STONY BROOK UNIVERSITY
COLLEGE OF ENGINEERING AND APPLIED SCIENCES

Numerical Investigation of Control Surface's Subsonic External Aerodynamics

A Computational Fluid Dynamics Study using OpenFOAM v2412

Shafayat Alam
Undergraduate Researcher

Advanced Fluid Dynamics, Propulsion, and Energy Lab (AFPEL)
Advisor: Dr. Foluso Ladeinde

Department of Mechanical Engineering
Stony Brook University

Subject Area: Aerodynamics, CFD,
Keywords: OpenFOAM, Trapezoidal Fins, RANS

Abstract

.... Using the open-source software OpenFOAM v2412, we perform Reynolds-Averaged Navier–Stokes (RANS) simulations on a three-dimensional trapezoidal fin geometry with a root chord of 5.50 inches and an aspect ratio of 0.720 at a Mach number of 0.7. The work addresses the accessibility gap between simplified flight simulators and industrial-grade CFD tools by establishing a fully open-source workflow for student-led rocketry teams. Key aerodynamic coefficients—lift, drag, and moment—are decomposed into pressure and viscous components to provide detailed insight into force generation mechanisms. The Spalart–Allmaras turbulence model is employed to capture boundary-layer effects and trailing-edge separation. A finite volume discretization scheme ensures global conservation and numerical stability. The study validates the numerical approach through verification and physical validation protocols, integrating wind-tunnel data and subscale flight tests. Furthermore, the framework is designed to support subsequent shape optimization and sensitivity analysis using adjoint and variance-based methods. This research demonstrates that open-source CFD can serve as a robust, cost-effective alternative to proprietary software, thereby democratizing high-fidelity aerodynamic analysis for academic and student rocketry investigations.

1 Motivation

The design and optimization of high-power sounding rockets within student-led engineering organizations often encounter a significant technical bottleneck: the ”accessibility gap” between simplified flight simulation tools and high-fidelity aerodynamic analysis. While standard flight simulation software provides essential data regarding trajectory and static stability, these tools typically rely on semi-empirical methods and linearized aerodynamics. Such methods are often insufficient for capturing complex flow phenomena—including shock-body interactions, localized pressure distributions on non-standard geometries, and base drag characteristics—that are critical to the performance of a custom-built vehicle.

In the current academic landscape, high-fidelity Computational Fluid Dynamics (CFD) is frequently gated behind prohibitive licensing costs of proprietary software. While university research laboratories often possess these resources, independent student design teams operating on limited budgets find them financially unreachable. This creates a reliance on ”black-box” simulators, which limits a team’s ability to conduct rigorous sensitivity studies or optimize vehicle geometry for specific flight regimes.

As a Mechanical Engineering undergraduate and a member of the Vehicle Design team, I identified a unique opportunity to address this disparity. By pursuing independent research under the mentorship of a dedicated CFD research laboratory, this project seeks to demonstrate that open-source CFD frameworks, specifically OpenFOAM v2412, can serve as a robust, no-cost alternative to industrial software.

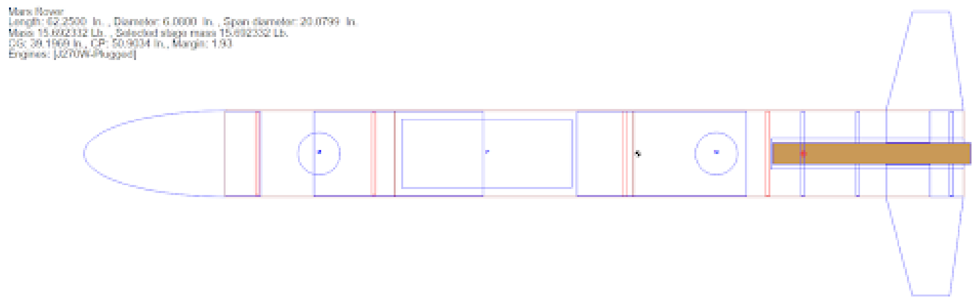
Mars Rover
Length: 62.2500 In. , Diameter: 6.0000 In. , Span diameter: 20.0799 In.
Mass 15.602332 Lb. , Selected stage mass 15.602332 Lb.
CG: 39.1969 In. , CP: 50.9034 In. , Margin: 1.93
Engines: [J270W-Plugged]

Figure 1: Sample Launch Vehicle System Design of Stony Brook University 2026 Rocket Team

Crucially, this research adopts the perspective that numerical algorithms are no substitute for physical experimentation. Every numerical algorithm possesses characteristic error patterns, such as numerical diffusion, false diffusion, or numerical flow[1], which can only be mitigated by comparing results with experimental data. To ensure the validity of these models, our team utilizes a multifaceted approach to verification and validation. We leverage both subscale flight launches and access to a localized wind tunnel facility capable of meeting our experimental needs. By synthesizing these physical measurements with high-fidelity CFD via the OpenFOAM v2412 open-source C++ library, we transform the simulation from a standalone estimate into a powerful additional problem-solving tool that enhances the reliability of the entire design cycle.

The primary motivation of this work is to democratize high-fidelity analysis for the student rocketry community. By developing a comprehensive workflow using open-source tools tailored specifically to rocket vehicle design, this paper aims to:

- **Empower** student teams to transcend the limitations of simplified simulators by providing a transparent, high-fidelity analysis pipeline.
- **Establish a Reference Framework** based on v2412 that lowers the barrier to entry for undergraduate engineers, allowing them to implement sophisticated CFD analysis without the burden of software overhead.

Ultimately, this investigation is driven by the philosophy that advanced engineering insights should be accessible to any student with the requisite curiosity and technical drive.

2 Introduction

The aerodynamic performance of control surfaces is a determining factor in the flight stability and payload capacity of launch vehicles. Among various planform options, the trapezoidal fin is frequently utilized due to its structural rigidity and efficient lift-to-drag characteristics across subsonic and transonic regimes. This study presents a high-fidelity numerical investigation of a 3D trapezoidal fin, defined by a root chord (c_r) of 5.50 in and an aspect ratio (AR) of 0.720, utilizing the OpenFOAM v2412 open-source CFD code to characterize its fundamental aerodynamic quantities.

2.1 Problem Statement and Scope

While analytical methods such as Barrowman's theory provide rapid estimations of center of pressure, they often fail to account for the complex three-dimensional flow structures at the fin tips and the trailing-edge separation zones. To address these limitations, this investigation employs the OpenFOAM v2412 framework to solve the Reynolds-Averaged Navier-Stokes (RANS) equations. The primary focus is the extraction and verification of the following basic foundational aerodynamic quantities, defined by their total coefficients and their respective pressure (p) and viscous (v) components.

These foundational quantities are derived from the integration of the normal pressure (P) and tangential shear (τ_w) stresses acting across the fin surface [2]. Specifically, the drag force (D) is defined as the component of the resultant force acting in the direction of the free-stream flow, representing the total aerodynamic resistance. Conversely, the lift force (L) is the component acting normal (perpendicular) to the flow direction, providing the stabilizing or control force for the vehicle. The aerodynamic moments account for the rotational torques generated about the reference axes: pitch (M_y) represents the rotation about the lateral axis affecting the angle of attack, roll (M_x) refers to the rotation about the longitudinal axis of the vehicle, and yaw (M_z) characterizes the side-to-side rotation about the vertical axis. Mathematically, the resultant forces are expressed as:

$$D = \int_A (P\cos\theta + \tau_w \sin\theta)\, dA, \quad L = \int_A (-P\sin\theta + \tau_w \cos\theta)\, dA \tag{1}$$

where θ is the angle between the surface normal and the flow direction [2]. The aerodynamic moments are similarly derived by integrating the moment arm—the distance from the reference center (x_{ref}, y_{ref}) to the surface element multiplied by the local surface forces. For the pitching moment (M), the integral is expressed as:

$$M = \int_A [(x - x_{ref})dL + (y - y_{ref})dD] \tag{2}$$

where (x, y) represents the coordinates of the surface element dA.

While these "bare" forces describe the physical loads, they are nondimensionalized into the following coefficients to allow for comparative analysis across different flight regimes and scales. For all calculations, the dynamic pressure is defined as $q_\infty = \frac{1}{2}\rho_\infty U_\infty^2$. In this investigation, the freestream velocity is derived from the target Mach number ($M = 0.7$) and the local speed of sound (a), which is determined by the relation $a = \sqrt{\gamma R T}$. Here, γ represents the ratio of specific heats for air (1.4), and R is the specific gas constant (287.05 J/kg·K). To reflect the operational environment at an altitude of 2,000 ft, standard atmospheric values are employed ($T = 284.19$ K, $\rho_\infty = 1.154$

kg/m^3)[3]. Under these conditions, the speed of sound is calculated as 338.0 m/s, yielding a freestream velocity $U_\infty = 236.6$ m/s.

Consequently, the simulation employs a constant dynamic pressure $q_\infty = 32,298$ Pa. To non-dimensionalize the resulting forces, the reference planform area S and the Mean Aerodynamic Chord c_{MAC} are utilized as the characteristic scales, ensuring the results are consistent with standard aerodynamic coefficients.

- **Drag Coefficient (C_D):** Quantifying the resistance force parallel to the free-stream flow, including the effects of the constant thickness ($t = 0.125$ in) profile.

$$C_D = \frac{D}{q_\infty S} \tag{3}$$

  The drag is analyzed through its pressure (form) drag and viscous (skin friction) drag components:

$$C_{D,p} = \frac{D_p}{q_\infty S}, \quad C_{D,v} = \frac{D_v}{q_\infty S} \tag{4}$$

- **Lift Coefficient (C_L):** Characterizing the force perpendicular to the free-stream flow.

$$C_L = \frac{L}{q_\infty S} \tag{5}$$

  The total lift is decomposed into the contributions from the surface pressure distribution and skin friction:

$$C_{L,p} = \frac{L_p}{q_\infty S}, \quad C_{L,v} = \frac{L_v}{q_\infty S} \tag{6}$$

- **Moment Coefficient (C_M):** Evaluating the pitching moment relative to the reference length c_{MAC}.

$$C_M = \frac{M}{q_\infty S c_{MAC}} \tag{7}$$

  The total moment is likewise decomposed into pressure and viscous moments:

$$C_{M,p} = \frac{M_p}{q_\infty S c_{MAC}}, \quad C_{M,v} = \frac{M_v}{q_\infty S c_{MAC}} \tag{8}$$

Through this framework, the total performance of the fin is evaluated as the summation of its components ($C_i = C_{i,p} + C_{i,v}$), providing a comprehensive view of the aerodynamic forces acting on the vehicle.

2.2 Geometric Definition

The study focuses on a three-dimensional trapezoidal fin. The planform is defined by the root chord c_r, the tip chord c_t, and the semi-span b. A constant thickness t is maintained across the span, effectively creating a modified flat-plate profile with a specified thickness-to-chord ratio. The primary geometric parameters are summarized in Table 1.

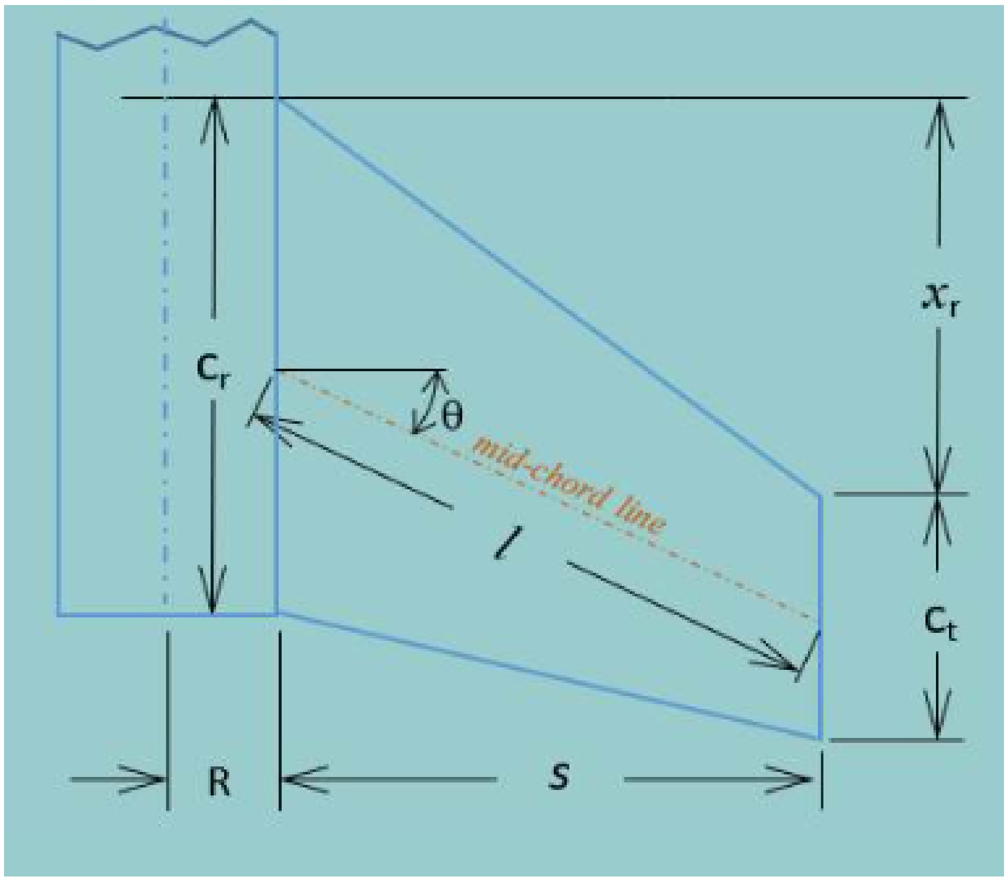

Figure 2: Schematic diagram of the trapezoidal fin planform illustrating the root chord (c_r), tip chord (c_t), semi-span (b), and constant thickness (t).

Table 1: Fin Geometric Parameters

Parameter	Symbol	Value
Root Chord	c_r	5.500 in
Tip Chord	c_t	2.144 in
Semi-Span	b	2.750 in
Root Leading Edge Offset	x_r	0.915 in
Constant Thickness	t	0.125 in
Taper Ratio	$\lambda = c_t/c_r$	0.389
Planform Area	S	10.51 in^2
Aspect Ratio	AR	0.720

For the calculation of the Reynolds number (Re) and the non-dimensionalization of aerodynamic coefficients (C_L, C_D, C_M), the Mean Aerodynamic Chord (c_{MAC}) is selected as the characteristic length scale. For a trapezoidal planform, c_{MAC} is calculated geometrically as:

$$c_{MAC} = \frac{2}{3}\left(c_r + c_t - \frac{c_r c_t}{c_r + c_t}\right) \tag{9}$$

Substituting the defined dimensions:

$$c_{MAC} = \frac{2}{3}\left(5.50 + 2.144 - \frac{5.50 \times 2.144}{5.50 + 2.144}\right) \approx 4.067 \text{ in} \tag{10}$$

This value serves as the reference length (L_{ref}) for the solver. The reference area (A_{ref}) is the planform area S. These scales are critical for ensuring that the turbulence model correctly predicts the boundary layer development and the separation characteristics at the trailing edge.

Blunt Fin This geometry is characterized by a rectangular cross-section featuring flat leading and trailing edges. While the planform dimensions remain as described and consistent to the alternative design, the blunt ends create sharp discontinuities in the profile. In this configuration, the flow must negotiate $90°$ corners, which typically results in immediate stagnation at the leading face and a large, turbulent wake downstream of the trailing edge. This design serves to highlight the impact of non-aerodynamic edge treatments on the total pressure drag. [2]

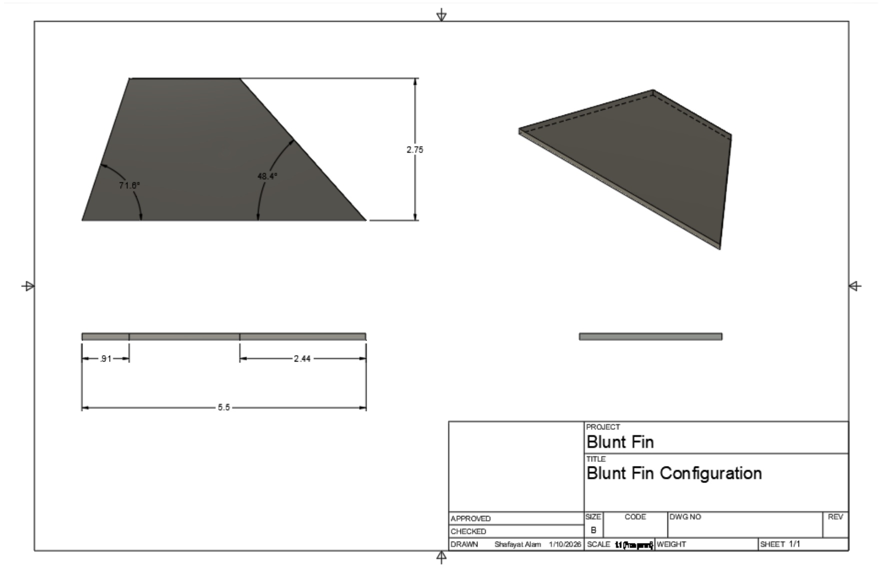

Figure 3: Schematic of the Blunt Fin Configuration

Aero Fin This geometry maintains the exact same planform dimensions as the blunt configuration but implements a curved leading edge and a cusped trailing edge. The leading-edge curvature is designed to allow the flow to accelerate around the profile with minimal separation at Mach 0.7, while the tapered trailing edge is intended to facilitate a gradual pressure recovery. By keeping the global dimensions constant, this configuration allows for a direct assessment of how airfoil profiling influences the pressure (p) and viscous (v) force components. [2]

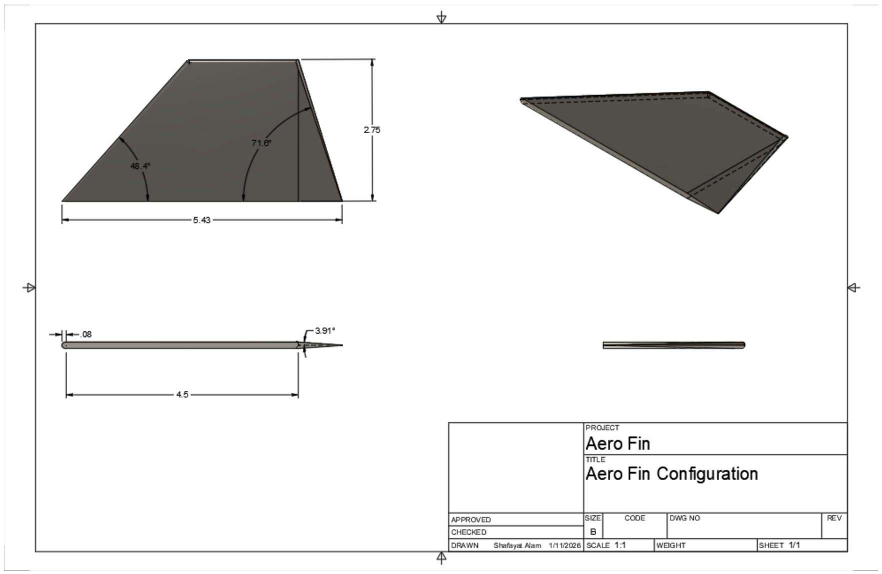

Figure 4: Schematic of the Aero Fin Configuration

3 Physical/Mathematical Model

The aerodynamic analysis of a trapezoidal fin is rooted in the fundamental laws of continuum mechanics. Before introducing numerical discretization or specific gas models, it is essential to establish the governing equations that describe the motion of a fluid. These equations, collectively known as the Navier-Stokes equations, represent the application of classical conservation laws to a fluid medium.

Mathematically, these are highly non-linear second-order partial differential equations (PDEs). While they are generally of a mixed type, they share properties with the three classic categories of second-order equations: elliptic, parabolic, and hyperbolic. In this study, the subsonic pressure field is primarily elliptic, where information travels in all directions; the boundary layer development is parabolic, governed by upstream conditions; and any transient fluctuations or potential local supersonic pockets exhibit hyperbolic behavior where information travels along characteristic lines [1]. This classification is fundamentally dictated by the Mach number (M), which represents the ratio of local flow velocity (u) to the local speed of sound (a):

$$M = \frac{u}{a} = \frac{u}{\sqrt{\gamma R T}} \tag{11}$$

Table 2: Classification of the main categories of fluid flow [BOOK REFERENCE REQUIRED]

	Steady flow	**Unsteady flow**
Viscous flow	Elliptic	Parabolic
Inviscid flow	$M < 1$, elliptic $M > 1$, hyperbolic	Hyperbolic
Thin shear layers	Parabolic	Parabolic

For the current mission profile at approximately $M = 0.7$, the condition $M < 1$ ensures that pressure disturbances (information) propagate faster than the bulk fluid motion. This reinforces the elliptic nature of the domain, confirming that every point in the solution field is influenced by its neighbors.

The use of a steady-state solver is justified in this analysis because the subsonic flow regime is characterized by an equilibrium state governed primarily by these elliptic properties. In such equilibrium problems, disturbances in the interior of the solution—such as the pressure variations around the fin geometry—travel in all directions, allowing every point in the domain to be influenced by its neighbors. This creates a smooth solution field where information propagates throughout the interior, a hallmark of boundary-value problems that can be accurately resolved using steady-state numerical techniques.

The transition from the Lagrangian perspective (tracking individual fluid particles) to the Eulerian perspective (observing a fixed control volume in space) is achieved via the Reynolds Transport Theorem (RTT). For any extensive property B, the rate of change for a system is related to the control volume CV and its control surface CS by:

$$\left.\frac{dB}{dt}\right|_{sys} = \frac{\partial}{\partial t}\iiint_{CV} \rho\beta dV + \iint_{CS} \rho\beta(\mathbf{u}\cdot\mathbf{n})dA \tag{12}$$

where β is the intensive property (quantity per unit mass), ρ is the density, and $\mathbf{u}$ is the velocity vector. This theorem serves as the mathematical engine that transforms Newton's laws and the laws of thermodynamics into the partial differential equations of fluid flow. To arrive at the local differential form, the surface integral in Eq. (9), representing the net flux across the control surface (CS), is converted into a volume integral over the control volume (CV) via the Divergence Theorem (Gauss's Theorem):

$$\iint_{CS} \rho\beta(\mathbf{u}\cdot\mathbf{n})dA = \iiint_{CV} \nabla\cdot(\rho\beta\mathbf{u})dV \tag{13}$$

By substituting this into the Reynolds Transport Theorem and assuming a fixed control volume, the general conservation law is expressed in a unified volume integral form:

$$\iiint_{CV} \left[\frac{\partial(\rho\beta)}{\partial t} + \nabla\cdot(\rho\beta\mathbf{u})\right] dV = \dot{B}_{net} \tag{14}$$

where $\dot{B}_{net}$ represents the net rate of change of the property B due to external sources or forces. Since the chosen control volume is arbitrary, the integrand itself must vanish to satisfy the equality at every point in space. This provides the basis for the steady-state, compressible governing equations used in this study, where the transient term ($\partial/\partial t$) is neglected to focus on equilibrium aerodynamic sensitivity.

3.1 Fluid Flow Governing Equations

Applying the RTT to the properties of mass, momentum, and energy yields the set of coupled non-linear equations that govern all fluid phenomena. Following the continuum assumption, we focus on macroscopic behavior where fluid properties are defined at every point in space.

3.1.1 Conservation of Mass (Continuity)

Substituting $B = m$ and $\beta = 1$ into the RTT, and noting that mass is conserved ($dm/dt = 0$), we obtain the continuity equation in its general form:

$$\underbrace{\frac{\partial\rho}{\partial t}}_{\text{Unsteady Density (Temporal)}} + \underbrace{\nabla\cdot(\rho\mathbf{u})}_{\text{Net Mass Flux (Convection)}} = 0 \tag{15}$$

This equation states that the rate of increase of mass within a control volume must equal the net rate of mass flux into the volume.

3.1.2 Conservation of Momentum

Substituting $B = m\mathbf{u}$ and $\beta = \mathbf{u}$ (Newton's Second Law), and accounting for surface forces (pressure and friction) and body forces (gravity), we arrive at the momentum equation:

$$\underbrace{\frac{\partial(\rho\mathbf{u})}{\partial t}}_{\text{Unsteady Acceleration (Temporal)}} + \underbrace{\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u})}_{\text{Convective Acceleration}} = \underbrace{-\nabla p}_{\text{Pressure Forces}} + \underbrace{\nabla\cdot\tau}_{\text{Viscous Forces}} + \underbrace{\rho\mathbf{f}_b}_{\text{Body Forces}} \tag{16}$$

Here, the term $\rho\mathbf{u}\otimes\mathbf{u}$ represents the convective transport of momentum (the advection term), which is the source of non-linearity in the flow. The viscous stress tensor τ represents the internal resistance of the fluid to deformation, characterized by a linear (Newtonian) relationship between shear stress and shear rate.

3.1.3 Conservation of Energy

The First Law of Thermodynamics is applied by setting $B = E$ (total energy) and $\beta = E = e + K$, where e is the specific internal energy and $K = \frac{1}{2}|\mathbf{u}|^2$ is the specific kinetic energy. The general conservation law for total energy in a control volume is expressed as:

$$\underbrace{\frac{\partial(\rho E)}{\partial t}}_{\text{Temporal Change}} + \underbrace{\nabla \cdot (\rho \mathbf{u} E)}_{\text{Convective Transport}} = \underbrace{\nabla \cdot (k \nabla T)}_{\text{Conduction}} + \underbrace{\nabla \cdot (\tau \cdot \mathbf{u})}_{\text{Viscous Work}} - \underbrace{\nabla \cdot (p\mathbf{u})}_{\text{Pressure Work}} + \underbrace{S_E}_{\text{Additional Source Term}} \tag{17}$$

In this form, the term $\nabla \cdot (p\mathbf{u})$ represents the rate of work done by pressure forces, which is fundamental for characterizing compressible flows where the fluid performs work through expansion or compression.

3.1.4 Model Assumptions

While the generalized conservation laws provide a complete mathematical description of fluid motion, the specific requirements of this aerodynamic study allow for the reduction of these equations into a steady-state ($\partial/\partial t = 0$), compressible, and adiabatic form. This simplified framework is strategically chosen to support the ultimate goal of the model: shape optimization (e.g. through global evolutionary/genetic algorithms and local adjoint methods) and geometry/aerodynamics sensitivity analysis (e.g. through local adjoint and global variance-based methods).

- **Steady-State:** The primary focus of this study is the evaluation of mean aerodynamic coefficients (C_L, C_D, C_M) and their sensitivity to geometric variations. A steady-state approach captures the time-averaged pressure distribution over the trapezoidal fin with high fidelity while maintaining the computational efficiency necessary for the iterative cycles inherent in shape optimization and sensitivity workflows.

- **Viscous Flow** While inviscid models (such as Euler equations) are computationally cheaper, they neglect the effects of fluid friction and boundary layer development, which are critical for predicting the aerodynamic performance of a fin. In a real flow at $M = 0.7$, the viscous effects lead to the formation of a boundary layer, flow separation near the trailing edge, and the generation of a wake. By solving the for viscosity rather than assuming ideal flow, the simulation can accurately capture the skin friction drag and the pressure distribution influenced by the boundary layer thickness. In high-subsonic flows, the temperature gradients within the boundary layer and near stagnation points become significant. Because molecular viscosity is physically dependent on temperature, assuming a constant viscosity would introduce errors in the calculation of the Reynolds number and the shear stresses. To maintain physical fidelity, the molecular viscosity μ is calculated using Sutherland's Law:

$$\mu = \mu_0 \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S}{T + S} \tag{18}$$

  where T is the local temperature, μ_0 is the reference viscosity at temperature T_0, and S is the Sutherland constant. This model is essential for coupling the thermal solution from the Energy equation back to the Momentum equations, ensuring that the drag and turbulence production terms are based on the actual local state of the fluid.

- **Compressibility:** In the high-subsonic flight regime, the interaction between the fluid and the fin geometry induces significant density variations ($\Delta\rho$), particularly at leading-edge stagnation points and regions of rapid acceleration. While flows at low Mach numbers ($M < 0.3$) are often approximated as incompressible[1], the current mission profile at $M = 0.7$ requires a fully compressible formulation. Modeling the flow as compressible via the Equation of State is mandatory to accurately resolve the pressure-density coupling:

$$\rho = f(p, T) \tag{19}$$

  This ensures that the sensitivity of the pressure field to shape changes is physically representative and captures the "thermal spring" effect of the air.

- **Adiabatic Boundary Condition:** An adiabatic wall boundary condition ($q_w = 0$) is implemented to decouple the complex physics of conjugate heat transfer from the aerodynamic performance metrics. This is a valid approximation for the short-duration flight profiles typical of rocket missions, where the thermal soak of the structure is secondary to the immediate pressure-driven forces. This decoupling allows the solver to focus computational resources on resolving the boundary layer and shock-boundary layer interactions critical for geometry based aerodynamic insight.

3.1.5 Reduced Governing Equations

Conservation of Mass (Continuity) For steady-state compressible flow, the rate of mass flux into the control volume must be perfectly balanced, reducing Equation (10) to:

$$\underbrace{\nabla \cdot (\rho \mathbf{u})}_{\text{Mass Flux (Convection)}} = 0 \tag{20}$$

Conservation of Momentum The momentum equation is reduced by neglecting temporal variations and body forces (as gravitational effects are negligible for high-speed aerodynamics of geometry that is lightweight), resulting in:

$$\underbrace{\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u})}_{\text{Convective Acceleration}} = \underbrace{-\nabla p}_{\text{Pressure Forces}} + \underbrace{\nabla \cdot \tau}_{\text{Viscous Forces}} \tag{21}$$

Conservation of Energy By applying the same steady-state assumption ($\partial/\partial t = 0$) used in the previous conservation laws and neglecting external heat sources ($S_E = 0$), the general energy equation (12) is reduced to a balance of convective transport and surface work:

$$\underbrace{\nabla \cdot (\rho \mathbf{u} E)}_{\text{Total Energy Convection}} = \underbrace{\nabla \cdot (k \nabla T)}_{\text{Conduction}} + \underbrace{\nabla \cdot (\tau \cdot \mathbf{u})}_{\text{Viscous Work}} - \underbrace{\nabla \cdot (p \mathbf{u})}_{\text{Pressure Work}} \tag{22}$$

This reduced form describes the equilibrium of energy transport within the control volume, where the advection of total specific energy is balanced by thermal conduction and the mechanical work performed by viscous and pressure forces. Defined as $E = e + K$, this is the most fundamental conservation form. It is typically chosen for general-purpose solvers as it tracks the complete energy state without decoupling mechanical and thermal energy components.

However, the energy equation can be mathematically cast into several alternative forms. By substituting different thermodynamic variables into the reduced energy balance (Eq. 15), the governing equation is optimized for specific physical regimes and numerical stability:

- **Specific Internal Energy ($\hat{e}$):**

$$\underbrace{\nabla \cdot (\rho \mathbf{u} \hat{e})}_{\text{Internal Energy Transport}} = \underbrace{\nabla \cdot (k \nabla T)}_{\text{Conduction}} - \underbrace{p(\nabla \cdot \mathbf{u})}_{\text{Expansion Work}} + \underbrace{\tau : \nabla \mathbf{u}}_{\text{Viscous Dissipation}} \tag{23}$$

  Derived by subtracting the mechanical energy equation from the total energy equation. This form is ideal for tracking the conversion between viscous work and internal heat, making it advantageous for thermal-heavy analysis where bulk kinetic energy transitions are secondary.

- **Total Enthalpy (h_0):**

$$\underbrace{\nabla \cdot (\rho \mathbf{u} h_0)}_{\text{Enthalpy Transport}} = \underbrace{\nabla \cdot (\alpha_{eff} \nabla h_0)}_{\text{Enthalpy Diffusion (Effective)}} \tag{24}$$

  Defined as $h_0 = \hat{h} + \frac{1}{2}|\mathbf{u}|^2$, enthalpy forms are frequently preferred in high-speed and compressible aerodynamics. Total enthalpy simplifies boundary conditions for stagnation points and remains nearly constant along streamlines in adiabatic flows, which significantly enhances numerical stability.

- **Temperature (T):**

$$\underbrace{\rho c_p (\mathbf{u} \cdot \nabla T)}_{\text{Sensible Heat Convection}} = \underbrace{\nabla \cdot (k \nabla T)}_{\text{Conduction}} + \underbrace{\Phi}_{\text{Viscous Dissipation}} + \underbrace{\dot{q}_v}_{\text{Volumetric Source}} \tag{25}$$

  By imposing caloric relations (e.g., $d\hat{h} = c_p dT$), the equation can be solved directly for temperature. This is the standard choice for incompressible solvers or constant-pressure systems where temperature is the primary variable of interest for experimental validation.

In the present study, the Total Enthalpy (h_0) form, as defined and expressed in Eq. (18), is selected as the primary energy variable. For steady-state, high-subsonic aerodynamics, solving for h_0 is preferred because it significantly simplifies numerical boundary conditions at stagnation points—such as the leading edge of the trapezoidal fin—where kinetic energy is converted into static enthalpy. Furthermore, in adiabatic flows, total enthalpy remains nearly constant along streamlines, which enhances the numerical stability of the solver and provides a robust framework for capturing the compressibility effects inherent in high-velocity flight regimes. By neglecting temporal variations and external heat sources, the transport of total enthalpy implemented in this study is governed by the reduced balance between advection and effective thermal diffusion, where α_{eff} accounts for both molecular and turbulent transport effects.

3.1.6 Thermodynamic Closure and Equation of State

The Navier-Stokes system contains more unknowns (ρ, u, v, w, p, h_0) than available conservation equations. The Equation of State (EOS) provides the critical ”closure” by relating the thermodynamic variables and transport properties through three primary functions:

1. **Thermal Closure:** Relates density to pressure and temperature. Consistent with the *perfectGas* model, this is defined as:
$$\rho = \frac{p}{RT} \tag{26}$$
where R is the specific gas constant for air (287.05 J/(kg·K)).

2. **Caloric Closure:** Relates the enthalpy solved in the energy equation to the local static temperature. For a calorically perfect gas, the static enthalpy (h) is a linear function of temperature:
$$h = C_p T \tag{27}$$
Given that the solver computes the total enthalpy (h_0), the static temperature is extracted by subtracting the local kinetic energy: $T = (h_0 - \frac{1}{2}|\mathbf{u}|^2)/C_p$.

3. **Transport Coupling (Prandtl Number):** The Prandtl number (Pr) is a dimensionless parameter representing the ratio of momentum diffusivity (kinematic viscosity) to thermal diffusivity. Mathematically:
$$Pr = \frac{\nu}{\alpha} = \frac{C_p \mu}{k} \tag{28}$$
In the present study, a value of $Pr = 0.72$ is utilized (consistent with air at standard conditions). This value is critical for the energy equation closure, as it allows for the calculation of the effective thermal diffusivity α_{eff} from the dynamic viscosity μ, thereby coupling the thickness of the thermal boundary layer to the momentum boundary layer.

Modeling with the Perfect Gas Law in this study ensures that any change in the velocity field that alters the pressure distribution (∇p) immediately updates the density and temperature. This ensures the model correctly accounts for the ”work” done by the fluid as it accelerates over the fin geometry.

Modeling with the Perfect Gas Law In this study, the use of the Perfect Gas EOS couples the momentum and energy equations; any change in the velocity field that alters the pressure distribution (∇p) immediately updates the density and temperature. This ensures that the energy equation correctly accounts for the ”work” done by the fluid as it accelerates over the fin planform.

3.2 Turbulence Modeling and Statistical Closure

The flow over a rocket fin at high subsonic speeds is inherently turbulent. As discussed in the previous section, the non-linearity of the Navier-Stokes equations leads to a chaotic state characterized by three-dimensional eddy structures spanning a vast range of spatial and temporal scales. To capture the influence of these eddies on the aerodynamic coefficients (C_L, C_D, C_M), a mathematical strategy must be established for the modeling of turbulence.

3.2.1 Classification of Flow Regimes

For external flow over a fin, the flow field is categorized into three distinct regimes:

1. **Laminar Regime ($Re_x < 5 \times 10^5$):** Characterized by smooth, parallel layers. The equations here are primarily **parabolic** as information marches downstream within the boundary layer.
2. **Transition Regime ($5 \times 10^5 \leq Re_x \leq 3 \times 10^6$):** A stochastic region where inertial terms dominate and Tollmien-Schlichting waves amplify into turbulent spots.
3. **Fully Turbulent Regime ($Re_x > 3 \times 10^6$):** Characterized by chaotic, multi-scale fluctuations. These flows are often **elliptic** in recirculating zones (like the wake) where information travels both upstream and downstream, affecting the entire domain.

3.2.2 Significance for the Current Model

Given the mission profile, the global Reynolds number is to exceed $Re_{crit} \approx 5 \times 10^5$. Consequently, the boundary layer is expected to be predominantly turbulent. This mixed-type mathematical nature necessitates an iterative numerical scheme (such as the SIMPLE algorithm) that can resolve the elliptic pressure-velocity coupling while accurately ”marching” through the parabolic boundary layer properties.

3.2.3 A Selection of Turbulence Modeling Approaches

In computational fluid dynamics, there are three primary approaches to handling the turbulent scales, differing in their treatment of the Navier-Stokes equations:

1. **Direct Numerical Simulation (DNS):** Resolves all scales of motion from the domain size down to the Kolmogorov microscale. While physically exact, the computational cost scales as Re^3, making it extremely expensive for applications like a fin with $c_{MAC} \approx 4$ in.
2. **Large Eddy Simulation (LES):** Employs a spatial filtering technique to resolve large energy-containing eddies while modeling the smaller, universal sub-grid scales. This is computationally intensive and typically reserved for transient aeroacoustic studies.
3. **Reynolds-Averaged Navier-Stokes (RANS):** The approach utilized in this study. RANS does not attempt to resolve any turbulent fluctuations. Instead, it solves for the time-averaged (or ensemble-averaged) flow field, modeling the effect of all turbulent scales through a ”turbulence model.”

   The selection of the RANS approach is strategically aligned with the objectives of geometry optimization and sensitivity analysis. By solving for the time-averaged flow field, RANS provides a computationally efficient pathway to obtain stable aerodynamic coefficients (C_L, C_D, C_M). While DNS and LES offer higher fidelity for transient structures, the RANS framework effectively captures the mean pressure distributions and skin friction effects required to evaluate how small changes in fin planform impact the overall vehicle stability.

3.2.4 The RANS Framework and Reynolds Decomposition

The RANS approach begins with the Reynolds Decomposition, where every instantaneous flow variable ϕ (velocity, pressure, etc.) is split into a mean component $\bar{\phi}$ and a fluctuating component ϕ':

$$\phi(x,t) = \bar{\phi}(x) + \phi'(x,t) \tag{29}$$

The mean component $\bar{\phi}$ is defined as the time-average of the instantaneous quantity over a period T that is large relative to the time scale of the turbulent fluctuations:

$$\bar{\phi}(\mathbf{x}) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \phi(\mathbf{x},t)dt \tag{30}$$

By definition, the Reynolds averaging process follows a set of mathematical identities that are critical for simplifying the governing equations. For any two variables ϕ and ψ and a constant c, the following rules apply:

1. The mean of a fluctuating component is zero: $\overline{\phi'} = 0$.
2. The mean of a mean is the mean itself: $\overline{\bar{\phi}} = \bar{\phi}$.
3. The mean of a sum is the sum of the means: $\overline{\phi + \psi} = \bar{\phi} + \bar{\psi}$.
4. The mean of the product of a mean and a fluctuation vanishes: $\overline{\bar{\phi}\psi'} = 0$.
5. The mean of the product of two fluctuations does not necessarily vanish: $\overline{\phi'\psi'} \neq 0$. This term represents the covariance (correlation) between the turbulent fluctuations of two variables.

Applying the Reynolds decomposition ($\rho = \bar{\rho} + \rho'$ and $\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}'$) and taking the ensemble average of the continuity equation yields:

$$\overline{\nabla \cdot [(\bar{\rho} + \rho')(\bar{\mathbf{u}} + \mathbf{u}')]} = 0 \tag{31}$$

Expanding the product and applying the averaging rules where $\overline{\rho'} = 0$ and $\overline{\mathbf{u}'} = 0$, the expression simplifies to:

$$\nabla \cdot (\bar{\rho}\bar{\mathbf{u}}) + \nabla \cdot (\overline{\rho'\mathbf{u}'}) = 0 \tag{32}$$

Under the assumption of moderate compressibility for high-subsonic flows ($M \approx 0.7$), the turbulent mass flux term $\overline{\rho'\mathbf{u}'}$ is considered negligible Morkovin's Hypothesis. This results in the final RANS continuity formulation used in this study:

$$\nabla \cdot (\bar{\rho}\bar{\mathbf{u}}) = 0 \tag{33}$$

The averaging process is most clearly demonstrated by expanding the x-component of the momentum equation as an example. By substituting the decomposed variables $\phi = \bar{\phi} + \phi'$ for density, velocity, and pressure, the instantaneous momentum balance is expressed as:

$$\begin{gathered}\overline{\frac{\partial}{\partial x}[(\bar{\rho} + \rho')(\bar{u} + u')(\bar{u} + u')]} + \overline{\frac{\partial}{\partial y}[(\bar{\rho} + \rho')(\bar{v} + v')(\bar{u} + u')]} \\ + \overline{\frac{\partial}{\partial z}[(\bar{\rho} + \rho')(\bar{w} + w')(\bar{u} + u')]} = \overline{-\frac{\partial(\bar{p} + p')}{\partial x} + \frac{\partial(\bar{\tau}_{xx} + \tau'_{xx})}{\partial x} + \frac{\partial(\bar{\tau}_{yx} + \tau'_{yx})}{\partial y} + \frac{\partial(\bar{\tau}_{zx} + \tau'_{zx})}{\partial z}}\end{gathered} \tag{34}$$

By applying the averaging rules discussed earlier, the expanded momentum equation simplifies the reduced momentum equations expressed in differential form as:

$$\bar{\rho}(\bar{\mathbf{u}} \cdot \nabla\bar{\mathbf{u}}) = -\nabla\bar{p} + \nabla \cdot \bar{\tau} \underbrace{-\nabla \cdot (\bar{\rho}\overline{\mathbf{u}' \otimes \mathbf{u}'})}_{\text{Reynolds Stresses}} \tag{35}$$

In this reduced form, the left-hand side represents the convective transport of the mean momentum. The right-hand side accounts for the mean pressure gradient, the molecular viscous stresses, and the apparent stresses arising from turbulent fluctuations, specifically the Reynolds Stress tensor ($\bar{\rho}\overline{\mathbf{u}' \otimes \mathbf{u}'}$).

The energy equation follows an identical averaging process. By substituting the decomposed variables into the steady-state total enthalpy equation, we obtain:

$$\overline{\nabla \cdot [(\bar{\rho} + \rho')(\bar{\mathbf{u}} + \mathbf{u}')(\bar{h}_0 + h'_0)]} = \overline{\nabla \cdot [(\bar{\alpha} + \alpha')\nabla(\bar{h}_0 + h'_0)]} \tag{36}$$

Applying the Reynolds averaging rules and eliminating terms that vanish ($\overline{\bar{\phi}\psi'} = 0$), the reduced energy equation in differential form is expressed as:

$$\nabla \cdot (\bar{\rho}\bar{\mathbf{u}}\bar{h}_0) = \nabla \cdot (\bar{\alpha}\nabla\bar{h}_0) \underbrace{-\nabla \cdot (\bar{\rho}\overline{\mathbf{u}'h'_0})}_{\text{Turbulent Heat Flux}} \tag{37}$$

Similar to the Reynolds stress in the momentum equation, the term $-\bar{\rho}\overline{\mathbf{u}'h'_0}$ represents the Turbulent Heat Flux, which accounts for the transport of energy due to turbulent eddies and requires a closure model.

3.2.5 Selection of Turbulence Closure Models

The Boussinesq Hypothesis provides the mathematical "bridge" to close the RANS equations by relating the chaotic Reynolds stresses to the mean flow gradients. It assumes that the effect of turbulence can be modeled as an increased viscosity, known as the turbulent eddy viscosity (μ_t).

For the momentum equation, the Reynolds stress tensor is modeled as:

$$\underbrace{-\bar{\rho}\overline{\mathbf{u}' \otimes \mathbf{u}'}}_{\text{Chaotic Stress}} = \mu_t \left[\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T\right] - \frac{2}{3}\bar{\rho}k\mathbf{I} \tag{38}$$

Similarly, the turbulent heat flux in the energy equation is modeled using a turbulent thermal diffusivity (k_t):

$$\underbrace{-\bar{\rho}\overline{\mathbf{u}' h_0'}}_{\text{Chaotic Heat Flux}} = k_t \nabla \bar{T} \tag{39}$$

Unlike molecular viscosity (μ), which is a fluid-specific constant (at a given temperature), μ_t is a *local property of the flow field.* To determine its distribution without manual tuning, turbulence closure models are employed. These models range from two-equation formulations, including the standard k-ϵ, k-ω, and the hybrid k-ω SST, to one-equation transport models such as Spalart-Allmaras, all of which solve additional partial differential equations to compute μ_t dynamically.

The k-ϵ Model The k-ϵ model closes the RANS equations by solving two additional transport equations for the turbulent kinetic energy (k) and the dissipation rate (ϵ). The eddy viscosity is calculated as a function of these two variables:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{40}$$

The transport equations for k and ϵ are expressed as follows:

$$\underbrace{\frac{\partial(\rho k)}{\partial t}}_{\text{Transient}} + \underbrace{\nabla \cdot (\rho k \bar{\mathbf{u}})}_{\text{Convection}} = \underbrace{\nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right]}_{\text{Diffusion}} + \underbrace{P_k}_{\text{Production}} - \underbrace{\rho\epsilon}_{\text{Dissipation}} \tag{41}$$

$$\underbrace{\frac{\partial(\rho \epsilon)}{\partial t}}_{\text{Transient}} + \underbrace{\nabla \cdot (\rho \epsilon \bar{\mathbf{u}})}_{\text{Convection}} = \underbrace{\nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\nabla \epsilon\right]}_{\text{Diffusion}} + \underbrace{C_{1\epsilon}\frac{\epsilon}{k}P_k}_{\text{Generation}} - \underbrace{C_{2\epsilon}\rho\frac{\epsilon^2}{k}}_{\text{Destruction}} \tag{42}$$

where P_k represents the production of turbulent kinetic energy due to mean velocity gradients, and C_μ, $C_{1\epsilon}$, $C_{2\epsilon}$, σ_k, and σ_ϵ are semi-empirical constants.

The k-ϵ model is highly robust and computationally stable, making it the preferred choice for free-shear flows and far-field turbulence far from solid boundaries. However, it is fundamentally limited in the near-wall region. It struggles to accurately predict flow behavior in the presence of adverse pressure gradients and often fails to capture boundary layer separation accurately—a critical factor for aerodynamic surfaces like fins.

The k-ω Model The k-ω model utilizes the specific dissipation rate (ω) as the second scale-determining variable. In this formulation, the turbulent eddy viscosity is defined as:

$$\mu_t = \rho \frac{k}{\omega} \tag{43}$$

The transport equations for the turbulent kinetic energy (k) and the specific dissipation rate (ω) are:

$$\underbrace{\frac{\partial(\rho k)}{\partial t}}_{\text{Transient}} + \underbrace{\nabla \cdot (\rho k \bar{\mathbf{u}})}_{\text{Convection}} = \underbrace{\nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right]}_{\text{Diffusion}} + \underbrace{P_k}_{\text{Production}} - \underbrace{\beta^* \rho k \omega}_{\text{Dissipation}} \tag{44}$$

$$\underbrace{\frac{\partial(\rho\omega)}{\partial t}}_{\text{Transient}} + \underbrace{\nabla \cdot (\rho\omega\bar{\mathbf{u}})}_{\text{Convection}} = \underbrace{\nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\nabla\omega\right]}_{\text{Diffusion}} + \underbrace{\alpha\frac{\omega}{k}P_k}_{\text{Generation}} - \underbrace{\beta\rho\omega^2}_{\text{Destruction}} \tag{45}$$

where α, β, β^*, σ_k, and σ_ω are the model constants.

The k-ω model is superior for internal flows and boundary layers, particularly under adverse pressure gradients where it predicts separation much more accurately than k-ϵ. However, its major drawback is ”freestream sensitivity”: the results can change significantly based on the arbitrary values of ω assigned at the far-field boundaries.

The k-ω SST model is a hybrid formulation designed to combine the individual strengths of the k-ϵ and k-ω models while mitigating their respective weaknesses. It utilizes a blending function that allows the solver to transition between models based on the distance from the wall: it employs the k-ω formulation in the inner parts of the boundary layer to maintain accuracy near the surface, and switches to a k-ϵ behavior in the freestream. By doing so, it effectively eliminates the freestream sensitivity issues inherent in the standard k-ω model while retaining its superior ability to predict flow separation under adverse pressure gradients.

The Spalart-Allmaras Model The Spalart-Allmaras model solves a single transport equation for a modified viscosity variable, $\tilde{\nu}$. The actual kinematic eddy viscosity ($\nu_t = \mu_t/\rho$) is then derived using a damping function f_{v1}:

$$\nu_t = \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu} \tag{46}$$

The transport equation for the working variable $\tilde{\nu}$ is expressed as:

$$\underbrace{\frac{\partial(\rho\tilde{\nu})}{\partial t}}_{\text{Transient}} + \underbrace{\nabla \cdot (\rho\tilde{\nu}\bar{\mathbf{u}})}_{\text{Convection}} = \underbrace{\frac{1}{\sigma_{\tilde{\nu}}}\nabla \cdot [(\mu + \rho\tilde{\nu})\nabla\tilde{\nu}] + C_{b2}\rho(\nabla\tilde{\nu})^2}_{\text{Diffusion and Non-conservative terms}} + \underbrace{P_{\tilde{\nu}}}_{\text{Production}} - \underbrace{D_{\tilde{\nu}}}_{\text{Destruction}} \tag{47}$$

where $P_{\tilde{\nu}}$ is the production term and $D_{\tilde{\nu}}$ is the wall-destruction term.

The primary advantage of the Spalart-Allmaras model is its computational efficiency and numerical robustness, especially on the relatively coarse meshes often used in preliminary design phases. It provides excellent results for aerodynamic flows with mild separation, and so thus is employed in this investigation. Nevertheless, it is less accurate for complex internal flows or flows with massive separation and high curvature compared to the k-ω SST model.

3.2.6 Wall Functions and y/y+

[Report wall functions and y+]

3.3 Initial/Boundary Conditions

The Reynolds-Averaged Navier-Stokes (RANS) equations and the Spalart-Allmaras transport model form a system of Partial Differential Equations (PDEs) that describe the conservation laws within the computational domain. However, these equations are general; to obtain a unique solution for the flow over a specific rocket fin geometry, we must prescribe the state of the fluid at the domain boundaries.

In a system of governing equations the boundary conditions serve two critical roles:

1. **Mathematical Closure:** They provide the necessary constraints to solve the system of algebraic equations, transforming a general physical model into a specific simulation of a $M \approx 0.7$ flow field.

2. **Physical Injection:** They represent the physical environment of the investigation, such as the high-velocity air entering the inlet, the atmospheric pressure at the outlet, and the stationary, solid surface of the fin.

Numerical implementation of these conditions involves modifying the flux evaluations at the boundary faces. For any cell P adjacent to a boundary, the neighbor link in the $[A_N]$ matrix is typically replaced or augmented by terms that account for the fixed values or gradients imposed at the edge of the mesh.

Dirichlet Boundary Conditions For a Dirichlet boundary, the value of the transport variable ϕ is explicitly prescribed at the boundary face f. In the context of the rocket fin simulation, this is applied at the inlet to define the free-stream conditions:

$$\phi_f = \phi_{specified} \tag{48}$$

In the discretized system, this known value is moved to the source vector $\{b\}$, and the corresponding neighbor coefficient a_N is set to zero, "breaking" the link to the exterior and forcing the boundary cell to respect the physical constraint. A primary example in this study is the no-slip condition applied at the fin walls. By setting the velocity vector $\mathbf{u}_f = 0$, the fluid is mathematically "pinned" to the surface. This prescription removes the face velocity as an unknown, zeroing the convective flux through the wall and shifting the viscous shear contributions into the source term of the adjacent cell P.

Neumann Boundary Conditions A Neumann boundary condition prescribes the normal gradient of the variable ϕ at the boundary face. This is primarily utilized at the outlet of the computational domain to represent a "zero-gradient" or fully developed flow condition:

$$\left(\frac{\partial \phi}{\partial n}\right)_f = g_{specified} \tag{49}$$

For a $M \approx 0.7$ investigation, we typically set $g = 0$. In the matrix assembly, this implies that the boundary face value is equal to the interior cell value ($\phi_f = \phi_P$). This ensures that the flow properties "exit" the domain smoothly without non-physical reflections or back-pressure. In the algebraic system, the boundary face contribution is eliminated from the neighbor coefficient a_N and the diagonal coefficient a_P is adjusted to maintain the zero-flux constraint, effectively allowing the matrix to solve for the boundary values implicitly based on internal field trends.

Mixed Boundary Conditions A Mixed (or Robin) boundary condition is a linear combination of Dirichlet and Neumann formulations. It prescribes a relationship between the value of the variable and its normal gradient at the face:

$$w\phi_f + (1-w)\left(\frac{\partial \phi}{\partial n}\right)_f = f_{specified} \tag{50}$$

Symmetry Boundary Conditions The symmetry boundary condition enforces a zero-flux constraint across a plane, effectively acting as a "mirror" for the flow field. Mathematically, this is a specialized Neumann condition where the scalar gradient and the normal component of velocity are zero:

$$\frac{\partial \phi}{\partial n} = 0, \quad \mathbf{u} \cdot \mathbf{n} = 0 \tag{51}$$

This condition is applied to the side boundaries of the domain to simulate a semi-infinite fluid environment around the fin without the computational expense of modeling the entire atmosphere. In the algebraic system, the link to the "outside" neighbor is removed ($a_N = 0$), and because the normal velocity is zero, the convective flux across the boundary face is identically zero. This ensures the matrix solution remains perfectly symmetric across the defined plane.

3.3.1 Boundary Conditions Configuration

Initial Conditions The internal field of the computational domain is initialized using a uniform cold start approach. To facilitate numerical stability and accelerate convergence, the *internalField* for all transport variables is set to match the prescribed free-stream conditions at the inlet. Specifically, the domain is initialized at $P_{atm} = 94212$ Pa and $U_\infty = 236.6$ m/s. This ensures that the iterative solver begins with a physically consistent state, reducing the magnitude of the initial residuals in the first iteration.

The selection of boundary values is strictly governed by the physics of compressible flow at a target Mach number of $M = 0.7$. To ensure experimental consistency, the simulation is initialized using International Standard Atmosphere (ISA) sea-level conditions. The thermodynamic state is anchored by an ambient static pressure of $P_\infty = 94212$ Pa

Table 3: Boundary Conditions Configuration including Wall Functions

Variable	**Inlet**	**Outlet**	**Fin**	**Walls**	**Symmetry**
Velocity ($\mathbf{u}$)	Dirichlet (236.6 m/s)	Neumann (zeroGradient)	Dirichlet (noSlip)	slip (zeroGradient)	Symmetry
Pressure (p)	Neumann (zeroGradient)	Dirichlet (94212 Pa)	Neumann (zeroGradient)	Neumann (zeroGradient)	Symmetry
Temperature (T)	Dirichlet (283.19 K)	Neumann (zeroGradient)	Neumann (Adiabatic)	Neumann (zeroGradient)	Symmetry
S-A Variable ($\tilde{\nu}$)	Dirichlet (4.695×10^{-5})	Neumann (zeroGradient)	Dirichlet ($0\ m^2/s$)	Neumann (zeroGradient)	Symmetry
Turb. Visc. (ν_t)	Dirichlet (1.565×10^{-6} m^2/s)	Dirichlet (1.565×10^{-6} m^2/s)	Wall Function (alphatJayatillekeWallFunction)	Neumann (zeroGradient)	Symmetry
Therm. Diff. (α_t)	Dirichlet (1.841×10^{-6} m^2/s)	Dirichlet (1.841×10^{-6} m^2/s)	Wall Function (nutLowReWallFunction)	Neumann (zeroGradient)	Symmetry

and a static temperature of $T_\infty = 283.19$ K. These thermodynamic properties determine the local speed of sound (a), calculated via the ideal gas relation for air ($\gamma = 1.4$, $R = 287.05$ J/kg·K):

$$a = \sqrt{\gamma R T_\infty} = \sqrt{1.4 \cdot 287.05 \cdot 288.15} \approx 340.3 \text{ m/s} \tag{52}$$

The inlet free-stream velocity (U_∞) is then derived directly from the definition of the Mach number:

$$U_\infty = M \cdot a = 0.7 \cdot 340.3 = 238.21 \text{ m/s} \tag{53}$$

This magnitude is applied as a Dirichlet condition strictly in the longitudinal u-direction ($\mathbf{u} = (238.21, 0, 0)$) to simulate a zero-degree angle of attack. At the outlet, the static pressure is pinned to $101,325$ Pa (Dirichlet) to serve as the numerical reference for the domain, while velocity and temperature utilize Neumann (*zeroGradient*) conditions to allow the wake to exit without reflection.

For the Spalart-Allmaras turbulence model, the inlet modified turbulent viscosity ($\tilde{\nu}$) requires a non-zero value to prevent the solution from remaining laminar. We apply a heuristic ratio of approximately $3\nu_\infty$ relative to the kinematic viscosity of air ($\nu_\infty \approx 1.5 \times 10^{-5}$ m^2/s):

$$\tilde{\nu}_{inlet} \approx 3 \cdot \nu_\infty = 4.5 \times 10^{-5} \text{ m}^2/\text{s} \tag{54}$$

This provides the stable turbulence ”seed” necessary for model initialization. Consistent with this initialization, the turbulent kinematic viscosity (ν_t) and turbulent thermal diffusivity (α_t) are treated as calculated fields at the inlet and outlet boundaries. At the fin surface, these variables employ specific wall functions to model the near-wall behavior: the *alphatJayatillekeWallFunction* for ν_t and the *nutLowReWallFunction* for α_t. These functions ensure the correct damping and heat transport characteristics within the boundary layer while the modified viscosity $\tilde{\nu}$ is strictly pinned to zero at the wall to enforce physical damping.

The fin surface employs a strict *noSlip* condition ($U = 0$) and zero turbulent viscosity ($\tilde{\nu} = 0$) to enforce physical wall damping, while the outlet pressure is pinned to $101,325$ Pa to anchor the domain's thermodynamics. Finally, the far-field boundaries are assigned a Neumann condition. This choice is critical for open-air simulations to ensure that the artificial limits of the computational domain do not interfere with the physical flow field. This way the flow is allowed to pass through the boundaries without the formation of non-physical boundary layers or ”choking” effects that would otherwise arise from a Dirichlet *noSlip* condition, thereby maintaining the integrity of the free-stream characteristics and the wake development behind the fin.

4 Numerical Solution Method

The starting point of any numerical method is the mathematical model representing the physics of the problem to be solved. This model usually consists of a set of partial differential equations (PDEs) and a set of boundary conditions. The discretization process involves transforming these continuous equations into a system of algebraic equations that can be solved numerically. The discretization process can be divided into four main steps:

1. **Geometric and Physical Modeling:** Definition of the domain geometry and the physical models (governing equations and fluid properties).
2. **Domain Discretization:** The division of the physical domain into a finite number of non-overlapping control volumes (cells) forming the computational mesh.
3. **Equation Discretization:** The transformation of the PDEs into algebraic equations for each control volume.
4. **Solution of Algebraic Equations:** The assembly of the local equations into a global system and its solution using iterative or direct methods.

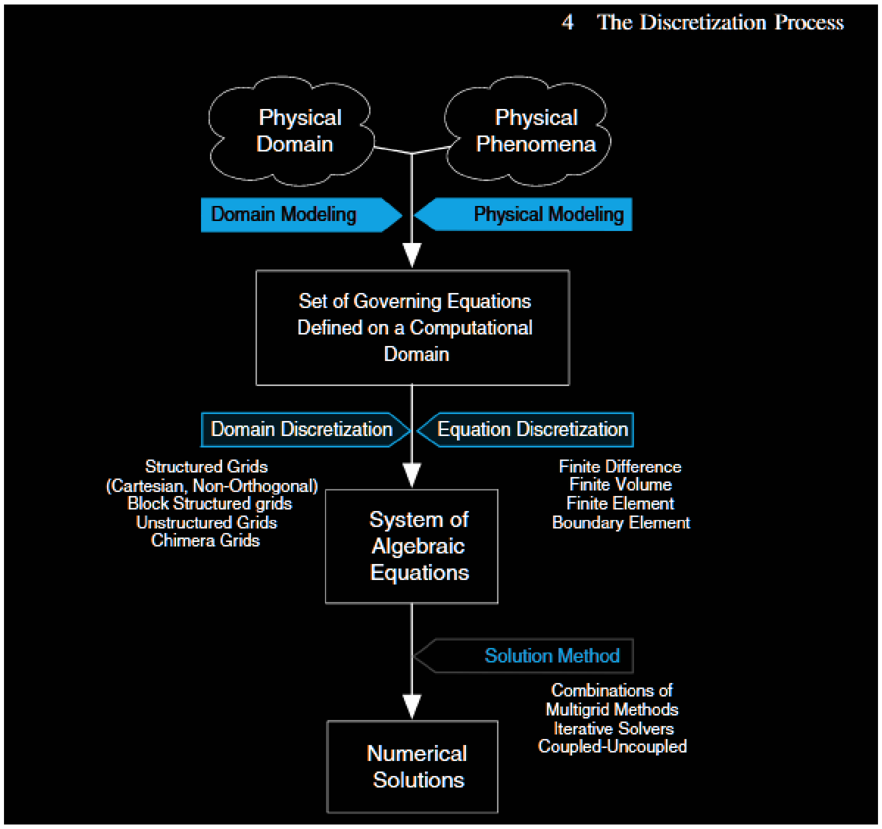

Figure 5: The Discretization Process

4.1 Numerical Model

The numerical solution of the flow field requires the simultaneous solution of the RANS equations, the equation of state, and the turbulence closure model. The RANS and Turbulence closue model equations are solved through a numerical iteration process until the residuals reach a predefined convergence criterion:

1. **Conservation of Mass (RANS):**

$$\nabla \cdot (\bar{\rho}\bar{\mathbf{u}}) = 0 \tag{55}$$

2. **Conservation of Momentum (RANS):**

$$\underbrace{\bar{\rho}(\bar{\mathbf{u}}\cdot\nabla\bar{\mathbf{u}})}_{\text{Convective Acceleration}} = \underbrace{-\nabla\bar{p}}_{\text{Pressure Forces}} + \underbrace{\nabla\cdot\bar{\tau}}_{\text{Viscous Forces}} \underbrace{-\nabla\cdot(\bar{\rho}\overline{\mathbf{u}'\otimes\mathbf{u}'})}_{\text{Reynolds Stresses}} \tag{56}$$

3. **Conservation of Energy (Total Enthalpy) (RANS):**

$$\underbrace{\nabla\cdot(\bar{\rho}\bar{\mathbf{u}}\bar{h}_0)}_{\text{Enthalpy Convection}} = \underbrace{\nabla\cdot(\bar{\alpha}\nabla\bar{h}_0)}_{\text{Mean Heat Diffusion}} - \underbrace{\nabla\cdot(\bar{\rho}\overline{\mathbf{u}'h_0'})}_{\text{Turbulent Heat Flux}} \tag{57}$$

4. **Spalart-Allmaras Turbulence Transport:**

$$\underbrace{\nabla\cdot(\rho\mathbf{u}\tilde{\nu})}_{\text{Advection}} = \underbrace{\frac{1}{\sigma}\nabla\cdot[\rho(\nu+\tilde{\nu})\nabla\tilde{\nu}] + \frac{c_{b2}\rho}{\sigma}(\nabla\tilde{\nu})^2}_{\text{Diffusion (Molecular and Turbulent)}} + \underbrace{\rho\tilde{\nu}P_{mod}}_{\text{Production}} - \underbrace{\rho\tilde{\nu}D_{mod}}_{\text{Destruction}} \tag{58}$$

Algebraic/Auxiliary Relations These equations are calculated directly at each step to update properties and ”close” the transport equations:

1. **Equation of State (Ideal Gas):** Links pressure, density, and temperature.

$$\bar{\rho} = \frac{\bar{p}}{R\bar{T}} \tag{59}$$

2. **Eddy Viscosity Calculation (S-A Closure):** The working variable $\tilde{\nu}$ is converted into the turbulent kinematic viscosity ν_t.

$$\nu_t = \tilde{\nu}f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu} \tag{60}$$

3. **Boussinesq Hypothesis:** Relates the kinematic eddy viscosity ν_t to the Reynolds Stress tensor in the momentum equation.

$$\tau_{ij}^R = \nu_t\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right) \tag{61}$$

4. **Sutherland’s Law:** A temperature-dependent viscosity model used to calculate the molecular viscosity μ as a function of the thermal field:

$$\mu = \mu_0\left(\frac{T}{T_0}\right)^{3/2}\frac{T_0+S}{T+S} \tag{62}$$

4.2 Discretization

4.2.1 Domain Discretization

Domain discretization, commonly referred to as meshing, is the process of subdividing the physical domain into a set of non-overlapping discrete elements. This process transforms the original geometry into a computational domain composed of grid elements or cells.

A mesh is fundamentally a collection of topological entities that define the space for numerical computation:

- **Elements (Cells):** The discrete volumes that cover the computational domain. In cell-centered Finite Volume Methods (FVM), the dependent variables are typically stored at the centroids of these cells.
- **Vertices (Nodes):** Specified points in space used to define the boundaries of the elements.
- **Faces:** The boundaries between adjacent elements. In 3D, these are surfaces; in 2D, they are lines.

For the discretization process to be implemented in a computer code, the mesh must contain information about how these entities are connected:

- **Internal Faces:** Faces shared by two adjacent elements. These allow for the calculation of fluxes between cells.
- **Boundary Faces:** Faces that coincide with the physical boundary of the domain. These are organized into patches (e.g., Patch #1, Patch #2) to which specific boundary conditions are applied.

The domain discretization step is the ”enabling ingredient” for the numerical solution. By dividing the domain into discrete structures, the governing partial differential equations can be integrated over each individual element C, leading to the local assembly of algebraic equations.

The computational domain is discretized into five distinct patch types to represent the virtual wind tunnel environment. The Inlet and Outlet govern the flow entry and exit, respectively. The Fin is defined as a no-slip physical boundary where the viscous boundary layer is resolved. The surrounding Walls (encompassing the top, bottom, front, and back extents) are treated as far-field boundaries. Finally, a Symmetry patch is applied to the longitudinal mid-plane. This reduces the computational cost by 50% by assuming flow symmetry, effectively removing one-half of the unknowns from the algebraic system of equations without loss of aerodynamic fidelity.

Table 4: Boundary Condition Configuration for the Computational Domain

Patch Name	**Physical Interpretation**
Inlet	Far-field Entry: Prescribes free-stream atmospheric conditions ($M = 0.7$).
Outlet	Far-field Exit: Enforces static pressure and prevents back-flow.
Fin	Solid Boundary: Resolved no-slip surface for force extraction.
Walls	Bounding Box: Far-field slip boundaries (Top, Bottom, Front, Back).
Symmetry	Mid-plane: Mirror boundary used to reduce computational cost by 50%.

[Report mesh information]

4.2.2 Equation Discretization

The algebraic equations resulting from the discretization process must be solved to obtain the discrete values of the dependent variables. The choice of solution methodology is generally independent of the discretization method itself and depends on the size and linearity of the system.

Direct methods, such as Gauss elimination or matrix inversion, obtain the solution by applying a complex algorithm once to a given set of coefficients. While these methods guarantee a solution if the matrix is non-singular, they are computationally expensive, requiring $O(N^3)$ operations for an $N \times N$ matrix. Consequently, they are rarely used in practical CFD problems, which typically involve hundreds of thousands of cells and nonlinear coefficients.

Instead, iterative methods are preferred as they follow a ”guess-and-correct” procedure to gradually refine the estimated solution. These methods, such as the Gauss-Seidel algorithm, visit each grid element and update the variable ϕ_P based on the prevailing values of its neighbors:

$$\phi_P = \frac{\sum_N a_N \phi_N + b_P}{a_P} \tag{63}$$

The domain is swept repeatedly until a specified convergence criterion is met. Iterative schemes are significantly more efficient for the large, sparse matrices encountered in CFD and are particularly well-suited for nonlinear problems, as coefficients can be updated using updated variable values as the iterations proceed. Consequently, CFD solvers primarily rely on iterative methods, which start with an initial guess and repeatedly refine the solution. These methods are significantly more memory-efficient and allow the solver to handle the non-linearities inherent in high-speed flows by updating coefficients at each step.

The Finite Difference Method (FDM) The Finite Difference Method (FDM) is one of the oldest and most computationally efficient numerical techniques for solving partial differential equations. Its primary advantage lies in its mathematical simplicity and the speed with which it can reach a converged solution. In FDM, the continuous derivatives in the governing equations are replaced by algebraic approximations based on Taylor series expansions. For example, a first-order derivative can be approximated as:

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi_{i+1} - \phi_i}{\Delta x} \tag{64}$$

FDM is often the fastest method to complete a simulation due to its minimal data overhead. By operating on discrete points rather than integrating over control volumes or elements, the framework maintains a low memory cost, as it avoids the storage of complex face-flux tables and large connectivity matrices. This streamlined structure allows algebraic operations to be highly vectorized, enabling rapid execution and high-speed iterations on hardware. Furthermore, FDM offers a distinct advantage in terms of cost-effective precision; achieving high-order accuracy is computationally inexpensive, allowing the solver to capture intricate flow physics with a relatively sparse distribution of points and significantly reducing the total time-to-solution.

The Finite Element Method (FEM) The Finite Element Method (FEM) is a numerical framework that discretizes the physical domain into a collection of sub-regions termed elements. While the Finite Volume Method (FVM) focuses on balancing fluxes across control volumes, FEM is built upon the approximation of the solution field throughout the entire continuum. The core of the FEM approach is the use of shape functions (N_i), which are used to interpolate the unknown variable ϕ within each element based on its values at discrete nodes:

$$\phi(x, y, z) \approx \sum_{i=1}^{n} N_i \phi_i \tag{65}$$

In this formulation, the governing partial differential equations are transformed into an integral "weak form" typically using the Galerkin method of weighted residuals. In engineering practice, FEM is particularly advantageous for structural analysis and heat conduction within the solid body of the rocket fin, as it naturally handles complex, curved geometries and provides high-order accuracy for stress and strain distributions.

Finite Volume Method (FVM) The Finite Volume numerical framework begins with the governing equations in their continuous differential form. For a steady-state scalar property ϕ, the transport equation is expressed as:

$$\underbrace{\nabla \cdot (\rho \mathbf{u} \phi)}_{\text{Convection}} = \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{Diffusion}} + \underbrace{S_\phi}_{\text{Source Term}} \tag{66}$$

To transition to the Finite Volume Method, this equation is integrated over a discrete control volume V_P:

$$\int_{V_P} \nabla \cdot (\rho \mathbf{u} \phi)\, dV = \int_{V_P} \nabla \cdot (\Gamma \nabla \phi)\, dV + \int_{V_P} S_\phi \, dV \tag{67}$$

By applying Gauss's Divergence Theorem, the volume integrals for the flux terms are transformed into surface integrals over the boundary S_P of the control volume:

$$\oint_{S_P} (\rho \mathbf{u} \phi) \cdot \mathbf{n}\, dS = \oint_{S_P} (\Gamma \nabla \phi) \cdot \mathbf{n}\, dS + \int_{V_P} S_\phi \, dV \tag{68}$$

This formulation establishes that the net convective flux leaving the control volume is balanced by the net diffusive flux and the internal generation or destruction provided by the source term. In the discrete limit, the integrals are approximated as the summation of fluxes across the faces f and a linearized source contribution:

$$\sum_f (\rho \mathbf{u} \phi \cdot \mathbf{n})_f A_f = \sum_f (\Gamma \nabla \phi \cdot \mathbf{n})_f A_f + S_\phi V_P \tag{69}$$

As illustrated in Figure 6, this discrete approach ensures that the flux leaving one control volume is identically the flux entering the adjacent volume, maintaining physical consistency across the entire mesh.

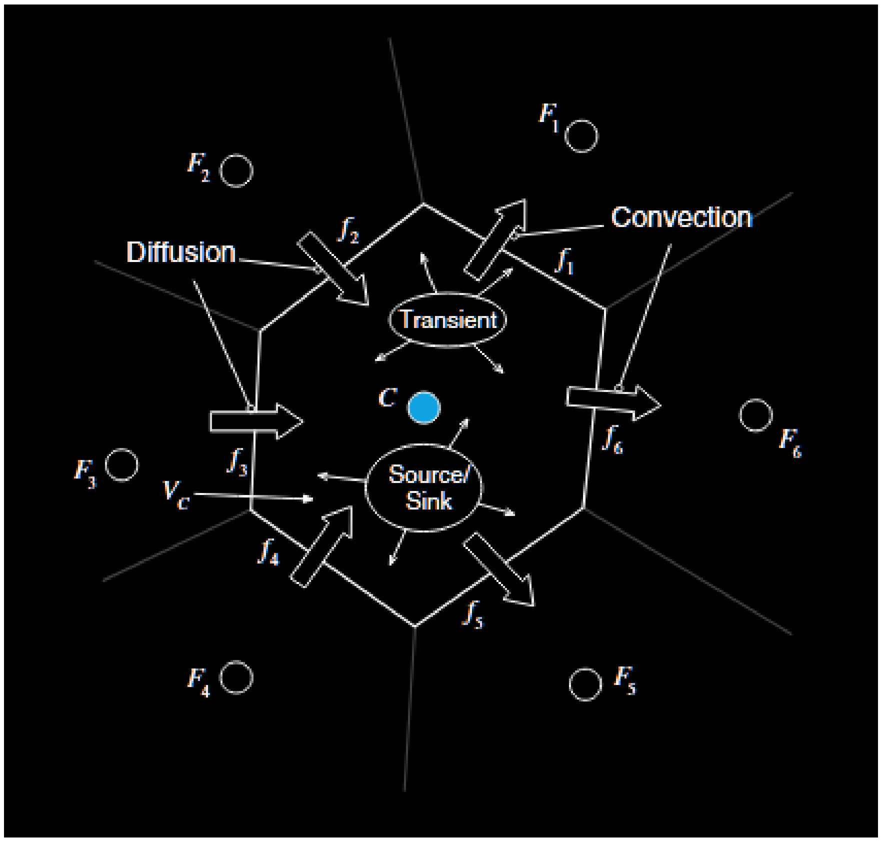

Figure 6: Conservation of fluxes between adjacent discrete elements in the Finite Volume Method.

In the Finite Volume Method, the accuracy of the discretized system is largely determined by how surface integrals are evaluated at the control volume faces. To transform the continuous surface integral into a discrete form, a Gaussian quadrature approach is employed:

$$\int_f (\mathbf{J}^\phi \cdot \mathbf{n}) dS = \sum_{ip \sim ip(f)} (\mathbf{J}^\phi \cdot \mathbf{n})_{ip} \omega_{ip} S_f \tag{70}$$

where ip denotes an integration point and ω_{ip} is the corresponding weighing function. For the current simulation, a single integration point ($ip = 1$) located at the centroid of the face is used, with a weighing function $\omega_{ip} = 1$.

This specific approximation is traditionally used by finite volume codes, also known as the trapezoidal rule, is second-order accurate and is applicable in both two and three dimensions. By utilizing this single-point integration at the face centroid, the solver maintains a balance between computational efficiency and the high-fidelity spatial accuracy required to capture the pressure gradients and viscous effects at $M \approx 0.7$.

Semi-Discretized Governing Equations To transform the continuous partial differential equations into a form solvable by the Finite Volume Method, each equation is integrated over a control volume V_P and converted into a surface integral balance using Gauss's Divergence Theorem. The semi-discretized forms of the RANS and Spalart-Allmaras equations are:

1. **Mass Conservation (Continuity):**

$$\underbrace{\sum_f (\bar{\rho} \bar{\mathbf{u}} \cdot \mathbf{n})_f A_f}_{\text{Net Mass Flux}} = 0 \tag{71}$$

2. **Momentum Conservation:**

$$\underbrace{\sum_f (\bar{\rho}\bar{\mathbf{u}} \otimes \bar{\mathbf{u}} \cdot \mathbf{n})_f A_f}_{\text{Convective Flux}} = \underbrace{-\sum_f (\bar{p}\mathbf{n})_f A_f}_{\text{Pressure Force}} + \underbrace{\sum_f (\bar{\tau}_{eff} \cdot \mathbf{n})_f A_f}_{\text{Viscous/Turbulent Diffusion}} \tag{72}$$

3. **Total Enthalpy Conservation:**

$$\underbrace{\sum_f (\bar{\rho}\bar{h}_0 \bar{\mathbf{u}} \cdot \mathbf{n})_f A_f}_{\text{Enthalpy Convection}} = \underbrace{\sum_f (\bar{\alpha}_{eff} \nabla \bar{h}_0 \cdot \mathbf{n})_f A_f}_{\text{Energy Diffusion}} \tag{73}$$

4. **Spalart-Allmaras Transport:**

$$\underbrace{\sum_f (\rho\tilde{\nu}\mathbf{u} \cdot \mathbf{n})_f A_f}_{\text{Advection}} = \underbrace{\sum_f (\Gamma_{\tilde{\nu}} \nabla \tilde{\nu} \cdot \mathbf{n})_f A_f}_{\text{Diffusion}} + \underbrace{\int_{V_P} (P_{\tilde{\nu}} - D_{\tilde{\nu}}) dV}_{\text{Net Source (Production - Destruction)}} \tag{74}$$

In these equations, the subscript f denotes values evaluated at the cell faces, $\mathbf{n}$ is the outward-pointing unit normal vector, and A_f is the face area. This summation represents the net flux through the control volume boundaries, which is the foundational step before applying numerical interpolation schemes.

4.3 Numerical Schemes and Matrix Assembly

To transform the integral conservation laws into a system of algebraic equations, the physical variables must be approximated based on their spatial location relative to the control volume. The terms within the volume integrals are evaluated using the mean value approximation, which assumes the value at the cell centroid P represents the average over the entire volume V_P:

$$\int_{V_P} \phi \, dV \approx \phi_P V_P \tag{75}$$

Conversely, the transport through the boundaries requires the evaluation of variables at the control volume faces f. Since these values are not stored directly, they are determined using interpolation schemes that relate the face value ϕ_f to the known values at the adjacent cell centroids:

$$\phi_f = f(\phi_{centroids}) \tag{76}$$

Similarly, the surface gradients required for diffusion terms are approximated using the centroid data to determine the change across the interface. This combination of volume-averaged values and interpolated face values allows for the assembly of a closed linear system for each discrete element. Thus, to close this system numerical interpolation schemes are employed to approximate face values and gradients.

4.3.1 Mean-Value Approximation (Source Terms)

The volume integral representing the production and destruction of turbulent viscosity in the Spalart-Allmaras model is evaluated at the cell centroid P. Using the Mean-Value Theorem, the term is approximated as:

$$\int_{V_P} (P_{\tilde{\nu}} - D_{\tilde{\nu}}) \, dV \approx (P_{\tilde{\nu}} - D_{\tilde{\nu}})_P V_P \tag{77}$$

This assumes the value at the centroid is representative of the entire volume. Since this data is already stored at the centroid, no interpolation is required.

4.3.2 Convection Schemes (Advection Terms)

Convective terms, such as the momentum flux $\sum_f (\bar{\rho}\bar{\mathbf{u}} \otimes \bar{\mathbf{u}} \cdot \mathbf{n})_f A_f$ and enthalpy flux $\sum_f (\bar{\rho}\bar{h}_0\bar{\mathbf{u}} \cdot \mathbf{n})_f A_f$, require the evaluation of variables at cell faces. Because the flow has a distinct direction, interpolation must account for the "upwind" information:

$$\phi_f = f(\phi_P, \phi_N, \mathbf{u}_f) \tag{78}$$

Interpolation is the process of reconstructing the variable value at the cell face f from the known values at the centroids P and N. In the Upwind approach, the face value ϕ_f is determined solely by the direction of the flow. Mathematically, for a mass flux $\dot{m}_f = \rho_f \mathbf{u}_f \cdot \mathbf{n}_f$, the first-order upwind scheme is expressed as:

$$\phi_f = \underbrace{\frac{1}{2}\left(\phi_P + \phi_N\right)}_{\text{Central Average}} + \underbrace{\frac{1}{2}\text{sgn}(\dot{m}_f)\left(\phi_P - \phi_N\right)}_{\text{Upwind Directional Bias}} \tag{79}$$

This ensures that ϕ_f takes the value of the upstream cell, providing high numerical stability but introducing significant numerical diffusion. To improve accuracy, the Linear Upwind scheme (second-order) uses a linear Taylor series expansion from the upstream centroid to the face:

$$\phi_f = \phi_{\text{upwind}} + \nabla\phi_{\text{upwind}} \cdot \mathbf{d}_f \tag{80}$$

where $\mathbf{d}_f$ is the vector from the upwind cell centroid to the face center. The term "linear" refers to this linear reconstruction using the local gradient $\nabla\phi$, which allows the scheme to better capture the actual distribution of the variable across the cell.

For this study, the LinearUpwind scheme is applied to the momentum flux $\sum_f (\bar{\rho}\bar{\mathbf{u}} \otimes \bar{\mathbf{u}} \cdot \mathbf{n})_f A_f$ and enthalpy flux $\sum_f (\bar{\rho}\bar{h}_0\bar{\mathbf{u}} \cdot \mathbf{n})_f A_f$ to provide second-order accuracy for the primary flow variables at $M = 0.7$. Conversely, the standard Upwind scheme is utilized for the transport of the Spalart-Allmaras variable $\tilde{\nu}$ to ensure boundedness and prevent non-physical results. Additionally,a standard Upwind approach is applied to the pressure-velocity coupling terms and the density-based mass flux ϕ; this provides the necessary numerical damping to stabilize the pressure-density relationship inherent in high-subsonic compressible flow.

4.3.3 Diffusion Schemes (Laplacian Terms)

Diffusion terms represent the transport of variables due to molecular or turbulent gradients. Based on the governing equations, these include the Viscous/Turbulent Diffusion in momentum $\sum_f (\bar{\tau}_{eff} \cdot \mathbf{n})_f A_f$, Energy Diffusion $\sum_f (\bar{\alpha}_{eff}\nabla\bar{h}_0 \cdot \mathbf{n})_f A_f$, and the Spalart-Allmaras turbulent diffusion $\sum_f (\Gamma_{\tilde{\nu}}\nabla\tilde{\nu} \cdot \mathbf{n})_f A_f$.

The evaluation of the face gradient $(\nabla\phi)_f$ is typically performed using the values of the two cells sharing the face, P and N. For an orthogonal mesh, the surface normal gradient is approximated using a second-order Central Differencing scheme:

$$(\nabla\phi \cdot \mathbf{n})_f \approx \frac{\phi_N - \phi_P}{|\mathbf{d}_{PN}|} \tag{81}$$

where $|\mathbf{d}_{PN}|$ is the distance between the cell centroids. This formulation assumes that the vector connecting the centroids is parallel to the face normal vector $\mathbf{n}$.

However, for the complex geometry of a trapezoidal fin, the mesh contains non-orthogonal cells where the centroid-to-centroid vector is not aligned with the face normal. To maintain accuracy, a non-orthogonal correction is applied to the general diffusion coefficient Γ:

$$\int_V \nabla \cdot (\Gamma\nabla\phi) dV = \sum_f \Gamma_f \underbrace{(\nabla\phi \cdot \mathbf{n})_f A_f}_{\text{Orthogonal}} + \sum_f \Gamma_f \underbrace{(\nabla\phi \cdot \mathbf{n}_{corr})_f A_f}_{\text{Non-orthogonal correction}} \tag{82}$$

In this study, the Gauss linear corrected scheme is employed for all Laplacian terms. This choice applies second-order central differencing for the orthogonal component while adding a correction term to account for mesh skewness. This ensures that the viscous stresses and heat conduction are accurately resolved across the boundary layers of the fin without introducing non-physical numerical errors due to mesh topology.

4.3.4 Gradient Schemes

Gradients are required for evaluating the pressure force $-\sum_f (\bar{p}\mathbf{n})_f A_f$, for the second-order reconstruction in the Linear Upwind convection scheme, and as a mechanism to quantify spatial variation whenever non-uniformity of the flow variables across the mesh. This is critical for evaluating forces and ensuring high-order accuracy in the solution. To compute these gradients numerically, the Gauss Linear scheme is employed, which is a second-order accurate method based on the Green Gauss Theorem. In a discretized CFD mesh, we simplify the integral by assuming the gradient is constant over the cell volume. This allows us to calculate the gradient at the cell center ($\nabla\phi_P$) by summing the values at the center of each face (f):

$$\nabla\phi_P = \frac{1}{V_P}\sum_f \phi_f \mathbf{S}_f \tag{83}$$

where V_P is the cell volume, and $\mathbf{S}_f$ is the face area vector (pointing outward).

The "Linear" part of the Gauss Linear scheme refers to how the value ϕ_f at the face is determined. Since the solver only knows the values at the cell centers (P and its neighbor N), it uses Linear Interpolation to find the value exactly at the face between them:

$$\phi_f = w_P\phi_P + (1 - w_P)\phi_N \tag{84}$$

where w_P is a weighting factor based on the distance from the cell centers to the face. By combining the Green-Gauss surface summation with linear interpolation, the solver achieves a robust, second-order accurate representation of the flow physics, which is essential for capturing the sharp pressure gradients present in $M = 0.7$ transonic flow.

4.4 The Algebraic System of Equations

The transition from the semi-discretized conservation laws to a solvable linear algebraic system is achieved through the substitution of numerical schemes and the linearization of non-linear coefficients. For a control volume V_P, the semi-discretized equation represents a summation of face fluxes that must be transformed into a linear relationship between the unknown value at cell centroid P and its immediate neighbors N:

$$a_P\phi_P^{k+1} + \sum_N a_N\phi_N^{k+1} = b_P \tag{85}$$

To reach this form, two distinct operations are performed on the flux terms:

1. **Substitution of Numerical Schemes:** The unknown face values ϕ_f and gradients $(\nabla\phi)_f$ are replaced by expressions involving the cell center unknowns ϕ_P and ϕ_N.

2. **Freezing of Non-Linear Coefficients:** Terms involving products of unknowns are linearized by using values from the previous iteration (k):

$$\dot{m}_f\phi_f \approx \underbrace{\dot{m}_f^k}_{\text{Known}}\ \underbrace{\phi_f^{k+1}}_{\text{Unknown}} \tag{86}$$

   By treating $\dot{m}_f^k$ as a known constant multiplier, the equation remains linear for the variable ϕ at the current iteration $k + 1$.

The final algebraic coefficients are categorized based on their role in the system:

- **Diagonal Coefficient (a_P):** Represents the self-influence of cell P. It sums all terms multiplying the unknown ϕ_P^{k+1}, including outgoing convective fluxes and the center-related parts of diffusive gradients.

- **Neighbor Coefficients (a_N):** Represents the influence of adjacent cells. These are populated by incoming convective fluxes and the neighbor-related parts of the diffusion terms.

- **Source Term (b_P):** Represents the "Right-Hand Side" (RHS) containing all fully known values, such as constant pressure gradients, the $\nabla\phi \cdot \mathbf{d}$ explicit corrections from the *Linear Upwind* scheme, and non-orthogonal mesh corrections.

At this stage, the governing physical laws have been reduced to a system of arithmetic relationships, where the accuracy of the coefficients a_P and a_N is refined iteratively as the solution progresses. By applying this linearization to every control volume P in the domain, the localized physical balances are converted into a global sparse matrix system of the form $[A]\{\Phi\} = \{B\}$.

For a steady-state simulation, the solver assembles the discretized equations into a sparse matrix system of the form $\mathbf{A\Phi} = \mathbf{B}$. In this framework, the matrix $\mathbf{A}$ contains the coefficients a_P and a_N representing the convective and diffusive links between cells, while $\mathbf{B}$ contains the linearized source terms and boundary condition contributions. The resulting system for a mesh of n cells is represented as:

$$\begin{bmatrix} a_{P,1} & a_{N,12} & \cdots & a_{N,1n} \\ a_{N,21} & a_{P,2} & \cdots & a_{N,2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,n1} & a_{N,n2} & \cdots & a_{P,n} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix} \tag{87}$$

For the steady-state, compressible flow investigation at $M \approx 0.7$, the governing transport equations are discretized using the Finite Volume Method. For any specific control volume P, the system of equations (Continuity, Momentum (u, v, w), Total Enthalpy, and S-A Turbulence) is represented as a local linear system. For brevity, the iteration superscript $k + 1$ is omitted in the following notation:

$$\underbrace{\begin{bmatrix} a_P^{pp} & a_P^{pu} & a_P^{pv} & a_P^{pw} & 0 & 0 \\ a_P^{up} & a_P^{uu} & 0 & 0 & 0 & 0 \\ a_P^{vp} & 0 & a_P^{vv} & 0 & 0 & 0 \\ a_P^{wp} & 0 & 0 & a_P^{ww} & 0 & 0 \\ 0 & a_P^{hu} & a_P^{hv} & a_P^{hw} & a_P^{hh} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_P^{\tilde{\nu}\tilde{\nu}} \end{bmatrix}}_{\text{Local Coefficient Matrix } [A_P]} \underbrace{\begin{Bmatrix} p \\ u \\ v \\ w \\ h \\ \tilde{\nu} \end{Bmatrix}_P}_{\{\phi\}_P} + \sum_N \underbrace{\begin{bmatrix} a_N^{pp} & a_N^{pu} & a_N^{pv} & a_N^{pw} & 0 & 0 \\ a_N^{up} & a_N^{uu} & 0 & 0 & 0 & 0 \\ a_N^{vp} & 0 & a_N^{vv} & 0 & 0 & 0 \\ a_N^{wp} & 0 & 0 & a_N^{ww} & 0 & 0 \\ 0 & a_N^{hu} & a_N^{hv} & a_N^{hw} & a_N^{hh} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_N^{\tilde{\nu}\tilde{\nu}} \end{bmatrix}}_{[A_N]} \underbrace{\begin{Bmatrix} p \\ u \\ v \\ w \\ h \\ \tilde{\nu} \end{Bmatrix}_N}_{\{\phi\}_N} = \begin{Bmatrix} b_p \\ b_u \\ b_v \\ b_w \\ b_h \\ b_{\tilde{\nu}} \end{Bmatrix} \tag{88}$$

The components of the algebraic system correspond to the physical mechanisms defined in the reduced RANS and S-A equations:

- **Convection-Diffusion Balance ($a_P^{\phi\phi}, a_N^{\phi\phi}$):** These diagonal entries represent the net transport of property ϕ (velocity, energy, or $\tilde{\nu}$) due to fluid motion and molecular/turbulent diffusion across the faces.
- **Pressure-Velocity Coupling (a_P^{up}, a_P^{pu}):** These off-diagonal blocks represent the gradient of pressure ∇p acting as a source in the momentum equations, and the velocity influence in the continuity (pressure) equation. This coupling is critical for capturing the aerodynamic forces on the fin.
- **Energy-Flow Coupling ($a_P^{hu}, a_P^{hv}, a_P^{hw}$):** These terms account for the work done by pressure and viscous forces, as well as the convective transport of total enthalpy.
- **S-A Turbulence ($a_P^{\tilde{\nu}\tilde{\nu}}, b_{\tilde{\nu}}$):** The turbulence equation is linked through the updated density and velocity fields. The source term $b_{\tilde{\nu}}$ contains the production and destruction of turbulent viscosity, where destruction is often linearized into $a_P^{\tilde{\nu}\tilde{\nu}}$ to enhance diagonal dominance.
- **Cell State Vectors ($\{\phi\}_P$ and $\{\phi\}_N$):** These vectors represent the primary variables $(p, u, v, w, h, \tilde{\nu})$ at the centroids of the owner cell P and its neighbors N. Based on the Mean Value Theorem, these centroid values are treated as the average value for the entire control volume.
- **Linearized Sources (b):** The right-hand side vector contains all terms that do not depend linearly on the current iteration's variables, such as constant body forces or boundary condition values.

4.5 Global Conservation

The primary advantage of the Finite Volume Method is its inherent property of global conservation. For a steady-state transport equation discretized using the Finite Volume Method, the net flux of a property ϕ through the global domain boundaries is identically zero in the absence of internal sources, regardless of the mesh resolution.

Consider the steady-state conservation law for a general property ϕ in a domain Ω:

$$\nabla \cdot \mathbf{J} = S_\phi \tag{89}$$

where $\mathbf{J}$ is the total flux (convective and diffusive) and S_ϕ is the source term. In the Finite Volume Method, the domain Ω is partitioned into a set of N non-overlapping control volumes $\{V_i\}_{i=1}^{N}$ such that $\Omega = \bigcup V_i$.

Integrating over an arbitrary control volume V_i and applying the Gauss Divergence Theorem:

$$\int_{V_i} \nabla \cdot \mathbf{J} \, dV = \oint_{\partial V_i} \mathbf{J} \cdot \mathbf{n} \, dS = \int_{V_i} S_\phi \, dV \tag{90}$$

where ∂V_i represents the boundary of the cell and $\mathbf{n}$ is the outward-pointing unit normal vector. The surface integral is discretized as a summation of fluxes over the faces f of the cell:

$$\sum_{f \in \partial V_i} \mathbf{J}_f \cdot \mathbf{n}_f A_f = Q_i \tag{91}$$

Summing this relation over all N control volumes in the domain yields:

$$\sum_{i=1}^{N} \left(\sum_{f \in \partial V_i} \mathbf{J}_f \cdot \mathbf{n}_f A_f \right) = \sum_{i=1}^{N} Q_i \tag{92}$$

The double summation can be decomposed into two distinct sets of faces: internal faces ($\mathcal{F}_{int}$) and boundary faces ($\mathcal{F}_{ext}$). For any internal face f shared by adjacent cells V_i and V_j, the outward normal vectors satisfy $\mathbf{n}_{f,i} = -\mathbf{n}_{f,j}$. Since the FVM enforces a unique flux calculation at each face, the internal contributions cancel pairwise:

$$(\mathbf{J}_f \cdot \mathbf{n}_{f,i} A_f) + (\mathbf{J}_f \cdot \mathbf{n}_{f,j} A_f) = 0, \quad \forall f \in \mathcal{F}_{int} \tag{93}$$

Substituting this into the global summation, all internal terms vanish, leaving only the flux through the physical boundaries of the domain:

$$\sum_{f \in \mathcal{F}_{ext}} \mathbf{J}_f \cdot \mathbf{n}_f A_f = \sum_{i=1}^{N} Q_i \tag{94}$$

Thus, the net flux across the global boundary is exactly balanced by the total integrated source within the domain. In the limit of zero sources, the global flux is identically conserved. ■

5 Numerical Solution

5.1 Numerical Algorithm

5.1.1 Direct Methods

Direct methods for solving the system $\mathbf{A}\boldsymbol{\phi} = \mathbf{b}$ are characterized by a finite sequence of operations that, in the absence of round-off errors, would yield the exact solution. These methods involve the systematic transformation of the coefficient matrix $\mathbf{A}$ into a simpler form that allows for easy extraction of the unknown vector $\boldsymbol{\phi}$.

Gaussian Elimination and Forward Elimination Gaussian Elimination is the most fundamental direct algorithm. The first phase, Forward Elimination, reduces the matrix $\mathbf{A}$ to an upper triangular form $\mathbf{U}$. For a system of n equations, the element a_{ij} is eliminated in the k-th step using a multiplier $m_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)}$. The row operations are defined as:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right) a_{kj}^{(k)} \quad \text{and} \quad b_i^{(k+1)} = b_i^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}\right) b_k^{(k)} \tag{95}$$

where $i, j = k+1, \ldots, n$. This process is repeated until the system takes the form:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b'_2 \\ \vdots \\ b_n^{(n-1)} \end{Bmatrix} \tag{96}$$

Backward Substitution Once the matrix is in upper triangular form, the unknowns are solved in reverse order. The last unknown ϕ_n is computed directly, and subsequent variables are found by substituting the known values back into the previous equations:

$$\phi_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \quad ; \quad \phi_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j=i+1}^{n} a_{ij}\phi_j\right) \tag{97}$$

This step requires $O(n^2)$ operations, whereas the preceding forward elimination requires $O(n^3)$, making the elimination phase the computational bottleneck.

LU Decomposition LU Decomposition is a more efficient variation of Gaussian Elimination, particularly when solving the same matrix $\mathbf{A}$ for multiple right-hand side vectors $\mathbf{b}$. The matrix $\mathbf{A}$ is factored into two matrices: $\mathbf{L}$ (Lower triangular with unit diagonals) and $\mathbf{U}$ (Upper triangular).

$$\mathbf{A}\boldsymbol{\phi} = (\mathbf{L}\mathbf{U})\boldsymbol{\phi} = \mathbf{b} \tag{98}$$

The solution is obtained via a two-step process:

1. Solve $\mathbf{L}\mathbf{y} = \mathbf{b}$ using Forward Substitution: $y_i = b_i - \sum_{j=1}^{i-1} L_{ij} y_j$.
2. Solve $\mathbf{U}\boldsymbol{\phi} = \mathbf{y}$ using Backward Substitution: $\phi_i = \frac{1}{U_{ii}}(y_i - \sum_{j=i+1}^{n} U_{ij}\phi_j)$.

This decomposition effectively ”stores” the elimination steps, allowing the solver to bypass the $O(n^3)$ cost if the boundary conditions (the $\mathbf{b}$ vector) change while the geometry and mesh (the $\mathbf{A}$ matrix) remain constant.

5.1.2 Iterative Methods

[Report on the methods used.]

Direct methods are generally inappropriate for solving large systems of equations, particularly when the coefficient matrix is sparse i.e. most elements are zero. For a 3D simulation, computing a direct matrix inverse is computationally expensive, requiring $O(N^3)$ operations and massive memory overhead. Furthermore, fluid flow problems are highly non-linear with solution-dependent coefficients, necessitating an iterative process where a pre-assigned level of convergence is reached rather than seeking a fully converged solution at every step.

Iterative methods compute a series of solutions $\phi^{(n)}$ that converge toward the exact solution ϕ from a starting guess $\phi^{(0)}$. This process relies on matrix decomposition, where the coefficient matrix $\mathbf{A}$ is written as:

$$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U} \tag{99}$$

where $\mathbf{D}$, $\mathbf{L}$, and $\mathbf{U}$ refer to the diagonal, strictly lower, and strictly upper matrices, respectively. The general iterative form is then:

$$\mathbf{M}\phi^{(n)} = \mathbf{N}\phi^{(n-1)} + \mathbf{b} \tag{100}$$

where $\mathbf{A} = \mathbf{M} - \mathbf{N}$.

Jacobi Method The Jacobi method is the simplest iterative technique, where the diagonal elements of $\mathbf{A}$ are used to solve for each unknown independently using values from the previous iteration. In matrix form, $\mathbf{M} = \mathbf{D}$ and $\mathbf{N} = -(\mathbf{L} + \mathbf{U})$. The local update for each cell i is:

$$\phi_i^{(n)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} \phi_j^{(n-1)} \right) \tag{101}$$

Gauss-Seidel Method The Gauss-Seidel method improves upon Jacobi by using the most recently updated values within the same iteration. This essentially splits the matrix such that $\mathbf{M} = (\mathbf{D} + \mathbf{L})$ and $\mathbf{N} = -\mathbf{U}$. The mathematical formulation is:

$$\phi_i^{(n)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} \phi_j^{(n)} - \sum_{j>i} a_{ij} \phi_j^{(n-1)} \right) \tag{102}$$

This method generally converges twice as fast as Jacobi but remains sensitive to the spectral radius of the iteration matrix.

Preconditioning of Iterative Methods To accelerate convergence, preconditioning is applied to reduce the condition number of the system. Instead of solving $\mathbf{A}\phi = \mathbf{b}$, the solver handles $\mathbf{P}^{-1}\mathbf{A}\phi = \mathbf{P}^{-1}\mathbf{b}$, where $\mathbf{P}$ is a preconditioning matrix that approximates $\mathbf{A}$ but is easy to invert. Effective preconditioning, such as Incomplete LU (ILU) factorization, ensures the eigenvalues of the iteration matrix remain well within the unit circle, accelerating the reduction of the residual error.

Steepest Descent (Gradient Methods) Gradient methods view the solution of $\mathbf{A}\phi = \mathbf{b}$ as a minimization problem of a quadratic functional. The Steepest Descent method iteratively updates the solution by moving in the direction of the local negative gradient, or the residual $\mathbf{r}^{(n)} = \mathbf{b} - \mathbf{A}\phi^{(n)}$:

$$\phi^{(n+1)} = \phi^{(n)} + \alpha^{(n)} \mathbf{r}^{(n)} \tag{103}$$

where $\alpha^{(n)}$ is a step-size parameter calculated to minimize the error along the search direction. While foundational, this is often superseded in CFD by the Conjugate Gradient method for better efficiency in high-dimensional spaces.

Conjugate Gradient Family (Symmetric Systems) The Conjugate Gradient (CG) method is an orthogonal search direction algorithm used for symmetric, positive-definite matrices, such as the pressure Poisson equation. Unlike Steepest Descent, which may re-traverse previous directions, CG ensures each new search direction $\mathbf{p}^{(n)}$ is $\mathbf{A}$-orthogonal (conjugate) to all previous directions:

$$\mathbf{p}^{(n)} = \mathbf{r}^{(n)} + \beta^{(n-1)} \mathbf{p}^{(n-1)} \tag{104}$$

where β is a scalar that ensures conjugacy. This prevents the "zig-zag" convergence behavior and ensures the solution is found in at most N iterations for an N-dimensional system, though preconditioning typically reduces this to $\sqrt{N}$ in practice.

Krylov Subspace Family (Non-Symmetric Systems) For the momentum and turbulence equations, the presence of advection terms makes the system matrix non-symmetric, rendering standard CG ineffective. The Bi-Conjugate Gradient (BiCG) method and its successor, the Preconditioned Bi-Conjugate Gradient Stabilized (PBiCGStab) method, are employed. PBiCGStab applies a stabilization step to the BiCG residuals to smooth out oscillatory convergence. The update involves two search directions to minimize the residual $\mathbf{r}$ across the Krylov subspace:

$$\mathcal{K}_n(\mathbf{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{n-1}\mathbf{r}_0\} \tag{105}$$

This makes it the industry standard for high-speed compressible flows where convective transport dominates.

Multigrid Acceleration and Smoothers The Geometric-Algebraic Multi-Grid (GAMG) solver accelerates convergence by solving the system on a series of coarser meshes. Standard smoothers, such as Gauss-Seidel or the Diagonal Incomplete Cholesky (DIC), are used to remove high-frequency errors on the fine mesh, while the coarse levels handle the low-frequency errors that would otherwise stall a single-grid solver. The transition between levels follows a V-cycle, where the residual $\mathbf{r}$ is restricted to coarser grids and the correction $\mathbf{e}$ is prolonged back:

$$\mathbf{A}_{coarse}\mathbf{e}_{coarse} = \mathbf{r}_{coarse} \tag{106}$$

For this Mach 0.7 simulation, a hybrid solver strategy was selected for optimal efficiency and stability. The pressure field (p) is solved using the **GAMG** method to ensure rapid global pressure communication across the domain. For the non-symmetric transport equations, including velocity ($\mathbf{u}$), enthalpy (h), and the modified turbulent viscosity ($\tilde{\nu}$), the **PBiCGStab** solver is utilized. This is paired with a **Diagonal Incomplete LU (DILU)** preconditioner to cluster the eigenvalues of the system matrix near unity, providing a robust solution path for the stiff matrices generated by high-speed compressible flow.

5.1.3 Algorithm Architecture

The solution of the Navier-Stokes equations is complicated by the fact that the pressure field does not have an explicit transport equation; instead, it is implicitly coupled with the velocity field through the continuity equation. To resolve this, a *pressure-based segregated algorithm* is required to iteratively couple the pressure and velocity fields.

The coupling challenge is most evident when viewing the system in matrix form:

$$\mathbf{A}\mathbf{u} = \begin{pmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f}_b \\ 0 \end{pmatrix} \tag{107}$$

The presence of the zero diagonal block indicates a saddle point problem, which cannot be solved directly by standard iterative means. One approach to resolving this coupling is to reformulate the system by decomposing the matrix $\mathbf{A}$ into lower ($\mathbf{L}$) and upper ($\mathbf{U}$) triangular block matrices. Using the Schur complement of $\mathbf{F}$, the system can be factored as:

$$\mathbf{A} = \begin{pmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{B} & -\mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{F}^{-1}\mathbf{B}^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \mathbf{L}\mathbf{U} \tag{108}$$

In this decomposition, the term $-\mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T$ is the Schur complement matrix, which effectively maps the pressure onto the divergence of the velocity. This allows the monolithic system $\mathbf{A}\mathbf{u} = \mathbf{b}$ to be viewed as a two-step process:

1. **Forward Substitution (Lower System):** Solve for intermediate variables $(\mathbf{v}^*, p^*)^T$:

$$\begin{pmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{B} & -\mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T \end{pmatrix} \begin{pmatrix} \mathbf{v}^* \\ p^* \end{pmatrix} = \begin{pmatrix} \mathbf{f}_b \\ 0 \end{pmatrix} \tag{109}$$

   From this, we see that $\mathbf{F}\mathbf{v}^* = \mathbf{f}_b$ and the pressure equation is linked to the mass imbalance $\mathbf{B}\mathbf{v}^*$.

2. **Backward Substitution (Upper System):** Solve for the final fields $(\mathbf{v}, p)^T$:

$$\begin{pmatrix} \mathbf{I} & \mathbf{F}^{-1}\mathbf{B}^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{v}^* \\ p^* \end{pmatrix} \tag{110}$$

However, computing $\mathbf{F}^{-1}$ is computationally prohibitive for large-scale fluid problems. To make this tractable, the SIMPLE family of algorithms approximates the Schur complement by replacing $\mathbf{F}^{-1}$ with the inverse of its diagonal elements, $\mathbf{D}^{-1}$. This substitution transforms the exact block factorization into an efficient segregated solution procedure:

- **Solve Momentum:** $\mathbf{F}\mathbf{v}^* = \mathbf{f}_b$
- **Solve Pressure:** $(-\mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T)p^* = -\mathbf{B}\mathbf{v}^*$
- **Update Velocity:** $\mathbf{v} = \mathbf{v}^* - \mathbf{D}^{-1}\mathbf{B}^T p^*$
- **Update Pressure:** $p = p^*$

This segregated sequence is repeated within an outer loop until the residuals for both the momentum and continuity equations converge to a specified tolerance. Within each step, iterative linear solvers are employed to solve the discretized algebraic systems for $\mathbf{v}^*$ and p^*.

Compressibility and Turbulence Coupling To adapt the segregated framework for the compressible RANS equations with the Spalart-Allmaras (S-A) turbulence model, the Schur complement must be extended to account for variable density ρ and eddy viscosity ν_t. In this context, density is coupled to pressure and temperature through the ideal gas law, $\rho = p/(RT)$. This introduces a diagonal contribution to the pressure equation representing fluid compressibility. Furthermore, the momentum matrix $\mathbf{F}$ is updated to include the effective viscosity $\nu_{eff} = \nu + \nu_t$, where ν_t is derived from the S-A working variable $\tilde{\nu}$. The resulting sequence transforms the incompressible steps into a fully coupled thermophysical system:

- **Solve Momentum:** $\mathbf{F}(\tilde{\nu})\mathbf{v}^* = \mathbf{f}_b$
- **Solve Compressible Pressure:** $\left(\frac{1}{RT}\frac{1}{\Delta t}\mathbf{I} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T\right)p^* = -\mathbf{B}\mathbf{v}^* - \frac{\partial \rho}{\partial t}$
- **Update Velocity and Density:**
$$\mathbf{v} = \mathbf{v}^* - \mathbf{D}^{-1}\mathbf{B}^T p^*, \quad \rho = \frac{p^*}{RT} \tag{111}$$
- **Solve Energy:** $\mathbf{M}_T T = \mathbf{f}_T$
- **Solve Spalart-Allmaras:** Solve for the turbulence working variable $\tilde{\nu}$:
$$\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{v} \cdot \nabla \tilde{\nu} = \frac{1}{\sigma}\nabla \cdot [(\nu + \tilde{\nu})\nabla \tilde{\nu}] + S_{\tilde{\nu}} \tag{112}$$
- **Update Viscosity:** Compute $\nu_t = \tilde{\nu} f_{v1}$ to update the diffusion terms in $\mathbf{F}$ for the next outer iteration.

This architecture ensures that the non-linear dependencies between the flow field, the thermodynamics, and the turbulence structure are resolved through the outer sub-iterations.

5.1.4 The Compressible SIMPLE Algorithm

The collocated compressible SIMPLE algorithm extends the segregated pressure-velocity coupling to account for variations in density and temperature. This iterative process ensures that the conservation of mass, momentum, and energy, along with the equation of state, are simultaneously satisfied at each time step. The procedure for advancing the solution from time t to $t + \Delta t$ is summarized as follows:

1. **Initialization:** Start with the solution fields from the previous time step ($p^{(n)}, \mathbf{v}^{(n)}, \rho^{(n)}, T^{(n)}, \dot{m}^{(n)}$) as the initial guess for the new time level.
2. **Momentum Solve:** Assemble and solve the momentum equations to obtain the intermediate velocity field $\mathbf{v}^*$.
3. **Thermophysical Update:** Compute the intermediate density ρ^* using the equation of state and the intermediate mass flow rate $\dot{m}_f^*$ using Rhie-Chow interpolation to prevent checkerboard oscillations.

4. **Pressure Correction:** Assemble and solve the pressure correction equation for p'. This equation is derived by substituting the linearized momentum and density relations into the continuity equation to enforce mass conservation.

5. **Field Correction:** Correct the face mass flow rate ($\dot{m}_f''$), velocity ($\mathbf{v}''$), density (ρ''), and pressure (p^*) using the calculated pressure correction p'.

6. **Energy Solve:** Assemble and solve the energy equation for the temperature field T.

7. **Convergence Check:**

   - Set the updated values as the new guess: $\dot{m}_f^{(n)} = \dot{m}_f'', \mathbf{v}^{(n)} = \mathbf{v}'', \rho^{(n)} = \rho'', p^{(n)} = p^*$.
   - If the solution has not converged, repeat from Step 2.
   - If converged, advance in time: $t = t + \Delta t$ and repeat until the final simulation time is exceeded.

[Needs work.]

Under-Relaxation In the iterative solution of non-linear equations, large variations in variables between iterations can lead to numerical instability or divergence. To mitigate this, *under-relaxation* is employed to limit the change in a variable during a single iteration, effectively slowing down the changes to promote convergence and stabilize the solver. There are a few numerical approaches to under-relaxation where two ways are described as implicit and explicit. *Implicit* under-relaxation, often referred to as field relaxation, is applied to the solution vector ϕ after the system of equations has been solved. If $\phi^{(n-1)}$ is the value from the previous iteration and ϕ^* is the newly predicted value from the current matrix solve, the actual value $\phi^{(n)}$ passed to the next iteration is computed as:

$$\phi^{(n)} = \phi^{(n-1)} + \lambda \left(\phi^* - \phi^{(n-1)} \right) \tag{113}$$

where λ is the under-relaxation factor ($0 < \lambda < 1$). By rearranging this expression, the update can be viewed as a weighted average:

$$\phi^{(n)} = \lambda \phi^* + (1 - \lambda)\phi^{(n-1)} \tag{114}$$

In contrast, *explicit* under-relaxation (or equation relaxation) modifies the linearized algebraic equation $a_C\phi_C + \sum a_F\phi_F = b_C$ before it is passed to the linear solver. The central coefficient a_C and the source term b_C are adjusted as follows:

$$a_C^{new} = \frac{a_C}{\lambda} \quad \text{and} \quad b_C^{new} = b_C + \frac{(1-\lambda)a_C}{\lambda}\phi_C^{(n-1)} \tag{115}$$

This modification increases the magnitude of the diagonal coefficient a_C, thereby enhancing the *diagonal dominance* of the system. This is particularly essential for stabilizing highly non-linear problems, such as the pressure-velocity coupling in transonic flows. In the current simulation, low relaxation factors are typically applied to pressure ($\lambda_p \approx 0.3$) and momentum ($\lambda_u \approx 0.7$) to ensure a smooth path toward steady-state convergence.

Residuals and Solution Convergence In any iterative CFD process, it is critical to determine when the solution is accurate enough to be considered "converged". Convergence is quantified by the *residual*, which measures the error in the balance equation of the discretized system. For a specific cell C, the element residual error Res_C^{ϕ} is defined as the difference between the source term b_C and the algebraic sum of the current cell and its neighbors:

$$Res_C^{\phi} = b_C - \left(a_C\phi_C + \sum_{F\sim NB(C)} a_F\phi_F \right) \tag{116}$$

Ideally, as the solution reaches a converged state and the governing equations are satisfied, the value of Res_C^{ϕ} should approach zero. To evaluate the convergence across the entire computational domain, several indicators are utilized:

- **Absolute Residual (R_C^{ϕ}):** Because the sign of the error is immaterial, the absolute value is used to decide if the solution is converging or diverging.

$$R_C^{\phi} = \left| b_C - \left(a_C\phi_C + \sum a_F\phi_F \right) \right| \tag{117}$$

- **Maximum Residual ($R_{C,max}^{\phi}$):** This tracks the worst-performing cell in the entire mesh. Convergence is assumed when the maximum absolute residual over all cells drops below a small vanishing quantity ε:

$$R_{C,max}^{\phi} = \max_{allcells} (R_C^{\phi}) \leq \varepsilon \tag{118}$$

- **Root-Mean Square Residual ($R_{C,rms}^{\phi}$):** This provides an average representation of the error across the domain, reducing the impact of a single localized "bad" cell:

$$R_{C,rms}^{\phi} = \sqrt{\frac{\sum (R_C^{\phi})^2}{numberofelements}} \tag{119}$$

To make convergence criteria independent of the initial guess or the magnitude of ϕ, the solver often utilizes a *scaled residual*. This is calculated by dividing the local absolute error by a global normalization factor, typically the maximum flux across the domain:

$$R^{\phi}_{C,scaled} = \frac{|a_C \phi_C + \sum a_F \phi_F - b_C|}{\max_{allcells} |a_C \phi_C|} \tag{120}$$

It is common practice to require scaled residuals to drop to the order of 10^{-3} to 10^{-5} before declaring convergence. When the iterative process is stabilized using *under-relaxation*, the algebraic equation is modified to include the value from the previous iteration ϕ^*_C. This effectively changes the "sums" and "sources" being balanced. The modified residual balance for an under-relaxed system becomes:

$$\frac{a_C}{\lambda^{\phi}} \phi_C + \sum a_F \phi_F = b_C + \frac{(1-\lambda^{\phi}) a_C}{\lambda^{\phi}} \phi^*_C \tag{121}$$

In this state, the residual measures how well the current solution ϕ_C satisfies this "braked" version of the physics. At full convergence, ϕ_C and ϕ^*_C become equal, and the under-relaxation terms cancel out, ensuring the final converged solution satisfies the original, unmodified physical equations.

5.1.5 Numerical Solution

Reynolds Number Calculation

The Reynolds number (Re) is a dimensionless quantity used to predict flow patterns by relating inertial forces to viscous forces. It is defined as:

$$Re = \frac{\rho U L}{\mu} \tag{122}$$

For the current simulation at Mach 0.7, assuming a characteristic chord length of $L = 0.1\, m$ (10 cm), the calculation is as follows:

$$Re = \frac{1.225 \times 238.2 \times 0.1}{1.789 \times 10^{-5}} \tag{123}$$

$$Re \approx 1.63 \times 10^{6} \tag{124}$$

This high Reynolds number confirms that the flow over the fin is in the fully turbulent regime, which validates the selection of the Spalart-Allmaras turbulence model.

5.1.6 Convergence

Convergence Criteria

6 Verificaiton, Validation, and Uncertainty Quantification (VVUQ)

6.1 Verification

6.1.1 Code verification

OpenFOAM is verified code.

6.1.2 Solution verification

6.2 Physical Model Validation

6.3 Uncertainty Quantification

7 Shape Optimization

7.1 Global Optimization

7.1.1 Evolutionary-Genetic Algorithms

7.2 Local Optimization

7.2.1 The Adjoint method

8 Sensitivity Analysis

8.1 Global Sensitivity Analysis

8.1.1 Variance-Based Methodology utilizing Sobol Indices

8.2 Local Sensitivity Analysis

Acknowledgements

The author would like to express sincere gratitude to the following individuals and resources for their support throughout this investigation:

- **Dr. Foluso Ladeinde (Stony Brook University):** As the primary research advisor, Dr. Ladeinde provided the high-level roadmap for conducting rigorous CFD research. His feedback was instrumental in transitioning the author's approach from "black-box" CFD applications to a fundamental, first-principles modeling of the rocket fin. Through his mentorship, the author gained critical computational mechanics research experience and received vital access to the *Intruder* High-Performance Computing (HPC) cluster, which was essential for the computing resource required.

- **Spencer Zimmerman (Stony Brook University):** For providing the pedagogical foundation essential to this work. His instruction in undergraduate fluid mechanics established the theoretical scaffolding required to interpret the complex flow phenomena observed in this investigation. His dedication to fundamental principles provided the author with the necessary analytical tools to bridge the gap between classroom theory and high-fidelity computational analysis.

- **H.K. Versteeg and W. Malalasekera:** As authors of *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, their work served as the primary pedagogical entry point for the governing equations and turbulence modeling used in this report. Hendrik Versteeg (Loughborough University) and Weeratunge Malalasekera (Imperial College PhD) are renowned for bridging the gap between fluid mechanics theory and the practical discretization required for modern computer solvers.

- **F. Moukalled, L. Mangani, and M. Darwish:** Their text, *The Finite Volume Method in Computational Fluid Dynamics*, provided the advanced mathematical rigor necessary for understanding unstructured mesh formulations and pressure-based algorithms. Dr. Fadl Moukalled (American University of Beirut) and his colleagues are leaders in the development of all-speed flow solvers, and their insights into matrix assembly and numerical stability were critical for the high-fidelity modeling of the rocket fin.

- **Steven Brunton (University of Washington):** Served as a primary inspiration in the application of data-driven methods and applied mathematics to Fluid Mechanics/Dynamical Systems research. His work provided the clarity necessary to bridge the gap between abstract mathematical theory and the physical intuition required to model turbulence and dynamical systems.

- **Aidan Wimshurst (Fluid Mechanics 101):** Acted as a central resource for the technical implementation of Computational Fluid Dynamics. His focus on the foundational mechanics of the Finite Volume Method and the internal logic of modern solvers served as the primary inspiration for the numerical setup used in this investigation.

- **Dominique Thévenin (Otto-von-Guericke University Magdeburg):** A Professor at Otto-von-Guericke University Magdeburg and a global authority on the integration of Computational Fluid Dynamics with numerical optimization (CFD-O). His work is foundational in bridging the gap between raw flow simulation and automated design improvement. By utilizing advanced adjoint methods and genetic algorithms, Thévenin's methodologies allow for the iterative refinement of aerodynamic bodies, providing the framework necessary to move beyond trial-and-error design toward a mathematically optimized rocket fin configuration.

- **Andrea Saltelli:** A former lead at the European Commission and a primary architect of modern Global Sensitivity Analysis (GSA). Saltelli's development of variance-based methods, such as Sobol indices, provides the mathematical framework used to identify which geometric parameters most significantly influence the fluid-structure interaction of the rocket fin.

References

[1] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Prentice Hall, 2nd ed., 2007.

[2] Y. A. Çengel and J. M. Cimbala, *Fluid Mechanics: Fundamentals and Applications*. McGraw-Hill Higher Education, 2006.

[3] National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, and United States Air Force, "U.s. standard atmosphere, 1976," Tech. Rep. NASA-TM-X-74335, National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, United States Air Force, Washington, D.C., 1976.

[4] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*, vol. 113 of *Fluid Mechanics and Its Applications*. Springer International Publishing, 2016.

[5] P. R. Spalart and S. R. Allmaras, "A one-equation turbulence model for aerodynamic flows," *La Recherche Aérospatiale*, no. 1, pp. 5–21, 1994.

[6] F. Stern, R. V. Wilson, H. W. Coleman, and E. G. Paterson, "Verification and validation of cfd simulations," Tech. Rep. IIHR Report No. 407, Iowa Institute of Hydraulic Research, 1999.

[7] A. Wimshurst, "Fluid mechanics 101: Computational fluid dynamics tutorials." `https://www.fluidmechanics101.com/`, 2025.

[8] S. L. Brunton, "Department of mechanical engineering faculty profile." `https://www.me.washington.edu/facultyfinder/steve-brunton`, 2025. University of Washington.

[9] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, Ltd, 2008.

[10] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. John Wiley & Sons, Ltd, 2004.

[11] D. Thévenin and G. Janiga, eds., *Optimization and Computational Fluid Dynamics*. Springer-Verlag Berlin Heidelberg, 2008.

[12] J. E. V. Peter and R. P. Dwight, "Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches," *Computers & Fluids*, vol. 39, no. 3, pp. 373–391, 2010.

[13] X. Wu, W. Zhang, and S. Song, "Uncertainty quantification and sensitivity analysis of transonic aerodynamics with geometric uncertainty," *International Journal of Aerospace Engineering*, vol. 2017, 2017.

[14] S. Ju, C. Yan, X. Wang, Y. Qin, and Z. Ye, "Sensitivity analysis of geometric parameters upon the aerothermodynamic performances of mars entry vehicle," *International Journal of Heat and Mass Transfer*, vol. 120, pp. 597–607, 2018.