INTERNSHIP REPORT

Atmospheric reentry of a blunted cone at hypersonic speed:

From the characterization of the physical phenomena to the construction of an efficient solver on the COOLFluiD platform

Sana AMRI
Final year engineering student in
Applied Mathematics and Scientific Computing Methods
01-04-2018 to 30-09-2018

von Karman Institute for Fluid Dynamics
Aeronautics and Aerospace
Chaussée de Waterloo 72, 1640
Rhode-Saint-Genèse, BELGIUM

Supervisors
Thierry MAGIN[*]
Andrea LANI[†]
Fabio PINNA[‡]

Advisor
Vincent Fitzgerald GIANGASPERO

Sup Galilee, University Paris XIII
99 Avenue Jean Baptiste Clément,
93430 Villetaneuse, FRANCE

Academic supervisor
Emmanuel AUDUSSE [§]

---

[*] Associate Professor in the Aeronautics and Aerospace Department, VKI

[†] Research expert, Centre for mathematical Plasma Astrophysics, Mathematics Dept., KU Leuven
Visiting expert, Aeronautics & Aerospace Dept., VKI

[‡] Senior Research Engineer in the Aeronautics and Aerospace Department, VKI

[§] Associate professor in the Laboratoire d'Analyse, Géométrie et Applications (LAGA), University Paris XIII

Engineering: "The application of scientific principles to practical ends"

[The American Heritage Dictionary of the English Language, 1969]

Summary

This report summarizes the 6-month project made at the Institute von Karman to finalize the engineering degree in Applied Mathematics and Scientific Computing Methods from the Engineering School Sup Galilée (University, Paris XIII). The project consists of improving a solver (COOLFluiD) that simulate the atmospheric reentry of a blunt conical vehicle with local thermodynamic equilibrium assumption (LTE).

Contents

1 Introduction **6**
1.1 The von Karman Institute . . . 6
1.1.1 Structure of the organization . . . 6
1.1.2 Test Facilities and collaborative projects . . . 7
1.2 Vehicle's boundary layer during atmospheric reentry . . . 9
1.2.1 Notion of Hypersonic flow and earth's atmospheric reentry . . . 9
1.2.2 Case of an axisymmetric blunt cone . . . 11
1.3 Objectives . . . 12

2 Theoretical framework **13**
2.1 Conceptual model: Navier Stokes equations . . . 13
2.1.1 Three Laws of conservation for physical systems . . . 13
2.1.2 Gouverning PDE in cylindrical coordinates . . . 15
2.2 Numerical Methods used on the COOLFluiD platform . . . 15
2.2.1 Discretization in space and time . . . 15
2.2.2 Order of accuracy of the scheme . . . 17

3 Methodology **18**
3.1 State of the art . . . 18
3.1.1 Reentry problem . . . 18
3.1.2 Executive file in COOLFluiD . . . 18
3.2 Building an Executive file on COOLFluiD . . . 18
3.2.1 Restarting from an old executive file . . . 18
3.2.2 Mutation library version . . . 20
3.2.3 Interactive parameters in COOLFluiD . . . 22
3.3 Development of a lookup Table for thermodynamic properties . . . 24
3.3.1 Setting up the lookup Table for the thermodynamic variables . . . 25
3.3.2 Logarithmic scale for the pressure array . . . 27

4 Results and discussion **28**
4.1 Solution of the study case . . . 28
4.2 Residual temperature behavior with the lookup Table . . . 29
4.3 Discussion and suggestion of improvement . . . 31

5 APPENDIX **33**
5.1 Annex 1 . . . 33
5.2 Annex 2 . . . 34
5.3 Annex 3 . . . 35
5.4 Annex 4 . . . 36

Acknowledgment

I would like to express my grateful thanks to my supervisors. Special thanks should be given to Prof. Thierry Magin for his useful and constructive recommendations on this project.
My sincere gratitude to Andrea Lani for giving me the opportunity to implement tools in the COOLFluiD platform. You pushed me to behave less like a student and more as a “researcher”.
Thank you to Fabio Pinna for his patient guidance and useful critiques of this project.

My grateful thanks are also extended to Prof Emmanuel Audusse, my academic supervisor, for his assistance and quick responses by mail during this internship, but also for his listening skills and his ability to convey complex knowledge in a very simple way through all the engineering formation.

I would like to offer my special thanks to Vincent Giangaspero for his advice and assistance during this project. This internship has been a great experience mainly because of your daily support and availability.

I am particularly grateful to Firas Ben Ameur and Ray Vandenhoeck for their help with the COOLFluiD platform. Their willingness to give their time so generously has been very much appreciated.

My special thanks are also extended to the secretary staff members of the VKI, Simone Kratochvil and Dominique Landuyt, for their efficiency and their ability to make miracles!

Finally, I would like to thank my mother for her support and encouragement through this project and my studies.

1 Introduction

"Mathematics up to the present have been quite useless to use in regard to flying"
[14^{th} annual report of the aeronautical society of great britain, **1879**]

"Mathematical theories from the happy hunting grounds of pure mathematicians are found suitable to describe the airflow produced by aircraft with such excellent accuracy that they can be applied directly to airplane design"
[Theodore von Karman, **1954**]

As mentioned by Theodore von Karman in 1954 and way before by Galileo Galilee, Mathematics is the language of Nature. Amongst all the tools, nonlinear partial differential equations appear naturally to describe flows such as the earth atmosphere or the oceans. The absence of a general theory for the PDE, have led the researchers to have a new perspective on the equations. Instead of getting an analytical solution, the researchers are now focusing on the development of tools to resolve mathematical models. This scientific approach is known as applied mathematics.
Early, in the field of aerospace, it has already made its proof in 1962 at NASA. For the orbital mission of John Glenn[1], the specific position to return into the earth atmosphere and the right velocity were required, in order to make the astronaut come back safe. The mathematician Katherine Johnson used the Euler's method to get these parameters for the safe transition of the space capsule from an orbital trajectory, to a reentry path back to earth. The massive quantity of computation required, were made by the firsts IBM[2] computers programmed in the Fortran language.

Today, with the power of computers and the large panel of numerical methods, we are able to have good descriptions of a various physical phenomenon in the field of aerospace. These results are then validate by experiments through facilities and uncertainty studies. Even with all these new methods, describing the physical phenomenon of a vehicle during the atmospheric reentry is still a challenge. The pressure distribution, the shear stress, and the temperature on the surface of vehicles are the primary source to describe the airflow. But these parameters tend to reach critical values in hypersonic speed, which means that their behavior becomes unpredictable.
For this internship, we focused on the atmospheric reentry of an axisymmetric cone in a local thermodynamic equilibrium regime. The first chapter introduces the VKI, the study project and the objectives briefly. In section 2, the theoretical framework is recalled. In chapter 3, the different steps of the project are detailed, from getting used to COOLFluiD to the improvement of the solver by implementing new tools for the computation of the thermodynamic variables. In chapter 5, the results of our study case with and without the tools implemented are presented.

1.1 The von Karman Institute

1.1.1 Structure of the organization

VKI is a non-profit international educational and scientific organisation, hosting three departments (aeronautics and aerospace, environmental and applied fluid dynamics, and turbomachinery & propulsion). It provides post-graduate education in fluid dynamics (research master in fluid dynamics, former "VKI Diploma Course", doctoral program, short training program and lecture series) and encourages "training in research through research". The von Karman Institute undertakes and promotes research in the field of fluid dynamics. Extensive research on experimental, computational and theoretical aspects of gas and liquid flows is carried out at the VKI under the direction of the faculty and research engineers, sponsored mainly by governmental and international agencies as well as industries.

The von Karman Institute organizes each year 8 to 12 one-week Lecture Series on specialized topics in the field of aerodynamics, fluid mechanics and heat transfer with application to aeronautics, space, turbomachinery, the environment and industrial fluid dynamics. These courses have gained over the years world wide recognition for their high quality which is the result of a careful choice of subjects of current interest and lecturers known for their excellency in that field and willing to co-operate in building up well-structured courses.

VKI has a permanent staff of approximately 100 persons, among them 21 research engineers and 16 professors. Besides the permanent staff about 190 students and temporary researchers are involved in the different academic programmes. Students involved in these programmes have been recently awarded numerous prestigious grants such as those provided by the Belgian agencies FNRS, FRIA, FWO, IWT and the European Union ERC and Marie-Curie.

---

[1] the first American to orbit the Earth
[2] International Business Machines

The von Karman Institute hosts three departments:

Aeronautics and Aerospace The Aeronautics and Aerospace departement is a wide spectrum of facilities and computational tools covering the flow range from the low-speed regime of commercial aircraft to the supersonic and hypersonic regime of atmospheric space entry. The department focuses in particular on the modelling, simulation and experimental validation of atmospheric entry flows and thermal protection systems, including transition to turbulence and stability. The experimental studies are carried out in its world-class Mach 14, Mach 6 and Inductively Coupled Plasma windtunnels, for which dedicated measurement techniques have been developed, e.g. involving spectroscopic laser techniques.

Turbomachinery and Propulsion The Turbomachinery and Propulsion department specializes in the aero-thermal aspects of turbomachinery components for aero-engines and industrial gas turbines, space propulsion units, steam turbines and process industry compressors and pumps. It has accumulated wide skills in wind tunnel testing over a wide range of Mach and Reynolds numbers and related measurement techniques development and application.

For several years, the Turbomachinery and Propulsion Department has broadly specialized in activities related to aero-propulsion and energy conversion by means of rotating machinery. At the present time, the department teams up with the major European engine/energy manufacturers, either within European Commission co-funded projects or through bi-lateral collaborations.

Environmental and applied Fluid Dynamics The Environmental and Applied Fluid Dynamics department studies and teaches fluid dynamic aspects of environmental and industrial processes. Research themes cover a wide range of domains in response to the demands of the industry. Multiphase flows, aeroacoustics, wind engineering and cryogenic flows are some of the disciplines covered by this multidisciplinary department.

The department has 40 years of expertise in experimental fluid dynamics at full-scale and model-scale, including the design, construction and testing of dedicated experimental facilities, and the development of traditional and advanced, laser-based and acoustic measurement techniques.

1.1.2 Test Facilities and collaborative projects

World unique test facilities

The von Karman Institute for Fluid Dynamics has some of the most important and largest wind tunnel testing facilties in the world. Those facilities include low speed and high speed wind tunnels, plasma facilties as the well-know "plasmatron", water tunnel, turbomachinery facilities such as the biggest compression tube in the world "CT-3", ground vehicle facility, wind gallery, water spray, aeroacoustics and solid propulsion facilities, industrial test rigs.

For our study case, it is interesting to mention the VKI hypersonic tunnel H-3:

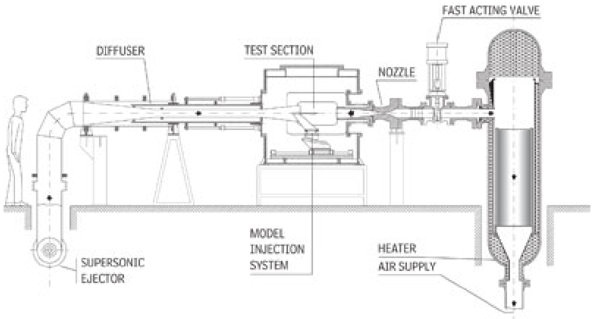


Figure 1: H3 facility

General Facts:

- The H3 tunnel is a blow-down facility with an axisymmetric nozzle giving a uniform Mach 6 free jet 12 cm in diameter.
- Air is supplied from a pebble-bed heater at stagnation pressures from 7 to 35 bar
- Maximum stagnation temperature of 550 K.
- Reynolds number may be varied from $3 \times 10^6/m$ to $30 \times 10^6/m$

- The test section contains a three-degree-of freedom traversing system for model and/or probe support that also allows the angle of incidence to vary between -5 to +5 degrees.

It is also interesting to mention the German High Enthalpy Shock Tunnel of Göttingen. To validate our project, data from this tunnel have been used **[13]** for this specific case.

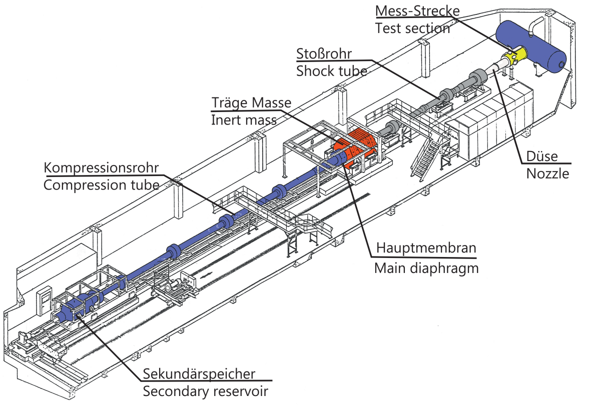


Figure 2: HEG facility

General Facts:

- Total length of 62m and total mass of 260t Four hypersonic nozzles (**Ma 6** to Ma 10)
- Two test sections and high speed H2 supply for wind tunnel models
- Typical test gases : **Air**, Nitrogen and CO2, basically no restriction for the type of gas
- Optical measurement systems for flow visualization and optical high speed tracking Fields of Application
- **Investigation of high temperature effects in entry and re-entry flows**
- Study of hypersonic flow at Ma 8 und Ma 10 in 20km up to 40 km altitude

Numerical solvers

On the computational side the Aeronautic and Aerospace department has developed an extendable software platform, **COOLFluiD** for high-performance computational flow, and the **library Mutation**++ for the computation of different chemical properties.

COOLFluiD (Computational Object-Oriented Libraries for Fluid Dynamics) is a component-based framework for scientific high-performance computing, CFD and multi-physics applications, developed initially at the von Karman Institute for Fluid Dynamics. This platform is an open-end collaborative project providing a robust set of tools for:

- solving complex applications with existing numerical solvers;
- building entirely new or customized models/solvers with arbitrary data-structures;
- easily defining reusable components (e.g., algorithms, models, BCs, wrappers);
- building virtual prototypes and quickly test new algorithms or models;
- interfacing or coupling other libraries/solvers to tackle more complex problems;
- large-scale simulations.

With more than one hundred contributors since 2002 from various institutions and more than one million lines of code, the COOLFluiD platform is an open source downloadable from Github[3].

Mutation++ is a c++ library designed to provide efficient algorithms in the fields of hypersonic and combustion Computational Fluid Dynamics for the computation of various properties including:

- Thermodynamic properties
- Multicomponent transport properties
- Finite rate chemistry in thermal nonequilibrium
- A highly robust multiphase equilibrium solver

To access directly, to Mutation++ from the platform COOLFluiD, a plugin has been enclosed in COOLFluiD itself. Mutation++ is treated as a black box from COOLFluiD, and the data exchange is controlled by the Mutation-COOLFluiD interface. During my project, I worked on this interface structure in order to implement new tools.

1.2 Vehicle's boundary layer during atmospheric reentry

1.2.1 Notion of Hypersonic flow and earth's atmospheric reentry

What's a hypersonic flow? The most common definition is to say that flow in a specific medium reach a hypersonic speed if the Mach number in the corresponding medium is greater than 5. In his book **[4]**, John D. Anderson Jr explains that this definition is not accurate.

> “ *When a flow is accelerated from* $M = 4.99$ *to* $M = 5.01$*, there is no 'clash of thunder', and the flow does not 'instantly turn from green to red'.* ”

For the definition of a hypersonic flow, we will refer to **[4]**. A hypersonic flow is defined as a regime where all or some of the following phenomena become important proportionally with the increase of the Mach number.

- **Thin shock layer**: When a hypersonic flow invests a body, the distance between the generated shock wave and the body is smaller compared to supersonic conditions.
- **Entropy layer**: Across a shock wave, the entropy tends to increase. In the case of a blunt cone (our study case), the shock wave is strong and highly curved in the nose region. This leads to strong entropy gradients generated in the nose region. This phenomenon is a source of analytical problems when we wish to perform a standard boundary layer calculation (What should be the proper condition at the outer edge of the boundary condition?).
- **Viscous interaction**: The considered vehicle reach a high velocity which implies a large amount of kinetic energy. But when the flow is slowed by viscous effect within the boundary layer, then the loss of kinetic energy is transformed, in part, into the internal energy of the gas. That's the viscous dissipation. The viscosity coefficient increases with temperature which leads to thicker boundary layer (because of the equation of state $p = \rho RT$).
- **High-temperature**: The severe post-shock conditions can create a very high heat that can cause dissociation and even ionization within the gas. The nose region of a blunt body is a high-temperature flow behind the strong shock wave.
- **Low-density flow**: At high altitude (above 92km), the assumption of a continuous medium to apply the Navier-Stokes equations becomes tenuous. At a hypersonic speed when a vehicle enters the atmosphere, it will pass through different regime: Above 150km high, the Navier-Stokes equations are no longer valid. Between about 92 and 150 km, there is a tenuous phase where the medium can be considered continuous. But, because of the low density, the conventional viscous flow no-slip boundary conditions at the surface of the vehicle doesn't hold.

---

[3] https://github.com/andrealani/COOLFluiD/

The following scheme, from **[4]**, summarizes the essential physical phenomena associated with hypersonic flight.

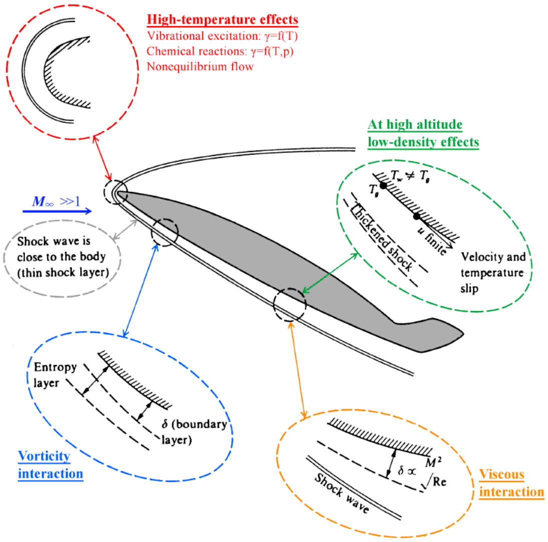


Figure 3: Physical effects characteristic of hypersonic flow (from **[4]**)

The reasons to distinguish hypersonic flow from supersonic flows ($Mach > 1$) is, indeed, the apparition of these critical phenomena that can have a severe impact on the vehicle itself. As mentioned by Guillaume Grossir, this differentiation has been pointed out by experiments. To illustrate the effect of the high temperatures characteristic of hypersonic flows is interesting to report the following anecdote [7]:

"*In 1949, a V-2 rocket brought from Germany was equipped with a second stage rocket (WAC Corporal) and fired from New Mexico desert. After reaching an altitude of 452km, it reentered the atmosphere at a velocity on the order of 2 200m/s. No remains of the slender rocket could be found except from few charred elements. This underlines the severe thermal environment associated with hypersonic flows, often described as a "heat barrier" [...]. All hypersonic vehicles therefore require some kind of thermal protection system. Allen and Eggers (1953) demonstrated few years later that blunt geometries should be preferred*[4] *in order to survive the atmosphere entry. This contrasts with the slender ones commonly used in the supersonic regime where the minimization of the drag coefficient prevails over the heat transfer.*"

A vehicle at hypersonic speed encounter several critical phenomena, and the study of these phenomena is mandatory for the design of re-entry capsules and hypersonic vehicles.

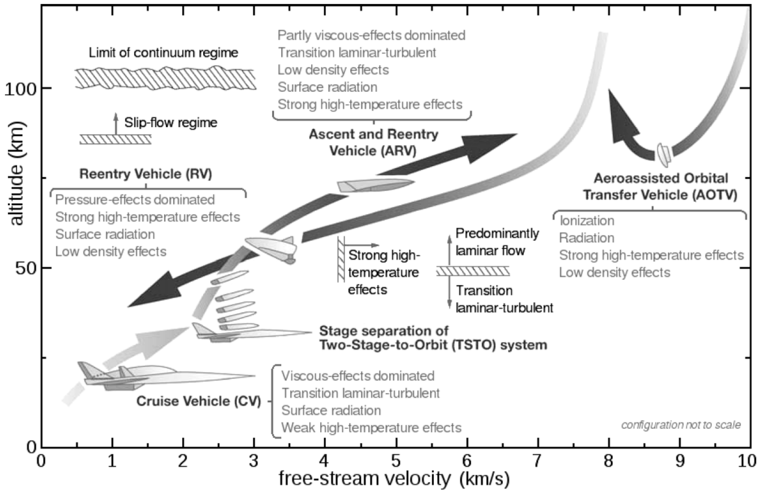


Figure 4: Physical effects characteristic of hypersonic flow **[4]**

---

[4] That's also one of the reason why the following considered study will be on a blunted cone.

Also having a better knowledge of the medium itself can be a real asset for the design of a vehicle. For instance, space vehicles encounter the earth atmosphere during their blastoff from the earth surface and again during their reentries. Therefore, the properties of the atmosphere must be taken into account. As shown in the map above, the earth's atmosphere is a dynamically changing system.

The pressure and the temperature of the atmosphere depend on many variables like the location on the globe or the time of the day. It also depends on the altitude of the vehicle and its velocity.

For example, at high altitude and velocity, there is ionization phenomena, strong high-temperature effect but also low-density effect. At low altitude and velocity, the leading critical phenomena will be the pressure and the viscous effect.

1.2.2 Case of an axisymmetric blunt cone

For the project, we consider the study of an axisymmetric blunt cone during the earth's atmospheric[5] reentry. The physical model considered is described in this section.

The 2D physical model is presented in the following scheme. The shape in blue corresponds to the area of interest around the cone. For the mesh, we will consider only the half of it because of the axisymmetric property of the cone. More information about the experimental setup are reported in **[13]**.

We consider a standard atmosphere with 5 chemical elements: O, N, NO, N_2, O_2 (AIR5)

The properties of the flow field are: $\overrightarrow{v} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 4224 \\ 0 \end{pmatrix} \; [m/s], \qquad T = 1192 \; [K] \qquad p = 6880 \; [Pa]$

On the surface of the cone (corresponding to the wall in the scheme), we fixed a temperature of 293K, and we make the no-slip wall assumption which means that the flow has no tangential velocity along the boundary of the cone.

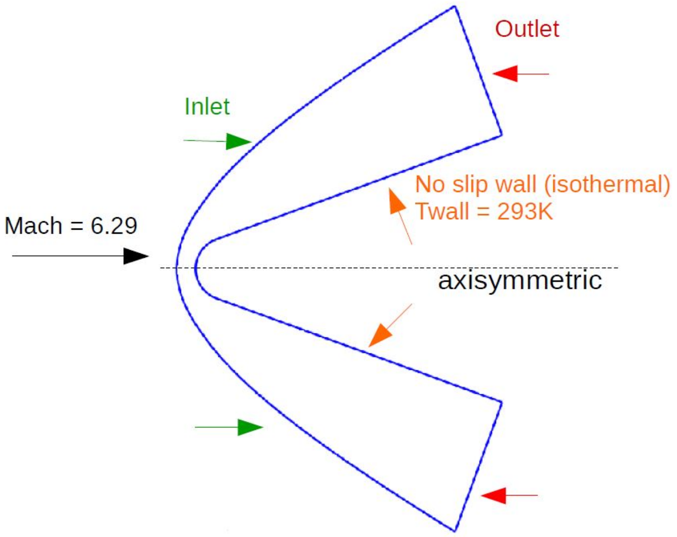


Figure 5: Physical model scheme

The scheme describes the main macroscopic properties of the system.

The thermodynamic properties (chemical properties such as the internal energy and the enthalpy) also have to be taken into account. Is there a link between these latter and the Mach number? We saw in **figure 4** that high velocity leads to a high temperature which causes ionization. The link between the Mach number and the regime of the flow is quite apparent. But, how can we quantify the regime of the flow from a chemical point of view??

---

[5] Standard atmosphere: AIR 5

To give an answer, we will first introduce an adimensional parameter, known as the Damköhler number and defined by:

$$Da := \frac{\text{characteristic time of the flow (macroscopic scale)}}{\text{characteristic time of the chemical reactions occuring in the flow}}$$

As described in **[1],** the Damköhler number is a dimensionless number used in chemical engineering to relate the chemical reaction timescale (reaction rate) to the transport phenomena rate occurring in a system. Indeed, at relatively high Mach numbers, the gas temperature may raise enough to cause dissociation (Mach≥ 7) and ionization (Mach≥ 12). Thus the gas becomes chemically active and electrically conductive. Three cases can be identified:

1. $Da \to 0$: the flow is frozen,meaning that no chemical reactions occur
2. $Da \to \infty$: the chemical reactions are fast enough to reach equilibrium conditions
3. $Da \simeq 1$: the flow is in chemical non equilibrium

In our study case, we presume that we are in the condition **2.** In other words, we suppose that in a local domain, the medium doesn't undergo spontaneous change. The medium is in chemical equilibrium.

Hypothesis:

- The flow field is a continuous medium (knusden number ≤ 0.2)
- The fluid is multispecies (air5) flow in LTE conditions.

1.3 Objectives

The general goal of the project is to develop and use numerical tools for simulating the flow over a body in a hypersonic facility in order to provide a better understanding of the physical phenomena inside the facility. But also, to be a key component in fully specifying the facility's test flow conditions.

The following scheme established by Ernst Heinrich Hirschel, show the subtle link between a considered physical phenomena, its conceptual vision (i.e., modelization) and the scientific computing method used to solve, with a specific error, the conceptual model.

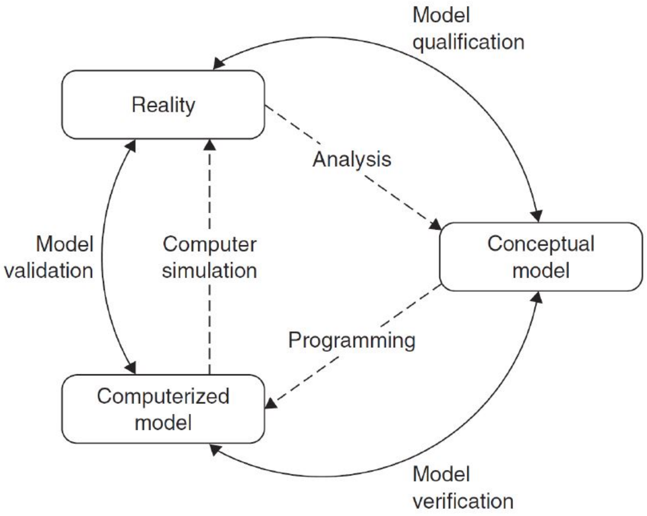


Figure 6: Relational scheme **[Ernst Heinrich Hirschel, 2005]**

The main objectives of this project are:

- Discover the world of aerospace by studying the main phenomena occurring during the atmospheric reentry of a vehicle
- Get familiar with the COOLFluiD platform to be able to carry on numerical investigations of the specific test-case
- Optimize and adapt the choice of the different numerical methods and tools used in order to achieve an optimal set-up for the simulations
- Development and implementation of a Lookup table to enhance the performance of the solver in terms of memory usage and computational time

2 Theoretical framework

2.1 Conceptual model: Navier Stokes equations

The Navier-Stokes equations are over one hundred seventy years old. The name Navier-Stokes initially referred to the conservation equation of linear momentum. Today it denotes collectively the conservation equations of mass, momentum, and energy. As written in **[14]**, this model can be used in a wide range of fluid flow configurations, *"whether it is the flow in a hurricane or in a turbomachine, around an airplane or a submarine, in arteries or in lungs, in pumps or in compressors, the Navier-Stokes equations can describe all these phenomena."*

2.1.1 Three Laws of conservation for physical systems

The **three laws of conservation [6]**, are listed below:

1. **Conservation of mass** (equation of continuity)

   The equation of continuity, in Lagrangian term is : $m = \rho \upsilon = cte$
   Where υ is the volume of a single particle and ρ is the density and cte is a constant.

   Recall, that the medium that we consider is a flow field with millions of separate particles where each follows its own path! It is hence necessary to consider a new point of view of the medium. Instead of following the path of each particle, we will consider the entire flow at every fixed point as a function of time. That is the Eulerian formulation of motion.
   Therefore in the Eulerian term, we will use the particle derivative **[Prof. Mekkas's course]** : $\frac{DX}{Dt} = \frac{\partial X}{\partial t} + (V \cdot \nabla X)$

   We have $\frac{Dm}{Dt} = \frac{D}{Dt}(\rho \upsilon) = 0 = \rho \frac{D\upsilon}{Dt} + \upsilon \frac{D\rho}{Dt}$

   Moreover, knowing that: $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \frac{1}{\upsilon}\frac{D\upsilon}{Dt} = div(\mathbf{V})$ and $div(\rho \boldsymbol{I}) = 0$

   We easily deduce that: $\frac{D\rho}{Dt} + \rho div(\boldsymbol{V}) = 0 \Longleftrightarrow \frac{\partial \rho}{\partial t} + div(\rho \boldsymbol{V}) = 0$

2. **Conservation of momentum** (Newton's second law)
   Recall the Newton's second law: $\boldsymbol{F} = m\boldsymbol{a}$

   Because the medium considered is a fluid particule, it is convienent to divide the previous relation by the volume of the particule υ, which leads to:

   $\boldsymbol{F} = m\boldsymbol{a} \Longleftrightarrow \frac{m}{\upsilon}\boldsymbol{a} = \frac{\boldsymbol{F}}{\upsilon} \Longleftrightarrow \rho \frac{d\boldsymbol{v}}{dt} = \boldsymbol{f}_{body} + \boldsymbol{f}_{surface}$

   where $\boldsymbol{f}_{body}$ and $\boldsymbol{f}_{surface}$ are forces ber unit volume.

   Most of the time the body force are: $\boldsymbol{f}_{body} = \rho \boldsymbol{g}$, with $\boldsymbol{g}$: the vector acceleration of gravity. But, for our study case, the body force is neglected.

   The surface forces are those applied to the vehicle itself by external stresses $\boldsymbol{\tau}$ (tensor).

3. **Conservation of energy** (first law of thermodynamics)

The first law of thermodynamics told us that the sum of the work done dW and heat added dQ to the system, will increase the total energy (internal, kinetic and potential) of the system dE_t.
For fluid particle, the energy per unit volume is: $E_t = \rho(e + \frac{1}{2}V^2 - \boldsymbol{g} \cdot \boldsymbol{r})$ where e is the internal energy and $\boldsymbol{r}$ describes the displacement of a particle.
By using the particle derivative on the first law of thermodynamics, we obtain:

$$\frac{DE_t}{Dt} = \frac{DQ}{Dt} + \frac{DW}{Dt}$$

To express the heat transfer Q, Frank M. White, uses the **Fourier's Law**:

$$\boldsymbol{q} = -k\nabla T$$

where $\boldsymbol{q}$ is the vector rate flow per unit area and k is the thermal conductivity

By writing the heat flow at each face of the considered body **[6]**, Franck M. White obtains the desired expression for the heat transfer term:

$$\frac{DQ}{Dt} = -div(\boldsymbol{q})$$

The net rate of work is is obtained by also writing the rate of work per unit area on each face of the body (using the the tensor τ):

$$\frac{DW}{Dt} = \nabla \cdot (\boldsymbol{U} \cdot \boldsymbol{\tau}_{ij}) \quad \text{(indicial notation)}$$

The 3 unknown parameters which **must be obtained simultaneously** from these 3 equalities are:

- the velocity $\overrightarrow{\boldsymbol{v}}$
- the thermodynamic pressure p
- the absolute temperature T

However, these equations contain also 4 thermodynamic variables that need to be computed. We recall these latter below:

- the density ρ
- the enthalpy h
- the viscosity μ
- the conductivity k

Because the mixture is at a local thermodynamic equilibrium state, each of these thermodynamic variables **depends only** on the temperature T and the pressure p obtained. Otherwise,

$\rho := \rho(T,p)$, $h := h(T,p)$, $\mu := \mu(T,p)$ and $k := k(T,p)$

The course of Bernard Grossman **[8]**, contains the detail formulation of each of these parameters. We learn through this course that even if each component i of the mixture is thermally perfect (equation of state: $p = \rho_i RT$). The mixture itself is not. That is why the enthalpy and the internal energy don't only depend on the temperature T.

These parameters are computed in Mutation++. During the project, we used this library from an interface built in COOLFluiD, without having access to the Mutation++'s code itself.

Before enoncing the sets of gouverning PDE's for our case, we need to precise that these relations are fairly general and involve only a few restrictive assumptions **[6]**.

Hypothesis:

- The flow forms a (mathematical) continuum
- The particles are essentially in thermodynamic equilibrium
- The heat conduction follows Fourier's law
- the body forces are neglected

2.1.2 Gouverning PDE in cylindrical coordinates

For our study case, the gouverning PDE's that will be considered, are expressed from these previous 3 conservative laws.

Recall that our steady case is an axisymmetric case in 2D. In cylindrical coordinates, the system that we will considered is[6]:

$$\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{P}} \times \frac{\partial r\boldsymbol{P}}{\partial t} + \frac{\partial r\boldsymbol{F}_x^c(\boldsymbol{U})}{\partial x} + \frac{\partial r\boldsymbol{F}_r^c(\boldsymbol{U})}{\partial r} = \frac{\partial r\boldsymbol{F}_x^d(\boldsymbol{U})}{\partial x} + \frac{\partial r\boldsymbol{F}_r^d(\boldsymbol{U})}{\partial r} + \boldsymbol{S} \tag{1}$$

where,

- x and r are the axial and radial directions
- $\boldsymbol{U} = \begin{pmatrix} \rho \\ \boldsymbol{u} \\ \rho E_t \end{pmatrix}$ are the conservative variables, $\boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix}$ contain the velocity component and $\rho E_t = \frac{p}{\gamma-1} + \rho\frac{|\boldsymbol{u}|^2}{2}$: total energy per unit volume (γ: indice adiabatique)
- $\boldsymbol{P} = \begin{pmatrix} p \\ u \\ v \\ T \end{pmatrix}$ are the *update* variable in COOLFluiD (Andrea Lani's notation)
- $\boldsymbol{F}_i^c$ *and* $\boldsymbol{F}_i^d$, $i \in \{x, r\}$ are respectively the convective and diffusive fluxes in the direction i.
- $\boldsymbol{S}$ is the source term

To avoid cluttering the report, you'll find the detail of the hypervectorial form in the **annex 1**.

2.2 Numerical Methods used on the COOLFluiD platform

The COOLFluiD platform contains a wide range of numerical methods that allows solving the hyperbolic system (1).

To solve numerically the system, it must be discretized in time and space.

This section gives a global vision of the numerical methods used without going into the details. For more information, see **[1]**.

2.2.1 Discretization in space and time

In Computational Fluid Dynamics (CFD), the Finite Volume Method (FVM) is one of the most used because of its high flexibility and robustness as a discretization method. As mentioned by Andrea Lani, "*the success of this method is mainly due to the capability to adapt to every kind of meshes and to the good shock capturing properties, which make it suitable for handling compressible flows exhibiting complex shock interactions and discontinuities in general.*"

This method is a technique that transforms a system of integral equations on different volumes of controls from a continuous space (commonly on a Hilbert space), into a system of linear equations on discrete volumes (cells of the meshes

---

[6] We use the Notation from Andrea Lani's thesis

considered). On each volume of control, we approximate the analytical solution by a certain constant in its center. This method is known as **cell-centered finite volume method**.

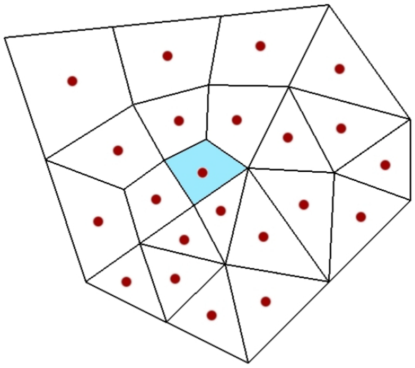


Figure 7: a cell centered Finite Volumes discretization [**chapitre 5.2 T1**]

The governing equations **(1)** are integrated over the finite volumes Ω_i into which the domain has been subdivided, then the Gauss theorem is applied to simplify the convective and diffusive term.

$$\boxed{\underbrace{\frac{d}{dt}\int_{\Omega_i}\boldsymbol{U}dw}_{\substack{transient\\ term}} + \underbrace{\int_{\Sigma_i}\boldsymbol{F}^c\boldsymbol{.}\boldsymbol{n}ds}_{\substack{convective\\ term}} = \underbrace{\int_{\Sigma_i}\boldsymbol{F}^d\boldsymbol{.}\boldsymbol{n}ds}_{\substack{diffusive\\ term}} + \underbrace{\int_{\Omega_i}\boldsymbol{S}dw}_{\substack{source\\ term}} \qquad (2)}$$

where Σ_i describes the area of the cell Ω_i.

Let's rewrite the gouverning equation in the form:

$$\tilde{\boldsymbol{R}}_s(U) = \frac{d}{dt}\int_{\Omega} U_s d\omega + \boldsymbol{R}_s(P) = 0 \text{ , s=1,...,}N_s$$

The system (2) is decomposed into N_s weakly coupled equation subsystems. For each subsystem, the convective, diffusive and source term (which are gathered into the term $R_s(P)$) are discretized **separately**. That's the *space discretization*.

Each pseudo-steady residual $\tilde{\boldsymbol{R}}_s(U)$ is then discretized in time by using an implicit time discretization (Backward Euler) that leads to a linear system solved by a Newton method for weakly coupled system detailed in **[1 - Chapter 5]**

Remarks: The linear system obtained **[1 - Chapter 5]** contains the flux Jacobian term in which will appear the CFL condition. The value of this parameter will have a not negligible impact on the numerical resolution.
The following functional scheme shows the main method used in our case.

The following functional scheme summarize briefly the method of discretization used.

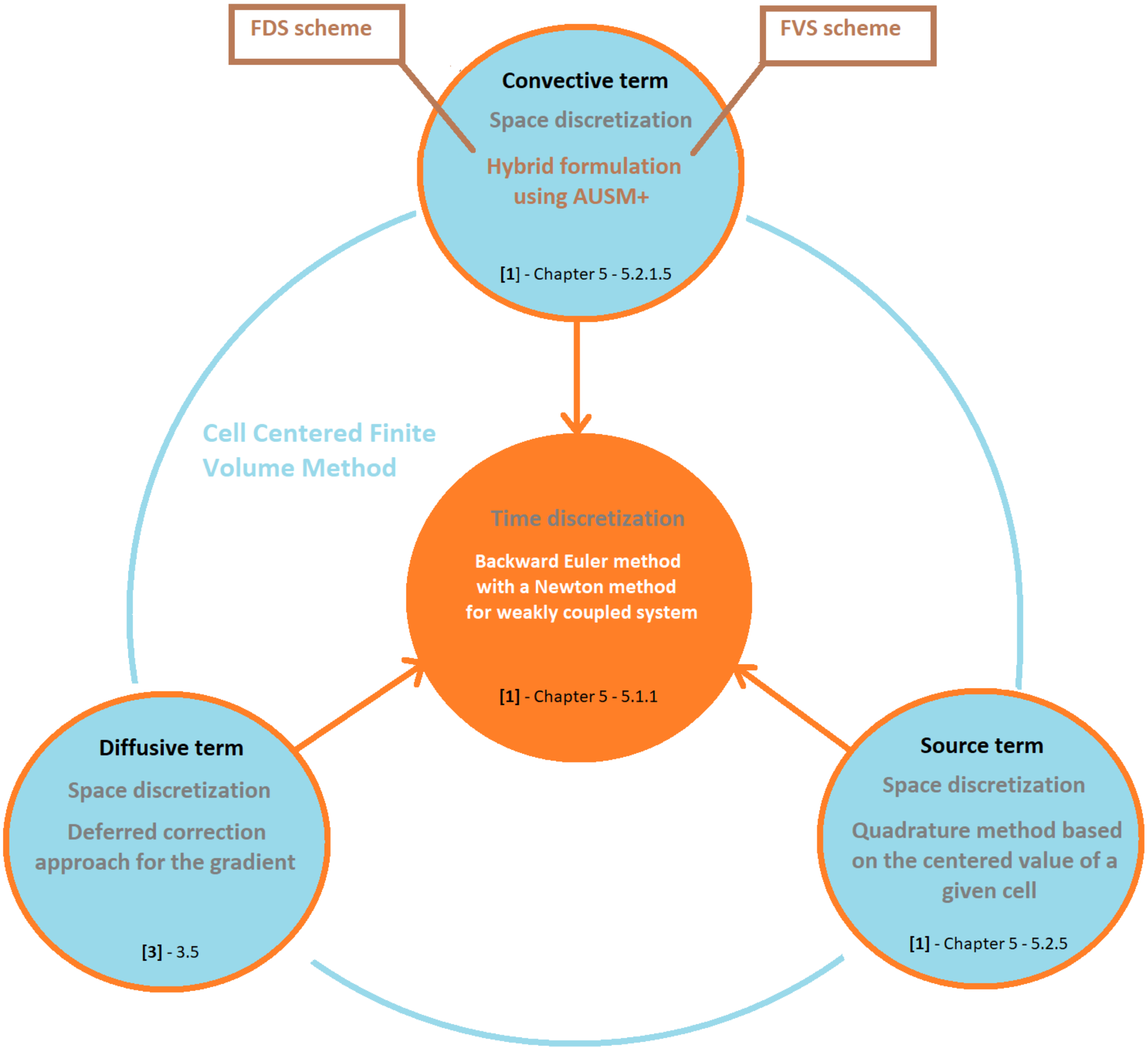

FDS scheme
FVS scheme
Convective term
Space discretization
Hybrid formulation using AUSM+
[1] - Chapter 5 - 5.2.1.5
Cell Centered Finite Volume Method
Time discretization
Backward Euler method with a Newton method for weakly coupled system
[1] - Chapter 5 - 5.1.1
Diffusive term
Space discretization
Deferred correction approach for the gradient
[3] - 3.5
Source term
Space discretization
Quadrature method based on the centered value of a given cell
[1] - Chapter 5 - 5.2.5

Figure 8: a cell centered Finite Volumes discretization [**chapitre 5.2 T1**]

2.2.2 Order of accuracy of the scheme

As written in **[13]**, the basic finite volume approximates each cell by a constant value which leads to a first-order accurate discretization in space. To get a higher order of accuracy we use a high order reconstruction method in which, each cell centered variable u_i is linearly extrapolated to the face quadrature points q. This method leads to a 2nd order accuracy. A MUSCL method is used in our case **[1 - Chapter 5]**.

$$\tilde{u}(x_q) = u_i + \nabla u_i \cdot (x_q - x_i)$$

where x_i denotes the centroid position of the control volume ω_i.

Also, when we deal with compressible flows, the linear reconstruction for a generic variable generates near shock waves the appearence of spurious oscillation. That's the Gibbs phenomenon. To prevent this situation, we use a Venkatakrishnan' Limiter that mainly consists of slightly modify the extrapolation (for the reconstruction) on each cell **[1 - Chapter 5]**.

$$\tilde{u}(x_q) = u_i + \phi_i \nabla u_i \cdot (x_q - x_i)$$

with $\phi \in [0, 1]$.

3 Methodology

3.1 State of the art

3.1.1 Reentry problem

Entering the VKI with mainly a mathematical background, I started to discover the field of aeronautics and aerospace. Used to the rigidity of mathematical books, I was pleasantly surprised to discover John Anderson's pen, the father of hypersonic.
I also discovered during this time that aerospace is a field that gathered the main field of mathematics: When the CFD solvers require plenty of numerical methods to solve the Navier Stokes equations in a specific domain (mesh topology), the experiment into the facilities requiere also the uncertainty studies (to take into account the error of measurement from the captors for example).

When I started the internship, the main goal was to learn the basic elements required to understand my study case. Then, be able to pose correctly the problem and get used to the COOLFluiD platform on which the case would be numerically solved.

3.1.2 Executive file in COOLFluiD

When I started to get used to the COOLFluiD platform, we built the executive file for our study case. Recall that COOLFluiD contains a very large panel of modules that cover everything needed to run a case. In this platform, we can activate options of different nature in the executive file.

Each test case is defined by a few input files:

- **CFcase file**: conventionally have the extension ".CFcase", provides all user-defined settings and parameters for configuring a simulation.
- **Interactive file**: conventionally with the extension ".inter", contains a few parameters (in the same format as in CFcase files) which can be modified on-the-fly by the end-users while the simulation is running. The interactive file is optional.
- **CFmesh file**: conventionally with the extension ".CFmesh", provide a mesh with or w/o solution in a native COOLFluiD format which supports parallel I/O. Those files can be used to start a simulation from scratch or to restart from a previous solution.

For more information, see **[1]**.

Some configurations like the CFL or the activation of the high reconstruction method in order to enhance the accuracy can be interactively modified during the computation.

3.2 Building an Executive file on COOLFluiD

3.2.1 Restarting from an old executive file

From the existing tools on COOLFluiD and Mutation++, we wanted to build an efficient executive file to run the study case. For that, we restarted a converged solution from an old executive file.

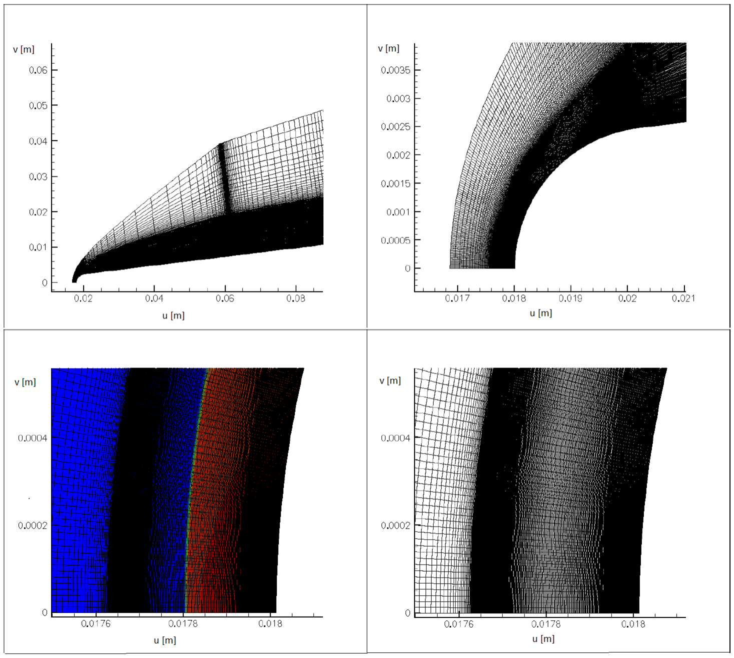


Figure 9: Restart Case

We realized that the mesh had an issue. As you can see in **figure 9**, it was not refined around the shock. Indeed, the shock refinement for this mesh was tailored for a perfect gas flow (we are here in LTE assumptions) where the nose shock was parabolic and where the downstream shock satisfies the Taylor Maccoll condition. That is why the mesh was not fitting the shock curve.

With the following unrefined mesh of 10K elements (for test purposes), we started the computation from scratch. The graph shows a zoom of the mesh in the nose region.

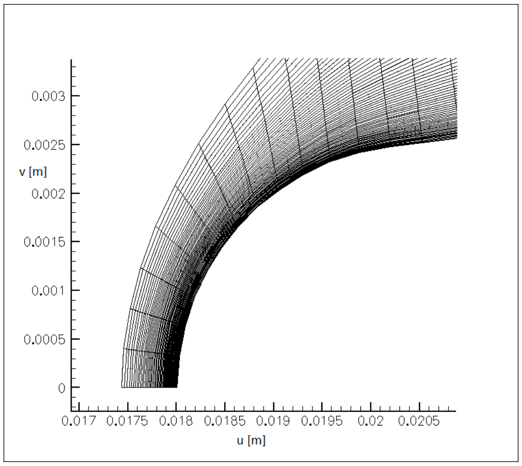


Figure 10: 10K elements Unrefined mesh

After converging from scratch in a first order of accuracy, I realized that the solution had an issue. As you can see on the left graph of the following figure, the shock didn't detach from the boundary layer.

Recall that we impose an isothermal wall. If we run the case only with this condition, the shock will not detache as shown in the **figure 11**.To detach the shock numerically from the boundary layer, the trick is to activate some adiabatic steps for a few iterations (in our case, we applied the adiabatic steps for the first 3000 iterations). The right graph, from **figure 11**, is the result after applying the adiabatic steps.

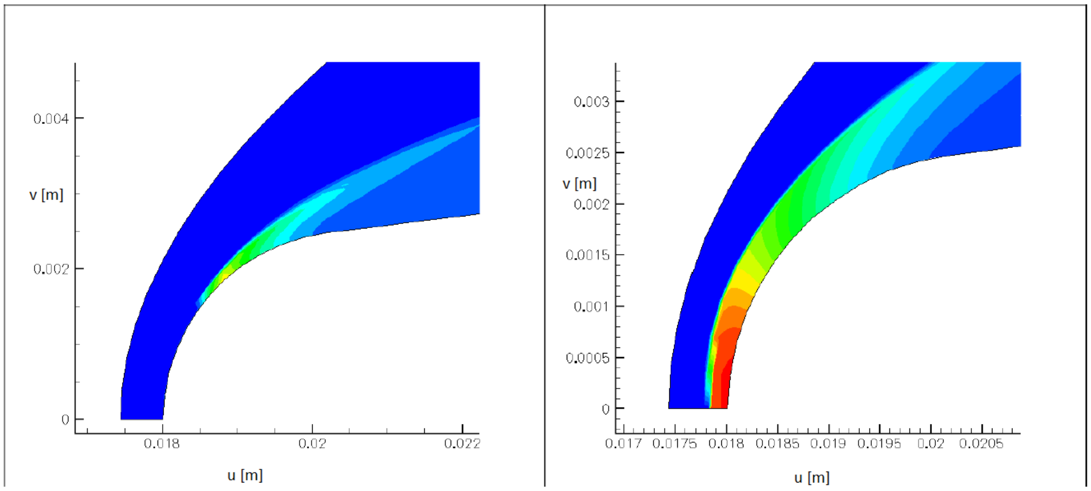


Figure 11: Impact of the adiabatic steps

3.2.2 Mutation library version

Recall that to access directly to Mutation++ from the platform COOLFluiD a plugin has been enclosed in COOLFluiD itself.

The COOLFluiD platform contains also plugins for the older versions of Mutation. The three main Mutation versions that we can access from COOLFluiD are summarized in the following table:

Mutation library version	**Interface name in COOLFluiD**	**Path in the COOLFluiD repository**
Mutation2 library	Mutation2OLD	plugins/Mutation2.0.0I/Mutation2OLD.*
Mutation library	Mutation	plugins/MutationI/MutationLibrary.*
Mutation++ library	Mutationpp	plugins/MutationppI/MutationLibrarypp.*

To compare the behavior of each Mutation version, we ran the case **without high order reconstruction and with a constant CFL value** through all the computation. We impose a precision of 10^{-4} for the residual temperature and an adiabatic wall for the first 3000 steps. The following figure shows the residual temperature obtained with each version of Mutation. We can see that for the first 8000 iterations the behavior is similar. Above these steps, the most stable residual temperature is the one computed with the Mutationpp version.

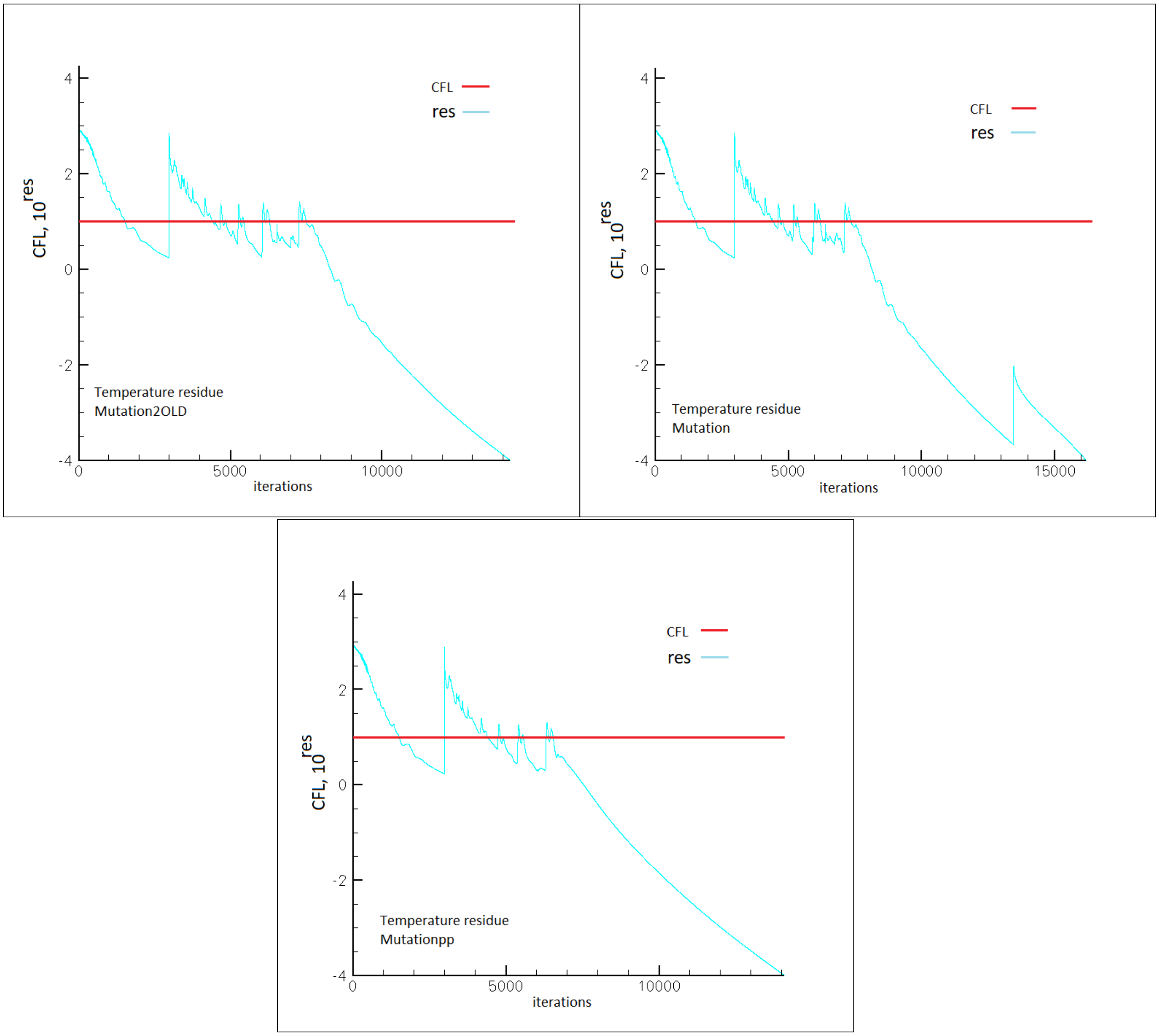


Figure 12: comparison Mutation version

The following table summarized the results obatined with the residual temperature computed with the different version of Mutation (without second order reconstruction).

Interface in COOLFluiD	**number of iterations for the residual temperature**	**CPU time**
Mutation2OLD	14 248	13 h 24 min 49.3914 sec
Mutation	16 198	14 h 8 min 2.48607 sec
Mutationpp	14 102	10 h 8 min 51.0333 sec

For the next simulations, we will only use the latest version of Mutation (Mutationpp module). Also, we will only discuss on the residual temperature because this parameter reaches the prescribed accuracy last. In particular, residual temperatures are more indicative of the solution accuracy because the temperature is the key quantity for hypersonic simulations and spacecraft design (amongst the pressure and the Mach number).

The following figure shows the residual temperature and solution obtained with the activation of the second order reconstruction and a constant CFL value. We recall that the solution is not accurate, because the mesh is very coarse. We use this mesh only for test purposes.

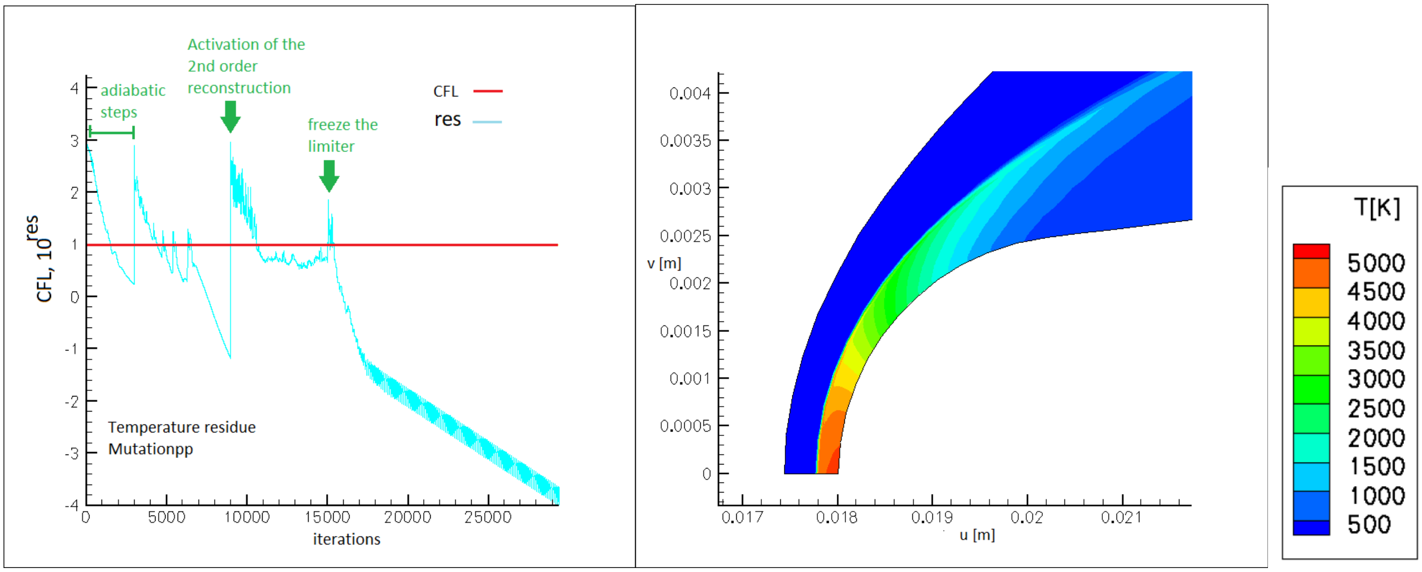


Figure 13: Mutationpp 2nd order - constant CFL

The convergence for the residual temperature, with second order reconstruction is reached after 29 456 iterations in 18 H 49 min 19.1234 sec.
Even for a coarse mesh, the simulation takes a lot of time. As shown after, it is possible to reduce considerably the CPU time by "playing" on the parameters listed below:

- The CFL parameter
- The activation of the second order reconstruction
- The limiter

For more information, see **[1 - Chapter 5]**.

3.2.3 Interactive parameters in COOLFluiD

To improve the speed of the convergence, I studied the impact on the residual temperature of the parameters mentioned in the last subsection. By using the interactive file, I modified each of these parameters and managed to enlighten specific patterns of the residual temperature. From that, after several tries, I built a function that changes the parameters automatically in order to reduce the CPU time as much as possible.

The following figure compares the residual temperature behavior for a simulation without second order reconstruction. The left graph represents the residual temperature with a fixed CFL value (that we saw previously in the section 3.2.2). The right graph shows the behavior of the residual temperature when the CFL value changes through the computation.

In this subsection, all the case are run in parallel (4 threads) using MPI.

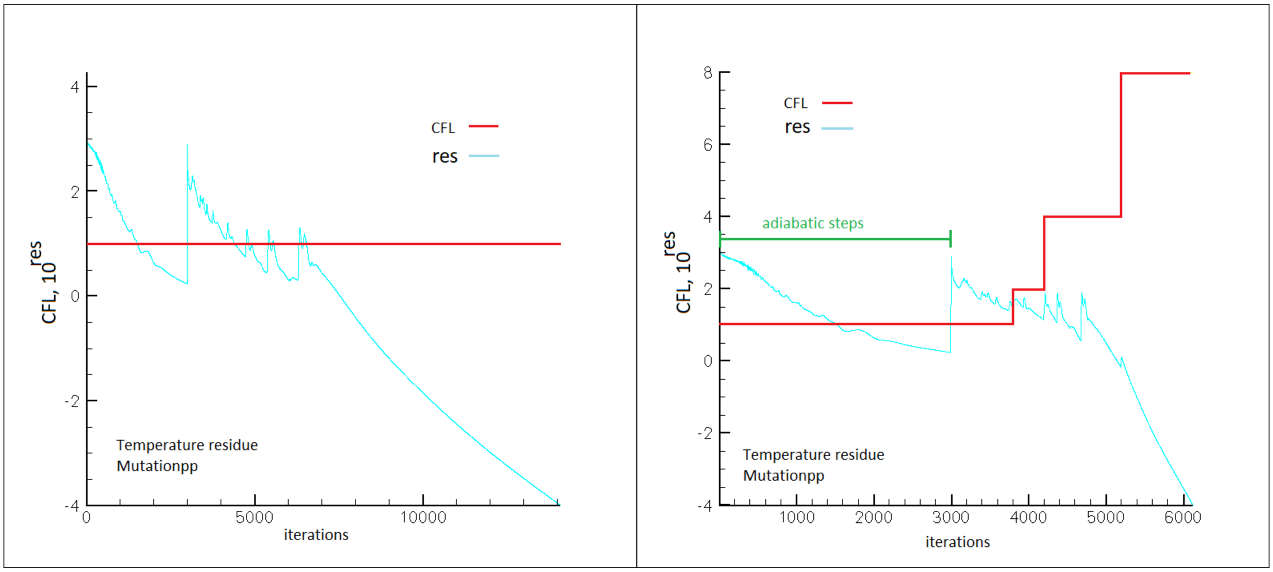


Figure 14: residual temperature

Without 2nd order reconstruction	FIXED PARAMETERS	VARIABLE PARAMETERS
Number of iterations	14 102	6130
CPU time	10 h 8 min 51.0333 sec	3 h 57 min 42. 8916 sec

We manage by increasing the CFL to reduce the oscillating part (after the adiabatic steps) and increase the slope of the straight line (after the oscillating part). Modifying the CFL value, during the computations, divided almost by 3 the CPU time.

As mentioned in the **subsection 2.2.1**,we can reach only a first order of accuracy with the cell-centered finite volume method. That is why we use a second order reconstruction method to enhance the accuracy. This method generates spurious oscillation near the shock wave (Gibbs phenomenon). That's why we use also a limiter to avoid this phenomenon.

When we run the simulation with the second order reconstruction we need to deal with three parameters. After several attempt, we manage to reduce considerably the CPU time. The following figure shows the residual temperature when we keep a constant CFL value and activate randomly the second order reconstruction (left graph). The right graph is the result obtained with the function built **(see annex 2)**

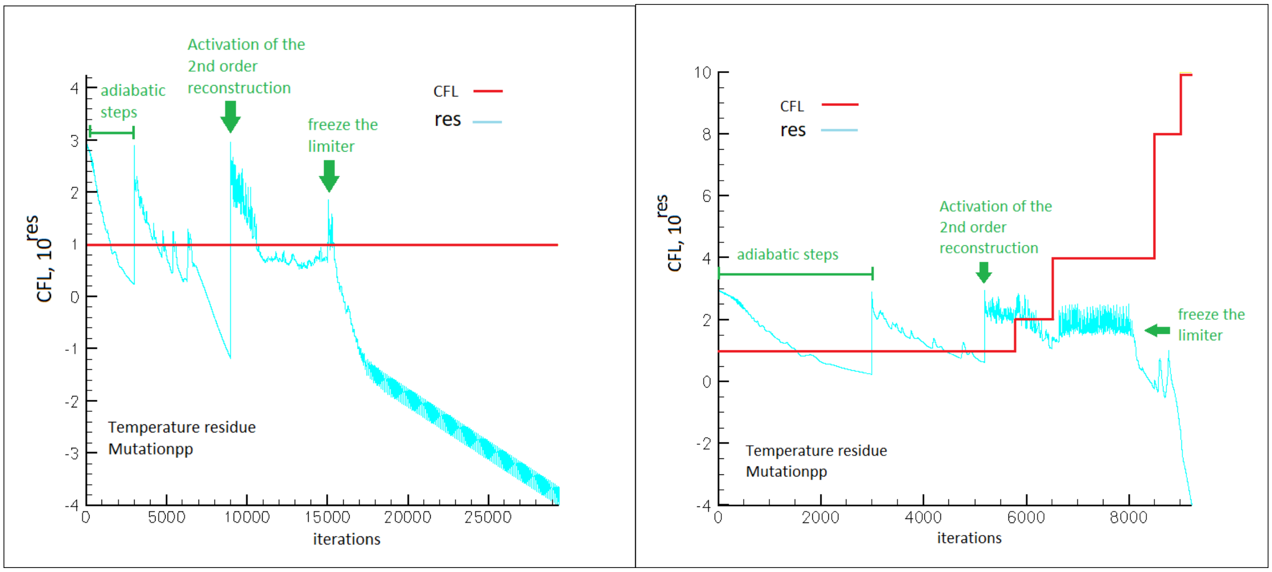


Figure 15: residual temperature (2nd order)

In the **Annex 2**, the algorithm used are detailed.

The following table summarize the results obtained:

With 2nd order reconstruction	FIXED PARAMETERS	VARIABLE PARAMETERS
Number of iterations	29 456	9232
CPU time	18 h 49 min 19.1234 sec	6 h 57 min 46.8631 sec

Again, by modifying the parameters, we divided by almost three the CPU time. Consequently, it is worth to spend time for a case on finding the right algorithm for the CFL, the activation of the 2nd order of accuracy and when to freeze the limiter.

Recall that we work with a coarse mesh (10K elements) that has been build only for test purposes. What if we adopt these configurations on a finer mesh? There is no mathematical assertion which assures that the residual temperature behavior will be similar with a finer mesh. The computation will require a lot of costs given that we will compute the case on a 500K elements mesh.

We first make the first try with fixed parameters. After 2 weeks of run (on the cluster), the precision of 10^{-4} was still not reached for the residual temperature. That is why we stopped the code and tried to apply what we learned from the 10K elements mesh to the 500K elements mesh. After less than 6 days (118h 41 min 46.7755 sec), the residual temperature reached a precision of 10^{-4}. In the following figure, we can see the behavior of the residual temperature with the 500K elements mesh (The complete solution will be shown in **section 4.1**).

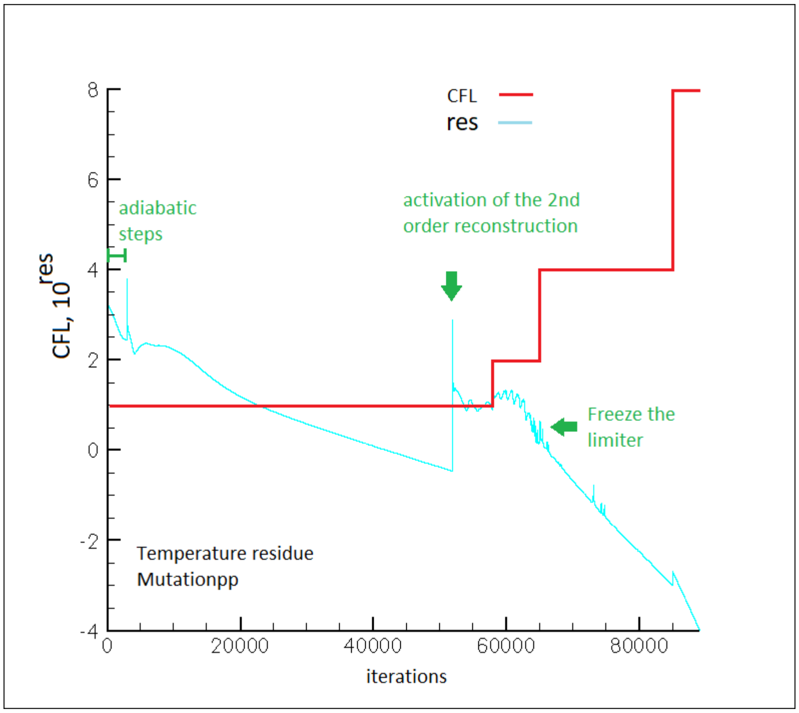


Figure 16: residual temperature with 500K mesh

3.3 Development of a lookup Table for thermodynamic properties

Even if “playing” with the interactive parameters reduce the CPU time of the simulation, the computation requires still a lot of time. This section shows how the time can be divided by two by using a lookup table to get the thermodynamic variables.

3.3.1 Setting up the lookup Table for the thermodynamic variables

As previously mentioned, the thermodynamic variables is highly nonlinear. Hence, at each iteration the computation of these latter requires a lot of CPU time. Recall that each of the thermodynamic variables depends only on the pressure and the temperature (LTE assumption). So, instead of computing the thermodynamic variable at each iteration, the idea is to create a lookup table in which we will get the corresponding values, at each step, by using an interpolation method. In this part, we will describe the method used to implement the lookup Table.
We previously mentioned that the platform COOLFluiD contains several modules to access different version of Mutation (see 3.2.2): Mutation, Mutation2OLD, and Mutationpp.

By using the **Mutation** modules (see the table in the subsection 3.2.2), I tried to activate the lookup table implemented by Andrea Lani (a few years ago), to my case. The code crashed after a few iterations. The reason was wether because of the lookup table itself or the fact that we were using an old version of Mutation.

From this observation, I decided to implement a lookup Table into the **Mutationpp** plugin by applying the same algorithm used by Andrea on the older version of Mutation++. That way, I had the main structure of the algorithm.

Also, if the "same" lookup table works into **Mutationpp**, it means that the older version of **Mutation** was not stable enough. If it doesn't work, it may be because of the interpolation method used (bilinear interpolation) to get the values from the table.

Implementation of the lookup table:

We want to interpolate the following thermodynamic variables:

- the sound speed: a
- the enthalpy: h
- the internal energy: e
- the density: ρ

The targeted files to be modified are: COOLFluiD/plugins/MutationppI/MutationLibrarypp.*

- For the Lookup Table, we use these main two modules:
  - **COOLFluiD/src/Common/LookupTable2D.***
    These files defines a class that represents a table where values can be looked up and if not found, are interpolated with the nearest ones.
  - **COOLFluiD/src/Common/CFMap.***
    These files allow to stock efficiently the table. Recall that Maps are associative containers that store elements formed by a combination of a key value and a mapped value, following a specific order. In our case, the key values will be the thermodynamic variables, and the mapped values will point to the corresponding function in Mutation++.

Method used to implement the Lookup Table:

**I - Steps BEFORE the computations of the Navier Stokes equations.
This step should only be done at once.**

1. We first generate two arrays containing all the keys for the temperature T and the pressure p. For that the user entered:
   - the boundary values for the temperature $[T_{min}, T_{max}]$ and the pressure $[p_{min}, p_{max}]$
   - the length of the subintervals (precision) of the temperature ΔT and the pressure Δp

   From these inputs, we generate two vectors defined as follow[7]:

---

[7] For now, we use a simple linear scale. We will see later the method to implement a logarithmic scale for the pressure.

Let $nbT = \frac{T_{max}-T_{min}}{\Delta T}$

If $T_{min} + nbT \times \Delta T = T_{max}$ then $T_{values} = [T_{min}, T_{min} + \Delta T, \dots, T_{min} + (nbT - 1) \times \Delta T, T_{max}]$

Else $T_{values} = [T_{min}, T_{min} + \Delta T, \dots, T_{min} + nbT \times \Delta T, T_{max}]$

We build, in a similar way, the vector p_{values}.

2. From these array we compute (with Mutationpp) the thermodynamic variables using all the combinaison possible (T_i, p_j). In other words:

   For all, $T_i \in T_{values}$ and $p_j \in p_{values}$, we compute: $a(T_i, p_j)$, $e(T_i, p_j)$, $h(T_i, p_j)$ and $\rho(T_i, p_j)$.

   Then, we stock the values in the lookup Table (using the CFMap module).

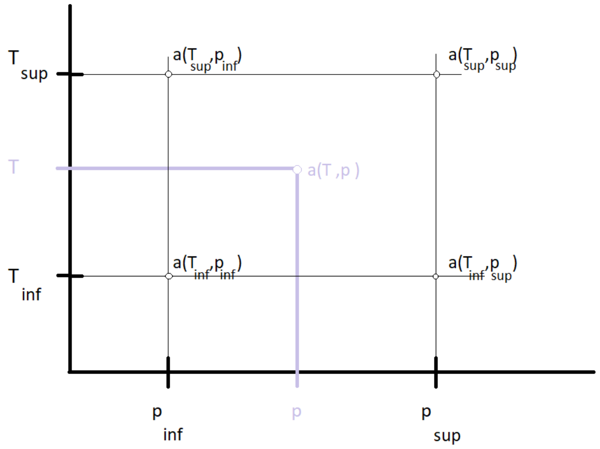
We now have generated a lookup table for each thermodynamic variables that we will interpolate at each step during the computation of the Navier Stokes equations.

II - Steps DURING the computations of the Navier Stokes equations.

1. After each numerical step of the linear system, we get a temperature T and a pressure p. For both of these values we will find the closest lower and upper boundary in the array $T_{values}$and p_{values} using the c++ function **lower_bound()**

   Let the respective boundaries for T and p, be $[T_{inf}, T_{sup}]$ and $[p_{inf}, p_{sup}]$

2. From these two set of boundaries, the thermodynamic variables are interpolated. In order to simplify the explanation, we will focus only on the interpolation of the sound speed $a(T, p)$. For the three other variables (e, h andρ) the method is exactly the same.



In the scheme the value of the sound speed are the ones known stocked in the lookup table. We want to interpolation the value $a(T, p)$ using the four values at each corner.

For that we will use 4-noded rectangular element using Lagrange element. During this step, the difficulty was to understand the method from the code itself. Here is the explanation.

Let $l_{T_{inf}}(T) = \frac{T-T_{sup}}{T_{inf}-T_{sup}}$ and $l_{p_{inf}}(p) = \frac{p-p_{sup}}{p_{inf}-p_{sup}}$ which respectevely checks:

$$l_{T_{inf}}(T) = \begin{cases} 1 & \text{if } T = T_{inf} \\ 0 & \text{if } T = T_{sup} \end{cases} \text{ and } l_{p_{inf}}(p) = \begin{cases} 1 & \text{if } p = p_{inf} \\ 0 & \text{if } p = p_{sup} \end{cases}$$

Also let's define:

Let $l_{T_{sup}}(T) = \frac{T-T_{inf}}{T_{sup}-T_{inf}}$ and $l_{p_{sup}}(p) = \frac{p-p_{inf}}{p_{sup}-p_{inf}}$ which respectevely checks also similar conditions.

These linear functions are the Lagrange basis functions of first order.

Note: In the code $l_{Tinf}(T) = -\frac{(1-csi)}{2}$, $l_{p_{inf}}(p) = -\frac{(1-eta)}{2}$, $l_{T_{sup}}(T) = \frac{(1+csi)}{2}$, $l_{p_{sup}}(p) = \frac{(1+eta)}{2}$

From the Lagrange polynomials, we associate a bilinear functions at each nodes:

  - $N_0(T,p) = l_{T_{inf}}(T) \times l_{p_{inf}}(p)$ corresponds to the node $n_0 = (T_{inf}, p_{inf})$
  - $N_1(T,p) = l_{T_{sup}}(T) \times l_{p_{inf}}(p)$ corresponds to the node $n_1 = (T_{sup}, p_{inf})$
  - $N_2(T,p) = l_{T_{sup}}(T) \times l_{p_{sup}}(p)$ corresponds to the node $n_2 = (T_{sup}, p_{sup})$
  - $N_3(T,p) = l_{T_{inf}}(T) \times l_{p_{sup}}(p)$ corresponds to the node $n_3 = (T_{inf}, p_{sup})$

- Properties of the shape function:
  - The shape functions N_0,N_1,N_2 and N_3 are bilinear functions of T and p.
  - For each $i \in \{0,1,2,3\}$, $N_i(n_j) = \delta_{ij}$

The interpolation for the speed sound, for a given (T,p), reads:

$$a_{approx}(T,p) = \frac{1}{4}\sum_{i=0}^{3}\left[N_i(T,p) \times a(n_i)\right]$$

3.3.2 Logarithmic scale for the pressure array

As previously mentioned, to run the case with the lookup table, the user needs to enter boundary values for the temperature and the pressure. For our specific application, the boundaries are the following:

- For the temperature: $T_{min} = 100K$ and $T_{max} = 7\,000K$
- For the pressure: $p_{min} = 800Pa$ and $p_{max} = 400\,000Pa$

As you can see, the interval for the pressure values is very large. Therefore, instead of building a uniform linear scale for the pressure, we decide to implement a logarithmic scale that allows a large range of values to be displayed. Here is the method that I used to implement a logarithmic scale for the pressure.

Method to implement a logarithmic scale illustrate by an example

INPUT: p_{min}, p_{max} and Δp

1. Write p_{min} and p_{max} in a scientific expression.

   $p_{min} = a_{min} + 10^{b_{min}}$ and $p_{max} = a_{max} + 10^{b_{max}}$

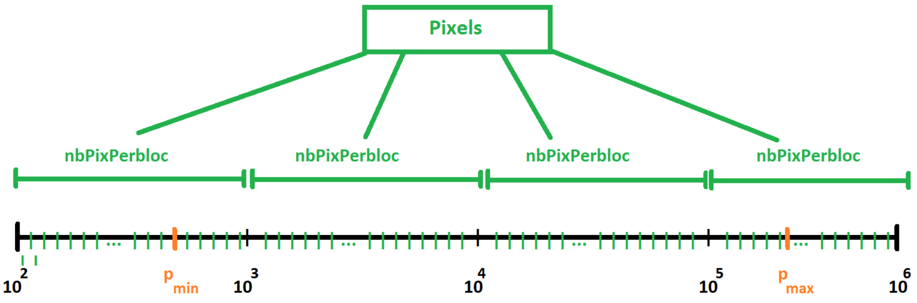
2. Get the number of subintervals, that we call pixels $= \frac{p_{max}-p_{min}}{\Delta p}$

3. In order to distribute the same values of pixels in each logarithmic blocs. We need first to get the number of blocs. For instance, if $p_{min} = 800$ and $p_{max} = 400\,000$, we have

   $10^2 \leq p_{min} \leq 10^3 \leq 10^4 \leq 10^5 \leq p_{max} \leq 10^6$

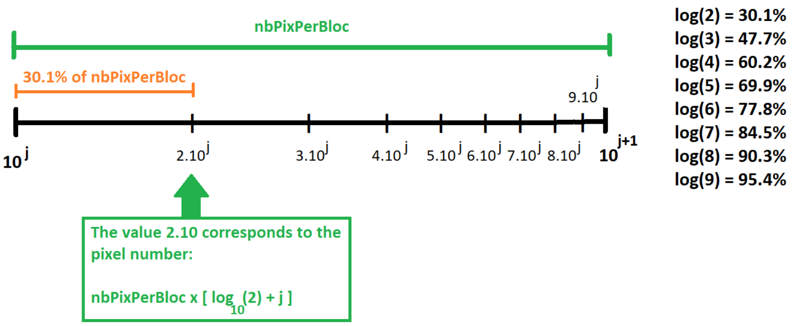
   which means that we have **4 logarithmic blocs**.

4. In each blocs, we distribute the same amount of pixels. In our example, each bloc will have the floor approximation of $nbPixPerBloc = \frac{pixels}{4}$

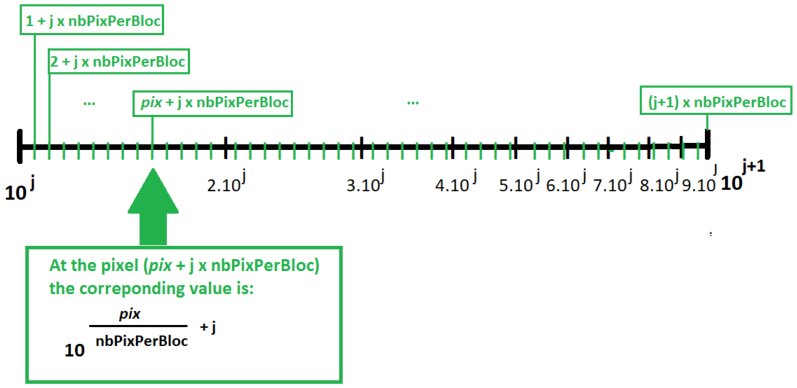


5. Using that $log(i * 10^j) = j + log(i)$. For each bloc, we set the values $i \times 10^j$ with $i \in \{1, 2, \ldots, 9\}$ and $j \in \{j_0 = 2, j_1 = 3, j_2 = 4, j_3 = 5\}$.

   Otherwise, in the block k, the value corresponding to $i \times 10^{j_k}$ is $nbPixPerBloc \times (log_{10}(i) + k)$, $k \in \{0, 1, 2, 3\}$ (index for each blocs)

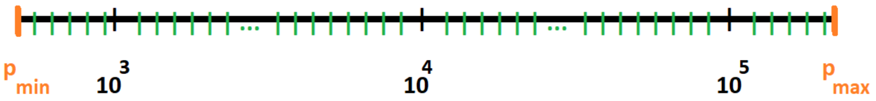


6. Then, we set the other values between the ones set in the previous step:
   For each bloc k, the corresponding value into the pixel $pix + k \times NbPixPerBloc$, where $pix \in [\![1, NbPixPerBloc]\!]$ is $10^{\frac{pix}{NbPixPerbloc} + j_k}$



7. Finally we trunc the excessive values that are less than p_{min} and up to p_{max} .

   In other words, we keep only the values $p_i \in [p_{min}, p_{max}]$



The relative error made with a fixed value of pressure, is shown in **annex 4**.

4 Results and discussion

4.1 Solution of the study case

This subsection shows the solution of the boundary flow of the blunted cone in a local thermodynamic regime. The computation has been made on a **500K element mesh**.

The following plots shown are the solution for the temperature, the pressure, and the Mach number. As John Anderson Jr., mentioned it in his book **[4]**, the flow reaches a very high-temperature flow in the nose region of the blunt cone. Note that the Mach number is null at the wall because of the no-slip boundary condition (no tangential speed).

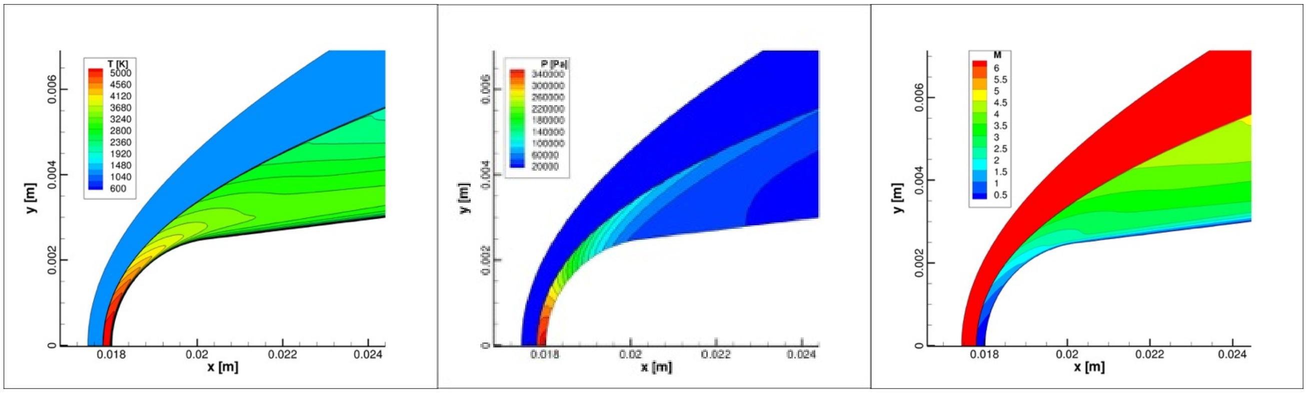


Figure 17: Solution of the temperature and the pressure

The **figure 18a and 18b** show the values of temperature and pressure along the stagnation line. We can see that on the wall, the temperature is equal to the isothermal temperature imposed (293K). The **figure 18c** shows the difference of shock position when computing the pressure stagnation line on a 10K and 500K meshes and highlights the importance of the use of a well-designed original grid to achieve the desired accuracy (i.e. accurate shock position)

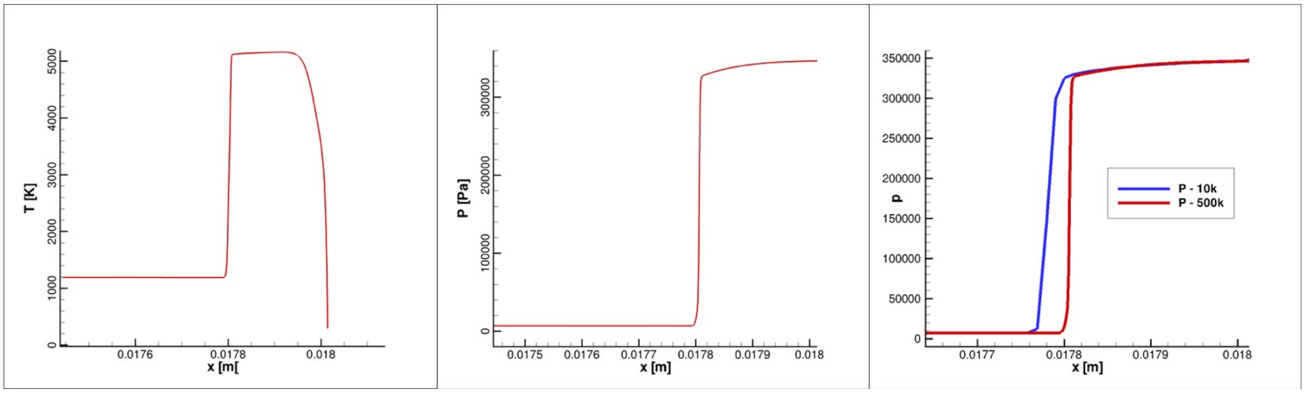


Figure 18: Results along the stagnation line

The mass fraction of the 5 species (AIR5) are in the Annex 3.

4.2 Residual temperature behavior with the lookup Table

In this subsection, with the 10K elements mesh, we show the results obtained by using the lookup table (**LKT**). Recall that this latter, interpolate the thermodynamic values in a table set before the computation. Knowing that these variables are highly non linear, the interpolation should save a lot of CPU time.

For this first test, we run, **in serial**, our case with and without the LKT with the same CFL behavior. We impose a residual temperature precision of 10^{-4}. Both cases have reached this precision. Recall that the LKT requires the following inputs:

- For the temperature: $T_{min} = 100K$, $T_{max} = 7\,000K$ and $\Delta T = 10$
- For the pressure: $p_{min} = 800Pa$ and $p_{max} = 400\,000Pa$ and $\Delta p = 100$

We use a linear scale for the range of temperature and the logarithmic scale (detailed in the subsection 3.3.2) for the pressure.

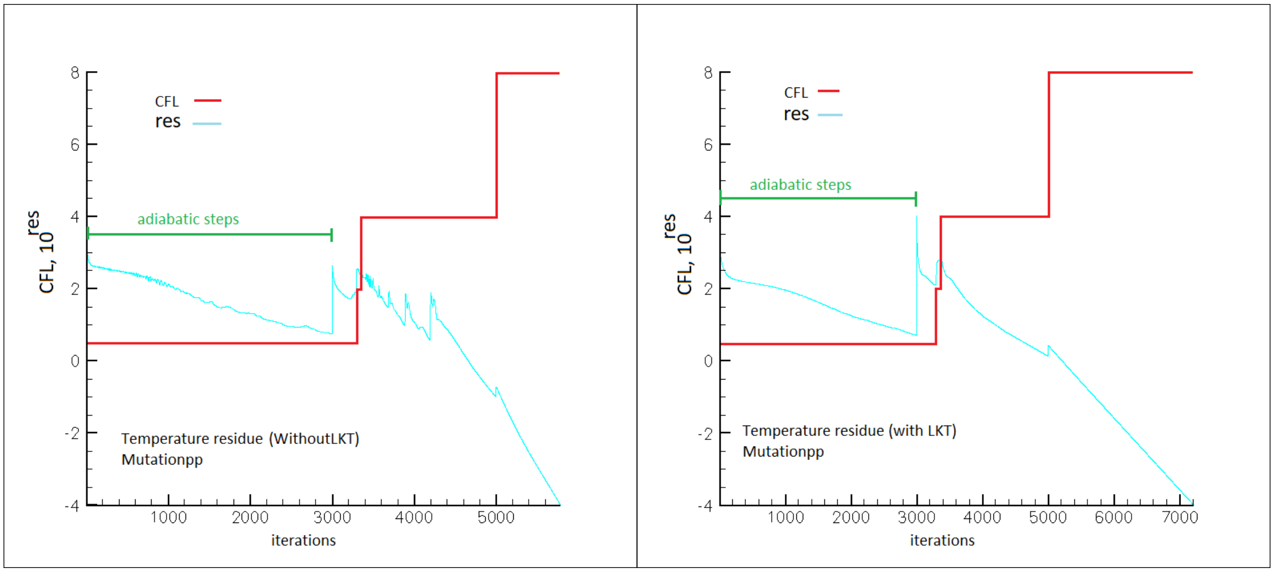


Figure 19: without and with LKT residual SERIAL

The caracteristics for the convergence are summarized in the following table:

Without 2nd order reconstruction	without LKT	with LKT
Number of iterations	5788	7218
CPU time	7 h 58 min 32.7446 sec	4 h 15 min 56.4669 sec

With the LKT, we reached the accuracy twice faster than without the LKT. Let us point out that the behavior of the residual temperature, after the adiabatic steps, is different. The transition to the isothermal steps is more critical with the LKT than without. That is why the computation crashed many times around 3000 iterations. The manipulation of the CFL behavior allowed to avoid the crash and reached the desired precision.

To run the case with the second order reconstruction, I used the same CFL behavior as before. As shown in figure 20, the desired precision has been reached after 9478 steps in 6 h 19 min 34.4027 sec.

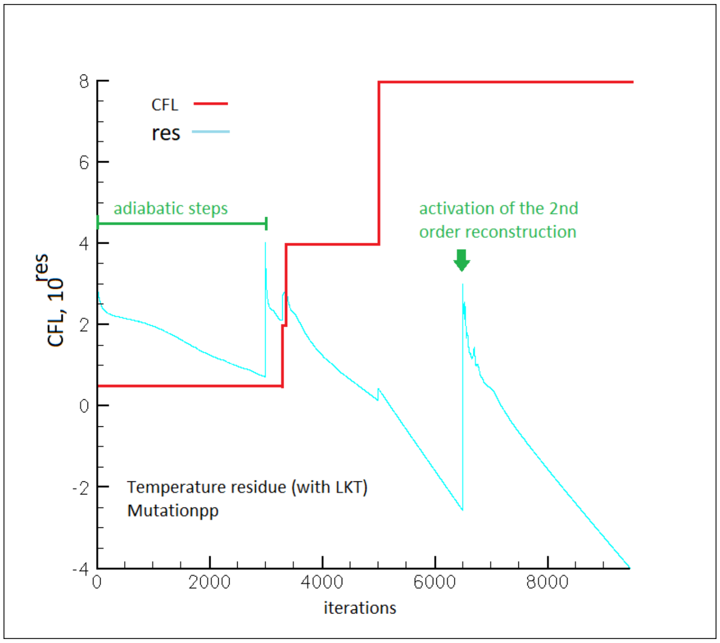


Figure 20: residual temperature 2nd order accuracy (with LKT - serial)

Similarly to what has been done in the **subsection 3.2.3**, I tried to reduce the number of iterations of the residual temperature obtained with the LKT (**figure 20**). For that, I built a function with the interactive parameters (CFL, second order reconstruction and the limiter). The result is shown in the following figure.

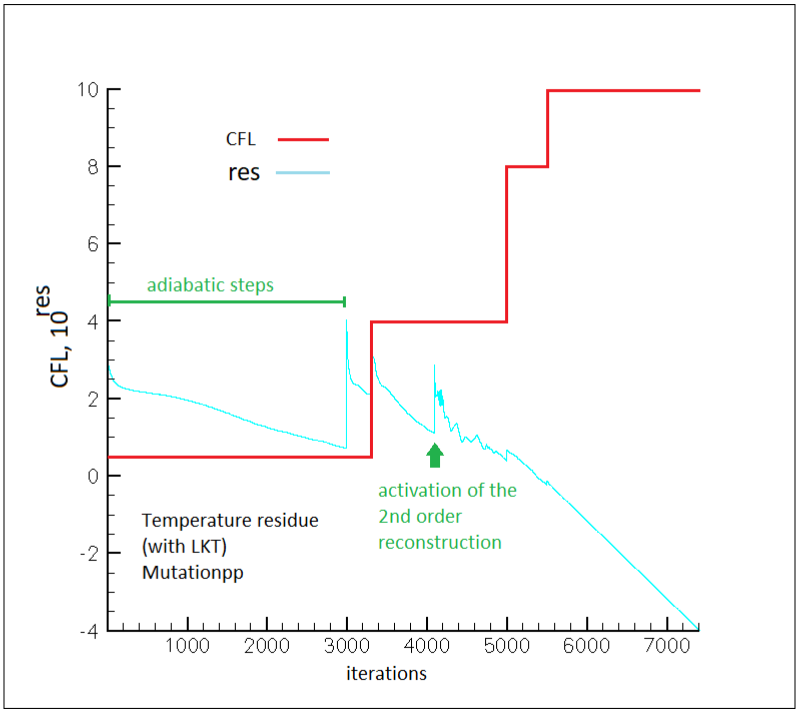


Figure 21: with LKT residual 2nd order serial

Number of iterations: 7413
CPU time: 4 h 35 min 26.6752 sec

As we can see, with controling the behavior of the CFL, we gain 2065 iterations and saved almost 2 hours (CPU time).

4.3 Discussion and suggestion of improvement

In the following table, we summarize the best results obtained with and without the LKT:

	without LKT (**parallel - 4** threads)	with LKT (**serial**)
WITHOUT 2nd Order reconstruction	**iter** = 6130 **; CPU time** = 3 h 57 min 42.8916 sec	**iter** = 7218 **; CPU time** = 4 h 15 min 56.4669 sec
WITH 2nd Order reconstruction	**iter** = 9232 **; CPU time** = 6 h 57 min 46.8631 sec	**iter** = 7413 **; CPU time** = 4 h 35 min 26.6752 sec

Without the second order reconstruction, we obtain roughly the same CPU time with and without the LKT. Recall that the case without LKT is computed in parallel with 4 CPU-threads whereas the case with LKT is computed in serial. It means that, for this case, the computation with the LKT is 4 times faster.

With the second order reconstruction, we can see that the convergence with the LKT, is also the better in terms of CPU time.

As shown by these results, the efficiency of the lookup table is non negligible. This method can be improved:

- by including other thermodynamic variables in the lookup Table (the viscosity and the conductivity)

- by implementing a new method of interpolation that will include the derivative of the polynomials in the bilinear Lagrange interpolation, like the bicubic spline method **[15]**.
- by giving a better measurement of the method accuracy (heat flux)

All the results file mentioned in this report are in the following Github repository:

`https://github.com/SanaAmri/LTE_NS_Cone/`

Conclusion

In the present work, a lookup table to interpolation the thermodynamic variables have been implemented into the **Mutationpp** plugin from COOLFLuiD. Recall that the thermodynamic variables require only a given set of temperature and pressure to be computed (LTE assumption). That is why the implementation of the lookup table needed a range of discretized values of temperature and pressure. Because of the large range of pressure values (800Pa → 400 000Pa), a logarithmic scale has been implemented to get the required array.

The objectives of this project were determined as follow:

- Discover the world of aerospace by studying the main phenomena occurring during the atmospheric reentry of a vehicle
- Get familiar with the COOLFluiD platform in order to be able to carry on numerical investigations of the specific test-case
- Optimize and adapt the choice of the different numerical methods and tools used to achieve an optimal set-up for the simulations
- Development and implementation of a Lookup table to enhance the performance of the solver in terms of memory usage and computational time

The engineering environment brought complementary skills to my applied mathematics and scientific computing method background. Using a research code (COOLFluiD) for industrial application allowed me to apply the theoretical concept and apply a systematic approach to define a configuration file (CFL, activation of the second order of accuracy, limiter). Finally, The implementation of a lookup table to get the thermodynamic variables proof to be very efficient in computation.

5 APPENDIX

5.1 Annex 1

From the technical note [2]

- **The convective flux vectors** in the axial direction x and the radial direction r are:

$$\boldsymbol{F}_x^c(\boldsymbol{U}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix} \qquad \boldsymbol{F}_r^c(\boldsymbol{U}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{pmatrix}$$

$H = E + \frac{p}{\rho}$ is the total ethalpy and p is the static pressure.

- **The diffusive flux vectors** in the axial direction x and the radial direction r are:

$$\boldsymbol{F}_x^d(\boldsymbol{U}) = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xr} \\ -q_x + \tau_{xx}u + \tau_{xr}v \end{pmatrix} \qquad \boldsymbol{F}_r^d(\boldsymbol{U}) = \begin{pmatrix} 0 \\ \tau_{rx} \\ \tau_{rr} \\ -q_r + \tau_{rx}u + \tau_{rr}v \end{pmatrix}$$

- The **source term** vector S is:

$$\boldsymbol{S} = \begin{pmatrix} 0 \\ 0 \\ p - \tau_{\theta\theta} \\ 0 \end{pmatrix}$$

- The **stresses** in the viscous flux are given by:

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r}\right) \qquad \tau_{rr} = 2\mu\frac{\partial v}{\partial r} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r}\right)$$

$$\tau_{xr} = \tau_{rx} = \mu\left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right) \qquad \tau_{\theta\theta} = -\frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} - 2\frac{v}{r}\right)$$

where $\mu = 1.458 \cdot 10^{-6}\frac{T^{\frac{3}{2}}}{T+110.4}$ [Kg/m s] is the laminar viscosity (according to Sutherland's law).

The heat transfer is modeled according to Fourier's law:

$$q_x = -k\frac{\partial T}{\partial x} \qquad q_r = -k\frac{\partial T}{\partial r}$$

with $k = \mu\frac{c_p}{Pr}$, P_r=0.72 is the Prandtl number and c_p the specific heat coefficient at constant pressure.

5.2 Annex 2

Refers to the figure 14.

Function without 2^{nd} order reconstruction

```
Gradient= 0. → Classic CCFVM, $1^{rst}$ order accuracy
WHILE residual temperature $> 10^{-4}$
    IF(iter<3800)
        CFL=1.0
    ELSE IF(3800≤iter<4200)
        CFL=2.0
    ELSE IF(4200≤iter<5200)
        CFL=4.0
    ELSE (9000≤iter)
        CFL=8.0
    END IF

END WHILE
```

Function with 2^{nd} order reconstruction

```
Gradient= 0. → Classic CCFVM, $1^{rst}$ order accuracy
WHILE residual temperature $> 10^{-4}$
    IF(iter<5800)
        CFL=1.0
        IF(iter=5200)
            Gradient=1. → Transition to second order accuracy
        END IF
    ELSE IF(5800≤iter<6500)
        CFL=2.0
    ELSE IF(6500≤iter<8500)
        CFL=4.0
        IF(iter=8000)
            → Freeze the factor limiter from the piecewise
              polynomials approximation of each cells
        END IF
    ELSE IF (8500≤iter<9000)
        CFL=8.0
    ELSE (9000≤iter)
        CFL=10.0
    END IF
END WHILE
```

5.3 Annex 3

Solution of the mass fractions for the 500K elements mesh.

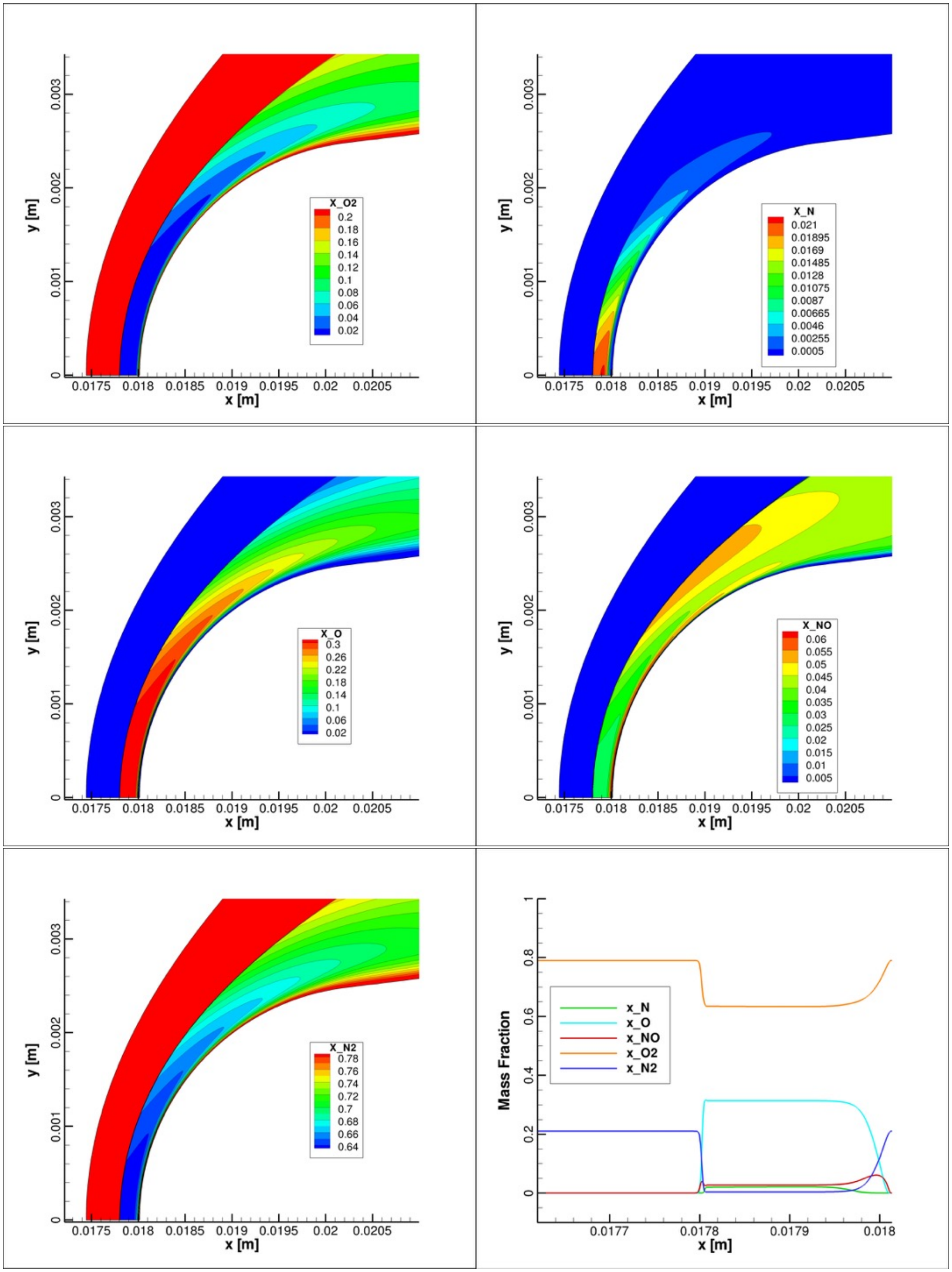


Figure 22: Mass fractions solutions - 500K mesh

5.4 Annex 4

Solution of the mass fractions for the 500K elements mesh.

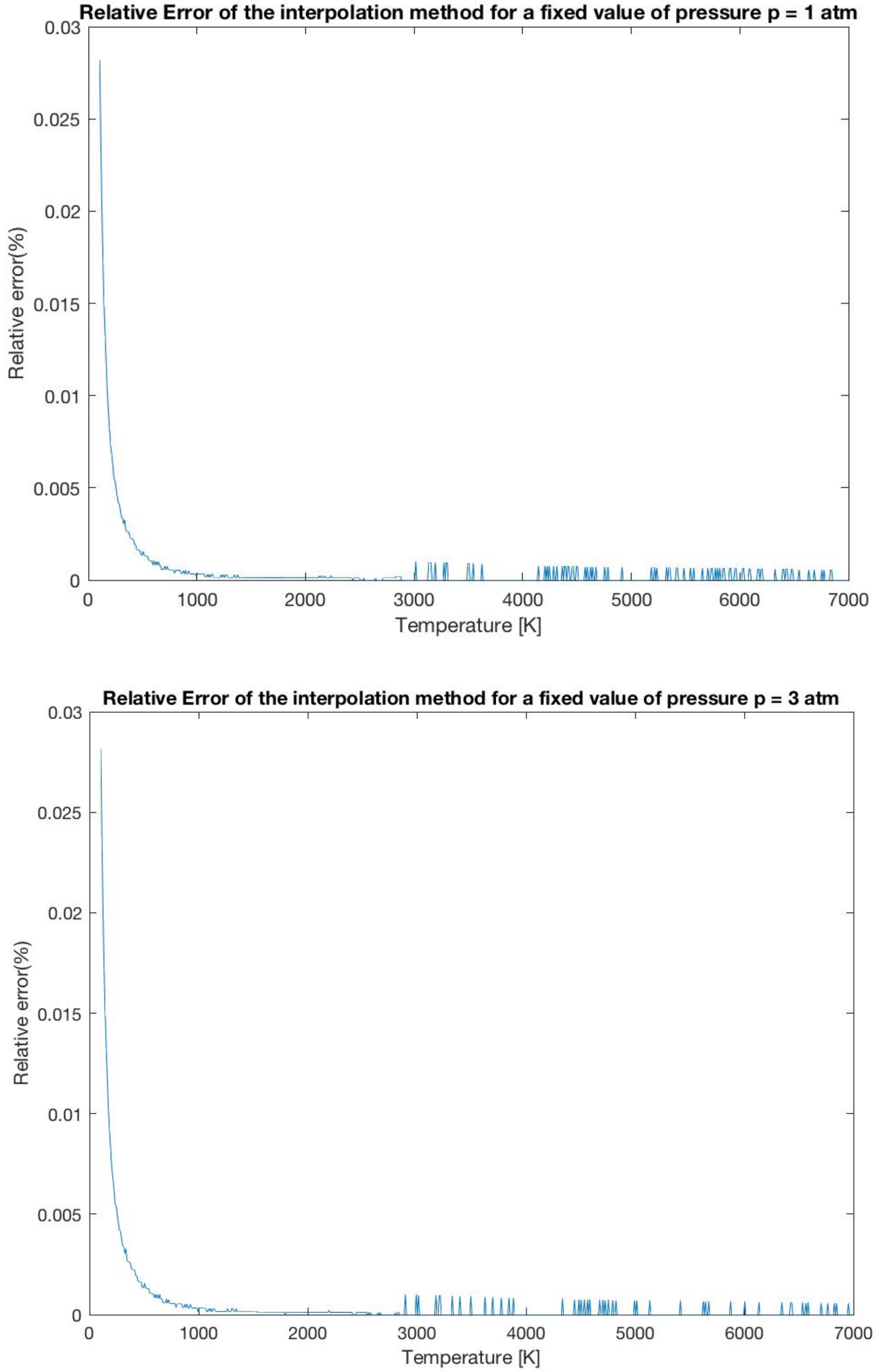


Figure 23: Relative error bilinear interpolation method

References

[1] Andrea Lani, An object Oriented and High Performance Platform for Aerothermodynamics Simulation. Phd thesis, Université Libre de Bruxelles, von Karman Institute for FluiD Dynamics, 2009.

[2] R. Broglia, M.Manna, H. Deconinck, G. Degrez, Development and validation of an axisymmetric Navier-Sytokes solver for hypersonic flows. Technical note 188, von Karman Institute for Fluid Dynamics, May 1995.

[3] A. Alvarez Laguna , A. Lani , H. Deconinck , N.N. Mansour , S. Poedts, A fully-implicit finite-volume method for multi-fluid reactive and collisional magnetized plasmas on unstructured meshes. Journal of Computaional Physics, 2016.

[4] John D. Anderson Jr., Hypersonic and high temperature gas dynamics, second edition. American Institute of aeronautics and Astronautics, Inc, 2016.

[5] John D. Anderson Jr., Fundamentals of aerodynamics, second edition. McGrw-Hill, Inc, 1991.

[6] Franck M.White, viscous fluid flow, second edition, McGrw-Hill, Inc, 1991.

[7] Guillaume Grossir, Longshot hypersonic wind tunel flow characterization and boundary layer stability investigation. PhD Thesis, Université libre de Bruxelles, von Karman Institute for fluid dynamics, July 2015

[8] Bernard Grossman, Fundamental concepts of real gasdynamics, Lecture notes, Virginia Tech, Blacksburg, US, January 2000.

[9] Culbert B.Laney, Computational gasdynamics. Cambridge University Press, 1998.

[10] Steven P.Shneider, Hypersonic boundary-layer transition on blunt bodies with roughness. AIAA Aerospace Sciences Meetin and Exhibit, January 2008.

[11] Andrea Lani, Janos Molnar, David Vanden Abeele, Pietro Rini, Thierry Magin, Gérard Degrez, Numerical study of elemental demixing in atmospheric entry flow regimes near local thermodynamic equilibrium, European conference on computational fluid dynamics ECCOMAS CFD, 2006.

[12] Davide Masutti, Fabio Pinna, Erkan Gunaydinoglu, Tamara Sopek, Oliver Chazot, Natural and induced transition on a 7deg half-cone at Mach 6, RTO-MP-AVT-200.

[13] Viola Wartemann, Alexander Wagner, Ross Wagnild, Fabio Pinna, Fernando Miro Miro, Hideyuki Tanno, Code to code comparison on hypersonic high enthalpy transitional boundary layers, American Institute of Aeronautics and Astronautics. (configuration of the experiment on the cone)

[14] F. Moukalled, L. Mangani, M.Darwish, The finite volume method in computational fluid dynamics, volume 113, Springer, 2016.

[15] Xiao Shu, Bicubic interpolation, course,Electrical and Computer Engineering McMaster Univerty, Canada. March, 2013