POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Implementation and Validation of a Transitional Model in ADflow for Aerodynamic Optimization

Supervisors

Prof. Domenic D'AMBROSIO

Prof. Luís EÇA

Prof. Marcello RIGHI

Candidate

Marcello ZANI

July 2025

Implementation and Validation of a Transitional Model in ADflow for Aerodynamic Optimization

Marcello Zani

Abstract

Transitional flows play a critical role in many low- to moderate-Reynolds-number aerodynamic applications, including unmanned aerial vehicles, gliders, and wind turbines. Traditional turbulence models often fail to capture laminar-to-turbulent transition accurately, leading to erroneous predictions of boundary-layer behavior and aerodynamic performance. To address this limitation, this work focuses on the integration of the Langtry–Menter k–ω–γ–$\mathrm{Re}_{\theta_\mathrm{t}}$ transition model into the high-fidelity RANS solver `ADflow`, with support for gradient-based design optimization via algorithmic differentiation.

The implementation enables the simulation of transitional flows using a local, correlation-based approach suitable for complex geometries and optimization tasks. Although the gradient verification for the transitional model remains incomplete, initial results show promise. Deviations in sensitivity accuracy and prediction fidelity indicate areas for further development and validation.

To demonstrate the framework's optimization capability, a shape optimization study was conducted on a supercritical airfoil using the Spalart–Allmaras model. The optimization successfully improved aerodynamic performance under constraints, showcasing the practical utility of `ADflow`'s gradient-based infrastructure. This study lays the groundwork for future efforts in adjoint-based transitional flow optimization and highlights the importance of transition modeling in modern aerodynamic design.

ACKNOWLEDGMENTS

The author sincerely thanks Professor Marcello Righi for his invaluable guidance that laid the groundwork for this research nearly a year ago. Special thanks also go to research assistant David Anderegg for his patience in explaining the code infrastructure and for his essential technical support—both affiliated with the *Zürcher Hochschule für Angewandte Wissenschaften (ZHAW)*.

Gratitude is extended to Professor Domenic D'Ambrosio and Ph.D. Manuel Carreño Ruiz (both from *Politecnico di Torino*) for their assistance with simulations and insightful technical advice, and to Professor Luís Eça (*Instituto Superior Técnico, Universidade de Lisboa*) for his suggestions in the code development.

Finally, the author acknowledges Professor Joaquim R. R. A. Martins for providing access to the MDOlab packages `ADflow` and `MACH-Aero`, as well as Postdoctoral Researcher Anil Yildirim (both at the *University of Michigan*) for his helpful suggestions during the implementation.

UMich MDO Lab	**ZHAW**	**PoliTo**	**IST**
Michigan, USA	Switzerland	Italy	Portugal

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol	Description	SI Unit
α	Angle of attack	°
β	Sideslip angle	°
$\mathrm{C_f}$	Skin friction coefficient, $\tau_\mathrm{w}/(0.5\rho \mathrm{U}_\mathrm{ref}^2)$	–
d	Distance to nearest wall	m
δ	Boundary-layer thickness	m
ε	Turbulence dissipation rate	$\mathrm{m^2/s^3}$
k	Turbulent kinetic energy	$\mathrm{m^2/s^2}$
λ_θ	Pressure gradient parameter, $(\rho\theta^2/\mu)(\mathrm{dU/ds})$	1/m
μ	Dynamic viscosity	kg/(m s)
μ_t	Turbulent (eddy) viscosity	kg/(m s)
ν	Kinematic viscosity, μ/ρ	$\mathrm{m^2/s}$
ν_t	Turbulent kinematic viscosity	$\mathrm{m^2/s}$
p	Flow pressure	Pa
P	Mean flow pressure	Pa
$\mathrm{p'}$	Fluctuating flow pressure	Pa
Re_θ	Momentum-thickness Reynolds number, $\rho\theta \mathrm{U}_0/\mu$	–
$\mathrm{Re}_{\theta_\mathrm{t}}^{\mathrm{eq}}$	Equilibrium transition onset momentum-thickness Reynolds number (from freestream conditions), $\rho\theta_\mathrm{t}\mathrm{U}_0/\mu$	–
$\hat{\mathrm{Re}}_{\theta_\mathrm{t}}$	Transition onset momentum-thickness Reynolds number (from transport equation)	–
ρ	Flow density	$\mathrm{kg/m^3}$
S	Strain rate magnitude, $\sqrt{2\mathrm{S_{ij}S_{ij}}}$	1/s
$\mathrm{S_{ij}}$	Strain-rate tensor, $0.5(\partial \mathrm{u_i}/\partial \mathrm{x_j} + \partial \mathrm{u_j}/\partial \mathrm{x_i})$	1/s
τ_w	Wall shear stress, $\mu(\partial \mathrm{U}/\partial \mathrm{y})_\mathrm{w}$	$\mathrm{kg/(m\,s^2)}$
θ	Momentum thickness	m
T	Flow temperature	K
Tu	Turbulence intensity, $100 \times \sqrt{2\mathrm{k}/3}/\mathrm{U}$	%
TI_∞	Freestream turbulence intensity, $100 \times \sqrt{2\mathrm{k}/3}/\mathrm{U}_\infty$	%

Symbol	Description	SI Unit
u	Flow velocity	m/s
U	Mean flow velocity	m/s
U_∞	Freestream flow velocity	m/s
u'	Fluctuating flow velocity	m/s
$\mathrm{x/c}$	Normalized axial coordinate	–
y^+	Wall coordinate, $\rho \mathrm{y} \mathrm{u}_\tau/\mu$	–
ω	Specific turbulence dissipation rate	1/s
Ω	Vorticity magnitude, $\sqrt{2\Omega_{\mathrm{ij}}\Omega_{\mathrm{ij}}}$	1/s
Ω_{ij}	Vorticity tensor, $0.5(\partial \mathrm{U}_\mathrm{i}/\partial \mathrm{x}_\mathrm{j} - \partial \mathrm{U}_\mathrm{j}/\partial \mathrm{x}_\mathrm{i})$	1/s

Chapter 1

Introduction

1.1 Motivation

The accurate prediction of transitional flows remains a key challenge in Computational Fluid Dynamics [1]. While high Reynolds number applications often exhibit minimal laminar regions, model-scale tests and many practical systems—such as UAVs [2], gliders [3], small submarines [4], and wind turbines [5]—operate at moderate or low Reynolds numbers, where transition is significant. In such regimes, traditional Reynolds-Averaged Navier-Stokes (RANS) models often mispredict early transition, especially near stagnation points, due to their fully turbulent assumptions.

In some contexts, e.g., marine propeller testing [6] or laminar flow wing design [7], transition prediction becomes critical, particularly when flow tripping is absent or laminar flow is desired. Consequently, transition-sensitive models have been developed to extend RANS applicability. However, these models are still under refinement and lack the maturity of conventional turbulence models.

Transition modeling has gained traction through efforts like the NASA Transition Modeling Workshop [8], and the AIAA Transition Modeling Workshops [9].

Standard turbulence models (k–ϵ, k–ω, Spalart–Allmaras) are inadequate for transition due to calibration against fully turbulent or isotropic turbulence data [10, 11]. For example, the k–ω model predicts transition at $\mathrm{Re} = 8100$ [12], resulting in a negligible laminar flow region for $\mathrm{Re} < 10^5$. Early transition prediction and sensitivity to tripping or initial conditions limit their reliability [13].

Transition is triggered by mechanisms like freestream turbulence, roughness, separation, wakes, and crossflow [14, 15], requiring models that incorporate instability physics. High-fidelity methods DNS offers accuracy but are too costly for routine use [16], while Large Eddy Simulation (LES) accuracy depends significantly on the subgrid-scale (SGS) model employed [17]. The e^{N} method accurately predicts natural transition [18]; however, its empirical basis and non-local nature hinder integration into RANS solvers and HPC frameworks employing parallel domain decomposition, where non-local computations are costly and difficult to implement efficiently [19].

Low-Re models with damping functions show limited success [20], and some k–ϵ variants suffer from instability [21]. Correlation-based models address these gaps. The γ–Re_θ model [22] is widely adopted for its locality and generality. Others, like k–$\mathrm{k_L}$–ω [23] and the Amplification Factor

Transport model [24], replicate e^N capabilities within RANS.

Several studies have employed the γ–Re_θ transition model for aerodynamic shape optimization. Khayatzadeh and Nadarajah [25, 26] applied this model in the adjoint-based optimization of the S809 and the natural laminar flow NLF(1)-0416 airfoils at low Reynolds numbers. Further applications of the Langtry–Menter model have been demonstrated in optimization frameworks using the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) as the search algorithm, with Kriging-based surrogate models introduced to enhance the efficiency of the optimization process [27, 28]. Robitaille et al. [29] relied on first-order finite-difference approximations for optimization, which tend to be computationally expensive. To address this limitation, Rashad and Zingg [30] developed an adjoint-based methodology using a two-dimensional RANS solver in combination with either a simplified e^N method or the Arnal–Habiballah–Delcourt (AHD) transition criterion [31], enabling efficient optimization of laminar flow airfoils. Halila et al. [32] developed a modified version of the original Amplification Factor Transport (AFT) model, termed AFT-S, for adjoint aerodynamic shape optimization using the RANS-based solver `ADflow` with derivatives calculated using algorithmic differentiation. Their fully integrated approach embeds transition modeling directly within the flow solver, incorporates the transition transport equations into the adjoint system, and removes the need for external transition modules. The optimization campaign spans a broad range of flow conditions—including subsonic and transonic regimes—and considers both fully turbulent and transitional flows. The results demonstrate clear aerodynamic benefits from including transition modeling. Moreover, the robustness of the optimized airfoils was assessed under early-transition scenarios induced by freestream disturbances, showing that designs optimized with transition effects retained favorable performance even under degraded conditions. Never a 4 equation transition model has been implemented with derivatives calculated using forward-mode algorithmic differentiation.

1.2 Objectives

This work pursues three main objectives:

1. **Implementation of a Transitional Turbulence Model:** This entails incorporating a transitional turbulence model— the Langtry-Menter γ–Re_{θ_t} model—into ADflow to effectively simulate laminar-to-turbulent transition. The objective is to improve prediction accuracy and solution stability in flows where transition significantly influences aerodynamic behavior.

2. **AD Gradient Verification in Forward Mode:** A critical step is verifying the correctness of forward mode AD gradient calculations within ADflow when employing the transitional model in the flow solver. This ensures that the computed derivatives are consistent, reliable, and physically meaningful for subsequent optimization tasks.

3. **Model Validation:** Finally, the implemented model will be validated against established benchmark cases from the literature by comparing ADflow's predictions with experimental data or high-fidelity numerical simulations. This step evaluates the model's ability to accurately represent transition behavior.

To further demonstrate the capabilities of adjoint-based optimization with derivatives computed via algorithmic differentiation in `ADflow`, an aerodynamic shape optimization will be performed using the fully turbulent Spalart–Allmaras model on an airfoil configuration.

Chapter 2

Transition - Theory and Modelling

2.1 Mechanisms of Flow Transition

Transition from laminar to turbulent flow results from the amplification of instabilities, triggered by environmental factors such as freestream turbulence, surface roughness, and pressure gradients [14, 15]. Several transition pathways exist:

- **Natural transition**: Initiated by low-amplitude disturbances evolving into Tollmien-Schlichting (TS) waves and turbulent spots. This can be analyzed using linear stability theory (LST) [33, 34]. Decomposing the flow into a steady base state $\mathrm{U(y)}$ and a perturbation $\tilde{\mathrm{u}}(\mathrm{x,y,t})$ leads to the Orr-Sommerfeld equation [33, 35]:

  $$(\mathrm{U}-\mathrm{c})(\phi''-\alpha^2\phi)-\mathrm{U}''\phi=-\frac{\mathrm{i}}{\alpha\mathrm{Re}}(\phi''''-2\alpha^2\phi''+\alpha^4\phi),$$

  where $\phi(\mathrm{y})$ is the disturbance shape function and $\mathrm{c}=\omega/\alpha$ the complex phase speed. *Receptivity theory* investigates how external disturbances—acoustic waves, freestream turbulence, or surface imperfections—introduce initial perturbations into the boundary layer. While linear theory captures the early growth phase, the full transition process, including nonlinear interactions and secondary instabilities, requires advanced models such as the Parabolized Stability Equations (PSE) [36, 37].

- **Bypass transition**: In high-disturbance environments, TS waves are bypassed and turbulence forms directly, typical in turbomachinery flows [38].

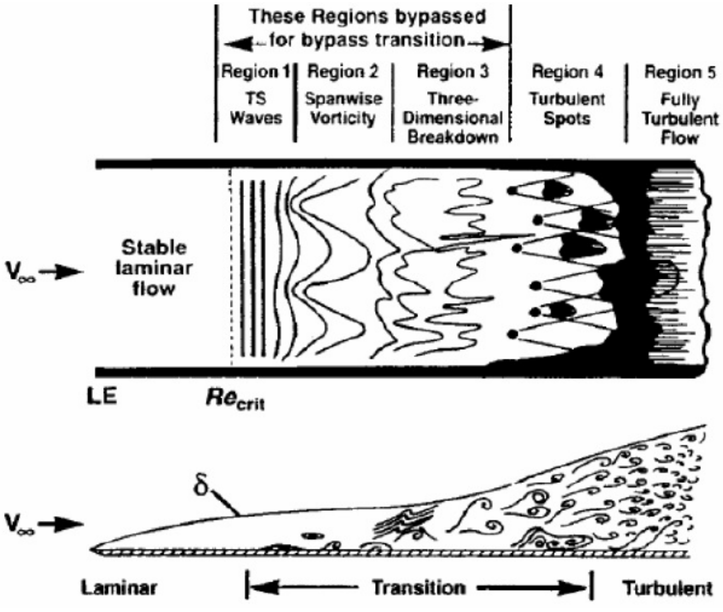

Figure 2.1: Illustration of natural and bypass transition mechanisms [33].

- **Crossflow transition**: Found in three-dimensional boundary layers (e.g., on swept wings), arising from crossflow velocity components and inviscid instabilities [39].

- **Separation-induced transition**: This mechanism occurs when a laminar boundary layer is subjected to an adverse pressure gradient and separates from the surface before becoming turbulent; the resulting free shear layer contains an inflection point in the velocity profile, making it highly susceptible to instability [40]. Transition can then occur through natural mechanisms, bypass transition, or Kelvin–Helmholtz instabilities (non linear mechanism [33])[40]. If the pressure gradient is not excessively strong, the turbulent shear layer may reattach, forming a laminar separation bubble (LSB) in a time-averaged sense. The size and impact of the LSB depend on Reynolds number, surface roughness, pressure gradient, and freestream turbulence intensity. Long bubbles—extending up to 20–30% of the chord—can induce a pressure plateau in the pressure distribution [41]. LSBs strongly affect airfoil performance, especially in UAV and wind turbine applications, through increased drag and modified lift and moment behavior [42].

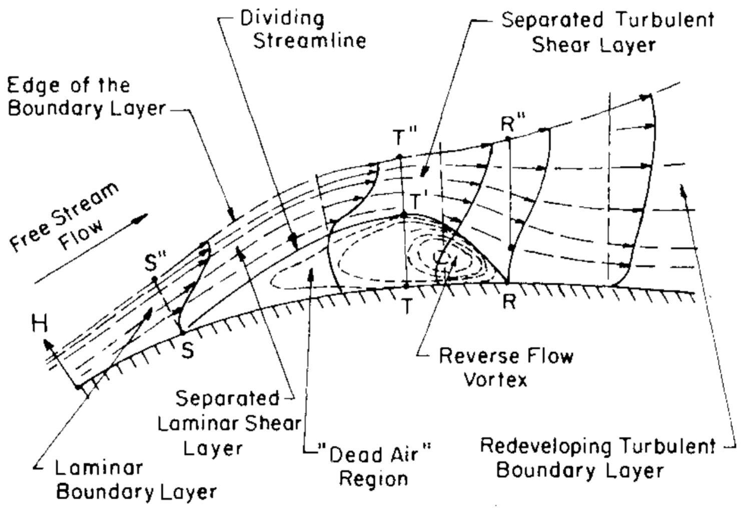
Dividing Streamline
Separated Turbulent Shear Layer
Edge of the Boundary Layer
Free Stream Flow
T"
R"
T'
S"
H
S
T
R
Reverse Flow Vortex
Separated Laminar Shear Layer
"Dead Air" Region
Redeveloping Turbulent Boundary Layer
Laminar Boundary Layer

Figure 2.2: Time-averaged flow visualization of a laminar separation bubble over an airfoil at low Reynolds number, illustrating separation, transition, and reattachment [40].

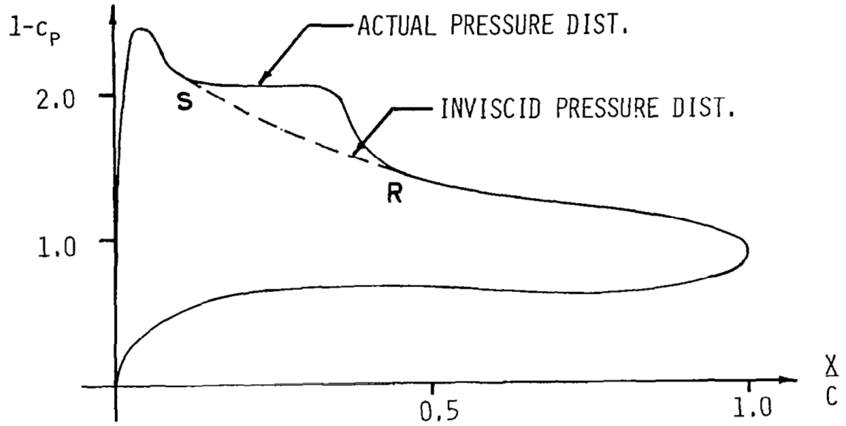

Figure 2.3: Pressure distribution over a "short" LSB [41].

2.2 Transition Modelling Techniques

Predicting laminar-to-turbulent transition is critical in high-fidelity aerodynamic simulations, as it significantly affects boundary layer development, separation, and skin-friction drag [43, 42]. Several modelling strategies are used, each balancing physical fidelity and computational cost.

- **Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES)**: DNS resolves the full range of spatial and temporal scales of turbulence by directly solving the Navier–Stokes equations without any turbulence model:

$$\frac{\partial \mathrm{u_i}}{\partial \mathrm{t}} + \mathrm{u_j}\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}} = -\frac{1}{\rho}\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}} + \nu \frac{\partial^2 \mathrm{u_i}}{\partial \mathrm{x_j^2}} \tag{2.1}$$

where $\mathrm{u_i}$ are the velocity components, p is pressure, ρ is density, and ν is kinematic viscosity. Large Eddy Simulation (LES) resolves the large scales of turbulence while modeling the smaller, more isotropic eddies through a subgrid-scale (SGS) model. Although LES offers higher accuracy than Reynolds-Averaged Navier-Stokes (RANS) methods, Direct Numerical Simulation (DNS) remains prohibitively expensive for high-Reynolds-number flows [44]. The accuracy of LES strongly depends on the choice of the SGS model [17].

- $\mathrm{e^N}$ **Method**: Based on linear stability theory (LST), the $\mathrm{e^N}$ method predicts the amplification of Tollmien–Schlichting (TS) waves in a laminar boundary layer. The transition onset is assumed to occur when the non-dimensional disturbance amplitude reaches a critical threshold:

$$\mathrm{A} = \mathrm{A_0 e^N} \tag{2.2}$$

where N is the amplification factor, $\mathrm{A_0}$ is the initial disturbance amplitude, and transition typically occurs when $\mathrm{N} \in [9, 11]$. This method is accurate in low-disturbance environments (e.g., wind tunnels) but not suited for general CFD applications due to its empirical basis and non-local nature [19].

- **Empirical Correlations**: These models use semi-empirical relations derived from experimental data to estimate transition onset. A well-known example is the Abu-Ghannam and Shaw correlation [45], which relates transition Reynolds number to turbulence intensity (Tu) and pressure gradient (λ):

$$\mathrm{Re}_{\theta,\mathrm{tr}} = \mathrm{f}(\mathrm{Tu}, \lambda) \tag{2.3}$$

where $\mathrm{Re}_\theta = \frac{\mathrm{U_e}\theta}{\nu}$ is the momentum-thickness Reynolds number.

- **Low-Reynolds-number (Low-**Re**) Turbulence Models**: These modify standard RANS models to explicitly resolve the viscous sublayer by introducing damping functions in turbulence models. For example, in the Wilcox k–ω model, near-wall damping is achieved by modifying the turbulent viscosity:

$$\nu_\mathrm{t} = \frac{\mathrm{k}}{\omega} \cdot \mathrm{f}_\mu(\mathrm{y}^+) \tag{2.4}$$

where k is turbulent kinetic energy, ω is specific dissipation rate, and f_μ is a damping function of the wall coordinate y^+. Models such as the k–ϵ can capture transition indirectly; however, their predictions are highly sensitive to initial conditions and the numerical solution procedure [21].

- **Transition Transport-Equation Models**: Transition prediction using transport equations represents a key advance in RANS-compatible methods. These models typically solve additional transport equations for intermittency and other transition-relevant quantities, enabling local formulations compatible with structured and unstructured grids [22].

The model proposed by Steelant and Dick [46] employs conditionally averaged Navier–Stokes equations and solves separate continuity and momentum equations for the laminar/transitional and turbulent flow regions. Intermittency is modeled using a transport equation guided by the intermittency distribution of Dhawan and Narasimha [47], while turbulence closure is achieved with a k–ϵ model.

Suzen and Huang [48] introduced a single intermittency transport equation that blends features from earlier models [49, 46]. Their approach multiplies the eddy viscosity from the k–ω SST model by intermittency, enabling accurate prediction of both streamwise and crossflow transition. Pecnik et al. [50] refined the Steelant and Dick model by incorporating a turbulence weighting factor that accounts for freestream turbulence intensity, resulting in improved transition prediction.

Menter et al.[51, 52] developed a local intermittency-based model coupled with the k–ω SST turbulence model, eliminating non-local dependencies. This formulation evolved into the γ–Re_θ model[22], which solves two transport equations: one for intermittency and another for the transition onset momentum-thickness Reynolds number, determined via empirical correlations. Although widely adopted, the model has limitations: it is not Galilean invariant, lacks surface roughness and crossflow sensitivity, and increases computational cost.

To overcome these drawbacks, Menter et al.[53] proposed the simplified γ model, replacing the Re_θ equation with an algebraic correlation. This version retains a local and robust formulation, is Galilean invariant, and can be coupled with both SST and Spalart–Allmaras turbulence models[54, 55, 56].

Alternative methods include the k–$\mathrm{k_L}$–ω models [23], which rely on laminar kinetic energy $\mathrm{k_L}$ to capture both bypass and natural transition without empirical correlations. These models use local transport equations and are fully RANS-compatible.

The Amplification Factor Transport (AFT) model [24] aims to replicate linear stability theory within a local RANS framework. Initially coupled with Spalart–Allmaras and later with SST [57], it estimates transition onset based on a critical amplification factor. While effective for natural transition, it performs less accurately in bypass-dominated regimes.

Among the various transition models, the most established are the γ–Reθ, γ, k–$\mathrm{k_L}$–ω, and AFT models, each tailored to specific transition mechanisms. The γ–Reθ model is widely used for general transition prediction, the simplified γ model offers improved robustness and locality, the AFT model targets natural transition, and the k–$\mathrm{k_L}$–ω model is effective for bypass transition [58]. This work focuses on models with local formulations, selecting the γ–Re_θ model as the representative of correlation-based methods.

Chapter 3

Governing Equations

3.1 Reynolds-Averaged Navier–Stokes (RANS) Framework

The Reynolds-Averaged Navier–Stokes (RANS) equations form a cornerstone in the modeling of turbulent flows, widely adopted in engineering due to their balance between computational tractability and predictive capability [12]. The nomenclature follows [58].

This study assumes a single-phase, incompressible, Newtonian fluid. Under these conditions, the governing equations for conservation of mass and momentum can be written as [59, 60]:

$$\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_i}} = 0, \tag{3.1}$$

$$\rho \frac{\partial \mathrm{u_i}}{\partial \mathrm{t}} + \rho \mathrm{u_j} \frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}} = -\frac{\partial \mathrm{p}}{\partial \mathrm{x_i}} + \frac{\partial}{\partial \mathrm{x_j}} \left[\mu \left(\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}} + \frac{\partial \mathrm{u_j}}{\partial \mathrm{x_i}} \right) \right], \tag{3.2}$$

where $\mathrm{u_i}$ denotes the instantaneous velocity components, p the pressure field relative to the hydrostatic baseline, ρ the fluid density, and μ the dynamic viscosity. The indices $\mathrm{i, j} = 1, 2, 3$ correspond to spatial Cartesian coordinates.

To capture turbulent fluctuations, each flow variable is decomposed into a mean and a fluctuating part via Reynolds decomposition:

$$\mathrm{u_i} = \mathrm{U_i} + \mathrm{u_i'}, \tag{3.3}$$

$$\mathrm{p} = \mathrm{P} + \mathrm{p'}, \tag{3.4}$$

where uppercase symbols ($\mathrm{U_i}$, P) represent time-averaged values, while primes indicate deviations from these means. For statistically stationary flows, the mean quantities are defined by long-time averaging:

$$\mathrm{U_i} = \overline{\mathrm{u_i}} = \lim_{\mathrm{T}\to\infty} \frac{1}{\mathrm{T}} \int_{\mathrm{t_0}}^{\mathrm{t_0+T}} \mathrm{u_i(t)}\, \mathrm{dt}. \tag{3.5}$$

Applying this averaging procedure to the instantaneous equations (3.1) and (3.2) and substituting the decomposed variables leads to the Reynolds-averaged equations:

$$\frac{\partial \mathrm{U_i}}{\partial \mathrm{x_i}} = 0, \tag{3.6}$$

$$\rho \frac{\partial \mathrm{U_i}}{\partial \mathrm{t}} + \rho \mathrm{U_j} \frac{\partial \mathrm{U_i}}{\partial \mathrm{x_j}} = -\frac{\partial \mathrm{P}}{\partial \mathrm{x_i}} + \frac{\partial}{\partial \mathrm{x_j}} \left[\mu \left(\frac{\partial \mathrm{U_i}}{\partial \mathrm{x_j}} + \frac{\partial \mathrm{U_j}}{\partial \mathrm{x_i}} \right) \right] - \frac{\partial}{\partial \mathrm{x_j}} \left(\rho \overline{\mathrm{u_i' u_j'}} \right). \tag{3.7}$$

Here, the Reynolds stress tensor $\overline{\mathrm{u_i' u_j'}}$ represents the momentum transfer caused by turbulent fluctuations. Since these new terms introduce additional unknowns, the system requires closure via turbulence models.

3.2 Turbulence Modeling

The Reynolds-Averaged Navier–Stokes (RANS) equations (Section 3.1) introduce the Reynolds stress tensor $\overline{\mathrm{u_i' u_j'}}$, which accounts for the effects of turbulent fluctuations on the mean flow. In three dimensions, this yields six additional unknowns, making the system underdetermined and necessitating closure models [60, 61].

Reynolds Stress Models (RSM) address this by solving transport equations for each stress component [62], capturing turbulence anisotropy with improved fidelity, but at significantly higher computational cost and complexity.

A more practical alternative are eddy viscosity models (EVM), which rely on the Boussinesq approximation to relate the Reynolds stresses to the mean strain rate:

$$-\rho \overline{\mathrm{u_i' u_j'}} = \mu_\mathrm{t} \left(\frac{\partial \mathrm{U_i}}{\partial \mathrm{x_j}} + \frac{\partial \mathrm{U_j}}{\partial \mathrm{x_i}} \right) - \frac{2}{3} \rho \mathrm{k} \delta_\mathrm{ij}, \tag{3.8}$$

where μ_t is the eddy viscosity, $\mathrm{k} = \frac{1}{2}\overline{\mathrm{u_i' u_i'}}$ the turbulent kinetic energy, and δ_ij the Kronecker delta. This approximation assumes isotropic turbulence and underpins most RANS models.

This formulation introduces μ_t and k as additional unknowns, which must be modeled to achieve closure.

3.3 The k–ω Shear Stress Transport (SST) Turbulence Model

The k–ω Shear Stress Transport (SST) model is a widely adopted two-equation turbulence model that combines the robustness of the k–ω model near walls with the free-stream insensitivity of the k–ϵ model. This hybrid strategy was developed to overcome the limitations inherent in using either formulation independently. Specifically, while the k–ω model performs well in resolving near-wall turbulence, it tends to be overly sensitive to free-stream values of ω. Conversely, the k–ϵ model is more stable in the free-stream but less accurate near solid boundaries. The SST model achieves a smooth transition between these two behaviors through the use of blending functions [63]. The version considered is the one of 2003, that follows [64].

The SST formulation is governed by two transport equations: one for the turbulent kinetic energy k, and another for the specific dissipation rate ω. These equations can be expressed as follows:

$$\frac{\partial(\rho \mathrm{k})}{\partial \mathrm{t}} + \frac{\partial(\rho \mathrm{U_j k})}{\partial \mathrm{x_j}} = \mathrm{P_k} - \mathrm{D_k} + \frac{\partial}{\partial \mathrm{x_j}}\left[(\mu + \sigma_\mathrm{k}\mu_\mathrm{t})\frac{\partial \mathrm{k}}{\partial \mathrm{x_j}}\right], \tag{3.9}$$

$$\begin{aligned}\frac{\partial(\rho\omega)}{\partial \mathrm{t}} + \frac{\partial(\rho \mathrm{U_j}\omega)}{\partial \mathrm{x_j}} &= \alpha\frac{\omega}{\mathrm{k}}\mathrm{P_k} - \rho\beta\omega^2 + \frac{\partial}{\partial \mathrm{x_j}}\left[(\mu + \sigma_\omega\mu_\mathrm{t})\frac{\partial\omega}{\partial \mathrm{x_j}}\right] \\ &+ 2(1-\mathrm{F_1})\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial \mathrm{k}}{\partial \mathrm{x_j}}\frac{\partial\omega}{\partial \mathrm{x_j}}.\end{aligned} \tag{3.10}$$

In these expressions, $\mathrm{P_k}$ denotes the production of turbulent kinetic energy, while $\mathrm{D_k}$ represents its dissipation. To prevent excessive turbulence production in regions of adverse pressure gradients or stagnation, the production term is limited:

$$\mathrm{P_k} = \min\left(\tau_\mathrm{ij}\frac{\partial \mathrm{U_i}}{\partial \mathrm{x_j}}, 10\beta^*\rho\omega\mathrm{k}\right). \tag{3.11}$$

The dissipation term $\mathrm{D_k}$ is modeled as:

$$\mathrm{D_k} = \rho\beta^*\mathrm{k}\omega. \tag{3.12}$$

The cross-diffusion term in Eq. (3.10) accounts for the transformation of the ϵ-based k–ϵ model into the ω-based framework and is activated by a blending function $\mathrm{F_1}$, which facilitates the transition between near-wall and free-stream modeling regions. This function is defined as:

$$\mathrm{F_1} = \tanh\left(\mathrm{arg}_1^4\right), \tag{3.13}$$

$$\mathrm{arg}_1 = \min\left\{\max\left(\frac{\sqrt{\mathrm{k}}}{\beta^*\omega \mathrm{d}}, \frac{500\nu}{\mathrm{d}^2\omega}\right), \frac{4\rho\sigma_{\omega 2}\mathrm{k}}{\mathrm{CD_{k\omega}d^2}}\right\}, \tag{3.14}$$

$$\mathrm{CD_{k\omega}} = \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial \mathrm{k}}{\partial \mathrm{x_j}}\frac{\partial\omega}{\partial \mathrm{x_j}}, 10^{-10}\right), \tag{3.15}$$

where d is the distance to the nearest wall and ν is the kinematic viscosity. The blending mechanism also modulates the model constants. Any such constant ϕ is computed as:

$$\phi = \mathrm{F_1}\phi_1 + (1-\mathrm{F_1})\phi_2, \tag{3.16}$$

allowing for a smooth interpolation between the two regimes.

The eddy viscosity, which is central to closing the RANS equations, is evaluated using a limiting formulation designed to suppress turbulence in regions of high strain rate anisotropy:

$$\mu_\mathrm{t} = \frac{\rho \mathrm{a}_1 \mathrm{k}}{\max\left(\mathrm{a}_1 \omega, \mathrm{S}\mathrm{F}_2\right)}, \tag{3.17}$$

where S is an invariant measure of the strain rate tensor and F_2 is a second blending function given by:

$$\mathrm{F}_2 = \tanh\left(\mathrm{arg}_2^2\right), \tag{3.18}$$

$$\mathrm{arg}_2 = \max\left(\frac{2\sqrt{\mathrm{k}}}{\beta^* \omega \mathrm{d}}, \frac{500\nu}{\mathrm{d}^2 \omega}\right). \tag{3.19}$$

The SST model constants are listed in Table 3.1, corresponding to the 2003 refinement of the model.

Table 3.1: Model constants for the 2003 version of the k–ω SST model [65].

α_1	α_2	$\sigma_{\mathrm{k}1}$	$\sigma_{\mathrm{k}2}$	$\sigma_{\omega 1}$	$\sigma_{\omega 2}$	β_1	β_2	β^*	κ	a_1
5/9	0.44	0.85	1.0	0.5	0.856	0.075	0.0828	0.09	0.41	0.31

3.4 The k–ω–γ–Re_{θ_t} Langtry–Menter Four-Equation Transitional SST Model

The k–ω–γ–Re_{θ_t} Langtry–Menter transitional shear stress transport (SST) turbulence model is coupled with the SST model, presented in Section 3.3. The nomenclature follows [66]. The 2009 version is considered.

3.4.1 Model Formulation of SST2003-LM2009

The Langtry–Menter SST transitional model extends the classical two-equation SST framework by adding two transport equations for the intermittency variable γ and the transition onset momentum thickness Reynolds number $\mathrm{Re}_{\theta t}$ [22]. The intermittency γ is a scalar field that controls the production of turbulence, where $\gamma = 0$ corresponds to laminar flow, $\gamma = 1$ to fully turbulent flow, and intermediate values indicate transitional regions. This enables prediction of laminar-to-turbulent transition by modeling the onset and extent of intermittency in the boundary layer.

The resulting four-equation system couples these new equations with the standard turbulent kinetic energy k and specific dissipation rate ω transport equations. The intermittency γ controls turbulence production, delaying it in laminar regions and activating it during transition.

The governing transport equations are:

$$\frac{\partial(\rho \mathrm{k})}{\partial \mathrm{t}} + \frac{\partial(\rho \mathrm{U_j k})}{\partial \mathrm{x_j}} = \hat{\mathrm{P}}_\mathrm{k} - \hat{\mathrm{D}}_\mathrm{k} + \frac{\partial}{\partial \mathrm{x_j}}\left[(\mu + \sigma_\mathrm{k}\mu_\mathrm{t})\frac{\partial \mathrm{k}}{\partial \mathrm{x_j}}\right] \tag{3.20}$$

$$\frac{\partial(\rho \omega)}{\partial \mathrm{t}} + \frac{\partial(\rho \mathrm{U_j}\omega)}{\partial \mathrm{x_j}} = \alpha\frac{\omega}{\mathrm{k}}\mathrm{P_k} - \rho\beta\omega^2 + \frac{\partial}{\partial \mathrm{x_j}}\left[(\mu + \sigma_\omega\mu_\mathrm{t})\frac{\partial \omega}{\partial \mathrm{x_j}}\right] + 2(1-\mathrm{F_1})\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial \mathrm{k}}{\partial \mathrm{x_j}}\frac{\partial \omega}{\partial \mathrm{x_j}} \tag{3.21}$$

$$\frac{\partial(\rho \gamma)}{\partial \mathrm{t}} + \frac{\partial(\rho \mathrm{U_j}\gamma)}{\partial \mathrm{x_j}} = \mathrm{P}_\gamma - \mathrm{E}_\gamma + \frac{\partial}{\partial \mathrm{x_j}}\left[\left(\mu + \frac{\mu_\mathrm{t}}{\sigma_\mathrm{f}}\right)\frac{\partial \gamma}{\partial \mathrm{x_j}}\right] \tag{3.22}$$

$$\frac{\partial(\rho \hat{\mathrm{Re}}_{\theta\mathrm{t}})}{\partial \mathrm{t}} + \frac{\partial(\rho \mathrm{U_j}\hat{\mathrm{Re}}_{\theta\mathrm{t}})}{\partial \mathrm{x_j}} = \mathrm{P}_{\theta\mathrm{t}} + \frac{\partial}{\partial \mathrm{x_j}}\left[\sigma_{\theta\mathrm{t}}\left(\mu + \mu_\mathrm{t}\right)\frac{\partial \hat{\mathrm{Re}}_{\theta\mathrm{t}}}{\partial \mathrm{x_j}}\right] \tag{3.23}$$

These equations are expressed in conservative form, consistent with standard turbulence modeling practice [63, 12].

3.4.2 Source Terms for the γ Equation

The source terms in the γ-equation govern the evolution of intermittency in the transition model. These include the production term P_γ, which initiates the transition process, and the destruction or relaminarization term E_γ, which suppresses turbulence in laminar regions. They are defined as follows:

$$
\begin{aligned}
&\mathrm{P}_{\gamma} = \mathrm{F}_{\mathrm{length}}\, \mathrm{c}_{\mathrm{a1}}\, \rho \mathrm{S} \left(\gamma \mathrm{F}_{\mathrm{onset}}\right)^{0.5} \left(1 - \mathrm{c}_{\mathrm{e1}}\gamma\right), \\
&\mathrm{E}_{\gamma} = \mathrm{c}_{\mathrm{a2}}\, \rho\, \Omega\, \gamma\, \mathrm{F}_{\mathrm{turb}} \left(\mathrm{c}_{\mathrm{e2}}\gamma - 1\right).
\end{aligned}
\tag{3.24}
$$

The source terms are augmented by several auxiliary functions, which are detailed below.

The critical momentum thickness Reynolds number, at which the flow begins instability [67], $\mathrm{Re}_{\theta\mathrm{c}}$, is obtained from an empirical correlation based on the transition momentum thickness Reynolds number $\hat{\mathrm{Re}}_{\theta\mathrm{t}}$:

$$
\mathrm{Re}_{\theta\mathrm{c}} = \begin{cases}
-3.96035 + 1.01207 \times 10^{-1}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}} - 8.68230 \times 10^{-4}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}}^{2} & \\
\quad +6.96506 \times 10^{-7}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}}^{3} - 1.74105 \times 10^{-10}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}}^{4}, & \hat{\mathrm{Re}}_{\theta\mathrm{t}} \leq 1870, \\
\hat{\mathrm{Re}}_{\theta\mathrm{t}} - \left(593.11 + 0.482 \left(\hat{\mathrm{Re}}_{\theta\mathrm{t}} - 1870\right)\right), & \hat{\mathrm{Re}}_{\theta\mathrm{t}} > 1870.
\end{cases}
\tag{3.25}
$$

Transition onset is triggered by the function $\mathrm{F}_{\mathrm{onset}}$, which comprises three components. The function $\mathrm{F}_{\mathrm{onset1}}$ compares the local vorticity-based Reynolds number Re_{V} with $\mathrm{Re}_{\theta\mathrm{c}}$:

$$
\mathrm{Re}_{\mathrm{V}} = \frac{\rho \mathrm{S} \mathrm{d}^{2}}{\mu}, \tag{3.26}
$$

$$
\mathrm{F}_{\mathrm{onset1}} = \frac{\mathrm{Re}_{\mathrm{V}}}{2.193 \mathrm{Re}_{\theta\mathrm{c}}}, \tag{3.27}
$$

A steep ramp-up is enforced via:

$$
\mathrm{F}_{\mathrm{onset2}} = \min\left(\max\left(\mathrm{F}_{\mathrm{onset1}}, \mathrm{F}_{\mathrm{onset1}}^{4}\right), 2.0\right), \tag{3.28}
$$

To account for feedback due to increased turbulence, a damping term based on the turbulence Reynolds number R_{T} is used:

$$
\mathrm{R}_{\mathrm{T}} = \frac{\rho \mathrm{k}}{\mu \omega}, \tag{3.29}
$$

$$
\mathrm{F}_{\mathrm{onset3}} = \max\left[1 - \left(\frac{\mathrm{R}_{\mathrm{T}}}{2.5}\right)^{3}, 0\right], \tag{3.30}
$$

The final onset function is then given by:

$$
\mathrm{F}_{\mathrm{onset}} = \max\left(\mathrm{F}_{\mathrm{onset2}} - \mathrm{F}_{\mathrm{onset3}}, 0\right). \tag{3.31}
$$

The transition length function $\mathrm{F}_{\mathrm{length}}$ governs the spatial growth of the intermittency variable γ and depends on the transition momentum thickness Reynolds number $\hat{\mathrm{Re}}_{\theta\mathrm{t}}$. To prevent nonphysical spikes in skin friction at low Reynolds numbers, the function $\mathrm{F}_{\mathrm{sublayer}}$ is introduced to ensure that $\mathrm{F}_{\mathrm{length}}$ attains a maximum value within the viscous sublayer.

The piecewise definition of the intermediate function $\mathrm{F}_{\mathrm{length,1}}$ is given by:

$$\mathrm{F_{length,1}} = \begin{cases} 39.8189 - 1.1927 \times 10^{-2}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}} - 1.32567 \times 10^{-4}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}}^{2}, & \hat{\mathrm{Re}}_{\theta\mathrm{t}} < 400, \\ 263.404 - 1.23939\, \hat{\mathrm{Re}}_{\theta\mathrm{t}} + 1.94548 \times 10^{-3}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}}^{2} - 1.01695 \times 10^{-6}\, \hat{\mathrm{Re}}_{\theta\mathrm{t}}^{3}, & 400 \leq \hat{\mathrm{Re}}_{\theta\mathrm{t}} < 596, \\ 0.5 - 3.0 \times 10^{-4} \left(\hat{\mathrm{Re}}_{\theta\mathrm{t}} - 596 \right), & 596 \leq \hat{\mathrm{Re}}_{\theta\mathrm{t}} < 1200, \\ 0.3188, & \hat{\mathrm{Re}}_{\theta\mathrm{t}} \geq 1200. \end{cases} \tag{3.32}$$

The Reynolds number Re_ω based on the specific dissipation rate ω and wall distance d is

$$\mathrm{Re}_\omega = \frac{\rho\, \omega\, \mathrm{d}^2}{\mu}. \tag{3.33}$$

Using Re_ω, the sublayer damping function $\mathrm{F_{sublayer}}$ is defined as

$$\mathrm{F_{sublayer}} = \exp\left[- \left(\frac{\mathrm{Re}_\omega}{200} \right)^2 \right]. \tag{3.34}$$

The final transition length function $\mathrm{F_{length}}$ combines these as

$$\mathrm{F_{length}} = \mathrm{F_{length,1}} \left(1 - \mathrm{F_{sublayer}} \right) + 40.0\, \mathrm{F_{sublayer}}. \tag{3.35}$$

Additionally, the intermittency production modifier $\mathrm{F_{turb}}$ is expressed as

$$\mathrm{F_{turb}} = \exp\left[- \left(\frac{\mathrm{R_T}}{4} \right)^4 \right], \tag{3.36}$$

3.4.3 Source Term of the $\hat{\mathrm{Re}}_{\theta\mathrm{t}}$ Equation

The source term for the $\hat{\mathrm{Re}}_{\theta\mathrm{t}}$ equation is given by:

$$\mathrm{P}_{\theta\mathrm{t}} = \mathrm{c}_{\theta\mathrm{t}} \frac{\rho}{\mathrm{T}} \left(\mathrm{Re}_{\theta\mathrm{t}}^{\mathrm{eq}} - \hat{\mathrm{Re}}_{\theta\mathrm{t}} \right) (1.0 - \mathrm{F}_{\theta\mathrm{t}}) \tag{3.37}$$

To ensure dimensional consistency in the transport equation for $\hat{\mathrm{Re}}_{\theta\mathrm{t}}$, a timescale T is introduced. The source term is formulated such that in the freestream, where $\mathrm{F}_{\theta\mathrm{t}} = 0$, the value of $\hat{\mathrm{Re}}_{\theta\mathrm{t}}$ asymptotically approaches its equilibrium value $\mathrm{Re}_{\theta\mathrm{t}}^{\mathrm{eq}}$. The timescale and associated auxiliary functions are defined as follows:

$$\mathrm{T} = \frac{500\, \mu}{\rho\, \mathrm{U}^2}, \tag{3.38}$$

where U is the local velocity magnitude, given by

$$\mathrm{U} = \sqrt{\mathrm{U_k U_k}}. \tag{3.39}$$

To damp transition production in the wake region, the wake suppression function is defined as

$$\mathrm{F_{wake}} = \exp\left[-\left(\frac{\mathrm{Re}_\omega}{1\times 10^5}\right)^2\right], \tag{3.40}$$

The transition suppression function also depends on the boundary layer edge thickness scale δ:

$$\delta = \frac{375\,\Omega\,\mu\,\hat{\mathrm{Re}}_{\theta\mathrm{t}}\,\mathrm{d}}{\rho\,\mathrm{U}^2}, \tag{3.41}$$

where Ω is the absolute value of vorticity magnitude.

The final form of the transition trigger function $\mathrm{F}_{\theta\mathrm{t}}$, which controls the source term in the $\hat{\mathrm{Re}}_{\theta\mathrm{t}}$ equation, is defined as

$$\mathrm{F}_{\theta\mathrm{t}} = \min\left[\max\left(\mathrm{F_{wake}}\exp\left[-\left(\frac{\mathrm{d}}{\delta}\right)^4\right],\ 1.0-\left(\frac{\mathrm{c_{e2}}\,\gamma-1}{\mathrm{c_{e2}}-1}\right)^2\right),\ 1.0\right], \tag{3.42}$$

The local acceleration along a streamline is:

$$\frac{\mathrm{dU}}{\mathrm{ds}} = \frac{\mathrm{U_m U_n}}{\mathrm{U}^2}\frac{\partial \mathrm{U_m}}{\partial \mathrm{x_n}} \tag{3.43}$$

The pressure-gradient parameter and turbulence intensity are defined as:

$$\lambda_\theta = \frac{\rho\theta_\mathrm{t}^2}{\mu}\frac{\mathrm{dU}}{\mathrm{ds}} \tag{3.44}$$

$$\mathrm{Tu} = 100\frac{\sqrt{\frac{2\mathrm{k}}{3}}}{\mathrm{U}} \tag{3.45}$$

The correlation function $\mathrm{F}(\lambda_\theta)$ accounts for pressure-gradient effects:

$$\mathrm{F}\left(\lambda_\theta\right) = \begin{cases} 1+\left(12.986\lambda_\theta+123.66\lambda_\theta^2+405.689\lambda_\theta^3\right)\exp\left(-\left(\frac{\mathrm{Tu}}{1.5}\right)^{1.5}\right), & \lambda_\theta \le 0 \\ 1+0.275\left(1-\exp\left(-35.0\lambda_\theta\right)\right)\exp\left(-\frac{\mathrm{Tu}}{0.5}\right), & \lambda_\theta > 0 \end{cases} \tag{3.46}$$

The correlation function $\mathrm{Re}_{\theta\mathrm{t}}^{\mathrm{eq}}$ is then given by turbulence intensity and pressure gradient:

$$\mathrm{Re}_{\theta\mathrm{t}}^{\mathrm{eq}} = \begin{cases} \left(1173.51-589.428\mathrm{Tu}+0.2196\mathrm{Tu}^{-2}\right)\mathrm{F}\left(\lambda_\theta\right), & \mathrm{Tu}\le 1.3 \\ 331.50\left(\mathrm{Tu}-0.5658\right)^{-0.671}\mathrm{F}\left(\lambda_\theta\right), & \mathrm{Tu} > 1.3 \end{cases} \tag{3.47}$$

Note that $\mathrm{Re}_{\theta\mathrm{t}}^{\mathrm{eq}}$ is an implicit function of θ_t, also expressible as:

$$\mathrm{Re}_{\theta\mathrm{t}}^{\mathrm{eq}} = \frac{\rho\mathrm{U}\theta_\mathrm{t}}{\mu} \tag{3.48}$$

This is solved iteratively using the secant method. While U is ideally the boundary-layer edge velocity, the model applies it as the local velocity. In regions close to the wall where U is small,

$\mathrm{Re}_{\theta t}^{\mathrm{eq}}$ is correspondingly small, and the $\mathrm{F}_{\theta t}$ term compensates by reducing the source term in the boundary layer.

3.4.4 Modification of the SST k Source Terms

To enable the SST turbulence model to capture laminar–turbulent transition, the production and dissipation source terms for the turbulent kinetic energy equation are modified based on intermittency control.

First, a reattachment factor is defined to suppress turbulent production in regions with low turbulent Reynolds number:

$$\mathrm{F}_{\mathrm{reattach}} = \exp\left[-\left(\frac{\mathrm{R}_{\mathrm{T}}}{20}\right)^4\right], \tag{3.49}$$

Next, a separation-sensitive intermittency function γ_{sep} is introduced:

$$\gamma_{\mathrm{sep}} = \min\left(\mathrm{s}_1 \max\left[0, \left(\frac{\mathrm{Re}_{\mathrm{V}}}{3.235\,\mathrm{Re}_{\theta\mathrm{c}}}\right) - 1\right] \mathrm{F}_{\mathrm{reattach}},\ 2\right) \mathrm{F}_{\theta\mathrm{t}}, \tag{3.50}$$

The term γ_{sep} captures the effect of separation-induced transition and can exceed 1, allowing for stronger activation of turbulent production during flow separation.

The final effective intermittency used to modulate production and dissipation is:

$$\gamma_{\mathrm{eff}} = \max\left(\gamma, \gamma_{\mathrm{sep}}\right), \tag{3.51}$$

ensuring that turbulence generation is driven by either natural transition (via γ) or separation-induced mechanisms (via γ_{sep}), whichever dominates.

This effective intermittency is then used to scale the k-equation source terms:

$$\hat{\mathrm{P}}_{\mathrm{k}} = \gamma_{\mathrm{eff}}\,\mathrm{P}_{\mathrm{k,SST}} \tag{3.52}$$

$$\hat{\mathrm{D}}_{\mathrm{k}} = \min\left(\max\left(\gamma_{\mathrm{eff}}, 0.1\right), 1.0\right) \mathrm{D}_{\mathrm{k,SST}} \tag{3.53}$$

where:

- $\hat{\mathrm{P}}_{\mathrm{k}}$ and $\hat{\mathrm{D}}_{\mathrm{k}}$ are the modified production and dissipation terms,
- $\mathrm{P}_{\mathrm{k,SST}}$ and $\mathrm{D}_{\mathrm{k,SST}}$ are the original SST source terms.

The constants are summarized in Table 3.2.

Table 3.2: Model constants for the k–ω–γ–Re_{θ_t} Langtry–Menter model [22].

c_{a1}	c_{a2}	c_{e1}	c_{e2}	$\mathrm{c}_{\theta\mathrm{t}}$	s_1	σ_{f}	$\sigma_{\theta\mathrm{t}}$
2.0	0.06	1.0	50	0.03	2.0	1.0	2.0

3.4.5 Modification of the SST $\mathrm{F_1}$ Blending Function

In laminar boundary layers, the original SST blending function $\mathrm{F_{1,SST}}$ may approach zero, deactivating turbulence modeling prematurely. To correct this, a modified auxiliary function $\mathrm{F_3}$ is introduced to enforce $\mathrm{F_1} = 1$ in laminar regions:

$$\mathrm{R_y} = \frac{\rho \mathrm{d}\sqrt{\mathrm{k}}}{\mu} \tag{3.54}$$

$$\mathrm{F_3} = \exp\left[-\left(\frac{\mathrm{R_y}}{120}\right)^8\right] \tag{3.55}$$

Here, $\mathrm{R_y}$ is a local Reynolds number based on the turbulent velocity scale $\sqrt{\mathrm{k}}$ and the wall distance d. The function $\mathrm{F_3}$ approaches 1 when k is very small (laminar flow), and rapidly decays as turbulence grows.

The modified blending function is defined as:

$$\mathrm{F_1} = \max\Big(\mathrm{F_{1,SST}}, \mathrm{F_3}\Big) \tag{3.56}$$

This formulation ensures that $\mathrm{F_1}$ remains close to 1 in both turbulent near-wall regions (via $\mathrm{F_{1,SST}}$) and laminar boundary layers (via $\mathrm{F_3}$), maintaining the activation of the SST model necessary for correctly modeling transition behavior.

3.4.6 Boundary Conditions for γ and $\hat{\mathrm{Re}}_{\theta \mathrm{t}}$

The boundary conditions for the intermittency function γ and the transition Reynolds number $\hat{\mathrm{Re}}_{\theta \mathrm{t}}$ are set as follows.

At the wall, the normal derivative of the intermittency function is zero,

$$\left.\frac{\partial \gamma}{\partial \mathrm{n}}\right|_{\mathrm{wall}} = 0, \tag{3.57}$$

which implies no flux of intermittency through the impermeable wall boundary.

In the farfield, the intermittency is set to unity,

$$\gamma_{\mathrm{farfield}} = 1, \tag{3.58}$$

representing fully turbulent flow entering the domain.

Similarly, the wall boundary condition for $\hat{\mathrm{Re}}_{\theta \mathrm{t}}$ enforces a zero normal gradient,

$$\left.\frac{\partial \hat{\mathrm{Re}}_{\theta \mathrm{t}}}{\partial \mathrm{n}}\right|_{\mathrm{wall}} = 0, \tag{3.59}$$

which prevents flux of the transition Reynolds number across the wall.

The farfield value of $\hat{\mathrm{Re}}_{\theta \mathrm{t}}$ depends on the freestream turbulence intensity Tu_∞ through a piecewise empirical correlation:

$$\hat{\mathrm{Re}}_{\theta\mathrm{t,farfield}} = \begin{cases} 1173.51 - 589.428\,\mathrm{Tu}_\infty + 0.2196\,\mathrm{Tu}_\infty^{-2}, & \mathrm{Tu}_\infty \leq 1.3, \\ 331.50\,(\mathrm{Tu}_\infty - 0.5658)^{-0.671}, & \mathrm{Tu}_\infty > 1.3. \end{cases} \tag{3.60}$$

This correlation captures different regimes of turbulence intensity, using a polynomial fit for low intensities and a power-law fit for higher intensities.

3.4.7 Numerical Limits for Robustness

For numerical robustness, the following three limits are enforced:

- $-0.1 \leq \lambda_\theta \leq 0.1$
- $\mathrm{Tu} \geq 0.027$
- $\mathrm{Re}_{\theta\mathrm{t}}^{\mathrm{eq}} \geq 20$

Chapter 4

Computing Derivatives

Accurately and efficiently computing derivatives is a fundamental task [68, 69]. Whether the interest is in understanding how an output changes with respect to input parameters or in optimizing a system, derivatives form the mathematical backbone of the analysis.

There are several methods for computing derivatives, each with its own advantages and limitations. The most familiar technique is symbolic differentiation, which involves manipulating mathematical expressions directly. This approach produces exact results, as in the classic example:

$$\frac{\mathrm{d}}{\mathrm{dx}}(\mathrm{x}^2 \sin \mathrm{x}) = 2\mathrm{x} \sin \mathrm{x} + \mathrm{x}^2 \cos \mathrm{x}. \tag{4.1}$$

However, symbolic differentiation quickly becomes impractical for large, complex codes, especially when the function of interest is defined algorithmically rather than analytically [70].

A more general approach is numerical differentiation, which approximates derivatives using differences between function evaluations. The simplest example is the forward finite difference:

$$\frac{\mathrm{df}}{\mathrm{dx}} \approx \frac{\mathrm{f}(\mathrm{x}+\mathrm{h}) - \mathrm{f}(\mathrm{x})}{\mathrm{h}}, \tag{4.2}$$

and its more accurate counterpart, the central difference:

$$\frac{\mathrm{df}}{\mathrm{dx}} \approx \frac{\mathrm{f}(\mathrm{x}+\mathrm{h}) - \mathrm{f}(\mathrm{x}-\mathrm{h})}{2\mathrm{h}}. \tag{4.3}$$

While easy to implement, these methods suffer from truncation and round-off errors, particularly when the choice of h is suboptimal [71]. Moreover, they require at least one additional function evaluation per input variable, making them computationally expensive in high-dimensional settings.

A lesser-known but powerful alternative is the complex step method, which leverages complex arithmetic to achieve very high accuracy [71]. By evaluating the function at a small imaginary perturbation,

$$\frac{\mathrm{df}}{\mathrm{dx}} \approx \frac{\Im(\mathrm{f}(\mathrm{x}+\mathrm{i}\epsilon))}{\epsilon}, \tag{4.4}$$

it can avoid the subtractive cancellation that plagues finite differences. This method is both simple and numerically stable, provided the function is analytic and can handle complex inputs.

4.1 Automatic Differentiation

Automatic Differentiation (AD) computes exact derivatives by systematically applying the chain rule to each elementary operation in a given algorithm. Unlike symbolic or numerical differentiation, AD avoids truncation errors, round-off errors, and the growth of symbolic expressions. It works by constructing and traversing the computational graph of the target function and operates in two primary modes: *forward* and *reverse*. The content of this section and its subsections is adapted from [72].

4.1.1 Forward Mode

In forward-mode AD, derivatives are propagated from inputs to outputs. Given a sequence of intermediate variables v_i, the total derivative with respect to an input v_j is defined as:

$$\dot{v}_i = \frac{dv_i}{dv_j}. \tag{4.5}$$

Using the chain rule, the propagation of derivatives follows:

$$\dot{v}_i = \sum_{k=j}^{i-1} \frac{\partial v_i}{\partial v_k} \dot{v}_k. \tag{4.6}$$

Assuming $\dot{v_1} = 1$ and incrementing i, let's illustrate forward-mode AD on a simple sequence of variables:

$$\begin{aligned}
\dot{v}_1 &= 1, \\
\dot{v}_2 &= \frac{\partial v_2}{\partial v_1}\dot{v}_1, \\
\dot{v}_3 &= \frac{\partial v_3}{\partial v_1}\dot{v}_1 + \frac{\partial v_3}{\partial v_2}\dot{v}_2, \\
\dot{v}_4 &= \frac{\partial v_4}{\partial v_1}\dot{v}_1 + \frac{\partial v_4}{\partial v_2}\dot{v}_2 + \frac{\partial v_4}{\partial v_3}\dot{v}_3 \equiv \frac{df}{dx}.
\end{aligned} \tag{4.7}$$

This process yields the following Jacobian structure:

$$J_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{dv_2}{dv_1} & 1 & 0 & 0 \\ \frac{dv_3}{dv_1} & \frac{dv_3}{dv_2} & 1 & 0 \\ \frac{dv_4}{dv_1} & \frac{dv_4}{dv_2} & \frac{dv_4}{dv_3} & 1 \end{bmatrix}. \tag{4.8}$$

The first column of the Jacobian has been computed using a forward sweep (from top to bottom) and corresponds to the directional derivative (or tangent) with respect to the input variable v_1. In forward-mode automatic differentiation, derivatives with respect to additional output variables can be obtained at no additional computational cost once the forward pass is performed.

4.1.2 Reverse Mode

In reverse-mode AD, sensitivities are propagated backward from outputs to inputs. For a seeded output $\mathrm{v_i}$, the adjoint variables $\bar{\mathrm{v}}_\mathrm{j}$ are defined as:

$$\bar{\mathrm{v}}_\mathrm{j} = \frac{\mathrm{dv_i}}{\mathrm{dv_j}}. \tag{4.9}$$

Using the chain rule in reverse:

$$\bar{\mathrm{v}}_\mathrm{j} = \sum_{\mathrm{k=j+1}}^{\mathrm{i}} \frac{\partial \mathrm{v_k}}{\partial \mathrm{v_j}} \bar{\mathrm{v}}_\mathrm{k}. \tag{4.10}$$

Starting with $\bar{\mathrm{v}}_4 = 1$, it propagates adjoints backward:

$$\begin{aligned}
\bar{\mathrm{v}}_4 &= 1, \\
\bar{\mathrm{v}}_3 &= \frac{\partial \mathrm{v_4}}{\partial \mathrm{v_3}} \bar{\mathrm{v}}_4, \\
\bar{\mathrm{v}}_2 &= \frac{\partial \mathrm{v_3}}{\partial \mathrm{v_2}} \bar{\mathrm{v}}_3 + \frac{\partial \mathrm{v_4}}{\partial \mathrm{v_2}} \bar{\mathrm{v}}_4, \\
\bar{\mathrm{v}}_1 &= \frac{\partial \mathrm{v_2}}{\partial \mathrm{v_1}} \bar{\mathrm{v}}_2 + \frac{\partial \mathrm{v_3}}{\partial \mathrm{v_1}} \bar{\mathrm{v}}_3 + \frac{\partial \mathrm{v_4}}{\partial \mathrm{v_1}} \bar{\mathrm{v}}_4 \equiv \frac{\mathrm{df}}{\mathrm{dx}}.
\end{aligned} \tag{4.11}$$

The corresponding Jacobian is:

$$\mathrm{J_v} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{\mathrm{dv_2}}{\mathrm{dv_1}} & 1 & 0 & 0 \\ \frac{\mathrm{dv_3}}{\mathrm{dv_1}} & \frac{\mathrm{dv_3}}{\mathrm{dv_2}} & 1 & 0 \\ \frac{\mathrm{dv_4}}{\mathrm{dv_1}} & \frac{\mathrm{dv_4}}{\mathrm{dv_2}} & \frac{\mathrm{dv_4}}{\mathrm{dv_3}} & 1 \end{bmatrix}. \tag{4.12}$$

The last row of the Jacobian has been computed using a reverse sweep (from right to left) and corresponds to the gradient of the scalar function $\mathrm{f} = \mathrm{v_4}$. In reverse-mode automatic differentiation, once the reverse pass is performed, computing derivatives with respect to additional input variables incurs negligible additional cost.

4.1.3 Forward vs. Reverse Mode

Let $\mathrm{J_f} = \frac{\partial \mathrm{f}}{\partial \mathrm{x}}$ denote the Jacobian of the outputs f with respect to the inputs x. In a computational graph, the full Jacobian of all intermediate variables is represented by $\mathrm{J_v}$, where the input variables x appear early and the output variables f appear later. The Jacobian of the output variables $\mathrm{J_f}$ is embedded as a sub-block in the lower-left portion of $\mathrm{J_v}$, as visualized below.

$$\mathrm{J_v} = \underbrace{\left.\begin{array}{|c|c|c|c|c|}\hline \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{J_f} & \cdots & \cdots & \cdots & \cdots \\ \hline \end{array}\right\rbrace \mathrm{n_f}}_{\mathrm{n_x}} \tag{4.13}$$

Forward and reverse modes of automatic differentiation differ in traversal and cost. Forward mode computes derivatives in a column-wise manner and is typically efficient when the number of inputs $\mathrm{n_x}$ is much smaller than the number of outputs $\mathrm{n_f}$. In contrast, reverse mode proceeds row-wise and is preferred when $\mathrm{n_f} \ll \mathrm{n_x}$. When $\mathrm{n_x} \approx \mathrm{n_f}$, both modes have comparable computational efficiency.

Each AD pass generally incurs a cost of about 2–3 times that of the original function evaluation. Reverse mode, however, requires storing the full computational graph for backpropagation, increasing memory usage. One solution could be couple it with implicit differentiation.

4.1.4 AD in the Context of Simulation and Optimization

In large-scale problems such as PDE-constrained simulations, AD is often used in combination with implicit differentiation in order to avoid storing the full computational graph and save memory. For a residual system $\mathrm{r(u,x)} = 0$, where the state u depends implicitly on the design variables x, the total derivative of an objective function $\mathrm{f(u,x)}$ is:

$$\frac{\mathrm{df}}{\mathrm{dx}} = \frac{\partial \mathrm{f}}{\partial \mathrm{x}} + \frac{\partial \mathrm{f}}{\partial \mathrm{u}}\frac{\mathrm{du}}{\mathrm{dx}}. \tag{4.14}$$

Differentiating the residual equation yields:

$$\frac{\partial \mathrm{r}}{\partial \mathrm{u}}\frac{\mathrm{du}}{\mathrm{dx}} + \frac{\partial \mathrm{r}}{\partial \mathrm{x}} = 0, \quad \Rightarrow \quad \frac{\mathrm{du}}{\mathrm{dx}} = -\left(\frac{\partial \mathrm{r}}{\partial \mathrm{u}}\right)^{-1}\frac{\partial \mathrm{r}}{\partial \mathrm{x}}. \tag{4.15}$$

Substituting this into the total derivative expression gives:

$$\frac{\mathrm{df}}{\mathrm{dx}} = \frac{\partial \mathrm{f}}{\partial \mathrm{x}} - \frac{\partial \mathrm{f}}{\partial \mathrm{u}}\left(\frac{\partial \mathrm{r}}{\partial \mathrm{u}}\right)^{-1}\frac{\partial \mathrm{r}}{\partial \mathrm{x}}. \tag{4.16}$$

When computing total derivatives in large-scale systems, two efficient strategies are commonly employed depending on the relative dimensions of the input and output spaces.

The *direct method* is advantageous when the number of design variables $\mathrm{n_x}$ is small. It involves solving for the sensitivity matrix

$$\phi = \left(\frac{\partial \mathrm{r}}{\partial \mathrm{u}}\right)^{-1}\frac{\partial \mathrm{r}}{\partial \mathrm{x}}, \tag{4.17}$$

which is then substituted into the total derivative expression:

$$\frac{\mathrm{df}}{\mathrm{dx}} = \frac{\partial \mathrm{f}}{\partial \mathrm{x}} - \frac{\partial \mathrm{f}}{\partial \mathrm{u}}\,\phi. \tag{4.18}$$

In contrast, the *adjoint method* is more efficient when the number of objective or constraint functions $\mathrm{n_f}$ is small. It avoids explicitly forming ϕ by solving the adjoint equation:

$$\left(\frac{\partial \mathrm{r}}{\partial \mathrm{u}}\right)^{\mathrm{T}} \psi = \left(\frac{\partial \mathrm{f}}{\partial \mathrm{u}}\right)^{\mathrm{T}}, \tag{4.19}$$

and computes the total derivative via:

$$\frac{\mathrm{df}}{\mathrm{dx}} = \frac{\partial \mathrm{f}}{\partial \mathrm{x}} - \psi^{\mathrm{T}}\frac{\partial \mathrm{r}}{\partial \mathrm{x}}. \tag{4.20}$$

4.1.5 Tools and Applications

Automatic differentiation (AD) is widely applied in scientific computing and machine learning [73]. Source-code transformation tools such as Tapenade [74] support languages like Fortran and C. By providing accurate and scalable sensitivity information, AD is essential in modern design optimization workflows [70, 75].

In the `ADflow` solver, forward mode AD is used to compute derivatives during simulation, while reverse mode AD is employed by the adjoint solver for efficient gradient-based optimization.

Chapter 5

Flow Solver and Numerical Setup

5.1 Overview of ADflow Solver

ADflow is a multi-block, overset structured flow solver originally developed at Stanford University as part of the Department of Energy's Advanced Strategic Computing (ASC) Initiative. It solves the compressible Euler, laminar Navier–Stokes, and Reynolds-Averaged Navier–Stokes (RANS) equations using a second-order finite volume discretization scheme [75].

Originally designed for simulating flow in rotating jet engine components, ADflow has evolved into a versatile CFD tool widely applied to external aerodynamic flows and other complex scenarios. It is optimized for massively parallel computing via the Message Passing Interface (MPI [76]), with parallelism abstracted from the user. Users interact with unified grid and solution files regardless of processor count, simply specifying the number of processors through commands such as `mpirun`.

ADflow supports steady compressible flows and a variety of turbulence models, including Spalart–Allmaras, k-ε, SST, and v^2-f. Input/output operations conform to the CGNS standard, with built-in MPI-IO support for efficient parallel file access, and utilities for data conversion between CGNS and other formats.The computational mesh, provided in metric units, is internally nondimensionalized.

Residuals can be solved via Runge–Kutta, the diagonalized-diagonally-dominant alternating direction implicit (D3ADI) scheme [77], or an approximate Newton–Krylov (ANK) solver [78], which also supports the full Newton–Krylov method [79]. This study adopts the ANK solver for its proven robustness and numerical efficiency [80]. Turbulence equations can be tightly coupled with the mean flow using the coupled ANK (CANK) approach, which supports second-order accurate (SANK) and coupled second-order accurate (CSANK) formulations [81].

Spatial discretization employs a second-order difference scheme, stabilized by multiple artificial dissipation techniques. Fully parallel preprocessing tools handle domain decomposition and load balancing, enabling scalability across thousands of processors.

Boundary conditions are automatically derived from flight parameters—reference length, reference area, Mach number, Reynolds number, and angle of attack—minimizing manual input and reducing errors.

5.2 Verification and Validation

The outcome of CFD simulations is a numerical solution to the governing equations. To assess its quality, a rigorous *Verification and Validation* (V&V) process is required [82]. Verification ensures that the equations are solved correctly (*"solving the equations right"*), whereas validation assesses whether the correct equations are being solved (*"solving the right equations"*) [82].

Verification is divided into *Code Verification* (checking for programming errors) and *Solution Verification* (quantifying numerical error). The total error E in a quantity of interest ϕ is

$$\mathrm{E} = \phi_{\mathrm{S}} - \phi_{\mathrm{T}}, \tag{5.1}$$

where ϕ_{S} is the simulation result and ϕ_{T} the true value. This error is decomposed as

$$\mathrm{E} = \mathrm{E}_{\mathrm{m}} + \mathrm{E}_{\mathrm{input}} + \mathrm{E}_{\mathrm{num}}, \tag{5.2}$$

where E_{m} is the modeling error, $\mathrm{E}_{\mathrm{input}}$ the input error, and $\mathrm{E}_{\mathrm{num}}$ the numerical error. The latter includes:

- Round-off error ($\mathrm{E}_{\text{round-off}}$) from finite precision arithmetic [83],
- Iterative error (E_{it}) from non-linear solver convergence [84],
- Discretization error ($\mathrm{E}_{\mathrm{disc}}$) due to spatial discretization [82],
- Statistical error ($\mathrm{E}_{\mathrm{stat}}$) from initial conditions or stochastic sampling [85].

$$\mathrm{E}_{\mathrm{num}} = \mathrm{E}_{\text{round-off}} + \mathrm{E}_{\mathrm{it}} + \mathrm{E}_{\mathrm{disc}} + \mathrm{E}_{\mathrm{stat}}. \tag{5.3}$$

In this work, double precision is used, rendering $\mathrm{E}_{\text{round-off}}$ negligible. Iterative convergence is enforced to residual norms normalized with respect to the values of the first 5 iterations below 10^{-6} (2D), minimizing E_{it}. The dominant contribution is $\mathrm{E}_{\mathrm{disc}}$, as simulations are mostly steady.

Discretization error is estimated via grid refinement studies [82], modeled as

$$\mathrm{E}_{\mathrm{disc}} = \phi_{\mathrm{i}} - \phi_{0} \approx \alpha \mathrm{h}_{\mathrm{i}}^{\mathrm{p}}, \tag{5.4}$$

where ϕ_{i} is the result on grid i, h_{i} its characteristic size, α a constant, and p the observed order of accuracy.

Code verification of the ADflow solver and its turbulence models has been extensively conducted in prior studies employing the order-of-accuracy assessments, and Jacobian consistency checks [86, 87, 75, 80]. These efforts confirm the correct implementation of the discretized governing equations within the solver. As this work focuses on the implementation and verification of a transition model within ADflow, the core solver code verification is considered complete, with the exception of the derivatives related to γ and $\mathrm{Re}_{\theta_{\mathrm{t}}}$. The derivatives, computed via the automatic differentiation tool in forward mode, are subject to verification against finite difference approximations to ensure their accuracy.

Validation compares numerical results to experimental and simulation data, which must include associated uncertainties [82]. However, due to incomplete experimental characterization and lack of inlet turbulence data, formal validation is not feasible in this work. Nevertheless, qualitative comparisons are made to highlight the benefits of transition modelling.

5.2.1 Grid Refinement Study

Mesh convergence is assessed using error estimations based on Richardson extrapolation and the Grid Convergence Index (GCI) [82, 88]. The refinement ratio between two grids is defined as:

$$\mathrm{r} = \frac{\mathrm{h_{coarse}}}{\mathrm{h_{fine}}}, \quad \text{with} \quad \mathrm{h} = \mathrm{N}^{-1/\mathrm{d}}, \tag{5.5}$$

where h is the representative grid spacing, N the number of cells, and d the domain dimension.

The estimated order of accuracy p from three grid levels is:

$$\mathrm{p} = \frac{\ln\left(\frac{\mathrm{f_{L_2}} - \mathrm{f_{L_1}}}{\mathrm{f_{L_1}} - \mathrm{f_{L_0}}}\right)}{\ln(\mathrm{r})}, \tag{5.6}$$

where $\mathrm{f_{L_2}}$, $\mathrm{f_{L_1}}$, and $\mathrm{f_{L_0}}$ are solutions on the coarse, medium, and fine grids, respectively.

Using p, the approximate discretization errors are:

$$\mathrm{E_1} = \frac{\mathrm{f_{L_2}} - \mathrm{f_{L_1}}}{1 - \mathrm{r^p}}, \tag{5.7}$$

$$\mathrm{E_2} = \mathrm{r^p E_1}. \tag{5.8}$$

The GCI quantifies the relative error:

$$\mathrm{GCI} = \mathrm{F_S E}, \tag{5.9}$$

where $\mathrm{F_S}$ is a safety factor and E is the estimated error.

Finally, the extrapolated solution as h → 0 is given by:

$$\mathrm{f(h = 0)} = \mathrm{f_{L_0}} + \frac{\mathrm{f_{L_0}} - \mathrm{f_{L_1}}}{\mathrm{r^p} - 1}, \tag{5.10}$$

assuming smooth and monotonic convergence.

5.3 Solution Procedure

The transport equation for a scalar quantity ϕ is given by [89]:

$$\frac{\partial(\rho\phi)}{\partial \mathrm{t}} + \nabla \cdot (\rho\phi\mathbf{U}) = \nabla \cdot (\Gamma\nabla\phi) + \rho\mathrm{F}, \tag{5.11}$$

where Γ denotes the effective diffusivity and F is a volumetric source term. Applying the finite volume method to a control volume V with boundary surface S and outward unit normal **n**, the integral form becomes [90]:

$$\frac{\partial}{\partial \mathrm{t}} \int_{\mathrm{V}} \rho\phi \, \mathrm{dV} + \int_{\mathrm{S}} \rho\phi \mathbf{U} \cdot \mathbf{n} \, \mathrm{dS} = \int_{\mathrm{S}} \Gamma \nabla \phi \cdot \mathbf{n} \, \mathrm{dS} + \int_{\mathrm{V}} \rho \mathrm{F} \, \mathrm{dV}. \tag{5.12}$$

The first term in the integral form, representing the time derivative, is neglected under the steady-state assumption and thus omitted from the discretized equation.

The source term is integrated over the control volume and approximated using the midpoint rule:

$$\int_{\mathrm{V}} \rho \mathrm{F} \, \mathrm{dV} \approx \rho_{\mathrm{c}} \mathrm{F}_{\mathrm{c}} \, \Delta \mathrm{V}_{\mathrm{c}}, \tag{5.13}$$

where ρ_{c}, F_{c}, and $\Delta \mathrm{V}_{\mathrm{c}}$ are the density, source term, and volume evaluated at the cell center, respectively.

The convective term is discretized as:

$$\int_{\mathrm{S}} \rho\phi \mathbf{U} \cdot \mathbf{n} \, \mathrm{dS} \approx \sum_{\mathrm{f}=1}^{\mathrm{N_f}} \rho_{\mathrm{f}} \phi_{\mathrm{f}} \, \mathbf{U}_{\mathrm{f}} \cdot \mathbf{S}_{\mathrm{f}}, \tag{5.14}$$

where $\mathbf{S}_{\mathrm{f}} = \mathrm{S}_{\mathrm{f}} \mathbf{n}_{\mathrm{f}}$ is the face area vector, defined as a vector normal to face f, pointing outward from the control volume, with magnitude equal to the area of the face. The scalar quantity ϕ_{f} is reconstructed at the face using a second-order upwind scheme with the Van Albada flux limiter [91] to reduce numerical oscillations near steep gradients. The face value is computed as:

$$\phi_{\mathrm{f}} = \phi_{\mathrm{U}} + \frac{1}{2} \psi(\mathrm{r}) (\phi_{\mathrm{D}} - \phi_{\mathrm{U}}), \tag{5.15}$$

where the limiter function is given by:

$$\psi(\mathrm{r}) = \frac{\mathrm{r}^2 + \mathrm{r}}{\mathrm{r}^2 + 1}, \quad \mathrm{r} = \frac{\phi_{\mathrm{U}} - \phi_{\mathrm{UU}}}{\phi_{\mathrm{D}} - \phi_{\mathrm{U}} + \epsilon}, \quad \epsilon \ll 1. \tag{5.16}$$

Here, ϕ_{U} is the value in the upwind cell, ϕ_{D} the value in the downwind cell, ϕ_{UU} the second upwind value, and ϵ is a small regularization parameter used to avoid division by zero.

The diffusive term is discretized as:

$$\int_{\mathrm{S}} \Gamma \nabla \phi \cdot \mathbf{n} \, \mathrm{dS} \approx \sum_{\mathrm{f}=1}^{\mathrm{N_f}} \Gamma_{\mathrm{f}} (\nabla \phi)_{\mathrm{f}} \cdot \mathbf{S}_{\mathrm{f}}, \tag{5.17}$$

where Γ_{f} is the diffusivity interpolated at face f, and $\mathbf{S}_{\mathrm{f}}$ is again the face area vector. The gradient $(\nabla\phi)_{\mathrm{f}}$ at the face is computed using the Gauss divergence theorem:

$$(\nabla \phi)_{\mathrm{f}} \approx \frac{1}{\Delta \mathrm{V}_{\mathrm{c}}} \sum_{\mathrm{j}=1}^{\mathrm{N_{f,c}}} \phi_{\mathrm{j}} \mathbf{S}_{\mathrm{j}}, \tag{5.18}$$

Combining all contributions, the steady-state discretized transport equation for each control volume is expressed as:

$$\sum_{\mathrm{f}=1}^{\mathrm{N_f}} \left(\rho_\mathrm{f}\phi_\mathrm{f}\,\mathbf{U}_\mathrm{f}\cdot\mathbf{S}_\mathrm{f} - \Gamma_\mathrm{f}(\nabla\phi)_\mathrm{f}\cdot\mathbf{S}_\mathrm{f}\right) = \rho_\mathrm{c}\mathrm{F_c}\,\Delta\mathrm{V_c}. \tag{5.19}$$

5.4 Freestream Decay of Turbulence Quantities in the SST k–ω Model

The downstream evolution of turbulent kinetic energy, k, and specific dissipation rate, ω, can be characterized using normalized variables defined as

$$\tilde{\mathrm{k}} = \frac{\mathrm{k}}{\mathrm{U}_\infty^2}, \quad \tilde{\omega} = \frac{\omega \mathrm{L}}{\mathrm{U}_\infty}, \quad \tilde{\mathrm{x}} = \frac{\mathrm{x}}{\mathrm{L}},$$

where L is a representative length scale. The decay behavior follows from the turbulence transport equations (fully derived in [58]) and is given by:

$$\tilde{\mathrm{k}}(\tilde{\mathrm{x}}) = \frac{\tilde{\mathrm{k}}_\infty}{\left[1 + \alpha(\tilde{\mathrm{x}} - \tilde{\mathrm{x}}_\infty)\tilde{\mathrm{k}}_\infty \left(\frac{\mu}{\mu_\mathrm{t}}\right)_\infty \mathrm{Re}\right]^{\frac{\alpha^*}{\alpha}}}, \tag{5.20}$$

$$\tilde{\omega}(\tilde{\mathrm{x}}) = \frac{\tilde{\omega}_\infty}{1 + \alpha(\tilde{\mathrm{x}} - \tilde{\mathrm{x}}_\infty)\tilde{\mathrm{k}}_\infty \left(\frac{\mu}{\mu_\mathrm{t}}\right)_\infty \mathrm{Re}}, \tag{5.21}$$

with the eddy-viscosity ratio decaying according to

$$\frac{\mu_\mathrm{t}}{\mu}(\tilde{\mathrm{x}}) = \frac{\left(\frac{\mu_\mathrm{t}}{\mu}\right)_\infty}{\left[1 + \alpha(\tilde{\mathrm{x}} - \tilde{\mathrm{x}}_\infty)\tilde{\mathrm{k}}_\infty \left(\frac{\mu}{\mu_\mathrm{t}}\right)_\infty \mathrm{Re}\right]^{\frac{\alpha^*}{\alpha}-1}}. \tag{5.22}$$

The empirical model constants are typically taken as

$$\alpha = 0.072, \quad \alpha^* = 0.09,$$

based on calibration against homogeneous isotropic turbulence decay [12].

These relations indicate that k and ω diminish rapidly downstream, whereas the eddy-viscosity ratio μ_t/μ decays more slowly, controlled by the exponent $\frac{\alpha^*}{\alpha} - 1$. The rate of decay increases with Reynolds number but can be alleviated by elevating the inlet eddy-viscosity ratio. This effect is especially important over long upstream domains where inflow boundaries are positioned to approximate far-field conditions or prevent artificial feedback into the computational region.

Since these decay laws stem from turbulence model calibration, they provide an approximate but not exact description of freestream turbulence decay in real flows [92].

5.5 Computation of y^+ from Skin Friction Coefficient and Reynolds Number

The dimensionless wall distance, y^+, is defined as [90]:

$$\mathrm{y}^+ = \frac{\mathrm{u}_\tau \mathrm{y}}{\nu}$$

where:

- $\mathrm{u}_\tau = \sqrt{\frac{\tau_\mathrm{w}}{\rho}}$ is the friction velocity,
- y is the distance from the wall to the center of the first cell (first cell height),
- $\nu = \mu/\rho$ is the kinematic viscosity.

The wall shear stress can be expressed using the skin friction coefficient $\mathrm{C_f}$:

$$\tau_\mathrm{w} = \frac{1}{2}\rho \mathrm{U}_\infty^2 \mathrm{C_f},$$

and thus,

$$\mathrm{u}_\tau = \mathrm{U}_\infty \sqrt{\frac{\mathrm{C_f}}{2}}.$$

Substituting into the y^+ expression:

$$\mathrm{y}^+ = \frac{\mathrm{U}_\infty \mathrm{y}}{\nu}\sqrt{\frac{\mathrm{C_f}}{2}}.$$

Using the Reynolds number definition,

$$\mathrm{Re_L} = \frac{\mathrm{U}_\infty \mathrm{L}}{\nu} \implies \frac{\mathrm{U}_\infty}{\nu} = \frac{\mathrm{Re_L}}{\mathrm{L}},$$

the final formula becomes

$$\mathrm{y}^+ = \left(\frac{\mathrm{y}\cdot \mathrm{Re_L}}{\mathrm{L}}\right)\sqrt{\frac{\mathrm{C_f}}{2}}.$$

This formulation follows standard boundary layer theory and wall scaling arguments [35, 12], commonly used in turbulence modeling and CFD meshing guidelines [33].

5.6 Grids for 2D Test Cases

The computational grids employed for the three 2D airfoil test cases are all O-type meshes, specifically designed to accurately resolve the boundary layers and wake regions. The first cell height is fixed at $\mathrm{s}_0 = 1 \times 10^{-6}$ meters to ensure sufficiently fine near-wall resolution. These meshes are generated using pyHyp [93], a hyperbolic mesh generator that automatically creates two- or three-dimensional meshes around simple geometric configurations. The approach begins with an initial surface (or curve) representing the geometry of interest, then grows or extrudes the mesh in successive layers until reaching a desired distance from the original surface, effectively meshing the entire surrounding domain. The grids, as well as the flow domains, are visualized using `ParaView` [94].

5.6.1 SD7003

The SD7003 is a low Reynolds number airfoil extensively studied in wind turbine and micro-air vehicle aerodynamics due to its favorable performance at low speeds [95, 96]. Simulations were conducted at angles of attack of 4° and 6°, with a Mach number of $\mathrm{M} = 0.1$ and a Reynolds number of $\mathrm{Re} = 6 \times 10^4$. The maximum skin friction coefficient for the baseline case is approximately $\mathrm{C_f} = 0.05$.

The primary grid used in this study, denoted `SD7003_L0`, is an O-type mesh with 119201 total cells, 240000 nodes, and 200 layers in the wall-normal direction. Near-wall resolution is ensured by a first cell height of $\mathrm{s_0} = 1 \times 10^{-6}$ m. To verify mesh-independence of the results, two additional coarser grids—`SD7003_L1` and `SD7003_L2`—were constructed while keeping the geometry and boundary layer clustering consistent.

Table 5.1: Grid parameters for SD7003 mesh convergence study

Grid ID	Cells	Nodes	Wall Cells	Wall Nodes	Layers	$\mathrm{s_0}$ [m]	Max $\mathrm{y^+}$
`SD7003_L0`	119201	240000	599	1200	200	1×10^{-6}	0.0095
`SD7003_L1`	66901	135000	449	900	150	1×10^{-6}	0.0095
`SD7003_L2`	44551	90000	299	600	150	1×10^{-6}	0.0095

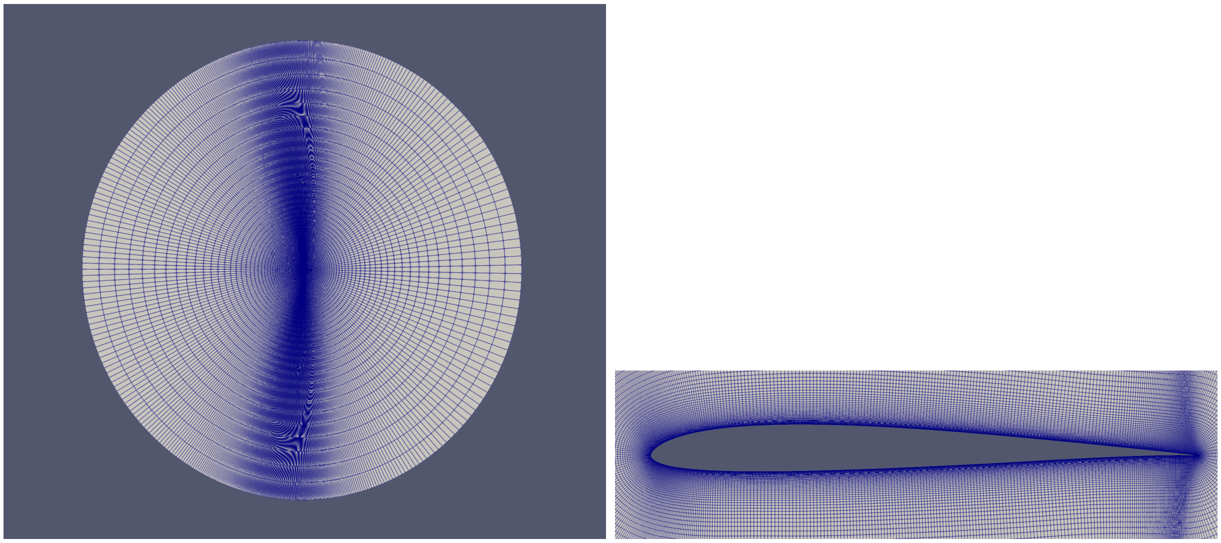

Figure 5.1: O-type grid for the SD7003 airfoil (`SD7003_L0`). Left: full domain view. Right: zoom near the airfoil surface. The inlet and outlet boundaries are placed at 100 chord lengths from the airfoil.

5.6.2 NACA 0015

The NACA 0015 is a symmetric airfoil with a 15% thickness ratio, frequently used as a baseline for aerodynamic studies [97]. Simulations were conducted at angles of attack of 3° and 10°, with a Mach number of $\mathrm{M} = 0.1$ and a Reynolds number of $\mathrm{Re} = 1.8 \times 10^5$. The maximum skin friction coefficient is approximately $\mathrm{C_f} = 0.03$.

Table 5.2: Grid parameters for the NACA0015 airfoil

Parameter	Cells	Nodes	Wall Cells	Wall Nodes	Layers	$\mathrm{s_0}$ [m]	Max $\mathrm{y^+}$
Value	119201	240000	599	1200	200	1×10^{-6}	0.0220

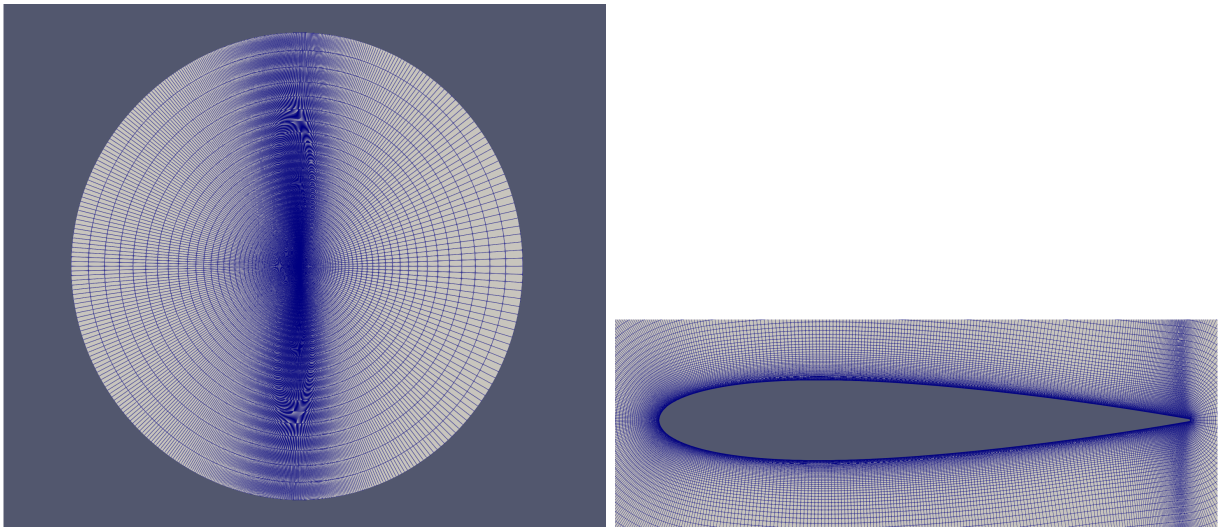

Figure 5.2: O-type grid for the NACA0015 airfoil. Left: full domain view. Right: zoom near the airfoil surface. The inlet and outlet boundaries are placed at 100 chord lengths from the airfoil.

5.6.3 Eppler 387

The Eppler 387 is a thick, high-lift airfoil designed specifically for wind turbine applications, notable for its gentle camber and thick trailing edge [98]. Simulations were performed at angles of attack of 0° and 4°, with a Mach number of $\mathrm{M} = 0.1$ and a Reynolds number of $\mathrm{Re} = 2 \times 10^5$. The maximum skin friction coefficient is approximately $\mathrm{C_f} = 0.03$.

Table 5.3: Grid parameters for the E387 airfoil

Parameter	Cells	Nodes	Wall Cells	Wall Nodes	Layers	$\mathrm{s_0}$ [m]	Max $\mathrm{y^+}$
Value	119201	240000	599	1200	200	1×10^{-6}	0.0245

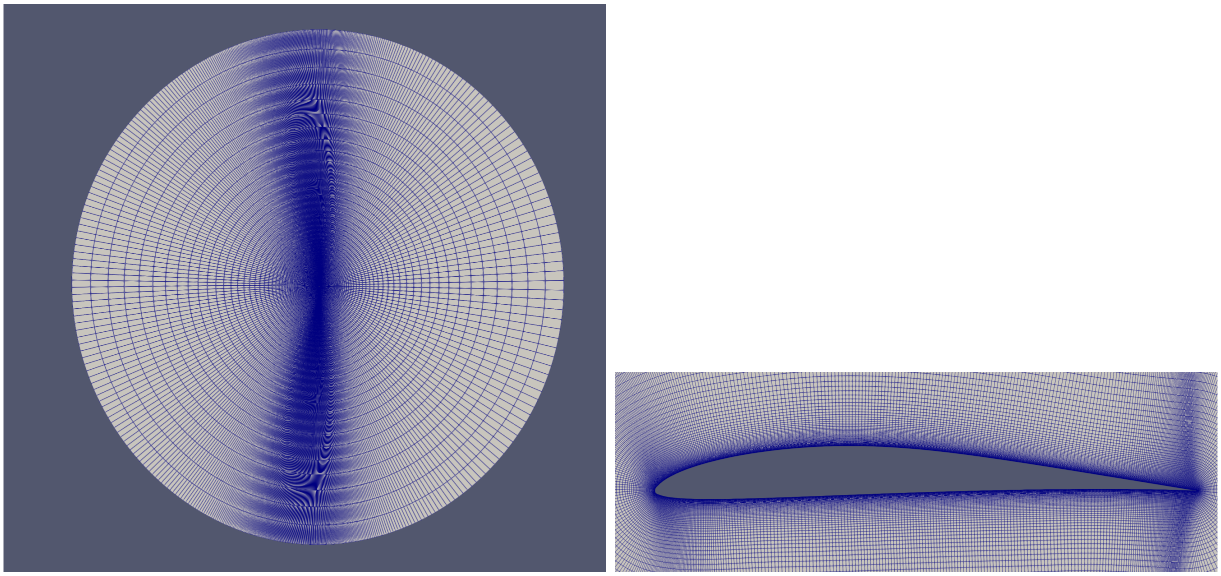

Figure 5.3: O-type grid for the E387 airfoil. Left: full domain view. Right: zoom near the airfoil surface. The inlet and outlet boundaries are placed at 100 chord lengths from the airfoil.

5.7 Aerodynamic Performance Quantities for Transition Analysis

The first aerodynamic quantity of interest is the pressure coefficient, $\mathrm{C_p}$, defined as:

$$\mathrm{C_p} = \frac{\mathrm{p} - \mathrm{p}_\infty}{\frac{1}{2}\rho \mathrm{V}_\infty^2} \tag{5.23}$$

The distribution of $\mathrm{C_p}$ is often indicative of the transition location, especially in cases of separation-induced transition [40]. Experimental data for $\mathrm{C_p}$ is available for all validation cases.

The skin-friction coefficient ($\mathrm{C_f}$) is defined as

$$\mathrm{C_f} = \frac{\tau_\mathrm{w}}{\frac{1}{2}\rho \mathrm{V}_\infty^2} \tag{5.24}$$

where τ_w is the wall shear stress, computed from the no-slip condition as

$$\tau_\mathrm{w} = \mu \left. \frac{\partial \mathrm{U_t}}{\partial \mathrm{x_n}} \right|_\mathrm{w} \tag{5.25}$$

with $\mathrm{U_t}$ the tangential velocity, $\mathrm{x_n}$ the wall-normal coordinate, and the subscript w denoting evaluation at the wall [59].

The skin-friction coefficient ($\mathrm{C_f}$) distribution is a valuable indicator for detecting flow separation and transition in two-dimensional boundary layers [59]. Regions where $\mathrm{C_f} < 0$ indicate reversed flow, with separation and reattachment points defined by the locations where $\mathrm{C_f} = 0$; the transition onset is typically identified just downstream of the local minimum in the $\mathrm{C_f}$ curve, where $\mathrm{C_f}$ begins to increase due to laminar breakdown and the development of turbulence [58]. This point corresponds to the *transition Reynolds number*, which differs from the *critical Reynolds number*—the location where small disturbances first become unstable. The point associated with the critical Reynolds number cannot be directly inferred from the $\mathrm{C_f}$ distribution; it is generally located upstream of the transition onset and must be identified through linear stability analysis [67]. The transition region typically ends at the local maximum of $\mathrm{C_f}$ [58]. The relevant points are illustrated in Figure 5.4.

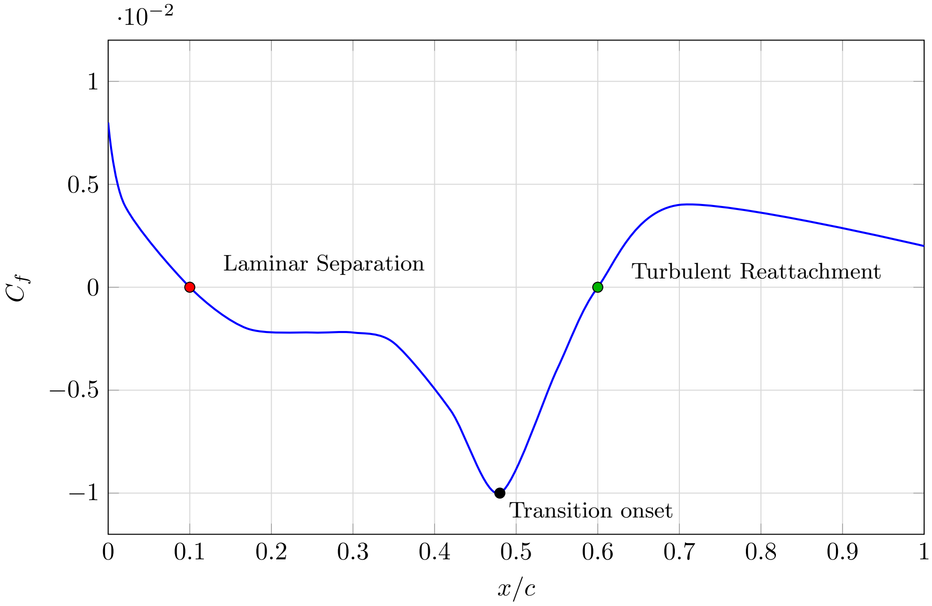

Figure 5.4: Skin-friction coefficient ($\mathrm{C_f}$) distribution along the surface, highlighting key points.

Lift and drag coefficients are defined as

$$\mathrm{C_l} = \frac{\mathrm{l}}{\frac{1}{2}\rho \mathrm{V}_\infty^2 \mathrm{c}}, \qquad \mathrm{C_d} = \frac{\mathrm{d}}{\frac{1}{2}\rho \mathrm{V}_\infty^2 \mathrm{c}} \tag{5.26}$$

where l and d are lift and drag per unit span, and c is the chord length [59].

Chapter 6

Results Part 1: Implementation, Calibration and Numerical Aspects

6.1 Code Implementation

The Fortran routines implementing the Langtry–Menter γ–Re_θ transition model have been developed and integrated into the `ADflow` solver framework [99]. The transition model is now compatible with all available solvers, including D3ADI, multigrid, and ANK. It is coupled to the mean flow variables and solved simultaneously via the coupled ANK formulation (CANK), and it also supports the second-order accurate and coupled second-order accurate implicit formulations (SANK and CSANK) [81]. In the Python interface `pyADflow.py`, key modifications enable the specification of freestream turbulence intensity (TI_∞) and freestream eddy viscosity ratio $(\mu_\mathrm{t}/\mu)_\infty$, allowing users to control these parameters directly through ADflow options in their run scripts.

6.2 Code Documentation

Clear and comprehensive documentation is essential to ensure that even well-designed code remains usable, maintainable, and extensible over time. This section provides detailed documentation for the implementation of the Langtry–Menter transition model, excluding minor components such as integrations within specific solver routines, that are documented in Appendix B. The goal is to present the documentation in a straightforward and descriptive manner, enabling future developers to understand, modify, or build upon the codebase effectively. The author hopes that this effort will facilitate ongoing development and foster collaboration within the community.

6.3 GammaRethetaModel module

The `GammaRethetaModel` module implements the Langtry–Menter correlation-based transition model. It includes routines to compute, diffuse, and scale the transition variables: the intermittency γ and the transition onset Reynolds number $\mathrm{Re}_{\theta\mathrm{t}}$.

6.3.1 Core Functionality

- `solve_local_Re_thetatt_eq`: Solves an implicit equation for the local transition Reynolds number $\mathrm{Re}_{\theta\mathrm{t}}$ using a secant method. This involves:
  - Computing local velocity magnitude and gradients.
  - Evaluating turbulence intensity and relevant flow parameters.
  - Iteratively solving the nonlinear correlation with convergence checks and clipping for numerical stability.
- `GammaRethetaSource`: Computes source terms for the transition transport equations. It:
  - Loops over all interior cells.
  - Calculates velocity gradients, vorticity, turbulence intensity, and local $\mathrm{Re}_{\theta\mathrm{t}}$.
  - Applies empirical correlations and blending functions to evaluate transition onset and progression.
- `GammaRethetaViscous`: Computes viscous diffusion terms for the transition variables in the ξ, η, ζ directions. Specifically:
  - Loops over all interior grid points (excluding ghost cells).
  - Computes direction-specific volume metrics, e.g.,

$$\begin{aligned} \mathrm{voli} &= \frac{1}{\mathrm{vol(i,j,k)}}, \\ \mathrm{volmi} &= \frac{2}{\mathrm{vol(i,j,k)} + \mathrm{vol(i,j,k-1)}}, \quad \text{etc.} \end{aligned}$$

  - Uses central differences to approximate second derivatives of the form:

$$\frac{\partial}{\partial \zeta}\left(\mu \frac{\partial \phi}{\partial \zeta}\right) \approx \frac{1}{\rho}\left[\mathrm{c}_{1\mathrm{m}}\phi_{\mathrm{k-1}} - \mathrm{c}_{10}\phi_{\mathrm{k}} + \mathrm{c}_{1\mathrm{p}}\phi_{\mathrm{k+1}}\right]$$

  - Includes two distinct contributions following the model:
    - The diffusion term in the γ-equation uses laminar and eddy viscosity, scaled by the calibration constant rLMsigmaf (defined in `paramTurb`).
    - The $\mathrm{Re}_{\theta\mathrm{t}}$-equation applies a similar diffusion model, scaled by rLMsigmathetat (also in `paramTurb`).
  - Updates the `scratch` array with directional contributions for each cell.
- `GammaRethetaResScale`: Scales residuals for consistency with the flow solver:
  - Iterates over interior points.

  - Scales the residuals by reference cell volume and applies masking:

$$\mathrm{dw}(\mathrm{i},\mathrm{j},\mathrm{k}) = -\mathrm{volRef}(\mathrm{i},\mathrm{j},\mathrm{k}) \cdot \mathrm{scratch}(\mathrm{i},\mathrm{j},\mathrm{k}) \cdot \mathrm{iblank}(\mathrm{i},\mathrm{j},\mathrm{k})$$

  - This ensures residuals are consistent in magnitude and structure with the rest of the solver.

6.3.2 Implementation Notes

- The secant method used avoids the need for analytical derivatives while retaining fast convergence (in less than ten iterations).
- All routines are tightly coupled to local flow conditions, enabling accurate modeling of transitional behavior.
- Numerical stability is maintained through limiting, clipping, and ghost-cell treatment.
- The module integrates seamlessly into the turbulence framework to modulate source terms during transition.

6.4 `SST_block_residuals` Subroutines with Transition Model Support in the module `SST`

These subroutines compute the residuals for the SST turbulence model, optionally coupled with the `gamma-Retheta` transition model. The `_d` variant is the automatically differentiated (AD) version used to compute derivatives.

6.4.0.1 Overview

- `SST_block_residuals`: Computes block residuals.
- `SST_block_residuals_d`: Computes residuals and their derivatives using Automatic Differentiation (AD) generated code .
- Both versions follow the same structure: evaluate transition model terms (if enabled), then compute standard SST turbulence contributions.

6.4.0.2 Transition Model: `gamma-Retheta`

When `transitionModel = gammaRetheta`, the following steps are executed:

- `strainNorm2` and `prodWmag2`: Compute local strain rate and vorticity magnitudes squared.
- `GammaRethetaSource`: Adds source terms for the transition variables based on empirical correlations and flow quantities.

- `turbAdvection`: Adds advection terms for the transition transport equations. Advection term subroutine is the same for all the models, just attention must be kept for the indices in w (array of the flow and turbulent variables).
- `GammaRethetaViscous`: Computes viscous diffusion terms for transition variables using variable viscosity and mesh metrics.
- `GammaRethetaResScale`: Scales transition residuals with cell volumes and applies overset masking.

In the differentiated version, the corresponding AD-aware routines are called: `GammaRethetaSource_d`, `GammaRethetaViscous_d`, `GammaRethetaResScale_d`, etc.

6.4.0.3 SST Turbulence Model Terms

Always executed regardless of transition model activation:

- `kwCDterm`: Computes the cross-diffusion term in the ω-equation.
- `f1SST`: Evaluates the blending function F_1 for SST model mixing.
- `prodSmag2`, `prodWmag2`, `prodKatoLaunder`: Compute turbulence production using the selected production model.
- `SSTSource`: Adds source terms for the k and ω transport equations.
- `turbAdvection`: Computes advection terms for k and ω.
- `SSTViscous`: Computes viscous diffusion terms for k and ω.
- `SSTResScale`: Scales residuals by cell volume for consistency and stability.

6.4.0.4 Automatic Differentiation (AD) Version

- The `_d` version mirrors the primal routine but uses AD differentiated subroutines (e.g., `prodSmag2_d`, `SSTSource_d`).
- Required for AD derivatives when ANK or NK solvers are used.
- The structure is kept consistent to ensure compatibility and correctness of the AD toolchain.

6.4.0.5 Cleanup and Memory Management

- `qq`: A central array used for temporary Jacobian or flux computations is allocated and deallocated based on the `cleanUp` flag.

6.5 `SSTSource` Subroutine: SST Source Term Evaluation with Transition Model Support in the module `SST`

`SSTSource` computes the source terms and partial Jacobians for the k–ω SST turbulence model in each cell of the computational block. The implementation supports both the standard SST model and its modification via the `gamma-Retheta` transition model.

6.5.1 Blending Constants

The constants γ_1, γ_2, β_1, β_2 are blended using the Menter function f_1:

$$\gamma = f_1\gamma_1 + (1-f_1)\gamma_2, \quad \beta = f_1\beta_1 + (1-f_1)\beta_2$$

6.5.2 Turbulent Kinetic Energy Production (P_k)

Production is computed using:

$$P_k = \texttt{rev} \cdot S \cdot \frac{1}{\rho}$$

and limited by:

$$P_k \leftarrow \min(P_k, \texttt{pklim} \cdot D_k)$$

where $D_k = \beta^* k\omega$.

6.5.3 Transition Model Activation

If `transitionModel == gammaRetheta`, the following steps are added:

- Reynolds number based on wall distance and vorticity are computed.
- Wake and transitional suppression functions:

$$F_{\text{wake}} = e^{-(\text{Re}_w/10^5)^2}, \quad F_{\theta t} = \min\left(\max\left(F_{\text{wake}} e^{-(y/\delta)^4}, 1 - \left(\frac{c\gamma - 1}{c-1}\right)^2\right), 1\right)$$

- Critical Reynolds number Re_{θ_c} is piecewise evaluated.
- Separation-induced intermittency:

$$\gamma_{\text{sep}} = \min\left(s_1 \max\left(0, \frac{\text{Re}_V}{3.235\text{Re}_{\theta c}} - 1\right) F_{\text{reattach}}, 2\right) F_{\theta t}$$

- Effective intermittency:

$$\gamma_{\text{eff}} = \max(\gamma, \gamma_{\text{sep}})$$

The production/destruction terms are then scaled by γ_{eff}:

$$P_k \leftarrow \gamma_{\text{eff}} P_k, \quad D_k \leftarrow \max(0.1, \min(\gamma_{\text{eff}}, 1)) D_k$$

6.5.4 Final Source Terms

$$S_k = P_k - D_k$$

$$S_\omega = \gamma \frac{P_k}{\texttt{rev}} + \texttt{crossDiffusion} - \beta\omega^2$$

6.5.5 Jacobian Entries (approximate)

Only source terms are differentiated:

$$\frac{\partial S_k}{\partial k} = \beta^* \omega, \quad \frac{\partial S_\omega}{\partial \omega} = 2\beta\omega$$

These are stored in the local block Jacobian array `qq(i,j,k,*,*)`.

6.5.6 AD-Compatible Looping

The loop is written to support both forward and reverse automatic differentiation (Tapenade) using loop macros around the main update block.

6.6 **`f1SST`** Subroutine: SST Blending Function f_1 Computation with Transition Model Support in the module **`SST`**

`f1SST` computes the f_1 blending function for the Menter SST turbulence model over all owned cells and first-layer halos in the computational domain. It also calculates the cross-diffusion term, stored in `scratch(:,:,:,icd)`, used later in the main SST solver.

6.6.1 Computation Range Setup

The update range $[\mathrm{iBeg} : \mathrm{iEnd}, \mathrm{jBeg} : \mathrm{jEnd}, \mathrm{kBeg} : \mathrm{kEnd}]$ is adjusted based on boundary conditions. For each boundary face, the owned domain range is restricted to exclude halo layers.

6.6.2 Cross-Diffusion Term and f_1 Evaluation

Let:

$$t_1 = \begin{cases} \frac{\sqrt{k}}{0.09\omega y} & \text{if } k > 0 \\ 0 & \text{otherwise} \end{cases}, \quad t_2 = \frac{500\nu}{\rho\omega y^2}$$

Then the maximum value is computed via:

$$t_1 \leftarrow \max(t_1, t_2)$$

Next, the cross-diffusion based term is:

$$t_2 = \frac{2k}{\max(\epsilon, CD) \cdot y^2}$$

where CD = `scratch(:,:,:,icd)`, and ϵ is either a small constant or a function of density depending on `use2003SST`.

A smooth minimum is applied:

$$\texttt{arg1} = \min(t_1, t_2)$$

$$f_1 = \tanh(\texttt{arg1}^4)$$

6.6.3 Transition Model Modification

If `transitionModel == gammaRetheta`, the value of f_1 is further modified by:

$$\mathrm{Re}_y = \frac{\rho y \sqrt{k}}{\nu}, \quad f_3 = \exp\left(-\left(\frac{\mathrm{Re}_y}{120}\right)^8\right)$$

$$f_1 \leftarrow \max(f_1, f_3)$$

6.6.4 Halo Cell Update (Neumann BC)

The halo values for f_1 are set by copying interior values using Neumann-type extrapolation:

$$f_1|_{\text{halo}} = f_1|_{\text{adjacent interior}}$$

This applies for all domain boundaries (`iMin`, `iMax`, etc.).

6.6.5 Automatic Differentiation Support

The subroutine is compatible with Tapenade reverse mode via loop macros, supporting vectorized loop traversal over the 3D domain:

`#ifdef TAPENADE_REVERSE` uses 1D indexing over the full 3D range

6.6.6 Output

The computed values of f_1 are stored in:

$$\texttt{scratch(i,j,k,if1SST)} = f_1$$

to be used in the SST source term evaluation (`SSTSource`) and coefficient blending.

6.7 Wall Boundary Condition Treatment for `gamma-Retheta` Transition Model in the module `TurbBCRoutines`

This section in the module `TurbBCRoutines` imposes Neumann (zero normal derivative) boundary conditions on the transition model variables γ and Re_{θ_t} at solid wall boundaries. These conditions are essential for ensuring that no artificial fluxes enter or exit the domain through the wall, maintaining physical consistency for transition onset prediction.

The implementation is activated only when the transition model is set to `gammaRetheta`. For each block boundary face, identified by `BCFaceID(nn)`, the solver selects the appropriate direction (`iMin`, `iMax`, `jMin`, `jMax`, `kMin`, or `kMax`). For each face, the solver loops over the corresponding face-aligned index ranges (`icBeg:icEnd` and `jcBeg:jcEnd`) defined in `BCData(nn)`.

For a given face direction, the ghost-cell values of γ and Re_{θ_t} (indexed by `iTransition1` and `iTransition2`, respectively) are set equal to the adjacent interior cell values. This is implemented using the appropriate face-normal derivative Jacobian arrays:

- `bmti1`, `bmti2` for `iMin`, `iMax` (streamwise faces)
- `bmtj1`, `bmtj2` for `jMin`, `jMax` (spanwise faces)
- `bmtk1`, `bmtk2` for `kMin`, `kMax` (normal/vertical faces in 3D)

For example, on the `iMin` face, the boundary values are set as:

```
bmti1(i, j, iTransition1, iTransition1) = &
     bmti1(i+1, j, iTransition1, iTransition1)
bmti1(i, j, iTransition2, iTransition2) = &
     bmti1(i+1, j, iTransition2, iTransition2)
```

This enforces $\partial\gamma/\partial \mathrm{n} = 0$ and $\partial \mathrm{Re}_{\theta_t}/\partial \mathrm{n} = 0$ by reflecting the adjacent interior value across the wall.

This pattern is repeated for each face direction:

- Copy from `i+1` on `iMin`, from `i-1` on `iMax`
- Copy from `j+1` on `jMin`, from `j-1` on `jMax`
- Copy from `k+1` on `kMin`, from `k-1` on `kMax`

This logic is essential for correctly applying wall boundary conditions to the transition model and must be kept in sync with both the residual computation and any automatic differentiation routines.

6.8 Freestream Boundary Conditions for Transition Model in the module **`initializeFlow`**

When the `gammaRetheta` transition model is active, `initializeFlow` sets the appropriate freestream values for the transition variables at inflow and freestream boundaries. These values are initialized using empirical correlations based on the freestream turbulence intensity.

6.8.1 Activation Condition

The logic is triggered when:

$$\texttt{transitionModel} = \texttt{gammaRetheta}$$

6.8.2 Turbulence Intensity Conversion

`turbIntensityInf` is given as a ratio (e.g., 0.02 for 2%). It is converted to percentage form for use in empirical correlations:

$$\text{Tu\_inf} = \texttt{turbIntensityInf} \times 100$$

6.8.3 Transition Variable Initialization

The transition-related freestream values are stored in the array `wInf`:

- Intermittency is set to unity:

$$\text{w\_inf}[\texttt{iTransition1}] = 1.0$$

- Transition onset momentum thickness Reynolds number, $\mathrm{Re}_{\theta \mathrm{t}}$, is computed using an empirical fit:

$$\text{w\_inf}[\texttt{iTransition2}] = \begin{cases} 331.50 \cdot (\mathrm{Tu}_\infty - 0.5658)^{-0.671}, & \text{if } \mathrm{TI}_\infty > 1.3 \\ 1173.51 - 589.428 \cdot \mathrm{Tu}_\infty + 0.2196 \cdot \mathrm{Tu}_\infty^{-2}, & \text{otherwise} \end{cases}$$

6.9 Solution Verification and Numerical Modeling Considerations

This transition model implementation is still under development and may contain errors that can propagate to the automatic differentiation (AD) process.

ADflow can handle approximately 40000 cells per gigabyte of main memory [100]. This limitation has influenced the present work, as all simulations were conducted on a machine with only 3 GB of RAM. Consequently, the finest grid that could be employed contained approximately 120000 cells. To avoid machine overheating, runtime issues and `MPI` communication errors [76], simulations were run using only two processors in parallel. As a result, each simulation required a considerable amount of time to complete.

The transition model employed in this study utilizes the calibration constants listed in Table 3.2, following the default values proposed by Langtry and Menter [22]. However, these constants are not universally transferable across different CFD solvers. Consequently, significant effort has been devoted to adapting and calibrating the model to suit the present implementation.

Compared to the fully turbulent SST model, the transition model exhibits reduced numerical robustness. It tends to produce stiffer linear systems and demonstrates slower convergence behavior, highlighting the importance of careful numerical treatment and model tuning [101, 51, 80].

One prominent numerical issue encountered involves the initial magnitude of the residuals. At startup, residuals for ω and γ can reach values as high as 10^{16}, while those for $\mathrm{Re}_{\theta_\mathrm{t}}$ are around 10^{22}. The encouraging residuals are the ones of k starting at 10^{-3}. Despite these extreme initial values, the solver typically achieves a reduction of approximately five orders of magnitude for the continuity equation residual (initially around 10^3), and reductions between six and fifteen orders of magnitude for the turbulence-related variables as seen in Figure 6.1. These levels of residual decrease generally precede the stabilization of the aerodynamic coefficients—namely, the lift ($\mathrm{C_l}$), drag ($\mathrm{C_d}$), and pitching moment ($\mathrm{C_{mz}}$) coefficients.

To improve convergence, several numerical strategies were investigated:

- Applying explicit under-relaxation to the effective intermittency variable, γ_{eff}, as done in [58], using the update formula:

  $$\gamma_{\mathrm{eff}}^{(\mathrm{n}+1)} = \gamma_{\mathrm{eff}}^{(\mathrm{n})} + \omega\left(\gamma_{\mathrm{eff}}^{\mathrm{new}} - \gamma_{\mathrm{eff}}^{(\mathrm{n})}\right)$$

  where $\omega \in (0, 1]$ is the relaxation factor, $\gamma_{\mathrm{eff}}^{(\mathrm{n})}$ is the current value, and $\gamma_{\mathrm{eff}}^{\mathrm{new}}$ is the value computed from the transition source terms.

- Testing an alternative formulation for the separation transition function, γ_{sep}, given by:

  $$\gamma_{\mathrm{sep}} = \min\left(\mathrm{s_1}\max\left[0, \frac{\mathrm{Re}_\nu}{2.193\mathrm{Re}_{\theta_\mathrm{c}}} - 1\right]\exp\left[-\left(\frac{\mathrm{R_T}}{15}\right)^4\right], 5\right)\mathrm{F}_{\theta_\mathrm{t}} \tag{6.1}$$

  Here, the constant 2.193 represents the maximum ratio between $\mathrm{Re_V}$ and $\mathrm{Re}_{\theta_\mathrm{c}}$ for a Blasius boundary layer, as introduced in the original model [101], and $\mathrm{s_1} = 2$.

- Increasing the scaling factor in Eq. (6.1) to $\mathrm{s_1} = 6$.

- Applying the modified γ_{sep} expression from Eq. (3.50) with $\mathrm{s_1} = 6$, following the approach used by Carreño Ruiz [102] to reduce excessive turbulence production in separated regions at high angles of attack, and consistent with the transition control strategy proposed by Suluksna and Juntasaro [103].

- Employing central differencing schemes for the convective terms, combined with either matrix dissipation or scalar dissipation formulations.

- Increasing artificial viscosity in the central scheme to enhance numerical stability.

- Raising the production term coefficient in the k-equation from 10 to 15, consistent with the strategy in [58], to prevent premature activation of the $\mathrm{P_k}$ limiter in turbulent regions.

- To improve the robustness and accuracy of forward-mode automatic differentiation (AD), the transition onset function $\mathrm{F_{onset}}$ is reformulated to be smooth, meaning infinitely differentiable. Discontinuities in differentiability have been shown to impair both flow and adjoint solver performance [104, 32, 80, 19].

  To enforce a steep ramp-up while maintaining differentiability, the nested `max` and `min` operations are approximated using smooth exponential functions. Let $\lambda = 20$ denote a smoothness parameter, where larger values of λ produce sharper transitions:

$$\mathrm{F}_{\mathrm{onset1,softmax}} = \frac{1}{\lambda}\log\left(\mathrm{e}^{\lambda \mathrm{F}_{\mathrm{onset1}}} + \mathrm{e}^{\lambda \mathrm{F}^{4}_{\mathrm{onset1}}}\right), \tag{6.2}$$

$$\mathrm{F}_{\mathrm{onset2}} = -\frac{1}{\lambda}\log\left(\mathrm{e}^{-\lambda \mathrm{F}_{\mathrm{onset1,softmax}}} + \mathrm{e}^{-\lambda \cdot 2.0}\right). \tag{6.3}$$

To model the damping effect due to increasing turbulence, a smooth approximation of the ReLU (Rectified Liner Unit) function [105] is applied:

$$\mathrm{F}_{\mathrm{onset3}} = \frac{1}{\lambda}\log\left(1 + \mathrm{e}^{\lambda\left(1-\left(\frac{\mathrm{R_T}}{2.5}\right)^{3}\right)}\right). \tag{6.4}$$

The final smooth expression for the onset function is then given by:

$$\mathrm{F}_{\mathrm{onset}} = \frac{1}{\lambda}\log\left(1 + \mathrm{e}^{\lambda(\mathrm{F}_{\mathrm{onset2}}-\mathrm{F}_{\mathrm{onset3}})}\right). \tag{6.5}$$

The version of this code can be found in [106]. Further investigation is warranted to improve convergence behavior and numerical stability. It is likely that other empirical correlation functions in the transition model may also benefit from similar smoothing strategies, and corresponding differentiable formulations should be developed.

Despite these efforts, none of the tested strategies yielded satisfactory convergence. As a result, the original formulation of the model—described in Chapter 3.4—was retained without modification.

The simulation was halted after a considerable number of iterations, and a restart-based approach was adopted. This involved overwriting the solution at each step using restart files. Consequently, a complete convergence history is not available.

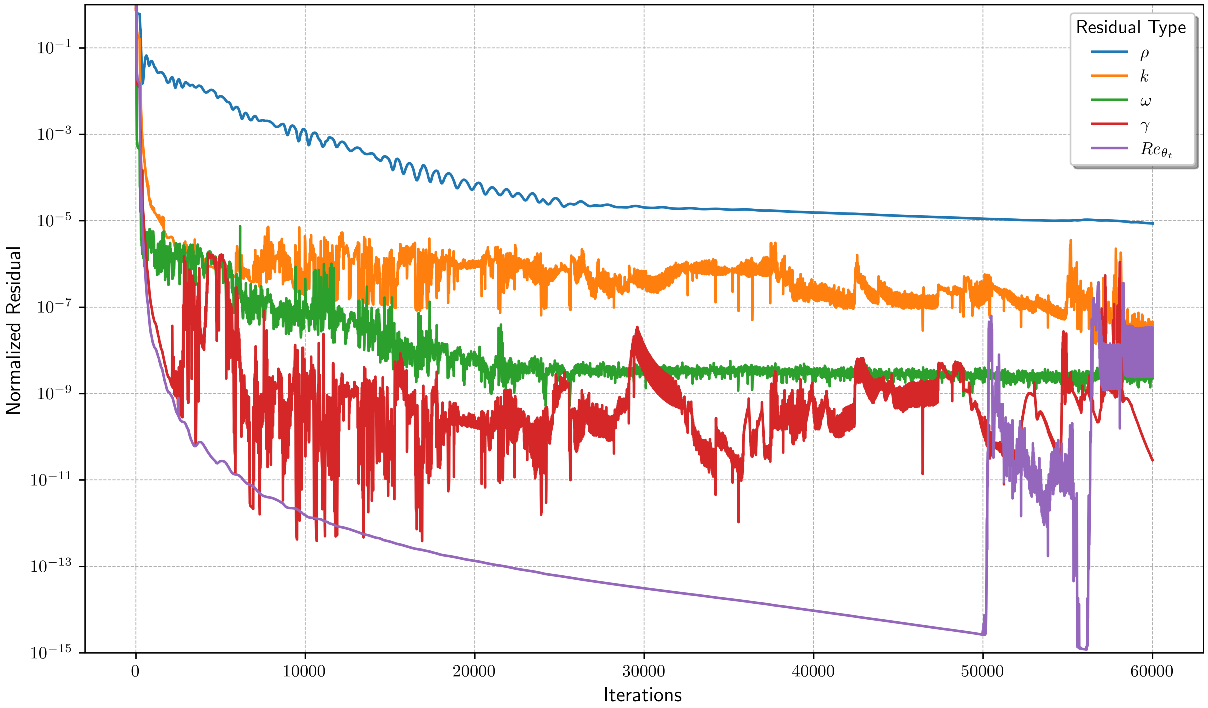

Figure 6.1: Residuals (normalized with respect to the mean over the first five iterations) for the finest mesh level (L0) used in the simulation of the SD7003 airfoil at Re = 60000 and $\alpha = 4°$.

A reference simulation by Carreño Ruiz, using the empirical correlations from [102], was performed with the commercial CFD solver *STAR-CCM+* [107]. The simulation employed the finest mesh level (L0) for the SD7003 airfoil at Re = 60000 and $\alpha = 4°$. The results were compared with numerical data from CIRA (Centro Italiano Ricerche Aerospaziali) to aid in the calibration and validation of the implemented transition model.

While the mesh used may be sufficient for simulations and sensitivity analysis based on finite differences (FD), automatic differentiation (AD) introduces stricter requirements on smoothness and numerical consistency [19, 80]. Even though non-differentiable expressions such as `min` and `max` are replaced with alternatives (`if-else` logic) in the Tapenade-generated code, the AD-generated derivatives can still be more sensitive to local solution irregularities, particularly in transitional flow regions. Therefore, to ensure consistency between FD and AD results and to obtain accurate derivative information, finer grids may be necessary when using AD with transition models [104, 32, 80].

6.10 Jacobian - Vector Product Verification, Forward Mode AD

Once the model is fully implemented—or when major modifications are introduced—certain Fortran routines (such as those in the turbulence or transition model modules) must be differentiated using Tapenade to obtain forward-mode automatic derivatives (AD). The resulting differentiated code is required by the ANK and NK solvers used for the flow analysis. While finite-difference (FD) derivatives are generally sufficient for post-converged sensitivity analysis, they are not reliable within the flow solver for achieving convergence in complex models like transition models. However, they ensure convergence for simpler models like Spalart–Allmaras (SA). Therefore, verifying the correctness of the AD-generated derivatives is critical for solver stability and accuracy.

A Python script for verifying Jacobian–vector products using forward mode automatic differentiation (AD) has been developed and is available in the author's associated repository [108]. The verification procedure applies perturbations to individual input parameters following the simulation and computes the corresponding sensitivities. This process was conducted for the SD7003 airfoil at a Reynolds number of $\mathrm{Re} = 60000$ and an angle of attack of $\alpha = 4^\circ$.

The following table compares derivatives computed using Automatic Differentiation (AD) against Finite Difference (FD) methods for various perturbations and design variables.

Perturbations include:

- $\dot{\mathrm{w}}$: perturbation of mean flow and turbulent variables,
- $\dot{\mathrm{x}}_{\mathrm{V}}$: perturbation of volume mesh,
- $\alpha, \beta, \mathrm{Mach}, \mathrm{P}, \mathrm{T}, \mathrm{x}_{\mathrm{ref}}, \mathrm{y}_{\mathrm{ref}}$: perturbation of geometric and flow design variables (DV).

The table shows norms of residual derivatives ($\|\partial \mathbf{R}/\partial \cdot\|$) and surface force derivatives ($\|\partial \mathbf{F}/\partial \cdot\|$), derivatives of functions such as drag ($\mathrm{C_d}$), lift ($\mathrm{C_l}$) and moment coefficients ($\mathrm{C_{M_z}}$) along with their relative errors. The relative error between results obtained using automatic differentiation (AD) and those computed via finite difference (FD) is defined as:

$$\mathrm{rel\_error(a, b)} = \frac{\|\mathrm{a} - \mathrm{b}\|}{\max(\|\mathrm{a}\|, \varepsilon)}, \tag{6.6}$$

where a denotes the vector of derivatives obtained from AD, and b represents the corresponding vector from FD. The norm $\|\cdot\|$ refers to the Euclidean norm (also known as the L_2 norm), which provides a scalar measure of vector magnitude. The small constant ε (typically set to 10^{-16}) ensures numerical stability by preventing division by zero when $\|\mathrm{a}\|$ is very small or vanishes. This formulation ensures a robust comparison even when derivative magnitudes are close to machine precision.

Table 6.1: Comparison of AD and FD derivatives across perturbations and design variables.

Quantity	**Value (AD)**	**Value (FD)**	**Relative Error**
Perturbation: $\dot{\mathrm{w}}$			
$\|\partial \mathrm{R}/\partial \mathrm{w} \cdot \dot{\mathrm{w}}\|$	1.77×10^{19}	4.00×10^{23}	0.99999998
$\partial \mathrm{C_d}/\partial \mathrm{w} \cdot \dot{\mathrm{w}}$	629.3754	629.3754	7.10×10^{-10}
$\partial \mathrm{C_l}/\partial \mathrm{w} \cdot \dot{\mathrm{w}}$	723.0253	723.0253	6.18×10^{-9}
$\mathrm{dC_{M_z}}/\mathrm{dw} \cdot \dot{\mathrm{w}}$	385.8471	385.8471	4.97×10^{-9}

Continued on next page

Table 6.1 – continued from previous page

Quantity	Value (AD)	Value (FD)	Relative Error
$\|\partial \mathrm{F}/\partial \mathrm{w} \cdot \dot{\mathrm{w}}\|$	2.88×10^{5}	2.88×10^{5}	2.61×10^{-8}
Perturbation: $\dot{\mathrm{x}}_\mathrm{V}$			
$\|\partial \mathrm{R}/\partial \mathrm{x}_\mathrm{V} \cdot \dot{\mathrm{x}}_\mathrm{V}\|$	1.38×10^{25}	1.38×10^{25}	1.37×10^{-3}
$\partial \mathrm{C_d}/\partial \mathrm{x}_\mathrm{V} \cdot \dot{\mathrm{x}}_\mathrm{V}$	-90.2438	-90.0360	2.30×10^{-3}
$\partial \mathrm{C_l}/\partial \mathrm{x}_\mathrm{V} \cdot \dot{\mathrm{x}}_\mathrm{V}$	-164.7931	-164.6824	6.72×10^{-4}
$\partial \mathrm{C_{M_z}}/\partial \mathrm{x}_\mathrm{V} \cdot \dot{\mathrm{x}}_\mathrm{V}$	14.0310	14.0586	1.96×10^{-3}
$\|\partial \mathrm{F}/\partial \mathrm{x}_\mathrm{V} \cdot \dot{\mathrm{x}}_\mathrm{V}\|$	3.18×10^{6}	3.18×10^{6}	5.02×10^{-4}
Perturbation on DV: α			
$\|\mathrm{dR}/\mathrm{d}\alpha\|$	0.04915	0.04915	3.27×10^{-5}
$\mathrm{dC_d}/\mathrm{d}\alpha$	1.44×10^{-6}	1.44×10^{-6}	1.01×10^{-8}
$\mathrm{dC_l}/\mathrm{d}\alpha$	-4.53×10^{-8}	-4.53×10^{-8}	1.68×10^{-5}
$\mathrm{dC_{M_z}}/\mathrm{d}\alpha$	0	0	0
$\|\mathrm{dF}/\mathrm{d}\alpha\|$	0	0	0
Perturbation on DV: β			
$\|\mathrm{dR}/\mathrm{d}\beta\|$	1.29×10^{-7}	1.88×10^{-6}	0.9977
$\mathrm{dC_d}/\mathrm{d}\beta$	-7.37×10^{-18}	0	7.37×10^{-4}
$\mathrm{dC_l}/\mathrm{d}\beta$	7.75×10^{-19}	0	7.75×10^{-5}
$\mathrm{dC_{M_z}}/\mathrm{d}\beta$	0	0	0
$\|\mathrm{dF}/\mathrm{d}\beta\|$	0	0	0
Perturbation on DV: Mach			
$\|\mathrm{dR}/\mathrm{dMach}\|$	10.0428	10.0428	1.85×10^{-6}
$\mathrm{dC_d}/\mathrm{dMach}$	-5.19×10^{-6}	-5.19×10^{-6}	4.06×10^{-7}
$\mathrm{dC_l}/\mathrm{dMach}$	-1.65×10^{-4}	-1.65×10^{-4}	1.41×10^{-7}
$\mathrm{dC_{M_z}}/\mathrm{dMach}$	-4.86×10^{-5}	-4.86×10^{-5}	1.68×10^{-7}
$\|\mathrm{dF}/\mathrm{dMach}\|$	0	0	0
Perturbation on DV: P			
$\|\mathrm{dR}/\mathrm{dP}\|$	1.21×10^{14}	1.21×10^{14}	2.21×10^{-6}
$\mathrm{dC_d}/\mathrm{dP}$	7.04×10^{-8}	7.04×10^{-8}	2.77×10^{-6}
$\mathrm{dC_l}/\mathrm{dP}$	1.83×10^{-8}	1.83×10^{-8}	3.00×10^{-6}
$\mathrm{dC_{M_z}}/\mathrm{dP}$	2.48×10^{-9}	2.48×10^{-9}	3.16×10^{-6}
$\|\mathrm{dF}/\mathrm{dP}\|$	6.96×10^{-3}	6.96×10^{-3}	2.00×10^{-8}
Perturbation on DV: T			
$\|\mathrm{dR}/\mathrm{dT}\|$	1.12×10^{-3}	5.91×10^{7}	1.00
$\mathrm{dC_d}/\mathrm{dT}$	2.05×10^{-25}	0	2.05×10^{-11}

Continued on next page

Table 6.1 – continued from previous page

Quantity	**Value (AD)**	**Value (FD)**	**Relative Error**
$\mathrm{dC_l/dT}$	6.55×10^{-26}	0	6.55×10^{-12}
$\mathrm{dC_{M_z}/dT}$	7.04×10^{-27}	0	7.04×10^{-13}
$\mathrm{\|dF/dT\|}$	5.91×10^{-22}	5.20×10^{-11}	1.00
Perturbation on DV: $\mathrm{x_{ref}}$			
$\mathrm{\|dR/dx_{ref}\|}$	0	0	0
$\mathrm{dC_d/dx_{ref}}$	0	0	0
$\mathrm{dC_l/dx_{ref}}$	0	0	0
$\mathrm{dC_{M_z}/dx_{ref}}$	-8.22×10^{-6}	-8.22×10^{-6}	3.06×10^{-6}
$\mathrm{\|dF/dx_{ref}\|}$	0	0	0
Perturbation on DV: $\mathrm{y_{ref}}$			
$\mathrm{\|dR/dy_{ref}\|}$	0	0	0
$\mathrm{dC_d/dy_{ref}}$	0	0	0
$\mathrm{dC_l/dy_{ref}}$	0	0	0
$\mathrm{dC_{M_z}/dy_{ref}}$	-6.03×10^{-7}	-6.03×10^{-7}	1.04×10^{-5}
$\mathrm{\|dF/dy_{ref}\|}$	0	0	0

The AD-derived derivatives exhibit excellent agreement with finite difference (FD) results for aerodynamic coefficients ($\mathrm{C_d}$, $\mathrm{C_l}$, $\mathrm{C_{M_z}}$) and force-related quantities across most perturbations and design variables. Relative errors for these quantities are generally on the order of 10^{-8} to 10^{-5}, indicating that AD correctly captures the sensitivities.

However, the residual norm sensitivities computed by AD show significant deviations from FD, with relative errors approaching unity in some cases (e.g., perturbation $\dot{\mathrm{w}}$ and variable β). This discrepancy suggests potential issues in the implementation or numerical stability of the AD forward mode for residuals, possibly due to scale differences or nonlinearities in the solver.

Furthermore, derivatives with respect to certain variables, such as freestream temperature (T), reveal large relative errors or near-zero FD values, implying that AD gradients for these variables are currently unreliable.

Overall, while AD demonstrates strong performance for aerodynamic function derivatives, caution is warranted for residual sensitivities and specific design variables. Further investigation and refinement of the AD implementation are necessary to ensure consistent and robust gradient computation across all components of the solver.

After verifying forward-mode AD derivatives, selected routines—particularly those not supported by Tapenade [68, 70]—will be manually differentiated in reverse mode to enable efficient gradient-based aerodynamic shape optimization within the existing framework.

6.11 Parametric study

To align transition model predictions with results from Large Eddy Simulations (LES), a parametric study involving the freestream turbulence intensity (TI_{∞}) and the freestream eddy viscosity ratio $(\mu_{\mathrm{t}}/\mu)_{\infty}$ is essential. Due to the inherent differences in modeling fidelity between Reynolds-Averaged Navier–Stokes (RANS) methods and LES—particularly in capturing surface pressure coefficients (C_{p}) and skin friction coefficients (C_{f})—significant discrepancies can arise. These differences are well-documented in the literature [109, 110, 111, 112], and are observed even among RANS-based transition models [102, 113], particularly in flows involving laminar-to-turbulent transition and flow separation.

In ADflow, the SST turbulence model uses a default value of $(\mu_{\mathrm{t}}/\mu)_{\infty} = 0.009$. However, Rumsey and Spalart [114] suggest setting this parameter as $2 \times 10^{-7} \times \mathrm{Re}$, a recommendation aimed at simulating transition in the absence of an explicit transition model. This approach manipulates the freestream turbulence characteristics such that turbulence production remains inactive near the leading edge and becomes effective downstream, thereby emulating transitional flow behavior. In light of this, a parametric sweep over $(\mu_{\mathrm{t}}/\mu)_{\infty}$ values is warranted to assess the model's sensitivity to this parameter.

Regarding TI_{∞}, Schlichting and Gersten [33] report that values below 0.1% have negligible effects on boundary layer development, based on experimental findings. Conversely, a minimum threshold of $\mathrm{TI}_{\infty} = 0.027\%$ ensures numerical stability in transition model simulations (see Section 3.4.7). Therefore, selecting TI_{∞} within the range of 0.027% to 0.1% is justified, particularly for comparison with LES studies. Although the empirical transition onset correlation $\mathrm{Re}_{\theta_{\mathrm{t}}} = \mathrm{f}(\mathrm{Tu}, \lambda_{\theta})$ is known to be sensitive within this interval, fixing TI_{∞} does not undermine the model's general validity, given that the correlation was calibrated on a broad set of experimental data with varying turbulence levels [22].

Initial simulations using $\mathrm{TI}_{\infty} = 0.1\%$—prior to the implementation of input control for this parameter, as discussed in Section 6.1—produced elevated lift coefficients, while drag coefficients remained consistent with reference data. Based on these outcomes, a further parametric investigation over the interval $\mathrm{TI}_{\infty} \in [0.03\%, 0.08\%]$ will be conducted. The goal is to determine a representative fixed value for TI_{∞} to be used in subsequent simulations.

Figure 6.2 shows the pressure and skin-friction distributions from the parametric study. In the C_{p} plot, flow instabilities appear near the typical transition points of the validated models, with an overshoot followed by an undershoot just downstream of the transition predicted by the present model. All ADflow simulations predict a delayed transition onset compared to fully validated results from STAR-CCM+ [107] and the CIRA LES and k–ω–γ models [113].

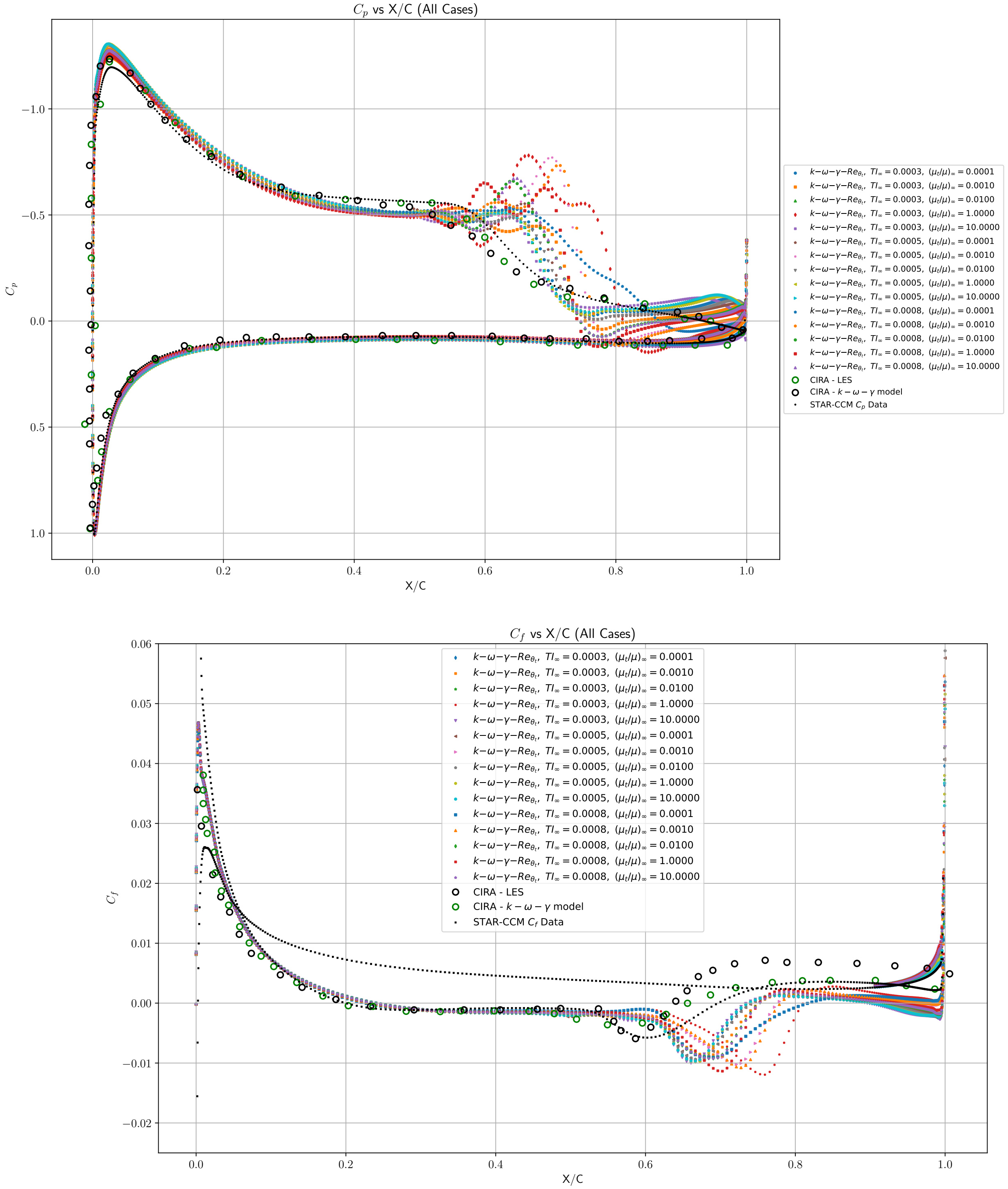

Figure 6.2: Distributions of pressure coefficient (C_p) and skin friction coefficient (C_f) over the SD7003 airfoil at Re = 60000, $\alpha = 4°$, comparing experimental measurements with numerical model predictions.

Table 6.2: Reference aerodynamic coefficients at $\alpha = 4°$

Source	C_l	C_d
LES (CIRA)	0.6273	0.0233
k–ω–γ (CIRA)	0.5993	0.0214
k–ω–γ–Re_{θ_t} (STAR-CCM+)	0.5940	0.0228

Table 6.3: Parametric study at Re = 60000, $\alpha = 4°$

Case	C_l	C_d
$TI_\infty = 0.0003$, $(\mu_t/\mu)_\infty = 0.0001$	0.6342	0.0196
$TI_\infty = 0.0003$, $(\mu_t/\mu)_\infty = 0.001$	0.6139	0.0190
$TI_\infty = 0.0003$, $(\mu_t/\mu)_\infty = 0.01$	0.6371	0.0196
$TI_\infty = 0.0003$, $(\mu_t/\mu)_\infty = 1.0$	0.6529	0.0258
$TI_\infty = 0.0003$, $(\mu_t/\mu)_\infty = 10.0$	0.6398	0.0195
$TI_\infty = 0.0005$, $(\mu_t/\mu)_\infty = 0.0001$	0.6256	0.0191
$TI_\infty = 0.0005$, $(\mu_t/\mu)_\infty = 0.001$	0.6185	0.0229
$TI_\infty = 0.0005$, $(\mu_t/\mu)_\infty = 0.01$	0.6245	0.0188
$TI_\infty = 0.0005$, $(\mu_t/\mu)_\infty = 1.0$	0.6328	0.0193
$TI_\infty = 0.0005$, $(\mu_t/\mu)_\infty = 10.0$	0.6375	0.0196
$TI_\infty = 0.0008$, $(\mu_t/\mu)_\infty = 0.0001$	0.6366	0.0246
$TI_\infty = 0.0008$, $(\mu_t/\mu)_\infty = 0.001$	**0.6223**	**0.0232**
$TI_\infty = 0.0008$, $(\mu_t/\mu)_\infty = 0.01$	0.6084	0.0190
$TI_\infty = 0.0008$, $(\mu_t/\mu)_\infty = 1.0$	0.6045	0.0203
$TI_\infty = 0.0008$, $(\mu_t/\mu)_\infty = 10.0$	0.6091	0.0191

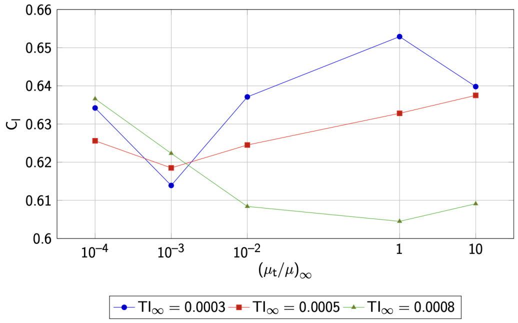

Figure 6.3: C_l variation with freestream eddy viscosity ratio for different turbulence intensities for SD7003 at Re = 60000, $\alpha = 4°$

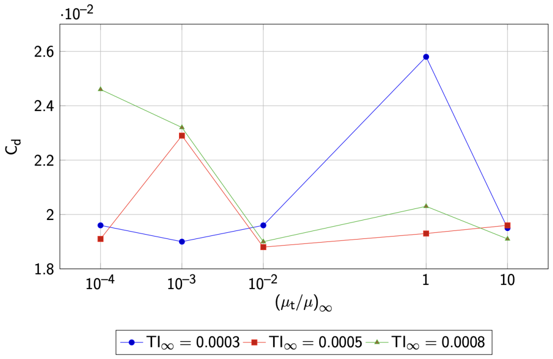

Figure 6.4: C_d variation with freestream eddy viscosity ratio for different turbulence intensities for SD7003 at Re = 60000, $\alpha = 4°$

A clear trend emerges from the parametric study: increasing the free-stream eddy viscosity ratio $(\mu_\mathrm{t}/\mu)_\infty$ generally leads to an increase in the lift coefficient $\mathrm{C_l}$, particularly at the lowest turbulence intensity $\mathrm{TI}_\infty = 0.0003$. This behavior aligns with established transition physics, where elevated ambient turbulence or eddy viscosity accelerates transition, reduces laminar separation, and enhances lift production [22]. However, the trend is not strictly monotonic. For example, at $(\mu_\mathrm{t}/\mu)_\infty = 1.0$, $\mathrm{C_l}$ peaks at 0.6529, but further increasing the eddy viscosity ratio to 10.0 leads to a plateau or slight reduction in lift, indicating diminishing returns due to increased skin-friction drag from premature transition.

The drag coefficient $\mathrm{C_d}$ shows a more complex, non-monotonic trend due to the competing effects of reduced pressure drag (via earlier transition) and increased skin-friction drag. For several cases, modest increases in $(\mu_\mathrm{t}/\mu)_\infty$ reduce drag, but excessive values can reverse this benefit. At $\mathrm{TI}_\infty = 0.0008$, for example, drag is minimized at $(\mu_\mathrm{t}/\mu)_\infty = 0.01$, but this comes at the cost of reduced lift. In contrast, the combination $\mathrm{TI}_\infty = 0.0008$ and $(\mu_\mathrm{t}/\mu)_\infty = 0.001$ offers a strong aerodynamic balance, achieving:

$$\mathrm{C_l} = 0.6223, \quad \mathrm{C_d} = 0.0232,$$

which represents high lift and moderate drag. These values closely match both LES benchmarks and results from the CIRA k–ω–γ model—within 0.8% and 0.5% of the LES values, respectively—and are in good agreement with STAR-CCM+ predictions. Therefore, this parameter combination is adopted for all subsequent simulations to ensure consistency and reliable model validation.

6.12 Mesh convergence analysis

The mesh convergence study for the SD7003 airfoil at $\mathrm{Re} = 60000$ and $\alpha = 4^\circ$ reveals contrasting sensitivities between the aerodynamic coefficients. As shown in Figure 6.5 and Tables 6.4 and 6.5, the drag coefficient $\mathrm{C_d}$ shows good grid convergence, with a relatively high estimated order of accuracy (the theoretical value is 2) and low GCI values. In contrast, the lift coefficient $\mathrm{C_l}$ exhibits poor convergence behavior, with large variations across mesh levels and high GCI values. This suggests that $\mathrm{C_l}$ is significantly more sensitive to mesh resolution, likely due to its stronger dependence on accurately capturing boundary-layer and separation features. For this reason, all computations are performed using the finest grid that is computationally feasible.

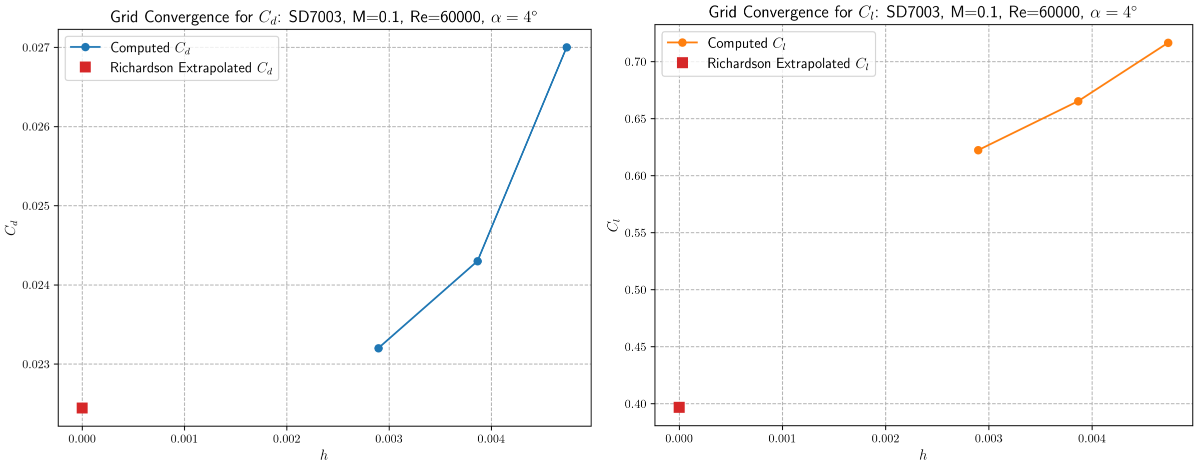

Figure 6.5: Drag and lift coefficients computed on three grids with different refinement levels — L0 (finest), L1, and L2 (coarsest) — along with the Richardson extrapolated value.

Table 6.4: Grid convergence for drag coefficient (C_d).

Quantity	**Value**
Cells (L0/L1/L2)	119201 / 66901 / 44551
Grid refinement ratio (r)	1.6357
C_d (L0/L1/L2)	0.0232 / 0.0243 / 0.0270
Achieved order (p)	1.8248
Richardson extrapolated C_d	0.0224
GCI L0–L1	3.11%
GCI L1–L2	6.88%

Table 6.5: Grid convergence for lift coefficient (C_l).

Quantity	**Value**
Cells (L0/L1/L2)	119201 / 66901 / 44551
Grid refinement ratio (r)	1.6357
C_l (L0/L1/L2)	0.6223 / 0.6653 / 0.7165
Achieved order (p)	0.3547
Richardson extrapolated C_l	0.3968
GCI L0–L1	33.89%
GCI L1–L2	37.47%

As illustrated in Figure 6.6, while the model is not yet fully verified (the AD gradients are not fully verified), the use of coarser grid resolutions results in a noticeable delay in transition onset, accompanied by a broader transition region that shifts downstream toward the trailing edge. This trend aligns with the findings reported by Langtry and Menter [22]. Further evidence is provided in Figures 6.7 and 6.8, which show a more diffuse distribution of turbulent kinetic energy, consistent with a larger and more delayed laminar separation bubble.

For the other cases analyzed, the isocontours can be found in Appendix A.

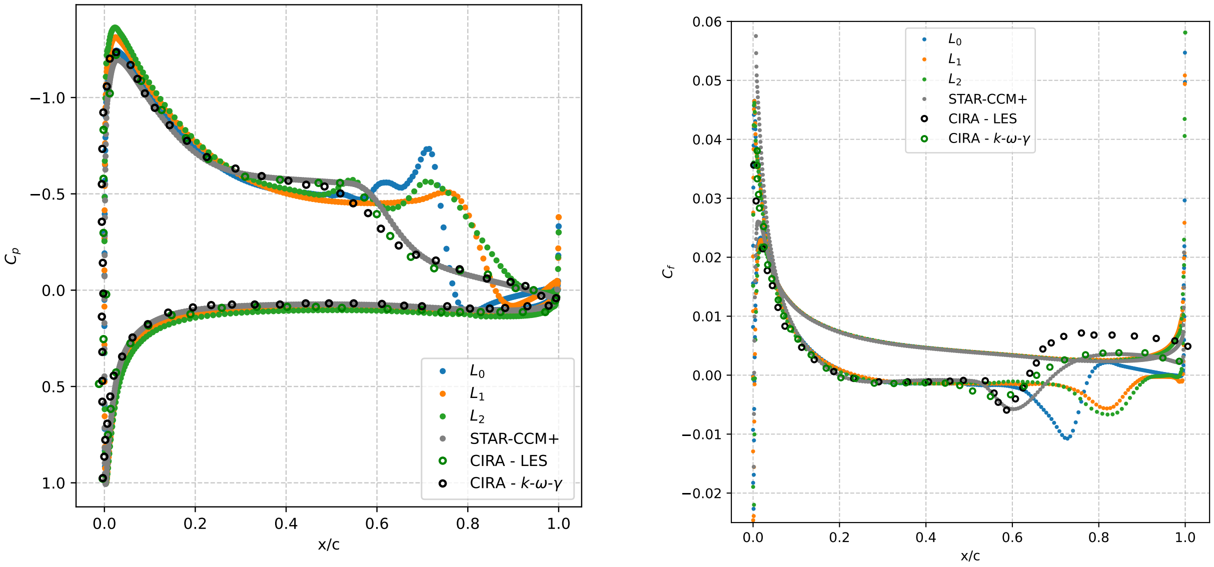

Figure 6.6: Distributions of pressure coefficient (C_p) and skin friction coefficient (C_f) over the SD7003 airfoil at Re = 60000, $\alpha = 4°$, comparing experimental measurements with numerical model predictions.

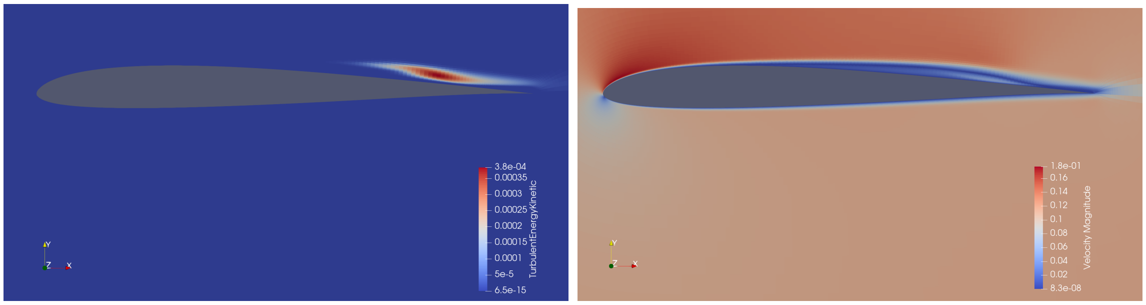

Figure 6.7: SD7003 — M = 0.1, Re = 6×10^4, $\alpha = 4°$, medium grid (`SD7003_L1`): k/U_∞^2 (left), U/U_∞ (right).

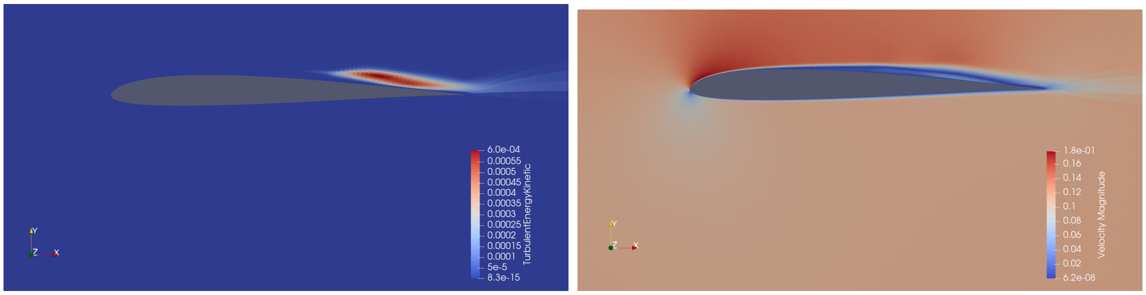

Figure 6.8: SD7003 — M = 0.1, Re = 6×10^4, $\alpha = 4°$, coarse grid (`SD7003_L2`): k/U_∞^2 (left), U/U_∞ (right).

Chapter 7

Results Part 2: Model Validation

The airfoil simulations are carried out under the conditions specified in Section 5.6. A reference temperature of 288 K is used for all cases to ensure consistent flow conditions. For each configuration, an additional simulation in ADflow is performed using the SST turbulence model to clearly highlight differences in transition location when a fully turbulent model is considered.

7.1 SD7003

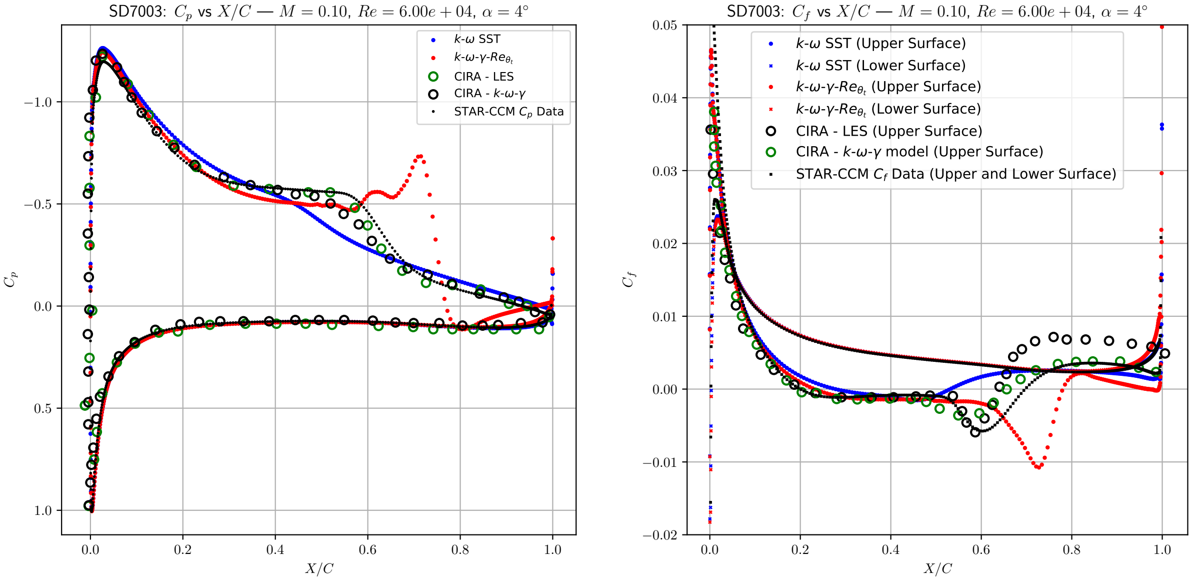

Figure 7.1: Distributions of pressure coefficient (C_p) and skin friction coefficient (C_f) over the SD7003 airfoil at $Re = 60000$, $\alpha = 4°$, comparing the developed k–ω–γ–Re_{θ_t} model with numerical model predictions [113, 107].

Reference	C_l	C_d	ε_{C_l} (%)	ε_{C_d} (%)	$x_{separation}/c$	$x_{transition}/c$	$x_{reattachment}/c$
LES (CIRA [113])	0.6273	0.0233	-	-	0.21	0.59	0.64
k–ω–γ (CIRA [113])	0.5993	0.0214	4.46	8.15	0.19	0.56	0.65
k–ω–γ–Re_{θ_t} (STAR-CCM+ [107])	0.5940	0.0228	5.31	2.15	0.20	0.60	0.70
k–ω–γ–Re_{θ_t} (this work)	0.6223	0.0232	0.80	0.43	0.22	0.72	0.78

Table 7.1: Comparison and relative error (with respect to LES) for the SD7003 airfoil at Re = 60000, $\alpha = 4°$.

The LES data provide a high-fidelity benchmark for assessing the transitional flow behavior over the SD7003 airfoil at $Re = 60000, \alpha = 4°$. The reference case reveals a typical laminar separation bubble with separation, transition, and reattachment occurring in close succession along the aft portion of the airfoil.

The k–ω–γ model underpredicts both lift and drag when compared to LES. This discrepancy is linked to its tendency to anticipate separation and transition earlier than observed in the reference, resulting in a compressed separation bubble. Although reattachment occurs in a comparable region, the premature transition limits the model's ability to capture the laminar flow regime accurately, particularly in the presence of mild adverse pressure gradients.

The STAR-CCM+ implementation of the k–ω–γ–Re_{θ_t} model improves drag prediction but still yields an underprediction in lift. Transition onset is predicted close to the LES location, but reattachment is delayed, leading to an over-extended bubble. This suggests a more gradual transition process, which may diffuse the instability growth and lead to a less abrupt reattachment. The overall aerodynamic performance is somewhat improved compared to the simpler transition model but still deviates from the LES reference.

In contrast, the k–ω–γ–Re_{θ_t} model developed in this work produces aerodynamic coefficients that closely match those of the LES, with minimal error in both lift and drag. It predicts a slightly delayed separation point and a transition onset that occurs noticeably further downstream. This results in a longer laminar separation bubble and a reattachment point significantly aft of the reference. This behavior is further reflected in the surface pressure coefficient (C_p) distribution; in the present work, the onset of flow instabilities is observed near the transition locations predicted by the other models; however, the response is notably less smooth. A distinct overshoot in C_p appears just upstream of the transition onset, followed by a localized undershoot near reattachment. While this non-monotonic pattern may suggest the rapid growth and breakdown of instabilities within the laminar shear layer, it is likely influenced by incomplete convergence of the simulation. Such artifacts can emerge near regions of strong gradients, particularly in transitional flows where resolving sharp changes in pressure recovery is numerically challenging. Nevertheless, the presence of these features indicates the model's heightened sensitivity to transitional dynamics—capturing localized behaviors that are often suppressed or overly diffused in traditional formulations.

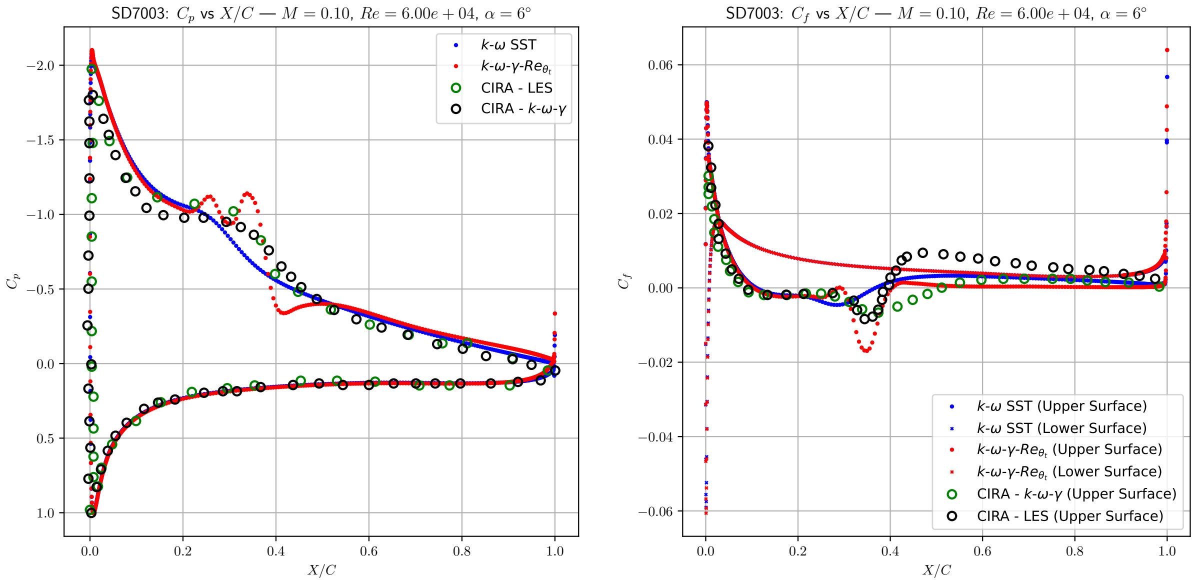

Figure 7.2: Distributions of pressure coefficient ($\mathrm{C_p}$) and skin friction coefficient ($\mathrm{C_f}$) over the SD7003 airfoil at $\mathrm{Re} = 60000$, $\alpha = 6°$, comparing the developed k–ω–γ–Re_{θ_t} model with numerical model predictions [113].

Table 7.2: Comparison of aerodynamic coefficients and relative errors with respect to LES for the SD7003 airfoil at $\mathrm{Re} = 60000$, $\alpha = 6°$.

Reference	$\mathrm{C_l}$	$\mathrm{C_d}$	$\varepsilon_{\mathrm{C_l}}$ (%)	$\varepsilon_{\mathrm{C_d}}$ (%)	$\mathrm{x_{separation}/c}$	$\mathrm{x_{transition}/c}$	$\mathrm{x_{reattachment}/c}$
LES (CIRA [113])	0.8078	0.0292	–	–	0.10	0.34	0.39
k–ω–γ (CIRA [113])	0.7828	0.0263	5.21	9.93	0.10	0.37	0.50
k–ω–γ–Re_{θ_t} (this work)	0.8236	0.0260	1.96	10.96	0.11	0.34	0.40

For the case at $\alpha = 6°$, the LES reference again provides a detailed representation of the transitional flow structure, characterized by early separation, rapid transition, and prompt reattachment. The k–ω–γ model predicts similar separation but delays transition slightly and significantly overestimates the reattachment length, resulting in a longer laminar separation bubble. This discrepancy contributes to a noticeable underprediction of both lift and drag.

The transition model developed in this work yields aerodynamic coefficients that more closely align with the LES values, particularly for lift. Separation occurs slightly later, which further supports the improved alignment with the physical flow topology. The predicted transition onset coincides with the reference, and the reattachment location is only marginally shifted, suggesting a more accurate reproduction of the laminar separation bubble dynamics.

In the surface pressure coefficient ($\mathrm{C_p}$) distribution, as in the previous case, the onset of flow instabilities is observed near the transition point identified by the LES and the other models. However, the pressure response in the present simulation is noticeably less smooth. A distinct overshoot in $\mathrm{C_p}$ appears just upstream of the transition onset, followed by a localized undershoot in the vicinity of the predicted reattachment. While this non-monotonic profile could be interpreted as a physical manifestation of instability amplification and shear-layer breakdown, it is more plausibly associated again with incomplete convergence of the simulation. Such features often arise in transitional regions with strong pressure gradients, where even small residual errors can produce local anomalies in the pressure field.

7.2 NACA 0015

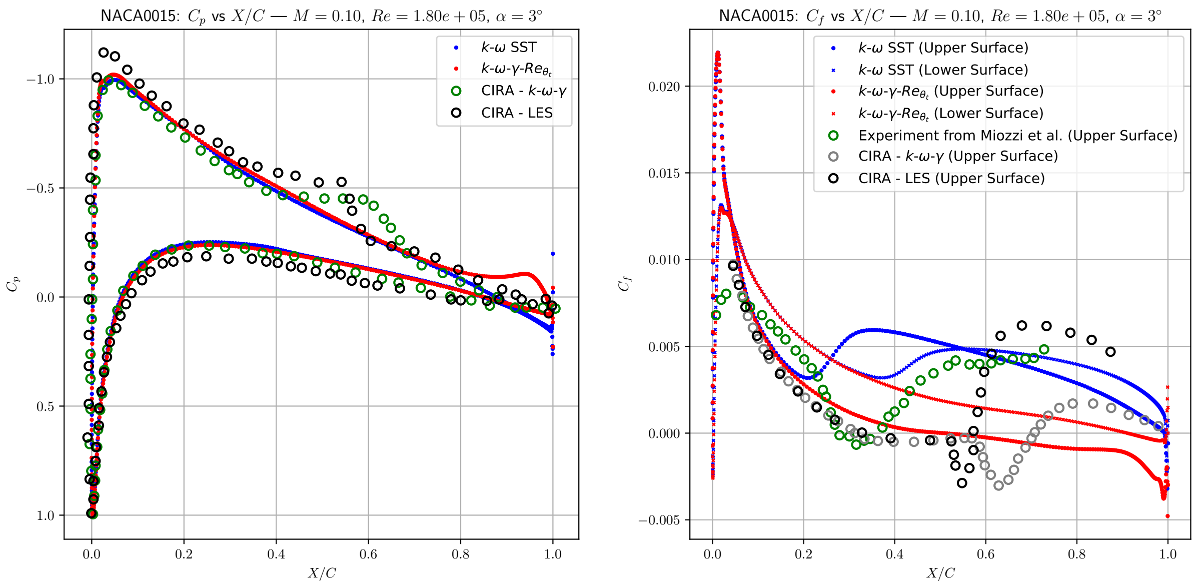

Figure 7.3: Distributions of pressure coefficient (C_p) and skin friction coefficient (C_f) over the NACA 0015 airfoil at Re = 180000, $\alpha = 3°$, comparing the developed k–ω–γ–Re_{θ_t} model with experimental measurements [115] and numerical model predictions [113].

Table 7.3: Comparison of aerodynamic coefficients and flow features for the NACA 0015 airfoil at Re = 180000, $\alpha = 3°$.

Reference	C_l	C_d	ε_{C_l} (%)	ε_{C_d} (%)	$x_{separation}/c$	$x_{transition}/c$	$x_{reattachment}/c$
LES (CIRA [113])	-	-	–	–	0.28	0.54	0.57
k–ω–γ (CIRA [113])	-	-	-	-	0.30	0.63	0.71
Experiment from Miozzi et al. [115]	-	-	-	-	0.27	0.31	0.35
k–ω–γ–Re_{θ_t} (this work)	0.3499	0.0159	–	–	0.48	0.98	1.0
XFOIL, $N_{crit} = 9$ (Airfoil Tools [116])	0.3803	0.0128	8.68	-19.50	-	0.55	-

For the NACA 0015 airfoil at Re = 180000 and a low angle of attack of 3°, the LES results from CIRA provide a reliable reference for transitional flow development. They capture a moderately sized laminar separation bubble, with separation occurring around mid-chord and transition and reattachment taking place shortly thereafter. This behavior reflects the expected flow structure for lightly loaded airfoils at moderate Reynolds numbers.

The k–ω–γ model predicts a slightly delayed separation point relative to LES and substantially later transition and reattachment locations. This results in an elongated laminar separation bubble, suggesting that the model underpredicts instability amplification and delays turbulence onset. Such behavior is characteristic of two-equation models that may lack sufficient sensitivity to shear-layer transition without additional transition correlations.

Experimental measurements from Miozzi et al. [115] indicate a markedly shorter bubble, with transition and reattachment occurring much earlier than in the LES or RANS-based predictions. This discrepancy can be attributed to differences in freestream turbulence levels and experimental boundary conditions, both of which strongly influence transition onset in separated flows.

The k–ω–γ–$\mathrm{Re}_{\theta_\mathrm{t}}$ model developed in this work predicts substantially delayed separation, transition, and reattachment. The entire bubble is displaced downstream, nearly reaching the trailing edge, implying a strongly suppressed instability growth. Although the predicted lift and drag coefficients remain consistent with expectations for weakly separated, low-incidence flow, the extended bubble length suggests a conservative transition trigger, potentially limiting the model's ability to respond to mild adverse pressure gradients. Notably, the pressure coefficient ($\mathrm{C_p}$) distribution shows no overshoot or undershoot, indicating a smooth and gradual transition. This absence of non-monotonic features suggests a less abrupt laminar–turbulent breakdown, although partial convergence effects cannot be ruled out.

The XFOIL prediction with a critical amplification factor of $\mathrm{N_{crit}} = 9$ shows earlier transition and significantly more optimistic aerodynamic performance, with higher lift and lower drag. However, given XFOIL's inviscid outer flow assumption and limited treatment of separation, these predictions tend to overestimate performance in regimes where separation-induced transition plays a dominant role.

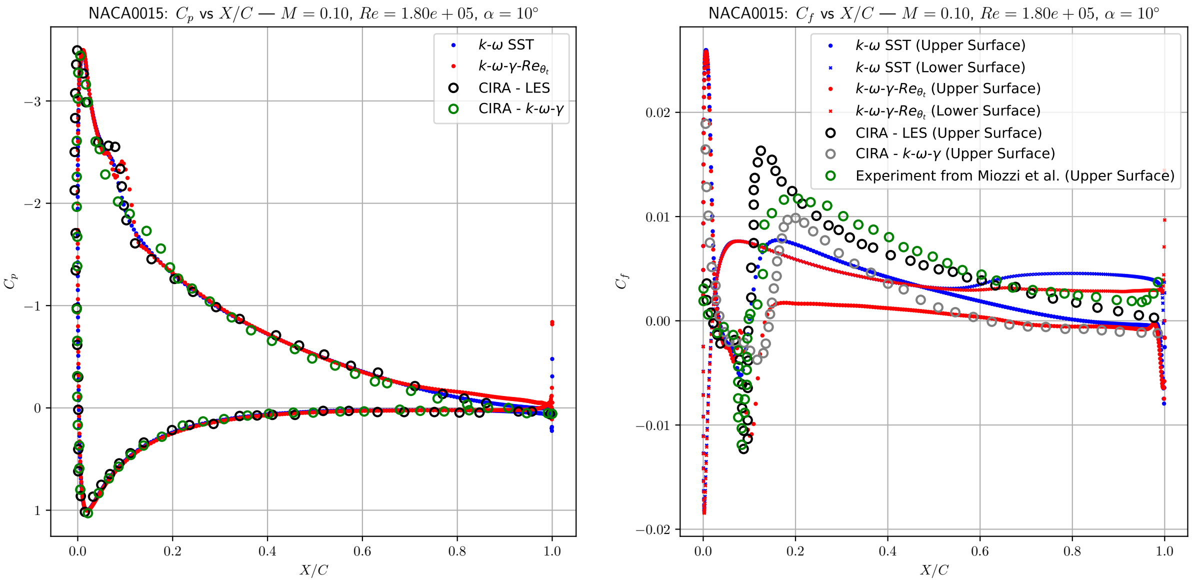

Figure 7.4: Distributions of pressure coefficient (C_p) and skin friction coefficient (C_f) over the NACA 0015 airfoil at Re = 180000, $\alpha = 10°$, comparing the developed k–ω–γ–Re_{θ_t} model with experimental measurements [115] and numerical simulations [113].

Table 7.4: Comparison of aerodynamic coefficients and flow features for the NACA 0015 airfoil at Re = 180000, $\alpha = 10°$.

Reference	C_l	C_d	ε_{C_l} (%)	ε_{C_d} (%)	$x_{separation}/c$	$x_{transition}/c$	$x_{reattachment}/c$
LES (CIRA [113])	–	–	–	–	0.02	0.09	0.10
k–ω–γ (CIRA [113])	–	–	–	–	0.04	0.11	0.14
Experiment (Miozzi et al. [115])	–	–	–	–	0.02	0.09	0.10
k–ω–γ–Re_{θ_t} (this work)	0.9670	0.0307	–	–	0.03	0.10	0.13
XFOIL, $N_{crit} = 9$ [116]	0.9987	0.0245	3.28	-20.19	–	0.09	–

At $\alpha = 10°$, the NACA 0015 airfoil operates in a regime dominated by early separation and rapid transition. Both LES and experimental data from Miozzi et al. indicate a laminar separation very close to the leading edge, followed by swift transition and reattachment, forming a short separation bubble confined to the front portion of the chord. This behavior is typical of moderately loaded airfoils at high angles of attack where strong adverse pressure gradients rapidly destabilize the boundary layer.

The k–ω–γ model from CIRA predicts a slightly delayed separation and transition compared to LES and experiments, leading to a marginally longer laminar separation bubble. The reattachment point is also shifted downstream, although the overall structure remains consistent with reference data. This delay is likely due to the model's more diffused response to instability amplification in steep pressure gradients.

The k–ω–γ–Re_{θ_t} model developed in this work provides predictions in very close agreement with the LES and experimental data. Separation and transition occur just slightly downstream of the reference locations, while the reattachment point is also within a narrow margin of deviation. The predicted lift and drag coefficients fall within a realistic range, reinforcing the consistency of the modeled flow features. Importantly, the pressure coefficient distribution for this case exhibits a minimal spurious overshoot followed by an undershoot near transition, suggesting that the simulation has achieved partial convergence.

XFOIL predictions, based on an $\mathrm{N_{crit}} = 9$ setting, estimate an earlier transition onset that matches the reference quite closely. However, the drag is significantly underpredicted, and no reattachment location is available. As XFOIL lacks a robust treatment of separated turbulent flow, especially in regions of rapid transition and reattachment, its applicability under these conditions is limited. While the lift coefficient is reasonably close to the value predicted by the present model, the optimistic drag prediction reflects the limitations of the boundary-layer formulation used in XFOIL.

Overall, the results confirm that the k–ω–γ–$\mathrm{Re}_{\theta_\mathrm{t}}$ model developed in this work performs well under high-incidence, early-transition conditions. The accurate capture of key transitional locations, combined with smooth pressure recovery and convergence behavior, supports the model's capability to handle strong adverse pressure gradients and rapid laminar-turbulent breakdown near the leading edge.

7.3 Eppler 387

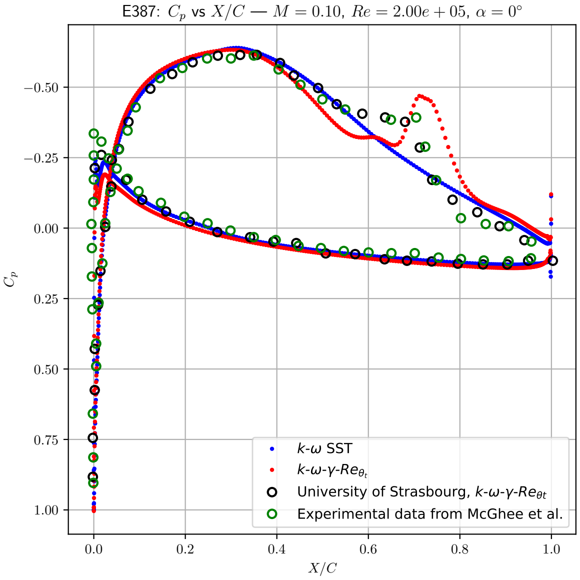

Figure 7.5: Distributions of pressure coefficient ($\mathrm{C_p}$) over the Eppler 387 airfoil at Re = 200000 and $\alpha = 0°$, comparing the present k–ω–γ–$\mathrm{Re}_{\theta_\mathrm{t}}$ model with experimental data [117] and numerical predictions from [113].

Table 7.5: Comparison of aerodynamic coefficients, relative errors, and transition-related flow features for the Eppler 387 airfoil at Re = 200000, $\alpha = 0°$.

Reference	C_l	C_d	ε_{C_l} (%)	ε_{C_d} (%)	$x_{separation}/c$	$x_{transition}/c$	$x_{reattachment}/c$
Experiment (McGhee [117])	0.3603	0.0108	–	–	0.55	0.70	0.75
k–ω–γ–Re_{θ_t} (University of Strasbourg [113])	0.3661	0.0119	1.61	10.19	0.53	0.67	0.76
k–ω–γ–Re_{θ_t} (this work)	0.3786	0.0110	8.17	1.85	0.63	0.65	0.82

For the Eppler 387 airfoil at Re = 200000 and zero angle of attack, the experimental data from McGhee [117] provide a solid benchmark, showing a laminar separation bubble that initiates just beyond mid-chord, with transition and reattachment occurring further downstream. The measured lift and drag coefficients reflect the moderate loading and relatively gentle separation characteristics typical of this configuration.

The k–ω–γ–Re_{θ_t} model from the University of Strasbourg [113] closely reproduces the experimental separation and transition locations, with a slightly earlier onset of transition and a somewhat extended separation bubble. The lift and drag predictions are in good agreement, although drag tends to be overestimated, possibly due to model sensitivity in the separated shear layer.

The k–ω–γ–Re_{θ_t} model developed in this work predicts delayed separation and transition compared to both experiment and Strasbourg results, with reattachment occurring further downstream. The lift coefficient is marginally overpredicted, while the drag is notably closer to experimental values, suggesting improved capture of viscous effects in the separation region.

Notably, an undershoot in the surface pressure coefficient C_p is observed near the separation point when compared to the other model and experimental data. This localized pressure dip may be indicative of sharper gradients and flow deceleration at separation, and could reflect the model's heightened sensitivity to shear-layer dynamics or minor numerical artifacts arising from flow detachment. Despite this, the overall flow topology and aerodynamic performance remain consistent with the expected physical behavior.

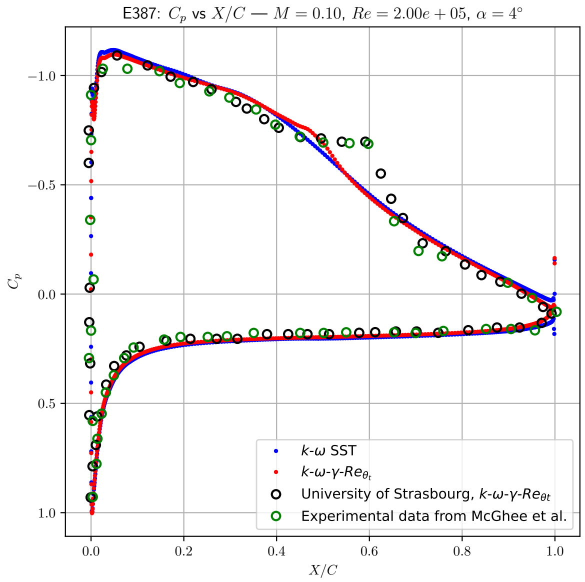

Figure 7.6: Distributions of pressure coefficient (C_p) over the Eppler 387 airfoil at Re = 200000, $\alpha = 4°$, comparing the developed k–ω–γ–Re_{θ_t} model with experimental measurements [117] and numerical model predictions [113].

Table 7.6: Comparison of aerodynamic coefficients, relative errors, and flow features for the Eppler 387 airfoil at Re = 200000, $\alpha = 4°$.

Reference	C_l	C_d	ε_{C_l} (%)	ε_{C_d} (%)	$x_{separation}/c$	$x_{transition}/c$	$x_{reattachment}/c$
Experiment (McGhee [117])	0.7988	0.0139	–	–	0.45	0.61	0.65
k–ω–γ–Re_{θ_t} (University of Strasbourg [113])	0.8060	0.0152	0.90	9.35	0.45	0.60	0.67
k–ω–γ–Re_{θ_t} (this work)	0.8036	0.0155	0.60	11.51	0.40	0.50	0.56

For the Eppler 387 airfoil at Re = 200000 and an angle of attack of 4°, the experimental data from McGhee [117] provide a robust baseline, indicating a laminar separation bubble initiating near mid-chord, followed by transition and reattachment downstream. The lift and drag coefficients measured reflect moderate aerodynamic loading and well-behaved boundary-layer behavior.

The k–ω–γ–Re_{θ_t} model from the University of Strasbourg [113] closely matches the experimental separation and transition locations, predicting a slightly extended separation bubble with minor overprediction of drag. The aerodynamic coefficients remain within reasonable agreement with the experiment, confirming the model's capability in capturing transitional flow physics.

The present k–ω–γ–Re_{θ_t} model predicts separation and transition locations somewhat upstream of the reference values, resulting in a shorter separation bubble. Lift and drag coefficients remain consistent with expectations, indicating the model's robustness in this flow regime. Importantly, the pressure coefficient distribution exhibits no overshoot or undershoot near transition or reattachment, indicating a smooth and stable laminar-to-turbulent breakdown, despite incomplete convergence.

Chapter 8

Results Part 3: Optimization

Optimization involving transition models remains particularly challenging [118, 32, 80]. Since forward- and reverse-mode derivatives have not yet been fully verified in the developed transition model, only a limited demonstration of airfoil optimization is presented using a fully turbulent model. A comprehensive multipoint, multi-objective optimization will follow. As a preliminary step, single-point and multipoint optimizations with drag coefficient C_d as the objective were performed to gain familiarity with the optimization framework.

8.1 Optimization Process Description

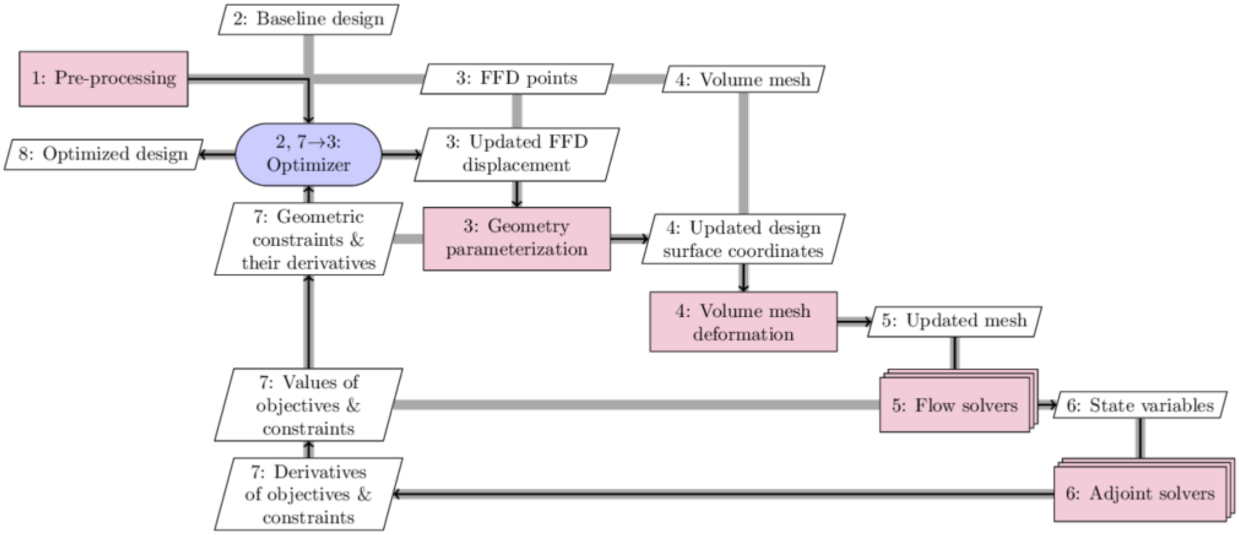
2: Baseline design
1: Pre-processing
3: FFD points
4: Volume mesh
8: Optimized design
2, 7→3: Optimizer
3: Updated FFD displacement
7: Geometric constraints & their derivatives
3: Geometry parameterization
4: Updated design surface coordinates
4: Volume mesh deformation
5: Updated mesh
7: Values of objectives & constraints
5: Flow solvers
6: State variables
7: Derivatives of objectives & constraints
6: Adjoint solvers

Figure 8.1: Workflow of the MACH-Aero framework for aerodynamic shape optimization [119].

The MACH-Aero [119] framework enables high-fidelity aerodynamic shape optimization by integrating modular tools into an automated and scalable design loop. The end-to-end process is outlined below:

1. **Mesh Generation (Pre-processing):** A structured multiblock volume mesh is generated

around the baseline geometry using the tool pyHyp [93]. This mesh is later used by the CFD solver (Step 5). Simultaneously, Free-Form Deformation (FFD) [120] control points are defined to enable shape parameterization in Step 3.

2. Design Variable Initialization: Design variables are initialized and passed to a gradient-based optimizer via the pyOptSparse [121] interface. Typical choices include SNOPT [122] and SLSQP [123], which are based on Sequential Quadratic Programming (SQP). The optimizer returns updated design variables to the geometry module.

3. Geometry Parameterization (pyGeo): pyGeo [124] performs geometry deformation using the current design variables and predefined FFD control points. It also evaluates geometric constraints and their derivatives. The deformed surface is then passed to the mesh deformation module (IDWarp, Step 4).

4. Mesh Deformation (IDWarp): IDWarp [93] updates the volume mesh to conform to the deformed geometry, preserving mesh quality. The resulting mesh is then used in the CFD analysis.

5. Flow Solution (ADflow The updated geometry and mesh are used to compute the aerodynamic flow field using a high-fidelity CFD solver. For compressible and transonic flows, ADflow [75] is commonly used, evaluating aerodynamic objectives (e.g., drag) and constraints (e.g., lift, moment).

6. Adjoint Derivative Computation: The adjoint solvers in ADflow [125] efficiently compute gradients of the objectives and constraints with respect to all design variables. This method is highly scalable, as the computational cost is largely independent of the number of design variables.

7. Optimization Update: The optimizer uses the function evaluations and gradients to update the design variables. This loop iterates until convergence is achieved, resulting in an optimized configuration.

8.2 Solvers and Discretization Details

The flow solver parameters were carefully tuned for accuracy and robustness. The additive Schwarz method was used as a global preconditioner, with incomplete LU (ILU) factorization applied locally at a fill level of two. The CFL number was capped at 100000.0 to ensure numerical stability during iterations. Nonlinear convergence was controlled using a maximum of 40 nonlinear iterations and a relative L2 norm convergence criterion of 1×10^{-16}. The linear systems arising within each nonlinear step were solved to a tolerance of 0.05.

For adjoint-based sensitivity analysis and optimization, the discrete adjoint solver is configured to use the GMRES (Generalized Minimal Residual) algorithm [126]. A tight convergence threshold of 1×10^{-12} ensures that the adjoint residuals are minimized effectively. The automatic differentiation (AD) preconditioner is enabled for the adjoint solver to enhance precision and stability, particularly in the presence of turbulent flow features. The adjoint solver is allowed to run up to 5000 iterations per design cycle.

The Spalart–Allmaras (SA) turbulence model is used for all RANS simulations. The total number of allowed solution cycles is set to 20000, providing sufficient room for full convergence even in complex flow regimes. The SLSQP (Sequential Least Squares Programming) algorithm [123] is employed as the optimizer in the design loop.

8.3 Grid Convergence Analysis for the NASA SC(2)-0714 Airfoil

The geometry selected is the NASA SC(2)-0714 a supercritical airfoil designed for transonic flight regimes, optimized to delay shock wave formation and minimize wave drag. Featuring a 14% thickness-to-chord ratio and carefully tailored camber, it achieves efficient aerodynamic performance near Mach 0.7–0.8, making it well-suited for high-speed subsonic and transonic applications [127].

A comprehensive grid convergence study was conducted to ensure numerical robustness and to minimize discretization errors. The study considered two representative flight conditions:

- FC0: Mach M = 0.70, altitude h = 36000 ft, angle of attack $\alpha = 1^\circ$
- FC1: Mach M = 0.75, altitude h = 40000 ft, angle of attack $\alpha = 4^\circ$

For each condition, three successively refined meshes containing 3978, 16195, and 31240 cells respectively were used. The lift (C_l) and drag (C_d) coefficients were tracked to evaluate sensitivity to mesh resolution.

Richardson extrapolation was employed to estimate the asymptotic drag coefficient, and the Grid Convergence Index (GCI) quantified the discretization uncertainty. Results, summarized in Table 8.1, indicate slightly reduced convergence orders relative to the ideal second order. This deviation is attributed to the airfoil's geometric complexity, mesh stretching strategies, and the influence of turbulence modeling in regions near shock waves.

Case	Refinement Levels	Achieved Order p	Richardson Extrapolation	GCI L0–L1 (%)	GCI L1–L2 (%)
FC0	[1.39, 1.0, 0.5]	1.6454	$C_d = 0.02579$	0.3429	1.7250
		1.8634	$C_l = 0.55041$	0.0280	0.1892
FC1	[1.39, 1.0, 0.5]	1.5049	$C_d = 0.07401$	0.0680	0.3167
		0.5511	$C_l = 0.58679$	1.2586	2.1838

Table 8.1: Grid convergence and GCI analysis for the NASA SC(2)-0714 airfoil under two flight conditions with three mesh refinement levels.

Since the achieved order of convergence of C_l for FC1 was very low, the finest mesh level was selected for all subsequent studies, offering a sound compromise between solution accuracy and computational cost. All simulations were performed in steady-state conditions.

8.4 Optimization Problem Definition

This section and relative subsections present a reformulation of the airfoil optimization problem following the framework outlined in the `MACH-Aero` tutorial [128]. The formulation specifies the objective, design variables, constraints, and governing parameters essential for aerodynamic shape optimization, providing a structured approach to improve airfoil performance systematically.

8.4.1 Design Variables

The aerodynamic optimization framework in this study employs Free-Form Deformation (FFD) [120] for shape parameterization. This approach allows smooth and flexible adjustments of the airfoil geometry by embedding it within a lattice of control points. During optimization, selected control points move along the surface-normal direction, serving as the primary design variables that enable shape modifications while maintaining geometric continuity and smoothness.

Besides geometric parameters, the angle of attack α is also treated as a design variable. Varying α within set bounds allows the optimizer to modify flow incidence, ensuring aerodynamic objectives are met. This joint manipulation of shape and angle guarantees that the airfoil sustains favorable aerodynamic performance and required lift throughout the optimization.

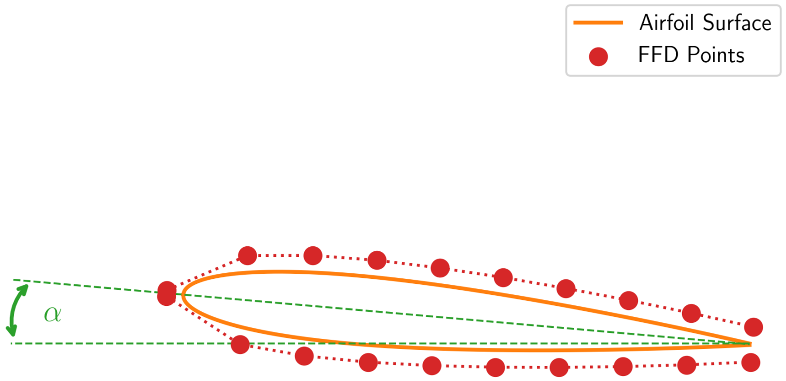

Figure 8.2: Design variables used in the optimization: FFD control points displacements and angle of attack α.

8.4.2 Constraints

To ensure that the resulting optimized shapes are physically meaningful and aerodynamically viable, a set of constraints is imposed on the optimization problem. Geometric constraints are applied to maintain the structural feasibility of the airfoil. A volume constraint ensures that the airfoil does not become unrealistically thin, while a minimum local thickness is enforced to preserve structural integrity and manufacturability. The geometry near the leading and trailing edges is held fixed by constraining the motion of points in these regions. Finally, symmetry about the mid-span is enforced through a set of linear constraints, requiring mirrored control point displacements across the spanwise direction. This maintains aerodynamic balance and simplifies the manufacturing process for symmetric configurations.

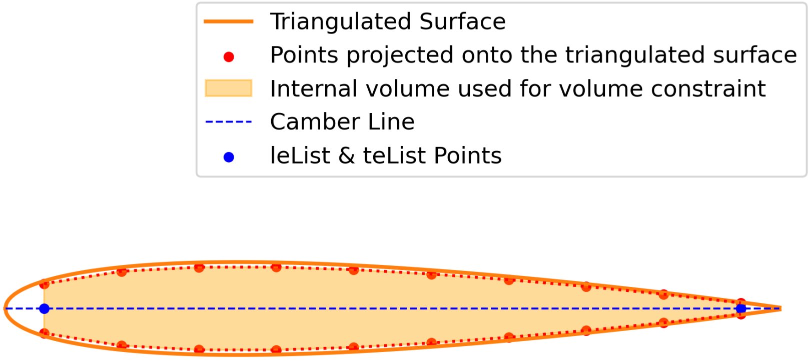
Triangulated Surface
Points projected onto the triangulated surface
Internal volume used for volume constraint
Camber Line
leList & teList Points

Figure 8.3: Constraints: selected points of each type are shown for clarity [128].

These constraints are implemented using a projection-based method, where a regularly spaced 2D grid of constraint points is projected onto the airfoil surface using surface normals derived from a ruled surface defined by the bounding corner points (from `leList` and `teList`). In non-planar regions, particularly near open edges, this projection may encounter difficulties due to geometric misalignments, potentially leading to failed intersection or constraint enforcement errors.

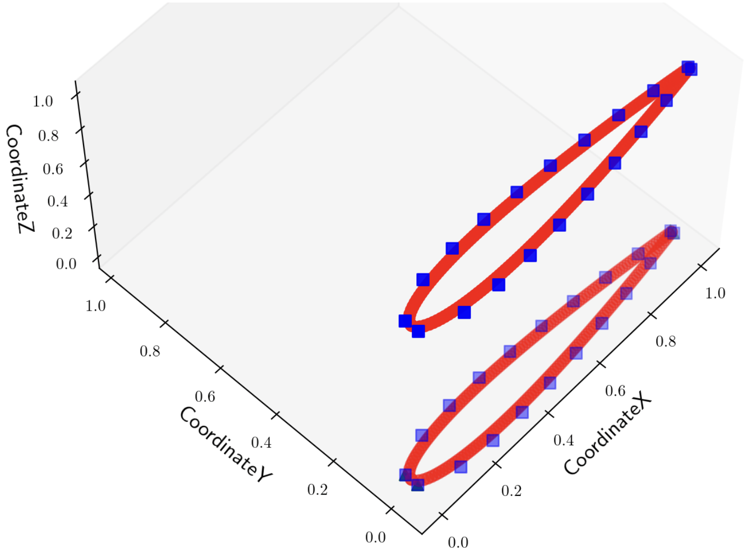

Figure 8.4: Geometric constraints in 3D: blue points enforce spanwise symmetry; red points enforce volume and thickness constraints

8.4.3 Objective Function

The objective of the optimization is to minimize aerodynamic drag while maximizing the lift. The scalar objective function is formulated as:

$$\min_{\mathbf{x}} (\mathrm{C_d} - \mathrm{C_l})(\mathbf{x})$$

Here, $\mathbf{x}$ denotes the full set of design variables, including the FFD-based shape parameters and the angle of attack. Both drag ($\mathrm{C_d}$) and lift ($\mathrm{C_l}$) coefficients are computed using a Reynolds-Averaged Navier–Stokes (RANS) solver coupled with the Spalart–Allmaras turbulence model. The objective function promotes drag reduction while maximizing lift, and its gradient is evaluated efficiently using an adjoint method, allowing the optimization to scale to high-dimensional design spaces introduced by FFD.

8.5 Optimization Problem Formulation

The aerodynamic shape optimization problem aims to identify a design vector

$$\mathbf{x} = \left[\alpha, \delta_{\mathrm{FFD},1}, \delta_{\mathrm{FFD},2}, \ldots\right],$$

where α is the angle of attack and $\delta_{\mathrm{FFD,i}}$ are the free-form deformation (FFD) control point displacements. The goal is to minimize the drag coefficient $\mathrm{C_d}(\mathbf{x})$ while maximizing the lift coefficient $\mathrm{C_l}(\mathbf{x})$, subject to relevant physical and geometric constraints.

Formally, the problem is expressed as:

Algorithm 1 Aerodynamic Optimization Problem

```
Given: Initial geometry, reference volume $\mathrm{V_0}$, and design variable bounds.
Find: Design vector $\mathbf{x}$.
Objectives: Minimize $\mathrm{C_d}(\mathbf{x})$, maximize $\mathrm{C_l}(\mathbf{x})$.
Subject to:
    $\alpha \in [0.0, 10.0]\ ^\circ,$
    $\delta_{\mathrm{FFD,i}} \in [-0.05, 0.05]\ \mathrm{m}, \quad \forall \mathrm{i},$
    $\mathrm{V}(\mathbf{x}) \geq \mathrm{V_0},$
    $\mathrm{t_{min}}(\mathbf{x}) \in [0.1, 3.0]\ \mathrm{t_0},$
    $\mathrm{L_{span}}(\mathbf{x}) = 0\ \mathrm{m}$
```

The constraints ensure the search remains within physically meaningful and manufacturable limits. Table 8.2 summarizes the variables and their bounds.

Symbol	Description	Quantity	Lower Bound	Upper Bound	Units
α	Angle of attack	1	0.0	10.0	°
δ_{FFD}	FFD control point displacements	40	-0.05	0.05	m
V_{enc}	Volume constraint	1	1.0	—	—
t_{min}	Minimum thickness	200	0.1 t_0	3.0 t_0	—
LE/TE	Leading/trailing edge constraints	2	—	—	—
L_{span}	Spanwise linear deformation	20	0.0	0.0	m

Table 8.2: Design Variables and Constraints

8.6 Multi-Objective and Multi-Point Optimization Results for the NASA SC(2)-0714 Airfoil

Figures 8.5 and 8.6 display the airfoil geometry and performance data before and after optimization, respectively. The results demonstrate the optimizer's ability to enhance aerodynamic efficiency by trading off drag and lift across multiple operating conditions.

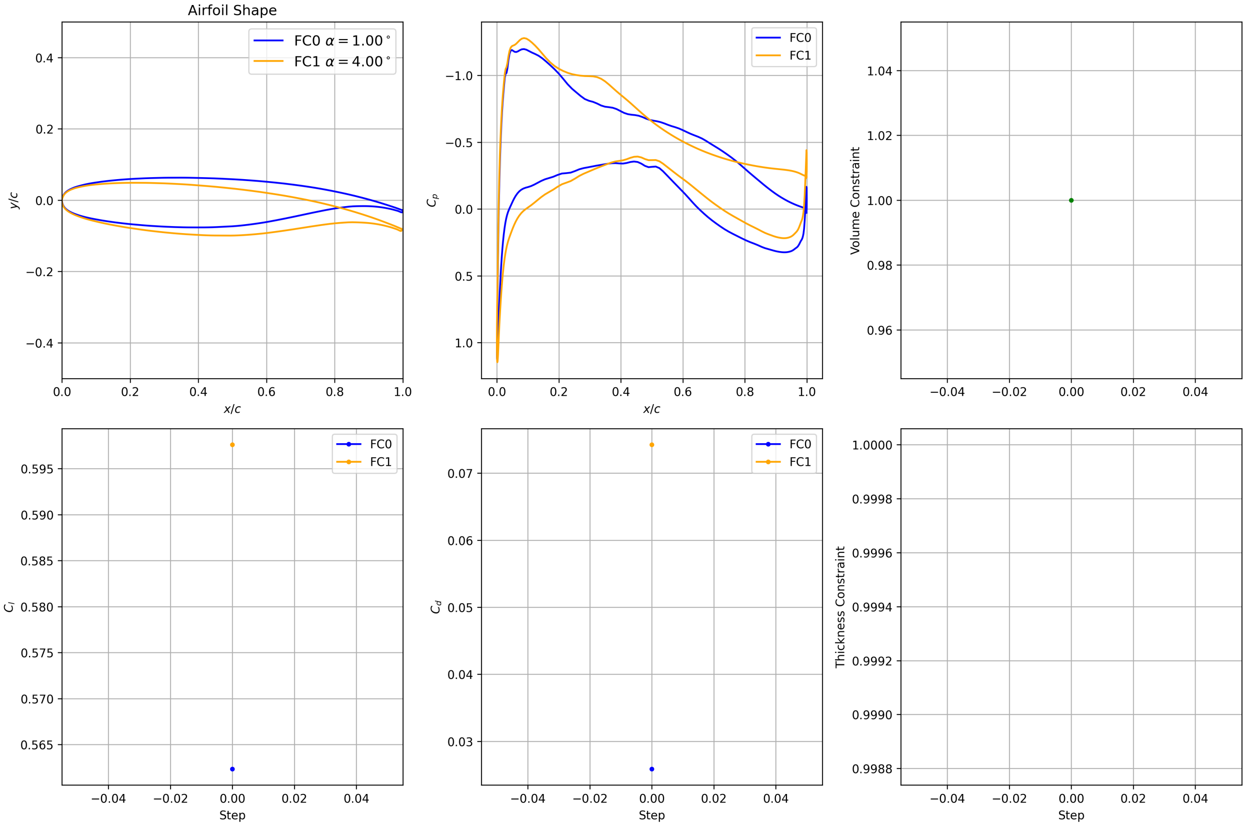

Figure 8.5: Initial airfoil geometry and performance data. The *Step* refers to the optimization iteration.

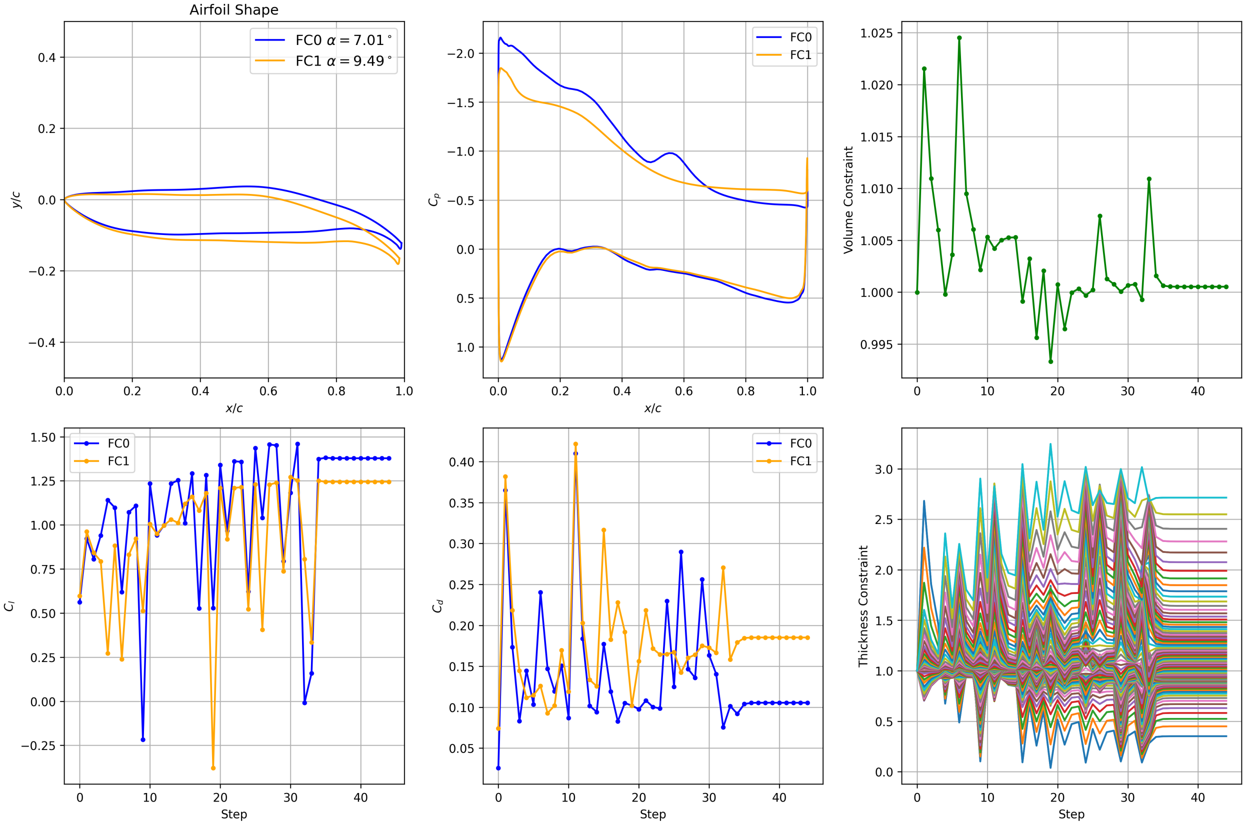

Figure 8.6: Optimized airfoil geometry and performance data. The *Step* refers to the optimization iteration.

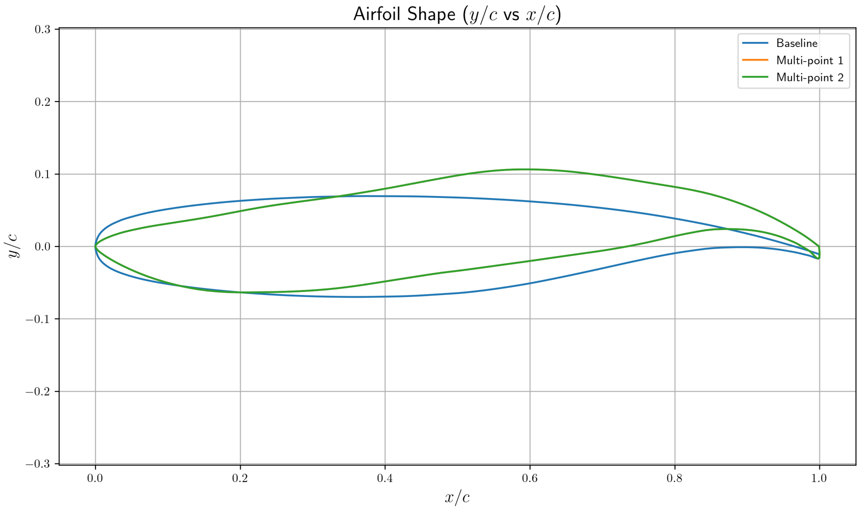

Figure 8.7: Comparison between the initial and optimized airfoil geometries.

Table 8.3 summarizes the initial and final values of key aerodynamic metrics and computational effort.

Table 8.3: Key optimization metrics at initial and final design points for the NASA SC(2)-0714 airfoil.

Quantity	**Initial**	**Final**
α_{FC0}	1.00 °	7.01 °
α_{FC1}	4.00 °	9.49 °
$\mathrm{C}_{\mathrm{d,FC0}}$	0.0259	0.1055
$\mathrm{C}_{\mathrm{d,FC1}}$	0.0742	0.1852
$\mathrm{C}_{\mathrm{l,FC0}}$	0.5624	1.3781
$\mathrm{C}_{\mathrm{l,FC1}}$	0.5976	1.2451
Objective value	-0.5299	-1.1663
Elapsed time	30.49 s	2128.72 s

Throughout the optimization process, the angle of attack was adjusted within the prescribed bounds, leading to more favorable lift and drag characteristics. Notably, the optimized design exhibits increased lift coefficients at multiple flight conditions, which indicates enhanced lift generation capabilities.

Conversely, while the drag coefficients at these conditions show a moderate increase, this trade-off is typical in multi-objective aerodynamic optimizations where improvements in lift often come at the expense of some drag penalty. However, the overall objective function, which balances these competing criteria, shows a marked improvement, validating the effectiveness of the chosen optimization approach.

The elapsed computational time reflects the complexity of the problem, including the multi-point evaluation and the high dimensionality of the design space. The results confirm that the optimization framework efficiently navigated the feasible design space to identify superior airfoil shapes that satisfy all constraints.

Visual inspection of the airfoil shapes before and after optimization reveals noticeable geometric modifications, which align with the observed aerodynamic performance changes. These modifications are consistent with the deformation allowed by the Free-Form Deformation (FFD) control points, illustrating the design flexibility embedded in the optimization process.

Overall, the results underscore the capability of the employed multi-objective and multi-point optimization methodology to achieve improved aerodynamic efficiency for the NASA SC(2)-0714 airfoil within the specified design constraints.

Chapter 9

Conclusions

The primary objective of this work was the implementation and integration of a transitional turbulence model—specifically, the Langtry-Menter k–ω–γ–$\mathrm{Re}_{\theta_\mathrm{t}}$ transition model—within the ADflow framework. This implementation was achieved, allowing for the simulation of laminar-to-turbulent transition phenomena in a high-fidelity environment.

However, a complete verification of the gradients produced via forward-mode automatic differentiation (AD) has not yet been accomplished. The current forward-mode AD implementation still requires careful verification, and the observed inconsistencies in gradient-based quantities reflect this limitation. These challenges have also impacted the model validation process, where deviations were observed when comparing simulation results with experimental data, LES reference solutions, and other transition modeling approaches.

Nevertheless, as a standalone example, the framework was used to conduct aerodynamic shape optimization of a supercritical airfoil with the Spalart-Allmaras turbulence model. Although the optimization primarily served as a demonstration of ADflow's capability to integrate and solve gradient-based design problems, the results indicate the solver's ability to converge to improved shapes even under constraints.

Aerodynamic shape optimization continues to be a subject of significant interest, particularly in aerospace and marine sectors. A notable recent example is the February 2025 study by Ng et al. [129], which applied high-fidelity optimization using ADflow and the Spalart-Allmaras model for the design of America's Cup AC75 hydrofoil sections with flaps. This work highlights the growing application of gradient-based optimization in real-world, performance-critical contexts such as competitive sailing and aerospace engineering.

9.1 Future Work

While the optimization of the airfoil shape using steady-state simulations was successful, there are several avenues for further improvement:

9.1.1 Integration of Transitional Modeling in Optimization

Following the verification of the flow solver's gradients using forward-mode automatic differentiation (AD), the next critical step is to verify the derivatives obtained via reverse-mode AD. This is essential for enabling consistent and efficient adjoint-based optimization with the transitional turbulence model. Achieving this integration requires that all components of the transition model be differentiable, and that derivative computations—both in forward and reverse modes—remain accurate, stable, and consistent.

To ensure the reliability of this integration, comprehensive validation strategies must be employed. These should include comparisons against experimental measurements, high-fidelity numerical data such as large-eddy simulation (LES) results, and cross-validation with established transition modeling approaches. Such validation is vital to build confidence in both the physical accuracy of the model and the correctness of the computed sensitivities.

A robust, fully verified and validated transitional modeling capability, integrated with gradient-based optimization, would significantly expand the applicability of the framework. It would enable high-fidelity design optimization in a wide range of engineering scenarios, particularly those involving low-Reynolds-number flows, transitional behavior, and complex geometry constraints.

9.1.2 Unsteady Optimization

The current optimization approach focuses on steady-state flow conditions. However, real-world aerodynamic performance often involves unsteady effects, especially at higher speeds or during transient maneuvers. Future work could explore unsteady optimization to account for time-varying aerodynamic forces and to improve performance under more dynamic conditions.

9.1.3 Use of Advanced Turbulence Models for Optimization

The Spalart-Allmaras (SA) turbulence model used in this aerodynamic optimization provides a good approximation for many aerodynamic applications. However, for higher accuracy, particularly in complex flow regimes, advanced turbulence models such as Large Eddy Simulation (LES) could be employed.

Appendix A

Field Functions from the Simulations

This chapter presents isocontours of nondimensional turbulent kinetic energy (k/U_∞^2) and nondimensional velocity magnitude (U/U_∞) obtained from simulations of various airfoils under different flow conditions. These fields provide valuable insight into laminar–turbulent transition, flow separation, and reattachment.

A.1 SD7003 Airfoil

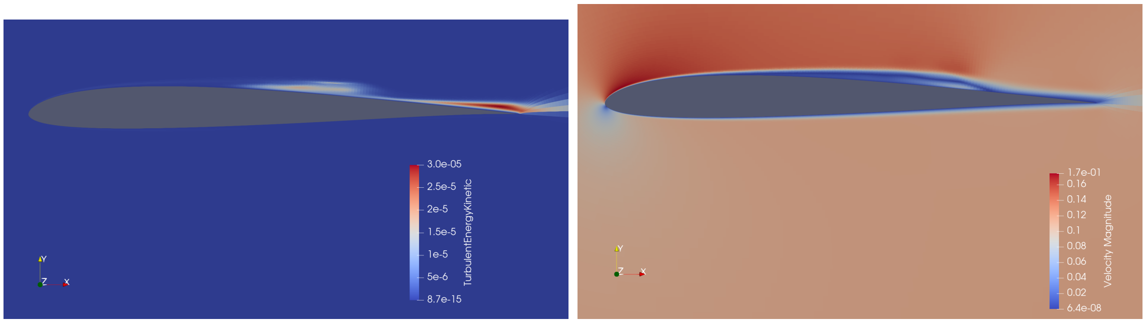

Figure A.1: SD7003 — $M = 0.1$, $Re = 6 \times 10^4$, $\alpha = 4°$, fine grid (`SD7003_L0`): k/U_∞^2 (left), U/U_∞ (right).

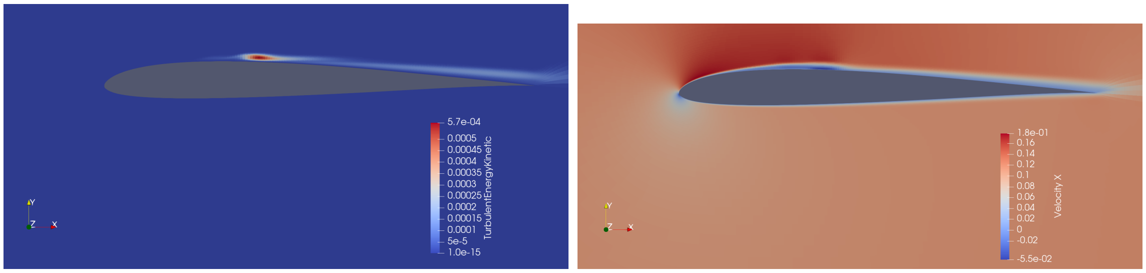

Figure A.2: SD7003 — $M = 0.1$, $Re = 6 \times 10^4$, $\alpha = 6°$: k/U_∞^2 (left), U/U_∞ (right).

A.2 NACA 0015 Airfoil

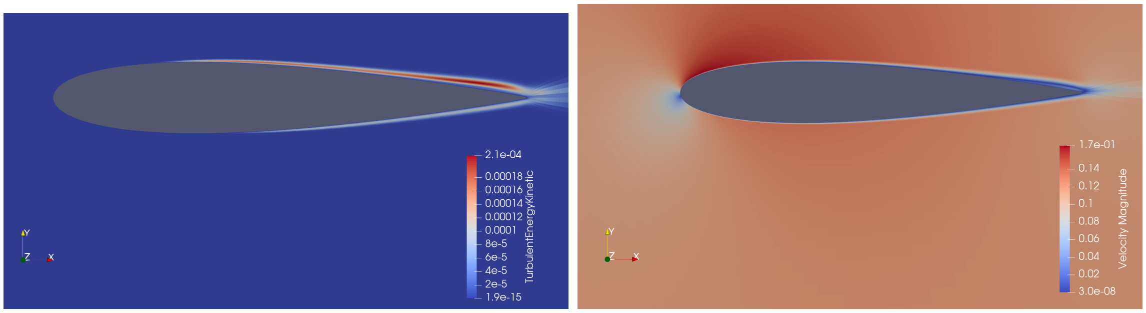

Figure A.3: NACA0015 — M = 0.1, Re = 1.8 × 10^5, $\alpha = 3°$: k/U_∞^2 (left), U/U_∞ (right).

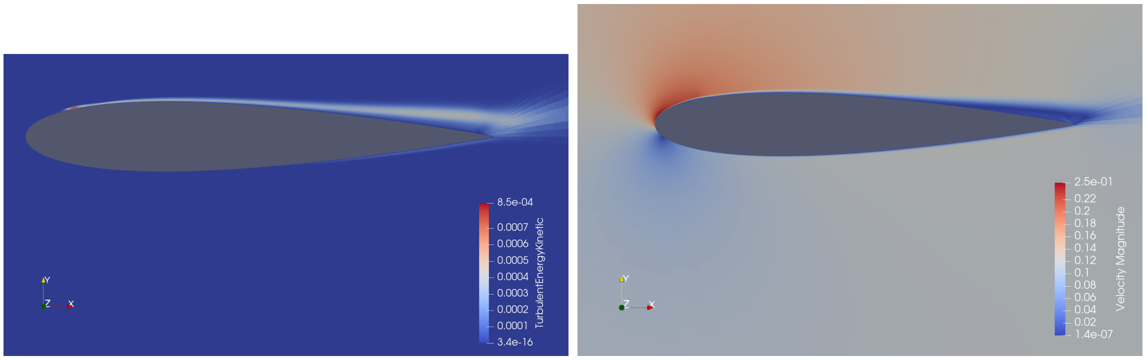

Figure A.4: NACA0015 — M = 0.1, Re = 1.8 × 10^5, $\alpha = 10°$: k/U_∞^2 (left), U/U_∞ (right).

A.3 Eppler 387 Airfoil

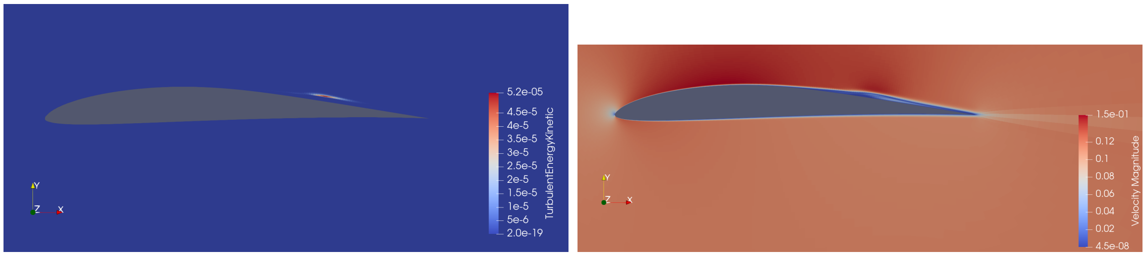

Figure A.5: Eppler 387 — M = 0.1, Re = 2 × 10^5, $\alpha = 0°$: k/U_∞^2 (left), U/U_∞ (right).

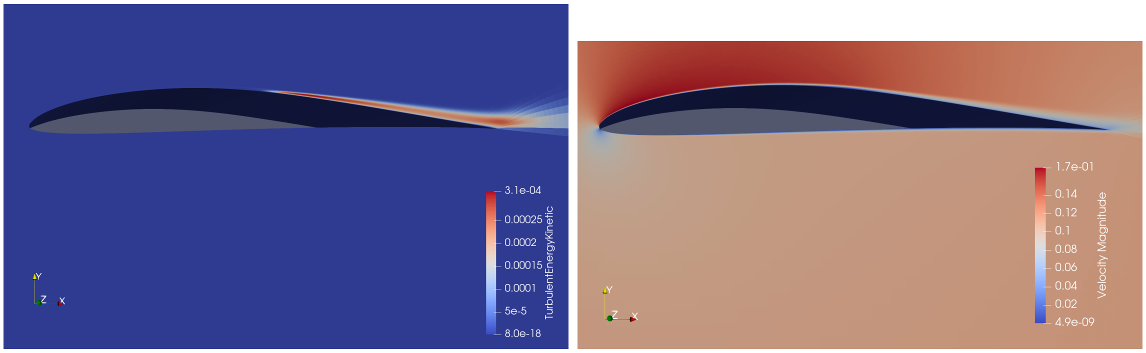

Figure A.6: Eppler 387 — $\mathrm{M} = 0.1$, $\mathrm{Re} = 2 \times 10^5$, $\alpha = 4^\circ$: $\mathrm{k}/\mathrm{U}_\infty^2$ (left), $\mathrm{U}/\mathrm{U}_\infty$ (right).

Appendix B

Minor Code Modification Documentation

This document summarizes the minor modifications for the Langtry-Menter SST transitional turbulence model in ADflow, detailing the affected modules and key subroutines with code excerpts. The major model implementation is documented discursively in Section 6.2.

B.1 Subroutine `turbSolveDDADI` in module `TurbAPI`

Handling for `langtryMenterSST` added:

Listing B.1: Excerpt from `turbSolveDDADI`

```fortran
select case (turbModel)
...
  case (komegaWilcox, komegaModified, menterSST, langtryMenterSST, ktau)
    call unsteadyTurbSpectral(itu1, itu2)  ! Compute unsteady turbulence spectral terms

  case (menterSST, langtryMenterSST)
    call SST_block_residuals(.false.)      ! Disable SST residual blocks for initial solve
    call SSTSolve                          ! Call SST solver for turbulence variables
end select
```

B.2 Subroutine `turbResidual` in module `TurbAPI`

Included `langtryMenterSST` case:

Listing B.2: Excerpt from `turbResidual`

```fortran
select case (turbModel)
...
  case (menterSST, langtryMenterSST)
```

```
    call SST_block_residuals(.True.)  ! Enable SST residual blocks to compute turbulence residuals
end select
```

B.3 Subroutine **`computeEddyViscosity`** in module **`turbUtils`**

Added `langtryMenterSST` to call SST eddy viscosity calculation:

Listing B.3: Excerpt from `computeEddyViscosity`

```
select case (turbModel)
...
  case (menterSST, langtryMenterSST)
    call SSTEddyViscosity(iBeg, iEnd, jBeg, jEnd, kBeg, kEnd) ! Compute eddy viscosity for SST
  ! other cases...
end select
```

B.4 Subroutine **`setBCVarNamesTurb`** in module **`BCData`**

Added BC variable names for transition variables:

Listing B.4: Excerpt from `setBCVarNamesTurb`

```
select case (turbModel)
...
  case (langtryMenterSST)
    bcVarNames(offset + 1) = cgnsTurbK               ! Turbulent kinetic energy
    bcVarNames(offset + 2) = cgnsTurbOmega           ! Specific dissipation rate
    bcVarNames(offset + 3) = cgnsTransitionGamma     ! Transition onset parameter
    bcVarNames(offset + 4) = cgnsTransitionReThetat ! Transition Reynolds number parameter
end select
```

B.5 Logical Function **`setBCVarTurb`** in module **`BCData`**

Freestream initialization for transition variables:

Listing B.5: Excerpt from `setBCVarTurb`

```
select case (turbModel)
...
  case (langtryMenterSST)
    ref(itu1) = pRef / rhoRef       ! Initialize turbulent kinetic energy reference
    ref(itu2) = ref(itu1) / nuRef   ! Initialize omega reference from k and viscosity
    ref(iTransition1) = 0
    ref(iTransition2) = 0
end select
```

B.6 Main residual loop for RANS equations in module **masterRoutines**

Listing B.6: Excerpt from main residual loop

```
if (equations == RANSEquations) then
...
  select case (turbModel)
    case (menterSST, langtryMenterSST)
      call SST_block_residuals(.True.)  ! Include SST residuals for turbulence model solve
  end select
end if
```

B.7 Subroutine **volSolNames** in module **outputMod**

Added volume solution variable names for Langtry-Menter SST model:

Listing B.7: Excerpt from `volSolNames`

```
case (langtryMenterSST)
...
  solNames(itu1) = cgnsTurbK                  ! Name for turbulent kinetic energy solution variable
  solNames(itu2) = cgnsTurbOmega              ! Name for omega solution variable
  solNames(iTransition1) = cgnsTransitionGamma    ! Name for transition gamma solution variable
  solNames(iTransition2) = cgnsTransitionReThetat ! Name for transition ReTheta solution variable

...

if (volWriteResTurb) then

            select case (turbModel)
            ...
            case (langtryMenterSST)
                nn = nn + 1
                solNames(nn) = cgnsResK

                nn = nn + 1
                solNames(nn) = cgnsResOmega

                nn = nn + 1
                solNames(nn) = cgnsTransitionGamma

                nn = nn + 1
                solNames(nn) = cgnsTransitionReThetat
```

B.8 Subroutine **`isoSurfNames`** in module **`outputMod`**

Added iso-surface solution variable names for Langtry-Menter SST model:

Listing B.8: Excerpt from `isoSurfNames`

```
...
if (isoWriteTurb) then

            select case (turbModel)
            ...
            case (langtryMenterSST)
                nn = nn + 1
                solNames(nn) = cgnsResK

                nn = nn + 1
                solNames(nn) = cgnsResOmega

                nn = nn + 1
                solNames(nn) = cgnsTransitionGamma

                nn = nn + 1
                solNames(nn) = cgnsTransitionReThetat

 ...

 if (isoWriteResTurb) then

            select case (turbModel)
            ...
            case (langtryMenterSST)
                nn = nn + 1
                solNames(nn) = cgnsResK

                nn = nn + 1
                solNames(nn) = cgnsResOmega

                nn = nn + 1
                solNames(nn) = cgnsTransitionGamma

                nn = nn + 1
                solNames(nn) = cgnsTransitionReThetat
```

B.9 Subroutine **writeCGNSHeader** in module **outputMod**

Added support for Langtry-Menter SST in CGNS metadata output:

Listing B.9: Excerpt from `writeCGNSHeader`

```
turbulentTest: if (equations == RANSEquations) then
  ...
  case (menterSST, langtryMenterSST)
    call writeCGNSMenterSSTInfo(cgnsInd, base)
```

B.10 Subroutine **blockResCore** in module **blockette**

Enabled residual block computation for Langtry-Menter SST in RANS mode:

Listing B.10: Excerpt from `blockResCore`

```
if (equations == RANSEquations .and. turbRes) then
  ...
  case (menterSST, langtryMenterSST)
    call SST_block_residuals(.True.)
```

B.11 Subroutine **referenceState** in module **initializeFlow**

Freestream turbulent variables initialization:

Listing B.11: Excerpt from `referenceState`

```
select case (turbModel)
....
  case (komegaWilcox, komegaModified, menterSST, langtryMenterSST)
    wInf(itu1) = 1.5_realType * uInf2 * turbIntensityInf**2  ! Calculate freestream turbulent kinetic energy
    wInf(itu2) = wInf(itu1) / (eddyVisInfRatio * nuInf)      ! Calculate freestream omega from k
end select
```

B.12 Subroutine **checkMonitor** in module **inputParamRoutines**

Added monitors for new turbulence variables:

Listing B.12: Excerpt from `checkMonitor`

```
...
case (langtryMenterSST)
  nMon = nMon + 4; nMonSum = nMonSum + 4
  monNames(nMon - 3) = cgnsL2ResK         ! Monitor residual of k
  monNames(nMon - 2) = cgnsL2ResOmega     ! Monitor residual of omega
  monNames(nMon - 1) = cgnsL2ResGamma     ! Monitor residual of transition gamma
  monNames(nMon)     = cgnsL2ResRethetat  ! Monitor residual of transition ReTheta
```

B.13 Subroutine **`setEquationParameters`** in module **`inputParamRoutines`**

Listing B.13: Excerpt from `setEquationParameters`

```
...
case (langtryMenterSST)
  nw = 9                     ! Number of variables including transition scalars
  nt2 = 9                    ! Total turbulence variable count
  iTransition1 = 8           ! Index for transition gamma variable
  iTransition2 = 9           ! Index for transition ReTheta variable
  kPresent = .true.          ! Turbulence kinetic energy present
  eddyModel = .true.         ! Eddy viscosity model active
  transitionModel = GammaRetheta  ! Use Gamma-ReTheta transition model
```

B.14 Subroutine **`readTurbvar`** in module **`variableReading`**

Added `langtryMenterSST` to support reading turbulence variables using `readTurbKwType`:

Listing B.14: Excerpt from `readTurbvar`

```
...
case (menterSST, langtryMenterSST)
  call readTurbKwType(nTypeMismatch)
```

B.15 **`transitionModel`** in module **`inputPhysics`**

Added `transitionModel` to support transition model:

Listing B.15: Excerpt from `constants`

```
       ! transitionModel      Which transition Model to use
       ....
       integer(kind=intType) :: turbModel, cpModel, turbProd, transitionModel
```

B.16 gammaretheta in module constants

Added gammaretheta as a transition model:

Listing B.16: Excerpt from constants

```
    integer(kind=intType), parameter :: &
        noTransitionModel = 0, &
        gammaretheta = 1
```

B.17 C Interface (libadflowmodule.c)

Listing B.17: Excerpt from libadflowmodule.c

```
 ...

 {"langtrymentersst",0,{{-1}},NPY_INT},
...
static void f2py_setup_constants(char *maxstringlen,..... *langtrymentersst,char *v2f
    ,char .....
...
 f2py_constants_def[i_f2py++].data = langtrymentersst;
...
"Fortran 90/95 modules:\n""  constants --- maxstringlen,maxcgnsnamelen,..,
    langtrymentersst,.....
```

B.18 Python Interface (pyADflow.py)

Listing B.18: Excerpt from pyADflow.py

```
"turbulencemodel": {
    ...
    "sa": self.adflow.constants.spalartallmaras,        # Spalart-Allmaras model
    "menter sst": self.adflow.constants.mentersst,      # Menter SST model
    "langtry menter sst": self.adflow.constants.langtrymentersst,  # Langtry-Menter
    SST model
    ...

"eddyVisInfRatio": [float, 0.009],     # Default freestream eddy viscosity ratio, but
     can be changed in run script
"turbIntensityInf": [float, 0.001],    # Default freestream turbulence intensity, but
     can be changed in run script
```

Bibliography

[1] Robert H. Bush, Thomas S. Chyczewski, Karthikeyan Duraisamy, Bernhard Eisfeld, Christopher L. Rumsey, and Brian R. Smith. “Recommendations for Future Efforts in RANS Modeling and Simulation”. In: (2019). AIAA Paper 2019-0317, Session: Modeling of Turbulence. DOI: 10.2514/6.2019-0317. URL: https://doi.org/10.2514/6.2019-0317 (cit. on p. 1).

[2] Y. Lian and W. Shyy. “Laminar-Turbulent Transition of a Low Reynolds Number Rigid or Flexible Airfoil”. In: *AIAA Journal* 45.7 (2007), pp. 1501–1513. DOI: 10.2514/1.25812 (cit. on p. 1).

[3] Thomas Henrik Hansen. “Modeling the Performance of the Standard Cirrus Glider using Navier–Stokes CFD”. In: *Technical Soaring* 38.1 (2014). Royal Institute of Technology (KTH), Stockholm, Sweden, pp. 5–14 (cit. on p. 1).

[4] Mahmoud Salari and Amin Rava. “Numerical Investigation of Hydrodynamic Flow Over an AUV Moving in the Water-surface Vicinity Considering the Laminar-turbulent Transition”. In: *Journal of Marine Science and Application* 16.3 (2017), pp. 298–304. DOI: 10.1007/s11804-017-1422-x (cit. on p. 1).

[5] Robert S. Ehrmann and E. B. White. “Effect of Blade Roughness on Transition and Wind Turbine Performance”. In: (2015). SNL-NM, Albuquerque, NM. URL: https://www.osti.gov/servlets/purl/1427238 (cit. on p. 1).

[6] Da-Qing Li, Per Lindell, and Sofia Werner. “Transitional Flow on Model Scale Propellers and Their Likely Influence on Performance Prediction”. In: *Proceedings of the Sixth International Symposium on Marine Propulsors (SMP’19)*. SessionWB2.3. CNR-INM / National Research Council of Italy. Rome, Italy, May 2019, II-125–II–125. DOI: 10.3390/jmse7120427. URL: https://doi.org/10.3390/jmse7120427 (cit. on p. 1).

[7] Jan-Sören Fischer. “Laminar Wing Design: A Framework for Transition Delay Using Linear Stability Theory and Adjoint Optimization”. PhD thesis. Enschede, Netherlands: University of Twente, June 2022. DOI: 10.3990/1.9789036553957 (cit. on p. 1).

[8] *NASA Transition Modeling Workshop*. https://transitionmodeling.larc.nasa.gov/nasa-transition-modeling-workshop-sep-2017/. 2017 (cit. on p. 1).

[9] Marie Denison. “1st AIAA CFD Transition Modeling and Prediction Workshop: OVERFLOW Results for the SST and Langtry-Menter Models”. In: *AIAA SciTech Forum and Exposition*. AIAA. San Diego, CA, Jan. 2022, p. 0908. DOI: `10.2514/6.2022-0908` (cit. on p. 1).

[10] David C. Wilcox. “Reassessment of the Scale-Determining Equation for Advanced Turbulence Models”. In: *AIAA Journal* 26.11 (1988). Published online 17 May 2012; original pub. Nov 1988, pp. 1299–1310. DOI: `10.2514/3.10041` (cit. on p. 1).

[11] Philippe Spalart and Steven Allmaras. “A One-Equation Turbulence Model for Aerodynamic Flows”. In: *AIAA* 439 (Jan. 1992). DOI: `10.2514/6.1992-439` (cit. on p. 1).

[12] David C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, 2006 (cit. on pp. 1, 9, 13, 29, 30).

[13] Christopher L. Rumsey. “Apparent Transition Behavior of Widely-used Turbulence Models”. In: *International Journal of Heat and Fluid Flow* 28.6 (2007), pp. 1460–1471. DOI: `10.1016/j.ijheatfluidflow.2007.04.003` (cit. on p. 1).

[14] Robert Edward Mayle. “The Role of Laminar-Turbulent Transition in Gas Turbine Engines”. In: (1991), V005T17A001. DOI: `10.1115/91-GT-261`. URL: `https://doi.org/10.1115/91-GT-261` (cit. on pp. 1, 4).

[15] Erik Dick and Slawomir Kubacki. “Transition Models for Turbomachinery Boundary Layer Flows: A Review”. In: *International Journal of Turbomachinery, Propulsion and Power* 2.2 (2017), p. 4. DOI: `10.3390/ijtpp2020004`. URL: `https://doi.org/10.3390/ijtpp2020004` (cit. on pp. 1, 4).

[16] Parviz Moin and Krishnan Mahesh. “Direct Numerical Simulation: A Tool in Turbulence Research”. In: *Annual Review of Fluid Mechanics* 30 (1998), pp. 539–578. DOI: `10.1146/annurev.fluid.30.1.539`. URL: `https://doi.org/10.1146/annurev.fluid.30.1.539` (cit. on p. 1).

[17] V. Michelassi, J. G. Wissink, J. Fröhlich, and W. Rodi. “Large-Eddy Simulation of Flow Around Low-Pressure Turbine Blade with Incoming Wakes”. In: *AIAA Journal* 50.5 (2012). Published Online: 17 May 2012, pp. 1066–1079. DOI: `10.2514/2.6832` (cit. on pp. 1, 7).

[18] J. L. van Ingen. “A Suggested Semi-Empirical Method for the Calculation of the Boundary Layer Transition Region”. In: *TU Delft Report* (1956) (cit. on p. 1).

[19] Michael Piotrowski and David W. Zingg. “Investigation of a Local Correlation-based Transition Model in a Newton-Krylov Algorithm”. In: *AIAA Aviation Forum, Aerodynamic Design: Analysis, Methodologies, and Optimization Techniques V*. AIAA 2019-2299, Published Online: 6 Jan 2019. Jan. 2019. DOI: `10.2514/6.2019-2299`. URL: `https://doi.org/10.2514/6.2019-2299` (cit. on pp. 1, 7, 45, 47).

[20] V. C. Patel, W. Rodi, and G. Scheuerer. “Turbulence Models for Near-Wall and Low Reynolds Number Flows - A Review”. In: *AIAA Journal* 23.9 (1985), pp. 1308–1319. DOI: `10.2514/3.10046` (cit. on p. 1).

[21] Christopher L. Rumsey, Barbara A. Pettersson Reif, and Thomas B. Gatski. “Arbitrary Steady-State Solutions with the k–ϵ Model”. In: *AIAA Journal* 44.7 (2006), pp. 1586–1592. DOI: `10.2514/1.17951` (cit. on pp. 1, 7).

[22] Robin B. Langtry and Florian R. Menter. “Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes”. In: *AIAA Journal* 47.12 (Dec. 2009), pp. 2894–2906. DOI: `10.2514/1.42362`. URL: `https://doi.org/10.2514/1.42362` (cit. on pp. 1, 8, 13, 17, 44, 51, 54, 55).

[23] D. Keith Walters and Davor Cokljat. “A Three-Equation Eddy-Viscosity Model for Reynolds-Averaged Navier–Stokes Simulations of Transitional Flow”. In: *Journal of Fluids Engineering* 130.12 (2008), p. 121401. DOI: `10.1115/1.2979230`. URL: `https://doi.org/10.1115/1.2979230` (cit. on pp. 1, 8).

[24] J. G. Coder and M. D. Maughmer. “Computational Fluid Dynamics Compatible Transition Modeling Using an Amplification Factor Transport Equation”. In: *AIAA Journal* 52.11 (2014), pp. 2506–2512. DOI: `10.2514/1.J052905` (cit. on pp. 2, 8).

[25] Peyman Khayatzadeh and Siva Nadarajah. “Aerodynamic Shape Optimization via Discrete Viscous Adjoint Equations for the k–ω SST Turbulence and γ–Re$_\theta$ Transition Models”. In: *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Published Online: 14 Jun 2012. AIAA. 2011. DOI: `10.2514/6.2011-1247`. URL: `https://doi.org/10.2514/6.2011-1247` (cit. on p. 2).

[26] Peyman Khayatzadeh and Sivakumaran Nadarajah. “Aerodynamic Shape Optimization of Natural Laminar Flow (NLF) Airfoils”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Published Online: 6 Nov 2012. AIAA. Jan. 2012. DOI: `10.2514/6.2012-0061`. URL: `https://doi.org/10.2514/6.2012-61` (cit. on p. 2).

[27] Jing Li, Zhenghong Gao, Jiangtao Huang, and Ke Zhao. “Robust Design of NLF Airfoils”. In: *Chinese Journal of Aeronautics* 26.2 (2013), pp. 309–318. DOI: `10.1016/j.cja.2013.02.007`. URL: `https://doi.org/10.1016/j.cja.2013.02.007` (cit. on p. 2).

[28] Ke Zhao, Zhenghong Gao, and Jiangtao Huang. “Robust Design of Natural Laminar Flow Supercritical Airfoil by Multi-objective Evolution Method”. In: *Applied Mathematics and Mechanics (English Edition)* 35 (2014), pp. 191–202. DOI: `10.1007/s10483-014-1783-6`. URL: `https://doi.org/10.1007/s10483-014-1783-6` (cit. on p. 2).

[29] Martin Robitaille, Ali Mosahebi, and Éric Laurendeau. “Design of Adaptive Transonic Laminar Airfoils Using the the -Re~t Transition Model”. In: *Aerospace Science and Technology* 47 (2015), pp. 31–39. DOI: `10.1016/j.ast.2015.06.027`. URL: `https://doi.org/10.1016/j.ast.2015.06.027` (cit. on p. 2).

[30] Ramy Rashad and David W. Zingg. “Aerodynamic Shape Optimization for Natural Laminar Flow Using a Discrete-Adjoint Approach”. In: *AIAA Journal* 54.10 (2016). Published Online: 5 Aug 2016, pp. 3153–3168. DOI: `10.2514/1.J054940` (cit. on p. 2).

[31] Julien Cliquet, Robert Houdeville, and Daniel Arnal. “Application of Laminar-Turbulent Transition Criteria in Navier-Stokes Computation”. In: (2007). Session: FD-13: CFD RANS Modeling and Application, Published Online: 18 Jun 2012, AIAA 2007–515. DOI: `10.2514/6.2007-515` (cit. on p. 2).

[32] Gustavo L.O. Halila, Joaquim R.R.A. Martins, and Krzysztof J. Fidkowski. “Adjoint-based aerodynamic shape optimization including transition to turbulence effects”. In: *Aerospace Science and Technology* 107 (2020), p. 106243. DOI: `10.1016/j.ast.2020.106243`. URL: `https://doi.org/10.1016/j.ast.2020.106243` (cit. on pp. 2, 45, 47, 66).

[33] H. Schlichting and K. Gersten. *Boundary-Layer Theory*. Springer, 2016 (cit. on pp. 4, 5, 30, 51).

[34] G. B. Schubauer and H. K. Skramstad. “Laminar Boundary-Layer Oscillations and Transition on a Flat Plate”. In: *Journal of Research of the National Bureau of Standards* 38 (1947), pp. 251–292 (cit. on p. 4).

[35] Stephen B. Pope. *Turbulent Flows*. Cambridge, UK: Cambridge University Press, 2000 (cit. on pp. 4, 30).

[36] Fei Li and Mujeeb R. Malik. “Spectral Analysis of Parabolized Stability Equations”. In: *Computers & Fluids* 25.4 (1996), pp. 361–378. DOI: `10.1016/S0045-7930(96)00044-8`. URL: `https://doi.org/10.1016/S0045-7930(96)00044-8` (cit. on p. 4).

[37] William S. Saric, Helen L. Reed, and Edward J. Kerschen. “Boundary-Layer Receptivity to Freestream Disturbances”. In: *Annual Review of Fluid Mechanics* 34 (2002), pp. 291–319. DOI: `10.1146/annurev.fluid.34.082701.161921`. URL: `https://doi.org/10.1146/annurev.fluid.34.082701.161921` (cit. on p. 4).

[38] P. S. Klebanoff. “Effects of Free-Stream Turbulence on a Laminar Boundary Layer”. In: *Bulletin of the American Physical Society* 16.11 (1971), p. 1323 (cit. on p. 4).

[39] William S. Saric, Helen L. Reed, and Edward B. White. “Stability and Transition of Three-Dimensional Boundary Layers”. In: *Annual Review of Fluid Mechanics* 35 (2003), pp. 413–440. DOI: `10.1146/annurev.fluid.35.101101.161045`. URL: `https://doi.org/10.1146/annurev.fluid.35.101101.161045` (cit. on p. 5).

[40] M. M. O’Meara and T. J. Mueller. “Laminar Separation Bubble Characteristics on an Airfoil at Low Reynolds Numbers”. In: *AIAA Journal* 25.8 (1987), pp. 1033–1041. DOI: `10.2514/3.9739` (cit. on pp. 5, 6, 34).

[41] J. M. Russell. “Length and Bursting of Separation Bubbles: A Physical Interpretation”. In: *Science and Technology of Low Speed Motorless Flight*. Vol. 2085. NASA Conference Publication. NASA, 1979 (cit. on pp. 5, 6).

[42] Thomas J. Mueller and James D. DeLaurier. “Aerodynamics of Small Vehicles”. In: *Annual Review of Fluid Mechanics* 35 (2003), pp. 89–111. DOI: `10.1146/annurev.fluid.35.101101.161102` (cit. on pp. 5, 6).

[43] Florian R. Menter, Richard B. Langtry, and Stefan Völker. “Transition modelling for general purpose CFD codes”. In: *Flow, Turbulence and Combustion* 77.1 (2006), pp. 277–303. DOI: 10.1007/s10494-006-9047-1 (cit. on p. 6).

[44] Parviz Moin and Krishnan Mahesh. “Direct Numerical Simulation: A Tool in Turbulence Research”. In: *Annual Review of Fluid Mechanics* 30 (1998), pp. 539–578. DOI: 10.1146/annurev.fluid.30.1.539 (cit. on p. 7).

[45] B. J. Abu-Ghannam and R. Shaw. “Natural Transition of Boundary Layers—The Effects of Turbulence, Pressure Gradient, and Flow History”. In: *Journal of Mechanical Engineering Science* 22.5 (1980), pp. 213–228. DOI: 10.1243/JMES_JOUR_1980_022_043_02 (cit. on p. 7).

[46] J. Steelant and E. Dick. “Modelling of Bypass Transition with Conditioned Navier–Stokes Equations Coupled to an Intermittency Transport Equation”. In: *International Journal for Numerical Methods in Fluids* 23.3 (1996), pp. 193–220. DOI: 10.1002/(SICI)1097-0363(19960915)23:3<193::AID-FLD383>3.0.CO;2-W (cit. on p. 8).

[47] S. J. Dhawan and R. Narasimha. “Some Properties of Boundary Layer Flow During the Transition from Laminar to Turbulent Motion”. In: *Journal of Fluid Mechanics* 3.4 (1958), pp. 418–436. DOI: 10.1017/S0022112058000180 (cit. on p. 8).

[48] Y. B. Suzen and P. G. Huang. “Modeling of Flow Transition Using an Intermittency Transport Equation”. In: *Journal of Fluids Engineering* 122.2 (2000), pp. 273–284. DOI: 10.1115/1.483255. URL: https://doi.org/10.1115/1.483255 (cit. on p. 8).

[49] Ji R. Cho and Myung K. Chung. “A k–ϵ–γ Equation Turbulence Model”. In: *Journal of Fluid Mechanics* 237 (1992), pp. 301–322. DOI: 10.1017/S0022112092003422. URL: https://doi.org/10.1017/S0022112092003422 (cit. on p. 8).

[50] R. Pecnik, W. Sanz, A. Gehrer, and H. Jericha. “Transition Modeling Using Two Different Intermittency Transport Equations”. In: *Flow, Turbulence and Combustion* 70 (2003), pp. 299–323. DOI: 10.1023/B:APPL.0000004983.49982.69. URL: https://doi.org/10.1023/B:APPL.0000004983.49982.69 (cit. on p. 8).

[51] F. R. Menter, R. B. Langtry, S. R. Likki, Y. B. Suzen, P. G. Huang, and S. Völker. “A Correlation-Based Transition Model Using Local Variables—Part I: Model Formulation”. In: *Journal of Turbomachinery* 128.3 (Mar. 2006), pp. 413–422. DOI: 10.1115/1.2184352. URL: https://doi.org/10.1115/1.2184352 (cit. on pp. 8, 44).

[52] R. B. Langtry, F. R. Menter, S. R. Likki, Y. B. Suzen, P. G. Huang, and S. Völker. “A Correlation-Based Transition Model Using Local Variables—Part II: Test Cases and Industrial Applications”. In: *Journal of Turbomachinery* 128.3 (Mar. 2006), pp. 423–432. DOI: 10.1115/1.2184353. URL: https://doi.org/10.1115/1.2184353 (cit. on p. 8).

[53] F. R. Menter, P. E. Smirnov, T. Liu, and R. Avancha. “A One-Equation Local Correlation-Based Transition Model”. In: *Flow, Turbulence and Combustion* 95.4 (Dec. 2015), pp. 583–619. DOI: 10.1007/s10494-015-9622-4 (cit. on p. 8).

[54] Shivaji Medida and James Baeder. "Application of the Correlation-Based γ–$\mathrm{Re}_{\theta_\mathrm{t}}$ Transition Model to the Spalart–Allmaras Turbulence Model". In: *AIAA Aviation Forum*. Published Online: 14 Jun 2012. June 2012. DOI: 10.2514/6.2011-3979. URL: https://doi.org/10.2514/6.2011-3979 (cit. on p. 8).

[55] S. Medida. "Correlation-based Transition Modeling for External Aerodynamic Flows". PhD thesis. University of Maryland, 2014 (cit. on p. 8).

[56] Shivaji Medida, David A. Corson, and Michael Barton. "Implementation and Validation of Correlation-Based Transition Models in AcuSolve". In: *AIAA Aviation Forum*. June 2016. DOI: 10.2514/6.2016-3478. URL: https://doi.org/10.2514/6.2016-3478 (cit. on p. 8).

[57] J. G. Coder and M. D. Maughmer. "Application of the Amplification Factor Transport Transition Model to the Shear Stress Transport Model". In: *53rd AIAA Aerospace Sciences Meeting*. 2015. DOI: 10.2514/6.2015-0588 (cit. on p. 8).

[58] Rui Miguel Alves Lopes. "Simulation of Transition from Laminar to Turbulent Regime in Practical Applications of Incompressible Flow". PhD thesis. Instituto Superior Técnico, Universidade de Lisboa, 2021 (cit. on pp. 8, 9, 29, 34, 35, 45).

[59] John D. Anderson. *Computational Fluid Dynamics*. McGraw-Hill, 1995 (cit. on pp. 9, 34, 35).

[60] Tae S. Chung. *Computational Fluid Dynamics: A Practical Approach*. Springer, 2002 (cit. on pp. 9, 10).

[61] Jiri Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2001 (cit. on p. 10).

[62] B. E. Launder, G. J. Reece, and W. Rodi. "Progress in the Development of a Reynolds-Stress Turbulence Closure". In: *Journal of Fluid Mechanics* 68.3 (1975), pp. 537–566. DOI: 10.1017/S0022112075001814. URL: https://doi.org/10.1017/S0022112075001814 (cit. on p. 10).

[63] F. R. Menter. "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications". In: vol. 32. 8. 1994, pp. 1598–1605. DOI: 10.2514/3.12149. URL: https://doi.org/10.2514/3.12149 (cit. on pp. 11, 13).

[64] F. R. Menter, M. Kuntz, and R. Langtry. "Ten Years of Industrial Experience with the SST Turbulence Model". In: *Turbulence, Heat and Mass Transfer 4*. Vol. 4. Begell House, 2003, pp. 625–632 (cit. on p. 11).

[65] F. R. Menter. *Review of the Shear-Stress Transport Turbulence Model Experience from an Industrial Perspective*. Tech. rep. 4. 2009, pp. 305–316. DOI: 10.1080/10618560902773387 (cit. on p. 12).

[66] Clark Pederson. *Langtry-Menter Four-Equation Transition Model*. https://turbmodels.larc.nasa.gov/langtrymenter_4eqn.html. Accessed: 2025-06-17. 2025 (cit. on p. 13).

[67] Alejandra Uranga. "Investigation of transition to turbulence at low Reynolds numbers using Implicit Large Eddy Simulations with a Discontinuous Galerkin method". Submitted to the Department of Aeronautics and Astronautics. Ph.D. thesis. Cambridge, MA: Massachusetts Institute of Technology, 2011 (cit. on pp. 14, 35).

[68] Joaquim R.R.A. Martins and Andrew Ning. "Engineering design optimization using a unified adjoint method". In: *AIAA Journal* 59.6 (2021), pp. 2193–2208 (cit. on pp. 20, 50).

[69] Michael B Giles and Niles A Pierce. "An introduction to the adjoint approach to design". In: *Flow, Turbulence and Combustion* 65 (2000), pp. 393–415 (cit. on p. 20).

[70] Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Second. USA: Society for Industrial and Applied Mathematics, 2008. ISBN: 0898716594 (cit. on pp. 20, 24, 50).

[71] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso. "The Complex-Step Derivative Approximation". In: *ACM Transactions on Mathematical Software* 29.3 (2003), pp. 245–262. DOI: 10.1145/838250.838251. URL: https://doi.org/10.1145/838250.838251 (cit. on p. 20).

[72] Joaquim R. R. A. Martins and Andrew Ning. *Engineering Design Optimization*. Cambridge University Press, 2021 (cit. on p. 21).

[73] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. "Automatic Differentiation in Machine Learning: A Survey". In: (2018). arXiv: 1502.05767 [cs.SC]. URL: https://arxiv.org/abs/1502.05767 (cit. on p. 24).

[74] INRIA. *Tapenade: Automatic Differentiation Software*. http://tapenade.inria.fr:8080/tapenade/index.jsp. Accessed: 2025-06-19. 2025 (cit. on p. 24).

[75] Charles A. Mader, Gaetan K. W. Kenway, Anil Yildirim, and Joaquim R. R. A. Martins. "ADflow: An Open-Source Computational Fluid Dynamics Solver for Aerodynamic and Multidisciplinary Optimization". In: *AIAA Journal* 58.6 (2020), pp. 2654–2669. DOI: 10.2514/1.I010796. URL: https://doi.org/10.2514/1.I010796 (cit. on pp. 24–26, 67).

[76] MPI Forum. *MPI: A Message Passing Interface Standard*. https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf. Version 3.1. 2015 (cit. on pp. 25, 44).

[77] G. Klopfer, C. Hung, R. Van der Wijngaart, and J. Onufer. "A Diagonalized Diagonal Dominant Alternating Direction Implicit (D3ADI) Scheme and Subiteration Correction". In: *29th AIAA Fluid Dynamics Conference*. 98-2610. AIAA. Albuquerque, NM, 1998 (cit. on p. 25).

[78] Anil Yildirim, Gaetan K. W. Kenway, Charles A. Mader, and Joaquim R. R. A. Martins. "A Jacobian-Free Approximate Newton–Krylov Startup Strategy for RANS Simulations". In: *Journal of Computational Physics* 397 (2019), p. 108741. DOI: 10.1016/j.jcp.2019.06.018. URL: https://doi.org/10.1016/j.jcp.2019.06.018 (cit. on p. 25).

[79] M. Nemec and D. W. Zingg. "Newton–Krylov Algorithm for Aerodynamic Design Using the Navier–Stokes Equations". In: *AIAA Journal* 40.6 (2002). Published online: 17 May 2012, pp. 1146–1154. DOI: `10.2514/2.1764` (cit. on p. 25).

[80] Gustavo Luiz Olichevis Halila, Anil Yildirim, Charles A. Mader, Krzysztof J. Fidkowski, and Joaquim R. R. A. Martins. "Linear Stability-Based Smooth Reynolds-Averaged Navier–Stokes Transition Model for Aerodynamic Flows". In: *AIAA Journal* (2021). Published Online: 22 Nov 2021. DOI: `10.2514/1.J060481`. URL: `https://doi.org/10.2514/1.J060481` (cit. on pp. 25, 26, 44, 45, 47, 66).

[81] MDO Lab. *ADflow Solvers Documentation.* `https://mdolab-adflow.readthedocs-hosted.com/en/latest/solvers.html`. Accessed: 2025-06-13. 2024 (cit. on pp. 25, 36).

[82] Patrick J. Roache. *Verification and Validation in Computational Science and Engineering.* Hermosa Publishers, 1998 (cit. on pp. 26, 27).

[83] David Goldberg. "What Every Computer Scientist Should Know About Floating-Point Arithmetic". In: *ACM Computing Surveys (CSUR)* 23.1 (Mar. 1991), pp. 5–48. DOI: `10.1145/103162.103163`. URL: `https://doi.org/10.1145/103162.103163` (cit. on p. 26).

[84] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing.* New York, NY, USA: Cambridge University Press, 2010. ISBN: 9780511760396. DOI: `10.1017/CBO9780511760396`. URL: `https://doi.org/10.1017/CBO9780511760396` (cit. on p. 26).

[85] Sai Hung Cheung, Todd A. Oliver, Ernesto E. Prudencio, Serge Prudhomme, and Robert D. Moser. "Bayesian Uncertainty Analysis with Applications to Turbulence Modeling". In: *Reliability Engineering & System Safety* 96.9 (2011), pp. 1137–1149. DOI: `10.1016/j.ress.2010.09.013` (cit. on p. 26).

[86] Gaetan K. W. Kenway, Graeme J. Kennedy, and Joaquim R. R. A. Martins. "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations". In: *AIAA Journal* 52.4 (2014), pp. 935–951. DOI: `10.2514/1.J052255`. URL: `https://doi.org/10.2514/1.J052255` (cit. on p. 26).

[87] James G. Coder, Thomas H. Pulliam, David Hue, Gaetan K. Kenway, and Anthony J. Sclafani. "Contributions to the 6th AIAA CFD Drag Prediction Workshop Using Structured Grid Methods". In: *AIAA Aviation Forum.* AIAA 2017-0960, Session: DPW-6 Results and Comparisons I, Published Online: 5 Jan 2017. 2017. DOI: `10.2514/6.2017-0960`. URL: `https://doi.org/10.2514/6.2017-0960` (cit. on p. 26).

[88] Ismail B. Celik, Urmila Ghia, Patrick J. Roache, Christopher J. Freitas, Hugh Coleman, and Peter E. Raad. "Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications". In: *ASME Journal of Fluids Engineering* 130.7 (2008). Published online July 22, 2008, p. 078001. DOI: `10.1115/1.2960953` (cit. on p. 27).

[89] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, 2002 (cit. on p. 27).

[90] Harry Versteeg and Weeratunge Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education, 2007 (cit. on pp. 28, 30).

[91] G. D. van Albada, B. van Leer, and W. W. Roberts. "A Comparative Study of Computational Methods in Cosmic Gas Dynamics". In: *Astronomy and Astrophysics* 108 (1982), pp. 76–84 (cit. on p. 28).

[92] P. R. Spalart and C. L. Rumsey. "Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations". In: *AIAA Journal* 45.10 (2007), pp. 2544–2553. DOI: 10.2514/1.29373 (cit. on p. 29).

[93] Ney Secco, Gaetan K. W. Kenway, Ping He, Charles A. Mader, and Joaquim R. R. A. Martins. *Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization*. 2021. DOI: 10.2514/1.J059491 (cit. on pp. 30, 67).

[94] Kitware, Inc. *ParaView: Open-source Scientific Visualization*. 2023. URL: https://www.paraview.org/ (cit. on p. 31).

[95] P. Catalano and R. Tognaccini. "RANS Analysis of the Low-Reynolds Number Flow Around the SD7003 Airfoil". In: *Aerospace Science and Technology* 15.8 (2011), pp. 615–626. DOI: 10.1016/j.ast.2011.01.009 (cit. on p. 31).

[96] E. Kadivar and E. Kadivar. "Computational study of the laminar to turbulent transition over the SD7003 airfoil in ground effect". In: *Thermophysics and Aeromechanics* 25.4 (2018), pp. 497–505. DOI: 10.1134/S0869864318040030. URL: https://doi.org/10.1134/S0869864318040030 (cit. on p. 31).

[97] Md Islam, Md Amzad Hossain, Md Uddin, and Mohammad Mashud. "Experimental Evaluation of Aerodynamics Characteristics of a Baseline Airfoil". In: *American Journal of Engineering Research (AJER)* 4 (Jan. 2015), pp. 91–96 (cit. on p. 32).

[98] M. S. Selig and B. D. McGranahan. *Wind Tunnel Aerodynamic Tests of Six Airfoils for Use on Small Wind Turbines*. Tech. rep. 4. Nov. 2004, pp. 986–1001. DOI: 10.1115/1.1793208. URL: https://doi.org/10.1115/1.1793208 (cit. on p. 33).

[99] Marcello Zani. *ADflow: Branch SST2003_LM2009 – SST 2003 and Langtry-Menter 2009 Transition Model Implementation*. https://github.com/marszani/adflow/tree/SST2003_LM2009. Accessed: 2025-06-12. 2025 (cit. on p. 36).

[100] MDO Lab. *ADflow Performance Guidelines*. https://mdolab-adflow.readthedocs-hosted.com/en/latest/performance.html#performance. Accessed: 2025-06-13. 2025 (cit. on p. 44).

[101] Richard B. Langtry. "A Correlation-Based Transition Model Using Local Variables for Unstructured Parallelized CFD Codes". PhD thesis. University of Stuttgart, 2006 (cit. on pp. 44, 45).

[102] M. Carreño Ruiz and D. D'Ambrosio. "Validation of the Transition Model for Airfoils Operating in the Very Low Reynolds Number Regime". In: *Flow, Turbulence and Combustion* 109 (2022), pp. 279–308. DOI: 10.1007/s10494-022-00331-z. URL: https://doi.org/10.1007/s10494-022-00331-z (cit. on pp. 45, 47, 51).

[103] Keerati Suluksna, Pramote Dechaumphai, and Ekachai Juntasaro. "Correlations for modeling transitional boundary layers under influences of freestream turbulence and pressure gradient". In: *International Journal of Heat and Fluid Flow* 30.1 (2009), pp. 64–75. DOI: 10.1016/j.ijheatfluidflow.2008.09.004. URL: https://doi.org/10.1016/j.ijheatfluidflow.2008.09.004 (cit. on p. 45).

[104] Yayun Shi, Charles A. Mader, Sicheng He, Gustavo L. O. Halila, and Joaquim R. R. A. Martins. "Natural Laminar-Flow Airfoil Optimization Design Using a Discrete Adjoint Approach". In: *AIAA Journal* (2020). Published Online: 24 Sep 2020. DOI: 10.2514/1.J058944. URL: https://doi.org/10.2514/1.J058944 (cit. on pp. 45, 47).

[105] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* See Chapter 6: Deep Feedforward Networks (Section 6.2.2) for discussion of the ReLU activation function. MIT Press, 2016. URL: https://www.deeplearningbook.org (cit. on p. 46).

[106] Marcello Zani. *ADflow - Smooth* F$_{onset}$ *modification for SST2003-LM2009.* https://github.com/marszani/adflow/tree/smooth_F_onset_SST2003_LM2009. Accessed: 2025-07-02 (cit. on p. 46).

[107] Siemens Digital Industries Software. *STAR-CCM+ User Guide.* Siemens PLM Software. 2023 (cit. on pp. 47, 51, 57, 58).

[108] Marcello Zani. *Transitional Flow Preprocessing, Simulation, and Postprocessing.* GitHub repository. 2025. URL: https://github.com/marszani/transitional-flow-preprocessing_simulation_postprocessing (visited on 06/16/2025) (cit. on p. 48).

[109] Heng Xiao and Paola Cinnella. "Quantification of Model Uncertainty in RANS Simulations: A Review". In: *Progress in Aerospace Sciences* 108 (2019), pp. 1–31. DOI: 10.1016/j.paerosci.2018.10.001. URL: https://doi.org/10.1016/j.paerosci.2018.10.001 (cit. on p. 51).

[110] Jeffrey P. Slotnick, Abdollah Khodadoust, Juan J. Alonso, David L. Darmofal, William D. Gropp, Elizabeth J. Lurie, and Dimitri J. Mavriplis. *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences.* Tech. rep. NASA, 2014 (cit. on p. 51).

[111] Atsushi Tateishi, Naoki Tani, Yasuhiro Okamura, and Masaaki Hamabe. "LES Prediction of Transitional Flows in LP Turbine Cascades: Flow Phenomena and Effects of Numerical Setup". In: (Apr. 2022). DOI: 10.33737/gpps21-tc-61 (cit. on p. 51).

[112] Y. Bin, G. I. Park, Y. Lv, and X. I. A. Yang. "Large Eddy Simulation of Separated Flows on Unconventionally Coarse Grids". In: *ASME Journal of Fluids Engineering* 146.9 (Sept. 2024). Published online March 26, 2024, p. 091501. DOI: 10.1115/1.4064851. URL: https://doi.org/10.1115/1.4064851 (cit. on p. 51).

[113] P. Catalano, D. de Rosa, V. D'Alessandro, A. Marouf, Y. Hoarau, M. Miozzi, and M. Righi. "Modelling Laminar Separation Bubbles at Low Reynolds Number". In: *AIAA SCITECH 2024 Forum*. 2024. DOI: `10.2514/6.2024-1347`. URL: `https://arc.aiaa.org/doi/10.2514/6.2024-1347` (cit. on pp. 51, 57–60, 62–65).

[114] Christopher L. Rumsey and Philippe R. Spalart. *Turbulence Model Behavior in Low Reynolds Number Regions of Aerodynamic Flowfields*. Tech. rep. 7. Published online May 2, 2012. May 2012, pp. 1534–1544. DOI: `10.2514/1.39947`. URL: `https://doi.org/10.2514/1.39947` (cit. on p. 51).

[115] Massimo Miozzi, Fabio Di Felice, Christian Klein, and Marco Costantini. "Taylor hypothesis applied to direct measurement of skin friction using data from Temperature Sensitive Paint". In: *Experimental Thermal and Fluid Science* 109 (2019), p. 109913. DOI: `10.1016/j.expthermflusci.2019.109913`. URL: `https://doi.org/10.1016/j.expthermflusci.2019.109913` (cit. on pp. 60–62).

[116] Airfoil Tools. *NACA 0015 Airfoil - XFOIL Calculations*. `http://airfoiltools.com/polar/details?polar=xf-naca0015-il-200000`. Accessed 2025-06-10. 2024 (cit. on pp. 60, 62).

[117] R. J. McGhee, B. S. Walker, and B. F. Millard. *Experimental Results for the Eppler 387 Airfoil at Low Reynolds Numbers in the Langley Low-Turbulence Pressure Tunnel*. NASA Technical Memorandum 4062. NASA Langley Research Center, 1988 (cit. on pp. 63–65).

[118] Jichao Li, Xiaosong Du, and Joaquim R. R. A. Martins. "Machine Learning in Aerodynamic Shape Optimization". In: *Progress in Aerospace Sciences* 135 (2022), p. 100849. DOI: `10.1016/j.paerosci.2022.100849` (cit. on p. 66).

[119] "MACH-Aero: High-Fidelity Multidisciplinary Design Optimization for Aircraft". In: *MDO Lab Documentation* (2021). `https://mdolab-mach-aero.readthedocs-hosted.com/en/latest//` (cit. on p. 66).

[120] Thomas W. Sederberg and Scott R. Parry. "Free-form Deformation of Solid Geometric Models". In: *SIGGRAPH Comput. Graph.* 20.4 (1986), pp. 151–160. DOI: `10.1145/15886.15903`. URL: `https://doi.org/10.1145/15886.15903` (cit. on pp. 67, 69).

[121] Ella Wu, Gaetan Kenway, Charles A. Mader, John Jasa, and Joaquim R. R. A. Martins. "pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems". In: *Journal of Open Source Software* 5.54 (2020), p. 2564. DOI: `10.21105/joss.02564` (cit. on p. 67).

[122] Philip E. Gill, Walter Murray, and Michael A. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". In: *SIAM Review* 47.1 (2005), pp. 99–131. DOI: `10.1137/S0036144504446096`. URL: `https://doi.org/10.1137/S0036144504446096` (cit. on p. 67).

[123] Dieter Kraft. *A Software Package for Sequential Quadratic Programming.* DFVLR-FB 88-28. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR), 1988 (cit. on pp. 67, 68).

[124] Hannah M. Hajdik et al. "pyGeo: A geometry package for multidisciplinary design optimization". In: *Journal of Open Source Software* 8.87 (2023), p. 5319. DOI: `10.21105/joss.05319` (cit. on p. 67).

[125] Gaetan K. W. Kenway, Charles A. Mader, Sicheng He, and Joaquim R. R. A. Martins. "Effective Adjoint Approaches for Computational Fluid Dynamics". In: *Progress in Aerospace Sciences* 110 (Oct. 2019), p. 100542. DOI: `10.1016/j.paerosci.2019.05.002` (cit. on p. 67).

[126] Milovan Peric, Joel H. Ferziger, and Robert L. Street. *Computational Methods for Fluid Dynamics.* Springer, 2002 (cit. on p. 67).

[127] Renaldo V. Jenkins. *NASA SC(2)-0714 Airfoil Data Corrected for Sidewall Boundary-Layer Effects in the Langley 0.3-Meter Transonic Cryogenic Tunnel.* NASA Technical Paper NASA-TM-102289, NASA TP-2890. Hampton, Virginia: NASA Langley Research Center, 1989. URL: `https://ntrs.nasa.gov/api/citations/19890008197/downloads/19890008197.pdf` (cit. on p. 68).

[128] *MACH-Aero Tutorials: Aerodynamic Shape Optimization.* `https://mdolab-mach-aero.readthedocs-hosted.com/en/latest/machAeroTutorials/opt_aero.html`. Accessed: 2025-06-15 (cit. on pp. 68, 70).

[129] Galen W. Ng, Anil Yildirim, Harold Youngren, Andrew Lamkin, Eirikur Jonsson, and Joaquim R. R. A. Martins. "Design Optimization of America's Cup AC75 Hydrofoil Sections with Flaps". In: *Journal of Sailing Technology* 10 (2025), pp. 50–73. DOI: `10.5957/jst/2025.10.1.50` (cit. on p. 75).

Dedications

This thesis is dedicated to my parents,
and especially to my dad —
he really wanted me to say this, so here it is:
***GRAZIE MAMMA, GRAZIE PAPÀ**,*
and to the rest of my family for bearing with me
and supporting me throughout this journey.