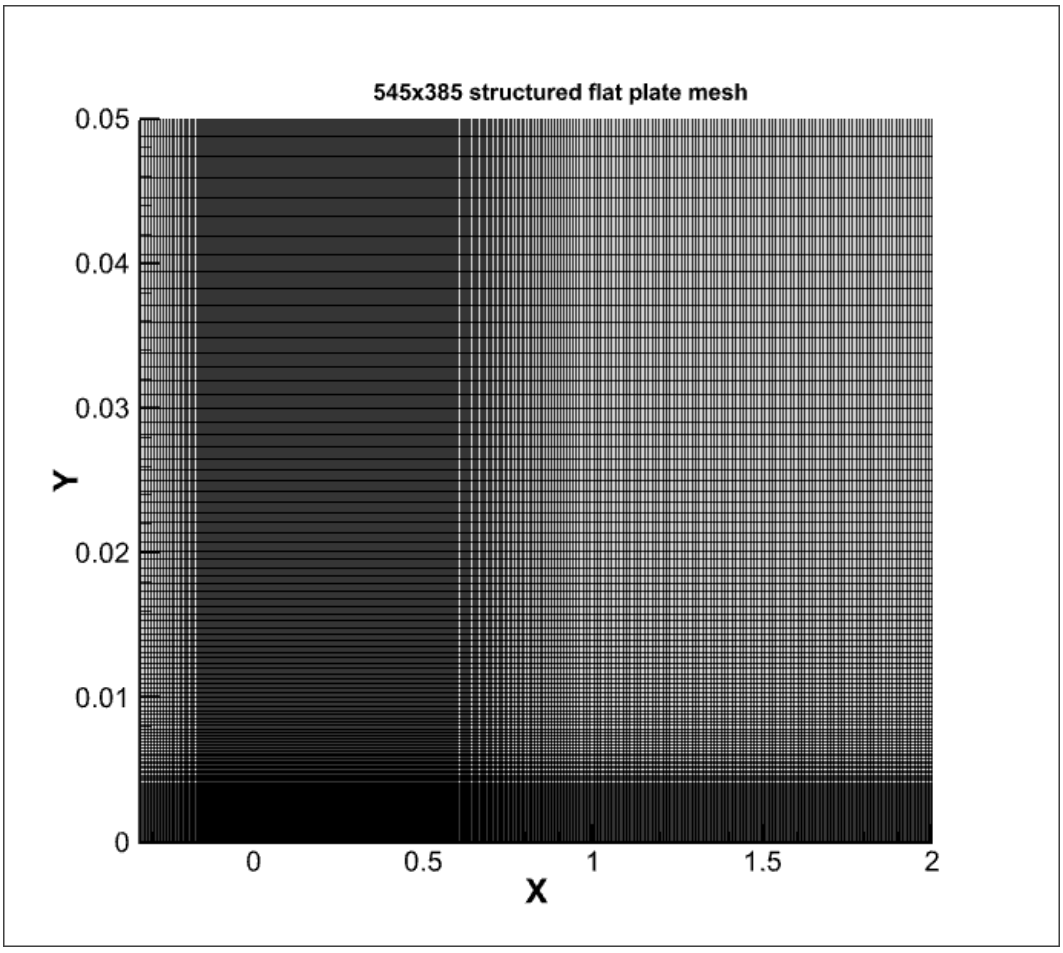

Figure 7.1 : Structured finest level 545 × 385 flat plate mesh (NASA Langley Research Center (n.d.)).

7.1.3 Numerics

Table 7.3 below provides a summary for the numerical solution parameters for flat plate case with subsonic inlet-outlet, symmetry, no-slip wall and farfield boundaries.

Table 7.3: Numerical parameters for the flat plate validation case.

Numerical Parameter	Value
BC for $x \in [-0.33, 0]$ at $y = 0$	Slip wall
BC for $x \in [0, 2]$ at $y = 0$	No-slip, adiabatic wall
BC for $x \in [-0.33, 2]$ at $y = 1$	Farfield
BC for $y \in [0, 1]$ at $x = -0.33$	Subsonic inlet
BC for $y \in [0, 1]$ at $x = 2$	Subsonic outlet
Time-dependence	Steady-state
Pseudo-time CFL strategy	Adaptive
CFL parameters (Start, Max, Up rate, Down rate)	[0.1, 10000, 1.125, 0.5]
Gradient scheme	Unweighted least-squares
Spatial order of accuracy for N-S equations	2^{nd} order
Spatial order of accuracy for turbulence equations	1^{st} order
Spatial upwind blending parameter (β_{upwind})	1.0
Flux scheme	Roe
Flux limiter	Off
ω scaling parameter for k-q SST model (A_0)	10^{-10}

7.1.4 Results

All following results are obtained without mesh adaptation, using the finest 545×385 structured grid provided by NASA Langley Research Center (n.d.). Figures 7.2 to 7.5 show the non-dimensional turbulent kinetic energy (k) and specific turbulent dissipation rate (ω) contours for HEMLAB and FUN3D solvers in boundary-layer where $y = [0, 0.05]$ from inlet to outlet.

Similarly, Figures 7.6 and 7.7 compare the nodal HEMLAB and FUN3D k and ω profiles in the boundary layer where a sectional cut is taken at x=0.97 m on the flat plate wall. In addition, skin-friction profiles along the flat plate are compared between HEMLAB, FUN3D solvers and theoretical expressions from NASA Langley Research Center (n.d.) on Figure 7.8 while Table 7.4 shows the drag coefficient comparison

between different solvers and turbulence models. Finally, Figures 7.9 and 7.10 display the convergence behaviour of HEMLAB SST implementation for the flat plate case.

For the non-dimensional k and ω contours shown in Figures 7.2 to 7.5, excellent agreement is observed in the boundary layer flow between HEMLAB k-q SST-m and FUN3D SST-Vm results.

Non-dimensional k and ω profiles for the section at x=0.97 on Figures 7.6 and 7.7 show that ”local” turbulence behaviour is in very good agreement between HEMLAB and FUN3D as well. The sudden switch near y=0.01 is the result of blending function activation near the boundary layer edge. k levels are at its highest near the middle of boundary layer where usually the eddy-viscosity is also highest (ANSYS, Inc., 2020).

In addition, quadratic near-wall behaviour of ω is captured accurately when compared to analytical value in Figure 7.7 as well. Although ω goes to infinity at the wall, it could not be shown due to post-processing limitations. When drag coefficient values are inspected from Table 7.4, it is clear that the integral quantity is predicted consistently for each solver and HEMLAB SST implementation output is very close to SA-neg value with differences being about half drag count. The skin friction plot also matches FUN3D results almost perfectly while both solvers follow the theoretical behaviour range closely. Finally, it can be seen from the Figures 7.9 and 7.10 that the rapid convergence for HEMLAB SST implementation is thanks to exact Jacobian construction in Newton method where L_2-norm of the non- linear residual easily falls to 1×10^{-9} levels and pseudo-CFL number can go up to 10000.

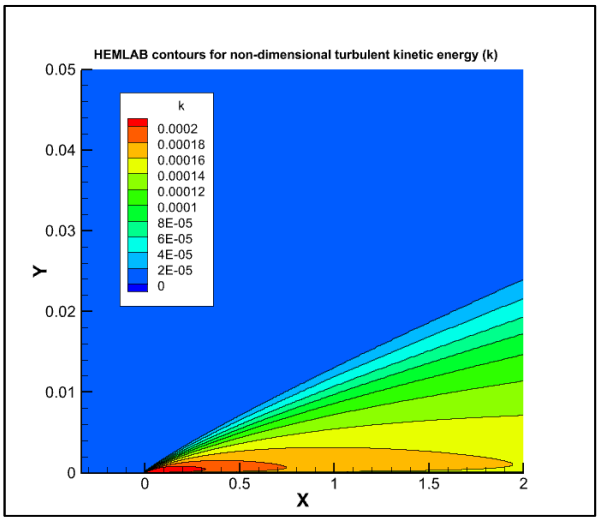

Figure 7.2 : *k*-contours along the boundary layer on the flat plate and profile at x=0.97 m for HEMLAB k-q SST-m model for 545 × 385 grid.

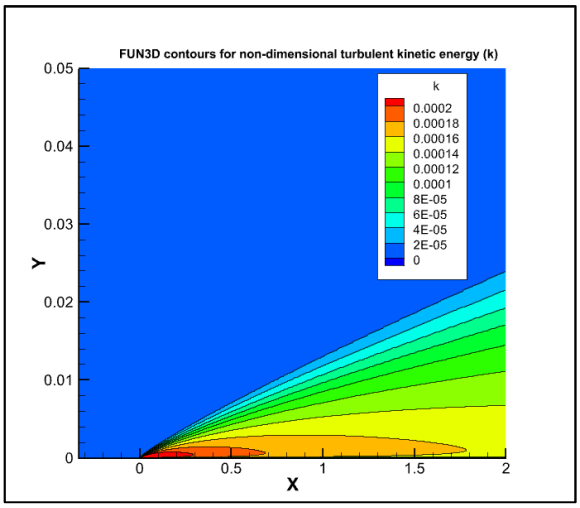

Figure 7.3 : *k*-contours along the boundary layer on the flat plate and profile at x=0.97 m for FUN3D SST-Vm model for 545 × 385 grid.

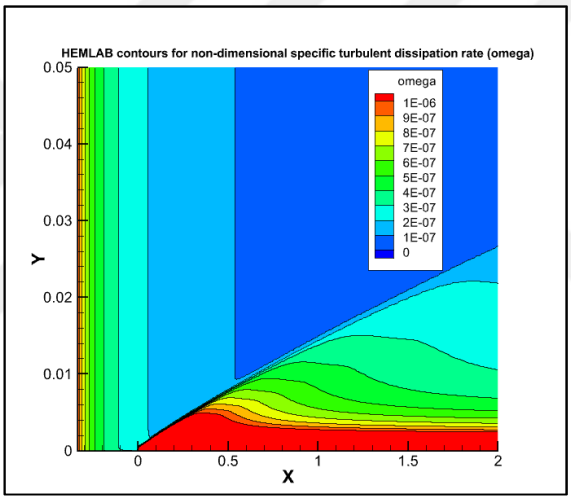

Figure 7.4 : ω-contours along the boundary layer on the flat plate and profile at x=0.97 m for HEMLAB for 545 × 385 grid.

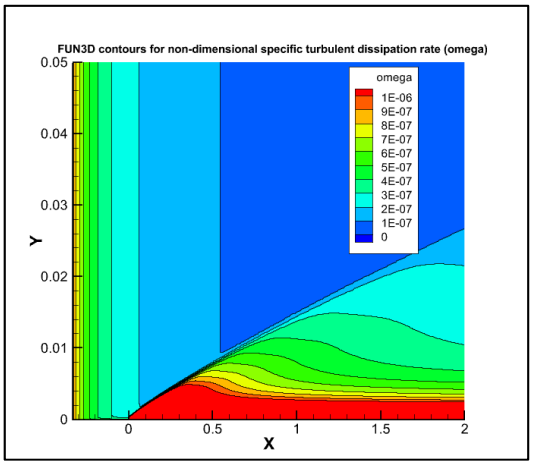

Figure 7.5 : ω-contours along the boundary layer on the flat plate and profile at x=0.97 m for FUN3D for 545 × 385 grid.

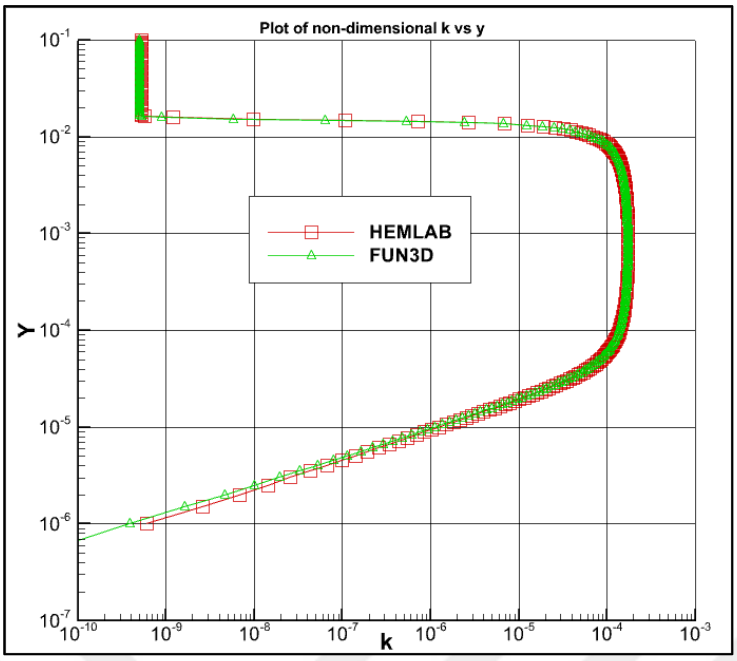

Figure 7.6 : Comparison of *k*-profiles at x=0.97 m for HEMLAB and FUN3D for 545 × 385 grid.

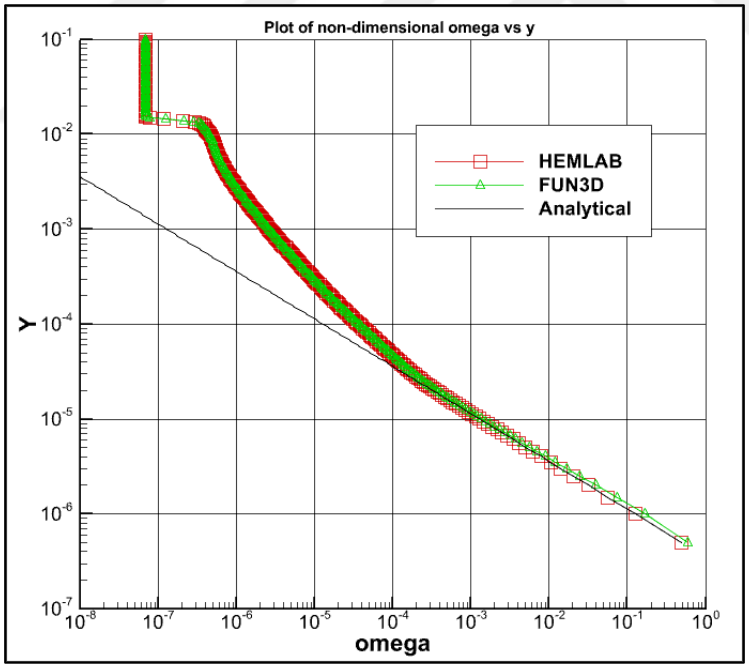

Figure 7.7 : Comparison of ω profiles at x=0.97 m for HEMLAB and FUN3D vs analytical solution at the wall for 545 × 385 grid.

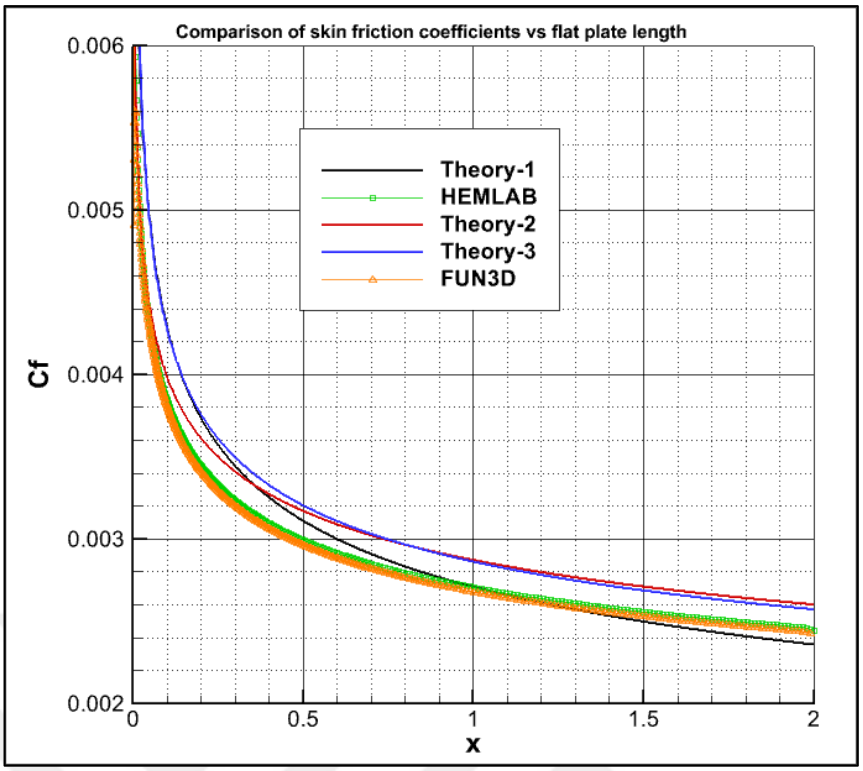

Figure 7.8 : Skin friction variation along the flat plate for HEMLAB, FUN3D and theory for 545 × 385 grid.

Table 7.4 : Comparison of drag coefficient C_d for different solvers and turbulence models for 545 × 385 grid.

Solver	Turbulence Model	C_d
FUN3D	SA-neg	0.00285
FUN3D	*k-ω* SST-Vm	0.00283
HEMLAB	SA-neg	0.00288
HEMLAB	*k-ω* SST-m	0.00289

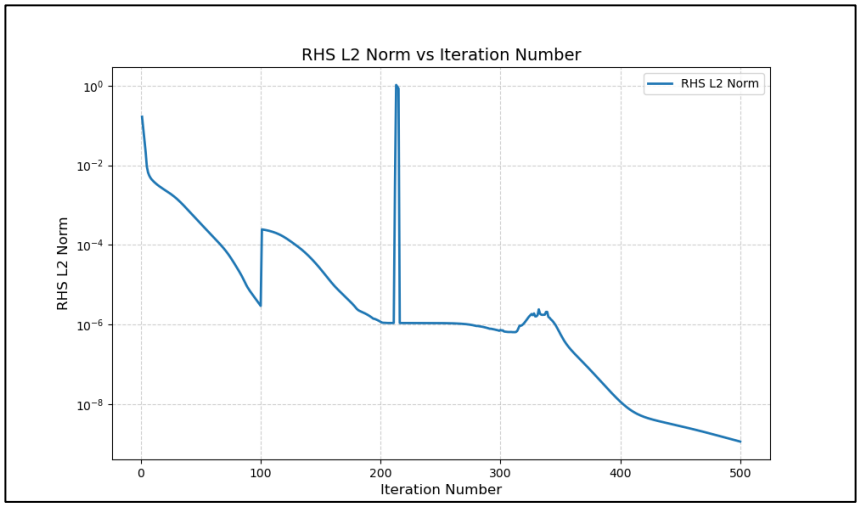

Figure 7.9 : L_2-norm convergence plot for HEMLAB SST model using 545 × 385 flat plate grid.

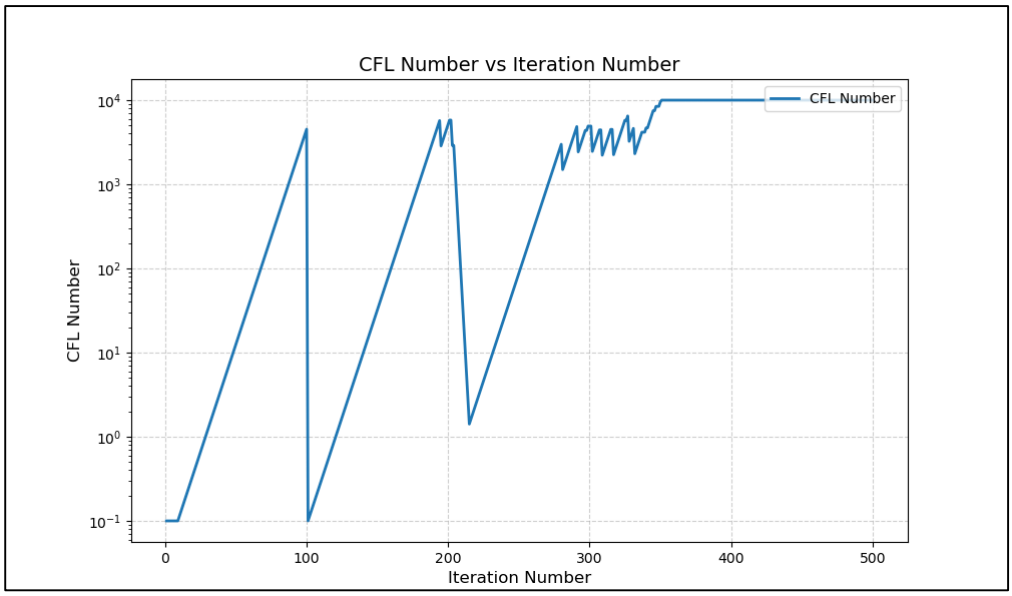

Figure 7.10 : Variation of CFL number vs iteration number for HEMLAB SST model using 545 × 385 flat plate grid.

7.2 4th High Lift Prediction Workshop CRM–HL Multi-Element Airfoil

7.2.1 Case conditions

For the three-element 2D CRM–HL multi-element airfoil, a moderately high-lift condition at $\alpha = 16^{\circ}$, $M_\infty = 0.20$ and $Re_c = 5 \times 10^6$ where c is the reference chord was investigated to assess the fully-coupled *k-ω* SST model performance on the canonical SA verification case from HLPW4 (NASA Langley Research Center, (n.d.)). Table 7.5 below provides a summary for the physical case conditions.

Table 7.5 : Physical conditions for the HLPW4 multi-element airfoil case.

Physical Condition	Value
Reynolds number (Re_c)	5×10^6
Freestream Mach number (M_∞)	0.2
Angle of attack (α)	16°
Angle of sideslip (β)	0°
Freestream non-dim. turbulent kinetic energy (k_∞)	9×10^{-9}
Freestream non-dim. specific dissipation rate (ω_∞)	1×10^{-6}
Reference non-dim. dynamic viscosity (μ)	1.0
Reference non-dim. static pressure (p_i)	0.7143
Reference non-dim. static temperature (T_i)	1.0
Turbulent Prandtl number (Pr_t)	0.9
Specific heat ratio (γ)	1.4

7.2.2 Mesh

Starting from a coarse, purely triangular mesh, feature-based anisotropic adaptive mesh refinement cycles were executed in HEMLAB using pyAMG library where a Mach-based sensor was utilized (Akkurt & Sahin, 2022). Farfield boundary was placed at 1000 chord lengths (NASA Langley Research Center, (n.d.)).

The first three adaptive sub-iterations are for building the boundary layer region without any solution, while the next three sub-iterations used a first-order spatial discretization to build a stable flowfield during refinement runs.

Finally, the rest of adaptive grid sub-iterations used second-order upwinding in flux calculations. Final grid at 9th adaptive subiteration has 515599 total nodes. Table 7.6 below provides a summary for the grid parameters.

Table 7.6: Mesh parameters for the HLPW4 multi-element airfoil case.

Mesh Parameter	Value
Grid type	Unstructured
Element type	Triangular
Fixed/Adaptive Grid	Adaptive
Domain extent in x	1000 × c
Reference chord length for Re_c (c)	1.0
Reference area (A_{ref})	1.0
Initial coarse mesh node count	80517
Finest adaptive mesh node count	515599

Figures 7.11 to Figure 7.18 describe the initial geometry, coarse mesh and final adapted meshes. Original unmeshed geometry for the three-element airfoil is provided where slat, main element and flap sections are tagged on Figure 7.11 below:

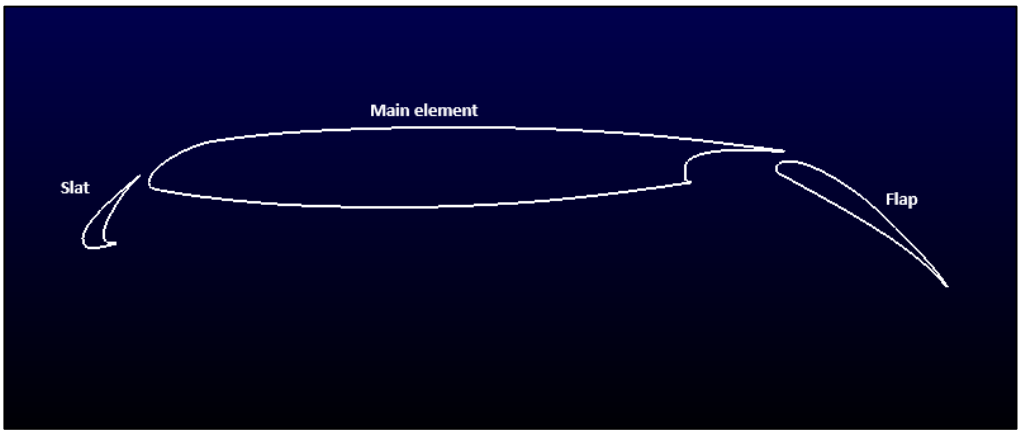

Figure 7.11 : CRM-HL multi-element airfoil geometry denoting slat, main element and flap regions.

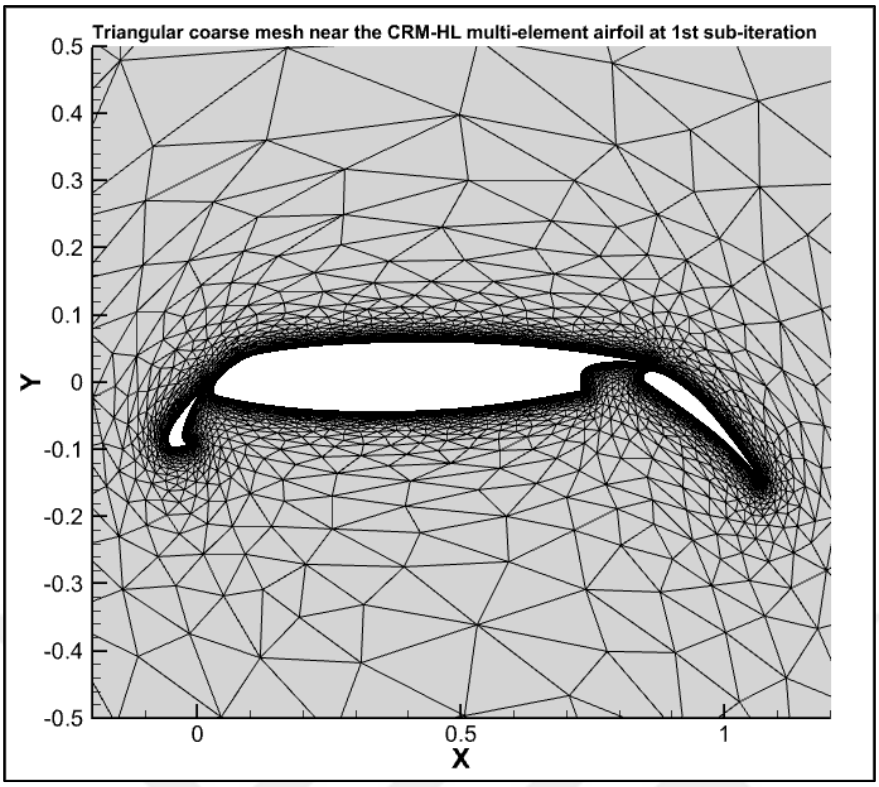

Figure 7.12 : Initial coarse triangular unstructured near-wall mesh including all three elements from 1st sub-iteration for 2D CRM-HL airfoil case.

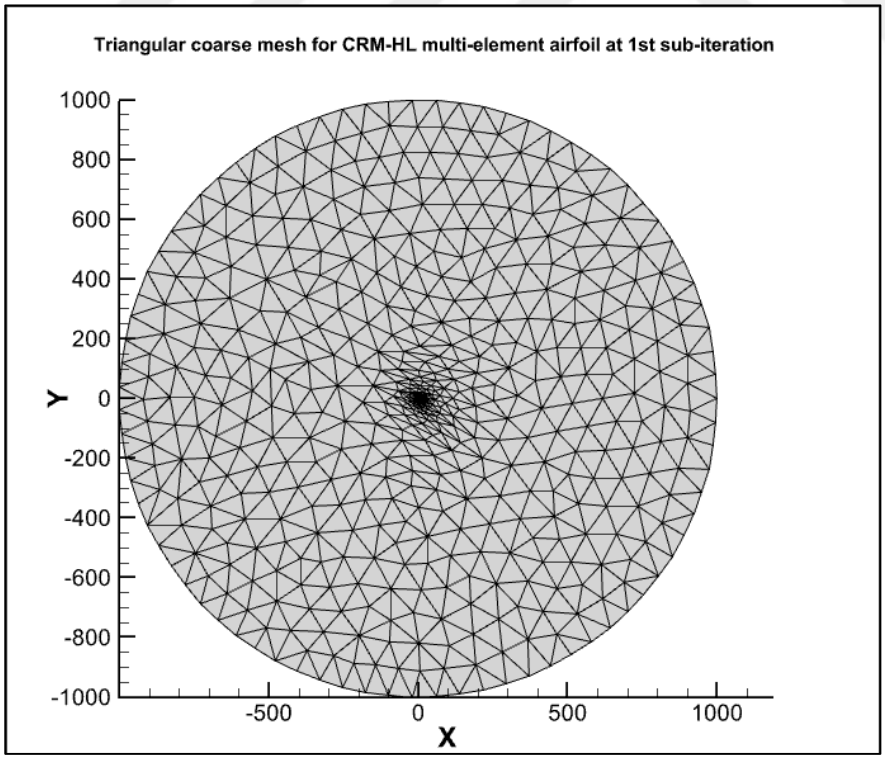

Figure 7.13 : Initial coarse triangular unstructured domain mesh from 1st sub-iteration for 2D CRM-HL airfoil case.

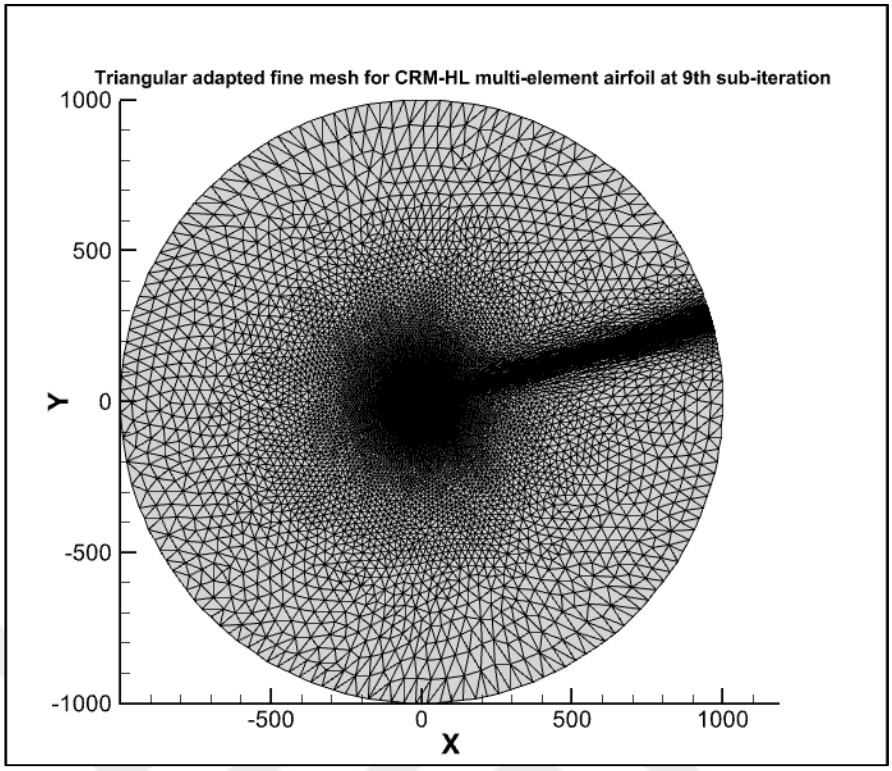

Figure 7.14 : Adapted unstructured mesh from 9th sub-iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16°.

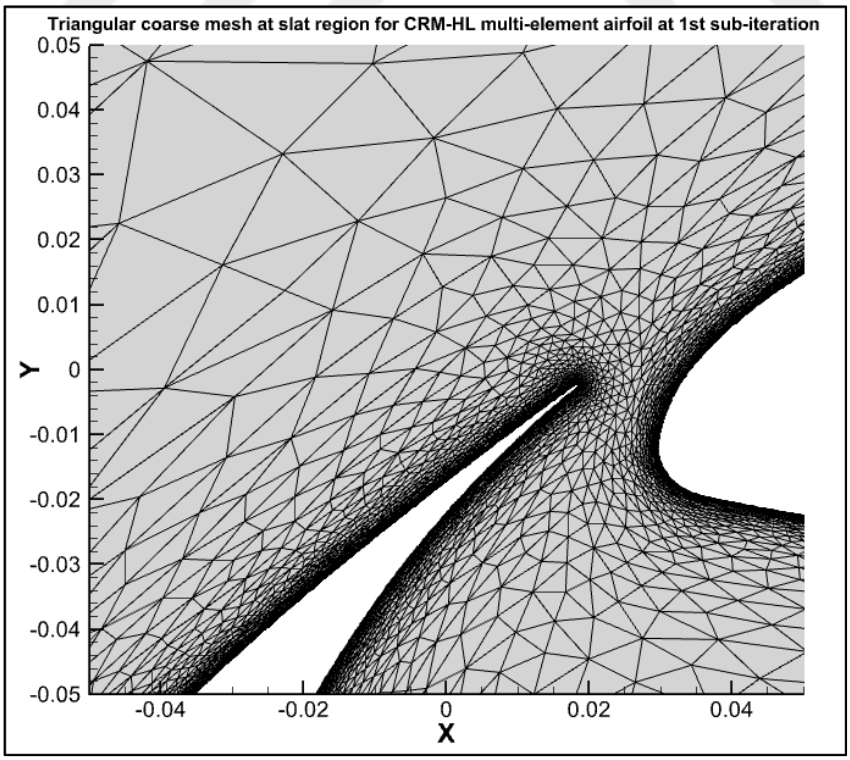

Figure 7.15 : Initial coarse triangular unstructured slat region mesh from 1st sub-iteration for 2D CRM-HL airfoil case.

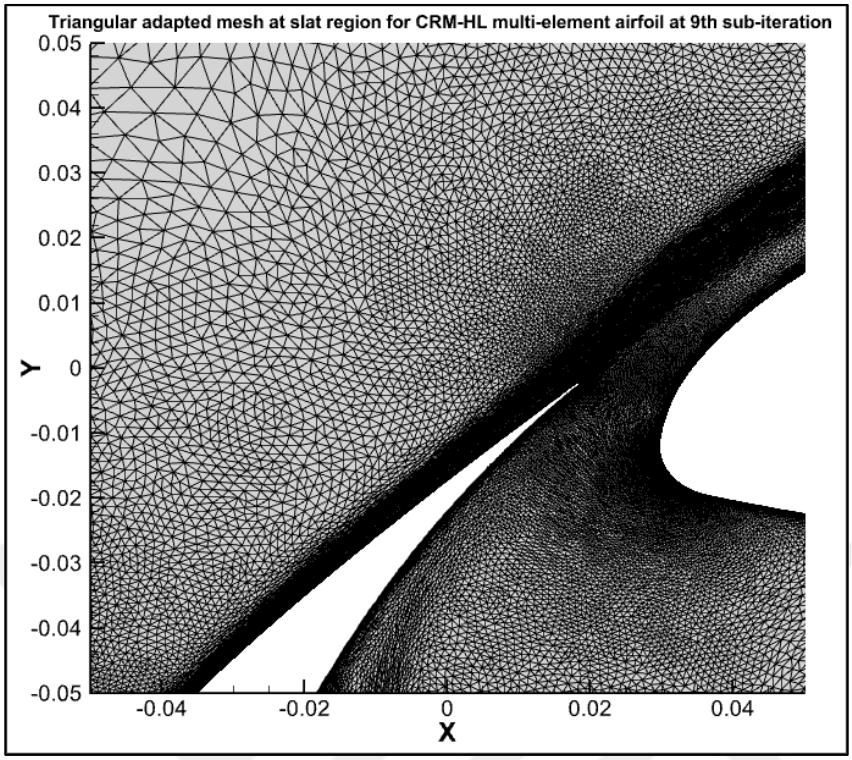

Figure 7.16 : Adapted slat-main element gap region mesh from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16°.

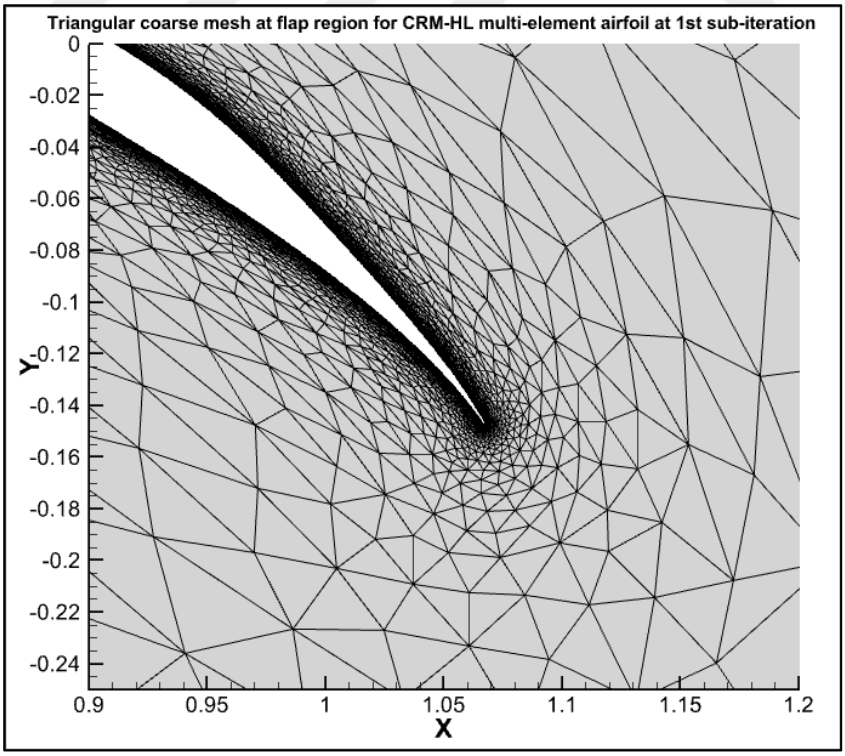

Figure 7.17 : Initial coarse triangular unstructured flap region mesh from 1st sub-iteration for 2D CRM-HL airfoil case.

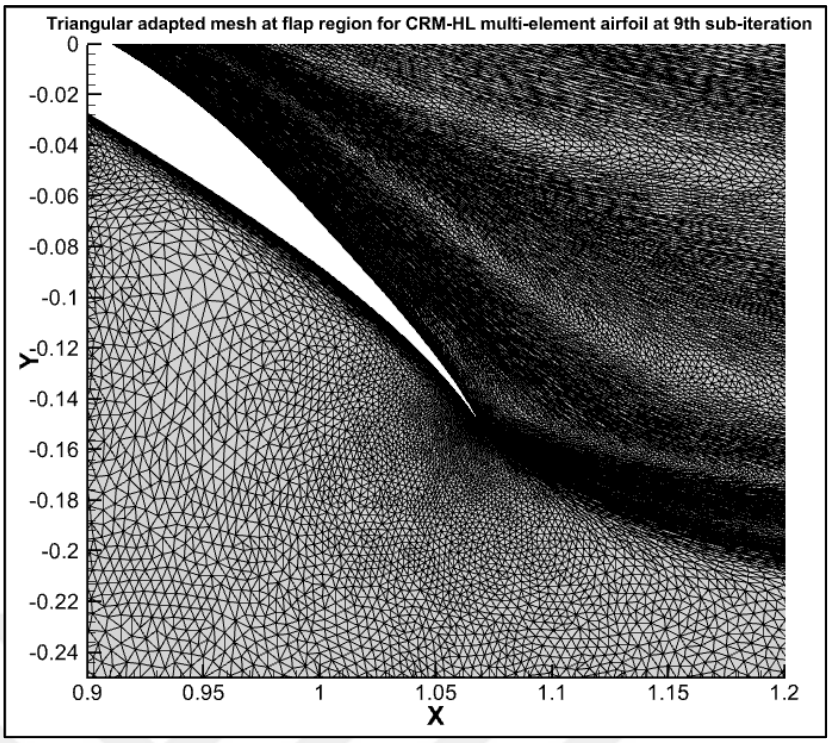

Figure 7.18 : Adapted flap trailing edge region mesh from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16°.

7.2.3 Numerics

Table 7.7 below provides a summary for the numerical solution parameters. For this case, adaptive algorithm was started and continued with 1st order accuracy for 3 sub-iterations for solution stabilization and later switched back to 2nd order for the remaining 3 adaptive sub-iterations.

Table 7.7 : Numerical parameters for the high lift multi-element airfoil case.

Numerical Parameter	Value
BC at r = 1000	Farfield
BC at airfoil boundaries	No-slip, adiabatic wall
Time-dependence	Steady-state
Pseudo-time CFL strategy	Adaptive
CFL parameters (Start, Max, Up rate, Down rate)	[0.1, 10000, 1.125, 0.5]
Gradient scheme	Unweighted least-squares
Spatial order of accuracy for N-S equations	2nd order
Spatial order of accuracy for turbulence equations	1st order
Spatial upwind blending parameter (β_{upwind})	1.0
Flux scheme	Roe
Flux limiter	Off
ω scaling parameter for k-q SST model (A_0)	10^{-12}

7.2.4 Results

All following results are obtained with mesh adaptation at AoA= 16° for the multi-element airfoil. Pressure coefficient distribution around the slat, main element and flap is also provided for comparison of HEMLAB SST adaptive grid results on Figure 7.19 with FUN3D fixed-grid SA-neg results (NASA Langley Research Center (n.d.)). In Figures 7.20 to 22, Mach contours are presented around the airfoil's slat, main element and flap regions. Furthermore, non-dimensional zoomed-in k and ω contours around the slat and flap are shown on Figures 23 to 26. Finally, Figures 27 and 28 display the convergence behaviour of HEMLAB SST implementation for the high-lift multi-element airfoil case.

Once again, unstructured adaptive refinement shines in such complex flow physics scenarios which is clearly seen from Figures 7.16 and 7.18 that show the 9th sub-iteration's adaptive grids for the high-lift case. The shear layer behaviour around the slat and flap are detected by the adaptive algorithm sensors, thereby automatically refining only those regions that require it, without ad-hoc assumptions about how to put the best "fixed mesh refinement zone" for those areas.

Pressure distributions on Figure 7.19 show very good agreement for HEMLAB and FUN3D although dif- ferent grids and different turbulence models have been used. Mach contour in Figure 7.20 shows the leading-edge suction and streamline skewness due to AoA= 16°.

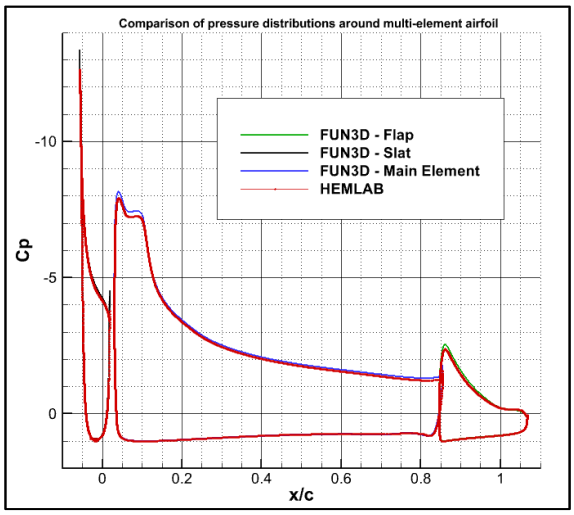

Figure 7.19 : Pressure coefficient distribution plot of the adaptive solution from 9th sub-iteration for 2D CRM-HL airfoil case using HEMLAB $k-\omega$ SST model at AoA=16◦ vs fixed-grid FUN3D SA-neg solution.

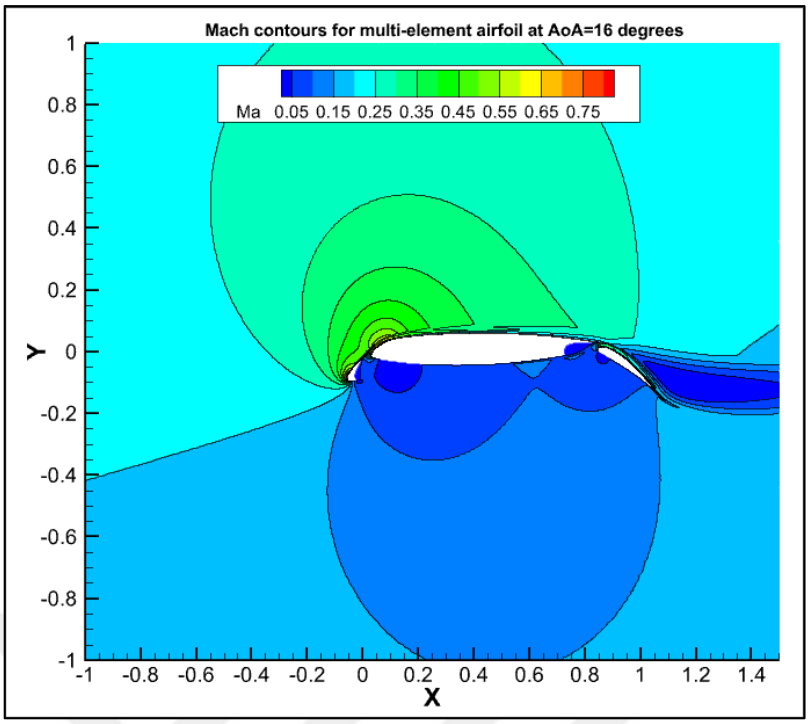

Figure 7.20 : Mach contours around all airfoil parts for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°.

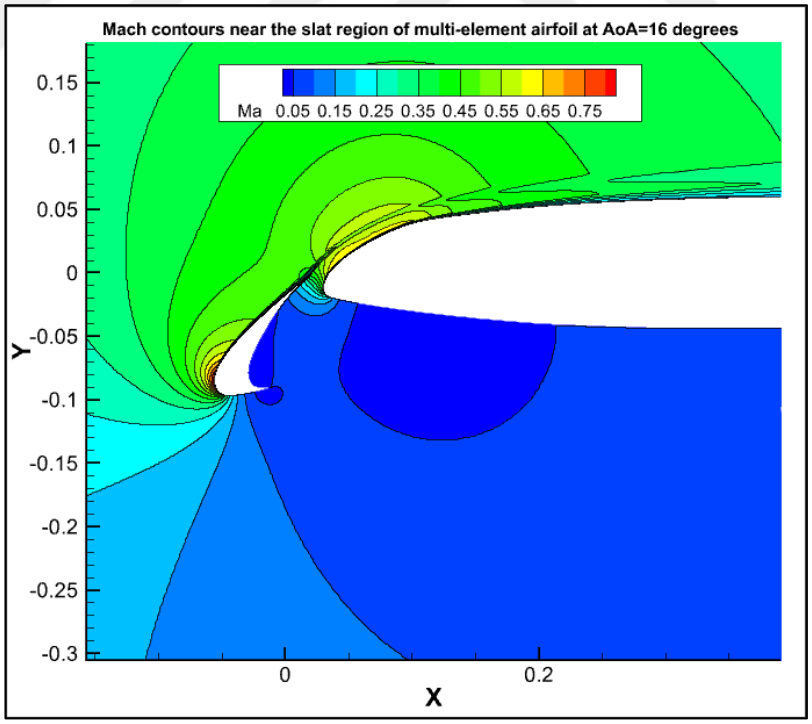

Figure 7.21 : Mach contours around airfoil slat region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16°.

It can be seen from Figure 7.21 above that thanks to the high-lift properties of the slat geometry where extreme curvatures result in increased effective angle of attack for multi-element airfoil configuration, the extent the local Mach number can reach is above 0.75 Mach at slat leading-edge for this subsonic (Ma_∞ = 0.2) case. Besides, a large wake zone with turbulent separated flow exists in Figure 7.22 downstream of flap due to high angle of attack as well as a recirculation zone right under the main-element's trailing edge from which flap was deployed.

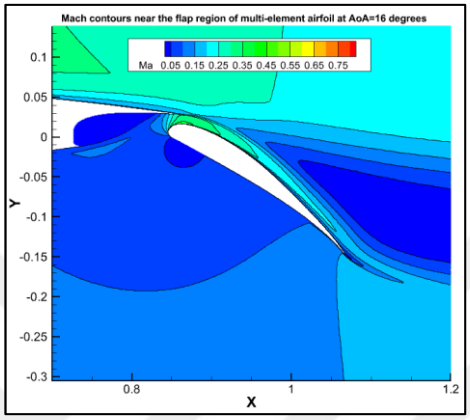

Figure 7.22 : Mach contours around airfoil flap region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°.

In addition, for the non-dimensional k contours shown in Figures 7.23 and 7.24, increased turbulence kinetic energy generation is visible due to shear layer originating from upper surface of the slat and reaching up to main element as well as separated zone behind the flap. Increased turbulent dissipation due to sharp edges of airfoil parts is also present in non-dimensional ω contours in Figures 7.25 and 7.26.

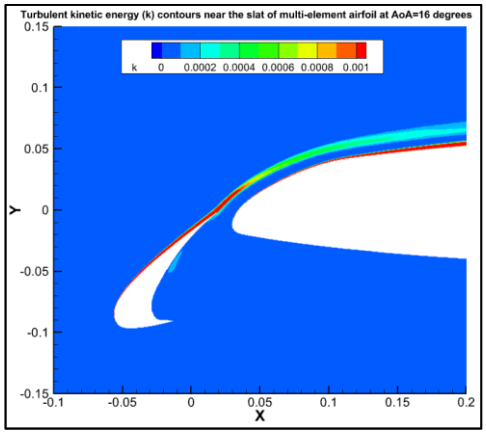

Figure 7.23 : k contours around airfoil slat region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°.

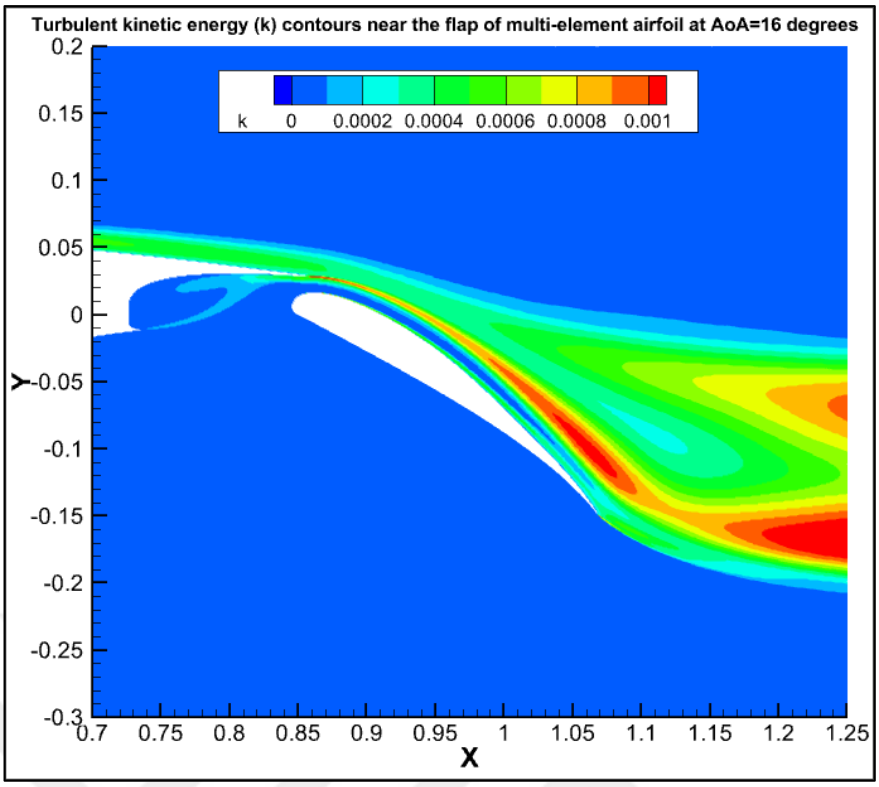

Figure 7.24 : k contours around airfoil flap region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°.

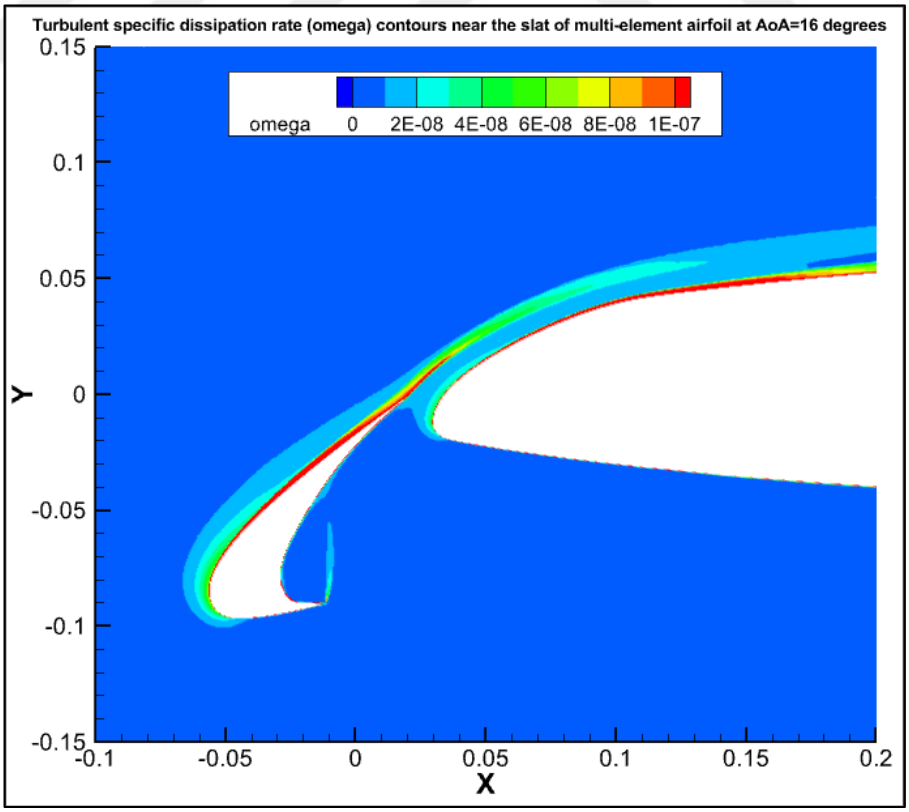

Figure 7.25 : *ω* contours around airfoil slat region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB *k-ω* SST model at AoA=16°.

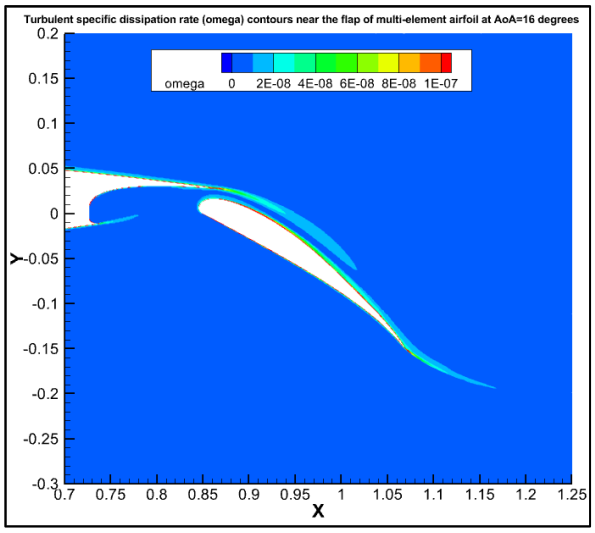

Figure 7.26 : ω contours around airfoil flap region for the adaptive solution from 9th iteration for 2D CRM-HL airfoil case using HEMLAB k-ω SST model at AoA=16°.

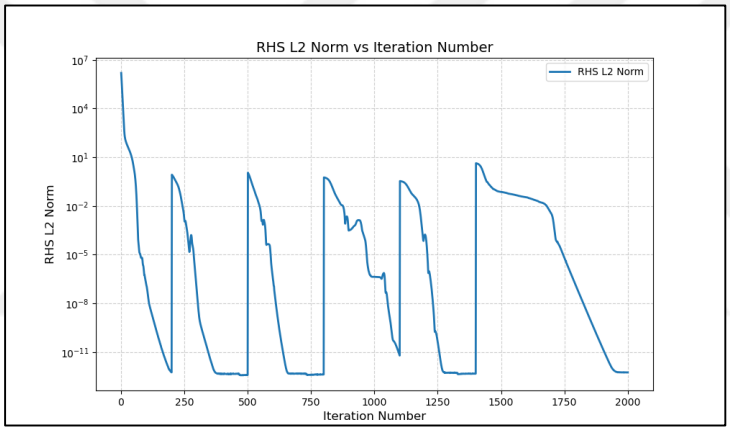

Figure 7.27 : L_2-norm convergence plot for HEMLAB SST model using consecutive anisotropic mesh adaptation for high-lift multi-element airfoil.

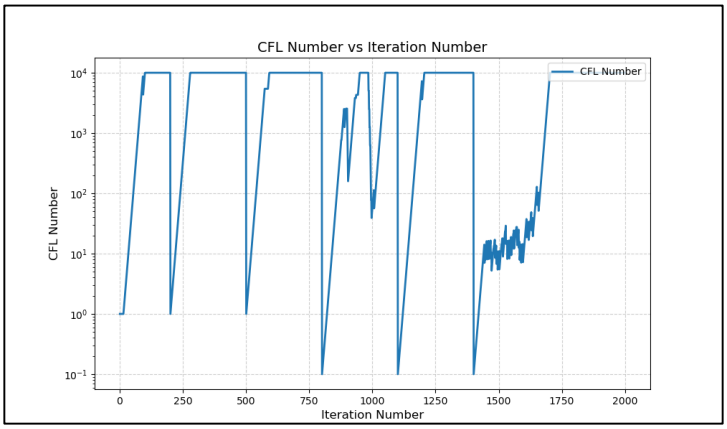

Figure 7.28 : Variation of CFL number vs iteration number for HEMLAB SST model using consecutive anisotropic mesh adaptation for high-lift multi-element airfoil.

Table 7.8: HEMLAB refinement data at $M_\infty = 0.2, \mathrm{Re} = 5 \times 10^6$ and $\alpha = 16^\circ$ with L4 norm, $\beta = 1$ and $\varepsilon = 0.001$.

Refinement level (Sensor function)	Number of vertices	Number of elements	Number of surface elements	C_d	C_l
ite0 (Initial mesh)	325794	642357	9156	-	-
ite1 (Distance)	80517	153908	7051	-	-
ite2 (Distance)	75178	144381	5900	-	-
ite3 (Distance)	69978	134655	5226	-	-
ite4 (Mach)	66675	127779	5496	0.945711	2.557647
ite5 (Mach)	62497	121299	3620	0.587681	2.152007
ite6 (Mach)	70450	138077	2742	0.536952	1.565434
ite7 (Mach)	77263	152101	2334	0.056270	3.798868
ite8 (Mach)	65191	127792	2499	0.063605	3.753702
ite9 (Mach)	515599	1023125	7878	0.065314	3.677077
Reference (SA model, Galbraith et al, 2022)	-	-	-	0.060529	3.802952

7.3 Transonic ONERA OAT15A Supercritical Airfoil (DPW–8 Test Case 1)

7.3.1 Case conditions

Another external aerodynamics verification case targeted the ONERA OAT15A super-critical airfoil prescribed for the 8th AIAA Drag Prediction Workshop (Jacquin et al., 2009; NASA Langley Research Center, (n.d.)). Simulations were conducted at M_∞ = 0.73, $Re_c = 3.0 \times 10^6$ for angle of attack $\alpha = 1.50^\circ$. Table 7.9 below summarizes the physical parameters for this case.

Table 7.9 : Physical conditions for DPW8 transonic ONERA OAT-15 airfoil case.

Physical Condition	Value
Reynolds number (Re_c)	3×10^6
Freestream Mach number (M_∞)	0.73
Angle of attack (α)	1.5°
Angle of sideslip (β)	0°
Freestream non-dim. turbulent kinetic energy (k_∞)	9×10^{-9}
Freestream non-dim. specific dissipation rate (ω_∞)	1×10^{-6}
Reference non-dim. dynamic viscosity (μ)	1.0
Reference non-dim. static pressure (p_i)	0.7143
Reference non-dim. static temperature (T_i)	1.0
Turbulent Prandtl number (Pr_t)	0.9
Specific heat ratio (γ)	1.4

7.3.2 Mesh

Mesh convergence followed the official DPW-8 6-level grid family where Level-2 grid is shown below in which y+ < 1 with a farfield boundary placed 1000 chord lengths away from the no-slip airfoil walls. Adaptive refinements were not used here to keep results directly comparable to committee baseline results. Table 7.10 below describes grid parameters while the Figures 7.29 to 7.31 shows the original unmeshed geometry and the near-wall regions of ONERA OAT-15 airfoil for Level-2 grid.

Table 7.10 : Mesh parameters for DPW8 transonic ONERA OAT-15 airfoil case.

Mesh Parameter	Value
Grid type	Structured, o-grid
Element type	Rectangular
Fixed/Adaptive Grid	Fixed
Domain extent	1000 × c
Reference chord length for Re_c (c)	0.23
Reference area (A_{ref})	0.23
Coarse mesh (Level-1) node count	152088
Finest mesh (Level-6) node count	1473761

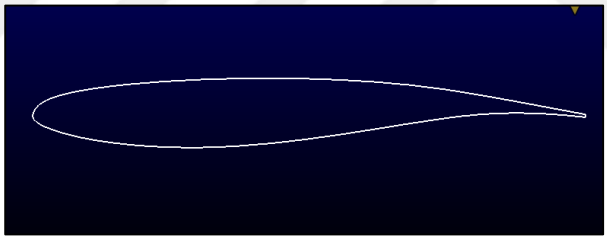

Figure 7.29 : ONERA OAT-15 airfoil geometry.

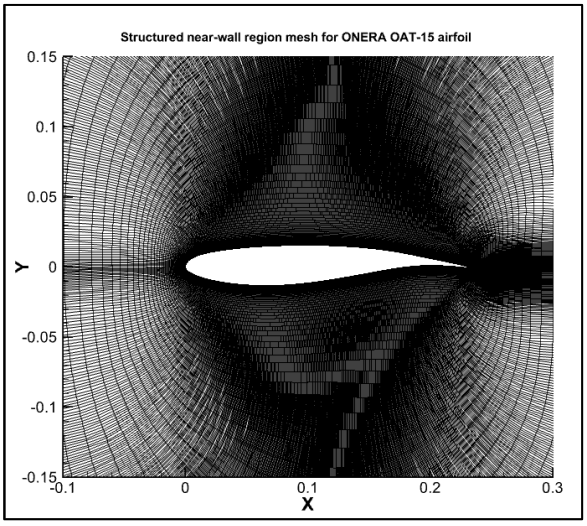

Figure 7.30 : Level-2 structured near-wall mesh for ONERA OAT-15 airfoil case.

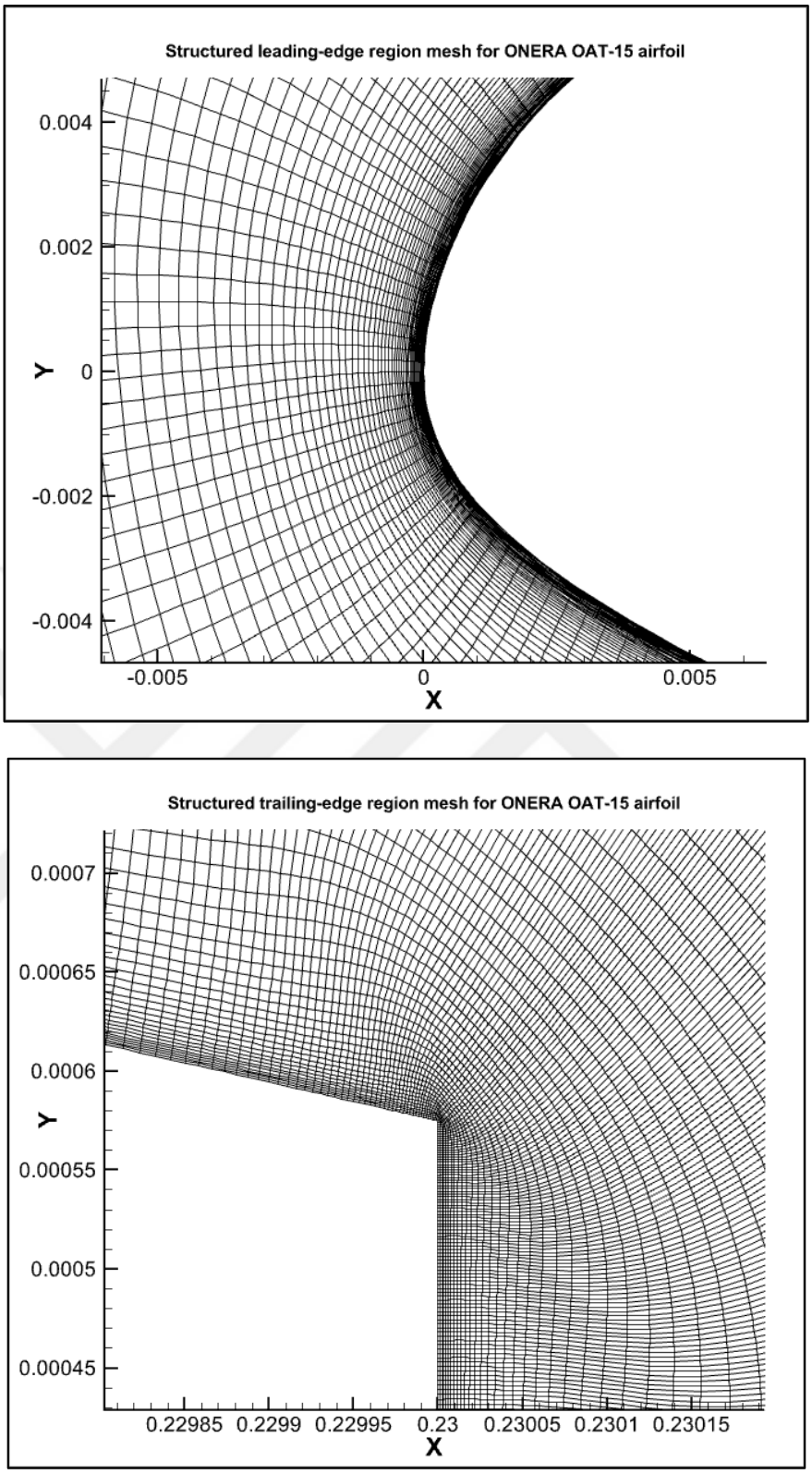

Figure 7.31 : Zoomed views for Level-2 structured ONERA OAT15A mesh: (top) leading and (bottom) trailing edge.

7.3.3 Numerics

Table 7.11 below provides a summary for the numerical solution parameters. For this case, adaptive sub-iterations were started with 1st order accuracy for N-S equations for solution stabilization and later switched back to 2nd order. to keep results directly comparable to committee baseline results.

Table 7.11 : Numerical parameters for the transonic ONERA OAT-15 airfoil case.

Numerical Parameter	Value
BC at r = 1000	Farfield
BC at airfoil boundaries	No-slip, adiabatic wall
Time-dependence	Steady-state
Pseudo-time CFL strategy	Adaptive
CFL parameters (Start, Max, Up rate, Down rate)	[0.1, 10000, 1.125, 0.5]
Gradient scheme	Unweighted least-squares
Spatial order of accuracy for N-S equations	2nd order
Spatial order of accuracy for turbulence equations	1st order
Spatial upwind blending parameter (β_{upwind})	1.0
Flux scheme	Roe
Flux limiter	Off
ω scaling parameter for k-q SST model (A_0)	10^{-12} and 10^{-12}

7.3.4 Results

All following results are obtained without mesh adaptation at AoA=1.5° for the transonic ONERA OAT15 airfoil for Level-2 structured grid except the pressure coefficient comparison which is at AoA= 2.5°. In Figures 7.32 to 7.35, Mach contours, C_p distributions and non-dimensional k and ω contours are presented around the airfoil. Table 7.12 presents drag and lift coefficient convergence of the HEMLAB SA-neg and k-ω SST models using the 6 structured grids provided for the ONERA airfoil. Experimental pressure coefficient data is provided for validation of HEMLAB SST implementation on Figure 7.33 (Jacquin et al., 2009). Finally, Figures 7.36 and 7.37 display the convergence behaviour of HEMLAB SST implementation for the ONERA OAT-15 transonic airfoil case.

Mach contour in Figure 7.32 shows the clear lambda shock structure near the airfoil wall with around the middle part of the upper airfoil surface which leads to a sudden deceleration of the local supersonic flow back to subsonic levels, although in a weak

manner since supercritical airfoils have the tendency to reduce wave drag by design compared to a standard transonic airfoil (Jacquin et al., 2009).

Upon inspection of Figure 7.33, the shock location and sudden pressure increase at AoA=2.5° is predicted slightly later than the experiment. Still, general trend of the surface pressure distribution is correctly captured by current SST model implementation and even much better than the existing SA-neg turbulence model in HEMLAB.

Similarly, there is no visible large-scale separation zone after the shock which can be seen from the relatively unchanged *k* contours in Figure 7.34 - which is desired for supercritical airfoils by design (Jacquin et al., 2009). However, the sudden increase in turbulent dissipation that is evident from the non-dimensional ω contours in Figure 7.35 corresponds to the lambda shock system's location.

Furthermore, convergence of the HEMLAB SST solution shown in Figures 7.36 and 7.37 is rather good considering the complex flow physics of the transonic airfoil case, where it can be seen that the L_2-norm of the residual can reach up to 1×10^{-10} and CFL up to 10000 even for 2nd order numerical accuracy.

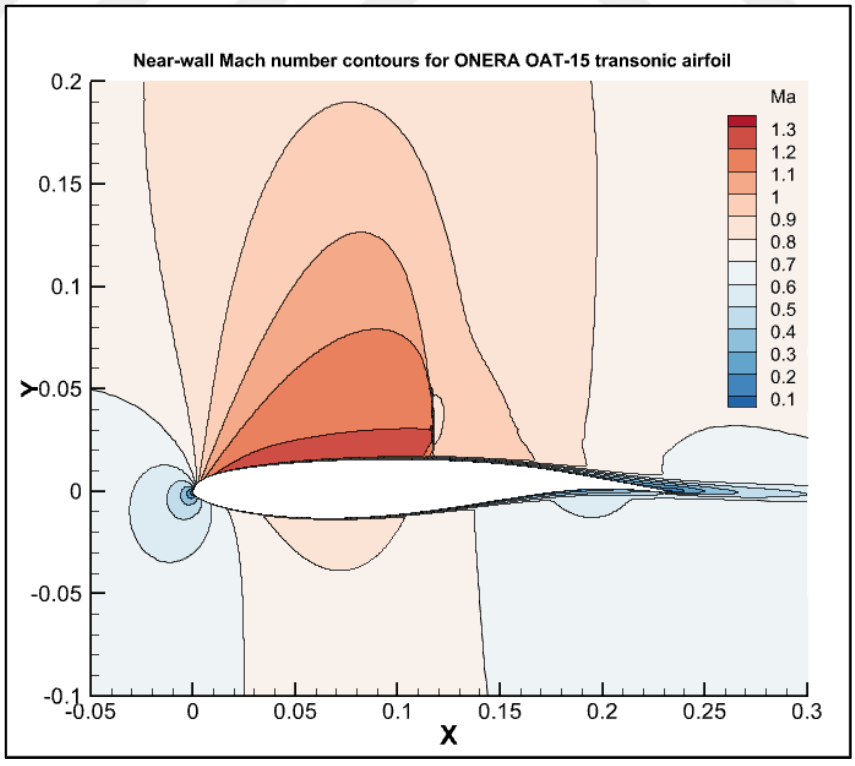

Figure 7.32 : Mach contours around ONERA airfoil for the Level-2 grid solution using HEMLAB *k*-ω SST model at AoA=1.5°.

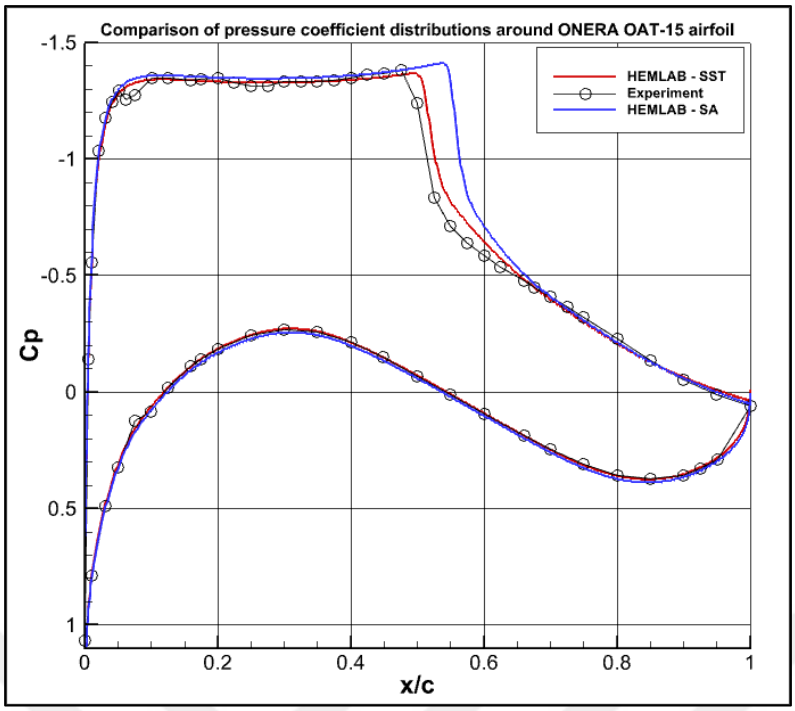

Figure 7.33 : Pressure coefficient distribution plots around the transonic ONERA OAT-15 airfoil using the Level-2 structured grid for HEMLAB *k-ω* SST model at AoA=2.5◦ vs experimental data and HEMLAB SA-neg model.

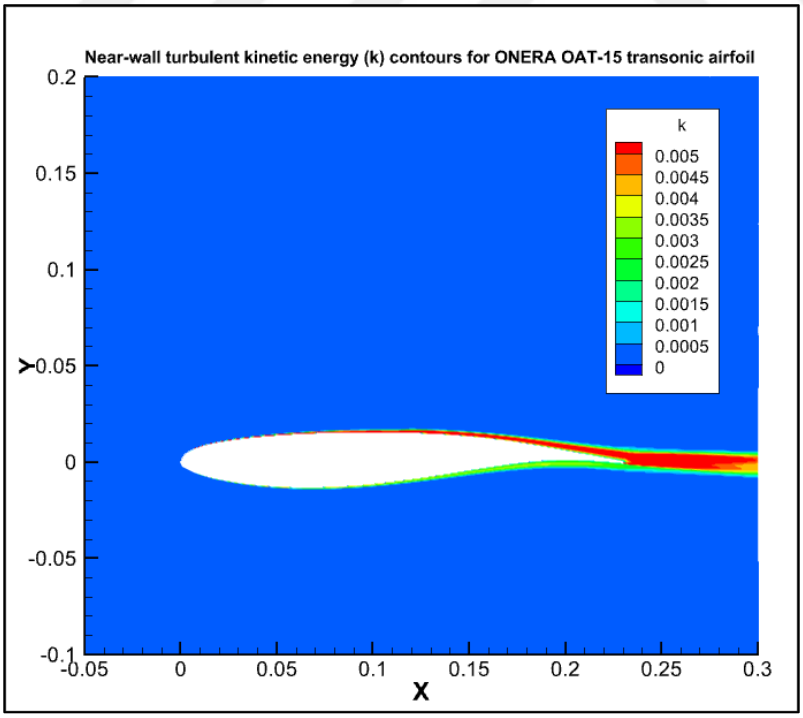

Figure 7.34 : *k* contours around ONERA airfoil for the Level-2 grid solution using HEMLAB *k-ω* SST model at AoA=1.5°.

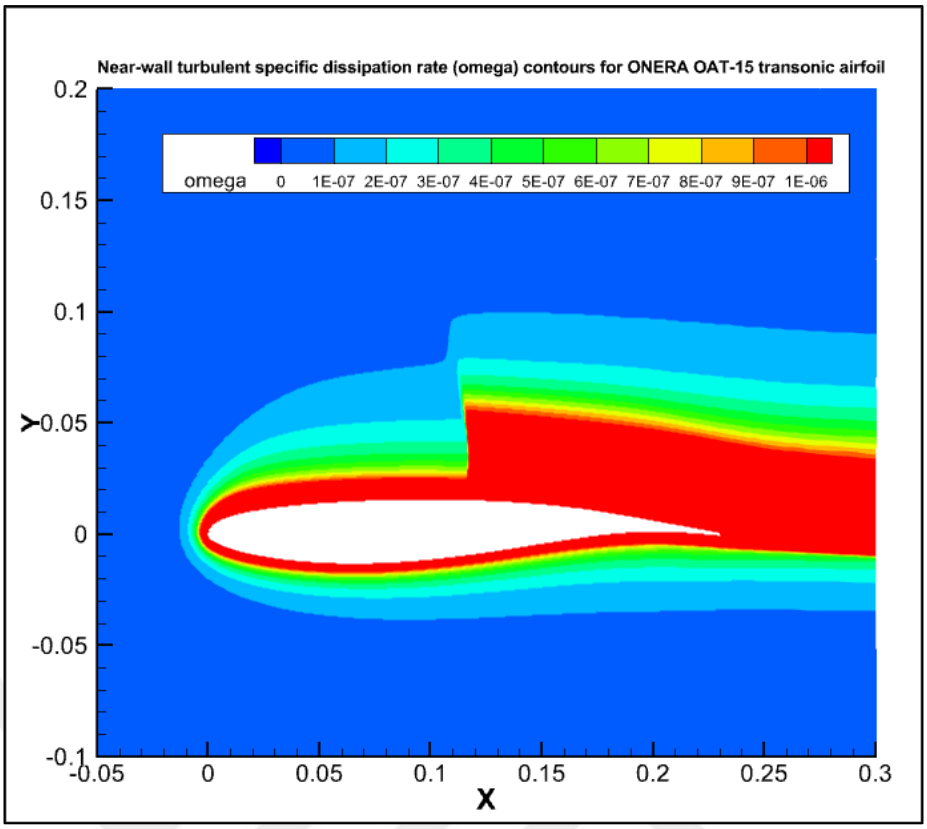

Figure 7.35 : ω contours around ONERA airfoil for the Level-2 grid solution using HEMLAB k-ω SST model at AoA=1.5°.

Table 7.12: Grid-convergence of HEMLAB drag and lift coefficients at $Ma_\infty = 0.73$ and $\alpha = 1.5^\circ$ for the ONERA OAT15 transonic airfoil study.

Mesh Level	Nodes	Turbulence Model	C_D	C_L
1	152 088	SA-neg	0.015230	0.8060
1	152 088	k–ω SST	0.014635	0.7662
2	241 845	SA-neg	0.015103	0.8086
2	241 845	k–ω SST	0.014523	0.7712
3	379 827	SA-neg	0.015082	0.8092
3	379 827	k–ω SST	0.014522	0.7732
4	598 567	SA-neg	0.015081	0.8097
4	598 567	k–ω SST	-	-
5	939 108	SA-neg	0.015084	0.8098
5	939 108	k–ω SST	-	-
6	1 473 761	SA-neg	0.015086	0.8099
6	1 473 761	k–ω SST	-	-

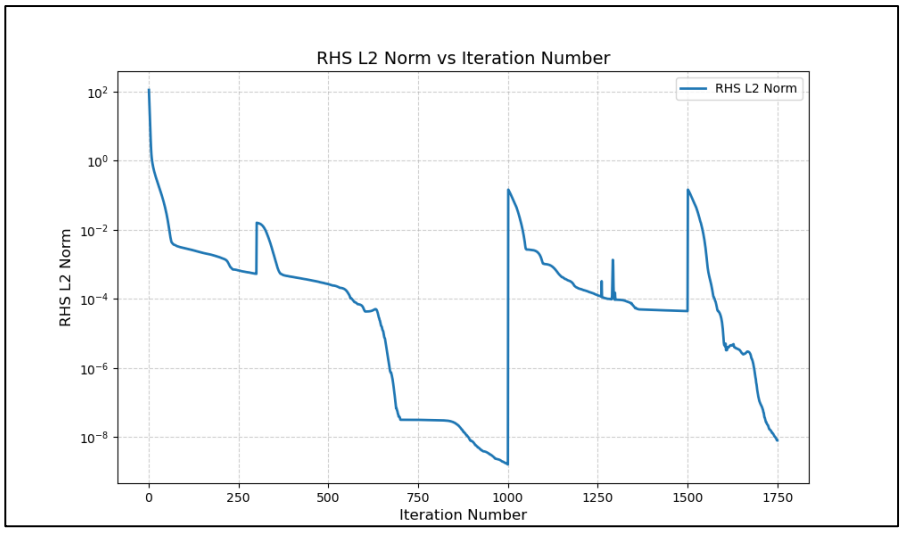

Figure 7.36 : L_2-norm convergence plot for HEMLAB SST model using the Level-2 structured grid for transonic ONERA OAT-15 airfoil..

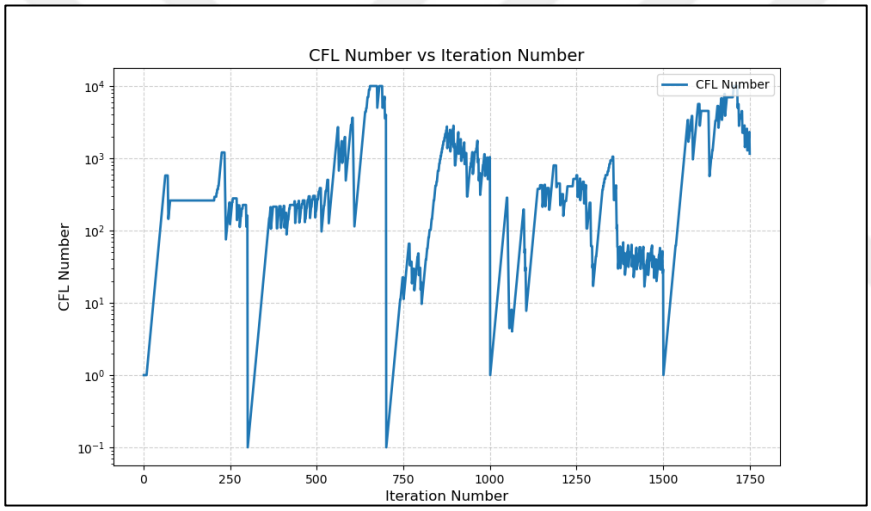

Figure 7.37 : Variation of CFL number vs iteration number for HEMLAB SST model using the Level-2 structured grid for transonic ONERA OAT-15 airfoil.

7.4 Subsonic NACA-0012 Airfoil

7.4.1 Case conditions

Subsonic flow around the NACA-0012 airfoil was also investigated under conditions $Re_c = 6 \times 10^6$ and $M_\infty = 0.15$ for the implementation of the HEMLAB *k-ω* SST model at angles of attack 0°,10°and 15°. The case conditions are summarized in Table 7.13 below.

Table 7.13 : Physical conditions for the subsonic NACA-0012 airfoil case.

Physical Condition	Value
Reynolds number (Re_c)	6×10^6
Freestream Mach number (M_∞)	0.15
Angle of attack (α)	[0°, 10°, 15°]
Angle of sideslip (β)	0°
Freestream non-dim. turbulent kinetic energy (k_∞)	9×10^{-9}
Freestream non-dim. specific dissipation rate (ω_∞)	1×10^{-6}
Reference non-dim. dynamic viscosity (μ)	1.0
Reference non-dim. static pressure (p_i)	0.7143
Reference non-dim. static temperature (T_i)	1.0
Turbulent Prandtl number (Pr_t)	0.9
Specific heat ratio (γ)	1.4

7.4.2 Mesh

A mesh independence study was simultaneously carried out at each angle of attack case by creating 5 different structured o-grid type mesh around the blunt trailing-edged airfoil. Firstly, number of surface points were decided on using a domain radius of 500 chord lengths of the airfoil and growth rate of 1.1. Then, effect of the first layer height of the boundary layer was tested by fixing the number of surface points. ”Level-4” grid was considered sufficient with 93796 total vertices and 400 points on airfoil surface with an estimated y+ below 0.5.

Table 7.14 below describes grid parameters while the Figure 7.38 shows drag and lift convergence for mesh independence and near-wall regions of NACA-0012 airfoil for the selected Level-4 grid are shown on Figures 7.39 to 7.41. Finally, Table 7.16 shows the details of mesh-independence study for NACA0012 airfoil.

Table 7.14 : Mesh parameters for the subsonic NACA-0012 airfoil case.

Mesh Parameter	Value
Grid type	Structured, o-grid
Element type	Rectangular
Fixed/Adaptive Grid	Fixed
Domain extent	500 × c
Reference chord length for Re_c (c)	1.0
Reference area (A_{ref})	1.0
Coarse mesh (Level-1) node count	40096
Finest mesh (Level-5) node count	111696

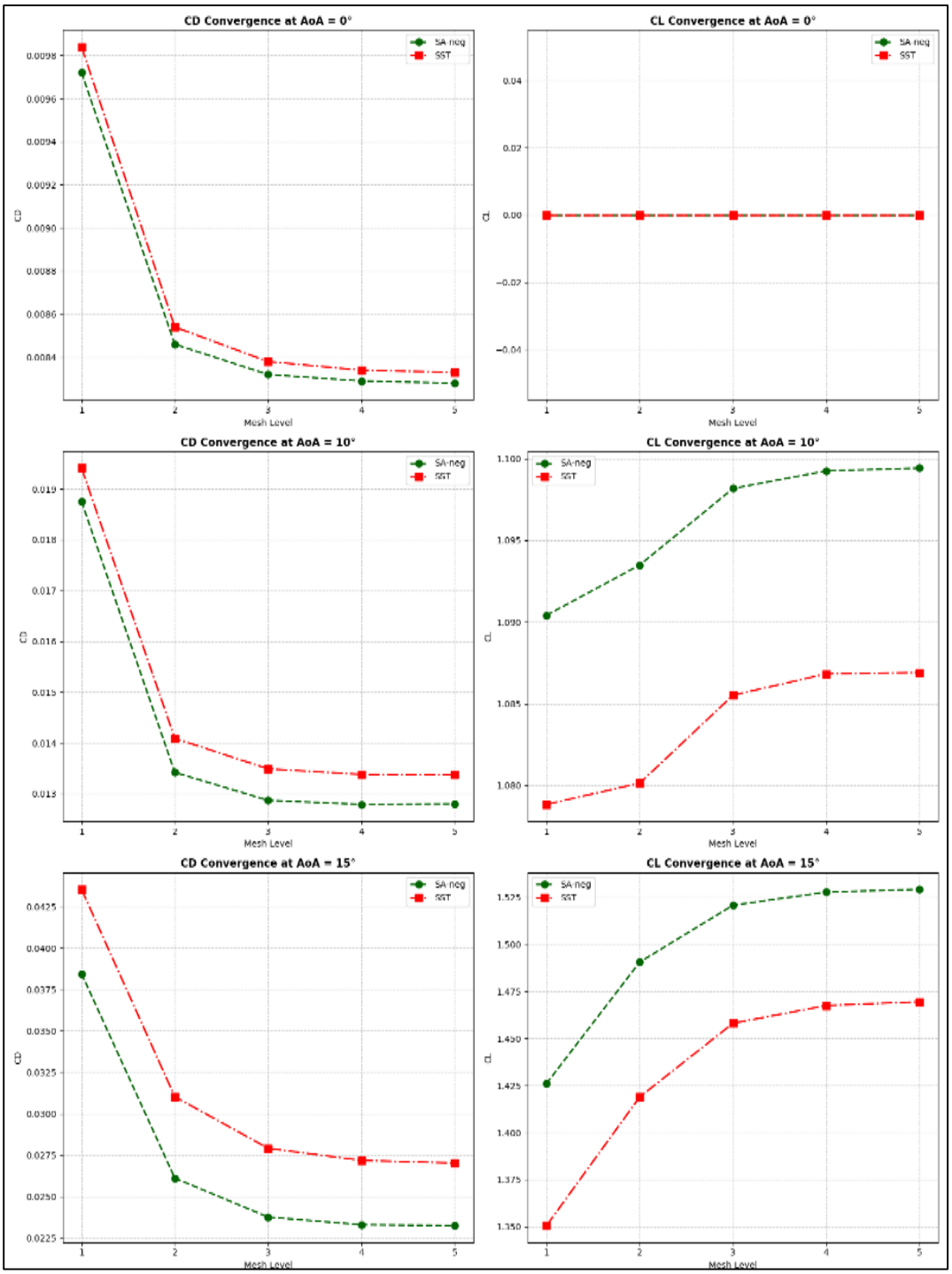

Figure 7.38 : Mesh independence study for HEMLAB using 5 different structured o-grid type meshes at different angles of attack.

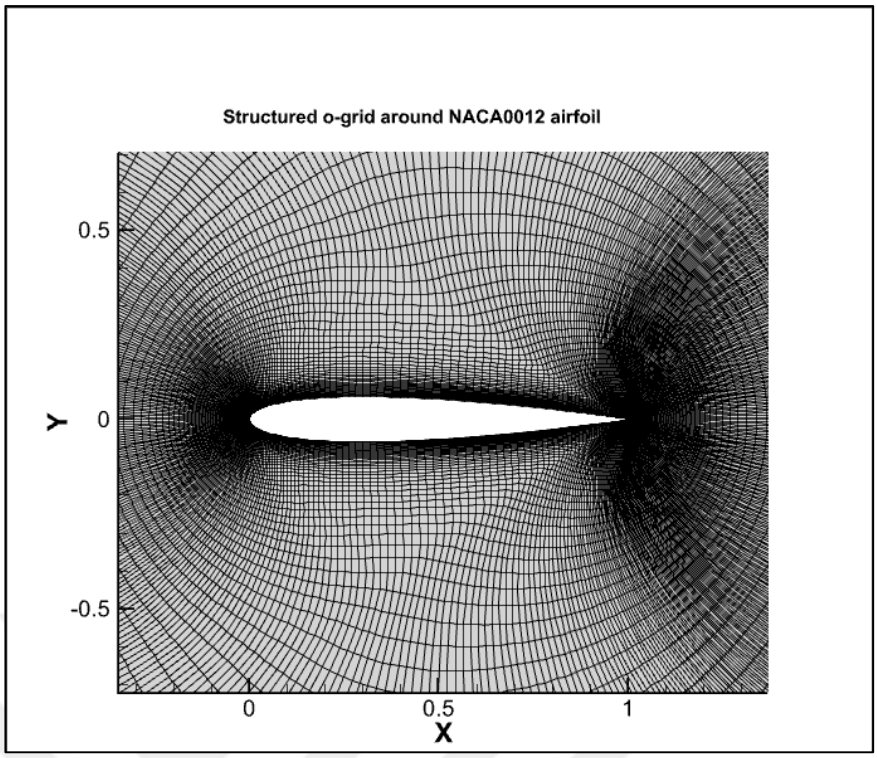

Figure 7.39 : Level-4 structured o-grid for NACA0012 airfoil near-wall region.

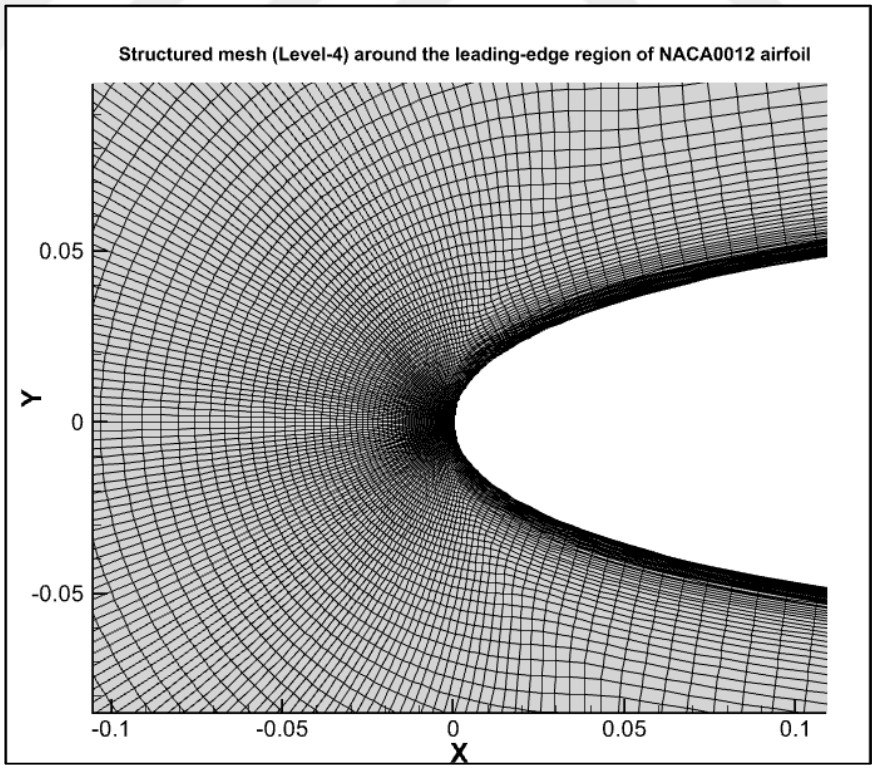

Figure 7.40 : Level-4 structured o-grid for NACA0012 airfoil leading-edge region.

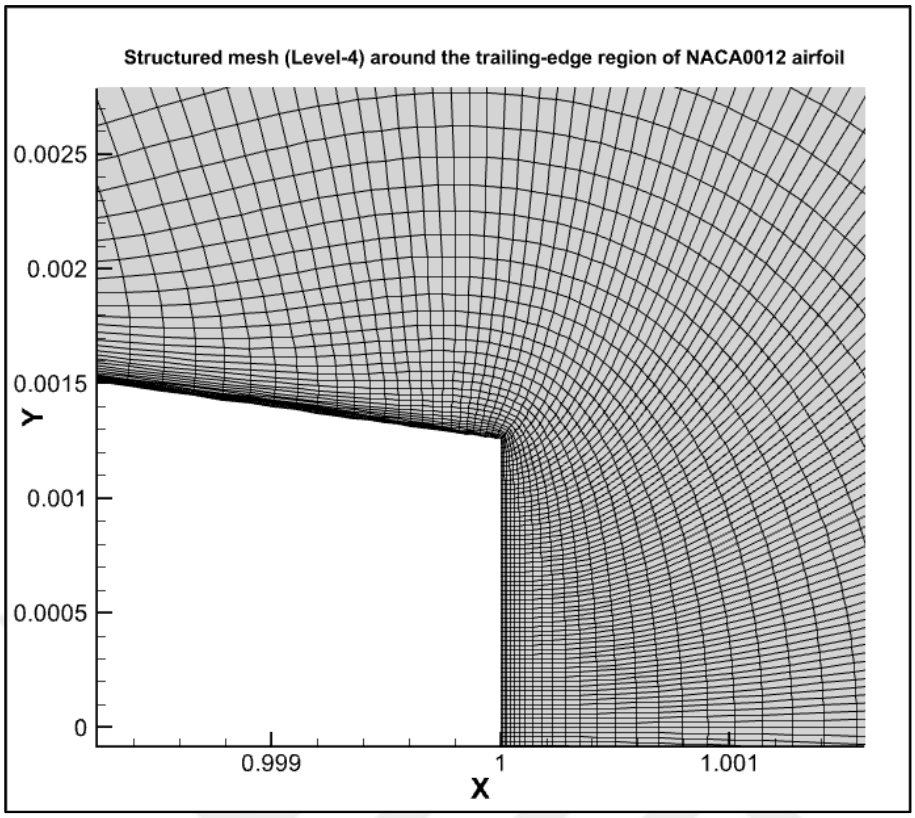

Figure 7.41 : Level-4 structured o-grid for NACA0012 airfoil trailing-edge region.

7.4.3 Numerics

Table 7. 15 below provides a summary for the numerical solution parameters. For this case, adaptive sub-iterations were started with 1st order accuracy for N-S equations for solution stabilization and later switched back to 2nd order for all angle of attack cases.

Table 7.15 : Numerical parameters for the subsonic NACA-0012 airfoil case.

Numerical Parameter	Value
BC at r = 500	Farfield
BC at airfoil boundaries	No-slip, adiabatic wall
Time-dependence	Steady-state
Pseudo-time CFL strategy	Adaptive
CFL parameters (Start, Max, Up rate, Down rate)	[0.1, 10000, 1.125, 0.5]
Gradient scheme	Unweighted least-squares
Spatial order of accuracy for N-S equations	2nd order
Spatial order of accuracy for turbulence equations	1st order
Spatial upwind blending parameter (β_{upwind})	1.0
Flux scheme	Roe
Flux limiter	Off
ω scaling parameter for k-q SST model (A_0)	10^{-10}

7.4.4 Results

All following results are obtained without mesh adaptation, using the Level-4 structured grid selected after the grid independence study. In addition, except the drag polar, all figures are obtained for the AoA= 10° case. Figure 7.42 shows a drag polar for comparing HEMLAB to various reference C_D and C_L data at all angles of attack (NASA Langley Research Center (n.d.)). A C_p plot is also provided for comparison with experimental data at AoA = 10° on Figure 7.43 (NASA Langley Research Center (n.d.)). In Figures 7.44 to 7.46, Mach contours and non-dimensional k and ω contours are also presented around the NACA0012 airfoil. Finally, Figures 7.47 and 7.48 displays the convergence behaviour of HEMLAB SST implementation for the NACA-0012 airfoil case.

The drag polar on Figure 7.42 clearly indicates that as angle of attack parameter is increased, scatter is more visible due to inadequacy of one or two-equation RANS model approximations. However, AoA=15° case needs further investigation for HEMLAB SST model since it is a bit far from the mean of other results as also observed in Figure 7.38. Indeed, drag count differences reach up to 50 counts when SA-neg and SST values are compared from the Table 7.16, although for AoA= 10° it is around 5 counts. This does not happen for FUN3D results, although it could be due to the fact that FUN3D and HEMLAB use different grids.

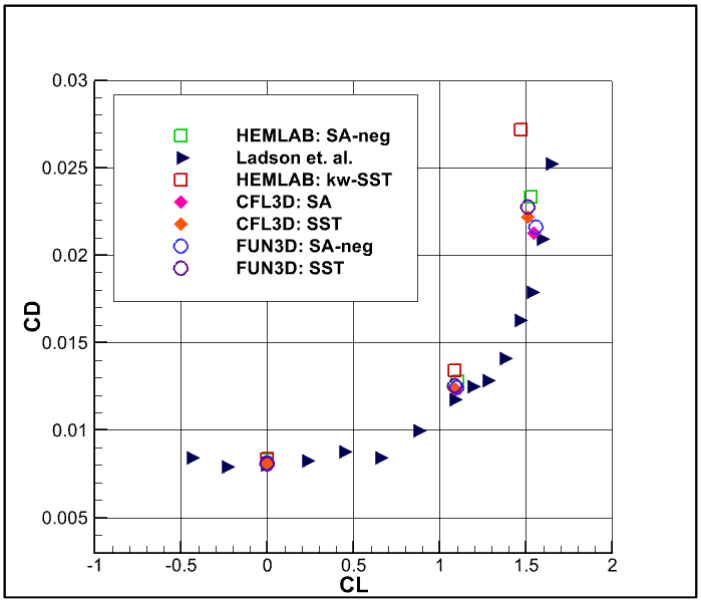

Figure 7.42 : Drag polar for NACA0012 case for different CFD solvers and experimental data.

Pressure coefficient versus x/c plot on Figure 7.43 is a solid indicator that HEMLAB SST implementation is able to resolve inviscid regions and pressure-related effects of the flow appropriately and compares well to HEMLAB SA-neg model distribution. This is validated by experimental distribution except the leading-edge overshoot where this is a known issue with the experimental results (NASA Langley Research Center (n.d.)).

Mach contour in Figure 7.44 shows the leading-edge suction and streamline skewness due to AoA=10°, where for the AoA=0° case a symmetric distribution is normally expected. In addition, for the non-dimensional k and ω contours shown in Figures 7.45 and 7.46, trailing-edge wake clearly shows the shear layer and increased turbulence generation for AoA= 10◦ case. Finally, it can be seen from the Figures 7.47 and 7.48 that the rapid convergence for HEMLAB SST implementation is thanks to exact Jacobian construction in Newton method where L_2-norm of the non-linear residual easily falls to 1×10^{-10} levels and pseudo-CFL number can go up to 10000.

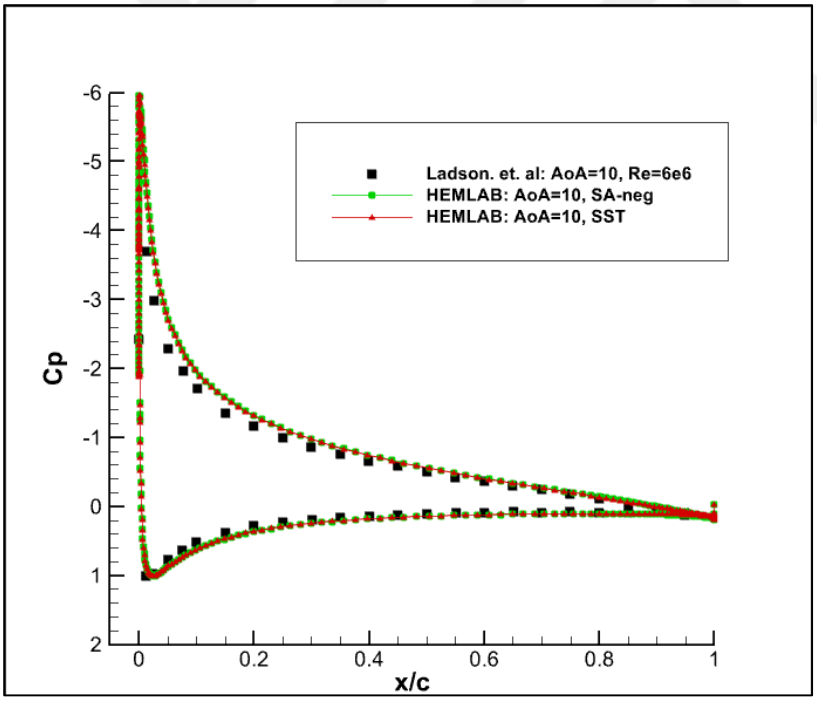

Figure 7.43 : Pressure coefficient plot for NACA0012 case using HEMLAB turbulence models and reference experimental data (NASA Langley Research Center (n.d.)) at AoA=10°.

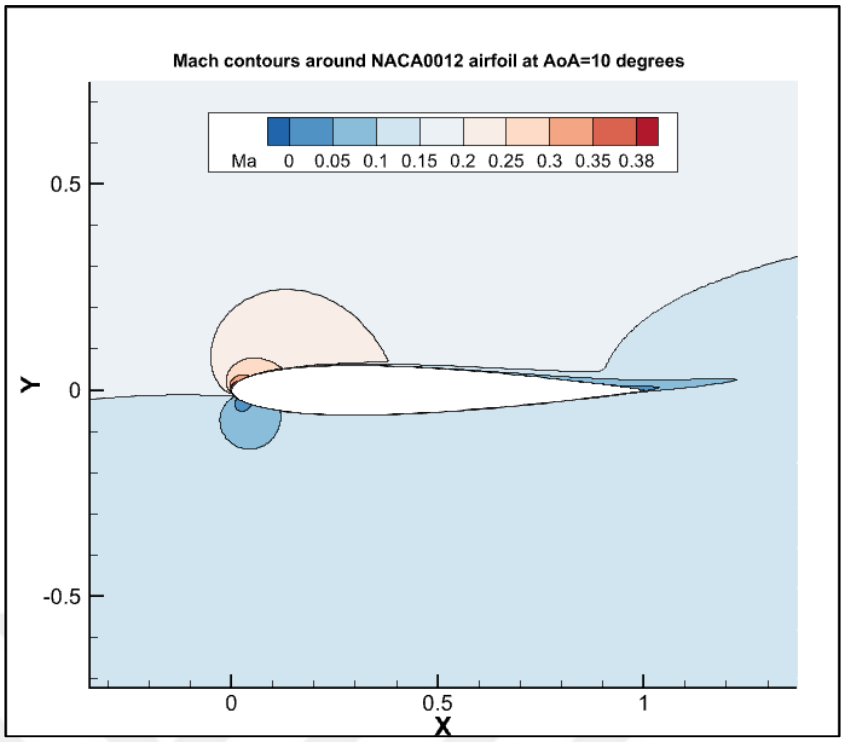

Figure 7.44 : Mach contour plot for NACA0012 case using HEMLAB *k-ω* SST model at AoA=10°.

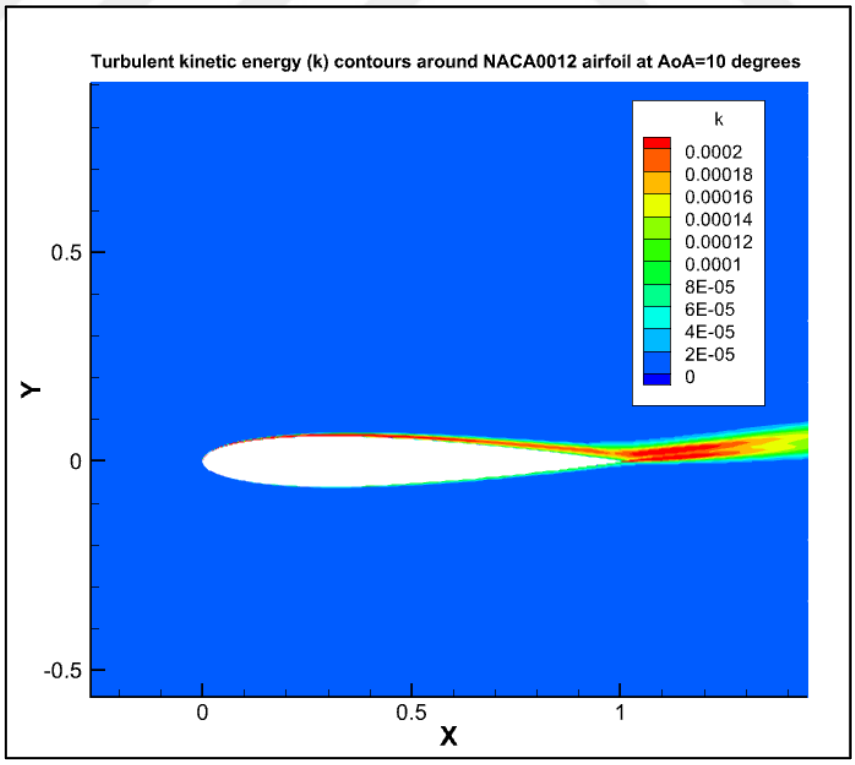

Figure 7.45 : *k* contour plot for NACA0012 case using HEMLAB *k-ω* SST model at AoA=10°.

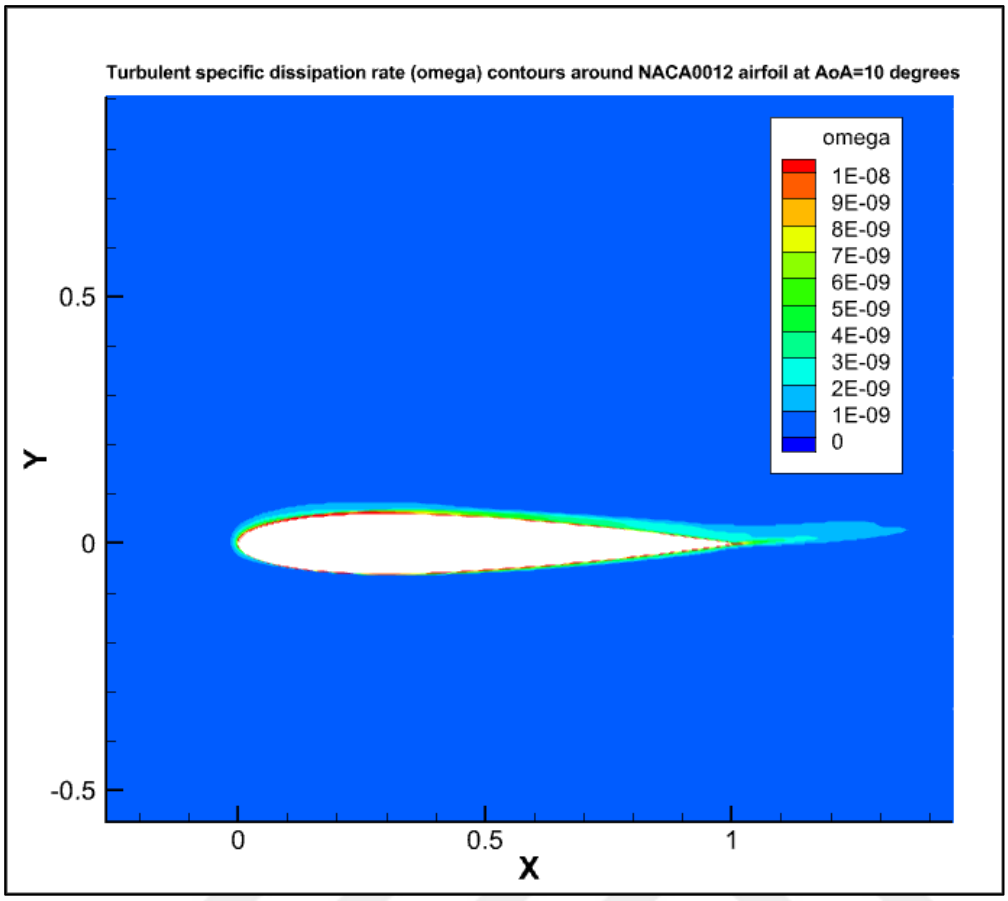

Figure 7.46 : ω contour plot for NACA0012 case using HEMLAB k-ω SST model at AoA=10°.

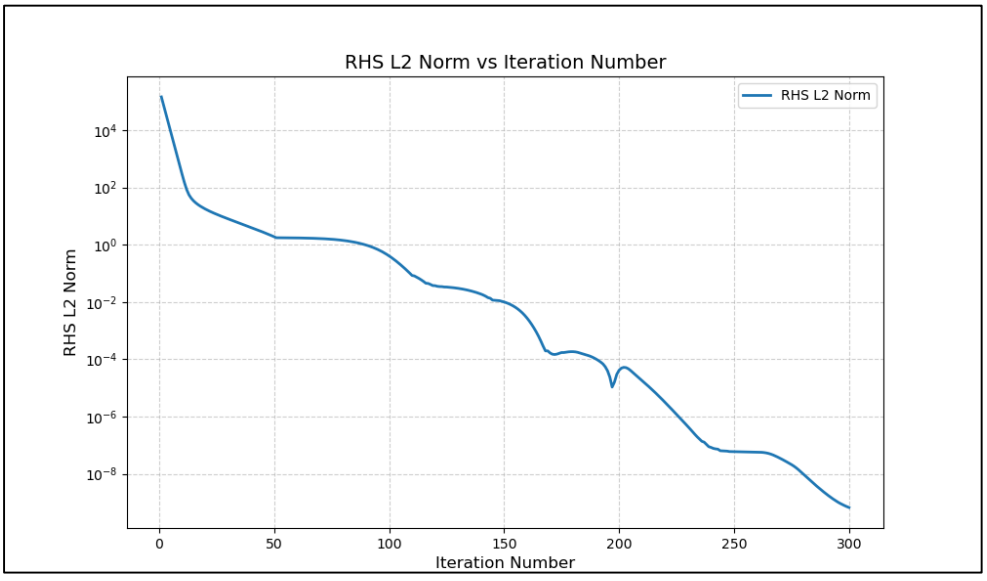

Figure 7.47 : L_2-norm convergence plot for HEMLAB SST model using NACA-0012 Level-4 structured grid.

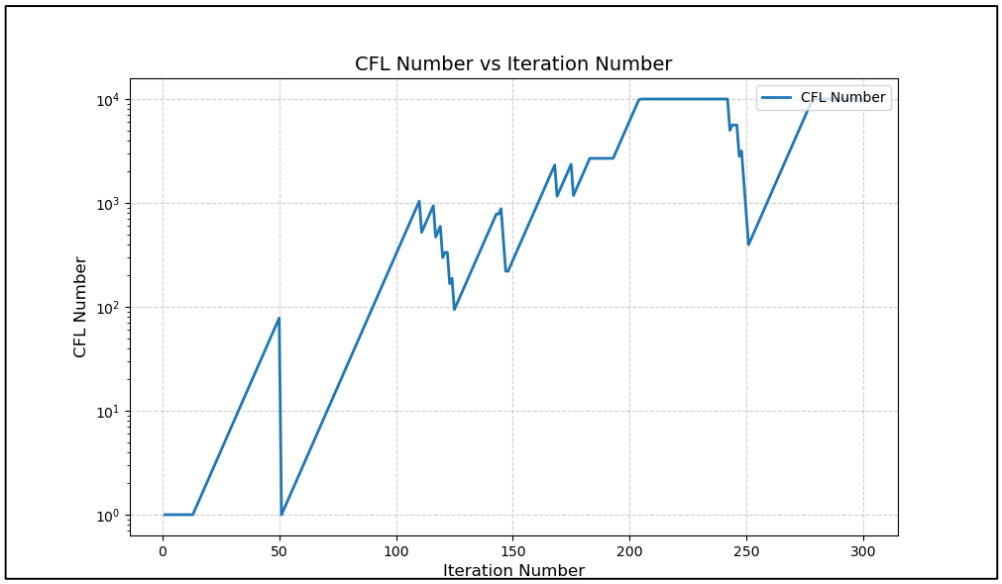

Figure 7.48 : Variation of CFL number for HEMLAB SST model using NACA-0012 Level-4 structured grid.

Table 7.16: HEMLAB mesh independence study for NACA0012 airfoil using the SA-neg & *k-ω* SST models.

Mesh Level	Nodes on Airfoil	1st Layer Height (m)	Growth Rate	Farfield Extent	Total Grid Nodes
1	100	2.00E-6	1.1	500c	40096
2	200	2.00E-6	1.1	500c	57996
3	300	2.00E-6	1.1	500c	75896
4	400	2.00E-6	1.1	500c	93796
5	500	2.00E-6	1.1	500c	111696

Mesh Level	Turbulence Model	C_D			C_L		
		0°	10°	15°	0°	10°	15°
1	SA-neg	0.00972	0.01875	0.03845	0.00000	1.09041	1.42607
1	*k-ω* SST	0.00984	0.01941	0.04354	0.00000	1.07884	1.35054
2	SA-neg	0.00846	0.01342	0.02612	0.00000	1.09349	1.49055
2	*k-ω* SST	0.00854	0.01409	0.03103	0.00000	1.08014	1.41892
3	SA-neg	0.00832	0.01287	0.02377	0.00000	1.09817	1.52074
3	*k-ω* SST	0.00838	0.01349	0.02792	0.00000	1.08552	1.45805
4	SA-neg	0.00829	0.01279	0.02331	0.00000	1.09925	1.52771
4	*k-ω* SST	0.00834	0.01338	0.02719	0.00000	1.08683	1.46747
5	SA-neg	0.00828	0.01280	0.02325	0.00000	1.09942	1.52917
5	*k-ω* SST	0.00833	0.01337	0.02703	0.00000	1.08691	1.46948

8. CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

As a conclusion, the *k-ω* SST model has been transformed into a *k-q* form and integrated into HEMLAB in fully-coupled (monolithic) fashion. The flux Jacobians use exact SST source-term contributions inside the non-linear Newton method implementation. Positivity of ω variable was maintained largely due to *k-q* model transform which results in a simple, Dirichlet BC on the walls. These together allowed the pseudo-CFL number to easily rise to up to 10 000 while the L_2-norm of residual also fell to 10^{-10}–10^{-12} levels for "all" four test cases that was selected in this study.

For the zero-pressure-gradient flat-plate benchmark ($Re = 5\times10^6$, $M_\infty = 0.20$, 545×385 grid), skin-friction plots match with theory and calculated flat plate drag coefficient ($C_D = 0.00289$) is within almost half drag count of FUN3D. In addition, boundary layer contours show detailed *k-ω* comparisons that match FUN3D profiles very well.

In the CRM-HL high-lift airfoil case ($Re = 5\times10^6$, $\alpha = 16°$, $M_\infty = 0.20$), anisotropic adaptation created an unstructured triangular mesh with 516000 nodes which automatically refined the slat gap shear layer and flap wake based on Mach sensor. The resulting pressure-coefficient distributions on slat, flap and main element match FUN3D results and convergence is excellent during each adaptive cycle despite the complex turbulent separated flow physics.

The transonic ONERA OAT15A super-critical airfoil ($Re = 3\times10^6$, $M_\infty = 0.73$, $\alpha = 1.5°$) test showed robustness of current SST implementation when shock discontinuities are present in the flow, where for the first 3 level grids, CFL was still able to reach up to 10000. Although the lambda-shock is captured slightly behind the experimental position, general trend is aligned well with experiment.

Finally, for the subsonic NACA-0012 airfoil ($M\infty = 0.15$, $Re = 6\times10^6$), a mesh convergence was shown for each of the 0–10-15° angle-of-attack sweeps, respectively. SST model shows very similar pressure distributions to experimental results at $\alpha =$

10°. In addition, drag differences are within 5 counts relative to HEMLAB SA-neg model at 10°, although model discrepancies become clearer at 15° flow physics.

These 2-D flat-plate, high-lift, transonic and subsonic airfoil verifications show that the fully-coupled *k-ω* SST implementation in HEMLAB solver produces physically accurate and mesh-independent results while being also numerically robust over a wide range of external-aerodynamics problems.

8.2 Future Work

The SST model verification studies in this thesis were restricted to two-dimensional, steady-state cases and used first-order upwinding for convective terms of turbulence model equations. In addition, numerical behaviour under transient flow conditions is still unexplored, and the SST model does not have transition modeling capability yet. System rotation and streamline curvature effects on turbulence production are also ignored in current SST model. Furthermore, model does not support wall-functions or high $y+$ meshes. Also, the HEMLAB input file syntax does not allow the user to select among variants of *k-ω* SST model at the moment. Therefore, these limitations show logical directions for extending the current SST model implementation in HEMLAB.

REFERENCES

Akkurt, S., & 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 (1), 13-31. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.5045 doi: https://doi.org/10.1002/fld.5045

Alauzet, F., & Frazza, L. (2021). Fature-based and goal-oriented anisotropic mesh adaptation for rans applications in aeronautics and aerospace. Journal of Computational Physics, 439, 110340. Retrieved from http://dx.doi.org/10.1016/j.jcp.2021.110340 doi: 10.1016/j.jcp.2021.110340.

Alauzet, F., & Loseille, A. (2016). A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. Computer-Aided Design, 72 , 13-39. Retrieved from http://dx.doi.org/10.1016/j.cad.2015.09.005 doi: 10.1016/j.cad.2015.09.005.

ANSYS, Inc.. (2020). Best practice: RANS turbulence modeling in ANSYS CFD. ANSYS, Inc. Retrieved from https://www.ansys.com/resource-center/technical-paper/best-practice-rans-turbulence-modeling-in-ansys- (Accessed: 2025-04-25)

Baker, T. (1997). Mesh adaptation strategies for problems in fluid dynamics. Finite Elements in Analysis and Design, 25 (3–4), 243–273. doi: 10.1016/S0168-874X(96)00032-7

Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., . . . Zhang, H. (2018). PETSC users manual (Tech. Rep. Nos. ANL-95/11 – Revision 3.10). Argonne National Laboratory. Retrieved from http://www.mcs.anl.gov/petsc

Blazek, J. (2001). Computational fluid dynamics: Principles and applications. Elsevier**.**

Coakley T. J. & Huang , P. G. (1992). Turbulence modeling for high speed ows. AIAA Paper 92-0436.

Content, C., Outtier, P.-Y., & Cinnella, P. (2013). Coupled/uncoupled solutions of rans equations using a jacobian-free newton-krylov method. In 21st aiaa computational fluid dynamics conference. American Institute of Aeronautics and Astronautics. Retrieved from https://doi.org/10.2514/6.2013-2423 doi: 10.2514/6.2013-2423.

Eça, L., & Hoekstra, M. (2004). On the grid sensitivity of the wall boundary condition of the k–ω turbulence model. Journal of Fluids Engineering, 126 (6), 900–910. doi: 10.1115/1.1845492

Economon, T. D., Palacios, F., Copeland, S. R., Mesnard, O., Jasa, J. P., & Alonso, J. J. (2016). SU2: An open-source suite for multiphysics simulation and design. AIAA Journal , 54 (3), 828–846. doi: 10.2514/1.J053813

Fidkowski, K., & Darmofal, D. (2011). Review of output-based error estimation and mesh adaptation in computational fluid dynamics. AIAA Journal , 49 (4), 673–694. doi: 10.2514/1.J050073.

Galbraith, M. C., Ursachi, C. I., Chandel, D., Allmaras, S. R., Darmofal, D. L., Glasby, R. S., ... & Alonso, J. J. (2022). Comparing multi-element airfoil flow solutions using multiple solvers with output-based adapted meshes. *AIAA Journal*, *60*(4), 2629-2643.

Habashi, W.G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M. and Vallet, M.-G. (2000), Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles. Int. J. Numer. Meth. Fluids, 32: 725-744. https://doi.org/10.1002/(SICI)1097-0363(20000330)32:6<725::AID-FLD935>3.0.CO;2-4.

Harten, A. (1983). High resolution schemes for hyperbolic conservation laws. Journal of Computational Physics, 49 (3), 357-393. Retrieved from https://www.sciencedirect.com/science/article/pii/0021999183901365 doi: https://doi.org/10.1016/0021-9991(83)90136-5.

Hellsten, A. (1998). Some improvements in Menter's k-omega SST turbulence model. In 36th aiaa aerospace sciences meeting and exhibit. Reno, NV

Ilinca, F., & Pelletier, D. (1999). Positivity preservation and adaptive solution of two-equation models of turbulence. *International Journal of Thermal Sciences*, *38*(7), 560–571. https://doi.org/10.1016/S0035-3159(99)80036-1

Inria. (n.d.). Tapenade Automatic Differentiation Engine. https://team.inria.fr/ecuador/en/tapenade/. (Accessed: 2025-03-23)

Jacquin, L., Molton, P., Deck, S., Maury, B., & Soulevant, D. (2009). Experimental study of shock oscillation over a transonic supercritical profile. AIAA Journal, 47 (9), 1985–1994. Retrieved from https://doi.org/10.2514/1.30190 doi: 10.2514/1.30190

Kalitzin, G., Medic, G., Iaccarino, G., & Durbin, P. (2005). Near-wall behavior of RANS turbulence models and implications for wall functions. *Journal of Computational Physics, 204*, 265–291.

Kato, M., & Launder, B. E. (1993). The modelling of turbulent flow around stationary and vibrating square cylinders. In Proceedings of the 9th symposium on turbulent shear flows. Kyoto, Japan.

Knoll, D., & Keyes, D. (2004). Jacobian-free Newton–Krylov methods: a survey of approaches and applications. Journal of Computational Physics, 193 (2), 357-397.

Kok, J. C. (2000). Resolving the dependence on freestream values for the *k–ω* turbulence model. AIAA Journal , 38 (7), 1292–1295. doi: 10.2514/2.1101.

Langer, S., & Swanson, R. C. (2024). Implementation, realization and an effective solver of two-equation turbulence models. Journal of Scientific Computing, 98 (8). doi: 10.1007/s10915-023-02394-0

Langtry, R. B., & Menter, F. R. (2009). Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. AIAA Journal , 47 (12), 2894–2906. Retrieved from https://doi.org/10.2514/1.42362 doi: 10.2514/1.42362

Lian, C., Xia, G., & Merkle, C. L. (2010). Impact of source terms on reliability of CFD algorithms. Computers & Fluids, 39, 1909–1922. https://doi.org/10.1016/j.compfluid.2010.06.021

Löhner, R. (1995). Feature-based error estimation and adaptive mesh refinement for transient compress- ible flows. Computer Methods in Applied Mechanics and Engineering, 119 (1–2), 123–139

Loseille, A. (2014). Metric-orthogonal anisotropic mesh generation. Procedia Engineering, 163 , 403–415

Mavriplis, D. (1995). Adaptive unstructured mesh techniques for the euler equations. AIAA Journal , 33 (4), 549–556. doi: 10.2514/3.12615

Menter, F. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. AIAA Journal , 32 (8), 1598-1605. Retrieved from https://doi.org/10.2514/3.12149 doi: 10.2514/3.12149

Merci, B., Steelant, J., Vierendeels, J., Riemslagh, K., & Dick, E. (2000). Computational treatment of source terms in two-equation turbulence models. AIAA Journal , 38 (11), 2085-2093. Retrieved from https://doi.org/10.2514/2.870 doi: 10.2514/2.870

Moryossef, Y., & Levy, Y. (2006). Unconditionally positive implicit procedure for two-equation turbulence models: Application to k–ω models. Journal of Computational Physics, 220, 88–108. doi:10.1016/j.jcp.2006.05.001

NASA. (2025). Nasa refine: Mesh adaptation library. https://github.com/nasa/refine. (Accessed: 2025-03-24)

NASA Langley Research Center. (n.d.). NASA Turbulence Modeling Resource. https://turbmodels.larc.nasa.gov/. (Accessed: 2025-03-23)

Park, M. A., Loseille, A., Krakos, J., Michal, T. R., & Alonso, J. J. (2016). Unstructured grid adaptation: Status, potential impacts, and recommended investments towards cfd 2030. In 46th aiaa fluid dynamics conference. American Institute of Aeronautics and Astronautics. Retrieved from http://dx.doi.org/10.2514/6.2016-3323 doi: 10.2514/6.2016-3323

Powell, K., & Roe, P. (1993). Adaptive–multigrid solution of the euler equations using a solution-based error indicator. In 11th aiaa computational fluid dynamics conference

Roe, P. L. (1982). Approximate riemann solvers, parameter vectors, and difference schemes. Journal of Computational Physics, 43 (2), 357–372

Roy, C. (2010). Review of discretization error estimators in scientific computing. In 48th AIAA aerospace sciences meeting.

Sinha, K., & Candler, G. V. (1998). Convergence improvement of two-equation turbulence model calculations (AIAA Paper 1998-2649). Paper presented at the 29th AIAA Fluid Dynamics Conference, Albuquerque, NM, United States. https://doi.org/10.2514/6.1998-2649

Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., & Mavriplis, D. (2014). CFD Vision 2030 Study: A path to revolutionary computational aerosciences (Tech. Rep. No. NASA/CR-2014-218178). NASA Langley Research Center. Retrieved from https://ntrs.nasa.gov/citations/20140003093

Smirnov, P. E., & Menter, F. R. (2009). Sensitization of the sst turbulence model to rotation and curvature by applying the spalart-shur correction term. ASME Journal of Turbomachinery, 131 (4), 041010. Retrieved from https://doi.org/10.1115/1.3070573 doi: 10.1115/1.3070573

Spalart, P. R., & Rumsey, C. L. (2007). Effective inflow conditions for turbulence models in aerodynamic calculations. AIAA Journal , 45 (10), 2544–2553. doi: 10.2514/1.29373

Suarez, G. M. (2020). Coupling strategies for solving the rans equations (Tech. Rep.). Institut für Aerodynamik und Strömungstechnik. Retrieved from https://elib.dlr.de/134238/

Sukas, H., & Sahin, M. (2025). A robust monolithic nonlinear newton method for the com- pressible reynolds averaged navier–stokes equations. Computers Fluids, 289, 106549. Retrieved from https://www.sciencedirect.com/science/article/pii/S0045793025000106 doi: https://doi.org/10.1016/j.compfluid.2025.106549

Swanson, R. C. (2021). Solving two-equation turbulence models: With a perspective on solving transport equations (NASA Technical Memorandum Nos. NASA/TM–20210016636). Hampton, VA: NASA Langley Research Center. Retrieved from https://ntrs.nasa.gov/citations/20210016636

Thompson, K. B., & O'Connell, M. (2019). Streamlined convergence acceleration for CFD codes. In AIAA Aviation 2019 Forum. Retrieved from https://arc.aiaa.org/doi/abs/10.2514/6.2019-3709 doi: 10.2514/6.2019-3709

Wang, Z.J. (2000). A fast nested multi-grid viscous flow solver for adaptive Cartesian/Quad grids. Int. J. Numer. Meth. Fluids, 33: 657-680. https://doi.org/10.1002/1097-0363(20000715)33:5<657::AID-FLD24>3.0.CO;2-G

Warren, M. (1991). Solution-adaptive grid refinement in unsteady transonic flows. In 29th aiaa aerospace sciences meeting.

Wilcox, D. C. (1988). Reassessment of the scale-determining equation for advanced turbulence mod- els. AIAA Journal , 26 (11), 1299-1310. Retrieved from https://doi.org/10.2514/3.10041 doi: 10.2514/3.10041

Wilcox, D. C. (1993). Turbulence modelling for CFD. DCW Industries, La Canada

APPENDICES

APPENDIX A: Analytical Derivations for Source Terms of Non-Dimensional *k-ω* SST-m Model Equations

APPENDIX B: Integral Forms of Non-Dimensional *k-ω* SST-m Model Equations for Newton Method

APPENDIX A

The non-dimensionalized production and destruction terms in the k-equation are:

$$P_k = \frac{Ma}{Re} \cdot (\mu_t S^2)$$
$$D_k = \frac{Re}{Ma_\infty} \cdot (\beta^* \rho w k)$$

where S is the strain-rate magnitude and μ_t is the turbulent viscosity, given by:

$$\mu_t = \frac{\rho a_1 k}{X}$$

where:

$$X = max\left(a_1 w, \frac{Ma}{Re} \cdot \Omega \cdot F_2\right)$$

Derivatives of P_k:

Applying the chain rule:

$$\frac{\partial P_k}{\partial k} = \frac{Ma}{Re} \cdot S^2 \cdot \frac{\partial \mu_t}{\partial k}$$
$$\frac{\partial P_k}{\partial w} = \frac{Ma}{Re} \cdot S^2 \cdot \frac{\partial \mu_t}{\partial w}$$

Since $\mu_t = \frac{\rho a_1 k}{X}$, applying the quotient rule:

$$\frac{\partial \mu_t}{\partial k} = \frac{\rho a_1 X - \rho a_1 k \frac{\partial X}{\partial k}}{X^2}$$
$$\frac{\partial \mu_t}{\partial w} = \frac{0 - \rho a_1 k \frac{\partial X}{\partial w}}{X^2}$$

Case-1: $X = a_1 w$

If $a_1 w \geq \frac{Ma}{Re} \cdot \Omega \cdot F_2$:

$$X = a_1 w, \frac{\partial X}{\partial k} = 0, \frac{\partial X}{\partial w} = a_1$$

Thus:

$$\frac{\partial \mu_t}{\partial k} = \frac{\rho a_1}{X}$$
$$\frac{\partial \mu_t}{\partial w} = \frac{0 - \rho a_1 k a_1}{X^2}$$

Case-2: $X = \frac{Ma}{Re} \cdot S \cdot F_2$:

If $a_1 w < \frac{Ma}{Re} \cdot \Omega \cdot F_2$:

$$X = \frac{Ma}{Re} \cdot \Omega \cdot F_2$$

Applying the chain rule:

$$\frac{\partial X}{\partial k} = \frac{Ma}{Re} \cdot \Omega \cdot \frac{\partial F_2}{\partial k}$$
$$\frac{\partial X}{\partial w} = \frac{Ma}{Re} \cdot \Omega \cdot \frac{\partial F_2}{\partial w}$$

Derivative of F_2 :

$$F_2 = tanh(arg2^2)$$

where:

$$\frac{\partial F_2}{\partial k} = \frac{\partial F_2}{\partial arg\,2} \cdot \frac{\partial arg2}{\partial k}$$
$$\frac{\partial F_2}{\partial w} = \frac{\partial F_2}{\partial arg\,2} \cdot \frac{\partial arg2}{\partial w}$$
$$arg2 = max(2G_3, G_1)$$
$$G_3 = \frac{\frac{Ma}{Re} \cdot \sqrt{k}}{\beta^* w d}$$
$$G_1 = \frac{\left(\frac{Ma}{Re}\right)^2 \cdot 500 \cdot \nu}{d^2 w}$$

Explicit derivatives of G_1 and G_3 :

$$\frac{\partial G_3}{\partial k} = \frac{\frac{Ma}{Re}}{\beta^* w d} \cdot \frac{1}{2\sqrt{k}}$$

$$\frac{\partial G_3}{\partial w} = -\frac{\frac{Ma}{Re} \cdot \sqrt{k}}{\beta^* d w^2}$$

$$\frac{\partial G_1}{\partial k} = 0$$

$$\frac{\partial G_1}{\partial w} = -\frac{\left(\frac{Ma}{Re}\right)^2 \cdot 500 \cdot \nu}{d^2 w^2}$$

Applying the chain rule, if $2G_3 \geq G_1$, then:

$$\frac{\partial arg2}{\partial k} = 2 \cdot \frac{\partial G_3}{\partial k},$$

$$\frac{\partial arg2}{\partial w} = 2 \cdot \frac{\partial G_3}{\partial w}.$$

Otherwise:

$$\frac{\partial arg2}{\partial k} = 0, \frac{\partial arg2}{\partial w} = \frac{\partial G_1}{\partial w}$$

Using the *sech-tanh* identity:

$$\frac{d}{dx} tanh(x^2) = 2x \cdot sech^2(x^2) = 2x \cdot \left(1 - tanh^2(x^2)\right)$$

Thus:

$$\frac{\partial F_2}{\partial arg2} = 2 \cdot arg2 \cdot \left(1 - tanh^2(arg2^2)\right)$$

Derivatives of D_k:

$$D_k = \frac{Re}{Ma_\infty} \cdot (\beta^* \rho w k)$$

Applying the product rule:

$$\frac{\partial D_k}{\partial k} = \frac{Re}{Ma_\infty} \cdot \beta^* \rho w$$

$$\frac{\partial D_k}{\partial w} = \frac{Re}{Ma_\infty} \cdot \beta^* \rho k$$

APPENDIX B

If $\left(\frac{M_\infty}{Re_\infty}\mu_t S^2 < \frac{Re}{M_\infty}20\beta^*\rho\omega k\right)$:

***k*-equation:**

$$\begin{aligned}
&\iiint_\Omega \frac{\Delta(\rho k)}{\Delta t}dV + \iint_{\partial\Omega} [\boldsymbol{n}\cdot(\rho\boldsymbol{u}\Delta k) + \boldsymbol{n}\cdot(\rho k\Delta\boldsymbol{u}) + \boldsymbol{n}\cdot(\boldsymbol{u}k\Delta\rho)]dS \\
&\quad -\frac{M_\infty}{Re}\iiint_\Omega [S\Delta\mu_t + 2\mu_t S\Delta S]dV \\
&\quad +\frac{Re}{M_\infty}\iiint_\Omega [\beta^*\rho\omega\Delta k + \beta^*\rho k\Delta\omega + \beta^* k\omega\Delta\rho]dV \\
&\quad -\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla k[(\Delta\mu + \sigma_k\mu_t) + (\mu + \sigma_k\Delta\mu_t) + (\Delta\mu + \sigma_k\Delta\mu_t) \\
&\quad +(\mu + \mu_t\Delta\sigma_k) + (\Delta\mu + \mu_t\Delta\sigma_k)]dS \\
&\quad -\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla(\Delta k)(\mu + \sigma_k\mu_t)dS \\
&\quad = -\iint_{\partial\Omega} \boldsymbol{n}\cdot(\rho\boldsymbol{u}k)dS + \frac{M_\infty}{Re}\iiint_\Omega \mu_t S^2 dV - \frac{Re}{M_\infty}\iiint_\Omega \beta^*\rho\omega k dV \\
&\quad +\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla k(\mu + \sigma_k\mu_t)dS
\end{aligned}$$

***ω*-equation:**

$$\begin{aligned}
&\iiint_\Omega \frac{\Delta(\rho\omega)}{\Delta t}dV + \iint_{\partial\Omega} [\boldsymbol{n}\cdot(\rho\boldsymbol{u}\Delta\omega) + \boldsymbol{n}\cdot(\rho\omega\Delta\boldsymbol{u}) + \boldsymbol{n}\cdot(\boldsymbol{u}\omega\Delta\rho)]dS \\
&\quad -\frac{M_\infty}{Re}\iiint_\Omega [\gamma S^2\Delta\rho + 2\gamma\rho S\Delta S + \rho S^2\Delta\gamma]dV \\
&\quad +\frac{Re}{M_\infty}\iiint_\Omega [\beta\omega^2\Delta\rho + 2\beta\rho\omega\Delta\omega + \rho\omega^2\Delta\beta]dV \\
&\quad -\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla\omega[(\Delta\mu + \sigma_\omega\mu_t) + (\mu + \sigma_\omega\Delta\mu_t) + (\Delta\mu + \sigma_\omega\Delta\mu_t) \\
&\quad +\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla(\Delta\omega)(\mu + \sigma_\omega\mu_t)dS \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-\Delta F_1)\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega dV \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\Delta\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega dV \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\rho\sigma_{\omega 2}}{\omega}\nabla(\Delta k)\cdot\nabla\omega dV \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla(\Delta\omega)dV \\
&\quad = -\iint_{\partial\Omega} \boldsymbol{n}\cdot(\rho\boldsymbol{u}\omega)dS + \frac{M_\infty}{Re}\iiint_\Omega \gamma\rho S^2 dV - \frac{Re}{M_\infty}\iiint_\Omega \beta\rho\omega^2 dV \\
&\quad +\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla\omega(\mu + \sigma_\omega\mu_t)dS + \frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega dV
\end{aligned}$$

Else if $\left(\frac{Re}{M_\infty}20\boldsymbol{\beta}^*\boldsymbol{\rho\omega k} < \frac{M_\infty}{Re_\infty}\boldsymbol{\mu_t S^2}\right)$:

***k*-equation**

$$\begin{aligned}
&\iiint_\Omega \frac{\Delta(\rho k)}{\Delta t}dV + \iint_{\partial\Omega} [\boldsymbol{n}\cdot(\rho\boldsymbol{u}\Delta k) + \boldsymbol{n}\cdot(\rho k\Delta\boldsymbol{u}) + \boldsymbol{n}\cdot(\boldsymbol{u}k\Delta\rho)]dS \\
&\quad -\frac{Re}{M_\infty}20\beta^*\iiint_\Omega [\rho\omega\Delta k + \rho k\Delta\omega + k\omega\Delta\rho]dV \\
&\quad +\frac{Re}{M_\infty}\beta^*\iiint_\Omega [\rho\omega\Delta k + \rho k\Delta\omega + k\omega\Delta\rho]dV \\
&\quad -\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla k[(\Delta\mu + \sigma_k\mu_t) + (\mu + \sigma_k\Delta\mu_t) + (\Delta\mu + \sigma_k\Delta\mu_t) \\
&\quad +(\mu + \mu_t\Delta\sigma_k) + (\Delta\mu + \mu_t\Delta\sigma_k)]dS \\
&\quad -\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla(\Delta k)(\mu + \sigma_k\mu_t)dS \\
&\quad = -\iint_{\partial\Omega} \boldsymbol{n}\cdot(\rho\boldsymbol{u}k)dS + \frac{Re}{M_\infty}20\beta^*\iiint_\Omega \rho\omega k dV - \frac{Re}{M_\infty}\beta^*\iiint_\Omega \rho\omega k dV \\
&\quad +\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla k(\mu + \sigma_k\mu_t)dS
\end{aligned}$$

***ω*-equation**

$$\begin{aligned}
&\iiint_\Omega \frac{\Delta(\rho\omega)}{\Delta t}dV + \iint_{\partial\Omega} [\boldsymbol{n}\cdot(\rho\boldsymbol{u}\Delta\omega) + \boldsymbol{n}\cdot(\rho\omega\Delta\boldsymbol{u}) + \boldsymbol{n}\cdot(\boldsymbol{u}\omega\Delta\rho)]dS \\
&\quad -\frac{M_\infty}{Re}\iiint_\Omega [\gamma S^2\Delta\rho + 2\gamma\rho S\Delta S + \rho S^2\Delta\gamma]dV \\
&\quad +\frac{Re}{M_\infty}\iiint_\Omega [\beta\omega^2\Delta\rho + 2\beta\rho\omega\Delta\omega + \rho\omega^2\Delta\beta]dV \\
&\quad -\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla\omega[(\Delta\mu + \sigma_\omega\mu_t) + (\mu + \sigma_\omega\Delta\mu_t) + (\Delta\mu + \sigma_\omega\Delta\mu_t) \\
&\quad +\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla(\Delta\omega)(\mu + \mu_t\Delta\sigma_\omega) + (\Delta\mu + \mu_t\Delta\sigma_\omega)\Big]dS \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-\Delta F_1)\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega dV \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\Delta\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega dV \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\rho\sigma_{\omega 2}}{\omega}\nabla(\Delta k)\cdot\nabla\omega dV \\
&\quad -\frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla(\Delta\omega)dV \\
&\quad = -\iint_{\partial\Omega} \boldsymbol{n}\cdot(\rho\boldsymbol{u}\omega)dS + \frac{M_\infty}{Re}\iiint_\Omega \gamma\rho S^2 dV - \frac{Re}{M_\infty}\iiint_\Omega \beta\rho\omega^2 dV \\
&\quad +\frac{M_\infty}{Re}\iint_{\partial\Omega} \boldsymbol{n}\cdot\nabla\omega(\mu + \sigma_\omega\mu_t)dS + \frac{M_\infty}{Re_\infty}\iiint_\Omega \frac{2(1-F_1)\rho\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega dV.
\end{aligned}$$

CURRICULUM VITAE

Name Surname: Atakan OKAN

EDUCATION:

- **B.Sc.:** 2020, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Astronautical Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2021 - Present: Computational Fluid Dynamics Engineer at Turkish Aerospace, Computational Mechanics Department.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Okan, A.** and Şahin, M. 2025: Fully-Coupled k–ω-SST Turbulence Model Implementation for Nonlinear Newton Method in Unstructured Vertex-Based Hemlab Algorithm. 4th International Graduate Research Symposium: IGRS'25, May 12-14, 2025 Istanbul, Turkey.