Politecnico di Torino

Master's Degree Thesis

Master's Degree course in
INGEGNERIA AEROSPAZIALE
Sistemi Propulsivi

Computational Study on the Instability of Supersonic Jet Impingement on a Flat Surface

Supervisor
Prof. Emanuele Martelli

Candidate
Matteo Obialero

A.A. 2024/2025

To everyone I've learned something from.

Abstract

The lift-off phase of a launch vehicle is characterized by the generation of intense acoustic waves due to the interaction between the supersonic exhaust jet and the launch pad deflector.
These pressure fluctuations pose a critical challenge, as they can induce structural fatigue on the launcher and surrounding facilities.
Understanding the underlying mechanisms governing the acoustic field in this scenario is essential for mitigating the risks associated with excessive noise levels.
In this study, a numerical approach is employed to simulate the supersonic exhaust flow of a Vega-type launcher and its interaction with the launch pad, assuming axisymmetric, inviscid flow.
The resulting acoustic field is analyzed using both frequency-domain and time-frequency techniques, providing insight into the unsteady pressure fluctuations.
The evolution of dominant frequencies is examined over time, allowing for a deeper understanding of the flow dynamics during lift-off.
The results highlight key features of the acoustic environment and provide useful information for future studies on noise mitigation strategies in rocket launches.

Contents

Chapter 1

Rocket Propulsion

1.1 Rocket Classification

The concept of propulsion, derived from the Latin *propellere* (meaning "to push forward"), broadly encompasses any method or system that initiates or modifies the motion of an object by counteracting resistive forces. Rockets, a distinct category of propulsion systems, generate thrust by expelling mass at high velocity through a nozzle, in accordance with Newton's Third Law of Motion.

Rocket propulsion systems can be classified based on the method used to produce thrust:

- **Thermal Rockets**: These systems generate thrust by heating a propellant, which is then expanded and accelerated through a nozzle. The heat source may vary, leading to subcategories such as chemical, nuclear, and solar thermal rockets.
- **Electric Propulsion**: Unlike thermal rockets, electric propulsion systems use electromagnetic fields to accelerate charged particles or neutral atoms, producing thrust with high efficiency but generally low thrust levels. This category includes ion thrusters, Hall-effect thrusters, and magnetoplasmadynamic (MPD) thrusters.
- **Nuclear Propulsion**: A conceptually powerful but complex method in which nuclear fission or fusion reactions provide the energy to heat and expel a propellant, offering significantly higher efficiency compared to chemical rockets.

Within the category of thermal rockets, further distinctions are made based on the energy source used to heat the propellant:

- **Chemical Rockets**: The most common propulsion method, where the stored chemical energy of a fuel-oxidizer mixture is released through combustion, producing high-temperature, high-pressure gases that expand through a nozzle to generate thrust.

- **Electrothermal Thrusters**: These systems rely on electrical energy to heat the propellant, either through resistive heating or arc discharge, before expanding it through a nozzle. While offering higher efficiency than traditional chemical rockets, they typically produce lower thrust.

- **Nuclear Thermal Rockets (NTRs)**: Utilize nuclear reactors to heat a working fluid (such as hydrogen) without combustion, providing superior efficiency in terms of specific impulse.

- Solar Thermal Rockets: Still largely experimental, these systems use concentrated solar radiation to heat and expand the propellant. While theoretically efficient, they are limited by solar energy availability and power density.

Another essential classification criterion is the thermal management system, which prevents critical engine components, particularly the nozzle, from overheating under extreme thermal loads:

- **Regenerative Cooling**: A widely used method in liquid rocket engines (LREs), where cryogenic or liquid propellants circulate through cooling channels embedded within the nozzle walls before being injected into the combustion chamber. This process enhances engine efficiency by preheating the fuel while keeping the nozzle from reaching destructive temperatures.

- **Ablative Cooling**: This method relies on sacrificial materials, such as graphite or phenolic composites, that gradually erode under intense heat, carrying away excess thermal energy. It is particularly useful for solid rocket motors (SRMs) and high-performance nozzles, especially in the throat region where thermal stress is at its peak.

- **Film Cooling**: A technique where a thin layer of cooler propellant is injected along the inner nozzle walls, forming a protective gas film that insulates the structure from the high-temperature exhaust. This method is often used in conjunction with regenerative cooling.

- **Heat Sink Cooling**: In this approach, the nozzle is made from materials with exceptionally high melting points, such as tungsten or niobium alloys, which can absorb and dissipate heat through conduction and radiation. While effective for short-duration burns and experimental engines, this method is generally impractical for sustained propulsion due to material limitations.

Rocket propulsion continues to evolve, with ongoing advancements in hybrid propulsion, aerospike nozzle technologies, and novel cooling strategies aimed at increasing efficiency and reliability. The choice of propulsion system depends on mission requirements, including thrust-to-weight ratio, efficiency (specific impulse), reusability, and environmental conditions encountered during flight.[1]

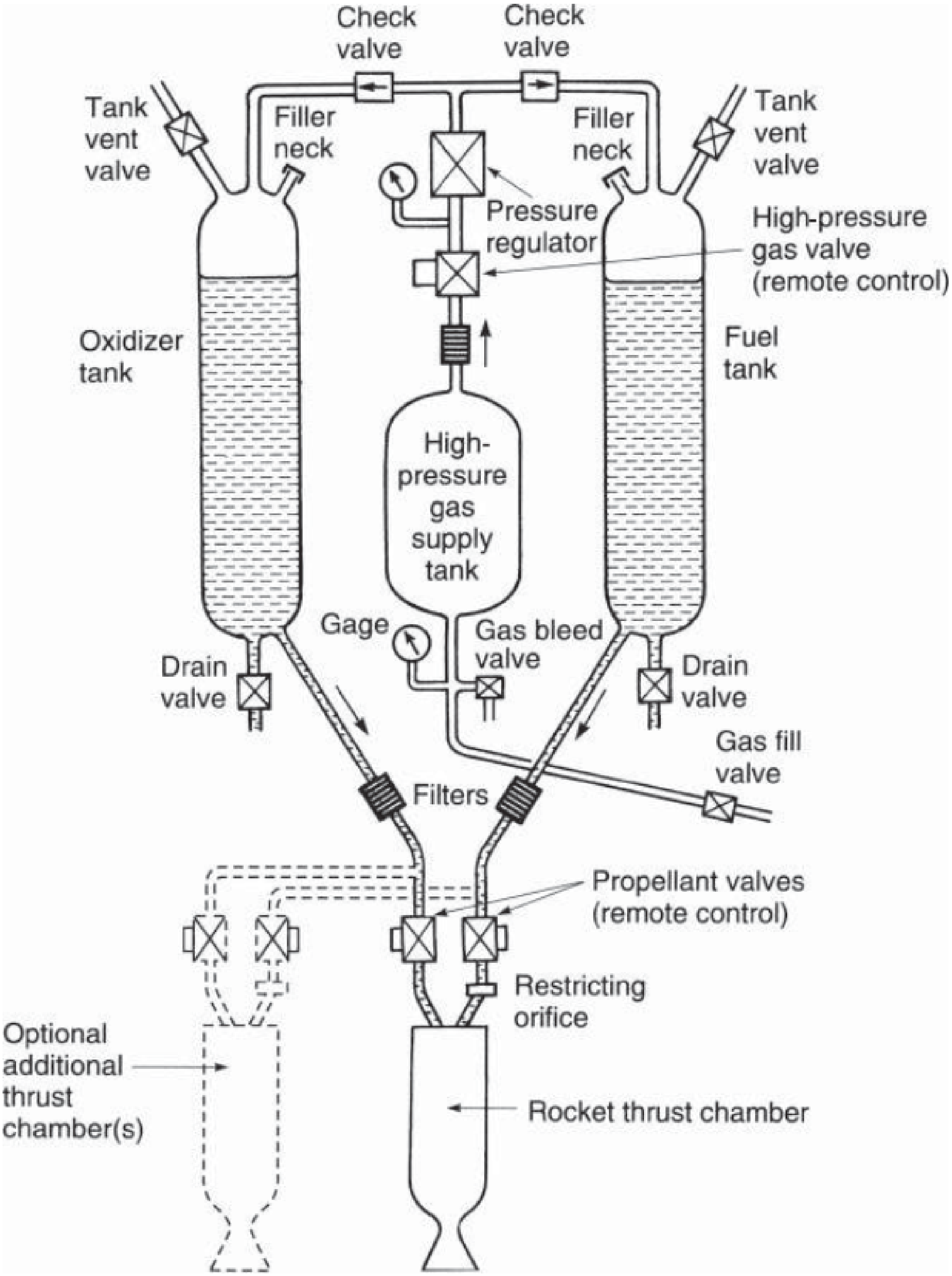


Figure 1.1: Schematic flow diagram of a liquid propellant rocket engine with a gas pressure feed system.-*from Sutton G.P., Biblartz O., "Rocket Propulsion Elements"*

1.2 Chemical Rocket Propulsion

Chemical propulsion systems rely on the conversion of stored chemical energy into thermal energy through combustion. This process occurs within a high-pressure combustion chamber, where a fuel and an oxidizer react exothermically, producing hot, high-speed gases that are expelled through a nozzle to generate thrust. The high temperatures and pressures involved necessitate careful design considerations to ensure structural integrity and efficiency.

Chemical rockets can be categorized based on the state of their propellants:

1.2.1 Liquid Rocket Engines (LREs)

Liquid rocket engines use propellants in liquid form, which are stored in separate tanks and fed into the combustion chamber through either a pressure-fed or pump-fed system. These systems are widely used in space applications due to their high specific impulse and the ability to regulate thrust by controlling the flow rate of the propellants.

- **Pressure-fed systems**: Utilize a pressurized gas (often helium or nitrogen) to force propellants into the combustion chamber. These systems are relatively simple and reliable but require robust, heavy tanks to withstand the high pressures.

- **Pump-fed systems**: Use turbopumps to increase the pressure of the propellants before injection. This allows for lighter tank designs and higher combustion chamber pressures, leading to improved performance. However, turbopumps introduce additional complexity and potential failure points.

1.2.2 Solid Rocket Motors (SRMs)

In solid propulsion systems, the propellant is cast or extruded into the motor casing itself, forming what is known as the propellant grain. Since the oxidizer and fuel are pre-mixed, solid rockets are inherently simpler than liquid systems and do not require complex plumbing or pumping mechanisms.

- **Advantages**: Solid rockets provide high thrust and rapid ignition, making them ideal for applications such as boosters in space launch vehicles and military missiles. They are also more mechanically robust and easier to store for long durations.

- **Disadvantages**: Once ignited, solid rocket motors cannot be throttled or shut down, leading to limited flexibility in mission profiles. Additionally, controlling thrust vectoring and burn rate requires precise grain shaping and composition adjustments.

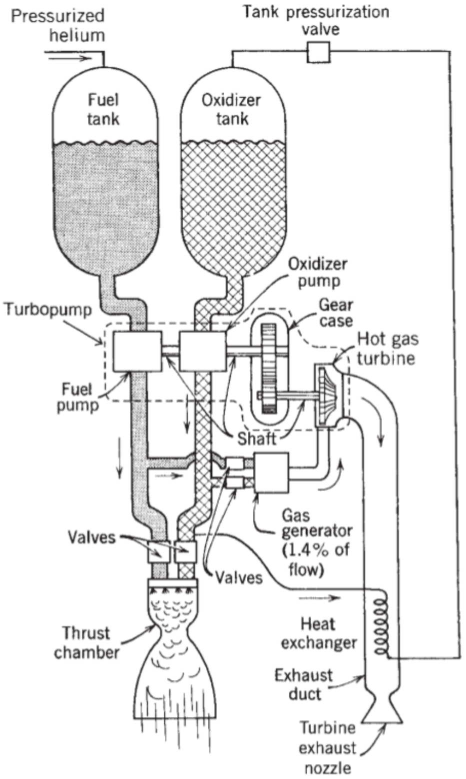


Figure 1.2: Simplified schematic diagram of a liquid propellant rocket engine with one type of turbopump feed system and a separate gas generator.*-from Sutton G.P., Biblartz O., "Rocket Propulsion Elements"*

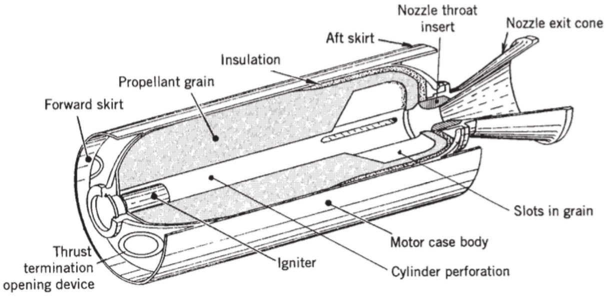


Figure 1.3: Simplified perspective three-quarter section view of a typical solid propellant rocket motor.*-from Sutton G.P., Biblartz O., "Rocket Propulsion Elements"*

1.2.3 Hybrid Rocket Engines (HREs)

Hybrid propulsion systems combine elements of both liquid and solid rockets by using a liquid oxidizer and a solid fuel. This configuration allows for some of the controllability of liquid systems while maintaining the relative simplicity of solid motors.

- **Advantages**: Hybrid rockets offer improved safety compared to fully liquid or solid systems, as the fuel and oxidizer are stored separately, reducing explosion risks. They can also be throttled and restarted in some designs.
- **Challenges**: The interaction between the liquid oxidizer and solid fuel introduces combustion instability challenges, requiring advanced engineering solutions.

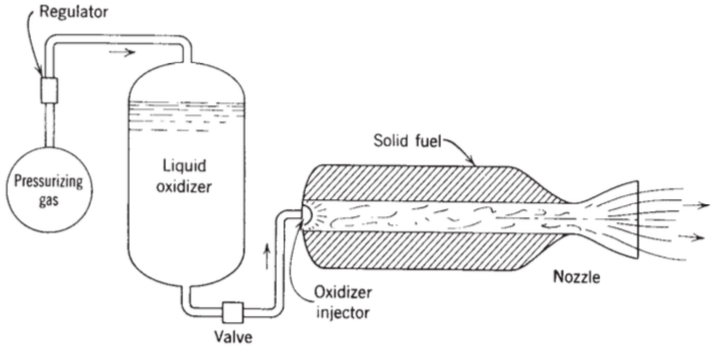


Figure 1.4: Schematic diagram of a typical hybrid rocket engine.*-from Sutton G.P., Biblartz O., "Rocket Propulsion Elements"*

Overall, chemical propulsion remains the dominant technology in modern space exploration due to its high energy density and proven reliability. Advances in materials, cooling techniques, and propellant formulations continue to enhance performance and efficiency, ensuring its continued relevance in future aerospace applications.

1.3 Fundamental Relations in Rocket Propulsion

Rocket propulsion is governed by fundamental physical principles, primarily derived from Newtonian mechanics, thermodynamics, and fluid dynamics. The performance of a rocket engine is characterized by key parameters such as thrust, specific impulse, and characteristic velocity, all of which are crucial for the design and optimization of space propulsion systems.

1.3.1 Thrust and Thrust Coefficient

Assuming a fixed geometry and steady-state nozzle operation, along with uniform atmospheric pressure, the total thrust F generated by a rocket engine can be expressed as:

$$F = \dot{m} v_e + (P_e - P_a) A_e \tag{1.1}$$

where:

- $\dot{m}$ is the mass flow rate of the propellant (kg/s),
- v_e is the effective exhaust velocity (m/s),
- P_e is the pressure at the nozzle exit (Pa),
- P_a is the ambient pressure (Pa),
- A_e is the nozzle exit area (m^2).

The first term represents the momentum thrust, while the second term accounts for the pressure difference at the nozzle exit, which becomes particularly relevant when operating at different altitudes.

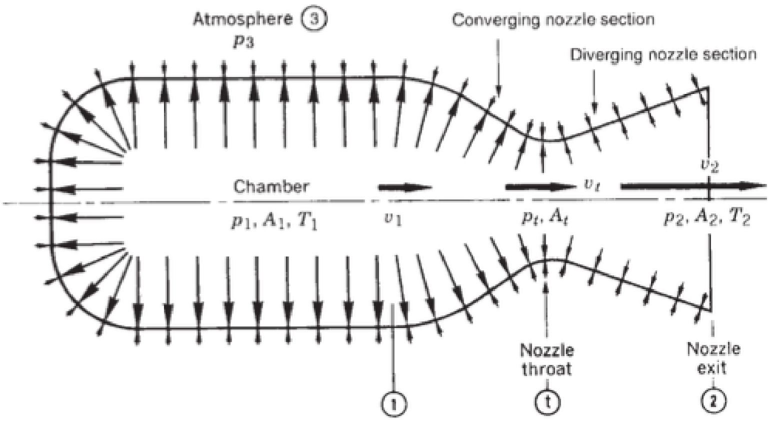


Figure 1.5: Pressure distribution for a simplified thrust chamber. *-from Sutton G.P., Biblartz O., "Rocket Propulsion Elements"*

Figure 1.5 illustrates the pressure distribution both inside and outside a simplified rocket, where the length of the arrows represents the relative pressure magnitude. The axial thrust can be determined by integrating the pressure components acting on the surfaces perpendicular to the nozzle axis.

In the absence of ambient pressure ($p_a = 0$), the thrust reaches its maximum value, which can be expressed as:

$$F = \dot{m}u_e + p_e A_e \tag{1.2}$$

To simplify the expression for thrust in terms of mass flow rate $\dot{m}$, the concept of an *equivalent exhaust velocity* is introduced:

$$u_{eq} = \frac{F}{\dot{m}} = u_e + \frac{(p_e - p_a)A_e}{\dot{m}} \tag{1.3}$$

The mass flux at any point in the nozzle can be derived using the continuity equation, the definition of the speed of sound $a = \sqrt{\gamma RT}$, and the Mach number $M = u/a$, along with isentropic flow relations:

$$\dot{m} = \sqrt{\gamma}\left(1 + \frac{\gamma - 1}{2}M^2\right)^{-\frac{\gamma+1}{2(\gamma-1)}} \frac{p_c A M}{\sqrt{RT_c}} \tag{1.4}$$

For the specific case of the throat section, where $A = A_t$ and $M = 1$, the equation simplifies to:

$$\dot{m} = \sqrt{\gamma}\left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} \frac{p_c A_t}{\sqrt{RT_c}} = \Gamma \frac{p_c A_t}{\sqrt{RT_c}} \tag{1.5}$$

Since mass flow rate $\dot{m}$ remains conserved throughout the nozzle, equating the expressions for different sections leads to the **area ratio relation**, which defines the ratio of the cross-sectional area A/A_t required to achieve a given Mach number M:

$$\frac{A}{A_t} = \frac{1}{M}\left(1 + \frac{\gamma - 1}{2}M^2\right)^{\frac{\gamma+1}{2(\gamma-1)}} \tag{1.6}$$

This relationship allows rewriting the thrust equation in terms of the chamber pressure p_c and nozzle expansion parameters:

$$F = p_c A_t \left[\sqrt{\frac{2\gamma}{\gamma - 1}}\Gamma\left(1 - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right) + \frac{A}{A_t}\left(\frac{p_e}{p_c} - \frac{p_a}{p_c}\right)\right] \tag{1.7}$$

This equation makes it clearer that thrust depends on the chamber pressure p_c and throat area A_t, but not explicitly on the combustion temperature.

From this relation, the **thrust coefficient** C_F can be defined, which describes the effectiveness of the nozzle in converting chamber pressure into thrust:

$$C_F = \frac{F}{p_c A_t} = \sqrt{\frac{2\gamma}{\gamma - 1}} \Gamma \left(1 - \left(\frac{p_e}{p_c}\right)^{\frac{\gamma-1}{\gamma}}\right) + \frac{A}{A_t} \left(\frac{p_e}{p_c} - \frac{p_a}{p_c}\right) \tag{1.8}$$

This expression highlights the dependence on the flow model (γ), the nozzle geometry (A/A_t), and the effect of external pressure (p_a/p_c), as shown by figure 1.6 below. By comparing the theoretical thrust coefficient (C_F) with experimental values, it is possible to assess the contribution of the nozzle's divergent section to the overall thrust performance.

1.3.2 Nozzle Expansion and Efficiency

Rocket nozzles play a crucial role in the conversion of thermal energy into directed kinetic energy. The efficiency of this conversion is largely determined by the expansion ratio:

$$\epsilon = \frac{A_e}{A_t} \tag{1.9}$$

where A_t is the throat area and A_e is the nozzle exit area. The optimal expansion ratio depends on the operating altitude. At sea level, nozzles with a lower expansion ratio are preferred, whereas in vacuum, larger expansion ratios maximize performance.

The relationship between the Mach number at the nozzle exit and the expansion ratio is given by:

$$\frac{A}{A_t} = \frac{1}{M} \left[\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2\right)\right]^{\frac{\gamma+1}{2(\gamma-1)}} \tag{1.10}$$

where γ is the ratio of specific heats. This equation is crucial for designing nozzles that efficiently expand the exhaust gases to achieve maximum thrust.

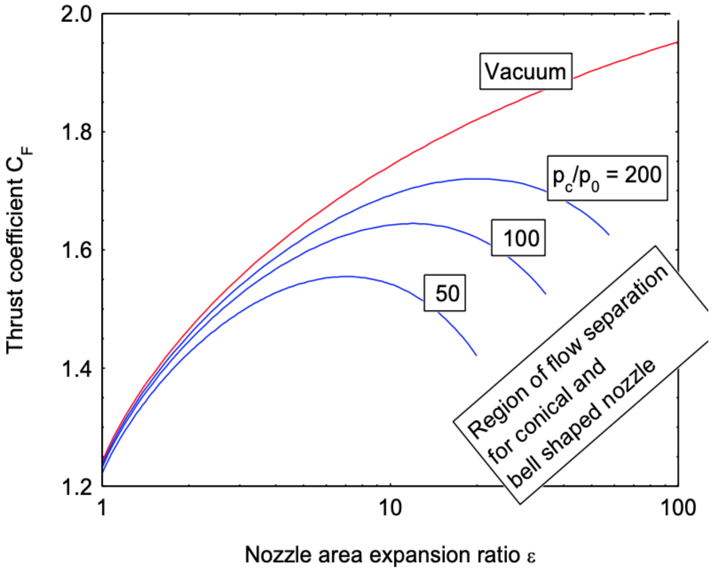


Figure 1.6: Diagram of (C_F) as a function of (A/A_t) at different values of pressure ratios *.-from Ferrero A., Dispense del corso di Endoreattori*

1.3.3 Specific Impulse and Characteristic Velocity

One of the most important performance parameters of a rocket engine is the specific impulse (I_{sp}), which represents the thrust generated per unit of propellant weight flow rate. It is mathematically defined as:

$$I_{sp} = \frac{\int_0^t F dt}{g_0 \int_0^t \dot{m} dt} = \frac{I}{M g_0} \tag{1.11}$$

where:

I is the total impulse of the rocket engine, Mg_0 represents the total weight flux, with g_0 being the standard acceleration due to gravity, t corresponds to the total burn time of the rocket. Rearranging this equation using the equivalent exhaust velocity (u_{eq}), we obtain:

$$I_{sp} = \frac{u_{eq}}{g_0} \tag{1.12}$$

This expression indicates that the specific impulse is directly proportional to the equivalent exhaust velocity. The presence of gravity in the denominator allows I_{sp} to be conveniently expressed in seconds, making it a widely used metric for evaluating propulsion system efficiency.

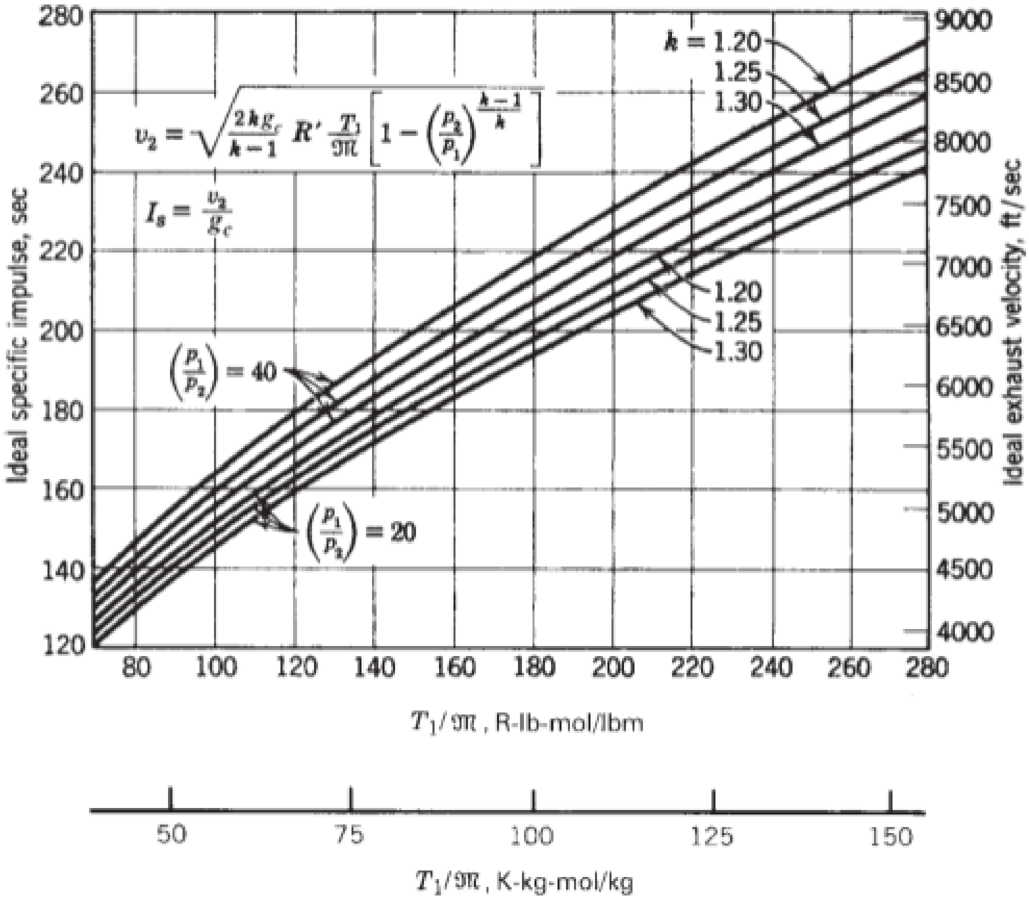


Figure 1.7: Relationship Between Ideal Specific Impulse and Exhaust Velocity. *-from Sutton G.P., Biblartz O., "Rocket Propulsion Elements"*

This figure illustrates the relationship between ideal specific impulse (I_{sp}) and ideal exhaust velocity (v_2) as a function of the temperature ratio ($T_1/\bar{T}$), where T_1 represents the combustion temperature and $\bar{T}$ is a reference temperature. The graph includes multiple curves corresponding to different values of the specific heat ratio (k), ranging from 1.20 to 1.30.

Additionally, the effect of the pressure ratio (p_2/p_1) is shown for values of 20 and 40, which influence the resulting exhaust velocity and specific impulse. The mathematical expressions for exhaust velocity and specific impulse are provided at the top left of the figure, showing their dependence on thermodynamic properties such as the specific gas constant (R) and heat capacity ratio (k).

This chart serves as a useful reference for estimating propulsion performance based on combustion conditions, nozzle expansion ratios, and thermodynamic properties of the working fluid.

Another important parameter is the characteristic velocity, c^*, which describes the internal performance of the rocket engine independent of the nozzle efficiency:

$$c^* = \frac{P_c A_t}{\dot{m}} \tag{1.13}$$

It is possible to express the specific impulse in terms of c^* and C_F:

$$I_{sp} = \frac{C_F c^*}{g_0} \tag{1.14}$$

This relation is particularly useful for separating the effects of combustion efficiency (c^*) from nozzle expansion efficiency (C_F).

1.3.4 Tsiolkovsky Rocket Equation

The fundamental equation governing the velocity change (Δv) of a rocket is given by the Tsiolkovsky rocket equation:

$$\Delta v = v_e \ln\left(\frac{m_0}{m_f}\right) \tag{1.15}$$

where:

- m_0 is the initial mass of the rocket (kg),
- m_f is the final mass after propellant consumption (kg),
- v_e is the effective exhaust velocity (m/s).

This equation highlights the importance of having a high mass ratio (m_0/m_f) and an efficient propulsion system with a high exhaust velocity to achieve significant velocity changes.

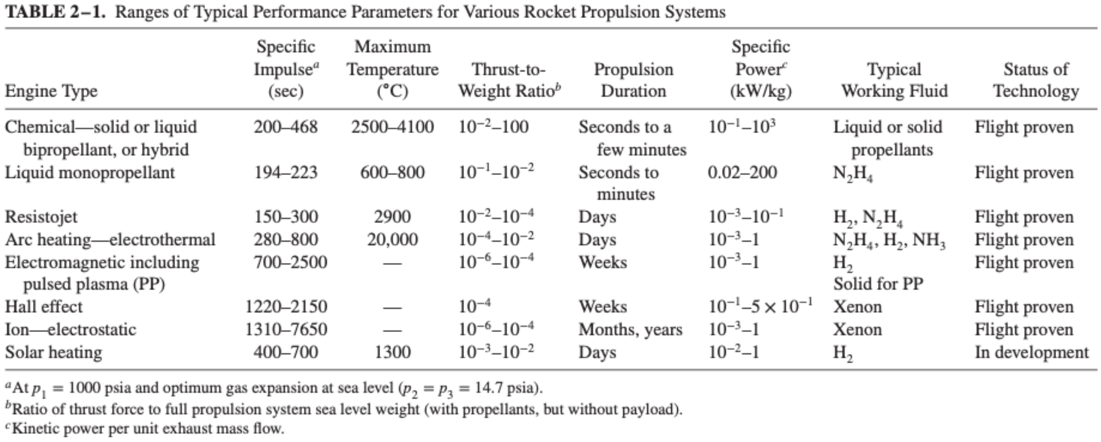


TABLE 2–1. Ranges of Typical Performance Parameters for Various Rocket Propulsion Systems

Engine Type	Specific Impulse[a] (sec)	Maximum Temperature (°C)	Thrust-to-Weight Ratio[b]	Propulsion Duration	Specific Power[c] (kW/kg)	Typical Working Fluid	Status of Technology
Chemical—solid or liquid bipropellant, or hybrid	200–468	2500–4100	10^{-2}–100	Seconds to a few minutes	10^{-1}–10^{3}	Liquid or solid propellants	Flight proven
Liquid monopropellant	194–223	600–800	10^{-1}–10^{-2}	Seconds to minutes	0.02–200	N_2H_4	Flight proven
Resistojet	150–300	2900	10^{-2}–10^{-4}	Days	10^{-3}–10^{-1}	H_2, N_2H_4	Flight proven
Arc heating—electrothermal	280–800	20,000	10^{-4}–10^{-2}	Days	10^{-3}–1	N_2H_4, H_2, NH_3	Flight proven
Electromagnetic including pulsed plasma (PP)	700–2500	—	10^{-6}–10^{-4}	Weeks	10^{-3}–1	H_2 Solid for PP	Flight proven
Hall effect	1220–2150	—	10^{-4}	Weeks	10^{-1}–5×10^{-1}	Xenon	Flight proven
Ion—electrostatic	1310–7650	—	10^{-6}–10^{-4}	Months, years	10^{-3}–1	Xenon	Flight proven
Solar heating	400–700	1300	10^{-3}–10^{-2}	Days	10^{-2}–1	H_2	In development

[a] At p_1 = 1000 psia and optimum gas expansion at sea level ($p_2 = p_3$ = 14.7 psia).
[b] Ratio of thrust force to full propulsion system sea level weight (with propellants, but without payload).
[c] Kinetic power per unit exhaust mass flow.

Figure 1.8: Ranges of Typical Performance Parameters for Various Rocket Propulsion Systems. *-from Sutton G.P., Biblartz O., "Rocket Propulsion Elements"*

The table in figure 1.8 presents a comparison of typical performance parameters for various rocket propulsion systems, categorizing them by engine type. It highlights key metrics such as specific impulse, maximum operating temperature, thrust-to-weight ratio, specific power, and the typical working fluids used in each system. Additionally, the propulsion duration is listed, providing insight into how long each type of engine can operate effectively.

Chemical propulsion systems, including solid, liquid bipropellant, and hybrid rockets, exhibit high thrust-to-weight ratios and relatively short operational durations. In contrast, electric and solar propulsion systems, such as ion thrusters and Hall effect thrusters, provide much higher specific impulses but at the cost of significantly lower thrust-to-weight ratios, making them more suitable for long-duration space missions.

The table also includes the status of technology for each propulsion system, indicating whether they are flight-proven or still under development, such as solar thermal propulsion. This comparative overview helps assess the trade-offs between efficiency, power, and operational feasibility in different mission profiles.

Chapter 2

Vettore Europeo di Generazione Avanzata - VEGA

2.1 History and Development

The *Vettore Europeo di Generazione Avanzata* (VEGA) is an expendable launch vehicle developed collaboratively by the Italian Space Agency (ASI) and the European Space Agency (ESA). The program was initiated to provide Europe with an independent capability to launch small to medium-sized payloads into low Earth orbit (LEO), complementing the larger Ariane launch vehicles. The development of VEGA began in the late 1990s, with official approval by ESA in 1998. Contracts for the project's development were signed in early 2003, with Italy funding 65% of the program, while France, Belgium, Switzerland, the Netherlands, Ukraine, Spain, and Sweden contributed the remaining share [2].

VEGA is operated by *Arianespace*, with industrial production led by *Avio*, an Italian aerospace company, through its subsidiary *European Launch Vehicle* (ELV), established in collaboration with ASI. The primary objective of VEGA was to address the growing demand for launching smaller satellites, particularly for scientific and Earth observation missions. Before VEGA, Europe lacked a dedicated launcher for such payloads and relied on foreign launch services or secondary payload slots on larger rockets [2].

Several iterations of the VEGA launcher have been developed:

- **VEGA**: The original configuration, which conducted its maiden flight on 13 February 2012 [2].

- **VEGA-C**: An enhanced version with increased payload capacity and performance, which first flew on 13 July 2022 [3].

- **VEGA-E**: A future iteration currently under development, aiming to further improve performance and reduce launch costs, scheduled for a maiden flight in 2024 [3].

Figure 2.1: VEGA Aeolus launch from Kourou, French Guyana on August 22 2018. *-from https://www.avio.com/it/press-release/vega-dodicesima-missione-consecutiva-con-lancio-aeolus*

2.2 Utilization, Production, and Launch History

VEGA is primarily used for launching small satellites into polar and Sun-synchronous orbits, ideal for Earth observation missions. Its ability to carry multiple payloads and deploy them into different orbits in a single mission makes it a versatile choice for institutional and commercial clients [2].

The rockets are assembled at the *Guiana Space Centre* in Kourou, French Guiana, utilizing the ELV launch complex, which was refurbished specifically for the VEGA program [2].

As of 2024, VEGA has completed 22 launches, with 20 successes and 2 failures. Notable missions include:

- **LARES (LAser RElativity Satellite)**: The maiden flight in 2012 deployed this ASI satellite designed to study Earth's gravitational field [4].
- **Sentinel-2A and Sentinel-2B**: Part of the European Union's Copernicus program, launched in 2015 and 2017, respectively, for land and vegetation monitoring [4].
- **Aeolus**: Launched in 2018, this ESA satellite was the first to directly measure global wind profiles from space [4].

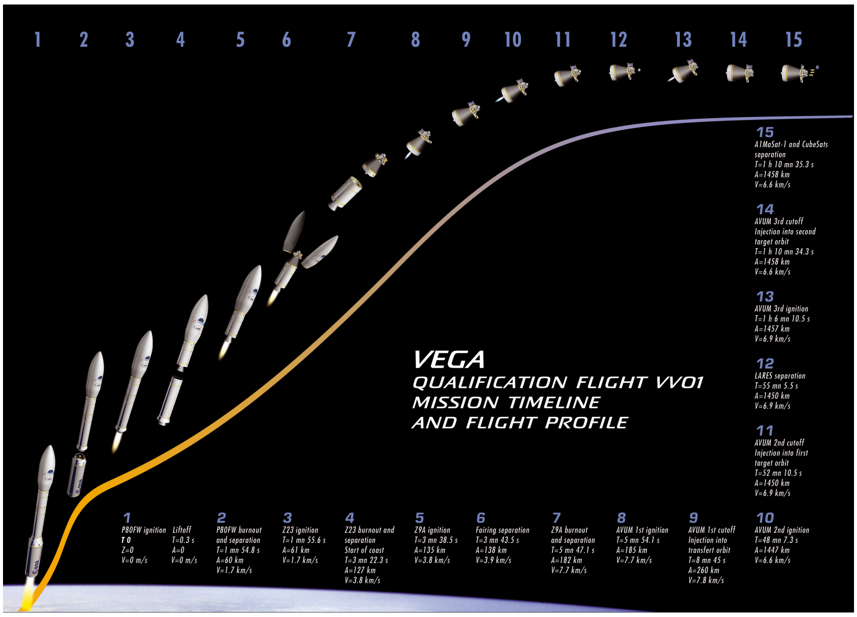


Figure 2.2: Sequence of events during VEGA's first launch, February 2012. *-from https://spaceflightnow.com/vega/vv01/launchtimeline.html*

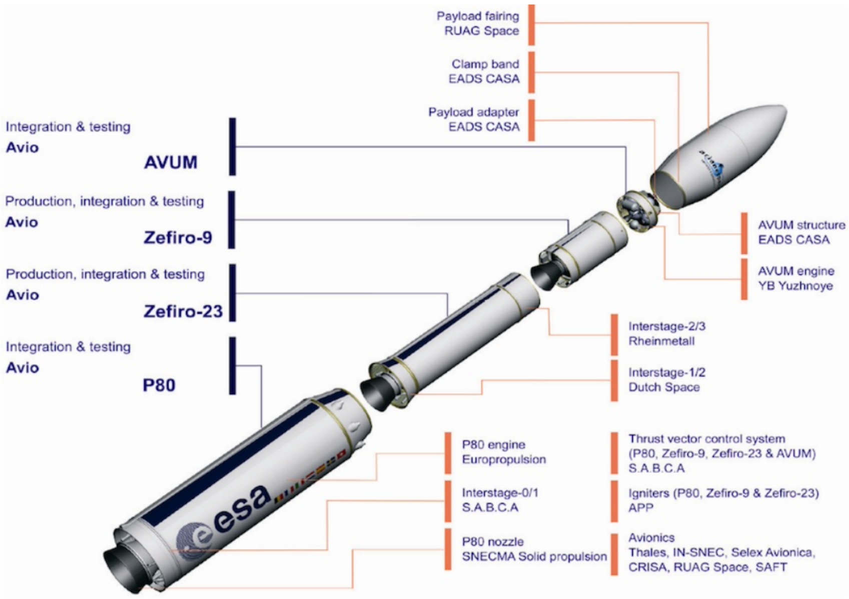


Figure 2.3: Diagram of the Vega rocket. The single stages are highlighted, along with the names of the companies that contributed to their components. *-from esa.int*

2.3 Propulsion Stages and Technical Characteristics

VEGA is a four-stage launch vehicle, with the first three stages utilizing solid rocket motors and the fourth stage employing a liquid-propellant engine for precise orbital insertion and attitude control.

2.3.1 First Stage: P80

The P80 is the first stage of VEGA, responsible for providing the initial thrust necessary to lift the rocket off the launch pad. It is a solid-fuel rocket motor, powered by hydroxyl-terminated polybutadiene (HTPB) solid propellant and generates a sea-level thrust of 2260 kN, with a specific impulse (I_{sp}) of 280 s. It is the largest and most powerful single-body solid rocket motor used in a launch vehicle.
The P80 motor features a carbon-epoxy case and a movable nozzle for thrust vector control, allowing for precise guidance. The P120C, a more advanced version, is currently used as a booster for Ariane 6 and as the main stage for VEGA-C [2].

During a standard mission, the P80 burns for approximately 114 seconds, with stage separation occurring at 53 km altitude. The motor shares design characteristics with the Ariane 5 solid rocket booster (SRB), having an identical length of 10.6 m and a diameter of 3 m. A more advanced variant of this rocket, the P120C, has been developed for use as both a Vega-C main stage and an Ariane 6 booster, providing enhanced performance and efficiency.

Parameter	Value
Propellant Type	HTPB (Solid)
Propellant Mass	88 t
Inert Mass	7.5 t
Average Thrust	2260 kN
Specific Impulse (I_{sp})	280 s
Burn Time	114 s

Figure 2.4: Specifications of the P80 First Stage

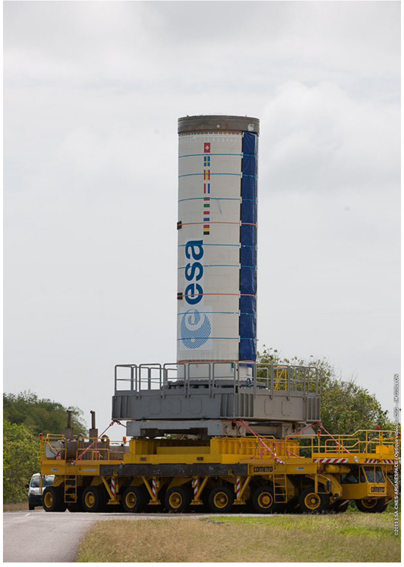


Figure 2.5: P80 transfer. - *from esa.int*

2.3.2 Second Stage: Zefiro 23

The Zefiro 23 (ZEro an FIrst Rocket motor) is the second stage of Vega and is also powered by HTPB solid propellant. It provides an average thrust of 900 kN and operates with a specific impulse of 290 s. The motor is 7.5 meters long and has a diameter of 1.9 meters.

The Zefiro 23 stage contains a 23-ton HTPB 1912 propellant grain, which consists of 19% aluminum powder, 12% HTPB, and 69% ammonium perchlorate, ensuring a burn time of approximately 103 seconds. It was completely developed and integrated by Avio. The name "Zefiro" originates from an earlier iteration of this rocket, which was originally planned as a first and second stage for the San Marco program of the Italian Space Agency (ASI).

The Zefiro 23 was developed by Avio and shares technological advancements with its successor, the Zefiro 40, which is used in VEGA-C [2].

Parameter	Value
Propellant Type	HTPB (Solid)
Propellant Mass	24 t
Inert Mass	1.9 t
Average Thrust	900 kN
Specific Impulse (I_{sp})	290 s
Burn Time	103 s

Figure 2.6: Specifications of the Zefiro 23 Second Stage

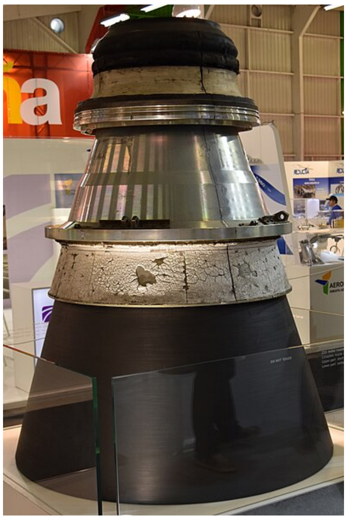


Figure 2.7: Zefiro 23 Nozzle at Paris Air Show 2015. *-from wikipedia.org*

2.3.3 Third Stage: Zefiro 9A

The Zefiro 9A is the third stage of the Vega launcher, providing continued acceleration after the first two stages. This solid-fuel rocket motor is 3.9 meters long and 1.9 meters in diameter. It burns 9 tonnes of HTPB 1912 propellant, delivering an average thrust of 230 kN, with a specific impulse of 290 s.

This stage has a nominal burn time of 117 seconds and plays a crucial role in finalizing the vehicle's trajectory before the payload is inserted into orbit by the AVUM stage.

An enhanced version, the Zefiro 40, is in development for future configurations [2].

Parameter	Value
Propellant Type	HTPB (Solid)
Propellant Mass	9 t
Inert Mass	0.7 t
Average Thrust	230 kN
Specific Impulse (I_{sp})	290 s
Burn Time	117 s

Figure 2.8: Specifications of the Zefiro 9A Third Stage

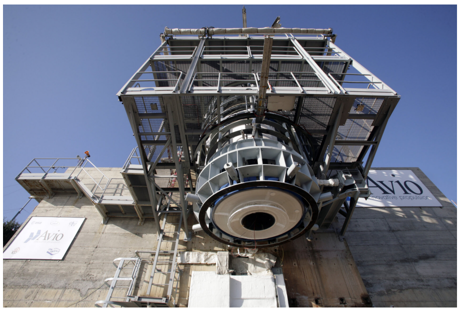


Figure 2.9: Zefiro 9-A on test stand. *-from esa.int*

2.3.4 Fourth Stage: AVUM

The AVUM (Attitude and Vernier Upper Module) is the final stage of the Vega launcher and is responsible for precise orbital injection and attitude control. Unlike the previous solid-propellant stages, AVUM operates using a liquid-propellant system, which provides greater flexibility for mission requirements.

The AVUM consists of two key units:

- **AVUM Propulsion Module (APM)** – which contains the main rocket engine.
- **AVUM Avionics Module (AAM)** – responsible for guidance, navigation, and control.

The APM is equipped with an RD-843 liquid rocket engine, developed by the Ukrainian company Yuzhnoye and manufactured by Yuzhmash. It burns unsymmetrical dimethylhydrazine (UDMH) and nitrogen tetroxide (N_2H_4), producing a thrust of 2.45 kN. The engine can restart multiple times, allowing the precise placement of payloads into different orbits.

The propellants are stored in four separate tanks, and the stage carries approximately 39 kg of hydrazine for monopropellant thrusters, which control the vehicle's roll, attitude, and final payload orientation before separation.

Additionally, the payload is protected by a fairing manufactured in Switzerland by RUAG Space. This 7.88-meter-high payload fairing weighs 540 kg and is designed to safeguard satellites during atmospheric ascent. AVUM can restart multiple times, enabling precise orbit insertion [2].

Parameter	Value
Propellant Type	UDMH + N_2O_4 (Liquid)
Propellant Mass	0.3 t
Inert Mass	0.3 t
Average Thrust	2.3 kN
Specific Impulse (I_{sp})	320 s
Burn Time	Variable (up to 667 s)

Table 2.1: Specifications of the AVUM Stage

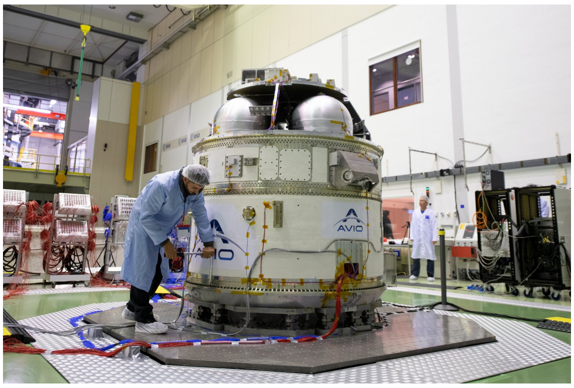


Figure 2.10: AVUM undergoing vibration testing at ESTEC. *-from esa.int*

2.4 Future Developments: VEGA-C and VEGA-E

VEGA-C is a more powerful configuration that increases payload capacity by 50%, up to 2,200 kg, and features the new P120C main stage and the Zefiro 40 second stage. Its maiden flight took place in 2022 [3]. The primary motivation behind its development was to enhance the cost-effectiveness and competitiveness of the Vega launcher, particularly in the rapidly growing small and medium satellite market. With the increasing demand for Earth observation, telecommunications, and scientific missions, VEGA-C was designed to provide a higher lift capability, improved flight performance, and reduced operational costs compared to its predecessor.

VEGA-E is an upcoming lightweight configuration, optimized for cost-effective small satellite launches. It will feature the P120C and Zefiro 40, along with a new cryogenic upper stage, the M10, powered by liquid methane, offering a specific impulse of 362 seconds. The first launch is expected in 2024 [3]. The rationale behind developing VEGA-E is to further reduce launch costs and increase operational flexibility, particularly for commercial and institutional customers requiring frequent, low-cost access to space. The adoption of methane-based propulsion also represents a technological step forward in terms of sustainability, efficiency, and reusability, aligning with future European spaceflight strategies.

Chapter 3

Aeroacoustic Characteristics of Supersonic Jets

3.1 Introduction

The generation of acoustic waves from a supersonic jet is a complex physical phenomenon related to compressible flows, which occurs in various aerospace applications, for example:

- Noise generated by an air-breathing engine's free exhaust jet, relevant for assessing the environmental impact of commercial aircraft and their acoustic pollution over residential areas.
- Noise from a rocket engine exhaust plume, where pressure disturbances may be intense enough to potentially damage the launcher or nearby structures, facilities, and sensitive electronic or mechanical equipment.

The latter is particularly critical during the rocket nozzle start-up phase when the unsteady turbulent flow exiting the nozzle impinges on an inclined plate, commonly referred to as a flame deflector. The deflector's design must mitigate feedback acoustic waves that could impact the rocket structure or surrounding infrastructures. Excessive acoustic noise with high energy content in the resonating frequency range of the launcher can lead to fatigue failure. In figure 3.1, we can see the launch pad at Europe's Spaceport in French Guiana, where Ariane 6 took his first flight on 9 July 2024, with its flame deflector.

In modern space launch facilities, sound suppression systems are implemented to reduce acoustic waves. A common approach involves stream jets of water or large water containers positioned around the base walls to absorb shock waves, as seen in the Kennedy Space Center's Sound Suppression Water System, shown in figure 3.2. Additionally, the recovery of reusable boosters and lower rocket stages has revived interest in studying vertical plume impingement, similar to that examined in VTOL (Vertical Take-Off and Landing) vehicles. In figure 3.3 a booster of the Falcon 9 launcher discending on an *autonomous spaceport drone ship* (ASDS) is shown.

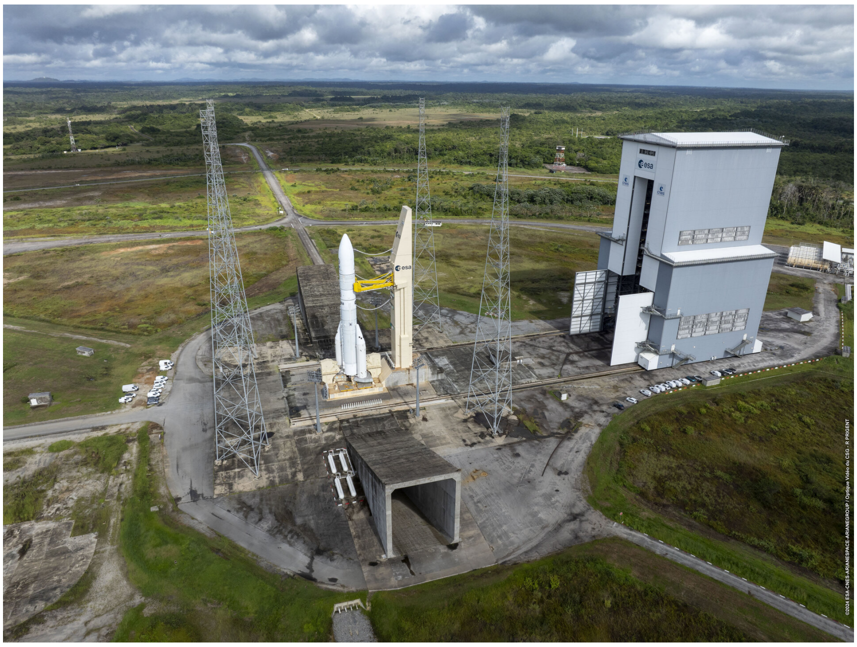


Figure 3.1: Ariane 6 on the launch pad at Europe's Spaceport in French Guiana. *-from ESA.int*

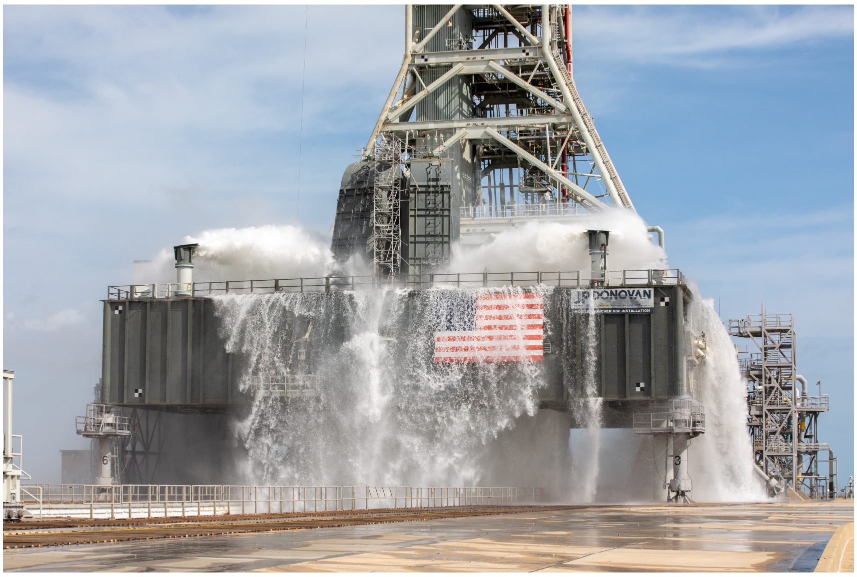


Figure 3.2: Kennedy Space Center's Sound Suppression Water System. *-from NASASpaceFlight.com*

Figure 3.3: Falcon 9 booster descending on the "Of Course I Still Love You" landing platform in the Atlantic Ocean. - *from spacex.com*

3.2 High-Speed Free Jet Noise

Numerous studies conducted over the past decades have demonstrated that when a jet impacts a solid surface, distinct tonal noise can be observed in the far field. This phenomenon arises due to the resonance between large-scale coherent turbulent structures moving downstream and the pressure waves produced by the impingement, which propagate upstream.

The physical characteristics of noise generated by high-speed jets can be categorized based on whether the jet is perfectly expanded or contains shock structures (i.e., overexpanded or underexpanded jets). This classification helps to identify distinct noise generation mechanisms, such as turbulent mixing, Mach radiation, and broadband shock-associated noise (BBSAN) [5, 6, 7].

3.2.1 Perfectly Expanded Jets

A supersonic jet is considered perfectly expanded when the nozzle pressure ratio (NPR), defined as the ratio between the stagnation pressure in the combustion chamber and the ambient pressure, results in a fully expanded flow without shocks. For a convergent nozzle operating with ideal air, the jet reaches sonic conditions when the NPR exceeds the critical value of 1.89. The corresponding Mach number for such a jet is given by:

$$M_j = \sqrt{\frac{2}{\gamma - 1}\left[(NPR)^{\frac{\gamma-1}{\gamma}} - 1\right]} \tag{3.1}$$

where γ represents the specific heat ratio. This Mach number corresponds to

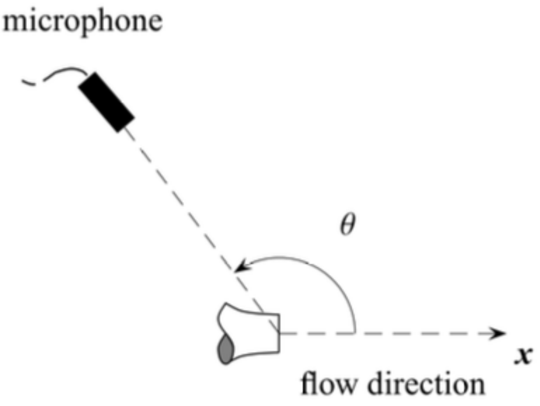


Figure 3.4: Sketch of the listener (microphone) disposition for the experimental setup in Tam et alia [7]

the design Mach number of a generic converging-diverging nozzle and is crucial for achieving a shock-free jet.

Ensuring a perfectly expanded flow—free of shocks—is particularly relevant to this study, as the acoustic noise generated under these conditions arises solely from turbulent mixing. Research conducted by Morris et al. (2007, 2009) [6, 5] identified two distinct mechanisms responsible for mixing noise generation.

The first mechanism is linked to the quasi-periodic passage of large turbulent structures near the end of the potential core of the jet. In this region, the flow is inherently unsteady and intermittent, causing surrounding air to be entrained and accelerated into the jet. As these large-scale turbulent structures propagate downstream, they induce Mach waves, akin to those generated by supersonic flow over a corrugated surface. This phenomenon, known as Mach radiation, is recognized as the dominant noise source in the downstream direction.

The second potential noise source arises from turbulent mixing within the shear layer, involving interactions between large-scale coherent turbulence and smaller-scale eddies, down to the Kolmogorov scale, through an energy cascade driven by nonlinear interactions. This component of noise is almost isotropic, as it is largely masked by the intermittent nature of the flow in the downstream direction, while being more prominent in the radial direction.

Both of these sound-generation mechanisms have been extensively studied over the past decades. Notably, Tam et al. (1996) [7] analyzed a comprehensive set of experimental data, leading to the development of semi-empirical spectral models for predicting jet noise. Further research by André, Castelain, and Bailly (2013) [8] investigated the acoustic spectra of a perfectly expanded jet at $M_j = 1.3$. By positioning a microphone at two different observer angles, $\theta = 30°$ and $\theta = 90°$ relative to the jet flow direction Figure 3.4, they obtained valuable acoustic spectra, later compared against semi-empirical predictions, with the results depicted in Figure 2.5.

The acoustic spectrum is typically characterized using the **Strouhal number**

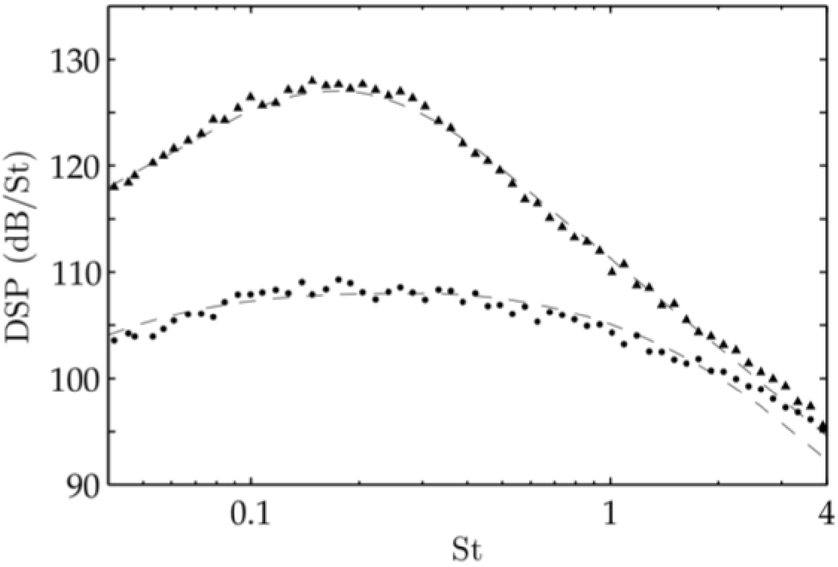


Figure 3.5: Acoustic spectra of a perfectly expanded jet at $M_j = 1.3$, measured at △ $\theta = 30^\circ$ and $\theta = 90^\circ$. The dashed lines represent the semi-empirical data obtained by Tam [7].

St, defined as:

$$St = \frac{fD}{U_j} \tag{3.2}$$

where:

- f is the frequency of the acoustic waves,
- D is the jet exit diameter,
- U_j is the jet exit velocity.

This dimensionless parameter is widely used to analyze jet noise characteristics, correlating frequency with flow parameters, allowing for comparisons across different flow conditions and jet configurations.
For both mechanisms responsible for acoustic noise generation, the peak Strouhal number occurs at approximately $St \approx 0.17$ in the downstream direction and $St \approx 0.29$ in the sideline direction. Additionally, the amplitude difference between these two contributions reaches nearly 20 dB.

3.2.2 Shock-Containing Jets (Over and Underexpanded)

The presence of shocks within the potential core of the jet introduces an additional source of noise compared to the previously described mechanisms. When the exit pressure does not match the ambient pressure, the jet must adapt through either an oblique shock in the case of an overexpanded jet, where the exit pressure is lower than the ambient pressure, or through an expansion fan in the case of an underexpanded jet, where the exit pressure is higher than the ambient pressure.

The noise generated by shock-cell structures typically consists of screech tones, characterized by high-amplitude discrete frequencies, along with a broadband component associated with the shocks themselves. Screech tones are particularly significant in supersonic jets, as they have a strong influence on the acoustic emission field and can alter the evolution of the turbulent jet flow. The generation of screech can be understood as a feedback loop occurring within the mixing layer of the jet.

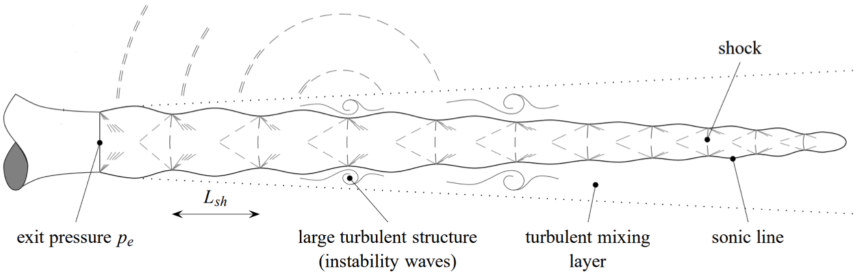


Figure 3.6: Screech generation diagram of an underexpanded converging nozzle.

As the jet evolves, vortical structures form at the nozzle lip and are convected downstream in the mixing layer of the jet plume. These structures interact with the shock-cell formations in the flow, producing acoustic waves. The disturbances propagate upstream through the surrounding medium and eventually reach the nozzle rim, where they induce new vortical structures, reinforcing the process in a self-sustaining cycle. The feedback acoustic waves typically originate between the second and fifth shock-cell, leading to the emission of discrete frequencies known as screech tones and their harmonics.

Several acoustic models have been developed to characterize the behavior of these noise components. Powell et al. (1992) [9] proposed a model to estimate the screech frequency, expressed as:

$$\frac{n}{f_s} = \frac{h}{u_c} + \frac{h}{c_\infty} = \frac{n_{sh} L_{sh}(1 + M_c)}{u_c} \tag{3.3}$$

where n is an integer, L_{sh} represents the average spacing between shock cells, u_c is the convection velocity, and c_∞ is the speed of sound in the surrounding medium. The convection Mach number is denoted as M_c, while $h = n_{sh} L_{sh}$ defines the position of the n-th shock within the nozzle.

This equation demonstrates how the convection time of vortical structures, defined by the ratio h/u_c, is a key factor in determining the resulting screech frequency. Additionally, the second term in the equation, h/c_∞, accounts for the upstream propagation of acoustic waves within the surrounding medium.

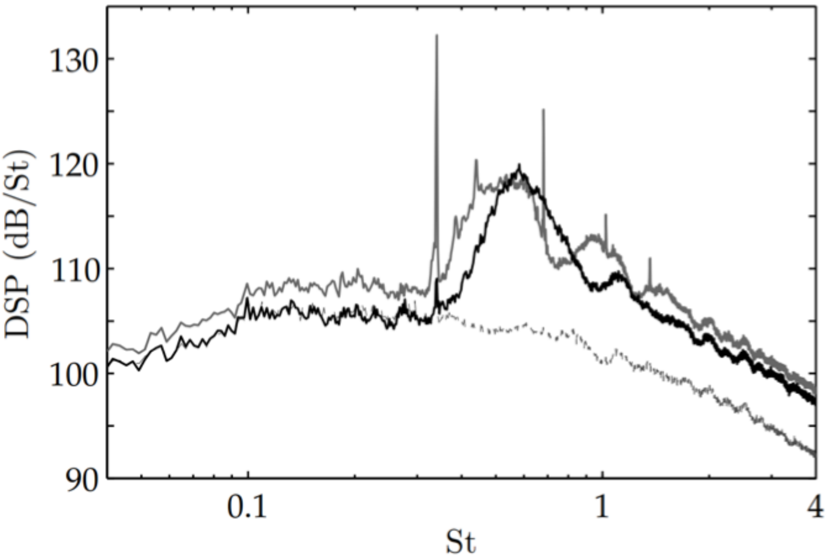


Figure 3.7: Far-field acoustic spectra measured at $r \approx 52D$ for a perfectly expanded nozzle with $M_j = 1.3$. *Smooth Converging Nozzle (solid line), Notched Converging Nozzle (dashed line), Perfectly Expanded Converging-Diverging Nozzle (dotted line)* from André et al. [8].

In jets containing shock structures, broadband shock-associated noise (BBSAN) represents another significant noise component. Although BBSAN shares similarities with screech in that both involve interactions between turbulent structures and shock cells, BBSAN does not involve a resonant feedback mechanism. The broadband noise component originates from the intersection of turbulent eddies with the shock structures within the jet shear layer.

As described by Harper-Bourne and Fisher (1974), BBSAN exhibits no strong directivity pattern but is instead distributed over a broad range of frequencies. The peak frequency f_p of this noise source is given by:

$$f_p = \frac{u_c}{L_{sh}(1 - M_c \cos\theta)} \tag{3.4}$$

where u_c is the convection velocity, L_{sh} is the average spacing between the shock cells, M_c is the convective Mach number, and θ represents the observer angle relative to the downstream jet axis. The peak frequency tends to increase as the observer moves toward the upstream direction, where BBSAN reaches its maximum intensity.

Figure 3.6 illustrates the mechanism of screech generation in an underexpanded jet. The figure provides a visual representation of the interactions between large turbulent structures, shock-cell formations, and the associated sound emission.

Experimental data comparing acoustic spectra from different nozzle configurations are presented in Figure 3.7. This comparison includes measurements for a smooth converging nozzle, a notched converging nozzle, and a perfectly expanded

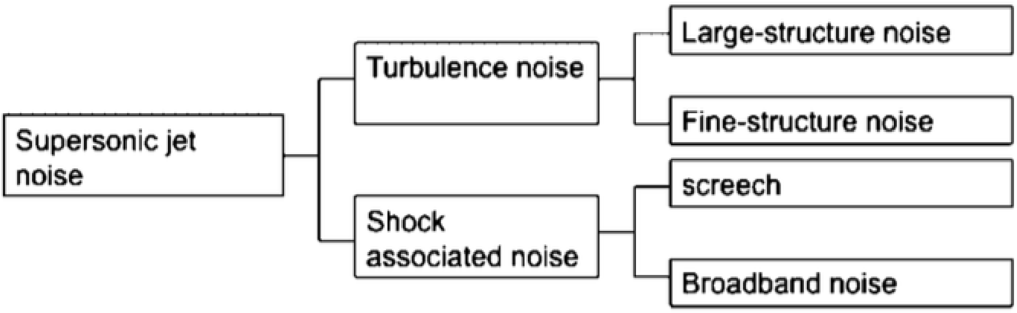


Figure 3.8: Chart showing the classification of supersonic jet noise and their causes, from Fujii et al. [10]

converging-diverging nozzle. The spectra were recorded in the far field at a distance of approximately 52 nozzle exit diameters, with an observer angle of $\theta \approx 110°$. The data indicate that screech tones are prominent in the upstream direction for over-expanded jets but are absent in the perfectly expanded converging-diverging nozzle, where no shock structures are present.

The table in Figure 3.8 and the schematic in Figure 3.9 provide a summary of the primary acoustic sources in supersonic jets and their directional characteristics.

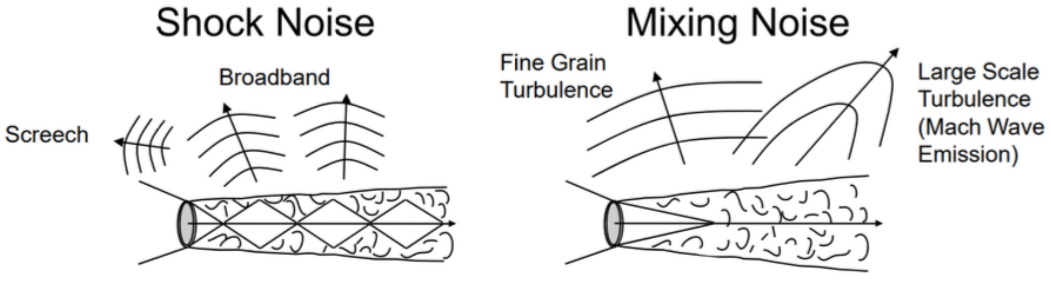


Figure 3.9: Diagram illustrating both shock-associated noise and turbulent mixing noise components, along with their primary propagation directions.[9]

3.2.3 Aeroacoustic Characteristics of Impinging Jets

This section presents an overview of the aeroacoustic properties of impinging jets, as documented in the literature.

During the ignition phase of both Liquid Rocket Engines (LRE) and Solid Rocket Boosters (SRB), intense pressure waves known as ignition overpressure (IOP) are generated and expelled through the nozzle [11]. Unlike the previously discussed free jet case, rocket ignition typically occurs within a silo or on a launch pad. In addition to IOPs, which consist of a singular strong pressure wave, continuous pressure waves emerge from the turbulent jet plume structure, along with waves resulting from the impingement of the jet on the launch pad. These waves represent a significant safety concern, as they expose both the rocket launcher and its payload—such as satellites, cargo, or crew—to severe vibrations induced by the surrounding acoustic field.

The design of launch structures and associated infrastructure plays a critical role in mitigating these acoustic effects. This challenge necessitates the combined use of experimental testing and computational aeroacoustic simulations to develop effective noise reduction strategies.

Recent studies have investigated the impingement of a perfectly expanded supersonic jet on a flat plate through both numerical simulations, using hybrid RANS/LES methods, and experimental approaches [10, 12, 13] . Analysis of both steady and unsteady flow fields has revealed several key insights.

As the jet exits the nozzle, it undergoes a transition process that initiates within the shear layer. As the potential core of the jet develops, the shear layer thickens until the jet reaches a fully developed state. Experimental studies conducted by Panda and Seasholtz (2002), as well as by Akamine et al. (2014), indicate that the potential core length of a jet is approximately $8D$, where D represents the nozzle exit diameter. Computational studies performed by Brehm [10] and Nonomura [12] positioned the flame deflector at a distance of $5D$ from the nozzle exit. Consequently, the jet interacts with the deflector before completing its transition phase, leading to the formation of an impingement shock. The unsteadiness of this shock structure, along with its interaction with the turbulent shear layer, generates substantial pressure fluctuations on the deflector wall.

Figure 3.10 presents the results for a perfectly expanded supersonic jet with $M_j = 2$ impacting an inclined surface at a 45° angle, as analyzed by Nonomura et al. [12]. The numerical simulations reveal three predominant propagation directions for the acoustic waves:

- (i) The arrows indicate Mach waves originating from the shear layer of the jet;
- (ii) Represents the main propagation path of the waves generated due to the impingement of the jet on the deflector;
- (iii) Depicts the propagation of Mach waves arising from the wall jet shock-cell interactions and the shear layer.

A comprehensive analysis was conducted over four different cases, considering variations in jet temperature and nozzle-to-plate distances of 5D and 10D. Regardless of these variations, all cases consistently demonstrated the presence of the three identified acoustic wave propagation directions. Additionally, the study on the *sound pressure level* confirmed that the most intense acoustic emissions are concentrated near the symmetry plane of the impingement region.

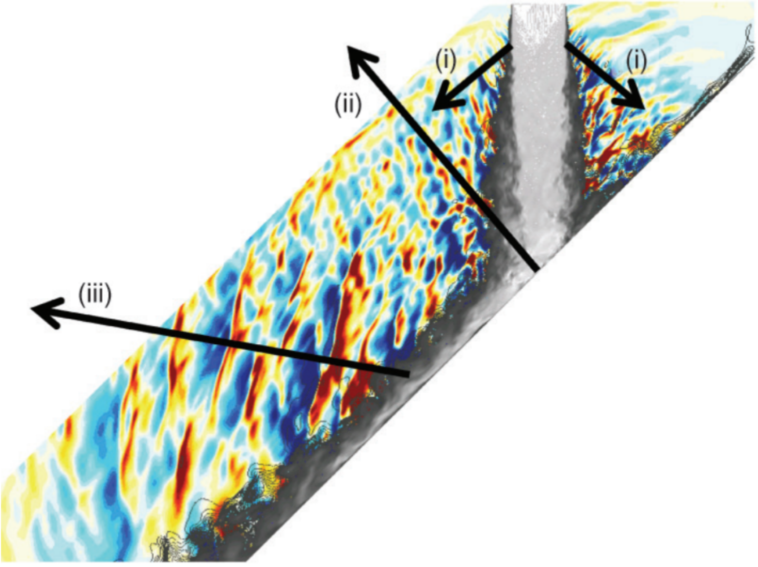


Figure 3.10: Instantaneous flow field visualization of a perfectly expanded jet with $M_j = 2$, impinging on a 45° inclined plate. The color contours represent static pressure distribution, while the monotone contours illustrate dynamic pressure distribution. *Adapted from Nonomura et al.* [12].

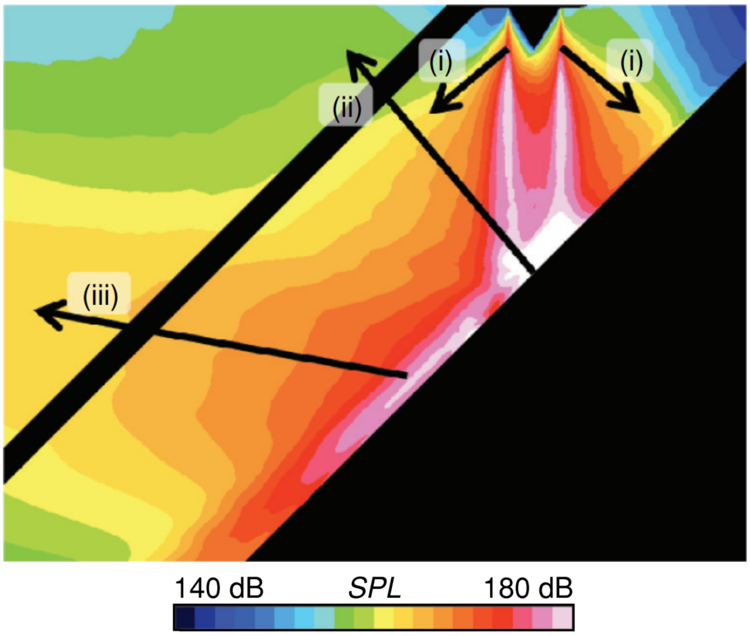


Figure 3.11: Overall Sound Pressure Level (OASPL) distribution in the symmetry plane for a fully expanded impinging hot jet. *Adapted from Nonomura et al., "Aeroacoustic Waves Generated from a Supersonic Jet Impinging on an Inclined Flat Plate"* [12].

Chapter 4

Computational Fluid Dynamics (CFD)

4.1 Introduction

Fluid dynamics plays a crucial role in numerous engineering and scientific applications, governing phenomena ranging from aerodynamics and propulsion to weather prediction and biomedical flows. However, solving fluid dynamic problems analytically is often impractical due to the complexity of the governing equations. In this context, Computational Fluid Dynamics (CFD) has emerged as a powerful tool for simulating and analyzing fluid behavior in a wide range of scenarios.

CFD aims to solve thermo-fluid dynamic problems by obtaining numerical solutions to partial differential (or integro-differential) equations that describe the motion of fluids and their associated physical phenomena, all derived from fundamental conservation principles.

To achieve numerical solutions, discretization methods are employed, transforming the governing differential equations into algebraic equations within a defined spatial and temporal domain. These approximations allow for the computation of flow field properties at discrete points in both space and time, enabling a detailed analysis of complex flow behaviors that would otherwise be impossible to study analytically.

With the continuous advancement of computational resources, CFD has become an essential tool in both academia and industry, offering a cost-effective and time-efficient alternative to experimental testing. It is widely applied in fields such as turbulence modeling, mechanical and aerospace engineering, environmental studies, atmospheric sciences for weather forecasting and climate research, astrophysics, and even medicine, where numerical simulations of blood flow assist in studying clot formation and vascular conditions. [14].

4.2 Governing Equations

The fundamental laws governing fluid flow can be derived by considering a specific spatial region referred to as the *control volume* (CV). The properties of interest within this volume can be categorized as *intensive*, such as density and velocity, which describe properties per unit volume or per unit mass. It is often more practical to analyze intensive properties rather than extensive ones, as they facilitate the study of flow within a defined volume (Eulerian approach) instead of tracking individual particles (Lagrangian approach) with their own mass and momentum.

If we denote ϕ as a generic intensive quantity (or an extensive quantity per unit mass), the corresponding extensive property Φ over the control volume V is given by:

$$\Phi = \int_V \rho\phi dV \tag{4.1}$$

where V represents the volume of the control volume. Based on this definition, the general conservation equation for a control volume can be formulated using the Reynolds transport theorem as follows:

$$\frac{d\Phi}{dt} = \frac{d}{dt}\int_V \rho\phi dV + \int_S \mathbf{n}\cdot\mathbf{v}(\rho\phi)dS = \sum f_\phi \tag{4.2}$$

Here, S represents the surface enclosing the control volume with outward unit normal $\mathbf{n}$, and $\mathbf{v}$ is the velocity field of the fluid. The last integral in Equation (3.2) accounts for the *convective flux* of the quantity ϕ. This equation essentially describes the rate of change of the property Φ within a control mass as a result of its variation inside the volume and its transport across the control surface. The term f_ϕ accounts for additional effects contributing to variations in ϕ, such as production, dissipation, or diffusion.

The integral conservation equations governing fluid flow can be derived from fundamental physical principles, namely:

- Conservation of mass;
- Newton's Second Law ($\sum \mathbf{F} = m\mathbf{a}$);
- Conservation of energy.

4.2.1 Mass Conservation

The *mass conservation* equation is obtained by setting $\phi = 1$ in Equation (4.1), leading to:

$$\frac{d}{dt}\int_V \rho dV + \int_S \rho\mathbf{v}\cdot\mathbf{n}dS = 0 \tag{4.3}$$

In this case, f_ϕ vanishes since no mass sources or sinks are assumed within the control volume.

4.2.2 Momentum Conservation

For the *momentum conservation* equation, we consider $\phi = \mathbf{v}$, the velocity vector, leading to:

$$\frac{d}{dt}\int_V \rho\mathbf{v}dV + \int_S \rho\mathbf{v}\mathbf{v}\cdot\mathbf{n}dS = \sum \mathbf{f} \tag{4.4}$$

This is equivalent to Newton's Second Law. The convective flux term introduces non-linearity into the equation. The forces acting on the control volume consist of surface forces, described by the Cauchy stress tensor $\mathbf{T}$, and body forces $\mathbf{b}$ (such as gravity, where $\mathbf{b} = \mathbf{g}$). Thus, Equation (4.4) can be rewritten as:

$$\frac{d}{dt}\int_V \rho\mathbf{v}dV + \int_S \rho\mathbf{v}\mathbf{v}\cdot\mathbf{n}dS = \int_S \mathbf{T}\cdot\mathbf{n}dS + \int_V \rho\mathbf{b}dV \tag{4.5}$$

The Cauchy stress tensor $\mathbf{T}$, in the context of a Newtonian fluid model, is defined as:

$$\mathbf{T} = (-p + \lambda\nabla\cdot\mathbf{v})\mathbf{I} + 2\mu\mathbf{D} \tag{4.6}$$

where p is the static pressure, μ and λ are the first and second viscosity coefficients, respectively, $\mathbf{I}$ is the identity tensor, and $\mathbf{D}$ is the deformation tensor:

$$\mathbf{D} = \frac{1}{2}(\nabla\mathbf{v} + \nabla\mathbf{v}^T) \tag{4.7}$$

Decomposing $\mathbf{D}$ into its isotropic and deviatoric components, we obtain:

$$\mathbf{T} = \left[-p + \left(\lambda + \frac{2}{3}\mu\right)\nabla\cdot\mathbf{v}\right]\mathbf{I} + 2\mu\mathbf{D}_0 \tag{4.8}$$

where $\mathbf{D}_0$ represents the deviatoric component of the stress tensor.

4.2.3 Energy Conservation

Energy conservation is generally expressed in terms of total energy $\phi = E$, where E represents the sum of kinetic, potential, and internal energy:

$$\frac{d}{dt}\int_V \rho E dV + \int_S \rho E\mathbf{v}\cdot\mathbf{n}dS = \int_S (\mathbf{v}\cdot\mathbf{T})\cdot\mathbf{n}dS - \int_S \mathbf{q}\cdot\mathbf{n}dS \tag{4.9}$$

The right-hand side terms correspond to:

- *Mechanical work* per unit time, with $\mathbf{T}$ decomposed into isotropic and deviatoric parts;

- *Heat flux*, modeled using Fourier's law as $\mathbf{q} = -k\nabla T$, where k is the thermal conductivity.

Using the definitions above, the energy conservation equation can be rewritten as:

$$\frac{d}{dt}\int_V \rho E dV + \int_S \rho\left(E+\frac{p}{\rho}\right)\mathbf{v}\cdot\mathbf{n}dS = \int_S \mathbf{v}\cdot(2\mu\mathbf{D}_0)\cdot\mathbf{n}dS + \int_S k\nabla T\cdot\mathbf{n}dS \quad (4.10)$$

By assuming differentiability, the integral conservation equations can be expressed in differential form, leading to the well-known Navier-Stokes equations:

$$\begin{cases} \frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0 \\ \frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) = \nabla\cdot\mathbf{T} + \rho\mathbf{g} \\ \frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left[\rho\left(E+\frac{p}{\rho}\right)\mathbf{v}\right] = \nabla\cdot(\mathbf{v}\cdot 2\mu\mathbf{D}_0) + \nabla\cdot(k\nabla T) \end{cases} \quad (4.11)$$

4.3 Numerical Methods in CFD

Since closed-form analytical solutions for these equations are generally not possible except for simplified cases, numerical methods are employed to discretize and solve them.

4.3.1 Discretization Methods

The analytical solution of the previously discussed equations is only feasible under highly simplified mathematical assumptions, such as considering a extitpotential flow (steady, incompressible, and inviscid conditions), where the flow field can be described by Laplace's equation. However, even under these constraints, solving Laplace's equation for complex geometries remains challenging.

To overcome this, numerical solutions are obtained by solving discretized versions of the governing equations at specific discrete locations within the computational domain.

The conservation laws in the form of partial differential equations (PDEs) are discretized using different approaches, among which the following are commonly used:

- **Finite Difference Method**: This is one of the earliest numerical techniques for solving PDEs and is particularly effective for simple geometric domains. In this approach, the computational domain is subdivided into a structured grid, where the discrete points of the grid, called nodes, represent the locations at which the solution is computed. The partial derivatives in the governing equations are approximated using finite difference formulas, which express derivatives in terms of the function values at neighboring nodes.

  One way to derive these approximations is by employing a Taylor series expansion for a generic function ϕ around a given node x_i:

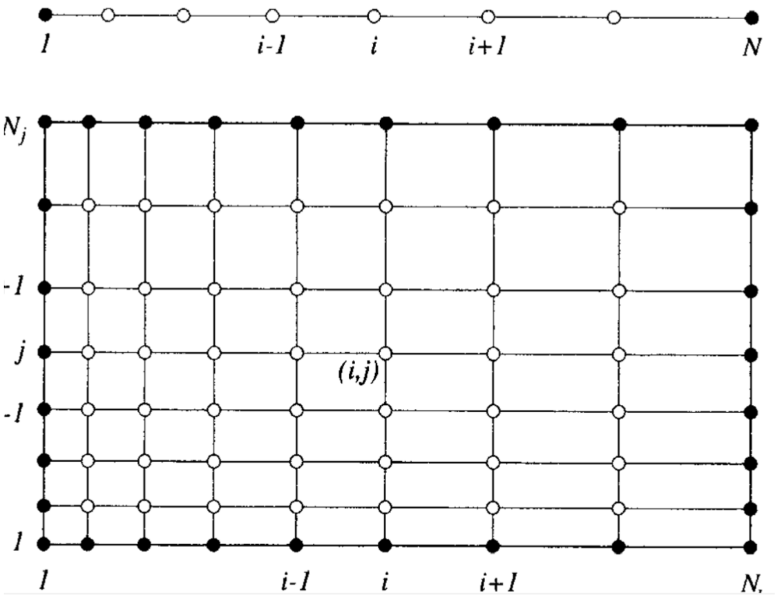


Figure 4.1: Illustration of a one-dimensional (1D) grid (top) and a two-dimensional (2D) grid utilized for Finite Difference methods. The 2D grid is non-uniform, as the spacing between nodes varies across the domain. [15]

$$\phi(x) = \phi(x_i) + (x - x_i)\frac{\partial \phi}{\partial x} + \frac{(x - x_i)^2}{2!}\frac{\partial^2 \phi}{\partial x^2} + \frac{(x - x_i)^3}{3!}\frac{\partial^3 \phi}{\partial x^3} + H \tag{4.12}$$

where H represents the higher-order terms in the expansion. By using neighboring points such as x_{i-1} or x_{i+1}, finite difference approximations for the first derivative of ϕ can be obtained. For example, considering the first derivative at the $(i+1)$-th node, we obtain:

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i} - \frac{x_{i+1} - x_i}{2}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i - \frac{(x_{i+1} - x_i)^2}{6}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i + H \tag{4.13}$$

These approximations form the basis for constructing numerical schemes that solve PDEs in a discretized space, allowing the flow variables to be computed iteratively across the grid.

A more practical approach to approximate the first derivative of ϕ can be obtained by assuming that higher-order derivative terms are negligible when the grid spacing is sufficiently small. This leads to an approximation given by:

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i} \tag{4.14}$$

This formulation is referred to as the forward-difference scheme (FDS), as it evaluates the derivative based on points forward of the node x_i. Similarly, alternative schemes exist, including the backward-difference scheme (BDS), which relies on values from previous nodes, and the central-difference scheme (CDS), which takes into account values from both forward and backward nodes. These formulations are expressed as:

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{\phi_i - \phi_{i-1}}{x_i - x_{i-1}} \quad (BDS) \tag{4.15}$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{\phi_{i+1} - \phi_{i-1}}{x_{i+1} - x_{i-1}} \quad (CDS) \tag{4.16}$$

These methods introduce an error due to the neglected higher-order terms, known as truncation errors. The first term of this truncation error generally dominates the approximation's accuracy, with the magnitude of the error being directly proportional to the grid spacing.

Both the forward and backward difference schemes are first-order accurate, while the central-difference scheme provides a second-order accurate approximation due to its symmetric nature. The choice of the numerical scheme depends on the balance between accuracy and computational cost.

An alternative method for discretization is polynomial fitting, which interpolates a function based on known data points. This approach provides a more refined approximation and is particularly useful for irregular grids. By applying this method, the first derivative can be expressed as:

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1}(x_i - x_{i-1})^2 - \phi_{i-1}(x_{i+1} - x_i)^2 + \phi_i[(x_{i+1} - x_i)^2 - (x_i - x_{i-1})^2]}{(x_{i+1} - x_i)(x_i - x_{i-1})(x_{i+1} - x_{i-1})} \tag{4.17}$$

This formulation can be derived while considering a second-order central difference scheme (CDS). Similar numerical techniques allow the discretization of higher-order derivatives, enabling the transformation of differential equations (such as steady one-dimensional problems or parabolic equations) into an algebraic system solvable through iterative methods, including Gauss, Jacobi, and Successive Over-Relaxation (SOR) methods.

For unsteady flow problems, an additional level of discretization is required in the time domain to ensure temporal progression of the solution. When solving transport equations, numerical methods typically employ either *explicit* or *implicit* schemes. In explicit methods, the solution at the new time step t^{n+1} is computed directly using known values from the previous time step t^n. This results in the algebraic formulation:

$$\phi^{n+1} = A\phi^n \tag{4.18}$$

Examples of explicit schemes include the first-order Euler method and the leapfrog method, which uses information from three time levels and employs a central difference scheme (CDS) for spatial discretization.

Conversely, implicit methods formulate the fluxes and source terms in terms of the unknown variables at the new time level, leading to a different algebraic representation:

$$A\phi^{n+1} = \phi^n \tag{4.19}$$

Although implicit methods require the inversion and storage of the coefficient matrix A at each time step, they are often preferred in cases where numerical stability is a primary concern. Among the most widely used implicit methods are the Euler implicit method and the second-order Crank-Nicholson method, which offer improved accuracy and stability.

- **Finite Volume Method**: This approach is based on the integral formulation of the conservation equations. Unlike the finite difference method, where computations are performed at discrete grid points, the finite volume method discretizes the fluid domain into a finite number of control volumes (CVs), with the grid defining the *boundaries* of these volumes rather than the computational nodes themselves.

  Typically, the grid is structured so that nodes are positioned at the center of each control volume, following a cell-centered approach. However, for improved accuracy in derivative approximations, computational nodes can be assigned beforehand, with the grid subsequently generated such that CV faces align midway between nodes, or alternatively, with nodes serving as the vertices of CVs: an approach known as the cell-vertex method.

  Figure 4.2 illustrates examples of notations for finite volume grids in both 2D and 3D Cartesian coordinates. To numerically solve the integral conservation equations, it is necessary to approximate both surface and volume integrals using quadrature formulas. Depending on the chosen approximation, the resulting equation may exhibit similarities with those obtained using the Finite Difference Method. In the case of surface integrals, the flux of a generic quantity f through the boundaries of a control volume is given by the sum of the flux contributions across each of the volume's faces:

$$\int_S f dS = \sum_k \int_{S_k} f dS \tag{4.20}$$

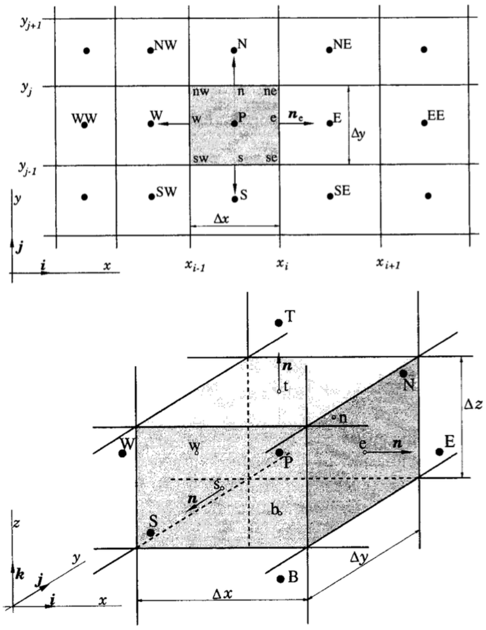


Figure 4.2: Illustration of standard 2D and 3D grid representations utilized in Finite Volume methods. [15].

where f represents either the convective or diffusive flux of the variable ϕ along the normal direction of the control volume face.

Considering a specific face e of the control volume (refer to Figure 4.2), evaluating the surface integral in Equation (4.20) requires knowledge of the function f at every location on S_e. However, since the values of ϕ are computed exclusively at discrete nodes, an approximation method is necessary. The most fundamental approach is the midpoint rule, which approximates the integral as the product of the integrand at the cell-face center and the respective face area:

$$F_e = \int_{S_e} f dS = \bar{f} S_e \approx f_e S_e \qquad (4.21)$$

where f is evaluated at the central position e. Taylor-series expansion can be used to demonstrate that the midpoint rule is second-order accurate [15].

A more refined approximation, offering second-order accuracy, is the trapezoidal rule:

$$\int_{S_e} f dS \approx \frac{S_e}{2}(f_{ne} + f_{se}) \qquad (4.22)$$

where the fluxes f are computed at the corners of the control volume.

An even higher accuracy can be achieved using a fourth-order approximation, such as Simpson's rule:

$$\int_{S_e} f dS \approx \frac{S_e}{6}(f_{ne} + 4f_e + f_{se}) \tag{4.23}$$

Beyond surface integrals, additional terms in the conservation equations involve volume integration. The most basic quadrature method for evaluating volume integrals consists of approximating the integral using the mean value of the function, computed at the center of the control volume, multiplied by the volume ΔV:

$$\int_V q dV \approx \bar{q}\Delta V = q_p \Delta V \tag{4.24}$$

where the subscript p denotes the nodal position at the centroid of the control volume. This approximation is second-order accurate and does not require interpolation, as all necessary variable values are directly available at node p. For higher accuracy, more sophisticated interpolation techniques, such as shape functions, can be employed to evaluate q at multiple locations within the control volume.

The Finite Volume Method includes various interpolation schemes, such as Central Differencing Scheme (CDS) for linear interpolation, QUICK (Quadratic Upstream Interpolation for Convective Kinematics) for higher-order accuracy, and others, each tailored for specific applications in numerical simulations.

- **Finite Elements Method**: This method relies on the subdivision of the computational domain into a set of sub-domains, known as finite elements, where the differential equations are approximated. Unlike the Finite Difference Method (FDM) and Finite Volume Method (FVM), FEM does not require a structured grid, allowing greater flexibility in handling complex geometries. The domain is discretized into elements of arbitrary shape (triangular or quadrilateral in 2D, tetrahedral or hexahedral in 3D), and the solution is approximated using basis functions (shape functions) defined locally on each element. The classical formulation of the method starts from the strong form of the problem and applies either the principle of virtual work or the Galerkin method, leading to a variational formulation of the equation. Integration over areas and volumes is typically performed using numerical quadrature techniques, such as the Gauss-Legendre method. Due to its flexibility, FEM is widely used in computational fluid dynamics, structural mechanics, and wave propagation problems.

4.3.2 Spatial Discretization and Computational Grids

In Computational Fluid Dynamics (CFD), solving the governing equations requires transforming the continuous physical domain into a discrete computational domain. This process, known as spatial discretization, enables numerical methods to approximate derivatives and integrals at a finite number of points. The accuracy and efficiency of the solution strongly depend on how the domain is discretized, which is defined by the computational grid. Grids can generally be classified into two main categories: structured grids and unstructured grids.

- **Structured Grids**: A structured grid is characterized by an orderly arrangement of nodes and elements, typically forming a Cartesian or curvilinear mesh. In this type of grid, each node follows a well-defined indexing pattern, allowing for straightforward connectivity between elements. Structured grids are widely used due to their simplicity in implementation, reduced memory requirements, and high computational efficiency. They are particularly advantageous when solving problems involving simple geometries or smoothly varying flow fields, where an orthogonal grid can minimize numerical diffusion. However, adapting structured grids to complex geometries often requires deformation techniques such as body-fitted grids or curvilinear transformations, which can introduce grid-induced errors.

- **Unstructured Grids**: Unlike structured grids, unstructured grids do not follow a regular pattern and are composed of arbitrarily shaped elements, such as triangles in 2D or tetrahedra in 3D. These grids provide greater flexibility in handling complex geometries, making them suitable for simulations involving intricate boundaries, irregular domains, or adaptive mesh refinement (AMR). Unstructured grids are generated using automatic meshing algorithms such as Delaunay triangulation or Voronoi tessellation, allowing for localized refinement where higher resolution is required. Despite their advantages, unstructured grids require additional computational effort for data storage and connectivity information, which can lead to increased memory usage and processing time compared to structured grids.

The choice between structured and unstructured grids depends on the nature of the problem, computational resources, and required solution accuracy. Hybrid approaches, combining both grid types, are sometimes employed to balance computational cost and geometric flexibility.
In figure 4.3, an example of a structured grid for a NACA 0012 airfoil is presented, while in Figure 4.4 is shown an unstructured grid for the same profile.

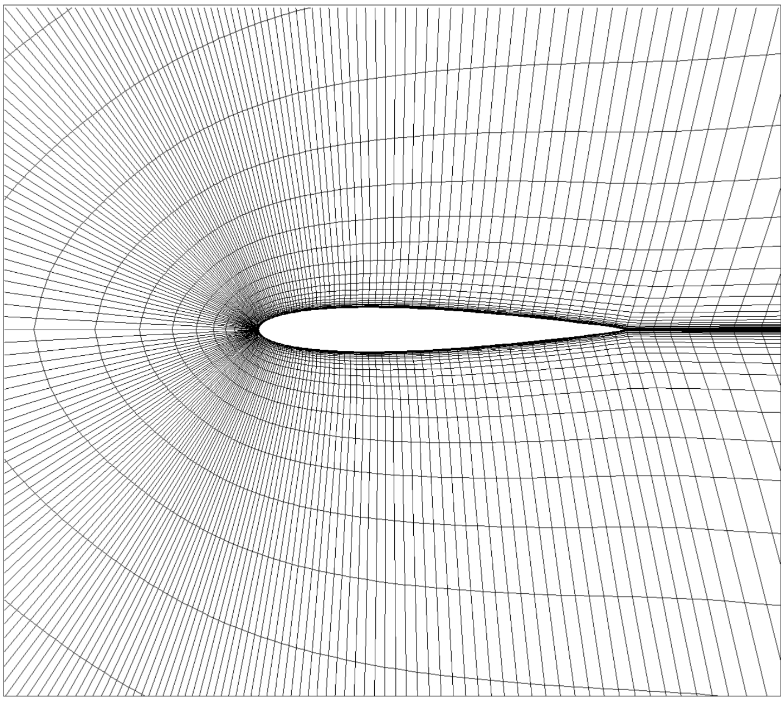


Figure 4.3: Structrured grid for a NACA 0012 arifoil

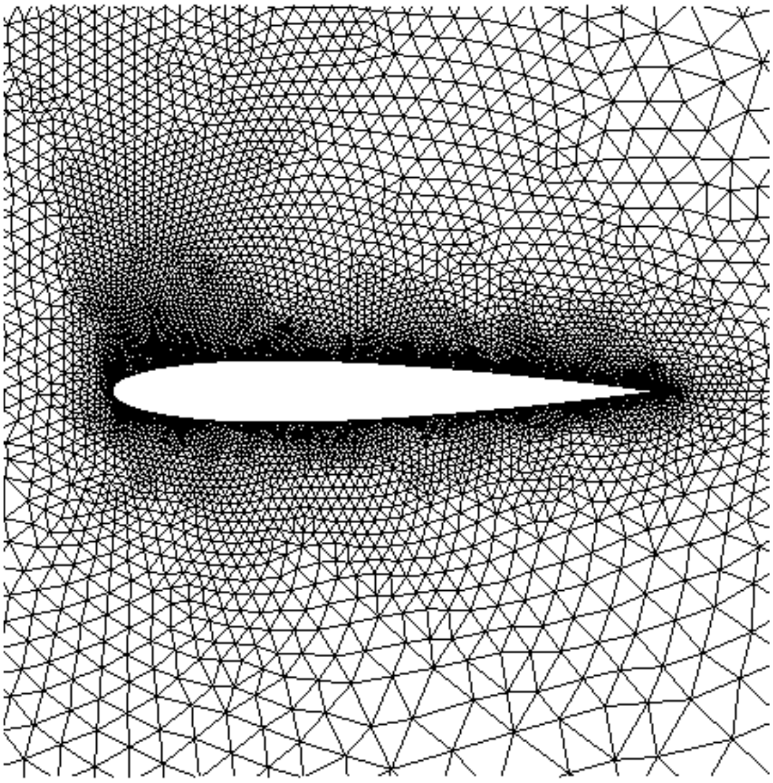


Figure 4.4: Unstructrured grid for a NACA 0012 arifoil

4.3.3 Turbulent Flows

Flow regimes describe the macroscopic behavior of fluid motion. One of the primary parameters used to classify flow regimes is the Reynolds number Re, which is defined as:

$$Re = \frac{\rho v L}{\mu} \tag{4.25}$$

where μ represents the dynamic viscosity of the fluid, ρ is the density, v is the characteristic velocity, and L is the characteristic length of the flow. Based on the Reynolds number and whether the flow is internal or external, the flow regime can be classified as *laminar*, *transitional*, or *turbulent.*

A flow is considered laminar when fluid layers move smoothly past each other without significant mixing or disruption, maintaining a well-ordered velocity profile. On the other hand, a turbulent flow exhibits chaotic behavior, characterized by the formation of vortices and eddies of varying scales. Between these two regimes, transitional flow combines features from both cases and is subject to instabilities that may evolve into turbulence.

The critical Reynolds number Re_{cr}, determined experimentally, helps identify the point at which transition from laminar to turbulent flow occurs. For air, external flows generally transition at $Re_{cr} \approx 5 \times 10^5$, while internal flows become turbulent at around $Re_{cr} \approx 2200$. Most fluid flows encountered in engineering applications are turbulent.

Turbulent flows possess distinctive characteristics, including high unsteadiness, three-dimensional effects, a wide range of velocity and energy scales, and significant mixing phenomena. Accurately simulating turbulence requires capturing these effects within CFD models. Various numerical approaches exist to model turbulence, some of which will be discussed below.

Direct Numerical Simulation (DNS)

A fundamental approach to solving turbulent flows is to directly solve the full Navier-Stokes equations without introducing additional approximations beyond numerical discretization. This approach is known as *Direct Numerical Simulation* (DNS). To fully resolve all turbulent structures, the computational domain must be refined enough to capture scales down to the smallest *Kolmogorov scale* η, which is viscosity-dependent. The grid resolution requirement is given by:

$$k_{max}\eta = \frac{\pi}{\Delta}\eta \geq 1.5 \tag{4.26}$$

where Δ is the minimum grid spacing. Ideally, $\Delta \leq 2\eta$ to ensure sufficient resolution of dissipation processes. Due to the computational expense associated with DNS, these simulations are typically limited to geometrically simple configurations and relatively low Reynolds numbers, making DNS impractical for most engineering applications, where Reynolds numbers are significantly larger.

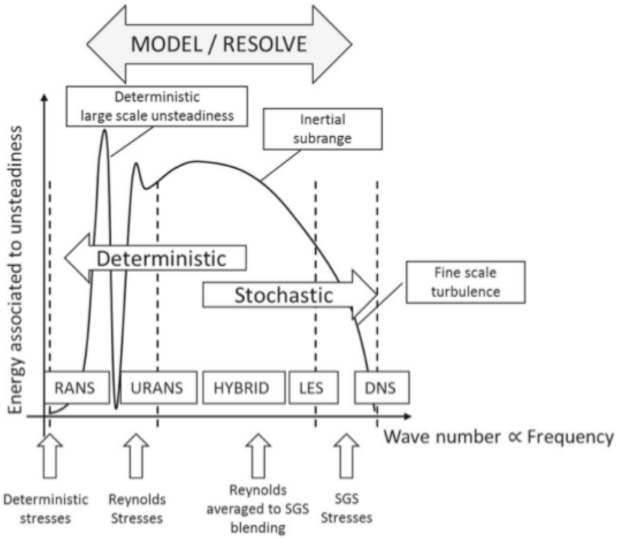


Figure 4.5: Comparison of turbulence modeling approaches (RANS, URANS, Hybrid, LES, DNS) based on their resolution of turbulence scales. The diagram illustrates the transition from deterministic large-scale unsteadiness to stochastic fine-scale turbulence as wave number increases. *-from Ferrero A., Dispense del corso di Fluidodinamica Computazionale dei Sistemi Propulsivi*

Reynolds-Averaged Navier-Stokes (RANS) Equations

A more practical approach for engineering applications is the *Reynolds-Averaged Navier-Stokes* (RANS) formulation, where the flow variables are decomposed into a time-averaged component and a fluctuating component:

$$\phi(x_i, t) = \overline{\phi}(x_i) + \phi'(x_i, t) \tag{4.27}$$

where $\overline{\phi}$ represents the time-averaged value, and ϕ' denotes the fluctuation. The time-averaged value is computed over an interval T sufficiently long compared to the timescale of the fluctuations:

$$\overline{\phi}(x_i) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \phi(x_i, t)dt \tag{4.28}$$

Applying this averaging technique to the Navier-Stokes equations leads to additional terms known as *Reynolds stresses*, represented as $\rho\overline{u_i'u_j'}$, which account for the effects of turbulence on the mean flow. Closure relations, commonly referred to as *turbulence models*, are needed to model these terms.

Several turbulence models exist, ranging from simple zero-equation models to more complex formulations:

- **Zero-equation models:** These define a velocity scale based on a mixing length, without requiring additional transport equations. They are applicable only to simple flows.

- **One-equation models:** These introduce a transport equation for a turbulence variable, such as the turbulent kinetic energy k.
- **Two-equation models:** These include models like k-ω and k-ε, which solve for both the turbulent kinetic energy k and its dissipation rate or specific dissipation rate.
- **More advanced models:** Additional closure relations are required for multiphase flows or chemically reacting turbulence.

Large Eddy Simulation (LES)

Another approach to turbulence modeling involves the spatial averaging of the Navier-Stokes equations and scalar transport equations. In this framework, the computational grid functions as a filter over the fluid domain, resolving turbulent structures larger than the filter scale while modeling the smaller ones. This method is referred to as *Large Eddy Simulation* (LES).

According to Pope [16], LES is valid when the computational grid is fine enough to resolve at least 80% of the turbulent kinetic energy throughout the domain. If this resolution criterion is not met, the simulation is more accurately classified as a *Very Large Eddy Simulation* (VLES). Both LES and VLES are inherently three-dimensional and time-dependent, making them computationally expensive, though still more feasible than Direct Numerical Simulation (DNS) for practical applications.

In LES, the governing equations need to be reformulated to include only large-scale flow structures while filtering out the small-scale fluctuations. The filtered velocity field $\bar{u}_i$ is expressed as:

$$\bar{u}_i(x) = \int G(x, x')u_i(x')dx', \tag{4.29}$$

where $G(x, x')$ is the filter kernel, which is a localized function, typically a Gaussian, a local average, or a cutoff function that removes wavenumbers exceeding a certain threshold.

It is important to note that the LES filtering process does not explicitly depend on the computational grid. However, for practical implementation, the filter width Δ should be at least as large as the grid spacing h, ensuring that only eddies smaller than Δ are modeled while larger structures are directly resolved:

$$\Delta \geq h. \tag{4.30}$$

Applying the filtering operation to the governing equations introduces additional terms, such as the filtered product of velocity components $\bar{u}_i\bar{u}_j$, which differs from

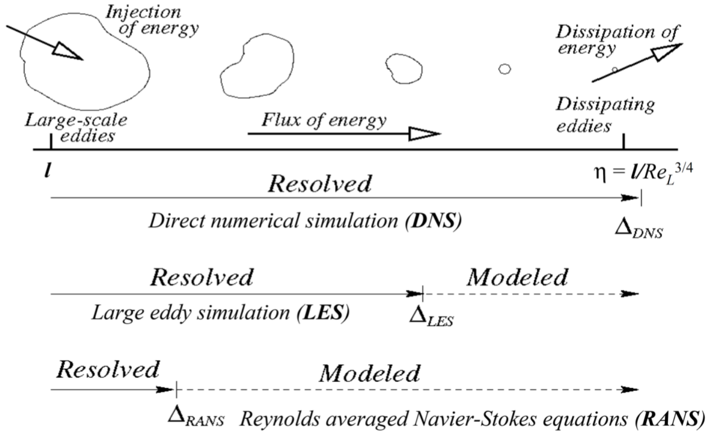


Figure 4.6: Comparison of turbulence modeling approaches (DNS, LES, and RANS) based on resolved and modeled scales, illustrating the energy cascade from large eddies to dissipation.[18]

$\overline{u_i u_j}$. This discrepancy necessitates the introduction of an additional quantity known as the subgrid-scale Reynolds stress τ_{ij}^s:

$$\tau_{ij}^s = -\rho(u_i u_j - \bar{u}_i \bar{u}_j). \tag{4.31}$$

This term represents the momentum flux generated by unresolved small-scale turbulence, which justifies the term ”stress” in the equations. To close the system, additional models are required to approximate the influence of these unresolved scales. One of the most widely adopted closure models for LES is the Smagorinsky eddy-viscosity model[17].

Chapter 5

Signal Analysis: Fourier and Wavelet

5.1 Introduction

The study of signal processing is fundamental in various scientific and engineering fields, enabling the analysis and transformation of data for numerous applications. Among the most widely used mathematical techniques for signal analysis are the Fourier Transform and Wavelet Transform, both of which offer unique insights into the frequency and time-frequency characteristics of signals.

The Fourier Transform has been the cornerstone of spectral analysis, decomposing signals into their constituent frequency components. However, it assumes signal stationarity and lacks time localization, making it insufficient for analyzing transient or non-stationary signals. To address this limitation, wavelet analysis has emerged as a powerful alternative, providing localized time-frequency representations of signals and enabling multiscale analysis.

This chapter presents an overview of both Fourier and wavelet analysis, discussing their mathematical foundations, properties, and applications. The motivation behind their use in scientific and engineering applications is explored, along with the advantages and limitations of each approach. Finally, key applications in fluid dynamics, turbulence analysis, and signal processing are highlighted [19, 20, 21].

5.2 Fourier Analysis

Fourier analysis is a fundamental tool in signal processing, physics, and engineering, enabling the decomposition of signals into their frequency components. The mathematical framework behind Fourier analysis provides essential insights into how signals evolve in the frequency domain and allows for efficient representation of periodic and aperiodic functions.

5.2.1 Mathematical Foundations

Definition of the Fourier Transform (FT)

The Fourier Transform (FT) is a mathematical operation that decomposes a time-domain signal into its constituent frequencies. For a continuous function $f(t)$, the Fourier Transform is defined as:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt, \tag{5.1}$$

where ω is the angular frequency, and j is the imaginary unit. This transformation maps a function from the time domain to the frequency domain, allowing for spectral analysis.

The inverse Fourier Transform enables the reconstruction of the original signal from its frequency components:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega. \tag{5.2}$$

Continuous and Discrete Fourier Transform (DFT)

In practical applications, signals are often sampled rather than continuous. The Discrete Fourier Transform (DFT) is the discrete counterpart of the Fourier Transform, applicable to signals represented by a finite set of samples:

$$F_k = \sum_{n=0}^{N-1} f_n e^{-j2\pi kn/N}, \quad k = 0, 1, ..., N-1. \tag{5.3}$$

where:

- N is the total number of samples,
- f_n are the sampled values of the function,
- F_k represents the frequency domain representation.

The inverse DFT is given by:

$$f_n = \frac{1}{N} \sum_{k=0}^{N-1} F_k e^{j2\pi kn/N}. \tag{5.4}$$

The DFT is computationally expensive for large N, which is why the Fast Fourier Transform (FFT) is commonly used to optimize computations [22, 23].

5.2.2 Properties of the Fourier Transform

The Fourier Transform has several important mathematical properties that make it a powerful analytical tool:

- **Linearity:** The Fourier Transform is a linear operator, meaning that for two functions $f(t)$ and $g(t)$, and constants a, b:

$$\mathcal{F}\{af(t) + bg(t)\} = aF(\omega) + bG(\omega). \tag{5.5}$$

- **Shift Theorem:** A shift in the time domain corresponds to a phase shift in the frequency domain:

$$\mathcal{F}\{f(t - t_0)\} = e^{-j\omega t_0} F(\omega). \tag{5.6}$$

- **Convolution Theorem:** The Fourier Transform of a convolution is the product of the individual Fourier Transforms:

$$\mathcal{F}\{f(t) * g(t)\} = F(\omega)G(\omega). \tag{5.7}$$

  This property is widely used in signal processing and filtering applications [22, 20].

5.2.3 Fast Fourier Transform (FFT)

Computational Efficiency

The Fast Fourier Transform (FFT) is an algorithm that significantly reduces the computational complexity of the DFT from $O(N^2)$ to $O(N \log N)$, making spectral analysis feasible for large datasets. The most commonly used FFT algorithm is the Cooley-Tukey algorithm, which recursively breaks down the DFT computation into smaller subproblems [24].

Applications in Numerical Simulations

The FFT is widely applied in numerical methods for solving partial differential equations (PDEs), image processing, filtering, and aeroacoustic analysis. In fluid dynamics, FFT-based spectral methods are often employed to solve the Navier-Stokes equations in turbulence modeling [16, 25].

5.2.4 Limitations of Fourier Analysis

Despite its widespread use, Fourier analysis has some significant limitations:

- **Loss of time localization:** The Fourier Transform provides information about the frequency content of a signal but does not indicate when specific frequencies occur. This is problematic for analyzing transient phenomena.
- **Issues with non-stationary signals:** Fourier analysis assumes that the signal is stationary, meaning that its frequency content does not change over time. This makes it unsuitable for analyzing signals with time-varying characteristics.

To address these issues, alternative methods such as the Wavelet Transform were developed, offering a better balance between time and frequency localization [19, 20].

5.3 Wavelet Analysis

Wavelet analysis is a powerful mathematical tool developed to overcome the limitations of Fourier analysis in handling non-stationary signals. Unlike the Fourier Transform, which provides only global frequency information, wavelet transforms allow for time-frequency localization, making them particularly useful in turbulence analysis, signal processing, and engineering applications [20, 19].

5.3.1 Introduction to Wavelets

Comparison with Fourier Methods

Traditional Fourier methods, such as the Fourier Transform (FT) and Short-Time Fourier Transform (STFT), decompose signals into sinusoidal components, assuming stationarity over the entire signal duration. This approach is suitable for periodic signals but fails to detect localized features or transient phenomena.

Wavelet analysis, in contrast, employs localized basis functions that can be stretched or compressed, offering a better balance between time and frequency resolution. The key difference lies in how these methods handle resolution:

- Fourier Transform: High-frequency resolution but poor time localization.
- Wavelet Transform: Adaptive resolution, allowing for better time-frequency trade-off.

Advantages in Time-Frequency Analysis

Wavelets possess several advantages over Fourier methods, including:

- Multiresolution analysis (MRA): Ability to analyze signals at multiple scales.

- Better handling of transient and non-stationary signals.
- Sparsity: Wavelet coefficients are often sparse, leading to efficient compression and denoising applications.

5.3.2 Continuous Wavelet Transform (CWT)

Definition and Formulation

The Continuous Wavelet Transform (CWT) is defined as:

$$W(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} f(t) \psi^* \left(\frac{t - b}{a} \right) dt, \tag{5.8}$$

where:

- a is the scale parameter, controlling the dilation or compression of the wavelet
- b is the translation parameter, shifting the wavelet in time
- $\psi(t)$ is the mother wavelet
- $W(a, b)$ represents the wavelet coefficients

Choice of Mother Wavelet Functions

The choice of mother wavelet significantly impacts the analysis. Some commonly used wavelets include:

- **Haar wavelet**: Simple step function, useful for signal compression.
- **Daubechies wavelets**: Compactly supported, ideal for smooth signals [20].
- **Morlet wavelet**: A complex exponential modulated by a Gaussian, suitable for time-frequency analysis.

5.3.3 Discrete Wavelet Transform (DWT)

Multiresolution Decomposition

Unlike the CWT, the Discrete Wavelet Transform (DWT) analyzes a signal using dyadic scaling and discrete shifts, reducing computational complexity. The DWT is given by:

$$W[j, k] = \sum_n f[n] \psi^*_{j,k}[n], \tag{5.9}$$

where:

- j represents the scale level
- k represents the shift index
- $\psi_{j,k}[n] = 2^{-j/2}\psi(2^{-j}n - k)$ are the scaled and translated wavelets

Applications in Numerical Simulations

The DWT is widely used in image compression (JPEG 2000), turbulence analysis, and numerical PDE solvers, where it provides efficient localized representations of fluid structures [21].

5.3.4 Wavelet Power Spectrum

Computation of Energy Distribution

The wavelet power spectrum (WPS) provides insights into the energy distribution of a signal over time and scales. It is defined as:

$$P(s) = |W_n(s)|^2, \tag{5.10}$$

where $W_n(s)$ are the wavelet coefficients at scale s.

Significance Testing

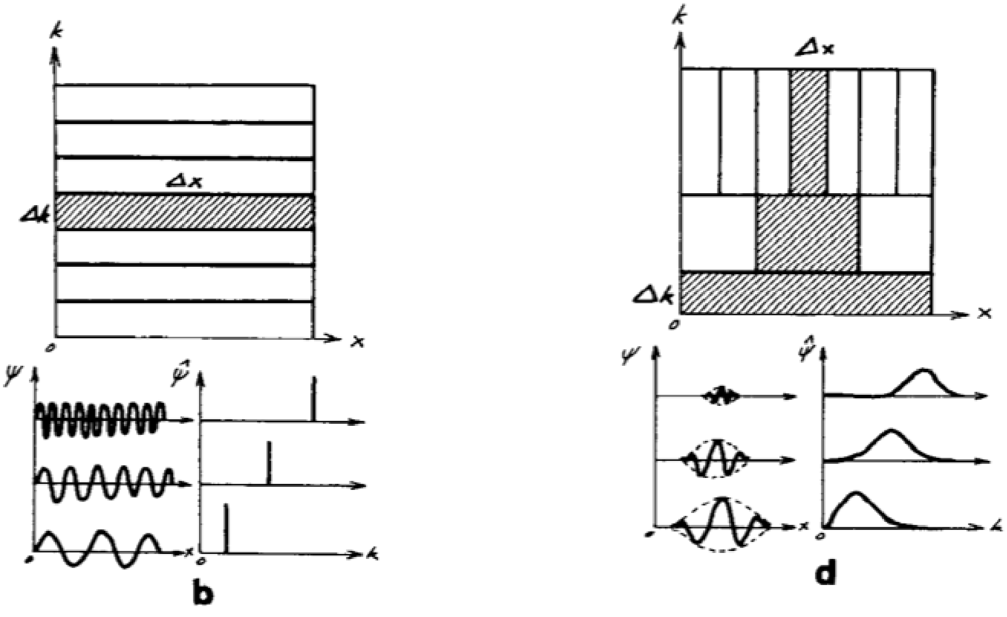
To determine if observed features are statistically significant, Monte Carlo simulations and background noise models (e.g., red noise) are commonly used for significance testing in wavelet analysis [26].

5.3.5 Time-Frequency Localization

Interpretation of Results

Wavelet transforms provide a scalogram, a time-frequency representation that reveals transient structures within signals. Unlike the spectrogram from STFT, wavelet scalograms offer superior time-frequency trade-offs, making them invaluable in analyzing turbulent flow structures and acoustic signals.

Wavelet analysis provides a robust framework for decomposing signals into multiple scales, overcoming the limitations of traditional Fourier methods. With applications in fluid dynamics, signal denoising, and numerical simulations, wavelets have become indispensable in modern engineering and scientific research [19, 20, 26].



(b)

(d)

Figure 5.1: Comparison between Fourier and wavelet decomposition. (b) Fourier transform decomposes a signal into sinusoidal functions of infinite support, resulting in a purely frequency-based representation. The grid in the frequency domain remains uniform, with a constant Δk resolution. (d) Wavelet transform, on the other hand, uses localized basis functions of varying scales, allowing simultaneous time-frequency analysis. The time-frequency resolution varies with scale, providing better localization for high-frequency components. [21]

Chapter 6

Numerical Setup and Methodology

6.1 Introduction

In this study, air is modeled as an ideal gas with a specific heat ratio of $\gamma = 1.4$, assuming a cold, initially stagnant flow at 300 K. This simplification enhances computational efficiency while preserving the accuracy of flow behavior analysis. The focus of this research is on the aeroacoustic characteristics of the exhaust jet, rather than the thermal effects of combustion.

Since the study examines the lift-off phase of a rocket launch, an overexpanded flow condition is simulated. During lift-off, the rocket exhaust exits into an environment where the ambient pressure exceeds the nozzle exit pressure, causing overexpansion. This scenario closely represents real launch conditions, making it a more practical choice than a perfectly expanded flow case.

The computational model was developed using the Design Modeler tool in ANSYS Fluent. The nozzle geometry was established starting with a throat radius of 1 cm, from which the inlet and exit sections were derived. The design follows a 15° half-angle conical nozzle configuration with a supersonic exit Mach number of 3. These parameters were selected to ensure a physically accurate nozzle flow while maintaining numerical stability.

The simulation assumes a stagnant inlet flow with an ambient pressure of 1 bar. The nozzle pressure ratio (NPR) is set to 10, corresponding to a stagnation pressure of 10 bar at the nozzle inlet.

6.2 Launcher Geometry Determination

To determine the dimension of the modelled launcher, we started from the nozzle. We employed isentropic flow relations, assuming a throat Mach number of $M_t = 1$ and an exit Mach number of $M_e = 3$. The mass flow rate through a generic section of the nozzle is given by:

$$\dot{m} = \sqrt{\gamma}\left(1 + \frac{\gamma - 1}{2} M^2\right)^{-\frac{\gamma+1}{2(\gamma-1)}} \frac{p_c A M}{\sqrt{R T_c}} \tag{6.1}$$

where:

- $\dot{m}$ is the mass flow rate,
- p_c is the combustion chamber pressure,
- A is the cross-sectional area at a given Mach number,
- M is the local Mach number,
- γ is the specific heat ratio (1.4 for air),
- T_c is the temperature in the combustion chamber,
- R is the specific gas constant.

Since the mass flow rate is conserved throughout the nozzle, at the throat where $M_t = 1$, the equation simplifies to:

$$\dot{m} = \Gamma \frac{p_c A_t}{\sqrt{R T_c}} \tag{6.2}$$

Combining equations 6.1 and 6.2, we obtain the relation for the required area expansion ratio to achieve a given Mach number:

$$\frac{A}{A_t} = \frac{1}{M}\left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma+1}{2(\gamma-1)}} \tag{6.3}$$

For $r_t = 1$ cm and $M = 1$ at the throat, and $M = 3$ at the exit, substituting these values yields the necessary expansion area ratio to determine the nozzle's exit radius, $r_e = 2.048$ cm. The nozzle length was then calculated assuming a 15° half-angle conical expansion to ensure proper flow expansion.

The flame deflector was placed at a distance of approximately 15 throat radii (r_t) downstream from the nozzle exit. This placement ensures a representative simulation of the impingement effects occurring during lift-off.

The model used in this study was designed to resemble the general shape of the Vega launcher, though it was not created as a perfectly scaled replica. The primary objective of this work was to focus on the methodologies for aeroacoustic analysis rather than achieving highly accurate quantitative results. However, instead of using a simple cylindrical geometry, it was preferred to model a shape that could plausibly represent a Vega-class launcher.

For the same reason, the nozzle within the first stage is not scaled proportionally to the actual vehicle. In reality, it is expected that the nozzle would be smaller relative to the internal volume of the first stage. Since the European Space Agency

(ESA) does not officially provide precise dimensional data on these components, the priority was given to designing a nozzle with realistic performance characteristics rather than an exact geometric reproduction of the launcher. The modeled rocket primarily serves as a reference object for Fourier and wavelet analysis, rather than a precisely scaled engineering model.

The main parameters of the problem are summarized in the table below.

Computational Domain Characteristics	**Value**
Throat Radius (r_t)	1 cm
Exit Radius (r_e)	2.058 cm
Inlet Radius (r_i)	1.426 cm
Nozzle Length (l_n)	4.0802 cm
Deflector Distance (d)	20.92 cm
Launcher Length (L)	42.67 cm
Launcher Radius (r_l)	2.67 cm

Table 6.1: Summary of Launcher Parameters

6.3 Computational Domain Definition and Mesh Generation

The initial phase of the study involved creating the axisymmetric computational domain. The geometry was constructed using Design Modeler in ANSYS Fluent, while the mesh generation was carried out with the ANSYS Fluent Meshing tool. Figure 6.1 presents the boundary conditions applied in the computational model. In this setup, the lower boundary of the domain serves as the symmetry axis, while the inclined flat plate, representing the impingement surface, is located on the right side of the figure.

A structured blocking strategy was applied to the geometry to accurately define the fluid domain, as depicted in Figure 6.2. The mesh was carefully partitioned to account for variations in the rocket's radius, ensuring increased resolution in critical regions while maintaining a computational cost that remains feasible. Although the mesh was structured and free of bias, this division allowed for greater precision in capturing flow features without excessive refinement.

The mesh consists of multiple blocks, with the highest refinement concentrated around the nozzle, followed by the region where the exhaust jet expands and subsequently interacts with the flame deflector. Due to the presence of the inclined ramp, the cells in the upper-right corner, where the sharp edge is located, exhibit higher distortion, which introduces some numerical viscosity in this region.

Additionally, four separate blocks are defined along the launcher's surface, with divisions occurring at the points of radius variation. Moving away from the nozzle, the mesh density gradually decreases while maintaining an acceptable aspect ratio to balance accuracy and computational efficiency. The minimum Element Quality recorded in the mesh is 0.47947, while the lowest Skewness value is 0.50052, both corresponding to the distorted cells near the upper-right edge of the ramp.

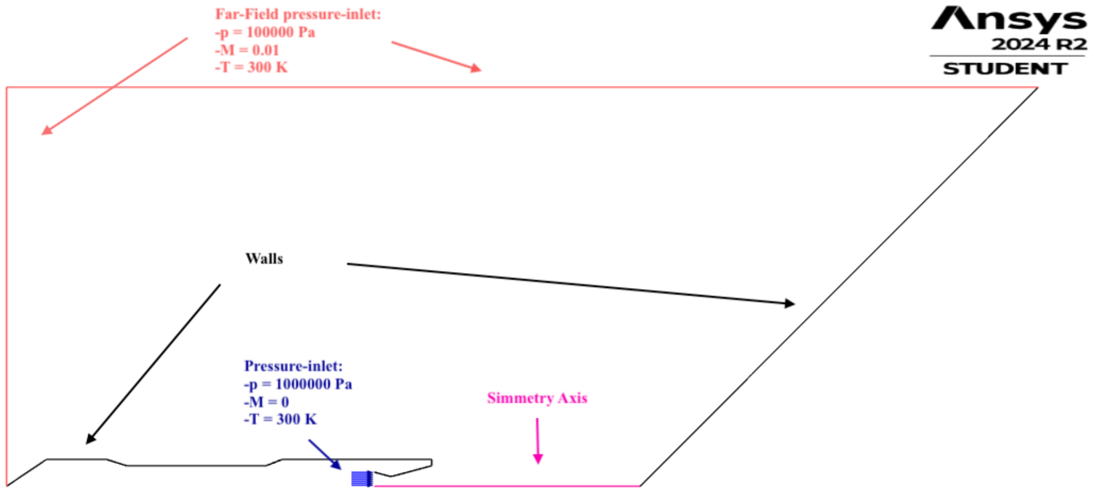


Figure 6.1: Boundary Conditions of the simulation

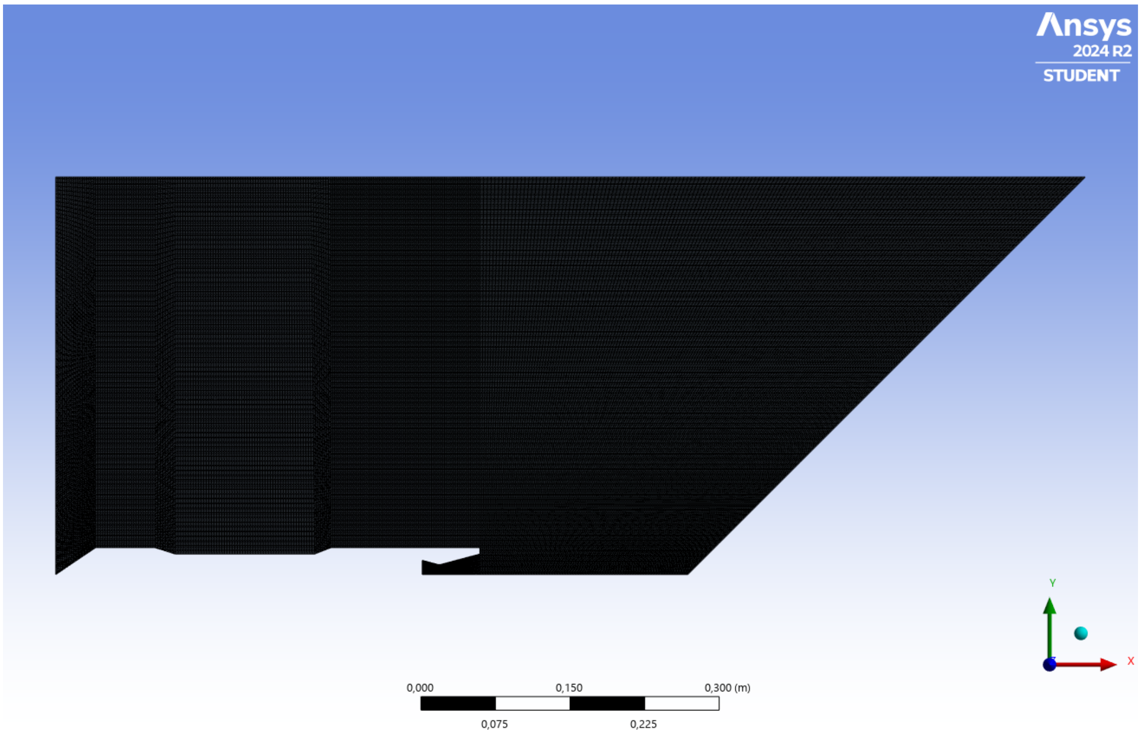


Figure 6.2: Mesh

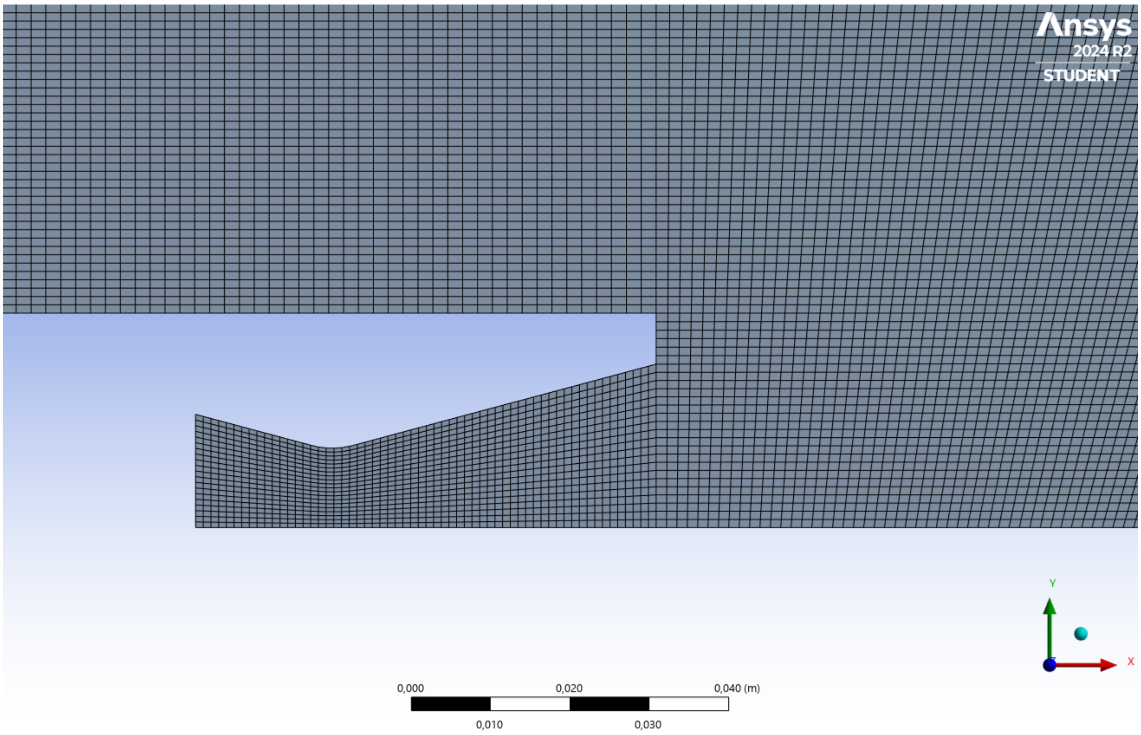


Figure 6.3: Mesh Details of Nozzle

6.4 Simulation Settings

In this study, the flow is assumed to be inviscid, meaning that viscous effects are neglected and the gas is treated as a single-phase ideal fluid. Under these assumptions, the governing equations used in the simulation are the Euler equations, which describe the behavior of an axisymmetric compressible flow.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{6.4}$$

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla p \tag{6.5}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho \mathbf{v} E) = -\nabla \cdot (p\mathbf{v}) \tag{6.6}$$

Since the flow is assumed to be single-phase and non-reacting, the generation terms in these equations are neglected. To solve these equations numerically, a Second Order Upwind discretization was applied for spatial accuracy, while Second Order Implicit discretization was used for time integration. This implicit formulation enhances numerical stability and ensures an accurate representation of acoustic wave propagation.

For flux calculation, the Roe Flux Differencing Scheme (Roe-FDS) was employed. The Roe scheme is based on solving a localized Riemann problem at each cell interface, ensuring a robust approximation of discontinuities, such as shock waves, while minimizing numerical diffusion. This method effectively captures compressible flow features without introducing excessive artificial viscosity, making it well-suited for high-speed aerodynamics problems.

A fixed time step of 5×10^{-6} s was used, and each case was simulated for a total physical time of 0.01 s, corresponding to 2000 time steps.

The computational parameters used in the simulations are summarized in Table 6.2.

Computational Methods	**Parameters**
Flow Model	Ideal, inviscid, non-reacting
Spatial Discretization	Second Order Upwind
CFL	1
Number of Elements	130675
Number of Faces	262144
Time Discretization	Implicit Second Order
Time Step	10^{-6} s
Total Simulation Time	0.01 s

Table 6.2: Summary of computational parameters used in the simulation.

In Figure 6.4, the time-averaged Mach contours are displayed in an axisymmetric view, providing a clearer and more intuitive representation of the flow behavior.

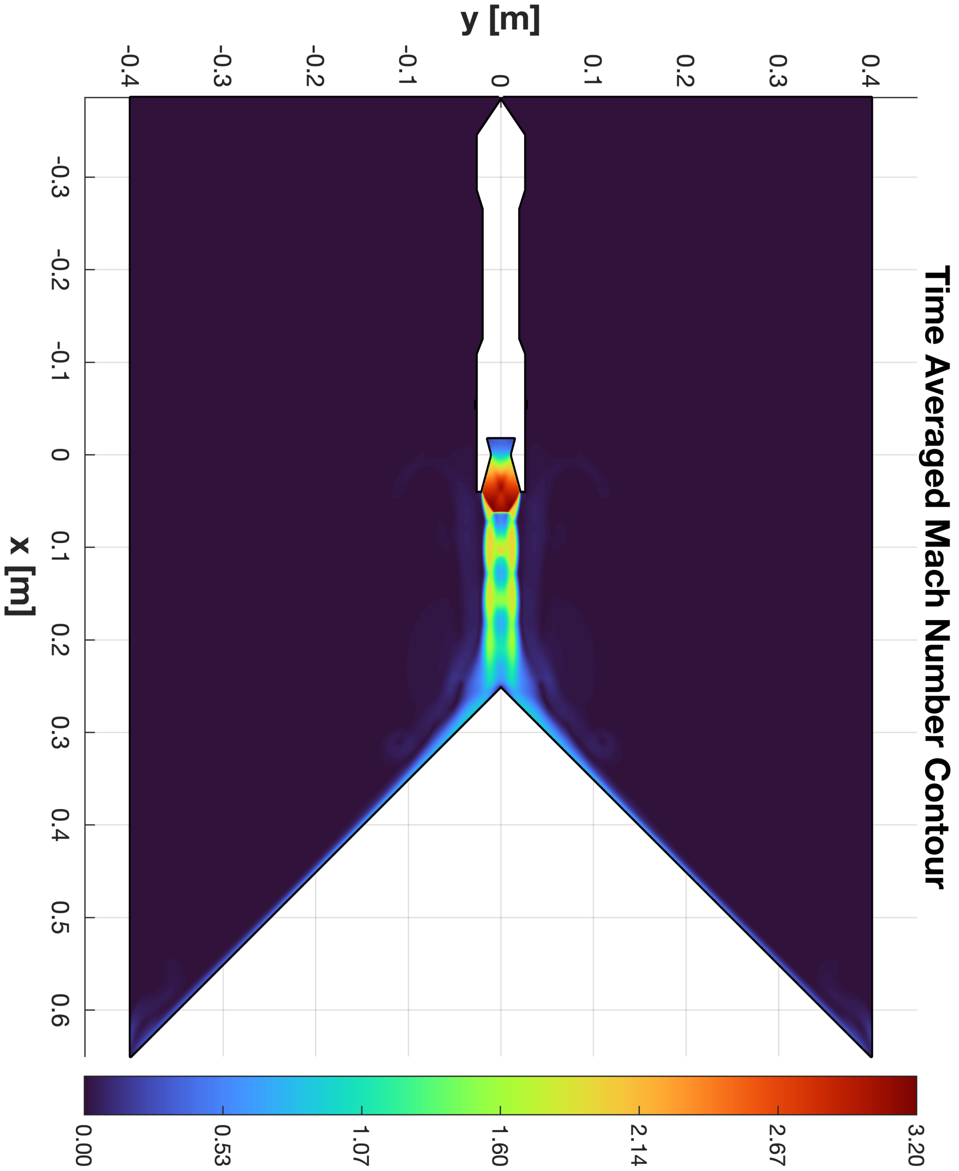


Figure 6.4: Time averaged Mach Number Contour

Chapter 7

Result Analysis

7.1 Analysis of Density Gradient Magnitude

In this section, the results of the simulation are presented and analyzed. The study begins with the examination of the density gradient magnitude, a crucial indicator of shock waves generated by the nozzle ignition and the subsequent impingement of the exhaust jet on the flame deflector. This interaction leads to shock reflections, making the visualization of the density gradient an essential tool for understanding the flow behavior.

To compute the density gradient magnitude, density data from ANSYS Fluent were recorded every three time steps. These cell-centered density values were then exported and post-processed in MATLAB using a Numerical Schlieren technique.

The Numerical Schlieren method is a post-processing technique that enhances the visibility of flow structures, particularly shock waves and turbulence regions, by highlighting regions of strong density gradients. This approach mimics traditional optical Schlieren imaging, which is widely used in experimental fluid dynamics to visualize variations in refractive index caused by density gradients.

In the numerical implementation, the density gradient is first computed using finite differences. The logarithm of the gradient magnitude is then taken to amplify weak density variations while maintaining the visibility of stronger shock features. This allows for a clearer representation of shock waves, expansion fans, and turbulent structures within the flow field.

By applying this method, the evolution of flow structures, such as the formation and propagation of shock waves and acoustic disturbances, can be accurately analyzed throughout the simulation.

The MATLAB script follows a structured approach for processing the exported density field:

1. **Domain Geometry and Data Import**
    - The script loads the boundary coordinates of the computational domain from an external file.

- The density field data is extracted from a CSV file containing simulation outputs.

2. **Grid Generation and Interpolation**

   - A regularized mesh of 2000 × 2000 points is created to resample the density values using linear interpolation.
   - A masking function is applied to remove points outside the domain, ensuring the visualization remains physically meaningful.

3. **Computation of the Density Gradient**

   - The script computes partial derivatives of the density field using finite differences to obtain density gradients in both x and y directions.
   - The gradient magnitude is then calculated to provide a scalar field representing shock intensity and flow discontinuities.

4. **Numerical Schlieren Implementation**

   - To enhance shock wave visualization, a logarithmic transformation of the gradient magnitude is applied, making weaker shock structures more visible.

5. **Plotting and Visualization**

   - The processed density gradient is displayed using an inverted grayscale colormap, mimicking traditional Schlieren imaging techniques used in experimental fluid dynamics.
   - The domain boundary is superimposed in blue to provide spatial reference, ensuring the visualization aligns with the computational setup.
   - Color scaling is automatically adjusted to optimize contrast, and a compact color bar is added to aid interpretation.

6. **Graph Export and High-Resolution Saving**

   - The final Numerical Schlieren image is saved as a high-resolution PNG file (800 dpi) for further analysis and documentation.

This post-processing methodology provides a detailed visualization of the shock wave structure, offering a clearer understanding of the interaction between the exhaust jet and the flame deflector at different time instants of the simulation.
The resulting visualizations are presented below:

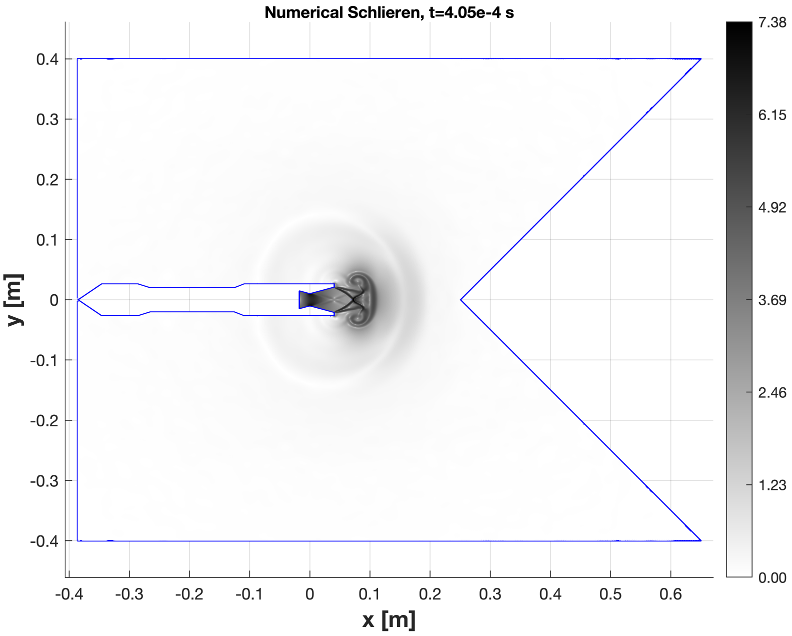


Figure 7.1: Density Gradient Magnitude contour at $t = 4.05 \times 10^{-4} s$. The first pressure wave generated by the rocket ignition is visible here.

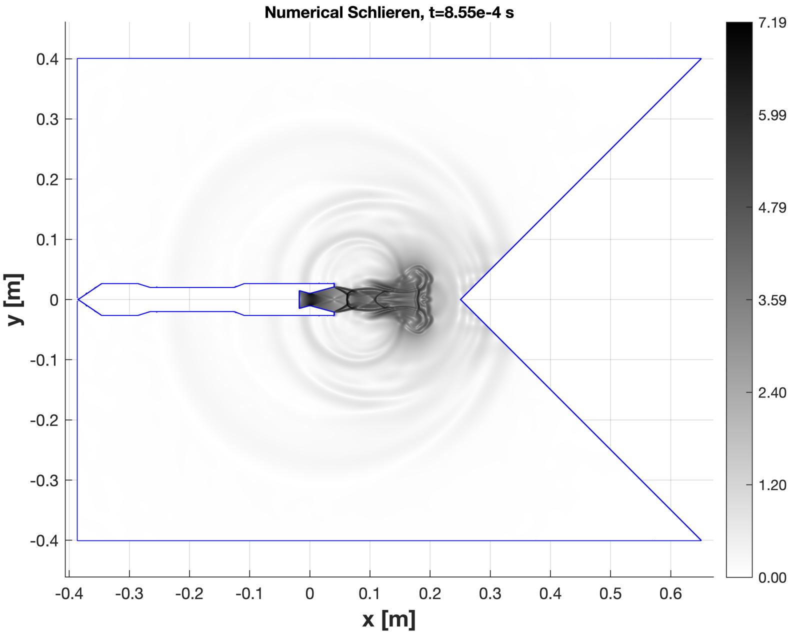


Figure 7.2: Density Gradient Magnitude contour at $t = 8.55 \times 10^{-4} s$. The pressure waves begin to impinge on the flame deflector.

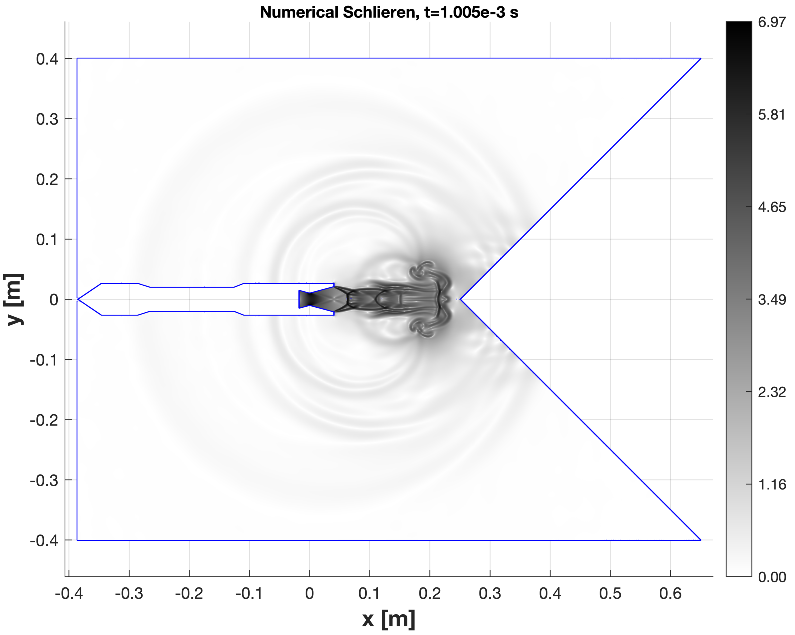


Figure 7.3: Density Gradient Magnitude contour at $t = 1.005 \times 10^{-3} s$. The interaction between impinging and reflected waves begins to emerge.

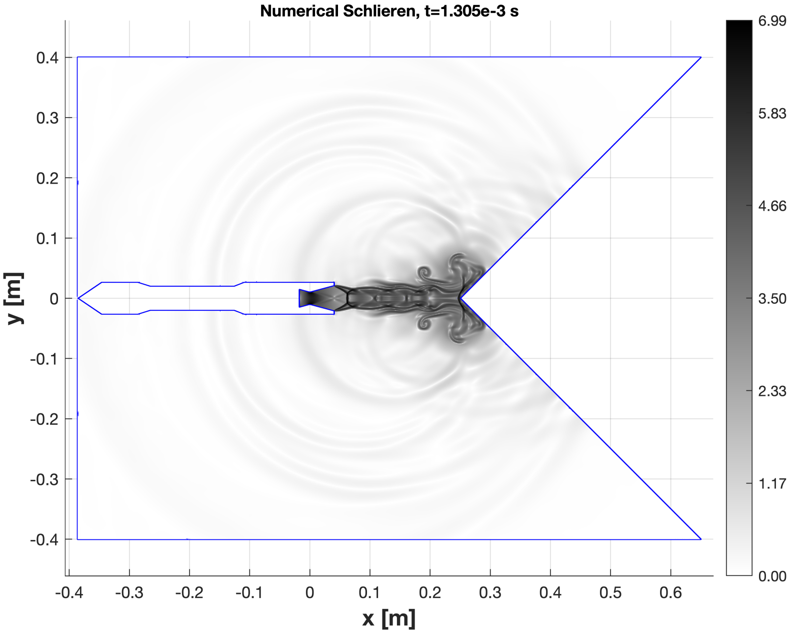


Figure 7.4: Density Gradient Magnitude contour at $t = 1.305 \times 10^{-3} s$. The interaction between impinging and reflected waves becomes more pronounced, while the formation of the Mach disk around the stagnation point becomes clearly appreciable.

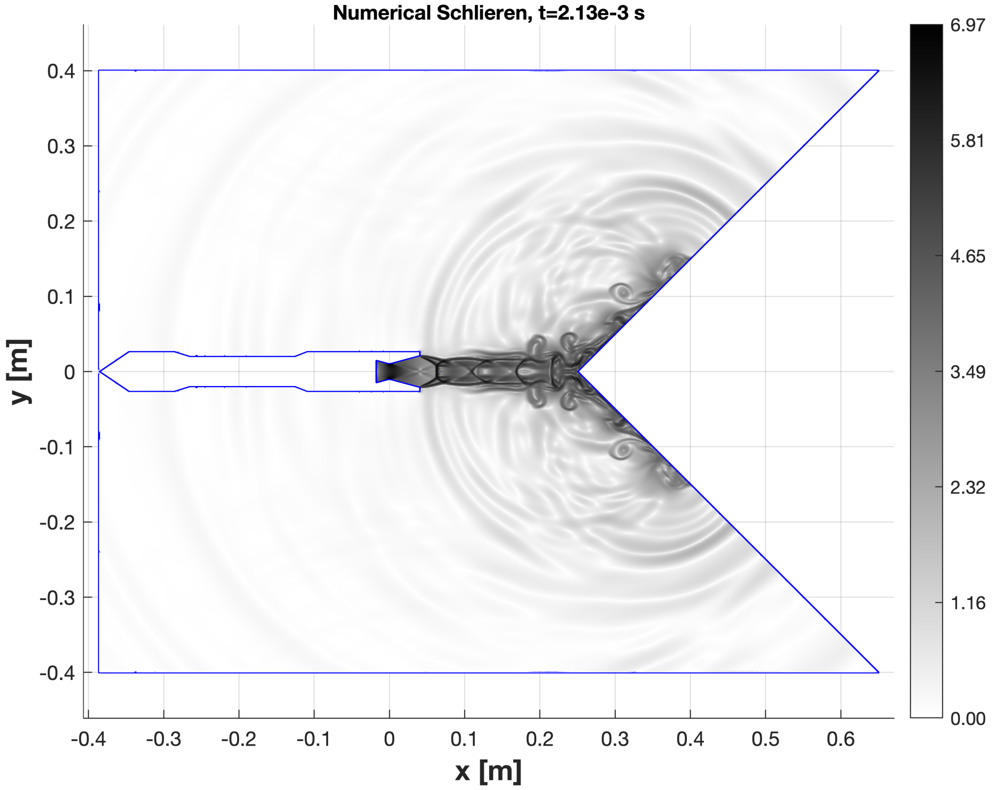


Figure 7.5: Density Gradient Magnitude contour at $t = 2.13 \times 10^{-3}s$. At this instant, the reflected acoustic waves are reaching the rocket and propagating further toward the launcher.

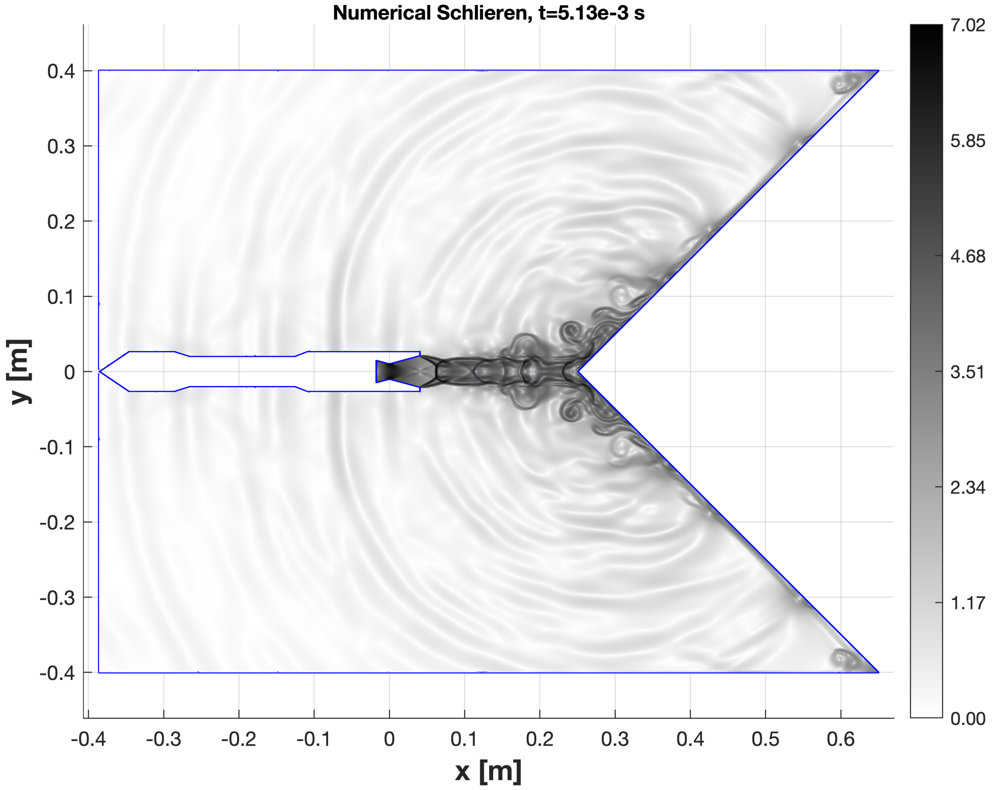


Figure 7.6: Density Gradient Magnitude contour at $t = 5.13 \times 10^{-3}s$. At this stage, the complex acoustic field is fully developed and clearly visible, with acoustic waves completely engulfing the launcher.

Figure 7.1, taken at $t = 4.05 \times 10^{-4}s$, illustrates the initial shock wave propagating from the nozzle, marking the beginning of the transient phase. In Figure 7.2, the first interactions between the expanding jet and the flame deflector ramp become visible.

As time progresses, Figure 7.3 reveals the reflection of shock waves off the ramp, a crucial step in the development of the flow structure. By Figure 7.4, at $t = 1.305 \times 10^{-3}s$, the formation of the Mach disk around the stagnation point can be observed, alongside the interaction between incident and reflected waves along the ramp. This interaction becomes even more evident in Figure 7.5, at $t = 2.13 \times 10^{-3}s$, where wave reflections and interferences intensify.

At this stage, approximately halfway through the simulation, Figure 7.5 clearly highlights the emergence of unsteady flow structures, generating additional wave formations. Plotting the Numerical Schlieren for this transient simulation allows us to identify the two main contributors to the acoustic field generation:

- The interaction of the jet with the flame deflector, which induces strong shock formations and wave reflections.

- The unsteadiness of the flow, as observed in Figure 7.5 and 7.6, which plays a significant role in turbulence-driven acoustic wave generation.

- The shedding and stretching of vortices within the shear layer of the plume, which would typically be absent in an inviscid framework. However, due to the inherent numerical dissipation of Euler-based solvers like ANSYS Fluent, this phenomenon is effectively reproduced, contributing to the overall acoustic emission.

7.2 Probes Analysis

To analyze the key characteristics of the acoustic field, multiple pressure probes were strategically placed throughout the computational domain to record time-series data of pressure fluctuations. The exact positions of these probes are marked as black crosses in Figure 7.7, illustrating their distribution within the domain.

- Probe 1 was placed on the P80 stage to monitor pressure fluctuations at the rocket body.
- Probes 2 and 3 were positioned between the nozzle and the flame deflector to capture the unsteady flow behavior in this critical interaction region.
- Probes 4, 5, and 6 were located on the flame deflector to track the evolution and propagation of shock waves across its surface.
- Probe 7 was placed on the fairing, as studying the sound pressure level (SPL) in this area is particularly important due to its role in structural and acoustic analysis.

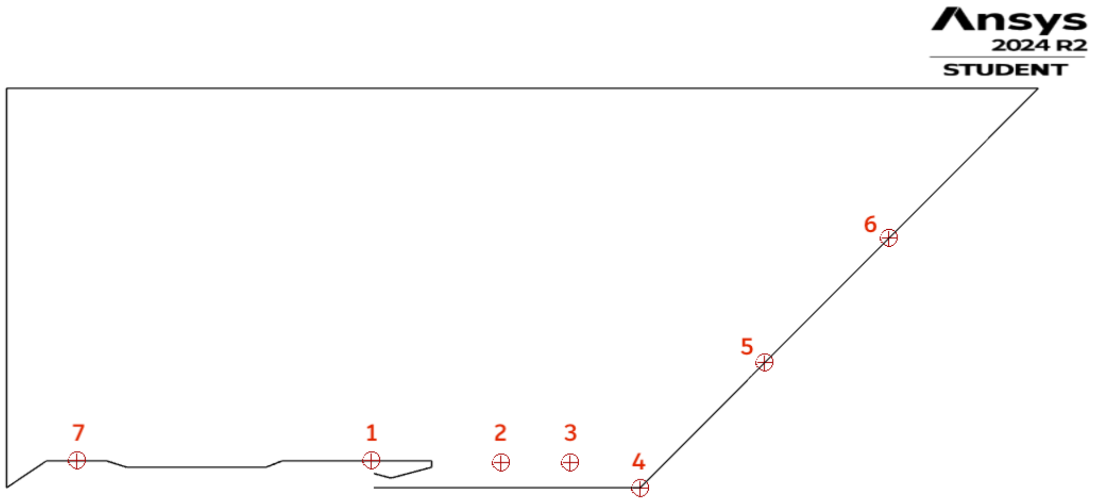


Figure 7.7: Probes position

With the probe locations established, we now proceed to analyze the recorded pressure data using both Fourier and Wavelet Transform techniques. The Fourier Transform is employed to decompose the pressure signals into their frequency components, providing insights into the dominant spectral content. The power spectral density (PSD) is computed using both Welch's method and the periodogram. Welch's method provides a smoothed and averaged estimate of the PSD, reducing

variance by segmenting the signal into overlapping windows and applying a windowing function. The periodogram, on the other hand, offers a higher frequency resolution by computing the PSD directly from the entire signal without averaging. Using both techniques allows for a more comprehensive spectral analysis, balancing the need for spectral resolution and statistical robustness.
The sound pressure level (SPL) is then derived as:

$$SPL(f) = 10 \log_{10} \left(\frac{P_{xx}(f)}{P_{ref}^2} \right) \tag{7.1}$$

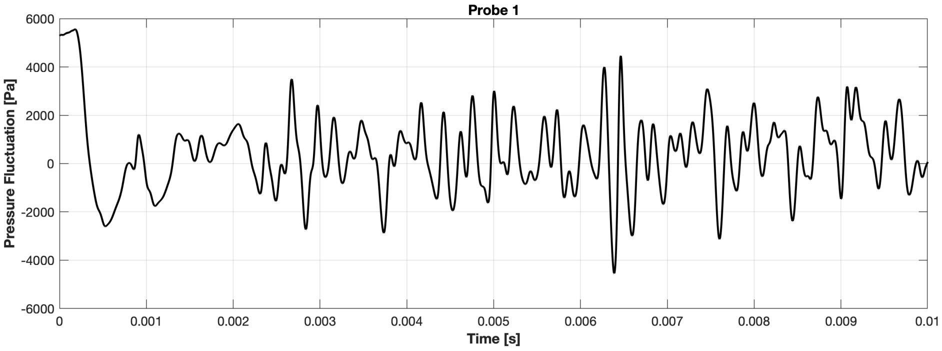
where $P_{xx}(f)$ represents the PSD of the pressure signal and P_{ref} is the reference pressure.

To complement this frequency-domain approach, a Wavelet Transform is applied using the Morlet wavelet function, which provides a time-frequency representation of the pressure fluctuations. The Continuous Wavelet Transform (CWT) is given by:

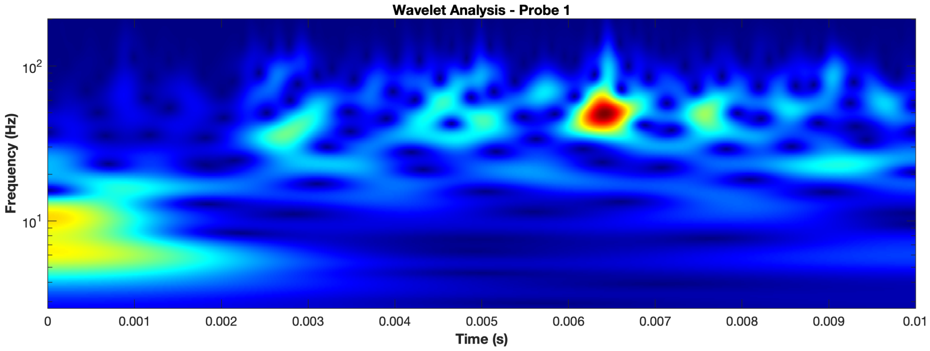
$$W(a, b) = \int_{-\infty}^{+\infty} p(t) \psi^* \left(\frac{t - b}{a} \right) dt \tag{7.2}$$

where a and b are the scale and translation parameters, and ψ is the mother wavelet. The wavelet decomposition allows us to track the evolution of the most energetic pressure waves over time, providing additional insight into transient acoustic phenomena. By combining these analyses, we can identify both the dominant frequencies and the temporal localization of high-energy acoustic events, improving the overall understanding of the aeroacoustic field generated during the simulation.

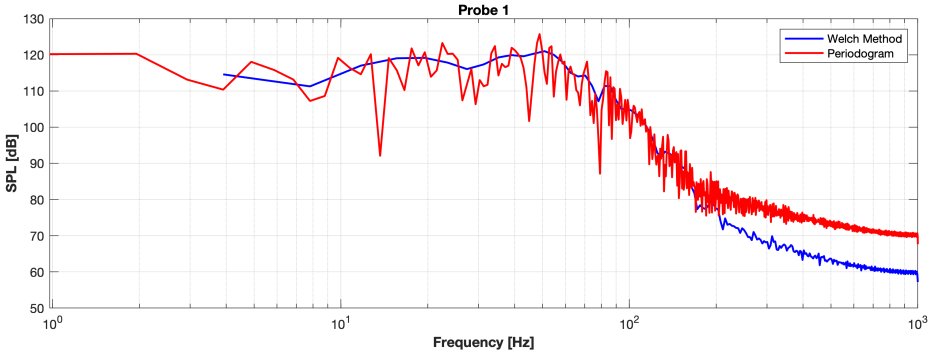
7.2.1 Probe 1



(a) Time evolution of fluctuating pressure measured by probe 1.

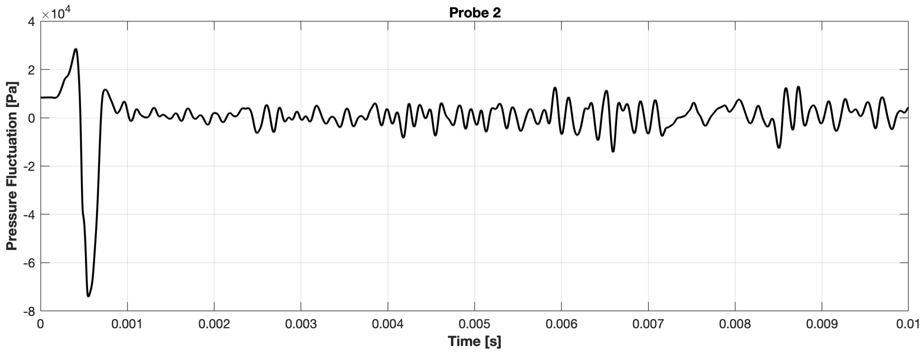


(b) Wavelet analysis of probe 1, showing the frequency distribution over time.

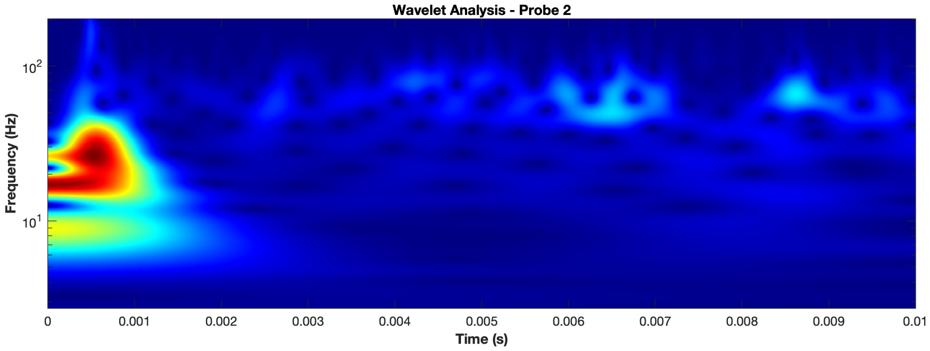


(c) Sound Pressure Level (SPL) measured by probe 1 as a function of frequency.

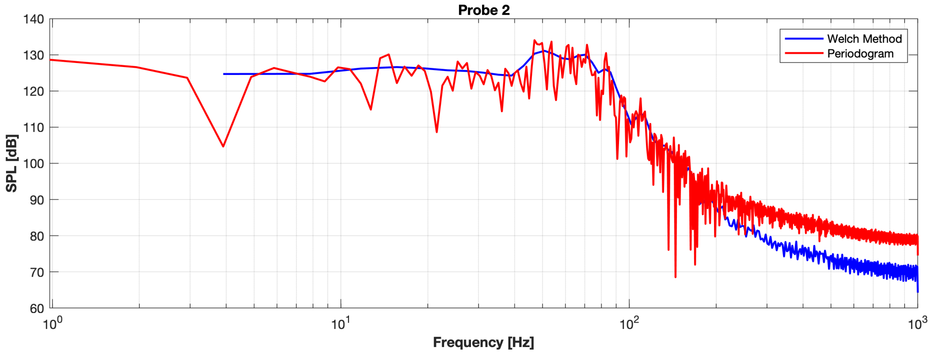
7.2.2 Probe 2



(a) Time evolution of fluctuating pressure measured by probe 2.

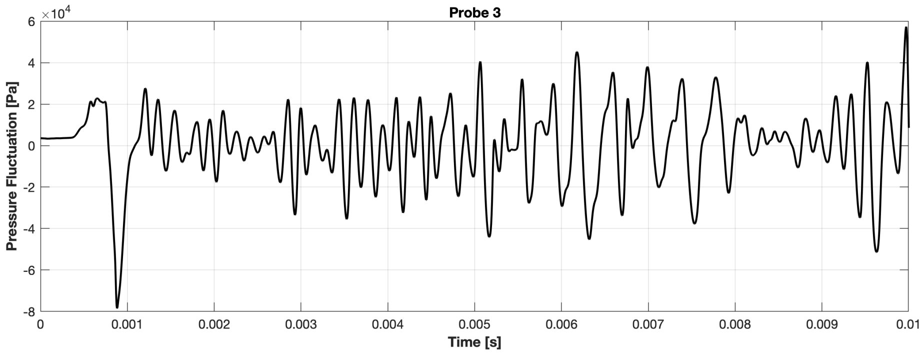


(b) Wavelet analysis of probe 2, showing the frequency distribution over time.

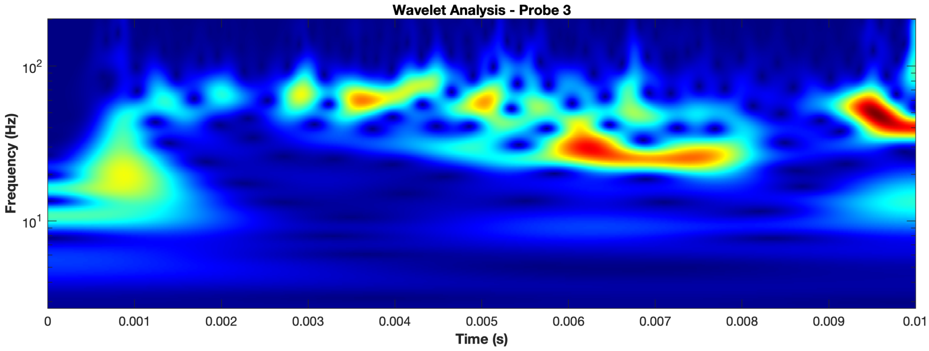


(c) Sound Pressure Level (SPL) measured by probe 2 as a function of frequency.

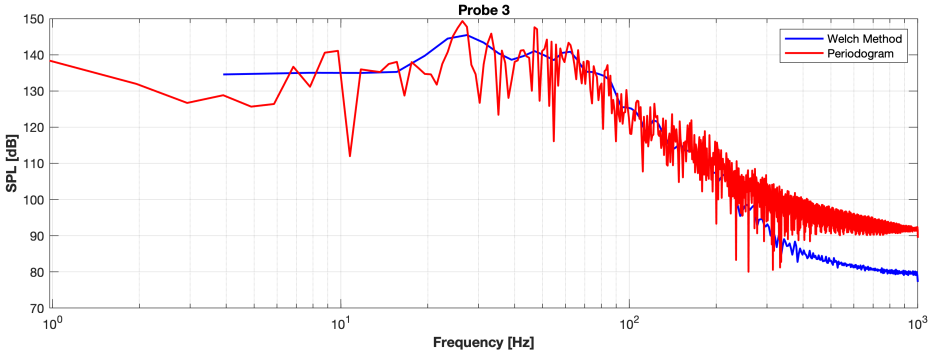
7.2.3 Probe 3



(a) Time evolution of fluctuating pressure measured by probe 3.

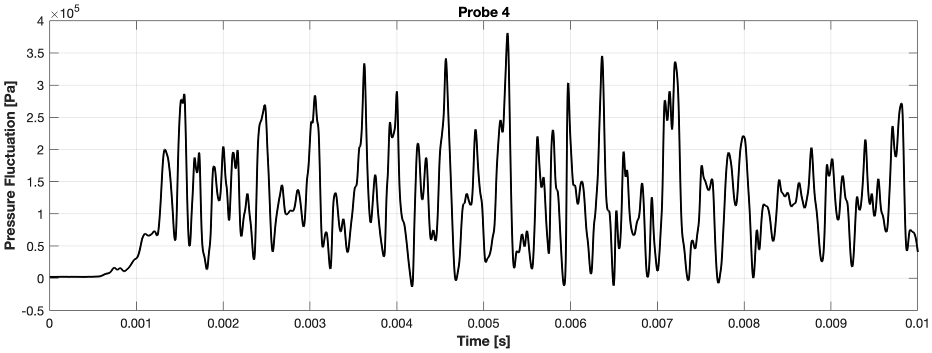


(b) Wavelet analysis of probe 3, showing the frequency distribution over time.

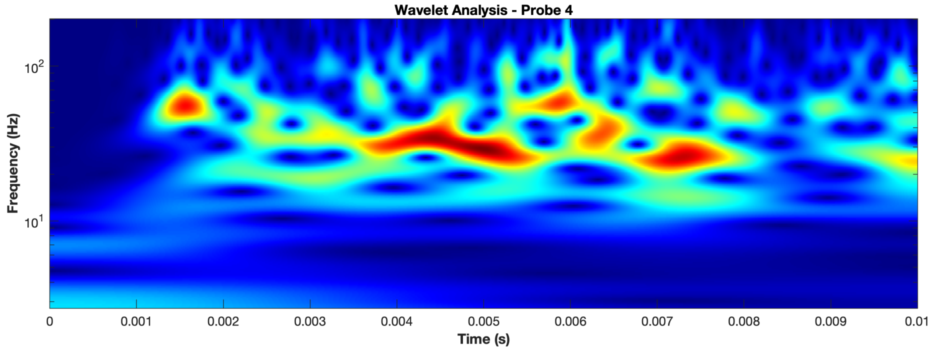


(c) Sound Pressure Level (SPL) measured by probe 3 as a function of frequency.

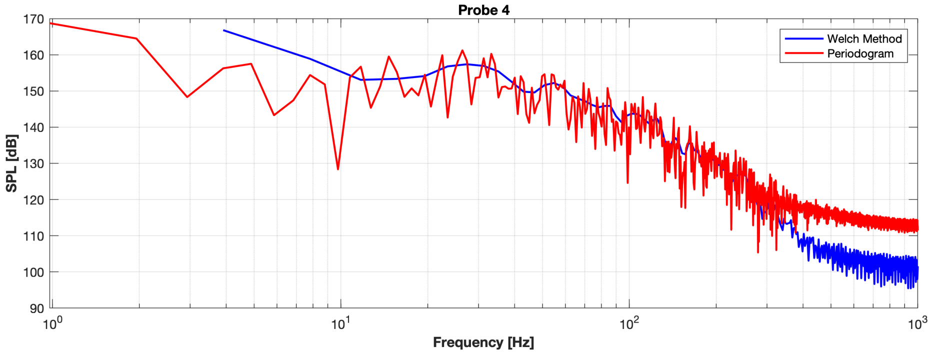
7.2.4 Probe 4



(a) Time evolution of fluctuating pressure measured by probe 4.

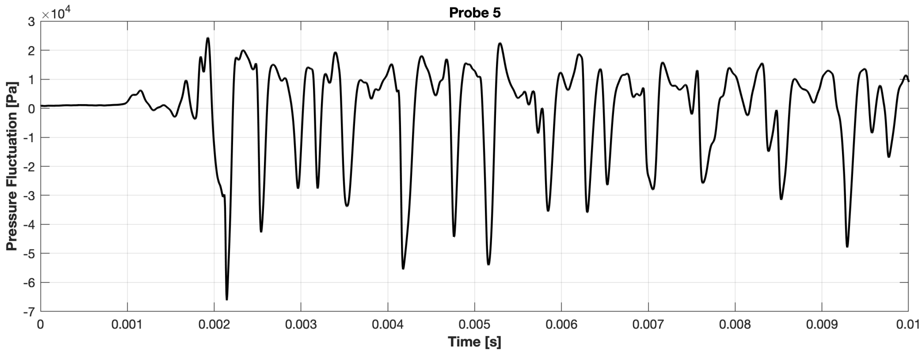


(b) Wavelet analysis of probe 4, showing the frequency distribution over time.

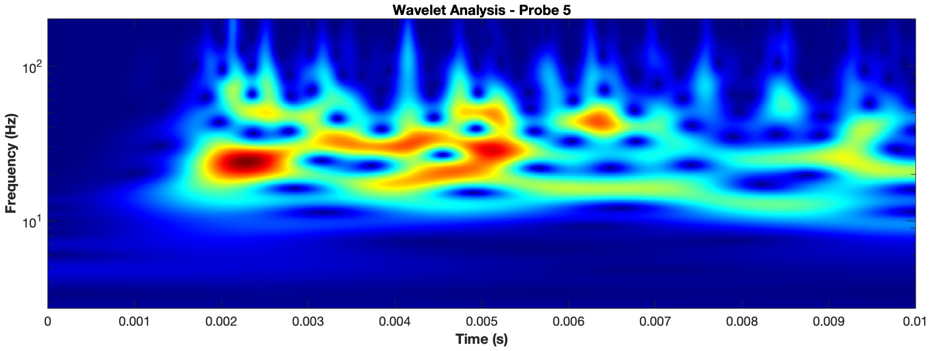


(c) Sound Pressure Level (SPL) measured by probe 4 as a function of frequency.

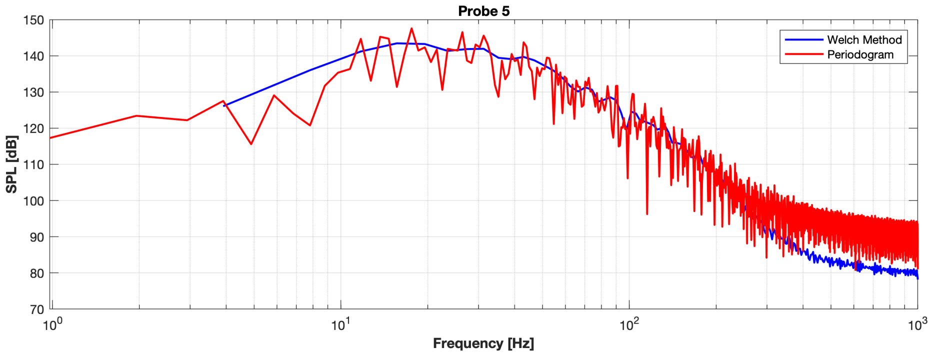
7.2.5 Probe 5



(a) Time evolution of fluctuating pressure measured by probe 5.

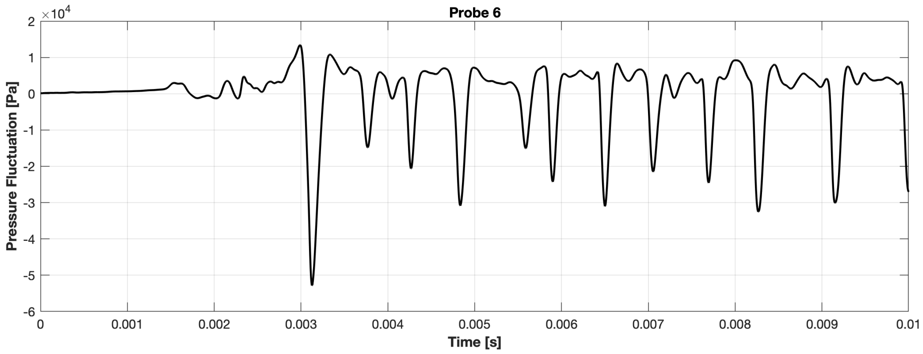


(b) Wavelet analysis of probe 5, showing the frequency distribution over time.

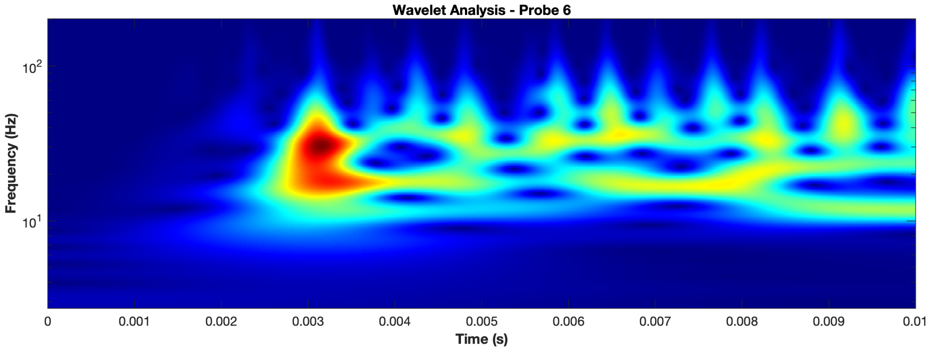


(c) Sound Pressure Level (SPL) measured by probe 5 as a function of frequency.

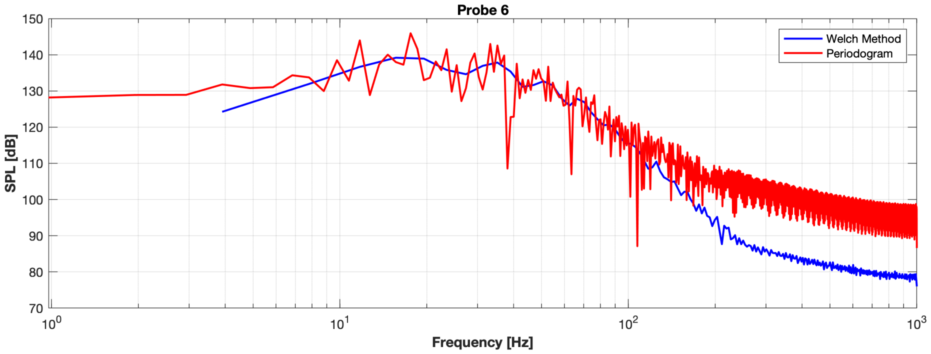
7.2.6 Probe 6



(a) Time evolution of fluctuating pressure measured by probe 6.

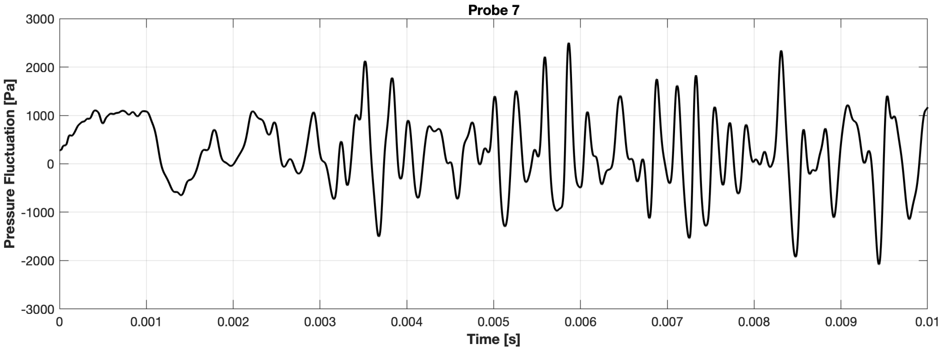


(b) Wavelet analysis of probe 6, showing the frequency distribution over time.

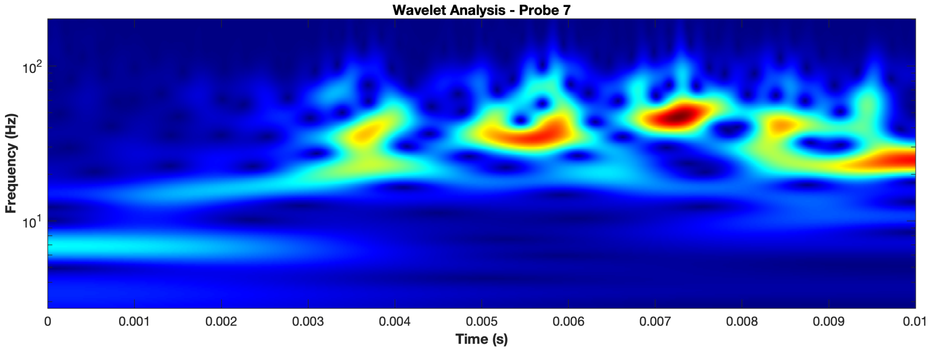


(c) Sound Pressure Level (SPL) measured by probe 6 as a function of frequency.

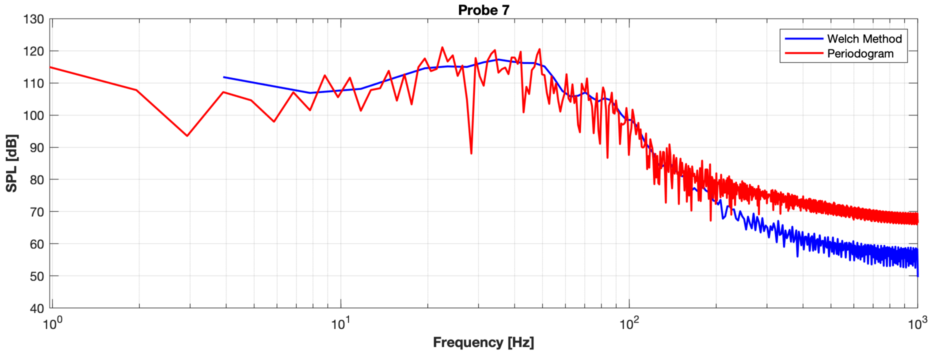
7.2.7 Probe 7



(a) Time evolution of fluctuating pressure measured by probe 7.



(b) Wavelet analysis of probe 7, showing the frequency distribution over time.



(c) Sound Pressure Level (SPL) measured by probe 7 as a function of frequency.

7.2.8 General Observations

From the spectral analysis of the pressure fluctuations recorded by the probes, we can derive significant insights into the aeroacoustic characteristics of the impinging supersonic jet. The combination of Fourier and Wavelet analysis provides a comprehensive understanding of the pressure loads and their evolution over time. The Fourier analysis provides a comprehensive frequency-domain representation of the pressure signals, allowing us to identify dominant spectral components. However, it does not capture the time-dependent nature of the fluctuations. The wavelet analysis, in contrast, enables time-frequency localization, allowing us to detect transient pressure events and their evolution over time.

7.2.9 Sound Pressure Levels (SPL) and Frequency Content

The SPL values derived from the recorded pressure signals reveal the intensity of the acoustic loads at different locations. The values observed in the flame deflector region (probes 4, 5, and 6) are particularly high, highlighting the strong interaction between the jet and the solid surface. These high SPL levels, especially in the low-frequency range, suggest that large-scale vortex shedding and shock reflections contribute significantly to the noise generation.

In contrast, the probes near the jet region (probes 1, 2, and 3) show dominant high-frequency components, indicative of turbulence and shear-layer instabilities. The presence of broadband noise in this region aligns with classical studies on supersonic jet aeroacoustics, where fine-scale turbulence and large-scale coherent structures contribute to the overall acoustic emission.

The fairing probe (probe 7) exhibits a combination of broadband noise and transient high-energy events, likely caused by the propagation of shock-induced waves from the flame deflector. This highlights the importance of studying acoustic loads in this region, as excessive pressure fluctuations can impact payload integrity.

7.2.10 Interpretation in Light of Aeroacoustic Theory

Based on the theoretical discussion on jet noise and impingement effects (Chapter 3), the observed frequency distribution corresponds well with expected physical mechanisms:

- The high-frequency content in the jet region is associated with turbulent mixing noise, as described in classical jet noise studies.
- The low-frequency dominance in the flame deflector region aligns with previous findings on impinging jets, where large-scale shock interactions and flow unsteadiness generate significant acoustic waves.
- The presence of discrete peaks in the Fourier spectrum, particularly in the flame deflector and fairing regions, suggests tonal noise contributions, which may originate from periodic vortex shedding or resonance effects.

7.2.11 Added Value of Wavelet Analysis

While the Fourier Transform provides a frequency decomposition of the signals, it lacks time localization. The wavelet analysis, using the Morlet wavelet, enhances our understanding by allowing us to track when specific frequency components appear during the simulation. The wavelet scalograms show that:

- In the jet region, high-frequency bursts occur intermittently, corresponding to transient turbulence structures.
- On the flame deflector, low-frequency energy spikes appear periodically, indicating the presence of unsteady shock reflections.
- On the fairing, high-energy events are detected at particular time instants, revealing when shock waves from the impinging jet reach this region.

From these results, it is evident that different regions experience distinct pressure fluctuation characteristics. The combined Fourier and wavelet analyses confirm that the dominant noise generation mechanisms in the domain are the jet-flame deflector interaction and the inherent unsteadiness of the flow. The SPL levels indicate significant acoustic loads, especially in the flame deflector region, emphasizing the importance of studying noise mitigation strategies. The flame deflector region is dominated by low-frequency shock interactions, while the jet and fairing regions exhibit a mix of broadband high-frequency noise and transient events. The use of both Fourier and wavelet analysis has allowed us to accurately identify and characterize these aeroacoustic phenomena, providing a comprehensive understanding of the pressure loads experienced during the simulated launch sequence.

7.2.12 Analysis of the Pressure Coefficient Distribution

To further understand the aerodynamic impact of the impinging jet, the pressure coefficient C_p was computed along both the flame deflector and the surface of the launcher. The pressure coefficient is defined for compressible flows as:

$$C_p = \frac{p - p_\infty}{0.5 \cdot \gamma M^2 p_\infty}, \tag{7.3}$$

where p is the local pressure, p_∞ is the free-stream pressure, γ is the ratio of specific heats, and M is the Mach number.

Flame Deflector The distribution of C_p along the flame deflector, shown in Figure 7.15, reveals a high-pressure region near the initial impact zone of the jet. This is expected due to the strong interaction between the impinging supersonic flow and the deflector surface, causing a localized increase in static pressure. Moving along the deflector, the pressure coefficient rapidly decreases, indicating that the jet expands and loses energy as it travels downstream. The reduction in C_p suggests that the strongest aerodynamic forces are concentrated at the initial impingement region, while the downstream region is characterized by weaker interactions and reduced pressure loads.

The calculation of C_p in this region is essential for evaluating the structural integrity of the deflector, as excessive pressure loads could lead to material stress, deformation, or increased acoustic emissions. Additionally, the observed pressure variations may contribute to the generation of secondary shock waves and acoustic disturbances that propagate throughout the domain.

Launcher Surface The computed C_p values along the launcher surface, presented in Figure 7.16, exhibit significantly lower magnitudes compared to the flame deflector. The pressure coefficient is nearly zero, confirming that the launcher is not subject to direct aerodynamic loads from the jet. The small oscillations observed in the C_p distribution may be attributed to weak acoustic waves or minor pressure fluctuations in the surrounding flow field.

This result is consistent with expectations, as the primary interaction occurs between the jet and the flame deflector, while the launcher primarily experiences indirect effects such as acoustic loading rather than direct aerodynamic forces.

The comparison between these two regions highlights the importance of studying the flame deflector's pressure distribution, as it is the main structural component exposed to significant aerodynamic loads. Understanding this behavior is crucial for optimizing deflector design to mitigate pressure-induced stresses and minimize secondary acoustic emissions.

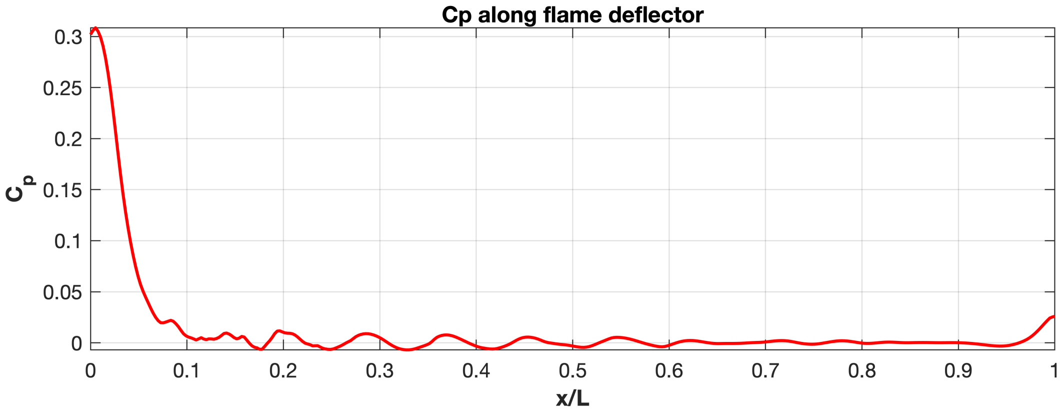


Figure 7.15: Time-averaged Cp along the deflector

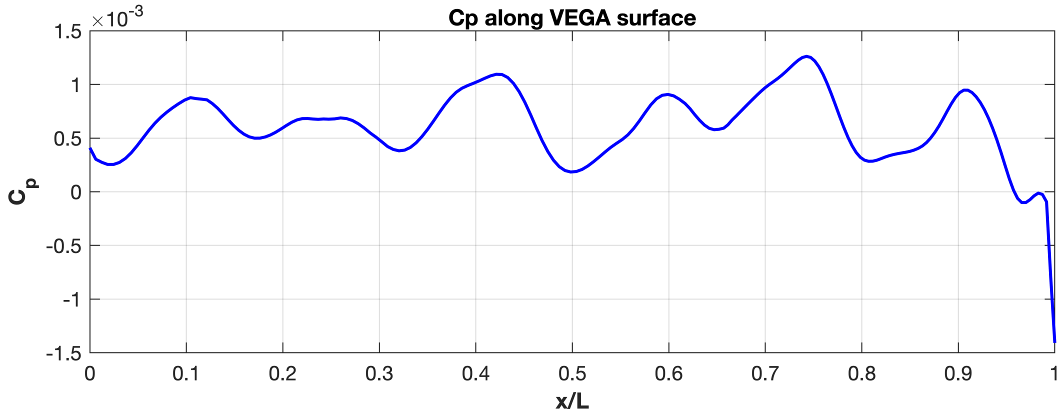


Figure 7.16: Time-averaged Cp along launcher surface

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This study presented a numerical investigation of the aeroacoustic field generated during the lift-off phase of a supersonic rocket jet impinging on a flame deflector. Using a relatively simple inviscid CFD model, we were able to capture the main flow structures, including shock waves, Mach disks, and the interaction between incident and reflected waves, which play a significant role in the generation of the acoustic field.

The post-processing analysis based on Numerical Schlieren visualization highlighted the formation and evolution of shock waves and their reflection on the flame deflector. Furthermore, the Fourier and Wavelet analyses of the pressure probes provided insight into the frequency content of the acoustic waves and their temporal evolution, offering a complementary approach for studying unsteady flow characteristics.

Despite the simplicity of the model, the results obtained demonstrate the effectiveness of computational tools in analyzing jet noise and its interaction with the surrounding structures. This work serves as a preliminary step toward more refined aeroacoustic studies, providing a foundation for future developments.

8.2 Potential Developments

While this study has provided valuable insights, several improvements and further investigations can enhance the accuracy and applicability of the results. Some key future developments include:

- Flame Deflector Optimization: Investigating different shapes and inclination angles for the flame deflector to assess their impact on wave reflection and noise mitigation.

- Varying Nozzle Pressure Ratio (NPR): Exploring different NPR values to analyze their influence on the expansion and shock structure of the jet, providing a more comprehensive characterization of launch conditions.

- Implementation of Turbulence Models: Including RANS, LES, or hybrid turbulence models to better capture vortex dynamics and shear-layer instabilities, which are crucial contributors to jet noise.

- Comparisons with Experimental Data: Validating the numerical results with available experimental datasets to assess the accuracy of the simplified approach and identify areas for improvement.

- Parametric Study on Jet and Deflector Interaction: Extending the analysis to different jet configurations, such as multi-nozzle arrangements, to evaluate their effect on noise generation.

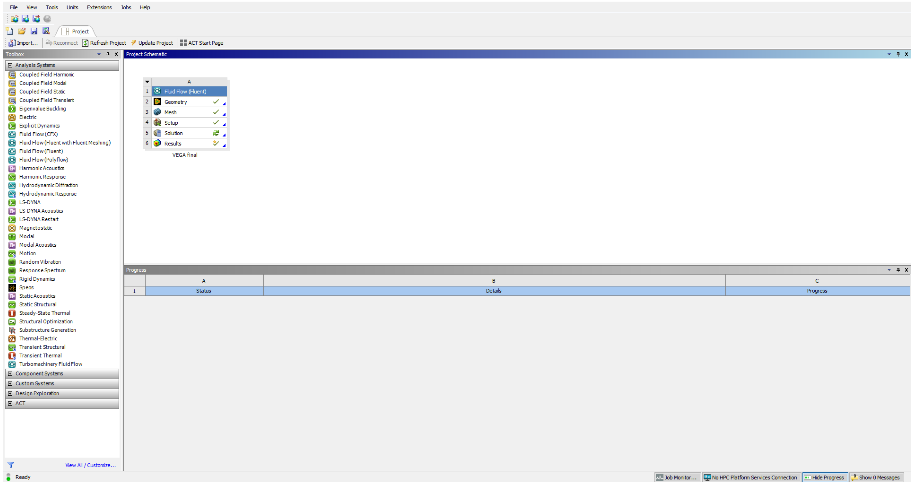
- Including Acoustic Treatment Strategies: Testing the effect of sound suppression techniques (e.g., water injection, deflector modifications) to assess their capability in reducing the acoustic loads on the launcher and surrounding infrastructure.

This work provides a solid starting point for further research in launch vehicle acoustics, demonstrating that even a simplified numerical approach can offer valuable insights into the physics of jet impingement noise. The integration of more advanced numerical models and experimental validation will further refine our understanding and contribute to the design of more efficient and quieter launch systems.

Appendix A: Simulation Workflow in ANSYS Fluent

A.1 Project Initialization and Geometry Creation

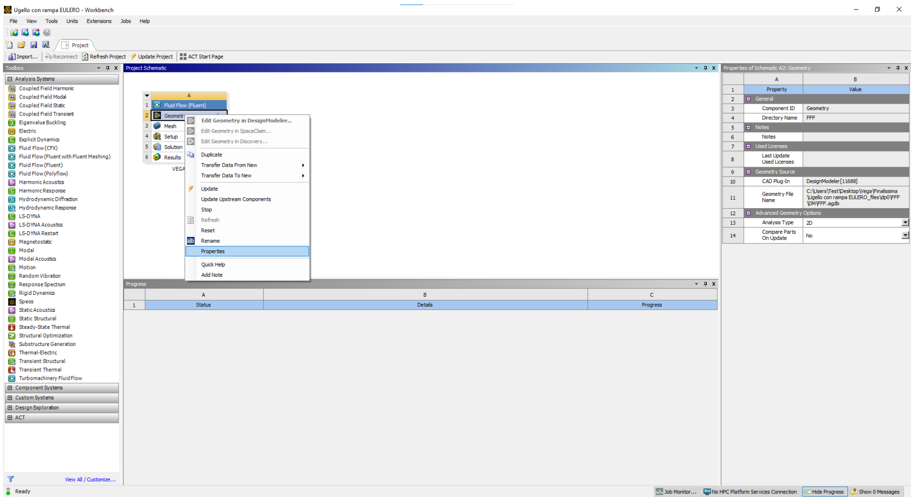
Once ANSYS Workbench is launched, the first step is to add a new Fluent analysis system. This is done by double-clicking on **Fluid Flow (Fluent)** under the *Analysis Systems* menu located on the left panel of the interface. As shown in Figure 8.1, this action will populate the project schematic with six components.



	A
1	Fluid Flow (Fluent)
2	Geometry ✓
3	Mesh ✓
4	Setup ✓
5	Solution
6	Results

VEGA final

	A	B	C
1	Status	Details	Progress

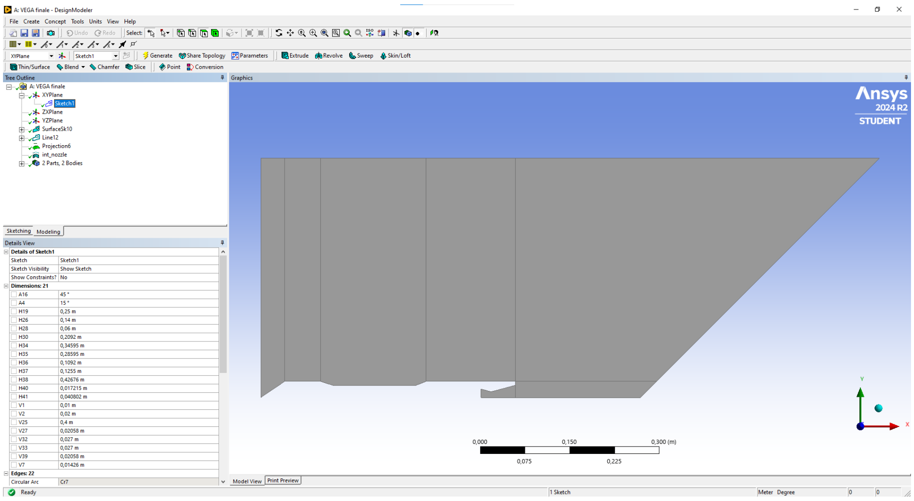


Properties of Schematic A2: Geometry

	A	B
1	Property	Value
2	General	
3	Component ID	Geometry
4	Directory Name	FFF
5	Notes	
6	Notes	
7	Used Licenses	
8	Last Update Used Licenses	
9	Geometry Source	
10	CAD Plug-In	DesignModeler[11688]
11	Geometry File Name	C:\Users\Test\Desktop\Vega\Finalissima \Ugello con rampa EULERO_files\dp0\FFF \DM\FFF.agdb
12	Advanced Geometry Options	
13	Analysis Type	2D
14	Compare Parts On Update	No

A.2 Geometry Setup in DesignModeler

At this stage, the required geometry might already be available, either as a text-based file or as an external CAD model. However, in the case of this thesis, the geometry was created directly within the Fluent environment.



Details View

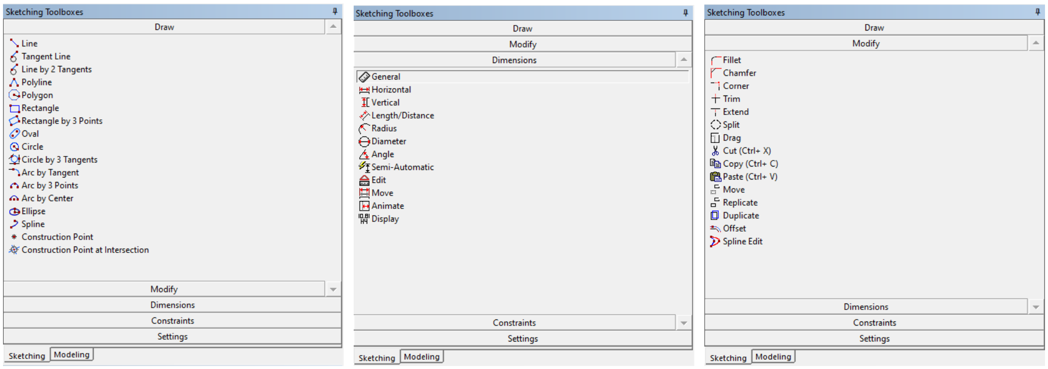
Details of Sketch1	
Sketch	Sketch1
Sketch Visibility	Show Sketch
Show Constraints?	No
Dimensions: 21	
□ A16	45 °
□ A4	15 °
□ H19	0,25 m
□ H26	0,14 m
□ H28	0,06 m
□ H30	0,2092 m
□ H34	0,34595 m
□ H35	0,28595 m
□ H36	0,1092 m
□ H37	0,1255 m
□ H38	0,42676 m
□ H40	0,017215 m
□ H41	0,040802 m
□ V1	0,01 m
□ V2	0,02 m
□ V25	0,4 m
□ V27	0,02058 m
□ V32	0,027 m
□ V33	0,027 m
□ V39	0,02058 m
□ V7	0,01426 m
Edges: 22	
Circular Arc	Cr7

To do this, double-click on the **Geometry** cell of the analysis system. Fluent provides three different tools for geometry creation: *DesignModeler*, *SpaceClaim*, and *Discovery*. For this work, *DesignModeler* was selected, and the tutorial will proceed using this tool.

Before proceeding, it is important to right-click on the **Geometry** cell and select **Properties**. In the panel that appears on the right side of the screen, set the *Analysis Type* to **2D** (Figure 8.2).

Once DesignModeler is opened, switch to the *XY plane*. This can be done either by selecting it from the tree menu on the left or by clicking on the *Z-axis* (the blue axis) in the triad located at the bottom-right corner of the screen.

To begin drawing, click on the **Sketching** tab located under the *Tree Outline* panel on the left. DesignModeler provides tools to create both simple lines and more complex shapes. A recommended approach is to start with a base sketch using the **Polyline** tool. This allows you to draw a series of connected straight-line segments by clicking at each desired corner point. To finish the polyline, right-click and choose either *Open End* or *Closed End*, depending on whether you want to leave the contour open or closed.



DesignModeler assists the user by indicating when a segment is aligned horizontally or vertically with respect to the reference plane, as well as when it intersects an existing line.

Note that to pan across the sketching plane, you should hold down **CTRL** and the mouse wheel. Pressing only the mouse wheel will shift the view across planes. If the view accidentally exits the 2D reference plane, you can realign it by clicking again on the **Z-axis** in the bottom-right triad.

A.2 Geometry Setup in DesignModeler

Before proceeding, it is important to right-click on the **Geometry** cell and select **Properties**. In the panel that appears on the right side of the screen, set the *Analysis Type* to **2D**.

Once DesignModeler is opened, switch to the *XY plane*. This can be done either by selecting it from the tree menu on the left or by clicking on the *Z-axis* (the blue axis) in the triad located at the bottom-right corner of the screen.

To begin drawing, click on the **Sketching** tab located under the *Tree Outline* panel on the left. DesignModeler provides tools to create both simple lines and more complex shapes. A recommended approach is to start with a base sketch using the **Polyline** tool. This allows you to draw a series of connected straight-line segments

by clicking at each desired corner point. To finish the polyline, right-click and choose either *Open End* or *Closed End*, depending on whether you want to leave the contour open or closed.

DesignModeler assists the user by indicating when a segment is aligned horizontally or vertically with respect to the reference plane, as well as when it intersects an existing line.

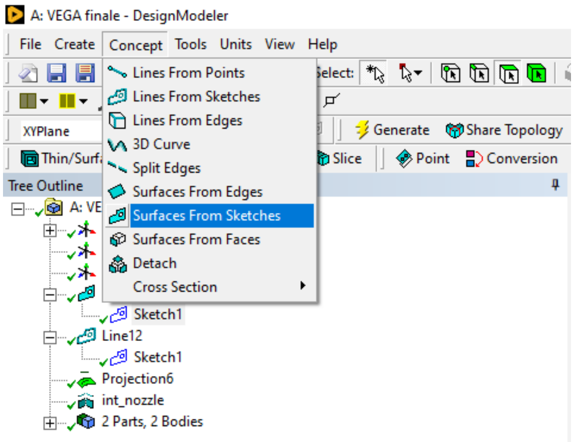
Note that to pan across the sketching plane, you should hold down **CTRL** and the mouse wheel. Pressing only the mouse wheel will shift the view across planes. If the view accidentally exits the 2D reference plane, you can realign it by clicking again on the **Z-axis** in the bottom-right triad.

Once the geometry has been defined, it can be dimensioned directly by clicking on the **Dimensions** option available in the *Sketching Toolboxes* on the left. In addition to specifying the lengths of line segments, it is also possible to define angle openings, diameters, and other geometric constraints.

If any part of the sketch is incorrect and needs to be redrawn, the **Modify** tool can be used to delete specific elements.

After completing the base sketch, click the **Generate** button, marked with a lightning bolt icon. This action generates the sketch, which will then appear in the *Tree Outline* under the **XYPlane** entry. If the sketch does not generate correctly, the issue is often due to an open contour—make sure the sketch is fully closed to proceed. When the sketch is properly generated, a green check mark should appear next to **Sketch1**.

At this point, the next step is to create the surface. To do this, go to the top menu and select **Concept** > **Surfaces from Sketches**. This will open the *Details View* panel at the bottom right. From there, select the sketch by clicking it in the *Tree Outline*, then click **Apply**, followed by **Generate**. The surface should now appear in the *Tree Outline*.



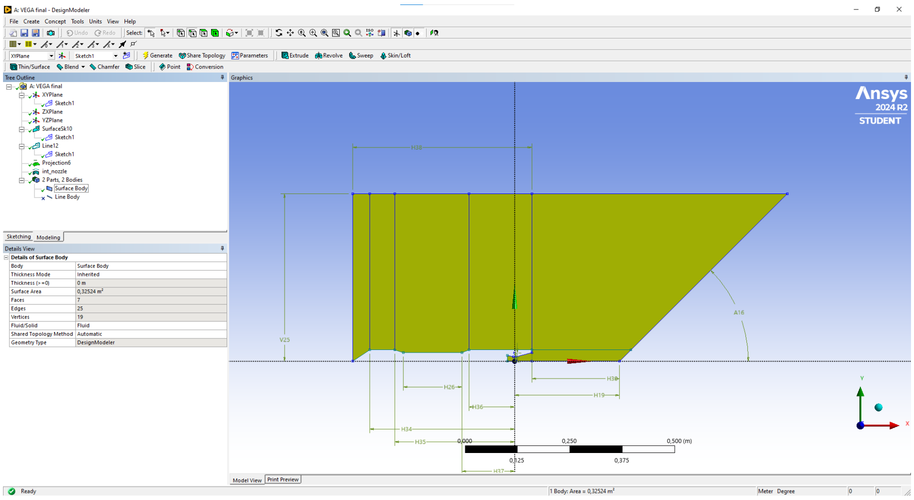
For convenience, especially in preparation for a multiblock mesh, the surface can be divided into subsurfaces. This is useful for better mesh control in complex geometries. To do this, draw additional lines over the surface in the regions where subdivision is desired. Once these lines are defined, go to **Concept** > **Lines from Sketches** to finalize the separation lines.

At this stage, after selecting the lines and the corresponding sketch, a new entry labeled **Line1** will appear in the *Tree Outline*. After clicking **Generate**, a green check mark will confirm its successful creation.

To actually split the surface, go to the top menu and click on **Tools** > **Projection**. In the lower-left panel, the software will prompt you to select the target edges—these correspond to the recently drawn lines—and the surface to be split.

It is worth noting that in the top toolbar, four cube icons are available, each highlighting a different element: a vertex, an edge, a face, and the entire body. Selecting the appropriate one helps streamline the selection process when working with complex geometries.

Once the projection is applied, the surface will be correctly subdivided. In the *Tree Outline*, a new entry labeled **2 Parts, 2 Bodies** should appear. Expanding this entry will reveal two sub-items: **Surface Body** and **Line Body**.



Details of Surface Body	
Body	Surface Body
Thickness Mode	Inherited
Thickness (>=0)	0 m
Surface Area	0,32524 m²
Faces	7
Edges	25
Vertices	19
Fluid/Solid	Fluid
Shared Topology Method	Automatic
Geometry Type	DesignModeler

1 Body: Area = 0,32524 m²

Make sure that when selecting **Surface Body**, the **Body Type** is set to *Fluid*, not *Solid*. Additionally, right-click on **Line Body** and select **Suppress Line Body** to avoid displaying unnecessary line details in later steps.

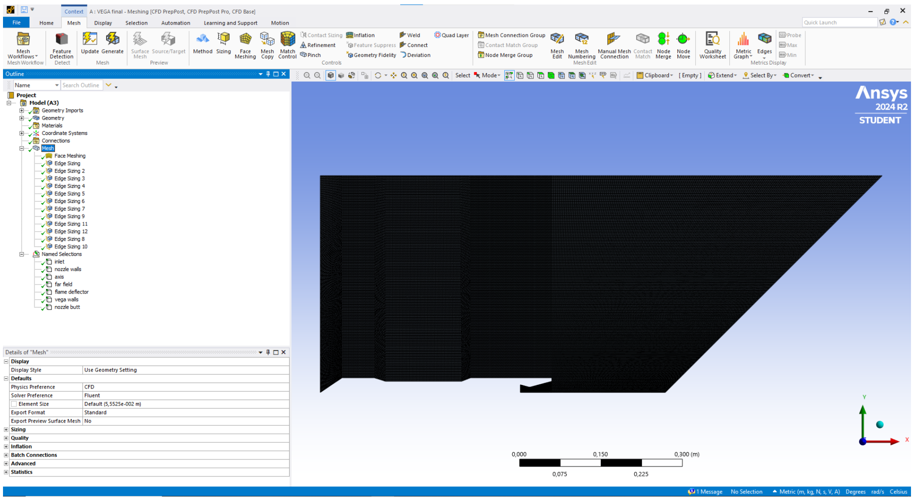
At this point, after verifying that all entries in the *Tree Outline* have green check marks (which indicates that each operation has been successfully generated), you can safely close DesignModeler.

A.3 Structured Meshing in ANSYS Mesher

At this point, you can proceed by opening the **Mesh** module from the Fluent workflow.

To correctly generate a structured mesh, the first step is to select the **Mesh** entry in the Outline menu on the left, highlight all the surfaces into which the geometry has been divided, and then click on the **Face Meshing** icon in the top toolbar. This operation ensures that, unless errors are present, all subsequent subdivisions of the mesh will maintain a structured configuration.

As always, it is important to click **Generate** after each modification to visualize the result.



A : VEGA final - Meshing [CFD PrepPost, CFD PrepPost Pro, CFD Base]

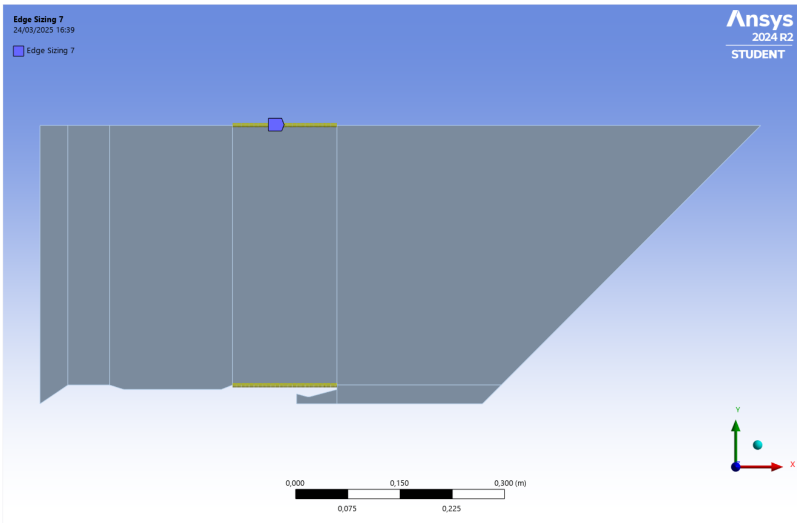
Outline

- Project
  - Model (A3)
    - Geometry Imports
    - Geometry
    - Materials
    - Coordinate Systems
    - Connections
    - Mesh
      - Face Meshing
      - Edge Sizing
      - Edge Sizing 2
      - Edge Sizing 3
      - Edge Sizing 4
      - Edge Sizing 5
      - Edge Sizing 6
      - Edge Sizing 7
      - Edge Sizing 9
      - Edge Sizing 11
      - Edge Sizing 12
      - Edge Sizing 8
      - Edge Sizing 10
    - Named Selections
      - inlet
      - nozzle walls
      - axis
      - far field
      - flame deflector
      - vega walls
      - nozzle butt

Details of "Mesh"

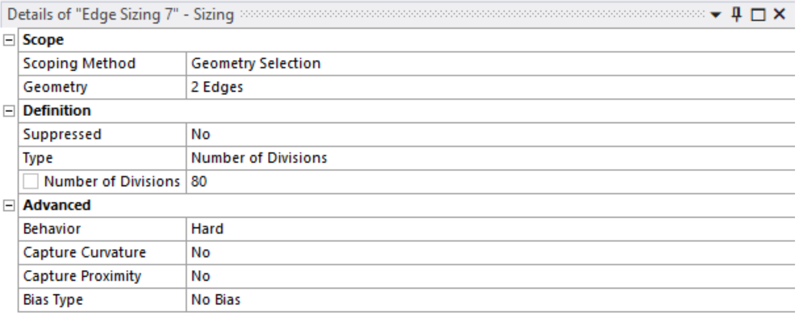
Display	
Display Style	Use Geometry Setting
Defaults	
Physics Preference	CFD
Solver Preference	Fluent
□ Element Size	Default (5,5525e-002 m)
Export Format	Standard
Export Preview Surface Mesh	No
Sizing	
Quality	
Inflation	
Batch Connections	
Advanced	
Statistics	

The most effective method—also used for this thesis—is to mesh the individual surfaces using the **Edge Sizing** tool. To do this, enable the edge selection mode, click on the desired edges, and then click on **Sizing**. This will open a panel at the bottom left where you can configure the mesh settings.



Although it is possible to define the mesh by specifying the element size, it is recommended to use the **Number of Divisions** option, which allows for finer or coarser meshing depending on the desired resolution.

It is also advised to double-click the **Behavior** setting until it reads **Hard**, and to set **Capture Curvature** to **No**. These settings should be applied to all created Edge Sizing entries.



Details of "Edge Sizing 7" - Sizing

⊟ **Scope**	
Scoping Method	Geometry Selection
Geometry	2 Edges
⊟ **Definition**	
Suppressed	No
Type	Number of Divisions
□ Number of Divisions	80
⊟ **Advanced**	
Behavior	Hard
Capture Curvature	No
Capture Proximity	No
Bias Type	No Bias

In this specific case, since the simulation was performed using an inviscid fluid model, it was not necessary to refine the mesh near the walls to resolve the boundary layer, which is absent. However, if such refinement had been needed, one could have enabled the **Bias Type** option and selected an appropriate biasing method. This allows the mesh to be refined near selected edges using a defined **Growth Rate**.

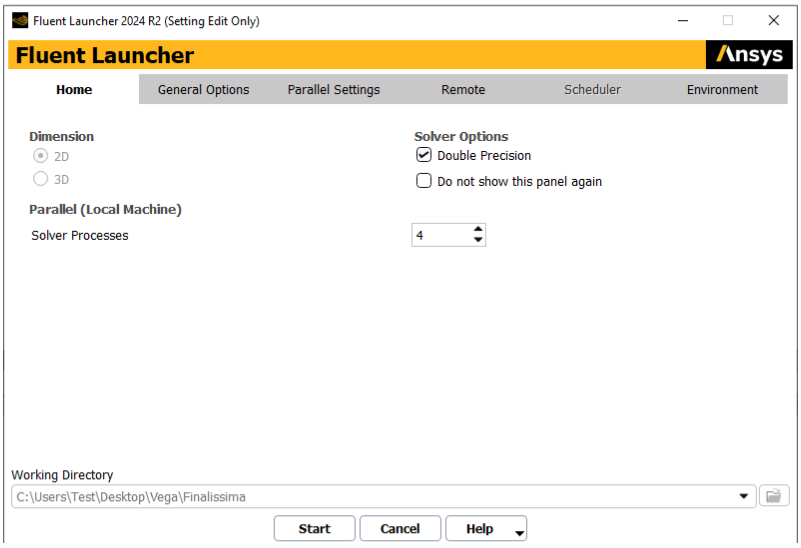
Once the mesh has been fully generated, its quality metrics can be inspected by selecting **Mesh** in the Outline menu and choosing a desired indicator from the **Display** section.

Before closing the Mesh module, it is good practice to assign names to the boundary edges of the domain. This facilitates the definition of boundary conditions later in the setup process.

To assign names, right-click on **Named Selection**, then select the desired edges and assign them appropriate labels. It is recommended to use standard names such as *"Inlet"*, *"Outlet"*, *"Farfield"*, and *"Axis"*, as Fluent can automatically recognize these labels and pre-fill the corresponding boundary condition types during the solver setup. This streamlines the configuration process and reduces the likelihood of errors.

A.4 Solver Setup and Simulation Configuration in Fluent

After completing the meshing process, close the meshing module and double-click on the **Setup** cell in the Fluent workflow. A window will appear asking for the number of solver processes to be used in parallel. Once configured, click **Start** to launch Fluent.



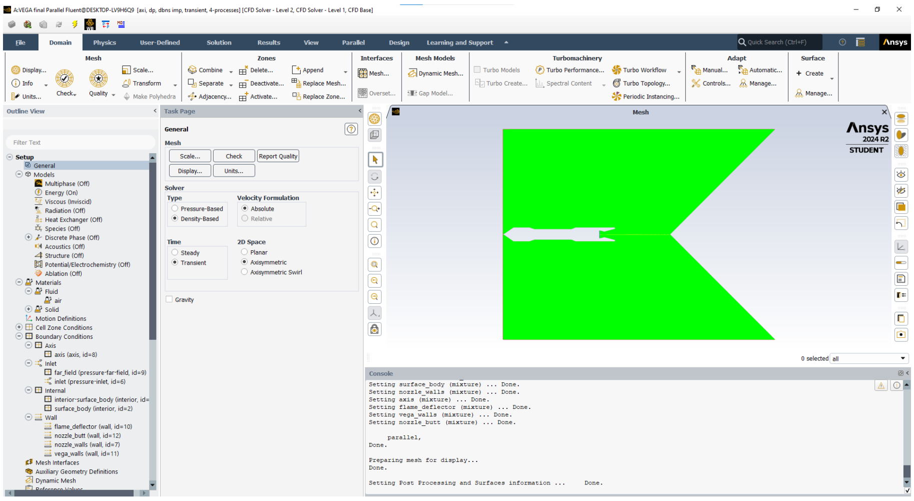
Dimension
○ 2D
○ 3D

Parallel (Local Machine)
Solver Processes 4

Solver Options
☑ Double Precision
□ Do not show this panel again

Working Directory
C:\Users\Test\Desktop\Vega\Finalissima

The first step in the *Outline View* is to select **Density-Based** solver and activate the **Axisymmetric** option. In this thesis, a **Transient** model was used, meaning that the simulation is driven by a total time duration rather than a predefined number of iterations.



A:VEGA final Parallel Fluent@DESKTOP-LV9H6Q9 [axi, dp, dbns imp, transient, 4-processes] [CFD Solver - Level 2, CFD Solver - Level 1, CFD Base]

Filter Text

- Setup
  - General
  - Models
    - Multiphase (Off)
    - Energy (On)
    - Viscous (Inviscid)
    - Radiation (Off)
    - Heat Exchanger (Off)
    - Species (Off)
    - Discrete Phase (Off)
    - Acoustics (Off)
    - Structure (Off)
    - Potential/Electrochemistry (Off)
    - Ablation (Off)
  - Materials
    - Fluid
      - air
    - Solid
  - Motion Definitions
  - Cell Zone Conditions
  - Boundary Conditions
    - Axis
      - axis (axis, id=8)
    - Inlet
      - far_field (pressure-far-field, id=9)
      - inlet (pressure-inlet, id=6)
    - Internal
      - interior-surface_body (interior, id=
      - surface_body (interior, id=2)
    - Wall
      - flame_deflector (wall, id=10)
      - nozzle_butt (wall, id=12)
      - nozzle_walls (wall, id=7)
      - vega_walls (wall, id=11)
  - Mesh Interfaces
  - Auxiliary Geometry Definitions
  - Dynamic Mesh
  - Reference Values

General

Mesh
Scale... Check Report Quality
Display... Units...

Solver

Type
○ Pressure-Based
● Density-Based

Velocity Formulation
● Absolute
○ Relative

Time
○ Steady
● Transient

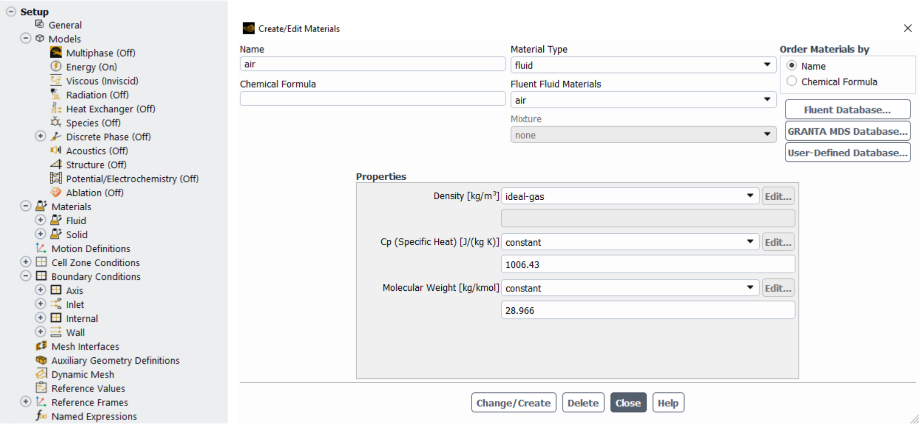
2D Space
○ Planar
● Axisymmetric
○ Axisymmetric Swirl

□ Gravity

```
Setting surface_body (mixture) ... Done.
Setting nozzle_walls (mixture) ... Done.
Setting axis (mixture) ... Done.
Setting flame_deflector (mixture) ... Done.
Setting vega_walls (mixture) ... Done.
Setting nozzle_butt (mixture) ... Done.

     parallel,
Done.

Preparing mesh for display...
Done.

Setting Post Processing and Surfaces information ...     Done.
```

Next, double-click on the **Energy** entry in the menu and set it to **Yes** to include the energy conservation equation in the simulation. Although it is possible to select a turbulence model at this stage, in this study the flow was considered **Inviscid**, and no turbulence model was activated.

Proceed to **Materials** > **Fluid** > **Air**. In the properties window, set the density model to **Ideal-gas**.



Create/Edit Materials

Name: air

Chemical Formula:

Material Type: fluid

Fluent Fluid Materials: air

Mixture: none

Order Materials by: ● Name ○ Chemical Formula

Properties

Density [kg/m^3]: ideal-gas

Cp (Specific Heat) [J/(kg K)]: constant, 1006.43

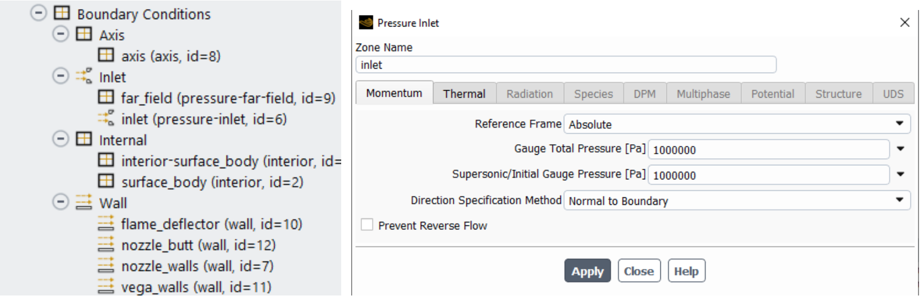
Molecular Weight [kg/kmol]: constant, 28.966

A.4.1 Boundary Conditions

Thanks to the earlier boundary naming, some conditions may already be assigned correctly. To modify, for example, the inlet conditions, double-click on the **Inlet** boundary. Before doing so, right-click on the entry and set the type to **Pressure-Inlet**. In the properties window, you can then define the total pressure, static pressure, and inlet temperature.

Similarly, right-click on the **Farfield** boundary and set its type to **Pressure Far-Field**. You can then specify the far-field pressure conditions. It is recommended to set a small but non-zero Mach number (e.g., 0.01).

Ensure that the axis of symmetry is correctly set under the **Axis** boundary and that any edges in contact with solid surfaces are classified as **Wall**.



Boundary Conditions
Axis
axis (axis, id=8)
Inlet
far_field (pressure-far-field, id=9)
inlet (pressure-inlet, id=6)
Internal
interior-surface_body (interior, id=
surface_body (interior, id=2)
Wall
flame_deflector (wall, id=10)
nozzle_butt (wall, id=12)
nozzle_walls (wall, id=7)
vega_walls (wall, id=11)
Momentum
Thermal
Radiation
Species
DPM
Multiphase
Potential
Structure
UDS
Apply
Close
Help

Pressure Inlet

Zone Name: inlet

Reference Frame: Absolute

Gauge Total Pressure [Pa]: 1000000

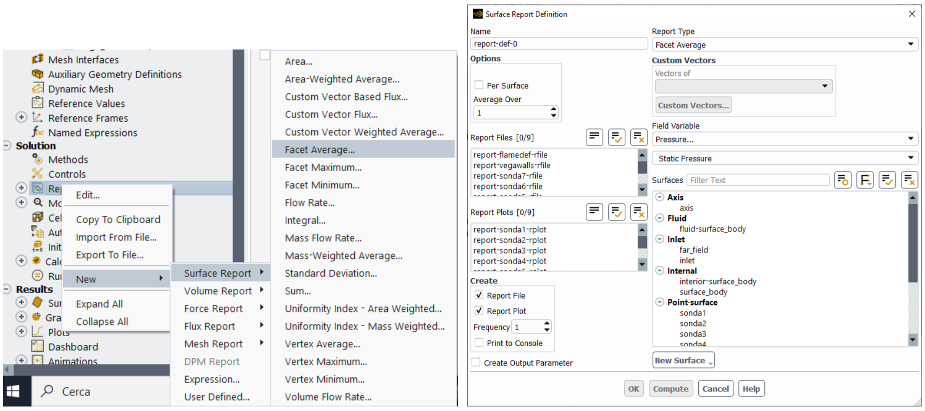
Supersonic/Initial Gauge Pressure [Pa]: 1000000

Direction Specification Method: Normal to Boundary

□ Prevent Reverse Flow

A.4.2 Probes and Data Monitoring

At this point, it is also possible to insert probes to record data at specific points during the simulation. To do so, right-click on **Report Definitions** > **New** > **Surface Reports** > **Facet Average**. In the window that appears, you can assign a name to the report, select an existing surface or create a new one (e.g., a point probe) by clicking **New Surface**. Then choose the quantity of interest and confirm with **OK**.



Surface Report Definition

Name: report-def-0

Report Type: Facet Average

Options: □ Per Surface; Average Over: 1

Custom Vectors: Vectors of: (blank)

Report Files [0/9]: report-flamedef-rfile, report-vegawalls-rfile, report-sonda7-rfile, report-sonda6-rfile

Report Plots [0/9]: report-sonda1-rplot, report-sonda2-rplot, report-sonda3-rplot, report-sonda4-rplot

Create: ☑ Report File; ☑ Report Plot; Frequency: 1; □ Print to Console

□ Create Output Parameter

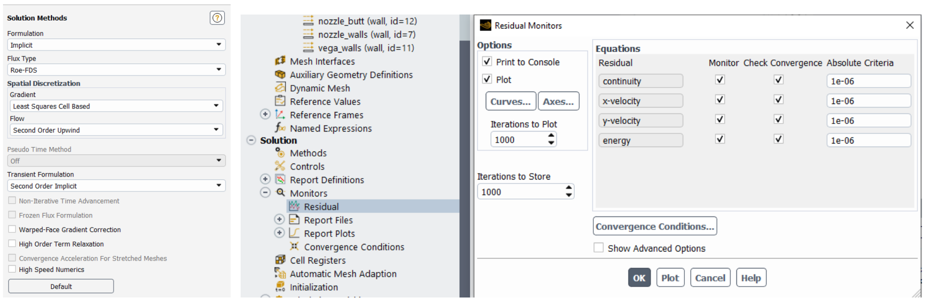
Field Variable: Pressure... / Static Pressure

Surfaces:
- Axis: axis
- Fluid: fluid-surface_body
- Inlet: far_field, inlet
- Internal: interior-surface_body, surface_body
- Point-surface: sonda1, sonda2, sonda3, sonda4

A.4.3 Numerical Methods and Controls

Double-click on **Methods** to choose the desired spatial and temporal discretization schemes. Then open the **Controls** menu to adjust the CFL number as needed.

Next, under the **Monitors** dropdown, open **Residuals** and set the convergence criteria. A value of `1e-6` is recommended to ensure solution accuracy.


Solution Methods

Formulation: Implicit

Flux Type: Roe-FDS

Spatial Discretization: Gradient: Least Squares Cell Based; Flow: Second Order Upwind

Pseudo Time Method: Off

Transient Formulation: Second Order Implicit

□ Non-Iterative Time Advancement
□ Frozen Flux Formulation
□ Warped-Face Gradient Correction
□ High Order Term Relaxation
□ Convergence Acceleration For Stretched Meshes
□ High Speed Numerics

Residual Monitors

Options: ☑ Print to Console; ☑ Plot; Iterations to Plot: 1000

Iterations to Store: 1000

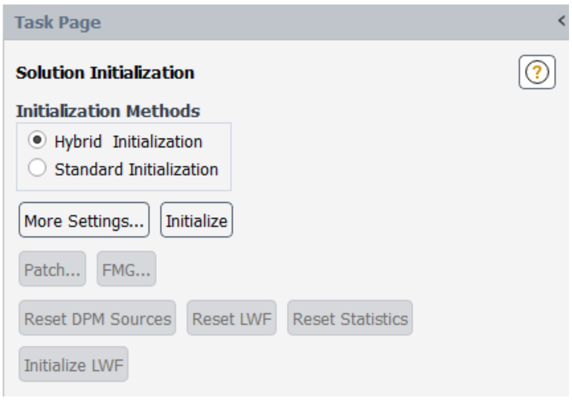
Equations

Residual	Monitor	Check Convergence	Absolute Criteria
continuity	☑	☑	1e-06
x-velocity	☑	☑	1e-06
y-velocity	☑	☑	1e-06
energy	☑	☑	1e-06

□ Show Advanced Options

A.4.4 Initialization and Simulation Run

To initialize the simulation, double-click on **Initialization**, select **Hybrid Initialization**, and click **Initialize**. This method initializes the flow field based on all boundary conditions rather than just the inlet, improving numerical stability.



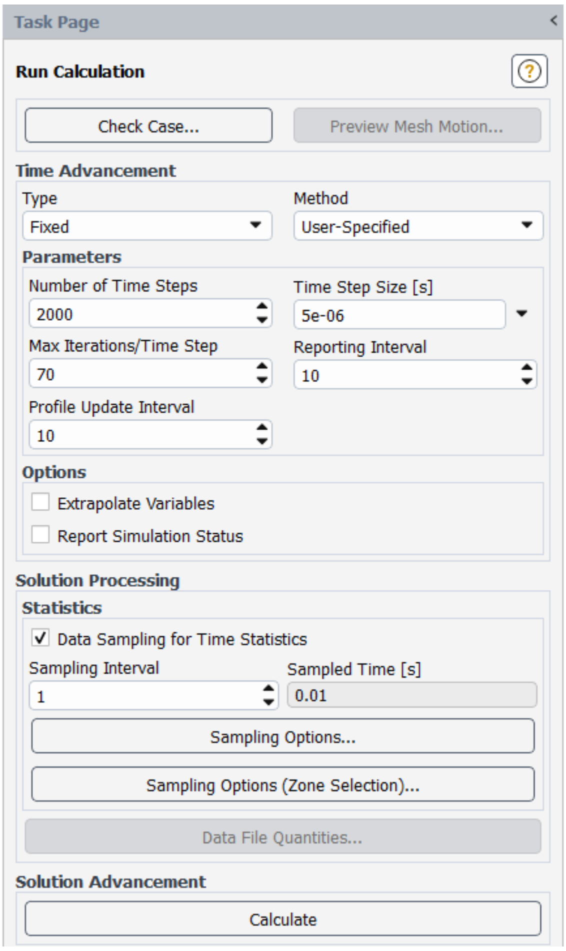
Task Page

Solution Initialization

Initialization Methods

● Hybrid Initialization
○ Standard Initialization

Then, double-click on **Run Calculation**. Here you can specify the number of **Time Steps**, their **size**, and the **maximum number of iterations per time step**. It is recommended to enable **Data Sampling for Time Statistics** to allow Fluent to save time-averaged quantities.



Task Page

Run Calculation

Time Advancement

Type: Fixed
Method: User-Specified

Parameters

Number of Time Steps: 2000
Time Step Size [s]: 5e-06
Max Iterations/Time Step: 70
Reporting Interval: 10
Profile Update Interval: 10

Options

□ Extrapolate Variables
□ Report Simulation Status

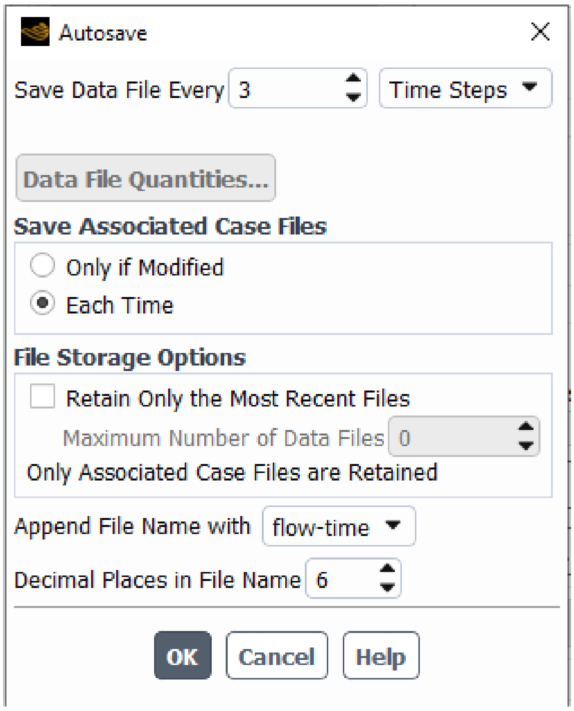
Solution Processing

Statistics

☑ Data Sampling for Time Statistics
Sampling Interval: 1
Sampled Time [s]: 0.01

Solution Advancement

If you wish to automatically save results at specific intervals, open **Autosave** under **Calculation Activities** and configure the save frequency. The output files will be stored in the path: Simulation_Name_files > dp0 > FFF > Fluent and can be reloaded after the simulation to extract results at specific times.



Autosave

Save Data File Every 3 Time Steps

Data File Quantities...

Save Associated Case Files

○ Only if Modified

● Each Time

File Storage Options

□ Retain Only the Most Recent Files

Maximum Number of Data Files 0

Only Associated Case Files are Retained

Append File Name with flow-time

Decimal Places in File Name 6

Once all parameters are set and verified, click **Calculate** to start the simulation.

List of Figures

Bibliography

[1] G. P. Sutton and O. Biblarz, *Rocket Propulsion Elements*, 9th ed. Wiley, 2016.

[2] E. S. Agency, “Vega launch vehicle overview,” 2025. [Online]. Available: https://www.esa.int/Enabling_Support/Space_Transportation/Launch_vehicles/Vega

[3] ——, “Vega-c: Europe’s next generation launcher,” 2025. [Online]. Available: https://www.esa.int/Enabling_Support/Space_Transportation/Launch_vehicles/Vega-C

[4] Arianespace, “Vega launch history and missions,” 2025. [Online]. Available: https://www.arianespace.com/vehicle/vega/

[5] P. J. Morris, *Broadband Shock-Associated Noise Prediction for Supersonic Jets.* Journal of Sound and Vibration, 2009.

[6] P. J. Morris and G. Miller, *Prediction of Broadband Shock-Associated Noise Using RANS CFD.* AIAA Journal, 2007.

[7] C. K. W. Tam and L. Auriault, *Jet Mixing Noise from Fine-Scale Turbulence.* AIAA Journal, 1996.

[8] B. André, T. Castelain, and C. Bailly, “Experimental study of mixing noise in subsonic and supersonic jets,” *Journal of Fluid Mechanics*, vol. 723, pp. 395–421, 2013. [Online]. Available: https://doi.org/10.1017/jfm.2013.110

[9] B. Henderson and T. Norum, “Impact of air injection on jet noise,” NASA Langley Research Center, Tech. Rep., 2007. [Online]. Available: https://ntrs.nasa.gov/citations/20070018374

[10] K. Fujii, T. Nonomura, and S. Tsutsumi, “Toward accurate simulation and analysis of strong acoustic wave phenomena - a review from the experience of our study on rocket problems,” *International Journal for Numerical Methods in Fluids*, 2010.

[11] A. S. Mouronval and A. Hadjadj, “Numerical study of the starting process in a supersonic nozzle,” *Journal of Propulsion and Power*, 2005.

[12] T. Nonomura and K. Fujii, *Aeroacoustic Waves from Supersonic Jet Impinging on an Inclined Plate.* Journal of Fluid Mechanics, 2011.

[13] C. B. et al., *Experimental and Numerical Investigation of Impinging Supersonic Jets.* AIAA Journal, 2016.

[14] C. Land, "What is computational fluid dynamics (cfd)," 2025. [Online]. Available: https://cfdland.com/what-is-computational-fluid-dynamics-cfd/

[15] J. H. Ferziger, M. Perić, and R. L. Street, *Computational Methods for Fluid Dynamics*, 4th ed. Springer, 2020. [Online]. Available: https://link.springer.com/book/10.1007/978-3-662-56336-4

[16] S. B. Pope, *Turbulent Flows.* Cambridge University Press, 2000. [Online]. Available: https://www.cambridge.org/core/books/turbulent-flows/91CB7DF4DFE2AE049BFD5E067161F1E9

[17] J. Smagorinsky, "General circulation experiments with the primitive equations: I. the basic experiment," *Monthly Weather Review*, vol. 91, no. 3, pp. 99–164, 1963. [Online]. Available: https://journals.ametsoc.org/view/journals/mwre/91/3/1520-0493_1963_091_0099_gcewtp_2_3_co_2.xml

[18] D. C. Wilcox, *Turbulence Modeling for CFD*, 3rd ed. DCW Industries, 2006.

[19] S. Mallat, *A Wavelet Tour of Signal Processing.* Academic Press, 1999. [Online]. Available: https://www.sciencedirect.com/book/9780123743701/a-wavelet-tour-of-signal-processing

[20] I. Daubechies, *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics (SIAM), 1992. [Online]. Available: https://epubs.siam.org/doi/book/10.1137/1.9781611970104

[21] M. Farge, "Wavelet transforms and their applications to turbulence," *Annual Review of Fluid Mechanics*, vol. 24, pp. 395–458, 1992. [Online]. Available: https://doi.org/10.1146/annurev.fl.24.010192.002143

[22] R. N. Bracewell, *The Fourier Transform and Its Applications*, 3rd ed. McGraw-Hill, 2000.

[23] L. N. G. Bachman and E. Beckenstein, *Fourier and Wavelet Analysis.* Springer, 2000. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4612-0505-0

[24] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[25] B. Wang and J. P. Ghil, "Wavelet analysis of enso variability in the last 100 years," *Climate Dynamics*, vol. 12, pp. 241–250, 1996. [Online]. Available: https://doi.org/10.1007/BF00231106

[26] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis," *Bulletin of the American Meteorological Society*, vol. 79, no. 1, pp. 61–78, 1998. [Online]. Available: https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Acknowledgements

I would like to express my sincere gratitude to Professor Emanuele Martelli for giving me the opportunity to complete this final chapter of my academic journey under his guidance. I am especially thankful for the chance to work on such an engaging project and to sharpen skills that will undoubtedly be valuable as I move forward in my career. I am also deeply appreciative of his patience and his total availability throughout these months, something I never took for granted.

I would also like to thank my family and all the people I hold dear for their unwavering support during these years of study, which were not without moments of difficulty. Without you by my side, this journey would have been, if not impossible, certainly much harder.