PlasFlowSolver: Data Reduction Aerothermodynamic Model of the Plasmatron X Wind Tunnel

Domenico Lanza

December 17, 2024

Abstract

In this manual I will discuss a data reduction aerothermodynamic model of the Plasmatron X Wind Tunnel, and the software that I built in order to compute the flow propeties from it. I will start by discussing the model and the hypotheses made in order to solve the problem. I will then discuss the software, how I built it, and how to use it. I will then discuss the code verification and validation, and I will conclude with some future improvements that can be made to the software.

Contents

Chapter 1

Introduction to the manual, motivations and previous works

In this manual I will present an engineering data reduction aerothermodynamic model of the Plasmatron X Wind Tunnel, an inductively-couple plasma wind tunnel built at the University of Illinois at Urbana-Champaign.
This wind tunnel is the largest inductively coupled plasma wind tunnel in the U.S.[2], and its main purpose is to test materials to find out whether they are survivable for extreme hypersonic flight and re-entry, and to develop profiles of the conditions that would be found throughout a mission or during re-entry.

1.1 High temperature effects in hypersonic flights

Although specific authors will disagree on its exact definition, the hypersonic regime can be identified for the present work as the range of Mach numbers for which there exist thin shock layers, entropy layers, important viscous interactions and where the flow temperature has a non-negligible impact on the aerodynamics[1].
The stagnation regions of an hypersonic vehicle are actually flying behind the shock wave, in a region of hot subsonic flow, where high temperature effects are important. The terms used to characterise such flow fiels, in which thermal effects and chemical effects are present and important for the vehicle aerodynmics, are *aerothermodynamics* and *aerothermochemistry*[12].
Among the high temperature effects, we can identify[4, 3]:

1. Change in the perfect gas behaviour: due to the excitation of vibrational states within molecules and electronic states within atoms, and molecules (at temperatures above 500 K), the perfect gas behaviour changes from *calorically perfect state*, known at ambient temperatures, for which one has:

$$c_p = \left(\frac{\partial h}{\partial T}\right)_p = \text{constant} \;\Rightarrow\; h = c_p T + h_0 \tag{1.1}$$

   to the thermally perfect state, for which the specific hat varies with the temperature:

$$c_p = \left(\frac{\partial h}{\partial T}\right)_p \;\Rightarrow\; h = \int_{T_0}^{T} c_p(T)dT + h_0 \tag{1.2}$$

   This change in the specific heat at constant pressure c_p is due to the excitation of the internal energy modes of the gas atoms and molecles, that at room temperature are dormant.

2. Change in molar mass: due to the dissociation of molecules into atoms and the ionisation of molecules and atoms into ions and ionised molecules (at temperatures above 2000 K and 6000 K respectively), the molar mass of the gas changes, and so does the specific gas constant:

$$p = \rho R T \text{ but } R = \frac{\mathcal{R}}{MW} = R(\rho, T) \tag{1.3}$$

   this increase the enthalpy content of the gas as the reverse recombination reactions release the formation energy of the species.
   All the thermodynamic properties become functions of the chemical composition, i.e. of two thermodynamic variables in the simple case of chemical equilibrium (i.e. the temperature and the pressure).

$$h = h(T, p) \text{ , } e = e(T, p) \text{ and } s = s(T, p) \tag{1.4}$$

3. Flow in chemical non-equilibrium: in regions of moderate to high velocity, the chemical times may be larger than the transit time of the atoms and molecules through the facility, resulting in a flow in chemical non-equilibrium.
4. Flow in thermal non-equilibrium: The relaxation times may also be larger, leading to thermal (usually vibrational) non-equilibrium. This means that the internal energy modes are out of euilibrium with respect to the transaltional one. For example, when a fuid element crosses a shock, the translational energy of the fluid particles is suddenly increased. It is well known that a high number of collisions is needed to equilibrate the internal energy modes with the translational one. Therefore, behind the shock, there will be a relaxation region where the internal energy modes will try to catch up the translational one. Another example is when the fluid experiences a strong expansion. In this case the translational energy will rapidly decrease because of the expansion, but the internal one will remain higher. When the mixture is ionized, an additional source of thermal nonequilibrium appears because energy exchange between mixture components and free electrons is highly inefficient due to the large mass disparity: in this case the translational temperature of electrons can be different from the one of heavy particles.
5. Plasma formation: If the temperature is high enough, ionization can occur and the gas becomes a partially ionized plasma, with a finite electrical conductivity. Therefore electromagnetic fields and associated forces, either selfinduced or applied from an external source can act on the fluid, appreciably changing its behaviour with respect to a neutral one.
6. Radiation: At high temperature (above 10'000 K for air) radiation emitted and absorbed by the gas can become important and eventually modify the energy distribuition in the flowfield.
7. Heat transfer rates: the heat transfer rates to the body are affected by radiation and by chemistry, through effects of diffusion of species across the boundary layer, of heat of recombination of the colder wall, and of catalytic effects of the wall material upon surface chemistry (usually such reactions have the negative property of increasing the heat flux experienced by the vehicle).

1.2 What is a ICP Wind Tunnel

It is impossible to perform exact simulations of the hypersonic reacting flows around hypersonic bodies in wind tunnels: this would require total enthalpy levels and Mach number ranges incompatible with the technology.

The TPS testing requires long-duration facilities, since that the degradation of the material over time must be carefully observerd. In addition, the flow field has to be as chemically pure as possible, the reason being the catalycity of the sample surface.

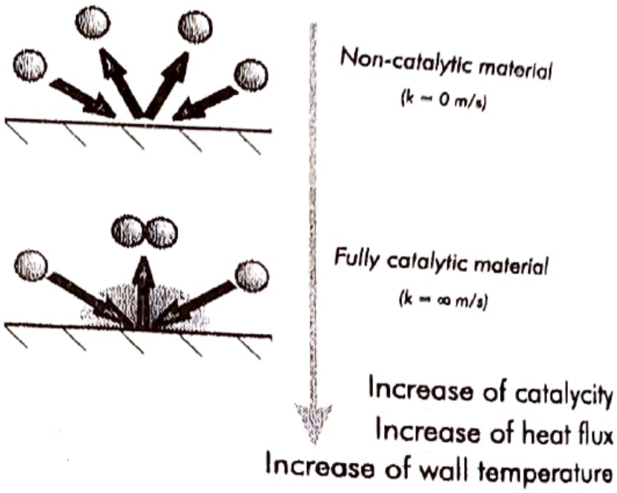

Figure 1.1: Effect of surface catalycity on heat flux

Figure 1.1 shows the difference between a non-catalytic and a fully catalytic surface. In the former case, the speed with which dissociated species recombine at the surface (noted k, catalytic recombination speed) is zero. Consenquently, there is no reaction taking place on the surface. In the latter case, that speed tends to infinity and the dissociated species will recombine at the surface. In so doing, they free up their formation energy, which goes into the wall. Thus, as wall catalycity increases, the conductive heat transfer is increased too, and so is the equilibrium surface temperature.

TPS materials are highly non-catalytic, in order to reduce the heat flux to the wall. However, catalycity also depends on the test gas, so combustion-heated facilities, which produce high-enthalpy gas by burning fuel, air and oxygen and therefore actually use products of combustion as test use gas, should be disregarded for TPS testing purposes.

In order to obtain long-duration, high enthapy flows suitable for re-entry TPS testing, one needs to use continuous tunnels capable of producing plasma, as: combustion-heated facilities, arc heaters, MGD thrusters or inductively-coupled plasma (ICP) heaters.

In contrast to arc-heated facilities, inductively-coupled plasma (ICP) facilities have **no contamination at all** as the plasma is never in direct contact with electrodes. The Plasmatron X Wind Tunnel is an ICP facility as said before.

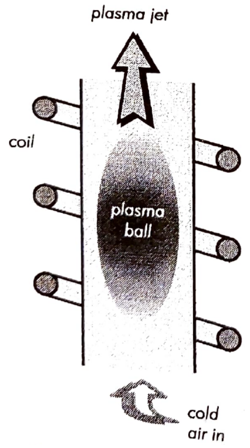

Figure 1.2: Concept of the inductively-coupled plasma torch.

The concept behind plasma generation by induction is sketched in Figure 1.2.
A coil surrounds a quartz tube in which cold gas in injected. This coil is connected to a high-frequency generator and is traversed by high voltage, high frequency current. Due to these conditions, induced electric and magnetic fields exist inside the quartz tube, with induction lines parallel to the axis, as in a classical solenoid. The periodic time variation of the induction creates circular, oscillating electric fields in planes perpendicular to the tube axis. These fields can move the free electrons existing in the gas, creating current loops (*eddy currents*) heating the gas by Joule effect. Once the gas heats up, it dissociates and ionises into a plasma. The first free electrons can be obtained by introducting an electrode into the plasma torch, or, **preferably**, by reducing the pressure under 0.1 mbar and applying voltage on the coil: the electric field gradients apperaring in the vicinity of the coil (between turns) are strong enough to cause a local ionisation of the gas in the tube. Once the first free electrons have appeared, the whole gas in rapidly heated by Joule effect. Argon is used in order to facilitate the initial electric discharge because of the longer lifetime of its free electrons at low pressure compared to the air plasma case. By these means plasma can be generated from low to atmospheric pressures with temperatures of the order of 6000 K to 12'000 K. The reader may refer to Bottìn B. Thesis[4] for a more detailed explanation of the ICP torch.

1.3 Motivations and needs for a data reduction model of the Plasmatron X Wind Tunnel

The purpose of this project is to provide a mathematical data reduction model capable of extrapolating the flow conditions inside of the Plasmatron X Wind Tunnel by experimental measuraments.

The complete, most accurate model in order to **predict** the flow properties should containg the elements listed below[4]:

- 3D coupled CFD solver for Navier-Stokes equations and Maxwell equations (the flow in the torch is governed by magnetohydrodynamic effects abd flow is injected in a swirling motion for plasma stability reasons).
- Revised equation of state and enthalpy definition to take into account the chemical effects.
- Transport properties for high-temperature gases (including radiation effects).
- Finite rate chemistry model and vibration relaxation as the flow can be in thermal and chemical non-equilibrium.
- Turbulence model suited to plasma flows, because the big temperature differences within the flow cause zones of laminar flow (high viscosity) and turbulent flow (low viscosity).
- multi-domain grid for the computation, as the complete flow solution in the test enclosure is required in order to make accurate predictions of sample stagnation point heat flux.

Clearly, such an elaborate model is too complex to obtain, but there are some researchers take achived in including many of the hypotheses described above in their models (i.e. D. Vanden Abeele work[14]).

While these complex models do provide useful information on the underlying physical phenomena and key values, they require an high computational power and high computational times. Due to this facts, simpler models, like 1-dimensional or 2-dimensionals models, that can provide good approximation of the solution with a shorter computational time and load, have been developed.

But we should always keep in mind that any prediction made with a computational model should be thoroughly validated using suitable measurement techniques. In this project, I will concentrate only on the measurement of stagnation pressure and heat flux, as these are the parameters on which the design requirements are based.

The main purpose of this project is to provide a data reduction model capable of extrapolate the flow conditions inside the Plasmatron X Wind Tunnel by experimental measurements, under a set of hypotheses that will be discussed in the next chapter. In fact, taking into account all the phenomena listed above would be too complex and computational expensive, so I focused on a simpler model, that can provide first order approximation of the solution with a shorter computational time and load.

***PlasFlowSolver** is an aerothermodynamic software meant to extrapolate the flow properties of the flow inside the Plasmatron X Wind Tunnel by using a data reduction model, in order to obtain first order approximations with fast computational times.*

The model used and the exact hypotheses are discussed in the next chapter.

1.4 Some applications for aerothermodynamic models

Space travel to the nearby Earth orbit or to distant celestial bodies imposes many stringent requirements upon technology. Among those are the so-called high-temperature effects that appear during either the vehicle ascent or descent phase, when flight velocity within the planetary atmosphere is large enough to cause wall heat fluxes and temperatures to increase. High temperature effects are responsible of tremendous heat loads to the vehicle front, which has to be adequately protected if it has to survive the descent phase. Thus a variety of thermal protection systems (TPS) have been developed, many consisting of tiles of ceramiclike materials such as SiC. It is of course essential that such materials be tested in facilities on the ground before certification for flight, as the inadequacy of such a material could prove fatal to the crew and/or to an expensive payload[4].

Other applications from Barbante's thesis[3] are:

Re-entry vehicles When an aerospace vehicle, as for example the Apollo capsule or the Space Shuttle, travels at high speed through the atmosphere, a strong shock is formed in front of it. A major part of the kinetic energy of the free stream is converted into thermal energy across the shock and therefore high temperature is reached in the region between the shock and the body (the shock layer), especially around the body nose, where the shock reaches its maximum intensity. Downstream of the nose region the gas expands and reaches again a high Mach number: the intense friction happening in the boundary layer increases its temperature triggering further chemical reactions. If temperature in the shock layer is high enough the gas can ionize: the free electrons that are created absorb radio waves and cause communication blackout to and from the vehicle. This is a serious problem and an accurate prediction of the electrons number density in the shock layer is important. Emission and absorption of radiation can occur and, besides advecting the state of the gas surrounding the vehicle, can raise the heat flux experienced by the vehicle itself. Radiation from the hot vehicle wall to the ambient atmosphere can have a significant cooling effect and must be taken into account in the thermal boundary condition.

High enthalpy wind tunnels: Conventional hypersonic wind tunnels are not able to simulate nonequilibriums over bodies, because the maximum attainable temperature is not enough to trigger high temperature effects. Facilities that can provide the needed temperature level in a continuous way (testing time of the order of minutes) are arc heaters and inductively coupled plasma (ICP) heaters. In arc heaters the testing gas is heated to high temperature by an electric arc discharge in the reservoir and then the chemically reacting mixture expands through a hypersonic nozzle in the test section where the model is located. In this kind of facility, temperature are both high, approaching real flight conditions. A main drawback of arc heaters is that the copper electrodes (used to generate the electric arc) are progressively eroded and the gas is contaminated by metal vapor. The metal vapor is deposited on the testing model and changes its catalytic properties: this effect is particularly adverse when aging tests on the model material are conducted. Although the pollution levels are very low in state-of-the-art arc heaters (copper fraction as low as 1 PPM), many researchers, especially from Russia, prefer to use ICP heaters. In a ICP heater the plasma is generated by electromagnetic induction: a coil, traversed by a high frequency, high voltage current, surrounds a quartz tube in which cold gas is injected. Induced electromagnetic oscillating fields exist inside the quartz tube. These fields move the free electrons present in the gas and create currents that heat the gas by Joule effect. The heated gas dissociates and ionizes into a plasma. These facilities show no pollution of the gas because there is no direct contact with the coils and the testing time can be, at least in theory, indefinitely long. Usually these facilities operate at low Mach number regime, even if they can be suitably adapted to produce supersonic flow.

Ramjet and Scramjet engines A Ramjet engine is essentially a duct where supersonic air is slowed down to subsonic speed at the entrance of the combustor. Fuel is injected in the combustor, the mixture burns and expands through the nozzle. Ramjets have advantages over conventional turbine engines in the Mach number regime from 2 to 5. However, some design concepts of hypersonic airbreathing transport vehicles assume a flight Mach number well in excess of 10. Under such conditions, if the incoming air is decelerated to subsonic speed, it attains a temperature that is above the adiabatic same temperature of the fuel-air burning process in the combustor and therefore no combustion can take place. A possible solution is to keep the incoming air stream at supersonic speed in the combustor: in this way air temperature is kept below the same adiabatic temperature and combustion can take place. The major drawback is that the combustion has to take place in a supersonic stream, leading to tremendous practical problems (same stabilization, efficient mixing and burning) that are still not solved nowadays.

1.5 Previous works

During this project, I stronly relied on the works of:

- Bottin B.[4]: in this thesis, the author discusses the development of a 1D model for the Plasmatron X Wind Tunnel, and of a data reduction model for the flow properties. His work was the starting point for my project, and I used his model as a reference for my work.
- Barbante P.[3]: in this thesis, the author discusses the development of a 4th order accuracy algorithm for the computation of the stagnation point heat flux. My heat flux algorithm is based on his work, with very little modifications.

Chapter 2

Position of the problem, model and hypotheses

In this chapter I am going to discuss the position of the problem, how to solve it and which model and hypotheses I used.

2.1 Position of the problem

The position of the problem is quite simple: given the values of:

- Static pressure p^o
- Pitot pressure p_t^o
- Stagnation heat flux q^o

as recorded by the Pitot probe and the heat flux probe, what are the values of the flow properties in front of the probe? More precisely, the parameters that are looked for using an iterative scheme are the flow:

- Total pressure p_t
- Total temperature T_t
- Static temperature T
- Velocity u

since that by using them, the flow conditions are completely defined under the hypotheses of the model that I am going to use.

2.2 Flow model and hypotheses

In this section I am going to discuss the flow model and the hypotheses used.

2.2.1 The flow as a continuum

The first hypothesis that I need to make is that the fluid can be described as a continuum. We know that a fluid is made of atoms and molecules that move around and interact with each other, so, in principle, the continuum hypothesis is not valid. However, studying each single molecule is not possible, since that the computational load would be immensively high. The solution is defining the fluid as a deformable continuum, and study the average manifestation of molecular motions. In a continuum, the fluid properties like temperature, density or velocity are defined at every point in space, and these properties are known to be appropriate averages of molecular characteristics in a small region surrounding the point of interest. For example, if we consider the flow velocity[11]:

$$\mathbf{u} = \frac{1}{n}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\mathbf{v}f(\mathbf{v})d^3v \tag{2.1}$$

where $f(\mathbf{v})$ is the Maxwell distribution function of the fluid velocity (random vector variable that represents possible molucular velocities), and n is the number density of the fluid particles.

The continuum approximation is valid at the length scale L (a body length, a pore diameter, a turning radius, etc.) when the **Knudsen number**:

$$Kn = \frac{\lambda}{L} \tag{2.2}$$

where λ is the mean free path, is much less than 1.

The length scale L is the characteristic length of the macroscopic variables gradients. Clearly the density variation is a good indicator of rarefaction of the medium i.e. how far apart the constituent particles are[3]. Thus for characterizing macroscopic variable gradients we can take:

$$L = \frac{\rho}{\frac{\partial \rho}{\partial x}} \tag{2.3}$$

The mean free path λ is the average distance a molecule travels between collisions, and it is given by[11]:

$$\lambda = \frac{\bar{v}}{n\pi \bar{d}^2 \bar{v_r}} \tag{2.4}$$

where $\bar{v}$ is the mean molecular speed, n is the number density, $\bar{d}$ is the average molecular collision diameter and $\bar{v_r}$ is the average relative velocity between molecules, $v_r = \sqrt{\bar{v}}$. If this hypothesis is not valid, a molecular-kinetic theory approach may be necessary for analyzing flows over very small objects or in very narrow passages (where L is small) or in tenuous gases at the upper reaches of the atmosphere (where λ is large).

The Navier-Stokes equations tend to become inapplicable for $Kn > 0.03$ and even when they are approximatly applicable, the usual no-slip boundary conditions are not valid anymore (a solution to this problem is to use a *slip boundary condition*, valid for $Kn < 0.1$[3]).

2.2.2 Governing equations for a mixture of reacting perfect gases

High temperature fluids are generally made of different species, and each one, in the range of pressure and temperature of our interest, behaves with good approximatrion as a **perfect gas**[3].

Plasma can be condisered as a chemically-reacting mixtures of individual species (atoms, molecules, ions and electrons)[4], with density ρ:

$$\rho = \sum_{i=1}^{N_s} \rho_i \tag{2.5}$$

where ρ_i is the density of the i-th species and N_s is the number of species in the mixture, and velocity vector $\mathbf{V}$:

$$\mathbf{V} = \frac{\sum_{i=1}^{N_s} \rho_i \mathbf{V_i}}{\rho} \tag{2.6}$$

where $\mathbf{V_i}$ is the velocity vector of the i-th species. These definitions can be seen as a density average in a control volume ΔV:

$$m = \sum_{i=1}^{N_s} m_i \tag{2.7}$$

$$\lim_{\Delta V \to 0} \frac{m}{\Delta V} = \rho = \lim_{\Delta V \to 0} \sum_{i=1}^{N_s} \frac{m_i}{\Delta V} = \sum_{i=1}^{N_s} \rho_i \tag{2.8}$$

and a linear momentum average in a control volume ΔV:

$$m\mathbf{V} = \sum_{i=1}^{N_s} m\mathbf{V_i} \tag{2.9}$$

$$\lim_{\Delta V \to 0} \frac{m\mathbf{V}}{\Delta V} = \rho \mathbf{V} = \sum_{i=1}^{N_s} \frac{m_i}{\Delta V} \mathbf{V_i} = \sum_{i=1}^{N_s} \rho_i \mathbf{V_i} \tag{2.10}$$

In principle, the governing equations should be derived from a suitable kinetic theory based on a generalized Chapmann-Enskog perturbative solution of the Boltzmann equation accounting for

- 1: thermal nonequilibrium of translational energy,
- 2: the influence of the electromagnetic field
- 3: excitation of internal energy modes through inelastic collisions and radiative processes,

- 4: reactive collisions,
- 5 photochemical processes.

However, no such uniform theory, incorporating all of these phenomena, currently exists. While significant progress has been made in this direction, current state-of-the-art models rely instead on the principle of generalized balancing to describe the evolution of extensive flow field quantities, namely the conservation of total or partial mass, momentum, and energy[13].

The governing equations for a mixture of reacting perfect gases can be derived with the control-volume method, and many textbooks provide a derivation. The 5 conservations equations that I am going to write are the conservation of mass (continuity equation), the conservation of species mass (species continuity equation), the conservation of charge, the conservation of momentum and the conservation of energy.

The continuity equation can be expressed in the conservative form as[3, 11]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \tag{2.11}$$

or in the convective form as:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot (\mathbf{V}) = 0 \tag{2.12}$$

where ρ is the density, $\mathbf{V}$ is the velocity vector and $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{V} \cdot \nabla$ is the material/convective derivative (the derivative taken by following a particular fluid particle). The right side of the global continuity equation is zero, since that there is no destruction or production of matter inside the mixture (no nuclear reactions are considered).

The species continuity equation can be expressed in the conservative form as[3, 13]:

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{V}_i) = \dot{\omega}_i \tag{2.13}$$

where $\dot{\omega}_i$ is the chemical production term (the term that describes the rate of production or depletion) of the i-th species due to (homogeneous) chemical reactions. If radiative processes are considered, a term $\dot{\Phi}_k$ should be added to the right side of the equation, which accounts for mass production due to radiative processes. We will not consider radiative processes, so this term is not included. Species mass production due to ablation and pyrolysis has also been neglected.

It can be useful to define the diffusion flux of the i-th species as:

$$\mathbf{J}_i = \rho_i \mathcal{V}_i \tag{2.14}$$

where $\mathcal{V}_i$ is the diffusion velocity of the i-th species:

$$\mathcal{V}_i = \mathbf{V} - \mathbf{V}_i \tag{2.15}$$

by using the velocity decomposition $\mathbf{V_i} = \mathbf{V} + \mathcal{V}_\rangle$.

By summing up all the species continuity equations it follows that:

$$\sum_{i=1}^{N_s} \dot{w}_i = 0 \tag{2.16}$$

for the conservation of mass.

Furthermore, it is easy to show that the following proprenty for the diffusion fluxes holds:

$$\sum_{i=1}^{N_s} \mathbf{J}_i = 0 \tag{2.17}$$

It is possibile to rewrite the species continuity equation as:

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{V} + \mathbf{J}_i) = \dot{w}_i \tag{2.18}$$

The charge conservation equation can be expressed as[3]:

$$\frac{\partial \mathcal{Q}}{\partial t} + \nabla \cdot (\mathcal{J}_q) = 0 \tag{2.19}$$

where $\mathcal{Q} = \sum_{i=1}^{N_s} q_i \rho_i$ is the total charge and $\mathcal{J}_q = \mathcal{J}_q^c + \mathcal{J}_q^d$ is the total current. Where:

$$\mathcal{J}_q^c = Q\mathbf{V} \tag{2.20}$$

And

$$\mathcal{J}_q^d = \sum_{i=1}^{N_s} q_i \mathbf{J}_i \tag{2.21}$$

It is useful if the mixture is ionized, so charges are present in the flow field. It can be obtained by multiplying the species continuity equation by the charge of the i-th species and summing up all the equations.

The momentum conservation equation can be expressed in the convervative form as[3, 13]:

$$\frac{\partial(\rho \mathbf{V})}{\partial t} + \nabla \cdot (\rho \mathbf{V} \otimes \mathbf{V}) + \nabla p = \nabla \cdot \overline{\overline{\tau}} + \sum_{i=1}^{N_s} \rho_i \left(\mathbf{F}_{gi} + \mathbf{F}_{ei} \right) \tag{2.22}$$

or in convective form as[3, 11]:

$$\rho \frac{D\mathbf{V}}{Dt} + \nabla p = \nabla \cdot \overline{\overline{\tau}} + \sum_{i=1}^{N_s} \rho_i \left(\mathbf{F}_{gi} + \mathbf{F}_{ei} \right) \tag{2.23}$$

where p is the mixture pressure, $\overline{\overline{\tau}}$ is the viscous stress tensor, $\mathbf{F}_{gi}$ is the nonelectromagnetic body force per unit mass and $\mathbf{F}_{ei}$ is the electromagnetic force per unit mass ($\mathbf{F}_{ei} = q_i \left(\mathbf{E} + \mathbf{V}_i \times \mathbf{B} \right)$). We will assume that no external electromagnetic forces are applied. This implies that $\mathbf{E}$ and $\mathbf{B}$ are generated only by phisical phenomena happening inside the flow field. In particular, I will assume that the magnetic field is negligible, so $\mathbf{B} = 0$ and that the electric field arises from the **ambipolar diffusion constraint**, i.e. $\sum_{i=1}^{N_s} q_i \mathbf{J}_i = \mathcal{J}_c^d = 0$ [3]. Furthermore, we assume negiglibe nonelectromagnetic body forces, so $\mathbf{F}_{gi} = 0$. The radiation pressure has also been neglected.

The energy conservation equation can be expressed in the conservative form as[3, 11]:

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{V} + p \mathbf{V}) = \nabla \cdot \left(\overline{\overline{\tau}} \cdot \mathbf{V} \right) + \sum_{i=1}^{N_s} (\rho_i \mathbf{V} + \mathbf{J}_i) \cdot (\mathbf{F}_{gi} + \mathbf{F}_{ei}) - \nabla \cdot \mathbf{q} \tag{2.24}$$

where E is the total energy per unit volume:

$$E = e + \frac{1}{2} \mathbf{V} \cdot \mathbf{V} = e + \frac{1}{2} |\mathbf{V}|^2 \tag{2.25}$$

and e is the internal energy per unit volume, $\mathbf{J}_i$ is the species diffusion flux and $\mathbf{q}$ is the heat flux vector. Thanks to the previous assumpitions, the work of both body forces is zero: $\sum_{i=1}^{N_s} \mathbf{J}_i \cdot \mathbf{F}_{ei} = 0$ for the ambipolar constraint, and $\sum_{i=1}^{N_s} \rho_i \mathbf{V} \cdot \mathbf{F}_{ei} = 0$ for strict charge neutrality, and so is approximate. Radiation power has been neglected.

It is possible to derive an expression for the conservation of entropy[11]:

$$\frac{Ds}{Dt} = -\frac{1}{\rho T} \nabla \cdot \mathbf{q} + \frac{\epsilon}{T} \tag{2.26}$$

where ϵ represents the mechanic energy to heat irreversible conversion due to the viscous stresses, so $\epsilon = 0$ for an inviscid flow. The reader can find the exact formula for ϵ in the literature[11], but it is not necessary for the present work. This formula is valid for a fluid in local thermodynamic equilibrium, that is an hypotesis that I am going to make in the next sections.

The missing terms in the governing equations are given in the next sections by using: Statistical mechanics (suitably coupled with quantum mechanics) provides the thermodynamic properties (internal energy, specific heat) and kinetic theory provides the transport properties (viscosity,thermal conductivity, diffusion).

2.2.3 Thermodynamic properties of perfect gas mixtures

Each species is assumed to behave as a perfect gas: a gas where interparticle forces and particle volume are negligible. For most problems in aerodynamics, this is a reasonable assumption, since that $p < 1000\ bar$ and $T > 30\ K$[3]. Conditions that require consideration of interparticle forces are very high pressure ($p > 1000$bar) due to volume considerations, or low temperature ($T < 30K$),due to forces considerations, both of which are far from the typical conditions met in aerospace applications

The equation of state for a perfect gas may be written as:

$$p_i = \rho_i \, R_i \, T \tag{2.27}$$

where p_i is the partial pressure of the i-th species, R_i is the specific gas constant of the i-th species and T is the temperature.

This means that I am assuming that the local thermodynamic state of each species can be described using two state variables, the temperature and the pressure.

The specific gas constant is given by:

$$R_i = \frac{\mathcal{R}}{M_i} \tag{2.28}$$

where $\mathcal{R}$ is the universal gas constant and M_i is the molar mass of the i-th species.

For Dalton's law of partial pressures, the total pressure is the sum of the partial pressures:

$$p = \sum_{i=1}^{N_s} p_i \tag{2.29}$$

and the same is true for the density, as already shown:

$$\rho = \sum_{i=1}^{N_s} \rho_i \tag{2.30}$$

The thermodynamic state of a mixture of perfect gases is uniquely defined by:

- The temperature T
- The pressure p or density ρ
- The chemical composition (mass/mole fractions)

We can define the mass fractions:

$$y_i = \lim_{\Delta V \to 0} \frac{m_i}{m} = \frac{\rho_i}{\rho} \tag{2.31}$$

and the mole fractions:

$$x_i = \frac{n_i}{n} \tag{2.32}$$

Clearly:

$$\sum_{i=1}^{N_s} y_i = 1 \tag{2.33}$$

$$\sum_{i=1}^{N_s} x_i = 1 \tag{2.34}$$

It is possibile to express the mass fractions in terms of the mole fractions:

$$y_i = \frac{M_i}{M} x_i \tag{2.35}$$

where:

$$M = \sum_{i=1}^{N_s} M_i x_i \tag{2.36}$$

is the mixture molar mass.

We can write the equation of state for the mixture as:

$$p = \rho R T \tag{2.37}$$

where R is the mixture specific gas constant:

$$R = \sum_{i=1}^{N_s} y_i R_i \tag{2.38}$$

2.2.4 Energy, enthalpy and specific heat

Quantum physics shows that atoms and molecules have different modes to store energy and that each mode is quantized, i.e. it can only take discrete values. In an atom there are 2 energy modes:

- Translational energy mode: associated with the motion of the center of mass, e_{tr}.
- Electronic energy mode: associated with the electrons orbiting around the nucleus, e_{el}.

For a molecule, there are additional energy modes:

- Rotational energy mode: associated with the rotation of the molecule around three orthogonal axes in space, e_{rot}.
- Vibrational energy mode: associated with the vibration of the atoms of the molecule with respect to equilibrium positions within the molecule, e_{vib}.

When the temperature is absolute zero ($0K$) each energy mode has a minimum energy content(so called *zero-point energy*)[4], zero in case of rotation, but different from zero for other modes. This is also called *ground state*. Let ϵ_i be the difference between the energy of level i and the zero point energy ϵ_0.

Every energy mode can assume an ensemble of, in theory infinite, discrete values, or levels. Each level, in its turn, may manifest itself in a number of different ways, which result in degeracy (g_i) of the levels (for example, an electron can spin clockwise or counter-clockwise). Consider now a system of N particles distributed among an ensemble of i energy levels, each of them with a different energy content, ϵ_i. The total energy of the system is[3]:

$$E = \sum_{i=0}^{\infty} \epsilon_i N_i \tag{2.39}$$

with the constrain $sum_{i=0}^{\infty} N_i = N$. Every distinguishable distribution of the particles over the energy levels is called a **microstate**.

A way of computation the possible macrostates is to set up a differential equation for each level, an exceedingly complex task:
The sensible internal energy of the system can thus be computed if the distribution N_i is known. However, there are many possible distributions N_i, that can satisfy the above equations. Each one is called a macrostate of the system, or discernible distribution of particles. As time passes, the macrostate of the system varies as particles change levels because of collisions between themselves. The value of all N, thus varies in time, although the equations are always satisfied. Ultimately, the equilibrium state will be reached and the system will not seem to evolve any more - although collisions will continue to produce level changes for individual particles. Statistical thermodynamics postulates that this equilibrium macrostate will be the most probable of all possible macrostates

To avoid this task, one can use statistical mechanics in order to find the **most probable macrostate**, i.e. the macrostate with an overwhelmingly higher probability than the others[3].In particular, this macrostate occours when the system is in thermodynamic equilibrium (this restricts the use of this macrostate to conditions of thermodynamic equilibrium or of slight nonequilibrium). Statistical thermodynamies makes the hypothesis that all microstates are equiprobable. Thus the most probable macrostate will be that with the greatest number of associated microstates.

In the cases where local thermal equilibrium cannot be assumed, multi-temperature models can be used (see works from Park and Lee[13]).

The distribution of the particles over the different level for the most probable macrostate is given by Bolzmann distribution:

$$N_i = N\,\frac{g_i e^{-\frac{\epsilon_i}{kT}}}{Q} \tag{2.40}$$

with k the Boltzmann constant and Q the system partition function:

$$Q = \sum_{i=0}^{\infty} g_i e^{-\frac{\epsilon_i}{kT}} \tag{2.41}$$

The reader should notice that this is true if we assume that the populations of internal energy levels satisfy the Maxwell-Bolzmann statistics, such that the Quantum effects differentiating boson (Bose-Einstein statistics) and fermions (Fermi-Dirac statistics) are negligible[13].

For atoms, the partition function has the form:

$$Q_A = \sum_{n=1}^{\infty}\sum_{e=0}^{\infty} g_{tr,n} g_{el,e} e^{\frac{-\epsilon_{tr,n}-\epsilon_{el,e}}{kT}} \tag{2.42}$$

For molecules, the partition function has the form:

$$Q_M = \sum_{n=1}^{\infty}\sum_{J=0}^{\infty}\sum_{v=0}^{\infty}\sum_{e=0}^{\infty} g_{tr,n} g_{rot,J} g_{vib,v} g_{el,e} e^{\frac{-\epsilon_{tr,n}-\epsilon_{rot,J}-\epsilon_{vib,v}-\epsilon_{el,e}}{kT}} \tag{2.43}$$

The partition function of the system can be hard to find: for perfect gases (weakly interacting particles, diluite gases[13]), the translational and the internal modes are independent of each others, so the partition function of the system is the product of the partition functions of the single modes: $Q = Q_{tr}Q_{int}$, where Q_{int} belongs to the internal modes. While translational energy levels are discrete, the spacing between levels is extremely small. For all practical purposes, this permits a semi-classical approach in which translational energy is assumed continuous while internal energy is left discrete[13]. In a real molecule, the internal energy modes are not truly independent of each other, so when computing Q_{int} the contribution of a single mode cannot be factored separately from the others. However, for the simplest diatomic molecule model, which is the **Rigid Rotator-Harmonic-Oscillator (RRHO)**, all the energy modes are considere to be independent, and the partition function of the system is the product of the partition functions of the single modes: $Q = Q_{tr} Q_{rot} Q_{vib} Q_{el}$.

Under this hypothesis, the internal energy of an atom can be written as[3]:

$$e_i = e_{tr,i} + e_{el,i} + e_{0,i} \tag{2.44}$$

while the internal energy of a molecule can be written as:

$$e_i = e_{tr,i} + e_{rot,i} + e_{vib,i} + e_{el,i} + e_{0,i} \tag{2.45}$$

where $e_{0,i}$ is the zero-point energy of the species i-th.

Using the partition function, it is possible to find an expression for each energy modes of the system.

The translational energy is the same for both atoms and molecules and its value per unit mass is:

$$e_{tr,i} = \frac{3}{2} R_i T \tag{2.46}$$

The electronic energy for atoms (and for molecules when it can be factored) is, per unit mass:

$$e_{el,i} = R_i \frac{\sum_{k=0}^{\infty} g_k \theta_{E,k} e^{-\frac{\theta_{E,k}}{T}}}{\sum_{k=0}^{\infty} g_k e^{-\frac{\theta_{E,k}}{T}}} \tag{2.47}$$

where $\theta_{E,k}$ is the characteristic electronic temperature for level k. The series in Eq. 2.47 diverges and has to be truncated. Different authors propose different truncation methods, for example:

- In Barbante P. thesis[3], the criteria is to take into account the strictly necessary minimum number of electronic levels that produce a non-negiglibe change of energy in the temperature $T < 20'000K$.
- In Bottin B. thesis[4], the author explains different cut-off criteria, like using valence states, geometrical criterion, advanced ionisation potential, etc. After a comparison, the author chooses to take into account the strict necessary number of levels and to avoid using levels which do not produce a significant change in enthalpy.

For a diatomic molecule behaving as a RRHO, the rotational energy per unit mass can be written as:

$$e_{rot,i} = R_i T \left(1 - \frac{\theta_R}{\theta_R + 3T} \right) \tag{2.48}$$

where θ_R is the characteristic rotational temperature and is usually equal to a few Kelvin, so its contribution is practically negligible.

The vibrational energy per unit mass can be written as:

$$e_{vib,i} = R_i \frac{\theta_V}{e^{\frac{\theta_V}{T}} - 1} \tag{2.49}$$

where θ_V is the characteristic vibrational temperature.

After computing the energy, the enthalpy can be simply computed as:

$$h_i = e_i + \frac{p_i}{\rho_i} = e_i + R_i T \tag{2.50}$$

The zero-point energy $e_{0,i}$ generally cannot be measured or computed, and it is generally set to the heat of formation of the species.

$$e_{0,i} = \Delta H^0_{F,i} \tag{2.51}$$

The reader should notice that the RRHO (Rigid Rotator-Harmonic Oscillator) model does not truly rapresent the reality, and so-called anharmonicity corrections can be used[3]. However, the coupled effect of the energy model is more pronounced above 6000 K, where usually the molecules are highly dissociated, so the correction has a small effect on the mixture properties and can often be neglected, as I will do.

It is interesting to notice that the pressure of the mixture reads:

$$pV = nRT \tag{2.52}$$

which strongly supports per se the validity of the statistical theory[4].

The mixture energy and enthalpy can be computed as the sum of the energy and enthalpy of the single species:

$$e = \sum_{i=1}^{N_s} y_i e_i \tag{2.53}$$

$$h = \sum_{i=1}^{N_s} y_i h_i \tag{2.54}$$

The single species heats are, by definition:

$$C_{v,i} = \left(\frac{\partial e_i}{\partial T}\right)_v \tag{2.55}$$

$$C_{p,i} = \left(\frac{\partial h_i}{\partial T}\right)_p \tag{2.56}$$

The mixture specific heat at constant pressure reads:

$$C_p = \left(\frac{\partial h}{\partial T}\right)_p = \sum_{i=1}^{N_s} \left\{ \left(\frac{\partial y_i}{\partial T}\right)_p h_i + y_i \left(\frac{\partial h_i}{\partial T}\right)_p \right\} \tag{2.57}$$

If no chemical reactions are taking place in the flow (the mixture is frozen) the first derivative is identically zero, and we have the frozen specific heat:

$$C_{p,fr} = \sum_{i=1}^{N_s} y_i \left(\frac{\partial h_i}{\partial T}\right)_p = \sum_{i=1}^{N_s} y_i C_{p,i} \tag{2.58}$$

In case of a frozen mixture, chemical composition does not change (yi is constant) and $C_{p,fr}$ is function only of temperature: a frozen mixture is therefore a thermally perfect gas.

If, on the opposite, chemical equilibrium is established, the chemical composition is function only of 2 thermodynamic variables, e.g. pressure and temperature: $y_i = y_i(p, T)$, and the specific heat is:

$$C_{p,eq} = C_{p,fr} + \sum_{i=1}^{N_s} \left(\frac{\partial y_i}{\partial T}\right)_p h_i \tag{2.59}$$

the latter term is also called *reactive specific heat.*

In the intermediate case of a finite rate chemically reacting mixture, the chemical composition is function not only of two thermodynamic variables, but also of the position and of the previous history.

All the thermodynamic properties of the mixture will be computed using the *Mutation++* library[13], which follows the hypotheses of the model that I am using. In particular, with the *Mutation++* library, the user can choose between:

- RRHO: Rigid Rotator Harmonic Oscillator model, exact computations are made.
- NASA-7: 7-coefficient polynomial model. Curve fitting of the thermodynamic data is made.
- NASA-9: 9-coefficient polynomial model. Curve fitting of the thermodynamic data is made.

The thermodynamic properties can be derived from the NASA polynomial by using the following formulas[13]:

$$C_{pi} = R_i \sum_{j=0}^{4+q} a_{ij} T^{j-q} \tag{2.60}$$

$$h_i = \int C_{pi} dT + b_{i1}\, R_i\, T \tag{2.61}$$

$$s_i = \int \frac{C_{pi}}{T} dT + b_{i2}\, R_i \tag{2.62}$$

2.2.5 Entropy

Similarly to the energy and enthalpy, the entropy can be computed as the sum of the entropies of the single species:

$$s = \sum_{i=1}^{N_s} y_i s_i \tag{2.63}$$

The entropy of the single species are computed using the statistical mechanics and the partition function of the system. I will not go into the details since that the procedure is the same as the one used for the energy and enthalpy.

2.2.6 Local Thermal equilibrium

As mentioned before, an important hypothesis that I need to make is that the flow is in **local thermal equilibrium**, i.e. the translation, rotation and vibration temperature of the molecules are the same.

In fact, in the hypersonic regime, after a shock wave, the translational and rotational modes have a low relaxation time, i.e. the time that the energy modes take to readjust is low, since that the readjustment requires only a few collisions, so these 2 modes have usually the same temperature. In contrast, vibrational and chemical processes take much longer to readjust, so extendend regions in the flow field may occur where the vibrational and chemical nonequilibrium exist. In general vibrational motion readjusts earlier than the chemical composition, although both are intricately coupled during the nonequilibrium stage. Successive relation of the different degrees of freedom motivetes the definition of alternative kinetic translation temperatures, which may be different from the translational temperature[9].

In my work, I will assume that the flow is in local thermal equilibrium, but the reader should notice that this cannot not always be assumed, and in those cases, multi-temperature models like the *Park two-temperature model* can be used[9].

The main idea of these models is that the energy associated with any internal level can be separated into various energy modes. This brings to the definition of different temperatures for the different energy modes. There exist other models called *State-Specific Models* which attempt to increase the validity range of MT models beyond small departures from equilibrium, by modelling some states as pseudo-species[13]. The most rigorous energy partitioning models are the so-called STS or collisional models which treat all internal energy levels as pseudo-species[13].

2.2.7 Chemistry model

In order to give an expression for the chemical production term $\dot{\omega}_i$, let's start by considering an elementary reaction (identified with index r), i.e. a reaction accomplished in one step only, which can be formally written as::

$$\sum_{i=1}^{N_s} \nu_{ir}^{'} X_i \rightleftarrows \sum_{i=1}^{N_s} \nu_{ir}^{''} X_i \tag{2.64}$$

The species appearing on the left hand side are the reactants, and the one appearing on the right hand side are the products. X_i is a dummy symbol for the i-th species, $\nu_{ir}^{'}$ and $\nu_{ir}^{''}$ are the stoichiometric coefficients of the i-th reactant and product, respectively. Notice that an elementary reaction can proceed in both directions and is always reversible; when there is perfect balance between dissociation and recombination, the reaction is said to be in **chemical equilibrium**.

A typical example is the dissociation recombination of oxygen, which is an important reaction in the context of hypersonic flows:

$$O_2 + X \rightleftarrows O + O + X \tag{2.65}$$

MOlecular oxygen O_2 collides with a third body X and dissociates into two oxygen atoms O if the energy of the collision in enough to activate the reaction. In the reverse reaction two oxygen atoms collide with the third body and recombine into molecular oxygen if the third body can carry out the energy released in the recombination. The third body does not change chemical nature in the reaction.

In accordance with the law of mass action and experimental evidence, the new rate of production of species i by the r-th elementary reaction is given by[3]:

$$\dot{\omega}_{ir} = M_i \left\{ \left(\nu_{ir}^{''} - \nu_{ir}^{'}\right) k_{fr} \prod_{j=1}^{N_s} \left(\frac{\rho_j}{M_j}\right)^{\nu_{jr}^{'}} - \left(\nu_{ir}^{''} - \nu_{ir}^{'}\right) k_{br} \prod_{j=1}^{N_s} \left(\frac{\rho_j}{M_j}\right)^{\nu_{jr}^{''}} \right\} \tag{2.66}$$

Considering N_r reactions for the i-th species, the total rate of production of the i-th species is the sum of the rates of production of the single reactions:

$$\dot{\omega}_i = \sum_{r=1}^{N_r} \dot{\omega}_{ir} \tag{2.67}$$

where M_i is the molar mass of the i-th species, k_{fr} and k_{br} are the forward and backward rate constants of the r-th elementary reaction, respectively.

We can link the forward and backward rate constants to the equilibrium constant K_{cr} of the r-th elementary reaction:

$$K_{cr} = \frac{k_{fr}}{k_{br}} \tag{2.68}$$

where the equilibrium constant is a function of temperature only for a perfect gas mixture:

$$\log K_{cr}(T) = -\sum_{i=1}^{N_s} \frac{\left(\nu_{ir}^{''} - \nu_{ir}^{'}\right) \hat{g}_i(T)}{\mathcal{R}T} - \log(\mathcal{R}T) \sum_{i=1}^{N_s} \left(\nu_{ir}^{''} - \nu_{ir}^{'}\right) \tag{2.69}$$

where $\hat{g}_i(T)$ is the Gibbs free energy per unit mole of the i-th species at temperature T and is equal to:

$$\hat{g}_i(T) = \hat{h}_i - T\hat{s}_i \tag{2.70}$$

where $\hat{h}_i$ is the specific enthalpy of the i-th species and $\hat{s}_i$ is the specific entropy of the i-th species per unit mole. The latters can be computed using the same statistical mechanics approach used for the energy modes.

It is possible to derive the forward and backward reaction coefficient from kinetic theory, assuming a known form for the interaction potential, the elementary dissociation-recombination probabilitiess and the exact distribution functions. In pratice, this approach is not possible because too much information is missing and a semiempirical formulation, the so called **Arrhenius law**, is used to compute the forward reactiong rates:

$$k_{fr} = A_r T^{b_r} e^{-\frac{E_{d,r}}{kT}} \tag{2.71}$$

where $A_r > 0$ is the pre-exponential factor, b_r is the temperature exponent and $E_{d,r}$ is the activation energy of the r-th elementary reaction. This expression is only valid for thermal equilibrium. Given the coefficients, one may compute the rate of production of the i-th species due to the r-th elementary reaction.

2.2.8 Local Chemical equilibrium

Let's now recall the species i continuity equation (Eq. 2.13) but without the moment, the convective and diffusive terms for now. The equation reduces to the ordinary differential equatio (ODE):

$$\frac{\partial \rho_i}{\partial t} = \dot{\omega}_i \tag{2.72}$$

If we consider all the species, we have a system of N_s ODEs, which is generally nonlinear because of the structure of the production term.

In first approximation, we can linearize the system around a known state ρ_i^0:

$$\frac{\partial}{\partial t}\{\rho_i\} = \{\dot{\omega}_i^0\} + [A]\{\rho_i\} \tag{2.73}$$

where $[A]$ is the Jacobian matrix of the system, evaluated at the known state ρ_i^0.

Let's observe that $\dot{\omega}_i$ has dimensions $\frac{Kg}{m^3 s}$, and thus each element of the Jacobian $[A]$ has dimensions $\frac{1}{s}$, and thus is an index of the characteristic time of chemical reactions. The element $[A]_{ij} = \frac{\partial \dot{\omega}_i}{\partial \rho_j}$ may also be seen as the sensitivity, due to chemical reactions, of the species i with respect to a variation of species j.

The time evolution of the N_s species is essentially given by[3]:

$$\{\rho_i\} \approx e^{[A]t}\{\rho_i^0\} \tag{2.74}$$

where $e^{[A]t}$ is the matrix exponential of $[A]t$. For practical purposes, we can think of having only a global value for the characteristic time of the chemical reactions, instead of the N_s by N_s given by the Jacobian matrix. The norm of $[A]$, $||[A]||$, is taken as the desired value.

The next step is to compare a characteristic flow time (e.g. the time that the flow needs to cross the region of interest) with the characteristic time of the chemical reactions. In order to do so, we can dimensionalise the species continuity equation (Eq. 2.13): we define a reference length L_∞, a reference speed V_∞, a reference density ρ_∞, and a reference chemistry time $\frac{1}{\tau_c} = ||[A]||$. The flow reference time is defined as $\tau_f = \frac{L_\infty}{V_\infty}$. The nondimensional variables are defined as:

$$\begin{cases} \tilde{\rho}_i = \frac{\rho_i}{\rho_\infty} \\ \tilde{t} = \frac{t}{\tau_f} \\ \tilde{\mathbf{V}}_i = \frac{\mathbf{V}_i}{V_\infty} \\ \tilde{\dot{\omega}}_i = \frac{\dot{\omega}_i}{\rho_\infty}\tau_c \\ \tilde{\nabla} = \frac{\nabla}{L_\infty} \end{cases} \tag{2.75}$$

Substituting the new variables, the species equation in nondimensional form read:

$$\frac{\partial \tilde{\rho}_i}{\partial \tilde{t}} + \tilde{\nabla} \cdot \left(\tilde{\rho}_i \tilde{\mathbf{V}}_i\right) = \tilde{\dot{\omega}}_i \frac{\tau_f}{\tau_c} \tag{2.76}$$

We can now define the **First Damköhler number** as the ratio of the flow time to the chemistry time:

$$Da_1 = \frac{\tau_f}{\tau_c} \tag{2.77}$$

By analyzing Eq. 2.76, we can see define the two following limiting cases:

- If $\tau_f \ll \tau_c$, or $Da_1 \to 0$, the flow time is much smaller than the chemistry time, so the chemical reactions are negiglibe, the fow is called **fronzen flow** and the species continuity equation becomes:

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{V}_i) = 0 \tag{2.78}$$

- $\tau_f \gg \tau_c$, or $Da_1 \to \infty$, the flow time is much greater than the chemistry time, so the flow tends towards a state of local chemical equilibrium, since that the reactions have enough time to reach equilibrium. The species continuity equation becomes:

$$\dot{\omega}_i = 0 \tag{2.79}$$

The hypothesis that I will make is that the flow is in local chemical equilibrium, i.e. $Da_1 \to \infty$. In this case, the chemical composition of the flow i uniquely determined by the local values of p and T or ρ and T. The species continuity equations may be eliminated from the system of governing equations and the chemical composition computed with an ad hoc algorithm (e.g. using the *Mutation++* library[13]).

It is important to notice that the local chemical equilibrium do not depend on the chemical reactions considered, but rather on the thermodynamics variables, i.e. pressure and temperature. Different methods can be used to compute the chemical composition, but the general approaches are:

- **Method 1: equilibrium constant formulation**: Given n_{sp} different chemical species, one needs n_{sp} independent equations to compute the chemical composition. Of these, n_c are obtained by writing the conservation of nuclei or charge in the mixture. The remaining $n_{sp} - n_c = n_r$ equations are reaction equations, describing possible set of reactions between the species. Since the gas is in equilibrium, they do not need to correspond to real reactions taking place in the mixture. Rather, they are used as a mere set used to solve the problem. Indeed, it seems logical that the rated at which the reactions take place have no impact on the equilibrium composition, which remain fixed at time[4]. So, artificial reactions will do just as well as real ones. Furthermore, real chemistry processes are described by more equations than needed, some of them obviously linearly dependent. The system of equations can be solved.

- **Method 2: free-energy minimization**: This method is not based on chemical reactions, but on the minimization of the Gibbs free energy of the mixture under proper constraints. Methods like Lagrange multipliers can be used. In particular, *Mutation++* uses the *Multiphase Gibbs function continuation (MPGFC) method* [13].

In PlasFlowSolver, the equilibrium chemical composition will be always computed by using the *Mutation++* library. Under these hypotheses, we can express the mixture equilibrium constant volume and pressure specific heat as:

$$C_{v,eq} = \left(\frac{\partial e}{\partial T}\right)_p = \sum_{i=1}^{N_s} \left[\left(\frac{\partial y_i}{\partial T}\right)_p e_i + y_i C_{v,i} \right] \tag{2.80}$$

where $C_{v,i}$ is the specific heat at constant volume of the i-th species.

$$C_{p,eq} = \left(\frac{\partial h}{\partial T}\right)_p = \sum_{i=1}^{N_s} \left[\left(\frac{\partial y_i}{\partial T}\right)_p h_i + y_i C_{p,i} \right] \tag{2.81}$$

where $C_{p,i}$ is the specific heat at constant pressure of the i-th species. And the equilibrium sound speed as:

$$c_{eq}^2 = \left(\frac{\partial p}{\partial \rho}\right)_s = \frac{C_{p,eq}}{C_{v,eq}} \frac{p}{\rho} \frac{1 - \frac{\rho^2}{p}\left(\frac{\partial e}{\partial \rho}\right)_T}{1 - \rho\left(\frac{\partial h}{\partial p}\right)_T} \tag{2.82}$$

The reader should notice that the local chemical equilibrium is not always true, and in those cases, other models can be used to compute the chemical composition.

2.2.9 Local Thermodynamic equilibrium

A flow is in **Local Thermodynamic Equilibrium** when it is in:

- **Local thermal equilibrium**
- **Local chemical equilibrium**
- **Local mechanical equilibrium**

I already went through the first two points, so I will now explain the third one.

A flow is in **Local mechanical equilibrium** if locally no net forces are acting on the fluid.

Since that a flow is moving, we cannot really have a state of local mechanical equilibrium, but we can have a state of **quasi-local mechanical equilibrium**, so that we can apply classical thermodynamic to small fluid volumes, the fluid particles. A fluid particle is a small deforming volume carried by the flow such that:

- It always contains the same fluid molecules
- It is large enough so that its thermodynamic properties are well defied when it is at equilibrium
- It is small enough so that its relaxation time (time taken by the system to adjust its thermodynamic state) is short compared to the time scales of the fluid-motion-induced thermodynamic changes.

This principle goes along with the continuum hypothesis, which states that the fluid is continuous and that the fluid properties are well defined at every point in the flow field.

In this way, it is possible to define a thermodynamic pressure p, that is the one that we use in the equation of state, since that it is well-defined only in thermodynamic equilibrium conditions.

The reader should notice that there are certain circumstances involving rarified gases, shock waves and high-frequency acousitc waves where one or more of the fluid particle requirements are not met and molecular-kinetic and quantum therories are needed.

2.2.10 Transport fluxes

In this subsection I will give the expression for the transport fluxes that apper in the governing equations. In particular, the diffusion flux $\mathbf{J}_i$, the stress tensor $\overline{\overline{\tau}}$ and the heat flux $\mathbf{q}$ are computed by the kinetic theory of gases and statistical mechanics[3].

Molecular description

The starting point of the thoery is the famous **Boltzmann equation** which describes a mixture from the molecular point of view. In particular, we seek a multiscale Chapman-Enskog perturbative solution of the Boltzmann equation.

The exact representation of the mixture state in not only impossible because it requires the knowledge of velocity, position and internal state of every particle in the mixture, but is also redundant for our continuum description that only needs the knowledge of some suitably defined average quantities. It seems therefoe more practical and convenient to use a statistical approach that, by its own nature, gives the *global* behaviour of the system under investigation.

Consider a particle belonging to species i, for simplicity we assume it has no internal degrees of freedom: its state is completely characterized by its position $\mathbf{r}$ and its velocity $\mathbf{c}_i$. The six-dimensional space having as components the three components of $\mathbf{r}$ and the three components of $\mathbf{c}_i$ is called phase space. In the spirit of the continuum decription, it would be enough to have a function $f_i(\mathbf{r}, \mathbf{c}_i, t)$ that gives the exprected amount of i species particles in an elementary volume $d\mathbf{r}d\mathbf{c}_i$ of the phase space. In other word, $N = f_i(\mathbf{r}, \mathbf{c}_i, t)d\mathbf{r}d\mathbf{c}_i$ is the expected number of i species particles in the volume element $d\mathbf{r}$ located at $\mathbf{r}$, whose velocities lies in the interval $d\mathbf{c}_i$ about velocity $\mathbf{c}_i$ at time t. Integration with respect to $\mathbf{r}$ and $\mathbf{c}_i$ gives the total number of i species particles in the system. Integration with respect to $\mathbf{c}_i$ gives the total number of i species particles in the volume element $d\mathbf{r}$ and the number density n_i of i species is this number divided by $d\mathbf{r}$:

$$n_i(\mathbf{r}, t) = \int f_i(\mathbf{r}, \mathbf{c}_i, t)d\mathbf{c}_i \tag{2.83}$$

The partial density ρ_i is $\rho_i = m_i n_i$, where m_i is the mass of the single i-th species particle. If $\varphi_i(\mathbf{r}, \mathbf{c}_i, t)$ is a generic property for species i function of the particle velocity, its average value is:

$$\bar{\varphi}_i(\mathbf{r}, t) = \frac{1}{n_i(\mathbf{r}, t)} \int \varphi_i(\mathbf{r}, \mathbf{c}_i, t) f_i(\mathbf{r}, \mathbf{c}_i, t)d\mathbf{c}_i \tag{2.84}$$

For example, the i species average velocity is:

$$\mathbf{V}_i(\mathbf{r}, t) = \frac{1}{n_i(\mathbf{r}, t)} \int \mathbf{c}_i f_i(\mathbf{r}, \mathbf{c}_i, t)d\mathbf{c}_i \tag{2.85}$$

The mixture mass average velocity is defined as:

$$\mathbf{V}(\mathbf{r}, t) = \sum_{i=1}^{N_s} m_i n_i(\mathbf{r}, t)\mathbf{V}_i(\mathbf{r}, t) \tag{2.86}$$

The difference between the i species particle velocity and the mixture average velocity is the **peculiar velocity** of species i:

$$\mathbf{C}_i = \mathbf{c}_i - \mathbf{V} \tag{2.87}$$

The peculiar velocity allows us to compute the diffusion velocity:

$$\mathcal{V}_i = \frac{1}{n_i(\mathbf{r},t)} \int \mathbf{C}_i f_i(\mathbf{r},\mathbf{c}_i,t) d\mathbf{c}_i \tag{2.88}$$

The peculiar velocity is linked with the thermal motion of the molecules: in a mixture at rest, without macroscopic gradients, particles are still suject to Brownian motion and this motion is nthing else than the peculiar velocity. The average kinetic energy assocated with the peculiar velocity may be identified as the translational component of the internal energy, therefore we have the equality:

$$\frac{3}{2}RT(\mathbf{r},t) = \frac{1}{\rho(\mathbf{r},t)} \sum_{i=1}^{N_s} \frac{1}{2} \int m_i C_i^2 f_i(\mathbf{r},\mathbf{c}_i,t) d\mathbf{c}_i \tag{2.89}$$

Transport fluxes definition

In a gas under nonequilibrium conditions (globally), gradients exist in one or more of the macroscopic physical properties of the system. The gradients of these properties result in the molecular transport of mass, momentum and energy through the mixture. The flux vector associated with the transport of the generic property ϕ_i is:

$$\mathbf{\Phi}_i(\mathbf{r},t) = \int \phi_i(\mathbf{r},\mathbf{c}_i,t)\mathbf{C}_i f_i(\mathbf{r},\mathbf{c}_i,t) d\mathbf{c}_i \tag{2.90}$$

Let's note that the velocity with which ϕ is transported is the peculiar velocity $\mathbf{C}_i$ of the i-th species particles.

The transport of mass is obtained by setting $\phi_i = m_i$ in Eq. 2.90 and gives the diffusion flux $\mathbf{J}_i$:

$$\mathbf{J}_i(\mathbf{r},t) = \int m_i \mathbf{C}_i f_i(\mathbf{r},\mathbf{c}_i,t) d\mathbf{c}_i \tag{2.91}$$

The transport of momentum is obtained by setting $\phi_i = m_i\mathbf{C}_i$ in Eq. 2.90 and gives the stress tensor $mathcalP$:

$$\mathcal{P}(\mathbf{r},t) = \int m_i \mathbf{C}_i \otimes \mathbf{C}_i f_i(\mathbf{r},\mathbf{c}_i,t) d\mathbf{c}_i = \rho_i \overline{\mathbf{C}_i \otimes \mathbf{C}_i} \tag{2.92}$$

This is a symmetric tensor and the diagonal elements are normal stresses and the off-diagonal elements are shear stresses. The arithmetic mean value of the tensor trace is the thermodynamic pressure p[3] of the i species. The pressure tensor for the whole mixture is the sum of the species pressure tensors.

The transport of the translational component of the internal energy is obtained by setting $\phi_i = \frac{1}{2}m_i C_i^2$ in Eq. 2.90 and gives the heat flux $\mathbf{q}$:

$$\mathbf{q}(\mathbf{r},t) = \int \frac{1}{2} m_i C_i^2 \mathbf{C}_i f_i(\mathbf{r},\mathbf{c}_i,t) d\mathbf{c}_i = \frac{1}{2}\rho_i \overline{C_i^2 \mathbf{C}_i} \tag{2.93}$$

The sum of the heat fluxes of the single species is the heat flux of the mixture.

Particles with internal degrees of freedom

The previous discussion is valid for particles without internal degrees of freedom. The simplest way of extending it is to treat classically the translational degrees of freedom and quantum mechanically the internal ones. We redefine the distribution function by introducting the internal quantum state I (each species having Q_i internal states). $f_i(\mathbf{r},\mathbf{c}_i,t,I)$ is now the expected number of molecules of type i in quantum state I in the volume element $d\mathbf{r}$ located at $\mathbf{r}$, whose velocity lies in the interval $d\mathbf{c}_i$ about $\mathbf{c}_i$ at time t. The number density of species i becomes:

$$n_i(\mathbf{r},t) = \sum_{I=1}^{Q_I} \int f_i(\mathbf{r},\mathbf{c}_i,t,I) d\mathbf{c}_i \tag{2.94}$$

The difference with respect to Eq. 2.83 is that now we have to sum over the internal states. THe same is vald for the others quantity previously defined.

A little bit more care has to be used for the species energy, which now includes also the contribution of the internal states. Identifying with E_I^i the energy linked with the I internal state (which, in general, is independent from the particle velocity $\mathbf{c}_i$) of i species, we have for the mixture total energy:

$$e_{tr}(\mathbf{r},t) + e_{int}(\mathbf{r},t) = \frac{1}{\rho(\mathbf{r},t)} \sum_{i=1}^{N_s} \sum_{I=1}^{Q_i} \int \left(\frac{1}{2} m_i C_i^2 + E_I^i\right) f_i(\mathbf{r},\mathbf{c}_i,t,I) d\mathbf{c}_i \tag{2.95}$$

The transport fluxes are obtained by summing over the interal states too and the formulas are similar to the one for a monoatomic mixture. The only slight modification is for the heat flux vector that reads:

$$\mathbf{q}_i(\mathbf{r}, t) = \sum_{I=1}^{Q_I} \int \left(\frac{1}{2} m_i C_i^2 + E_I^i \right) \mathbf{C}_i f_i(\mathbf{r}, \mathbf{c}_i, t, I) d\mathbf{c}_i \tag{2.96}$$

The transport of the internal states energy is also taken into account.

Boltzmann equation

The governing equation for the distribution function is the well know **Boltzmann equation**[3]:

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla_{\mathbf{r}} f_i + \mathbf{F}_i \cdot \nabla_{\mathbf{c}_i} f_i = \sum_{j=1}^{N_s} J(f_i f_j) \tag{2.97}$$

In compact notation it can be written as:

$$\mathcal{D}_i(f_i) = \mathcal{C}_i(f_i) \tag{2.98}$$

The left hand side is the *streaming operator* and gives the change of the distribution function due to convection and effect of the body force $\mathbf{F}_i$;th e right hand side is the *collision operator* and gives the change in the distribution function due to the collision processes happening into the flow.

Let $\varphi(\mathbf{r}, \mathbf{c}_i, t)$ be a generic property defined for every mixture component i. Sinche the collisional operator gives the change in the distribution function f_i due to collisions, the quantity:

$$\left(\overline{\frac{\partial \varphi_i}{\partial t}} \right)_{\text{coll}} = \frac{1}{n_i} \sum_{j=1}^{N_s} \int \varphi_i(\mathbf{r}, \mathbf{c}_i, t) J(f_i f_j) d\mathbf{c}_i \tag{2.99}$$

can be interpreted as the average rate of change of φ_i due to collisions. The rate of change of the mixture property $\varphi = \frac{1}{n} \sum_{i=1}^{N_s} n_i \varphi_i$ is given by:

$$\left(\overline{\frac{\partial \varphi}{\partial t}} \right)_{\text{coll}} = \frac{1}{n} \sum_{i=1}^{N_s} n_i \left(\overline{\frac{\partial \varphi_i}{\partial t}} \right)_{\text{coll}} = \frac{1}{n} \sum_{i,j=1}^{N_s} \int \varphi_i(\mathbf{r}, \mathbf{c}_i, t) J(f_i f_j) d\mathbf{c}_i \tag{2.100}$$

The total rate of change of φ_i, i.e. the rate of change due to both the streaming operator and the collisional operator, is obtained multiplying the Boltzmann equation (Eq. 2.97) by φ_i and integrating over $\mathbf{c}_i$. Doing so we obtain the **equation of change of the property** φ_i:

$$\int \varphi_i \mathcal{D}_i(f_i) d\mathbf{c}_i = \int \varphi_i \mathcal{C}_i(f_i) d\mathbf{c}_i = \left(\overline{\frac{\partial \varphi_i}{\partial t}} \right)_{\text{coll}} \tag{2.101}$$

The equation of change for the mixture property φ is obtained summing the equations of change for the single species properties.

It is interesting to note that the mixture governing equations can be obtained from the equations of change:

- If $\varphi_i = m_i$, the equation of change gives the species continuity equation (Eq. 2.13).
- If $\varphi_i = m_i \mathbf{c}_i$, and we sum over all the spcies, the equation of change gives the momentum equation (Eq. 2.22). Let's note that the change in momentum due to the collision operator is zero because of the principle of conservation of momentum during collisions.
- If $\varphi_i = \frac{1}{2} m_i c_i^2 + \sum_{I=1}^{Q_I} E_I^i$, and we sum over all the species, the equation of change gives the energy equation (Eq. 2.24). Let's note that also in this case the change in energy due to the collision operator is zero because of the principle of conservation of energy during collisions.

A comparison among the mixture equations just obtained and the one previously presented allows us to do the following identifications:

- The diffusion flux (Eq. 2.14) defined for the species continuity equation (Eq. 2.13) is given by Eq. 2.91.
- The stress tensor $\mathcal{P}$ defined for the momentum equation (Eq. 2.22) and energy equation (Eq. 2.24) is given by the sum of the species momentum fluxes $\mathcal{P}_i$ by Eq. 2.92.
- The heat flux $\mathbf{q}$ defined in the energy equation (Eq. 2.24) is given by the sum of the species heat fluxes $\mathbf{q}_i$ by Eq. 2.93.

The reader should notice that the transport fluxes are function, among other variables, of the distribution function f_i. If the distribution function, in its turn, has a one to one correspondence with the macroscopic variables characterizing the mixture, the governing equations become self-contained and can be solver with a suitable method.

Chapman-Enskog solution

In order to solve the Boltzmann equation and find the distribution function f_i, we can use the **Chapman-Enskog solution**. This solution gives the transport fluxes as linear functions of the macroscopic variables gradients through some proportionality scalar quantities, the transport coefficients.

The distribution function f_i is developed in a series expansion with respect to a small perturbation parameter, which is assumed to be the Knudsen number Kn. The procedure restricts the validity of the results to cases of small Knudsen number, but as I said before, the continuum hypothesis is valid for $Kn \ll 1$, so the Chapman-Enskog solution is a good approximation in this case. Stopping the expansion to the first two terms we have:

$$f_i = f_i^{(0)} + Kn\, f_i^{(1)} = f_i^{(0)}\,(1 + \phi_i) \tag{2.102}$$

The approximate solution of the Boltzmann equation is obtained by inserting the series expansion into Eq. 2.97 and equating the coefficients of equal powers of Kn[3]. The expression for $f_i^{(0)}$ is:

$$f_i^{(0)} = \frac{n_i}{Q_{int}} \left(\frac{m_i}{2\pi kT}\right)^{\frac{3}{2}} e^{-\frac{m_i C_i^2}{2kT} - \frac{E_I^i}{kT}} \tag{2.103}$$

where Q_{int} is the partition function of the internal modes. In case of particles without internal degrees of freedom, the partition function is $Q_{int} = 1$ and $E_I^i = 0$. This expression for the distribution function is called the **Maxwell-Boltzmann distribution** and is assumed by the gas in case of thermodynamic equilibrium. In fact, if the Maxwell-Boltzmann distribution is used in the equations of change Eq. 2.101, they reduce to the **Euler euqations**, which are characterized by the absence of the transport fluxes, and the stress tensor is equal to the thermodynamic pressure p_i.

The perturbation term ϕ_i is the solution of a linear integro-differential equation and its most general expression is[3]:

$$\phi_i = \frac{1}{n}\left[\sum_{j=1}^{N_s} D_i^j \mathbf{C}_i \cdot \mathbf{d}_j + A_i \mathbf{C}_i \cdot \nabla \log T + B_i \left(\mathbf{C}_i \otimes \mathbf{C}_i - \frac{1}{3} C_i^2\right) : \nabla \mathbf{V} + F_i \nabla \cdot \mathbf{V}\right] \tag{2.104}$$

where $\mathbf{d}_i$ is a vector of driving forces, the scalars D_i^j, A_i, B_i and F_i are functions of the species i peculiar velocity and they are determined from the integro-differential equation satisfied by ϕ_i, and the colon operator is the double dot product. We notice that in a mixture of N_s components there are N_s^2 D_i^j coefficients, and N_s A_i, B_i and F_i coefficients. The perturbation term ϕ_i is small with respect to $f_i^{(0)}$ and, therefore, the following discussion is strictly valid only for a weak deviation from thermal equilibrium.

If the approximate solution f_i is inserted into the equations of change Eq. 2.101, they reduce to the Navier-Stokes equations. The transport fluxes are obtained by using the approximate value of the distribution function.

A closed form expression for the coefficients D_i^j, A_i, B_i and F_i does not exist, so an approximate solution is sought in the form of a finite polynomial expansion. The **Sonine polynomials**, which have some useful orthogonality properties, are used for this purpose[3]. The accuracy of the approximation depends on how many terms are kept in the polynomial expansion THe sequence is monotonically increasing and converges to the exact solution of the integro-differential equation and so the transport properties computed with the Sonine expansion tend asymptotically to the properties computed with the Chapman-Enskog solution.

Diffusion flux

Since that with the local chemical equilibrium hypotesis I have eliminated the species continuity equations, but the diffusion fluxes are still needed for the heat flux computation as we will see later. From Barbante's thesis[3]:

The diffusion flux for species i in the **Chapman-Enskog approximation** is computed by means of Eq. 2.91 and Eq. 2.102:

$$\mathbf{J}_i = -\rho_i \left(\sum_{j=1}^{N_s} D_{ij} \mathbf{d}_j + D_i^T \nabla \log T\right) \tag{2.105}$$

where $D_{ij} = \frac{1}{3n}[D^i, D^j]$, $D_i^T = \frac{1}{3n}[D^i, A]$, where the symbol $[M, N]$ is called *bracket integral.* For the ONsager principle, $D_{ij} = D_{ji}$ and $D_{ii} \neq 0$.

The vector $\mathbf{d}_i$ of driving forces is:

$$\mathbf{d}_i = \nabla x_i + \left(x_i - \frac{\rho_i}{\rho}\right) \nabla \log p - \frac{\rho_i}{\rho p}\left(\rho \mathbf{F}_i - \sum_{k=1}^{N_s} \rho_k \mathbf{F}_k\right) \tag{2.106}$$

where x_i is the mole fraction of species i and $\mathbf{F}_i$ is the body force acting on species i. We notice that diffusion is generated by gradient in chemical composition, pressure, temperature (Soret effect) and body forces. Furthermore, $\sum_{i=1}^{N_s} \mathbf{d}_i = 0$.

The multicomponent diffusion and thermal diffusion coefficients can be computed by means of the polynomial expansions of D_i^j, and A_i, which read:

$$D_{ij} = \sum_{p=0}^{L-1} d_{i,p}^{j,(L)} \, S_{\frac{3}{2}}^{p}(W_i) \tag{2.107}$$

$$A_i = \sum_{p=0}^{L} a_{i,p}^{(L)} \, S_{\frac{3}{2}}^{p}(W_i) \tag{2.108}$$

where $S_{\frac{3}{2}}^{p}(W_i)$ are the Sonine polynomials, $d_{i,p}^{j,(L)}$ and $a_{i,p}^{(L)}$ are the expansion coefficients, indexes i and j are the species indexes, p is the expansion term, L is the order of the expansion and W_i is a nonsimensional speed:

$$W_i = \sqrt{\frac{m_i}{2kT}} C_i \tag{2.109}$$

Using the porperties of the Sonine polynomials, we can find the coefficients for the diffusion flux:

$$D_{ij} = \frac{1}{2n} d_{i,0}^{j,(L)} \tag{2.110}$$

$$D_i^T = -\frac{1}{2n} a_{i,0}^{(L)} \tag{2.111}$$

The coefficients for the polynomial expansion depend of the number L of terms used in the expansion, so, taken a L, the coefficients are determined by means of a linear system of dimension $N_s L by N_s L$. N_s of such system have to be solved in otder to compute all the coefficients. D_i^T is computed from a linear system of dimension $N_s(L+1)$ by $N_s(L+1)$. The coefficients of both linear system are function of the so-called **collision integrals**, which I will explain later.

Another approach, used also in the thermodynamic library that I will use, *Mutation++*, is to compute the diffusion fluxes from the **Stefan-Maxwell equations**. The Stefan-Maxwell equations are exactly equivalent to the complete diffusion equations[3] and they are obtained by inverting Eq. 2.105:

$$\frac{M}{\rho} \sum_{j=1}^{N_s} \left(\frac{x_i \mathbf{J}_j}{M_j \mathcal{D}_{ij} f_{ij}(L)} - \frac{x_j \mathbf{J}_i}{M_i \mathcal{D}_{ij} f_{ij}(L)} \right) = \mathbf{d}_i + k_i^T \nabla \log T \tag{2.112}$$

where $\mathcal{D}_{ij}$ are the binary diffusion coefficients, and they are symmetric, and $f_{ij}(L)$ is a symmetric factor that takes into account the contribution of Sonine polynomials of order $L \geq 2$ with the constraint $f_{ij}(1) = 1$. The thermal diffusion ratios k_i^T are defined as:

$$D_i^T = \sum_{j=1}^{N_s} D_{ij} k_j^T \tag{2.113}$$

with the constraint $\sum_{i=1}^{N_s} k_i^T = 0$. Since that the Stefan-Maxwell equations are not lineary independent, to properly close the system we add the condition:

$$\sum_{i=1}^{N_s} \mathbf{J}_i = 0 \tag{2.114}$$

By referring to Mutation++[13], the diffusion fluxes are in the form:

$$\mathbf{J}_i = -\rho_i \left(\sum_{j=1}^{N_s} D_{ij} \left(d_j + k_j^T \nabla \log T \right) \right) \tag{2.115}$$

which is equivalent(see thermal diffusion ratio definition). Instead, the driving forces vector is in the form:

$$\mathbf{d}_j = \frac{\nabla p_j}{n k_b T} - \frac{y_i p}{n k_b T} \nabla \log(p) - k_j \mathbf{E} \tag{2.116}$$

I have reasons to think that the 2 approaches are equivalent, since that the same hypotheses are made, but this should be verified in the future. It is good to notice that Mutation++ prefers the generalized Stefan-Maxwell system over the transport sustems for the multicomponent diffusion matrix.

Viscous stress tensor

The viscous stress tesor in the *Chapman Enskog approximation* is computed by means of Eq. 2.92 and Eq. 2.102:

$$\overline{\overline{\overline{\tau}}} = \mu \left(\nabla \mathbf{V} + \nabla \mathbf{V}^T\right) + \left(\eta - \frac{2}{3}\mu\right)(\nabla \cdot \mathbf{V})\, \mathbf{I} \tag{2.117}$$

where we identified the dynamic viscosity μ and the bulk viscosity η as:

$$\mu = \frac{kT}{10}[B, B] \tag{2.118}$$

$$\eta = kT\, [F, F] \tag{2.119}$$

The volume viscosity arises in polyatomic gases and it is absent in monoatomic ones. It is linked with the relaxation of internal degrees of freedom in molecules in the case of weak deviation from the equilibrium.

It is interesting to notice that the The contribution of the Maxwellian part of the distribution function reduces to the hydrostatic pressure.

The bulk/volume viscosity is usually neglected (**Stokes Hypothesis**) in multicomponent flow modeling: the main reason being that the necessary data to correctly computing it are lacking. Experimental results on acoustic waves absorption show that μ and η are of the same order of magnitude, so neglecting η has no justification a priori. What eventually can be neglected is the influence of the term $\eta \nabla \cdot \mathbf{VI}$, that is a good approximation for low Mach number flows.

Following a similar approach, the authors of *Mutation++* also include a term $-p^{\text{react}}\mathbf{I}$ in the viscous stress tensor, where p^{react} is the chemical pressure. They eventually neglect the term, after claiming that for gases in the Maxwelling regime, the chemical pressure terms vanishes, and it is generally thought to be small in comparison to the other terms. The authors also neglect η as we are doing in this work.

The shear viscosity is determined by means of a polynomial expansion of the coefficients B_i:

$$B_i = \sum_{p=0}^{L-1} b_{i,p}^{(L)}\, S_{\frac{5}{2}}^{p}(W_i) \tag{2.120}$$

Inserting the expansion in the definition of the dynamic viscosity, we get:

$$\mu = \frac{kT}{2} \sum_{i=1}^{N_s} \frac{n_i}{n} b_{i,0}^{(L)} \tag{2.121}$$

Like for the diffusion coefficients, the coefficients $b_{i,p}^{(L)}$ are determined by means of a linear system of dimension $N_s L$ by $N_s L$, which coefficients are function of the collision integrals.

By using *Mutation++*, we can choose between:

- CG: Conjugate-Gradient method,$O(mn^2)$, m is the number of iterations, iterative method. Exact solution is seaked from the linear system.
- LDLT: Cholesky solution via LDLT decomposition, $O(n^3/6)$, direct method. Exact solution is seaked from the linear system.
- Gupta-Yos: Approximate mixture rule. Consider the diagonal elements dominant wrt the non-diagonal ones.
- Wilke: Approximate mixture rule.

We should notice that by using this model we making the following assumptions:

- The fluid is Newtonian, i.e. the shear stress is linearly proportional to the velocity gradient.
- The stress tensor is symmetric, i.e. the fluid has no local torque proportial to the volume, as it would be possible In an electric field.
- The fluid is isotropic, i.e. there is no locally preferred direction, so the main axes of the stress and rate of deformation tensors coincide.
- The Stokes hypothesis is valid, i.e. the bulk viscosity is neglected (No relaxation processes may occur, so the relaxation times for internal transfer processes must be very small compared to the deformation times.)

Heat flux

The heat flux in the *Chapman-Enskog approximation* is computed by means of Eq. 2.93 and Eq. 2.102:

$$\mathbf{q} = -\lambda \nabla T + \sum_{i=1}^{N_s} h_i \mathbf{J_i} - p \sum_{i=1}^{N_s} D_i^T \mathbf{d}_i \tag{2.122}$$

where λ is the coefficient of partial thermal conductivity and is given by:

$$\lambda = \frac{k}{3}[A, A] \tag{2.123}$$

The first term in the heat flux is the Fourier term, the second term represents the transfer of enthalpy due to particles diffusion amd the third one the Dufour effect (the heat transfer due to concentration gradients).

The previous expression can be modified in order to use the thermal diffusion ratios k_i^T [3]:

$$\mathbf{q} = -\bar{\lambda} \nabla T + \sum_{i=1}^{N_s} h_i \mathbf{J}_i - p \sum_{i=1}^{N_s} k_i^T \mathcal{V}_i \tag{2.124}$$

where $\bar{\lambda}$ is the **coefficient of thermal conductivity** and is defined as:

$$\bar{\lambda} = \lambda - \frac{p}{T} \sum_{i=1}^{N_s} D_i^T k_i^T \tag{2.125}$$

The difference between λ and $\bar{\lambda}$ is often of the order $\left(D_i^T\right)^2$ and is therefore small, since that the diffusion coefficients are small.

The partial theraml conductivity is once more computed with a polynomial expansion of the A_i coefficients:

$$A_i = \sum_{p=0}^{L} a_{i,p}^{(L)} \, S_{\frac{3}{2}}^{p}(W_i) \tag{2.126}$$

Inserting the expansion in the definition of the partial thermal conductivity, we get:

$$\lambda = \frac{5}{4} k \sum_{i=1}^{N_s} \frac{n_i}{n} a_{i,1}^{(L)} \tag{2.127}$$

The coefficients $a_{i,p}^{(L)}$ are determined by means of a linear system of dimension $N_s(L+1)$ by $N_s(L+1)$, which coefficients are function of the collision integrals.

The presence, in the mixture, of particles with internal degrees of freedom influences the thermal conductivity coefficient. This happens because inelastic collisions affect the energy transfer between particles and therefore the coefficient of thermal conductivity. The modification of the coefficient of thermal conductivity depends upon the transition probabilities for the transfer of energy among the various degrees of freedom, particularly on the probability of transfer from the translational to the internal degrees of freedom. Because the transition probabilities are largely unknown, the additional contribution to the thermal conductivity is determined by using the so-called Eucken correction. This approximation is applicable when the transfer of energy is fast, i.e. when the gas is essentially in thermal equilibrium. The resulting formula for the additional internal thermal conductivity coefficient:

$$\lambda_h^{int} = k \sum_{i=1}^{n} \frac{\frac{c_{p,i}^{\mathrm{int}}}{\mathcal{R}} x_i}{\sum_{j=1}^{n} x_j \Delta_{ij}^1} \tag{2.128}$$

Ionized mixtures

In ionized mixtures, the interaction between charged particles is governed by the Coulomb force. Unfortunately, the decay of the Coulomb potential at large interparticle separations is slow ($\sim r^{-1}$), that collisions cannot be considered as short duration, binary encounters: on the opposite, each (charged) particle interacts simultaneously with many other particles.

A direct consequence of that, not only the Chapman-Enskog solution is invalid for ionized mixture because it is based on the binary collusion assumption, but also that even if it is used as a first approximation to copmute the transport properties, the collision integrals needed for their evaluation do not converge.

Fortunately, it is possible to show that collisions between charged particles can be treaded as binary encounters provided that the Coulomb potential is replaced by the **Screened Coulomb potential**:

$$V_{ij}(r) = \frac{q_i q_j}{r} e^{-\frac{r}{d}} \tag{2.129}$$

where q_i and q_j are the charges of the particles, r is the iterparticle distance and d is the **Debye length**, that in the maximum distance over which the mixture charge neutrality can be violated. Interactions between charged particles at a distance greater than the Debye length can be safely neglected since that the screened Couloumb potential is strongly attenuated,while those at a shorter distance may be considered as binary collusions.

The derivation of the transport properties (assuming that the effect of the magnetic field can be neglected, that is usually true for this kind of applcation) can be now carried out exactly as in the previous sections. Nevertheless, Devoto has shown that, in order to accurately compute the transport porperties (in particular the thermal conductivity), more terms have to be retained in the Sonine polynomial expansion than in the neutral mixture case. This increases the computational load assocuated with the comptutation of the transport properties. Fortunately, Devoto has shown that, by propertly taking into account the fact that the mass of the electrons is much smaller than the one of the heavy particles, the Champman-Enskog procedure can be consistenly simplified and the transport properties are computed as being the sum of two separate contributions (as we did for the thermal conductivity). It turns out that the heavy particles contribution is well approximated by one or two terms of the polynomial expansion, as before. Electrons contgribution needs more terms in the polynomial expansion, but explicit expressions, which are relatively straightforward to evaluate, are available. Furthermore, the effects on the viscosity can either be neglected or approximated by a low order expression.

In particular, Devoto's third approximation for the electron thermal conductivity can be expressed as follow[3]:

$$\lambda_e^3 = \frac{75 n_e^2 k}{8} \left(\frac{2\pi k T_e}{m_e} \right)^{1/2} \frac{1}{q^{11} - (q^{12})^2 / q^{22}} \tag{2.130}$$

Let's note that the rigorous expression as derived from Hirschfelder's theory are computationally expensive, and this is the reason why Devoto's approximation is used.

In the Mutation++ thesis[13], the heat flux is expressed has:

$$\mathbf{q} = \sum_k \rho_k h_k \mathcal{V}_k + \sum_k (\chi_k^h + \chi_k^e) \mathcal{V}_k - \sum_m \lambda_m \nabla T_m \tag{2.131}$$

which is equivalent to the one expressed in Eq. 2.124, with the electron contribution.

Collision integral

As I said multiple times, the coefficients of the polynomial expansions for the transport properties are found by solving linear systems of equations, which coefficients are functions of the collision integrals.

The collision integral can be computed accurately if an hypothesis can be made on the type of collision, i.e. of the type of interaction between the colliding particles. Figure 2.1 shows an example of collision:

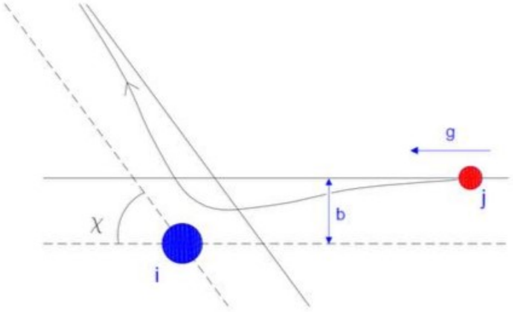

Figure 2.1: Binary elastic collision between particles i and j as seen in a frame moving with the particle i.

The scattering angle χ of the collision depends on the reduced mass μ of the two particles and on the interaction potential Φ for the collision under consideration. The expression is:

$$\chi(g_{ij}, b) = \pi - 2b \int_{r_m}^{+\infty} \frac{1/r^2 \, dr}{\left[1 - b^2/r^2 - 2\Phi_{ij}(r)/(\mu g_{ij}^2)\right]^{1/2}} \tag{2.132}$$

where g_{ij} is the modulus of the relative velocity between the two coliding particles. The lower limit r_m of the integral is the closest distance between i and j particles during the encounter and it may be computed from the expression:

$$1 - \frac{b^2}{r_m^2} - \frac{2\Phi_{ij}(r_m)}{\mu g_{ij}^2} = 0 \tag{2.133}$$

We define the generalised cross-section $Q_{ij}^l(g_{ij})$ as:

$$Q^l(g_{ij}) = 2\pi \int_0^{+\infty} \left(1 - cos^l(\chi)\right) b\, db \tag{2.134}$$

The collision integral is then defined as:

$$\Omega_{ij}^{l,k} = \sqrt(\frac{kT}{2\pi\mu}) \int_0^{+\infty} \gamma^{2k+3} e^{-\gamma^2} Q^l(g_{ij})\, d\gamma \tag{2.135}$$

where γ is defined as:

$$\gamma^2 = \frac{1}{2}\mu \frac{g_{ij}^2}{kT} \tag{2.136}$$

where k is the Boltzmann constant, T is the temperature and μ is the reduced mass of the two particles:

$$\mu = \frac{m_i m_j}{m_i + m_j} \tag{2.137}$$

The practical formulas use the reduced collision integral $\omega_{ij}^{\bar{l},k}$, which is defined as:

$$\omega_{ij}^{\bar{l},k} = \frac{4(l+1)}{(k+1)![2l+1-(-1)^l]} \sqrt{\frac{2\pi\mu}{kT}} \Omega_{ij}^{l,k} \tag{2.138}$$

The collision integral are hard to compute in most cases, so usually they are tabulated and interpolated in order to be used in the computation of the transport properties.

Comparing the collision integral used in the old VKI Pegase Library and the ones used in the Mutation++ library, we can see that the ones used in the Mutation++ library are more specific and detailed, and they are computed in a more accurate way. It is reasonable to assume that the ones used in the Mutation++ library are more accurate than the ones used in the VKI Pegase library.

Thermal conductivity contributions

Let's now focus on the expression for the thermal conductivity. In particular, we now focus on the **Frozen thermal conductivity** λ_{fr}, such that:

$$\mathbf{q} = -\lambda_{fr} \nabla T \tag{2.139}$$

This is the thermal conductivity of the gas when no diffusion is considered. We can split this thermal conductivity into two components:

$$\lambda_{fr} = \lambda_{fr}^t + \lambda_{fr}^{int} \tag{2.140}$$

The first one is the thermal conductivity due to the translational degrees of freedom, while the second one is the thermal conductivity due to the internal degrees of freedom. The translational thermal conductivity can be expressed as a sum of the heavy particles and the electrons contributions:

$$\lambda_{fr}^t = \lambda_h^t + \lambda_e^t \tag{2.141}$$

These thermal conductivities can be computed by means of the Chapman-Enskog expansion, as we have seen before. While the internal thermal conductivity is often computed using the **Euken corrections**, as we have seen before.

It is easy to compute this thermal conductivity by using Mutation++, which uses the collision integrals. By comparing to PEGASE, very similar results are found, since It also use the same procedure (but the collision integrals are different).

We have seen that the total heat flux is the sum of the heat flux due to the frozen thermal conductivity and the heat flux due to diffusion and thermal diffusion. It is not always easy to express the latter two contributions in terms of the thermal conductivity. Let's try to understand how this can be done.

Both contributions depend on the diffusion velocities, so we need to examine them further. I will now refer to Bottin's thesis[4] for the following discussion. We have seen that the diffusion velocities have the form:

$$\mathcal{V}_i = -\sum_{j=1}^{N_s} D_{ij} \left(\mathbf{d}_j + k_j^T \nabla \log T\right) \tag{2.142}$$

Let's also recall that the driven forces have the form:

$$\mathbf{d}_j = \nabla x_j + \left(x_j - \frac{\rho_j}{\rho} \right) \nabla \log p - \frac{\rho_j}{\rho p} \left(\rho \mathbf{F}_j - \sum_{k=1}^{N_s} \rho_k \mathbf{F}_k \right) \tag{2.143}$$

We can see that the diffusion is due to gradients in the chemical composition, pressure, temperature and body forces.

Now, it can be seen that the pressure diffusion is negligile. Indeed, the pressure gradient is somewhat "smoothed out", because it is expressed as a gradient of the logarithm of pressure. Moreover, in most flow applications, the pressure gradient is either null (boundary layer) or small compared to the temperature gradient. This leafs to the neglect of pressure diffusion.[4] By neglecting the body forces, we get

$$\mathbf{d}_j = \nabla x_j \tag{2.144}$$

This leads to the following expression for the diffusion fluxes:

$$\mathbf{J}_i = -\rho_i \sum_{j=1}^{N_s} D_{ij} \left(\nabla x_j + k_j^T \nabla \log T \right) \tag{2.145}$$

It seems like there is a sign discordance between the Bottin and Barbante in this equation for the gradient of the concentration, but the Stefan-Maxwell equations is the same, so I will ignore this discordance.

This yields to the Stefan-Maxwell relations for the diffusion fluxes:

$$\frac{M}{\rho} \sum_{j=1}^{N_s} \left(\frac{x_i \mathbf{J}_j}{M_j \mathcal{D}_{ij} f_{ij}(L)} - \frac{x_j \mathbf{J}_i}{M_i \mathcal{D}_{ij} f_{ij}(L)} \right) = \nabla x_i \tag{2.146}$$

where the thermal diffusion (Soret effect) has been neglected. This effect is usually neglected because the coefficient of diffusion can be obtained, like viscosity or thermal conductivity, with one non-vanishing Sonine polynomial in the series expansion. With this level of approximation, the thermal diffusion coefficient is identically zero. In order to obtain a non.zero value, 2 vanishing Sonine polynomials must be used. This effectively doubles the size of the system to be solved in order to account for an effect that is of second order only. This explains why thermal diffusion is neglected. Devoto quantifies the neglect of thermal diffusion to 1% of the total diffusion.[4]

Now, we would like to define a thermal conductivity to account for the diffusion fluxes in the heat flux.

Let's start by neglecting the thermal diffusion in the heat flux, that is the last term in Eq. 2.124.

So we want to focus on the term: $\sum_{i=1}^{N_s} h_i \mathbf{J}_i$. This transport of chemical energy by diffusion an also be seen as a sort of therml conduction. It has been shown by Butler and Brokaw that, under conditions of local chemical equilibrium, this effect could be properly taken into account by means of a reactive thermal conductivity λ_r. By doing so, the **equilibrium thermal conductivity** λ_{eq} can be defined as:

$$\lambda_{eq} = \lambda_{fr} + \lambda_r \tag{2.147}$$

in such a way that the heat flux can be expressed as:

$$\mathbf{q} = -\lambda_{eq} \nabla T \tag{2.148}$$

in the hypothesis of local chemical equilibrium.

In order to do so, Butler and Brokaw have suggested to obtain the reactive thermal conductivity from its definition in the expression of the heat flux rate. Indeed, they wanted to find:

$$\sum_{i=1}^{N_s} h_i \mathbf{J}_i = -\lambda_r \nabla T \tag{2.149}$$

The diffusion fluxes can be known by solving the Stefan-Maxwell equations, if the gradient of the mole fractions are known. However, the need of computing the thermal conductivity independently of any flow field data requires an alternative to computing this quantity a priori. Through the use of Van't Hoff's relation, Butler and Brokaw express the gradient of mole fractions in function of the gradient of temperature. Hence, the gradient of temperature appears on both sides of the equation, and the reactive thermal conductivity can be computed:

$$\lambda_r = k \sum_{i=1}^{n_r} \Delta \hat{h}_R^{(r)} \mathbf{W}_r \tag{2.150}$$

It is possible to use a mixture rule simplification of this approach proposed by Yos:

$$\lambda_r^Y = k \sum_{i=1}^{n_r} \frac{(\Delta H_{\text{react}}^{(r)} / (\mathcal{R} T))^2}{\sum_{i=1}^{N_s} \frac{\nu_i^{(r)}}{x_i \sum_{j=1}^{N_s} (\nu_i^{(r)} x_j - \nu_j^{(r)} x_i) \Delta_{ij}^{(1)}}} \tag{2.151}$$

However, this rule is not really accurate when the degree of ionisation becomes significan [4].

The above formualation supposes that the net diffuses fluxes of basic elements be zero, which may not be true in general but is verified in practical cases of chemical equilibrium. Butler and Brokaw derived these resuts for neutral mixtures. It is not a priori obvious that the result still holds true for ionised mixtures. Electron, due to their small mass, tend to diffuse much faster than heavy paricles. This diffusion causes slight violations of charge neutrality and generates corresposnding electric fields, which apper as additional diffusion driving forces (as we have seen from the Mutation++ formulation) which tend to dramatically slow down the diffusing electrons and, to a lesser extent, also affect the diffusing ions, such that overall quasi-neutrality is maintained in the plasma. This situation is referred to as **ambipolar diffusion**. It was shown by Vanden Abeele that Butler and Brokaw's hypothesis of zero elemental fluxes, when applied to ionisez mixtures, automatically yielded the ambipolar diffusion condition. Applying Butler and Brokaw's deruvation for ionised mixtures taking into account the electric field generated by ambipolar diffusion, it was found out that these extra terms were dropping out of the equations. Therefore, the same final result of Butler and Brokaw holds true for ionised mixtures as well.

From this discussion, by following Bottin's thesis, we now have a way to compute the equilibrium thermal conductivity. By taking a look to PEGASE library, we can see that the equilibrium thermal conductivity is computed by following the same procedure as the one described above, where the user can decide is the reactive thermal conductivity has to be computed with the exact Butler and Brokaw formula or with the Yos approximation.

Since I am going to use Mutation++ to compute the transport properties, let's see how the equilibrium thermal conductivity is computed in this library. Let's recall that in Mutation++, the diffusion velocity is expressed as:

$$\mathcal{V}_i = -\sum_{j=1}^{N_s} D_{ij}^T \left(\mathbf{d}_j + k_j^T \nabla \log T\right) \tag{2.152}$$

with:

$$\mathbf{d}_j = \frac{\nabla p_j}{nk_bT} - \frac{y_i p}{nk_bT}\nabla \log(p) - k_j\mathbf{E} \tag{2.153}$$

Now let's consider constant pressure (so the pressure gradient is also neglected here) and thermochemical equilibrium. We can rewrite the driving force as:

$$\mathbf{d}_j = \frac{\nabla p_j}{nk_bT} - k_j\mathbf{E} = \nabla x_j - k_j\mathbf{E} \tag{2.154}$$

since $p = nk_bT$ and $x_j = \frac{M}{M_i}y_i$. We can notice that we got the same expression as before (with Bottin), but here the electric field is not neglected. Now, by substituting this expression in the definition of the diffusion velocity, we get:

$$\mathcal{V}_i = -\sum_{j=1}^{N_s} D_{ij}^T \left(\frac{\partial x_j}{\partial T} + \frac{k_j^T}{T}\right)\nabla T + \sum_{j=1}^{N_s} D_{ij}^T k_j \mathbf{E} \tag{2.155}$$

This is because the gradient of the mole fractions has been written has:

$$\nabla x_j = \frac{\partial x_j}{\partial T}\nabla T \tag{2.156}$$

since the pressure gradient is neglected. Furthermore, the gradient of the logarithm of the temperature, has been written as:

$$\nabla \log T = \frac{1}{T}\nabla T \tag{2.157}$$

We should note that another hypothesis made is that the element fraction is constant (to clarify why).

Now, the electric field is specified or determined from the ambipolar constrain:

$$\mathbf{E}_a = E_a^{\nabla T}\nabla T \tag{2.158}$$

where the expression for $E_a^{\nabla T}$ can be found in the Mutation++ thesis[13], page 98.

Thanks to these hypotheses, we can define 2 new thermal conductivities, since now the thermal diffusion is not neglected:
The reactive thermal conductivy:

$$\lambda_r = \sum_{i=1}^{N_s}\sum_{j=1}^{N_s} \rho_i\, h_i\, D_{ik}\left(\frac{\partial x_j}{\partial T} + \frac{k_j^T}{T} - k_j E_a^{\nabla T}\right) \tag{2.159}$$

The Soret thermal conductivity:

$$\lambda_s = p\sum_{i=1}^{N_s}\sum_{j=1}^{N_s} k_i^T\, D_{ik}\left(\frac{\partial x_j}{\partial T} + \frac{k_j^T}{T} - k_j E_a^{\nabla T}\right) \tag{2.160}$$

Let's recall that $k_j^T = k_{j,h}^T + k_{j,e}^T$ are the thermal diffusion ratios.

The equilibrium thermal conductivity is then defined as:

$$\lambda_{eq} = \lambda_{fr} + \lambda_r + \lambda_s \tag{2.161}$$

The Soret thermal conductivity is usually neglectable, since it is very small compared to the other thermal conductivities.

So, in conclusion, with Mutation++ we can compute the equilibrium thermal conductivity, and, compared to PEGASE, we are not neglecting the thermal diffusion, which is usually very small, so the difference between the two should be very small.

By comparing the libraries, a big difference 20% is found in the reactive thermal conductivity. This is also noticed in the Mutation++ thesis when comparing the results with Wilke. This difference is due to the different collision integrals used in the two libraries.

Lastly, a note can be made on the reactive thermal conductivity computed with the Brokaw and Butler formula. This formula is easier to use because it doesn not require to compute the derivative of the composition with respect to the temperature. However, this formula does not account for *Elemental demixing*, which is important to consider in CFD simulation of flows in thermochemical equilibrium, since elemental demixing can occour.

In conclusion, we can compute the equilibrium heat flux has:

$$\mathbf{q} = -\lambda_{eq} \nabla T \tag{2.162}$$

where λ_{eq} is the equilibrium thermal conductivity, which is computed by Mutation++.

A lot of hypotesis have been made in order to compute the heat flux, but since we are using the Mutation++ library, we are constrained to them. One of them is that the pressure gradient can be neglected in the heat flux computation. Also according to [10], pressure diffusion can be neglected in agreement with boundary layer simplifications, which is the case we will work on.

Summary on thermal conductivities

In order to compute the heat flux in the simply form:

$$\mathbf{q} = -\lambda_{eq} \nabla T \tag{2.163}$$

we need to compute the **equilibrium thermal conductivity** λ_{eq}.

We can do this by defining three new coefficients for the thermal conductivity[3, 13, 4]:

1 **Frozen thermal conductivity** λ_{fr}: this is the effective thermal conductivity of the gas when no chemical reaction occurs, and it is the one that we called $\bar{lambda}$ in Eq. 2.124. This thermal conductivity can be split into two components:

$$\lambda_{fr} = \lambda_{fr}^t + \lambda_{fr}^{int} \tag{2.164}$$

where λ_{fr}^t is the thermal conductivity due to the translational degrees of freedom and λ_{fr}^{int} is the thermal conductivity due to the internal degrees of freedom (vibrational and electronic). The translational thermal conductivity can be expressed as:

$$\lambda_{fr}^t = \alpha_1 \lambda_h + \alpha_2 \lambda_e \tag{2.165}$$

where λ_h and λ_e are the thermal conductivities of the heavy species and electrons, respectively, and α_1 and α_2 are coefficients that indicate whether or not the global mode includes translational energy of heavy species and electrons respectively. The internal thermal conductivity can be found using the **Euken corrections**, that are exact when the internal energy modes are separable and inelastic collisions are neglected (it holds true inasmuch as the hypoteses of RRHO are made for neutral and ionised molecules and that local thermodynamic equilibrum is verified). The electrons thermal conductivity can be found using Devoto higher order formulas.

2. **Reactive thermal conductivity** λ_r: this is the thermal conductivity due to the reaction of the species in the mixture. It is defined as the λ_r such that:

$$\sum_{i=1}^{N_s} h_i \mathbf{J}_i = -\lambda_r \nabla T \tag{2.166}$$

The specific expression for λ_r can be found in the Mutation++ thesis[13]. Non negligible.

3. **Soret thermal conductivity** λ_s: this is the thermal conductivity due to heat diffusion due to concentration gradients. It is defined as the λ_s such that:

$$-p\sum_{i=1}^{N_s} D_i^T \mathbf{d}_i = -\lambda_s \nabla T \tag{2.167}$$

The specific expression for λ_r can be found in the Mutation++ thesis[13]. Usually neglectable.

The equilibrium thermal conductivity is then defined as:

$$\lambda_{eq} = \lambda_{fr} + \lambda_r + \lambda_s = \lambda_{fr}^t + \lambda_{fr}^{int} + \lambda_r + \lambda_s \tag{2.168}$$

2.2.11 Stationary flow

I will assume that the flow is stationary: this means that the flow fields are not time dependent, so, given the field $\mathbf{V}$, we have:

$$\frac{\partial \mathbf{V}}{\partial t} = 0 \tag{2.169}$$

This hypotesis simplify the governing equations, and it is good in first approximation. The reader should also recall that, for a stationary flow, streamlines, pathlines and streaklines coincide[11].

2.3 Model of the flow field for our problem

So far, I made a lot of hypotheses regarding the thermodynamic properties of the gas, but, aside for the hypotesis of stationary flow, I did not say anything about the flow field. In this section, I will describe the model of the flow field that I will use in this work.

I will only consider the stagnation line of the flow (this means the flow is 1D), that is the flow streamline that passes through the stagnation point, that is the point where the flow velocity is zero. Since that the flow is stationary, this is also a pathline for the flow. Figure 2.2 shows an example of stagnation line of a flow around a body.

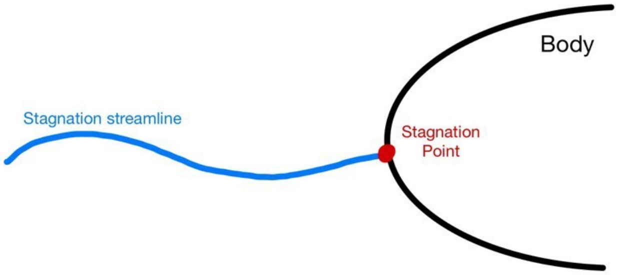

Figure 2.2: Stagnation line of a flow around a body.

In order to simplify the governing equation, I will assume that the flow is **Eulerian**, i.e. the flow is inviscid and adiabatic (no heat transfers) everywhere except for the boundary layer, next to the solid surface of the body. Figure 2.3 shows an example of Eulerian flow. Furthermore, the external flow is irrotational. No angle of attack is considered, so the flow is symmetric with respect to the stagnation line.

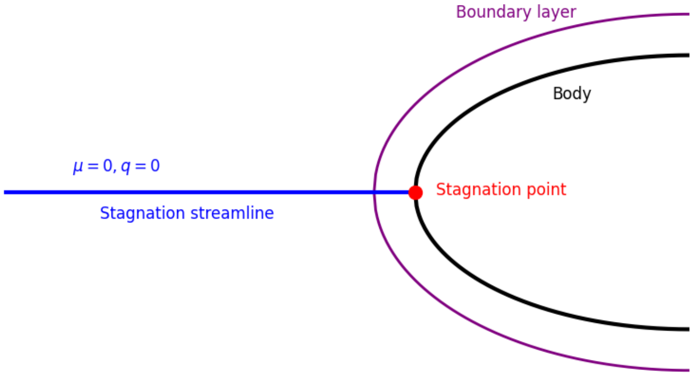

Figure 2.3: Eulerian flow around a body.

In the following subsections I am going to describe the model inside and outside the boundary layer.

2.3.1 Flow outside the boundary layer

As I said, I will consider the flow on the stagnation line outside the boundary layer as Eulerian, i.e. inviscid and adiabatic. Let p, T, u, be the pressure, temperature and velocity of the flow, respectively, of the free stream. In the stagnation point we will have $u = 0$, $p = p_0$ and $T = T_0$, where the subscript 0 indicates the stagnation values.

Now, let's take a look to Eq. 2.26 for the conservation of entropy. Since that outside the boundary layer the flow is adiabatic, $\mathbf{q} = \mathbf{0}$; furthermore, since that the flow is inviscid, $\overline{\overline{\tau}} = \mathbf{0}$, and so ϵ, the viscous dissipation, is zero. The equation becomes:

$$\frac{Ds}{Dt} = \mathbf{u} \cdot \nabla s = 0 \tag{2.170}$$

When can see this equation is two ways: the first is that the entropy saw by a fluid particle is constant, so this means that the entropy is constant along a pathline; the second is that the entropy is constant along a streamline, since that the gradient of the entropy has to be perpendicular (since that the dot product is 0) to the velocity, which il parallel to the streamline by definition. Since that the flow is stationary, the pathlines coincide with the streamlines, so the two concepts are equivalent. We understood that the flow is isentropic outside the boundary layer, so the entropy is constant from the freestream to the stagnation point (outside the boundary layer, since that inside, we have heat transfers and viscous stresses).

Now, we recall the definition of total conditions for a flow: the total conditions are the conditions that the flow would have if it were brought to rest (zero velocity) isentropicaly. The total conditions are denoted with the subscript t, p_t, T_t are the total pressure and temperature, respectively. We notice that in this case, the total conditions coincide with the stagnation conditions, since that the flow is isentropic. This is useful because we can compute the total conditions of the flow, like the enthalpy and entropy, by knowing the total pressure and temperature, p_t and T_t:

$$\begin{cases} h_t = h_0 = h(T_t, p_t) \\ s_t = s_0 = s(T_t, p_t) \end{cases} \tag{2.171}$$

Let's now recall the Eq. 2.24 for the conservation of energy. Since that the flow is steady, adiabatic, $\mathbf{q} = \mathbf{0}$, inviscid, $\overline{\overline{\tau}} = \mathbf{0}$, and we negled the body forces, $\mathbf{F}_{gi} = \mathbf{F}_{ei} = \mathbf{0}$, the equation becomes:

$$\nabla \cdot (\rho E \mathbf{u} + p\mathbf{u}) = \nabla \cdot \left(\rho\mathbf{u}\left(h + \frac{1}{2}|\mathbf{u}|^2\right)\right) = 0 \tag{2.172}$$

where $h = e + p/\rho$ is the specific enthalpy of the flow. Expanding the divergence, we get:

$$\rho\mathbf{u} \cdot \nabla \left(h + \frac{1}{2}|\mathbf{u}|^2\right) + \nabla \cdot (\rho\mathbf{u}) \left(h + \frac{1}{2}|\mathbf{u}|^2\right) = 0 \tag{2.173}$$

Let's recall Eq. 2.11 for the conservation of mass. Since that the flow is steady, the equation becomes:

$$\nabla \cdot (\rho\mathbf{u}) = 0 \tag{2.174}$$

So the energy equation becomes:

$$\rho\mathbf{u}\cdot\nabla\left(h+\frac{1}{2}|\mathbf{u}|^2\right)=\rho\frac{D}{Dt}\left(h+\frac{1}{2}|\mathbf{u}|^2\right)=0 \tag{2.175}$$

This, again, can be seen in two ways: the first is that the quantity $h+\frac{1}{2}|\mathbf{u}|^2$ is constant along a pathline, and the second one is that the quantity is constant along a streamline. Since that the flow is stationary, the two concepts are equivalent.

This last result tells us that the quantity $h+\frac{1}{2}|\mathbf{u}|^2$ is constant along the stagnation streamline. Since that the freestream flow has velocity u and enthalpy h, and the stagnation point has velocity 0 and enthalpy h_0, we have:

$$h_0=h_t=h+\frac{1}{2}u^2 \tag{2.176}$$

This equation tells us that the stagnation enthalpy, which is equal to the total enthalpy, is the sum of the enthalpy of the freestream flow and the kinetic energy of the flow.

At this point we sucsefully described the flow outside the boundary layer, and we found the relationships between the stagnation and freestream conditions.

$$\begin{cases} h_t(T_t,p_t)=h(T,p)+\frac{1}{2}u^2 \\ s_t(T_t,p_t)=s(T,p) \end{cases} \tag{2.177}$$

By solving this system of equations, we can link the stagnation (and total) and freestream conditions of the flow.

2.3.2 Flow inside the boundary layer

I will now describe the flow inside the boundary layer. I will assume that the flow inside the boundary layer can be described by the **Classical boundary layer theory**. This theory let us simplify the Navier-Stokes equations inside the boundary layer by making some assumptions. We can recall that this corresponds to an asymptotic expansion of the Navier-Stokes equations of order 0.

First of all, we define the Reynold number of the flow as:

$$Re=\frac{\rho uL}{\mu}=\frac{uL}{\nu} \tag{2.178}$$

where L is a characteristic length of the body, and ν is the kinematic viscosity of the fluid, defined as $\nu=\mu/\rho$. In the boundary layer theory, the **Reynold number** has to tend, in the limit, to infinity, and the flow has to be attached to the body. This means that the boundary layer thickness, $\delta\sim\frac{1}{\sqrt{Re}}$[7] as $Re\to+\infty$, has to tend to zero. In hypersonic regimes the boundary layer thickness can be nonnegligible even for high Reynold numbers, and this has to be watched out[3]. To improve the situation one can include in the boundary layer equation higher order terms (with respect to the Reynolds number: asymptotic expansions of higher orders) that take into account, for example, body curvature effects, vorticity entrainment at the boundary layer edge, etc.... A further step is to use the *Viscous Shock Layer* equations that are valid also outside of the boundary layer up to the shock location and are still less expensive to solve than the full Navier-Stokes equations (but more expensive than the boundary layer equations). Here, I will simply use the boundary layer theory, but I will show that this approach is a good approximation next to the stagnation point, that is where I need to solve them for the computation of the stagnation heat flux.

I am going to make the following assumptions:

- The flow is axisymmetric and laminar inside the boundary layer. This let us treat the boundary layer with only two coordinates.
- The recall some of the hypoteses made in the previous sections that are particularly useful in the boundary layer theory: the flow is steady, in quasi-thermodynamic equilibrium, no body forces due to external fields are considered.

Under this assumptions, the Navier-Stokes equations become:

$$\begin{cases} \nabla\cdot(\rho\mathbf{u})=0 \\ \rho\mathbf{u}\cdot\nabla\mathbf{u}=-\nabla p+\nabla\cdot\overline{\overline{\tau}} \\ \rho\mathbf{u}\cdot\nabla h=\mathbf{u}\cdot\nabla p-\nabla\cdot\mathbf{q}+\overline{\overline{\tau}}:\nabla\mathbf{u} \end{cases} \tag{2.179}$$

where the colon is the double dot product between two tensors, $\overline{\overline{a}}:\overline{\overline{b}}=a_{ij}b_{ij}$.

The first two equations of Eq. 2.179 simply derive by the conservation of mass and momentum, Eq. 2.11 and Eq. 2.22, respectively. The third one derive from the conservation of energy, Eq. 2.24, with some extra steps: As we saw in Eq.

2.175, when the flow is steady, the left hand side of the energy conservation equation can be rewritten as $\rho \frac{D}{Dt}\left(h + \frac{1}{2}|\mathbf{u}|^2\right)$. Now, since that we have heat transfers and viscous stresses, the right hand side is not zero:

$$\rho \frac{D}{Dt}\left(h + \frac{1}{2}|\mathbf{u}|^2\right) = \nabla \cdot \left(\overline{\overline{\tau}} \cdot \mathbf{u}\right) - \nabla \cdot \mathbf{q} \tag{2.180}$$

Now we would like to remove the term $\frac{D}{Dt}\left(\frac{1}{2}|\mathbf{u}|^2\right)$ from the equation. We can do this by multiply the conservation of the momentum by $\mathbf{u}$. By doing this and simplifying, we get[11]:

$$\frac{D}{Dt}\left(\frac{1}{2}|\mathbf{u}|^2\right) = \mathbf{u} \cdot \nabla \left(\frac{1}{2}|\mathbf{u}|^2\right) = \mathbf{u} \cdot \nabla \left(\frac{1}{2}|\mathbf{u}|^2\right) + \mathbf{u} \cdot \nabla \left(\frac{p}{\rho}\right) = -\mathbf{u} \cdot \nabla p + \mathbf{u} \cdot \left(\nabla \cdot \overline{\overline{\tau}}\right) \tag{2.181}$$

We can now subtract this equation from Eq. 2.180 to get the final form of the energy conservation equation as shown is Eq. 2.179.

Clerly these equations are valid for an arbitrary inertial coordiante system, if the correct form of the nabla operator (gradiet,divergence, etc...) is used. In particular, we are intered in the natural coordinate system in axisymmetric flows:

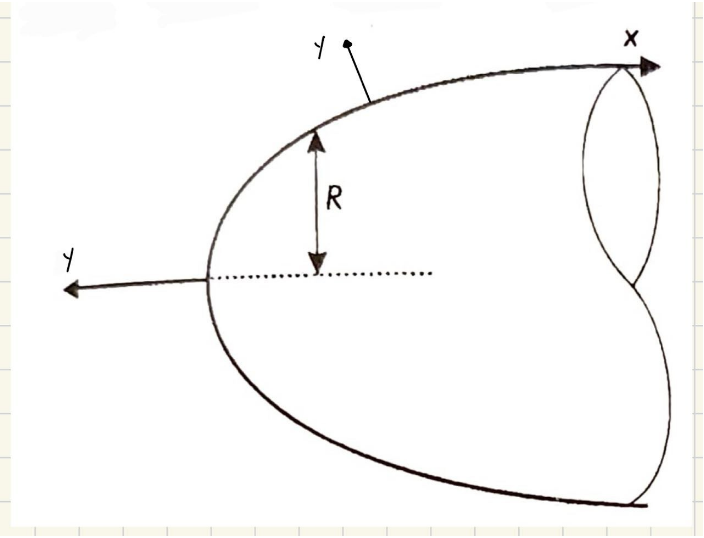

Figure 2.4: Natural coordinate system in axisymmetric flows.

In this coordinate system, the x axis lies on the body surface and the y axis normal to it is considered. The call u the tangential copmonent of the velocity vector and v the normal one.

With the assumption that $Re \to +\infty$, the boundary layer equations become[3]:

$$\frac{\partial}{\partial x}\left(\rho u R\right) + \frac{\partial}{\partial y}\left(\rho v R\right) = 0 \tag{2.182}$$

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{dp_e}{dx} + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) \tag{2.183}$$

$$\frac{\partial p}{\partial y} = 0 \tag{2.184}$$

$$\rho u \frac{\partial h}{\partial x} + \rho v \frac{\partial h}{\partial y} = u \frac{dp}{dx} + \mu \left(\frac{\partial u}{\partial y}\right)^2 - \frac{\partial q^y}{\partial y} \tag{2.185}$$

where $R = R(x)$ is the radius of the body and q_y is the heat flux in the y direction. These equations will be the starting point for the heat flux computation, and their derivation is provided in Appendix A.

We should also recall the following assumptions for the Boundary Layer theory:

- This equations are strictly valid only if the radius of curvature of the body is much larger than the boundary layer thickness, i.e. the boundary layer is relative thin, i.e. Re is relatively large. This means: $\delta \ll R$.
- $\frac{dp}{dx} = -\rho_e u_e \frac{du_e}{dx}$, i.e. the pressure gradient in the tangential direction is given by the momentum equation for the flow outside the boundary layer, the Eulerian flow.

- On the body surface, we assume a no-slip condition and no permeability for the velocity and a no-slip condtion for the temperature, so: $u_w = 0$, $T_w = T_w^{\text{body}}$.
- On the outer edge of the boundary layer, the boundary conditions are given by the Eulerian flow outside the boundary layer, so: $u_\infty = u_e$, $h_\infty = h_e$ (or $T_\infty = T_e$) and $P_\infty = P_e$.
- We did not include other conservation equations, like the species and charge conservation. This is because we assumed that the flow is in local chemical equilibrium, so the concentration of the species is fully determined by the temperature and pressure. We have 3 equations for the 3 unknowns, u, v and h (or T).

The reader can look at the Appendix A in order to understand why these assumptions are needed.

2.4 Stagnation heat flux computation

As we will see in the next chapter, we are interested in the computation of the stagnation heat flux inside the boundary layer. In this section I will start from the boundary layer equations and I will show how to compute the stagnation heat flux.

2.4.1 Definition of the heat flux

In general, there could be different types of heat transfers in a fluid flow, like conduction, convection, radiation and catalysis. In this work, I will neglect the radiation heat transfer, and I will compute the heat flux using the **Fourier law of heat conduction**:

$$\mathbf{q} = -\lambda \nabla T \tag{2.186}$$

But which λ should we use? Three limiting solutions can be of interest:

The **frozen boundary layer on a non-catalytic wall** is the simplest problem to treat: there are no chemical reactions taking place in the boundary layer or on the wall surface. Consequently, the concentration of species remains constant and, if thermal diffusion is neglected, there are no diffusion phenomena at all in the boundary layer. The heat flux on the surface is only due to heat conduction at the wall:

$$\mathbf{q} = -\lambda_{fr} \nabla T \tag{2.187}$$

For all practical purposes, the wall is colder than the flow outside the boundary layer: either the wall is water-cooled (heat flux probes) or it reaches an equilibrium temperature for which heat flux balance is achieved. A catalytic wall therefore imposes recombination of ions and atoms in the vicinity of the wall. Hence, in a frozen boundary layer on a catalytic wall, there is no chemistry source term in the flow, but there are chemical reactions near the wall and the species concentration varies across the boundary layer, reaching a steady-state distribution in accordance to diffusion effects due to concentration gradients. The species continuity equation has to be solved and the heat flux at the wall is made out of conduction and diffusion:

$$\mathbf{q} = -\lambda_{fr} \nabla T + \sum_{i=1}^{N_s} \mathbf{J}_i h_i \tag{2.188}$$

The third case is the equilibrium boundary layer, in which the chemical composition is imposed by the temperature and pressure profile across the boundary layer. In that limiting solution, the wall catalycity does not play any role. Diffusion effects are negiglible In the boundary layer but not near the wall, since diffusion of chemical energy adds to the heat flux just like in the case of the frozen boundary layer with catalytic wall. However, this effect can be taken into account, as we already seen, by means of the equilibrium thermal conductivity:

$$\mathbf{q} = -\lambda_{eq} \nabla T \tag{2.189}$$

We should note at this point that there is no much difference in species concentration near the wall between an equilibrium boundary layer and a frozen boundary layer with a fully catalytic wall. The values of heat flux in both cases will then be almost identical[4]. In the present work, we are mainly interested in the heat flux rate on a fully catalytic wall, namely the cold copper wall of the heat flux probe. The simplifying assumption of equilibrium boundary layer then seeems reasonable.

2.4.2 Change of variables

For the sake of clarity, let us repeat here the governing equations with the new considerations about the heat flux:

$$\frac{\partial}{\partial x}(\rho u R) + \frac{\partial}{\partial y}(\rho v R) = 0 \tag{2.190}$$

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{dp}{dx} + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) \tag{2.191}$$

$$\rho u \frac{\partial h}{\partial x} + \rho v \frac{\partial h}{\partial y} = u\frac{dp}{dx} + \mu\left(\frac{\partial u}{\partial y}\right)^2 + \frac{\partial}{\partial y}\left(\lambda_{eq}\frac{\partial T}{\partial y}\right) \tag{2.192}$$

I will now use a combination of the **Mangler transformation**, in order to transform an axisymmetric flow into an equivalent 2D flow, and the **Howard, Illingworth, Stewartson transformations**, in order to transform compressible flow to equivalent incompressible boundary layer equations. The **Levy-Lees cahgen of variable** is given by[4]:

$$\xi(x) = \int_0^x \rho_e \mu_e u_e R^2 dx' \tag{2.193}$$

$$\eta(x,y) = \frac{R u_e}{\sqrt{2\xi}} \int_0^y \rho dy' \tag{2.194}$$

We introduce the following adimensional functions:

$$F = \frac{u}{u_e} \tag{2.195}$$

$$g = \frac{T}{T_e} \tag{2.196}$$

$$V = \frac{2\xi}{\frac{\partial \xi}{\partial x}}\left(F\frac{\partial \eta}{\partial x} + \frac{\rho v R}{\sqrt{2\xi}}\right) \tag{2.197}$$

For the continuity equation, we get:

$$2\xi\frac{\partial F}{\partial \xi} + \frac{\partial V}{\partial \eta} + F = 0 \tag{2.198}$$

For the momentum equation, we get:

$$2\xi F\frac{\partial F}{\partial \xi} + V\frac{\partial F}{\partial \eta} = \alpha\left(\frac{\rho_e}{\rho} - F^2\right) + \frac{\partial}{\partial \eta}\left(l_0\frac{\partial F}{\partial \eta}\right) \tag{2.199}$$

where:

$$\alpha = \frac{2\xi}{u_e}\frac{du_e}{d\xi} \tag{2.200}$$

is a dimensionless parameter which takes into account the pressure gradient in the streamwise direction, and it is equal to $\frac{1}{2}$ for an axisymmetric body.

The term l_0 is the dimensionless Chapman-Rubesin parameter:

$$l_0 = \frac{\rho\mu}{\rho_e\mu_e} \tag{2.201}$$

For the energy equation, we get:

$$2\xi\, F\frac{\partial g}{\partial \xi} + V\frac{\partial g}{\partial \eta} = \frac{1}{C_p}\frac{\partial}{\partial \eta}\left(\chi\frac{\partial g}{\partial \eta}\right) + E\left(l_0\left(\frac{\partial F}{\partial \eta}\right)^2 + \alpha F\left(\rho_e\frac{\partial h}{\partial p} - \frac{\rho_e}{\rho}\right)\right) - 2\xi\frac{F}{T_e}g\frac{\partial T_e}{\partial \xi} \tag{2.202}$$

where:

$$E = \frac{u_e^2}{c_p T_e} \tag{2.203}$$

and:

$$\chi = \frac{\lambda}{\rho_e\mu_e} \tag{2.204}$$

Let's remember that all the quantities are to be considered at equilibrium conditions, and the subscript e indicates the edge of the boundary layer. The derivation of equations (2.198) and (2.199) and (2.202) is provided in Appendix B.

Since that we are in the stagnation point, $\xi = 0$, and the equations simplify to:

$$\frac{\partial V}{\partial \eta} + F = 0 \tag{2.205}$$

$$V\frac{\partial F}{\partial \eta} = \frac{1}{2}\left(\frac{\rho_e}{\rho} - F^2\right) + \frac{\partial}{\partial \eta}\left(l_0 \frac{\partial F}{\partial \eta}\right) \tag{2.206}$$

$$V\frac{\partial g}{\partial \eta} = \frac{1}{C_p}\frac{\partial}{\partial \eta}\left(\chi \frac{\partial g}{\partial \eta}\right) \tag{2.207}$$

with the boundary conditions:

$$\begin{cases} F(\eta = 0) = 0 \\ F(\eta \to +\infty) = 1 \\ g(\eta = 0) = \frac{T_w}{T_e} \\ g(\eta \to +\infty) = 1 \\ V(\eta = 0) = 0 \end{cases} \tag{2.208}$$

2.4.3 Actual heat flux computation

With the new variables, the heat flux on the wall can be computed as:

$$q_w = \sqrt{\frac{2\beta}{\rho_e \mu_e}} T_e \rho_w \lambda_w \left(\frac{\partial g}{\partial \eta}\right)_{\eta=0} \tag{2.209}$$

where β is the velocity gradient at the stagnation point:

$$\beta = \left(\frac{du_e}{dx}\right)_{x=0} \tag{2.210}$$

We can now solve the system of equation in order to find F,g and V, and then compute $\frac{\partial g}{\partial \eta}$ in order to find the heat flux on the wall. The whole derivation of these equations is provided in Appendix B.

The actual procedure to solve the equations will be described in the next chapters, where the code implementation will be discussed.

2.4.4 The velocity gradient

The stagnation point velocity gradient is defined as:

$$\beta = \left(\frac{du_e}{dx}\right)_{x=0} \tag{2.211}$$

and it is very important on the computation of the heat flux on the wall.

This term can be hard to evaluate, but for the subsonic case there are some approximations that can be used.

In particular the option implemented in the code is for the stagnation point velocity gradient in the subsonic ESA sample holder case (flat face probe perpendicular to the flow), and it was found by Kolesnikov[8]:

$$\frac{\beta R_m}{u} = \frac{1}{2 - \sigma - 1.68(\sigma - 1)^2 - 1.28(\sigma - 1)^3} \quad \text{if} \quad \sigma \leq 1 \tag{2.212}$$

$$\frac{\beta R_m}{u} = \sigma \quad \text{if} \quad \sigma > 1 \tag{2.213}$$

with:

$$\sigma = \frac{R_m}{R_j} \tag{2.214}$$

where R_j is the radius of the jet, R_m is the radius of the probe and u is the freestream velocity.

This formula is an approximation for the stagnation point velocity gradient, and it is valid for the subsonic case.

Other informations regarding the velocity gradient can be found in Appendix C.

Chapter 3

Resolution of the problem

3.1 Measurament of the input quantities

As I said before, the inputs of the problem are:

- The static (freestream) pressure, p^o.
- The stagnation pressure, p_t^o.
- The stagnation heat flux, q_o.
- The probe wall temperature T_w.
- The probe external radius, R_m.

I will now describe the problems encountered with total pressure measuraments, since that they are relevant for the solution of the problem, while I will assume that the other quantities are known.

3.1.1 Total pressure measurament

The measured pressure at the stagnation point is, by definition, the *stagnation pressure* on the probe, also called *Pitot pressure*, since that we measure it with a Pitot probe.

As I explained, the *stagnation pressure* is equal to the *total pressure* of the flow, p_t, only if the flow deceleration is isentropic, i.e. the flow is adiabatic and inviscid. In our model, this is true outside the boundary layer. But when we actually measure the stagnation pressure, we are measuring the pressure on the probe, which is clearly not obtained from an isentropic flow, since that, close to the probe, we have a deviation from the isentropic flow due to:

- **Flow chemistry**: If the flow is not truly in chemical equilibrium, then the reactions produce entropy and the deceleration cannot be considered isentropic. Hovever, if the flow remains subsonic, this effect will be small in the stagnation point, since that the flow is very close the the equilibrium.
- **Viscous effects**: As in low-temperature flows, a low value of the Reynolds number causes the Pitot pressure to be higher than the total pressure of the flow. This effect is known as **Barker's effect** and I am going to analyze it later.
- **High temperature effects**: The deceleration of the flow around the Pitot head cannot be considered adiabatic, due to an important stagnation-point heat flux from the plasma to the water-cooled probe (inside the boundary layer).ù

It is customary to add compressibility corrections to the above list, but it is actually irrelevant to do so in the present case, since that compressibility corrections are needed only when velocity has to be deducted from the Pitot measurament by using the incompressible Bernoulli equation, while I will actually solve the system with the total conditions of the flow.

The effects of heat transfer and high viscosity are now discussed in the following sections.

3.1.2 Effect of non-adiabatic deceleration

Considering the stream tube leading to the Pitot orifice diameter, non-adiabatic effects come in two aspects: heat transfer to the probe and dissipation of momentum into heat during the deceleration of the flow.

The first effect does not affect the total rpessure, as show by the compressible boundary layer equations, since that we found that $\frac{\partial P}{\partial y} = 0$ in the boundary layer. Hence, the pressure at the Pitot orifice is equal to the pressure outside the boundary layer, which is the stagnation pressure of the flow.

The dissipation of momentum into heat is due to the work performed by the viscous stresses. The relevance of this effect is linked to the magnitude of the **Brinkman number** of the flow, defined as:

$$Br = \frac{\mu u^2}{\lambda \Delta T} \tag{3.1}$$

This number is small. being of the order of ΔT^{-1} when reasonable values are used for the parameters [4]. The viscous dissipation of momentum into heat is hence negiglibe.

3.1.3 The Barker effect

At very low Reynolds numbers, it is no longer possible to ignore the effects of viscosity. The radial velocity gradient close to the stagnation point causes momentum to be transferred to the stagnation streamline from the neighbouring streamlines. This leads to an increase of total pressure in the stagnation point with respect to the ideal, isentropic, deceleration case.

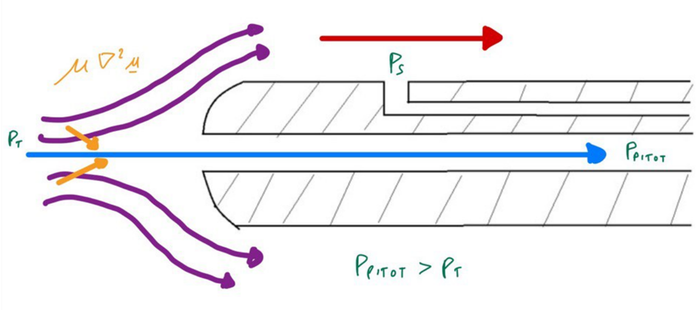

Figure 3.1: Graphical visualiation of the Barker effect

This effect appears for Reynold numbers lower than 1000, based of the external Pitot tube diameter, but only becomes significant under Re < 100. The increase of the Pitot pressure is expressible as:

$$P_{\text{stag}} = P_t + \Delta P_{\text{Barker}} \tag{3.2}$$

Usually the pressure increment is expressed as a function of an adimensional parameter C_p and the dynamic pressure of the flow close to the stagnation point:

$$C_p = \frac{P_{\text{stag}} - p}{\frac{1}{2}\rho u^2} \tag{3.3}$$

So, the Barker effect is expressed as:

$$P_{\text{stag}} = P_t + \frac{1}{2}\rho u^2 (C_p - 1) \tag{3.4}$$

C_p is a function of the Reynolds number of the flow:

$$C_p = C_p(Re) \tag{3.5}$$

Clearly, the pressure coefficient C_p depends on the Pitot probe shape:

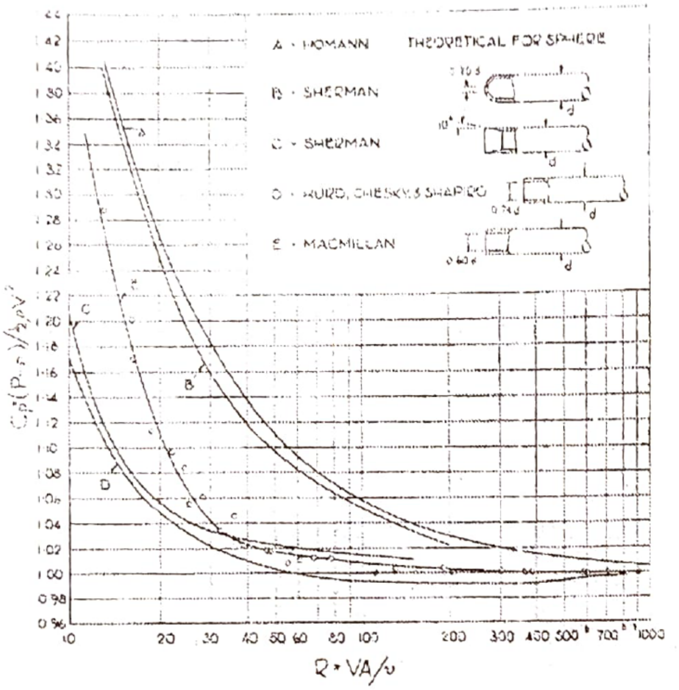

Figure 3.2: Pressure coefficient C_p as a function of the Reynolds number

For our case of interest, the Pitot Probe B, which has a round nose, is the most suitable and yields data of interest. The geometry is not exactly the same howerver, since Sherman used source-shaped probes but no hemispherical probes. Nevertheless, he obtained results which could be well fitted with Homann's theoretical calculations for a sphere.

The expression obtained by Homann almost exactly fits teh data labelled B. His expression is:

$$C_p = 1 + \frac{6}{Re + 0.455 Re^{0.5}} \tag{3.6}$$

where the Reynold number is expressed in terms of the Pitot probe nose radius R_p:

$$Re = \frac{\rho u R_p}{\mu} \tag{3.7}$$

Carleton propeses another approximation, checked in argon plasma flows, and in which Re must be computed at the temperature corresponding to the mean enthalpy of the boundary layer:

$$C_p = 1 + \frac{8}{Re + 0.5576 Re^{0.5}} \tag{3.8}$$

3.2 Solution withoout the Barker effect

The most efficient technique to solve the problem is obtained when the equations needed to compute total conditions are solved simultaneously with the heat flux equation, which can be seen as a single function of three paramenters:

$$q_w = q_w(p_t, T_t, u) \tag{3.9}$$

when the wall temperature T_w is given.

We can now write the following system of equations:

$$\begin{cases} q_w(p_t^o, T_t, u) - q^o = 0 \\ h_t(p_t^o, T_t) - [h(p^o, T) + 0.5u^2] = 0 \\ s_t(p_t^o, T_t) - s(p^o, T) = 0 \end{cases} \tag{3.10}$$

The system has 3 unknowns, T, T_t and u, and 3 equations, and can be solved using a Newton-Raphson iterative procedure.

3.3 Solution with the Barker effect

When the Barker effect is taken into account, the total pressure of the flow becomes an unknown of the system of equations. By including the Barker effect expression in the system, we get:

$$\begin{cases} q_w(p_t, T_t, u) - q^o = 0 \\ h_t(p_t, T_t) - [h(p^o, T) + 0.5u^2] = 0 \\ s_t(p_t, T_t) - s(p^o, T) = 0 \\ p^o_{\text{stag}} - p_{stag}(p^o, T, u, p_t) \end{cases} \tag{3.11}$$

The system has 4 unknowns, T, T_t, u and p_t, and 4 equations, and can be solved using a Newton-Raphson iterative procedure.

3.4 Jacobian matrix

The Jacobian matrix of the system without the Barker's effect is:

$$J = \begin{bmatrix} 0 & \frac{\partial q_w}{\partial u} & \frac{\partial q_w}{\partial T_t} \\ -\frac{\partial h}{\partial T} & -u & \frac{\partial h_t}{\partial T_t} \\ -\frac{\partial s}{\partial T} & 0 & \frac{\partial s_t}{\partial T_t} \end{bmatrix} \tag{3.12}$$

The Jacobian matrix of the system with the Barker's effect is:

$$J = \begin{bmatrix} 0 & \frac{\partial q_w}{\partial u} & \frac{\partial q_w}{\partial T_t} & \frac{\partial q_w}{\partial p_t} \\ -\frac{\partial h}{\partial T} & -u & \frac{\partial h_t}{\partial T_t} & \frac{\partial h_t}{\partial p_t} \\ -\frac{\partial s}{\partial T} & 0 & \frac{\partial s_t}{\partial T_t} & \frac{\partial s_t}{\partial p_t} \\ \frac{\partial p_{\text{stag}}}{\partial T} & \frac{\partial p_{\text{stag}}}{\partial u} & 0 & \frac{\partial p_{\text{stag}}}{\partial p_t} \end{bmatrix} \tag{3.13}$$

3.5 Output quantities

The output quantities of the problem are:

- The static temperature, T.
- The total temperature, T_t.
- The total pressure, p_t.
- The velocity, u.
- The enthalpy, h.
- The total enthalpy, h_t.
- The entropy, s.
- The total entropy, s_t.
- The Mach number, M.
- The Reynolds number, Re.

After solving the system of equations (with or without the Barker effect), we can compute the output quantities easily, since that the flow state is completely determined.

Chapter 4

How to use the software

4.1 Prerequisites

4.1.1 Work environment

I will start with a fresh installation of Ubuntu 20.04 LTS. In particular, I will use WSL (Windows Subsystem for Linux) to run Ubuntu on Windows.

The program can be run in 2 ways:

1 By running the Python script *main.py* in Python. This allows customization, but requires the installation of the dependencies.

2 By running the binary *PlasFlowSolver* in the terminal. This is very straightforward and does not require the installation of the dependencies. However, this does not allow customization.

4.1.2 Requirements

The main requirements are indicated in the *requirements.txt* file.
The following is a thorough list from a fresh installation of Ubuntu 20.04 LTS.

1 git:

```
sudo apt-get install git
```

Probably the requirement is already satisfied from the linux distribution.

2 Python:

```
sudo apt-get install python3-dev
```

Probably the requirement is already satisfied from the linux distribution, but it could be old and we need to 'dev' version.
The code was written in Python3.10.8, so we need at least Python3.10 to run it. Check your version with

```
python3 --version
```

You can install the latest version of Python3 with:

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
sudo apt-get install python3.10-dev
```

You can now check the version with:

```
python3.10 --version
```

3 PATH:
If you get a warning about your 'directory' not being in 'PATH', please do the following:

```
export PATH = "<directory>:$PATH"
```

and restart your shell.

4 pip3:

```
curl -sS https://bootstrap.pypa.io/get-pip.py | python3.10
```

You can now check the version with

```
python3.10 -m pip --version
```

5 numpy:

```
python3.10 -m pip install numpy
```

6 testresources:

```
python3.10 -m install testresources
```

7 wheel:

```
python3.10 -m pip install --upgrade pip setuptools wheel
```

8 pybind11:

```
python3.10 $-$m pip install pybind11
```

9 cmake:

```
sudo apt-get install cmake
```

10 mutationpp:
This is the trickiest package to install, since that we need the newer version from github, and we need to build it.
We go to GitHub page *https://github.com/mutationpp/Mutationpp.*
Follow the instructions to install the package. In particular, I would recommend to build it for C++ first, set up the environment variables and then build the Python package.
In particular, for the last step, navigate to the Mutationpp folder and run:

```
python3.10 -m pip install .
```

11 pandas:

```
python3.10 -m pip install pandas
```

12 scipy:

```
python3.10 -m pip install scipy
```

13 openpyxl:

```
python3.10 -m pip install openpyxl
```

14 h5py:

```
python3.10 -m pip install h5py
```

4.2 Installation

To install the software, after getting access to the GitHub repository, you can:

1 Download the zip file from Github, the link to the repository is `https://github.com/chess-uiuc/PlasFlow`.

2 Clone it with git:

```
git clone https://<your_username>:<your_token>@github.com/chess-uiuc/PlasFlow.git
```

Basing on your security settings for you GitHub account, you may not need a token and you could be able to simply do:

```
git clone https://github.com/chess-uiuc/PlasFlow.git
```

If you need a token, you can generate it from your account settings. In particular, developer options, personal access tokens, token (classic), generate new token (classic), tick the box for ”repo”.

At this point you can run the *main.py* script in Python (after installing all the dependencies). If you want to run the code without installing the dependencies, you can download the binaries from the release page on GitHub. At the moment, the latest release is v2.0.0.
You may need to give the permission to execute the file with:

```
sudo chmod +x PlasFlowSolver
```

A checksum is available with sha256 to check the integrity of the file.

4.3 Program inputs, outputs and settings

The program takes the following inputs:

- Measurements: the static pressure p, the stagnation pressure p_{stag} and the stagnation heat flux q.
- Heat flux probe properties: the wall temperature T_w, and the probe external radius R_m.
- Pitot probe properties: the Pitot probe external radius R_p.
- Initial conditions: the initial conditions of the flow for the Newton-Raphson method.
- Program settings: a list of the program settings, like the convergenze criteria (more details in the next sections).

These inputs are read from a file accordingly to the mode of the program (more details in the next sections).

The program outputs the following quantities:

- Static temperature T: the static temperature of the flow.
- Total temperature T_t: the total temperature of the flow.
- Total pressure p_t: the total pressure of the flow.
- Velocity u: the velocity of the flow.
- Enthalpy h: the enthalpy of the flow.
- Total enthalpy h_t: the total enthalpy of the flow.
- Entropy s: the entropy of the flow.
- Total entropy s_t: the total entropy of the flow.
- Mach number M: the Mach number of the flow.
- Reynolds number Re: the Reynolds number of the flow.
- Convergence status: a string that tells if the program has converged or not.
- Last residuals: the residuals of the last iteration of the Newton-Raphson method.

- Warnings: a string that tells if there are any warnings.
- Kn: the Knudsen number of the flow.
- mass fractions: the mass fractions of the flow.

The program outputs these quantities in a file accordingly to the mode of the program (more details in the next sections).

4.3.1 Input measurements and gas

In details, the measurements are the following:

- Comment: a comment that describes the current case.
- Static pressure p: the static pressure of the flow. It can be the chamber pressure or the static pressure from a Pitot probe. Default unit of measurament is kPa.
- Stagnation pressure *p_stag*: the stagnation pressure of the flow, read from the Pitot probe. The unit of measurament is kPa.
- Stagnation point cold-wall heat flux q: the stagnation heat flux of the flow read from the heat flux probe. The unit of measurament is W/cm^2.
- Plasma gas: the gas used during the test. It is needed to compute the properties of the gas. It has to be a mixture from the MUTATION++ library.

In some program modes it is possible to specify the dynamic pressure *p_dyn* instead of the static pressure p. The relationship between the two is:

$$p_{\text{stag}} = p + p_{\text{dyn}} \tag{4.1}$$

Please note that for the mixture files for the plasma gas, MUTATION++ will look in your environment variables for the path to the DATA folder of the MUTATION++ library, and THEN in the current directory and THEN in a subdirectory of the current directory. Some mixture files are already provided in the *mixtures* folder.
Please note that the equilibrium model should be used for this program. The same applies for the other MUTATION++ files, like The *mechanism* files (useless for the current program).

4.3.2 Heat flux and Pitot probe properties

The heat flux probe properties are the following:

- Wall temperature *T_w*: the wall temperature of the heat flux probe. It is assumed constant and given. The unit of measurament is K. It is needed to solve the stagnation heat flux computation (wall boundary condition).
- Probe external radius *R_m*: the external radius of the heat flux probe. It is assumed constant and given. The unit of measurament is *mm*. It is needed to compute the velocity gradient in front of the heat flux probe, in order to compute the stagnation heat flux.
- Jet radius *R_j*: the radius of the freestream jet. It is assumed constant and given. The unit of measurament is *mm*. It is needed to compute the velocity gradient in front of the heat flux probe, in order to compute the stagnation heat flux.
- Stagnation type *stag_type*: the type of the stagnation point. At the moment, only the *flat* option is available. It is needed to compute the velocity gradient in front of the heat flux probe, in order to compute the stagnation heat flux.
- Heat flux law *hf_law*: the law used to compute the heat flux. At the moment, only the *exact* option and the *fay_riddell* option are available. The *exact* option is the most accurate, but the *fay_riddell* option is faster. It is needed to compute the stagnation heat flux.

The Pitot probe properties are the following:

- Pitot probe external radius *R_p*: the external radius of the Pitot probe. It is assumed constant and given. The unit of measurament is *mm*. It is needed to compute the Barker effect in the stagnation pressure computation.
- Barker effect correction type *barker_type*: the type of the Barker effect correction. At the moment, the option supported are *none*, *homann* and *carleton*. The *none* option is used when the Barker effect is not considered. The *homann* option is used when the Homann expression for the Barker effect is used. The *carleton* option is used when the Carleton expression for the Barker effect is used.

4.3.3 Initial conditions

The Newton-Raphson method needs initial conditions to start the iteration. The initial conditions needed are the following:

- Static temperature *T_0*: the initial guess for the static temperature of the flow. The unit of measurament is K.
- Total temperature *T_t_0*: the initial guess for the total temperature of the flow. The unit of measurament is K.
- Velocity *u_0*: the initial guess for the velocity of the flow. The unit of measurament is m/s.
- Total pressure *p_t_0*: the initial guess for the total pressure of the flow. The unit of measurament is Pa. If the Barker effect is not considered, this has to be set to the stagnation pressure measurament. If the Barker effect is considered, and the value given is 0, the program will use the stagnation pressure measurament as initial guess.

Instead of manually specify the initial conditions, it is possible to extract them from an **Initial conditions database**, specified by a variable *ic_db_name*. In this case, only the name of the database is needed. The program will try to interpolate the previous runs to find the initial conditions. If this fails, extrapolation is tried and then nearest neighbour is used.

4.3.4 Program settings

At the moment, the program settings are the following:

- Number of points *N_p* the number of points for the η discretization of the boundary layer in the heat flux procedure.
- Maximum number of iterations *max_hf_iter* the maximum number of iterations for the heat flux procedure.
- Heat flux convergence criteria *hf_conv* the convergence criteria for the heat flux procedure. The normalized difference between iterations is used.
- Use previous iteration *use_prev_ite* this settings can be "yes" or "no". If it is "yes", the program will use the previous iteration of the heat flux procedure as initial guess for the current iteration if the heat flux procedure has already converged from a previous heat flux computation (this is valid only for the current case, not the previous ones).
- Eta max *eta_max* the maximum value of η for the boundary layer discretization in the heat flux procedure. Theoretically this value should be infinite, but in practice it is set to a large value.
- Log warning heat flux *log_warn_hf* this settings can be "yes" or "no". If it is "yes", the program will log a warning if the heat flux procedure does not converge. Otherwise, the program will not log a warning.
- Newton-Raphson convergence criteria *newton_conv* the convergence criteria for the Newton-Raphson method. The normalized residuals are used.
- Maximum number of Newton-Raphson iterations *max_newton_iter* the maximum number of iterations for the Newton-Raphson method.
- Jacobian difference *jac_diff* the difference used to compute the Jacobian matrix in the Newton-Raphson method.
- Minimum temperature relaxation *min_T_relax* the minimum value of the temperature relaxation parameter in the Newton-Raphson method.
- Maximum temperature relaxation *max_T_relax* the maximum value of the temperature relaxation parameter in the Newton-Raphson method.

4.3.5 Inputs and outputs files

The software can be run with 3 different I/O modes:

1 Single run mode (srun): in this mode, a single thermodynamic case is solved, by using a file with the extension *.srun* as input.

2 Excel mode (xlsx): in this mode, a series of thermodynamic cases are solved, by using an excel file with the extension *.xlsx* as input.

3 File mode (filerun): in this mode, a series of thermodynamic cases are solved, by using a file with the extension *.in* as input and a file with the extension *.pfs* as settings. This mode is a tribute to the old Plasprobes software.

Example files are provided in the **example_files** folder.
The software can be scripted by using a script file, *script.pfs*. More details later.

Single run mode

In this mode, the software is run with a single thermodynamic case, by using a single file with the extension *.srun* as input. The file **must** follow the following structure:

```
INPUTS:
comment = Example
P [kPa] = 6e0
P_dyn [kPa] = 1.8e-1
P_stag [kPa] =
q_target [W/cm^2] = 50.6e0
plasma_gas = air_11
INITIAL CONDITIONS:
ic_db_name = db_air_11.h5
T_0 [K] = 4000
T_t_0 [K] = 5000
u_0 [m/s] = 600
P_t_0 [kPa] = 0
PROBE PROPERTIES:
T_w [K] = 400
R_p [mm] = 4e0
R_m [mm] = 10.1e0
R_j [mm] = 50e0
stag_type = flat
hf_law = exact
barker_type = none
PROGRAM SETTINGS:
N_p = 500
max_hf_iter = 100
hf_conv = 1e-6
use_prev_ite = yes
eta_max = 8
log_warning_hf = yes
newton_conv = 1e-8
max_newton_iter = 30
jac_diff = 1e-2
min_T_relax [K] = 400
max_T_relax [K] = 18000
```

A file *example.srun* is already provided in the **example_files** folder as template.

It is important the the order of the inputs and the separator "=" are respected, since that the program reads the file line by line and splits the line by the separator. Also, a space before and after the separator is needed.

The user can decide to use the dynamic pressure *P_dyn* instead of the static pressure *P*. In this case, one of the two values must be left blank. If both values are given, the pressure must be consistent (1e-3 tollerance):

$$P_{\text{stag}} = P + P_{\text{dyn}} \tag{4.2}$$

For the initial conditions, if a database name *ic_db_name* is given, the program will extract the initial conditions from the database. If the database does not exist or is corrupted, the program will use the initial conditions specified in the file. If the Barker effect is not considered, the initial guess for the total pressure must be set to the stagnation pressure measurament, otherwise, the program will override the value with the stagnation pressure measurament. If the Barker effect is considered, and the value given is 0, the program will use the stagnation pressure measurament as initial guess.

The program will output a file with the extension *_out.srun* with the following structure:

```
has_converged_out: yes
rho_out: 6.461706826095365 g/m^3
T_out: 2984.4253840138012 K
h_out: 5031.776146194813 kJ/kg
u_out: 234.51004262158608 m/s
a_out: 1030.0548304536487 m/s
M_out: 0.2276675335023719
T_t_out: 2993.0232079343573 K
```

```
h_t_out: 5059.273626240002 kJ/kg
P_t_out: 6.18 kPa
Re_out: 126.43813239570865
Kn_out: 0.0006372461545838044
Species mass fraction composition:
e−: 1.7729892680267455e−12
N+: 9.0058705826078e−19
O+: 2.5267131981279414e−14
NO+: 9.694837986813029e−08
N2+: 4.938114748204303e−16
O2+: 3.025082674844172e−11
N: 2.3342420402008976e−05
O: 0.08825068024911226
NO: 0.035828340067088786
N2: 0.7503346078191594
O2: 0.12556293246380798
warnings_out: None
res_out: 4.2733011145389445e−10
```

The first line indicates if the program has converged or not. The following lines are the output quantities of the problem. The last line is the residuals of the last iteration of the Newton-Raphson method.
Thw warning line indicates if there are any warnings.

Excel mode

In this mode, the software is run with a series of thermodynamic cases, by using an excel file with the extension *.xlsx* as input. An example of input and output files is provided in the **example_files** folder.
The excel file **must** have 1 sheet with the following structure:

Table 4.1: Input Table

Inputs					
comment	P [kPa]	P_dyn [kPa]	P_stag [kPa]	q_target [W/cm2̂]	plasma_gas
Example case	6	0.2		100	N2

Table 4.2: Initial conditions table

Initial conditions				
ic_db_name	T_0 [K]	T_t_0 [K]	u_0 [m/s]	P_t_0 [kPa]
	3000	4000	200	0

Table 4.3: Probes settings table

Probe settings						
T_w [K]	R_p [mm]	R_m [mm]	R_j [mm]	stag_type	hf_law	barker_type
400	4.7	10	50	flat	exact	none

Table 4.4: Program settings table

Program settings										
N_p	max_hf_iter	hf_conv	use_prev_ite	eta_max	log_warning_hf	newton_conv	max_newton_iter	jac_diff	min_T_relax	max_T_relax
500	100	1.00E-04	yes	8	yes	1.00E-08	30	1.00E-02	400	20000

Please note that the file is case sensitive, so the names of the columns must be exactly as shown in the tables above. Each row is a different case.

The following rules apply:

- You can either use the dynamic pressure or the stagnation pressure, just leave blank the one you will not use. If you use both, they have to be consistent with each other (1e-3 tollerance).
- If an initial condition database is specified, the program will extract the initial conditions from the database. If the database does not exist, is corrupted or not specified, the program will use the initial conditions specified in the file.
- If an initial condition, probe property or setting is not found, it will be read from the file *std_value.pfs*. A warning will be logged in the output file if this happens. If the *std_value.pfs* file is not found (or is invalid), a default one is generated, but the user can freely modify it. This has been done so that the program can continue if some settings values are missing (by mistake or by choice).

The *std_value.pfs* file has the format:

```
plasma_gas = air_11
T_0 [K] = 400
T_t_0 [K] = 6000
u_0 [m/s] = 500
P_t_0 [kPa] = 0
T_w [K] = 400
R_p [mm] = 4.77e0
R_m [mm] = 10.16e0
R_j [mm] = 50e0
stag_type = 0
hf_law = 0
barker_type = 0
N_p = 400
max_hf_iter = 100
hf_conv = 1e-4
use_prev_ite = 1
eta_max = 6
log_warning_hf = 1
newton_conv = 1e-8
max_newton_iter = 30
jac_diff = 1e-3
min_T_relax [K] = 200
max_T_relax [K] = 18000
```

An example is provided in the **example_files** folder.
The output file is a file with the extension *_out.xlsx* with the same structure as the input file but new columns with the outputs.

File mode

In this mode, the software is run with a series of thermodynamic cases, by using a file with the extension *.in* as input and a file with the extension *.pfs* as settings. The input file only contains the measurements, while the settings file contains all the settings. The input file **must** follow the following structure:

- 20 characters for the comment
- 10 characters for the static pressure
- 20 characters for the dynamic pressure
- 20 characters for the stagnation heat flux

Each row is a case. A file *example_in.in* is already provided in the **example_files** folder as template. If one of the case (line in the input file) is not valid, the program will log a warning and skip the case.

The settings file **must** have the following format:

```
INITIAL CONDITIONS:
ic_db_name = db_air_11.h5
T_0 [K] = 3000
T_t_0 [K] = 4000
u_0 [m/s] = 300
P_t_0 [kPa] = 0
PROBES PROPERTIES:
T_w [K] = 400
R_p [mm] = 4e0
R_m [mm] = 10.1e0
R_j [mm] = 50e0
stagtype = flat
hf_law = exact
barker_type = none
PROGRAMS SETTINGS:
mixture_name = air_11
N_p = 251
max_hf_iter = 100
hf_conv = 1e-4
use_prev_ite = yes
eta_max = 6
log_warning_hf = yes
newton_conv = 1e-8
max_newton_iter = 30
jac_diff = 1e-2
min_T_relax [K] = 200
max_T_relax [K] = 18000
```

The structure is exactly the same as the single run mode, but the inputs are separated in different file. A file *example_settings.pfs* is already provided in the **example_files** folder as template.

It is important the the order of the inputs and the separator ”=” are respected, since that the program reads the file line by line and splits the line by the separator.

For the initial conditions, if a database name *ic_db_name* is given, the program will extract the initial conditions from the database. If the database does not exist or is corrupted, the program will use the initial conditions specified in the file. If the Barker effect is not considered, the initial guess for the total pressure must be set to the stagnation pressure measurement, otherwise, the program will override the value with the stagnation pressure measurement. If the Barker effect is considered, and the value given is 0, the program will use the stagnation pressure measurement as initial guess.

The output file is a file with the extension .out that contains the outputs of the software.

If a case is not valid, the line will be preceded by the string ”INVALID CASE” and the results will be ”-1”. If a case does not converge, the line will be preceded by the string ”NO CONVERGENCE”, with the results of the last iteration of the Newton-Raphson method.

Script file

The program can be automatized by using a script file, *script.pfs*.
The file has the following structure:

```
Mode: xlsx
File: excel_example.xlsx
Settings: settings.pfssx
```

If the program detect a valid *script.pfs* file, then the specified mode is executed with the specified input file/files (if the mode uses a settings file).
If one of the file is not found or invalid, the program will prompt the user to choose a new mode, input or setting file.

4.4 How to use the software

If a script file is detected, the program will simply run the mode and files specified.
If a script file is not detected or is invalid, the program will follow the following steps:

- Prompt the user for the program mode.

- Prompt the user for the input file.
- Prompt the user for the settings file (if the mode requires it).
- The program will start.

For each case, the program will show in the console the input conditions and the residual for each iteration of the Newton-Raphson loop, until convergence is reached or the maximum number of iteration is reached. The program will then terminate.

After the program has finished, the outputs will be written in the output file.

4.5 Database files

It is possible to use a database to store the program runs, and use them to build an initial conditions database. In particular, the program will look for the file *database_settings.pfs* with the following structure:

```
Database name: example_db.csv
Create database if not found: true
Update if lower time: true
Generate ic map: true
ic map name (Do not include the .h5): db_example
ic map mixture split: false
```

These settings are, respectively:

- Database name: this is the name of the database file which will store the program results for the next runs.
- Create database if not found: this is a boolean that tells the program if a database should be created if a previous one is not found.
- Update if lower time: if this flag is false, the program will not store the results of a case if it is already present in the database. If the flag is true, the program will measure the time of the case and store it only if the time is lower than the one in the database.
- Generate ic map: this is a boolean that tells the program if an initial conditions map should be generated at the end of the program, using the database (if previous cases are present in the database, they will be used too).
- ic map name: the name of the initial conditions map file.
- ic map mixture split: this is a boolean that tells the program if the initial conditions map should be split by mixture. If this flag is true, the program will generate a map for each mixture in the database, otherwise, it will generate a single map.

If the *database_settings.pfs* file is not found or is invalid, the program will not use the database.

4.6 Runner script

A runner script is provided, *runner.py*, in order to run, in sequence, different xlsx files. The script simply edits the *script.pfs* file and runs the program.

4.7 Final remarks

The code tries to handle all the possible exceptions, and it will try to not crash if one of the cases if invalid or an error occurs, so that the other cases can be run.
Please pay attention to your relaxation temperatures, since a too low or too high value could lead Mutation++ to hang; in this case, the program has to be manually killed.

Chapter 5

Code verification

To code was verified by comparison with an old version of the code in Fortran, made by B. Bottin. The old code was using PEGASE, so it has to be modified to use the MUTATION++ interface, in order to be able to compare the results. The code was tested with and without the Barker effect, and the results were consistent with the old code, with an error of the order of the machine precision.

At the moment, this verification is retained sufficient, since that it proves that the two codes are identical if the same thermodynamic library is used.

The differences between the thermodynamic libraries are explained briefly in the next sections, but mainly they are due to:

- Different collision integrals.
- Different implementation of the transport properties.
- Different cutting criteria for the computation of the electronic energy modes.
- Different implementation of the energy, enthalpy and entropy computation.

Since that both PESAGE and MUTATION++ are made from the research group VKI, but PESAGE is very old, it is reasonable to assume that MUTATION++ olds the most accurate implementation of the thermodynamic properties [13].

5.1 Thermodynamic library comparison

MAYBE TO DO A LITTLE BIT

5.2 Heat flux computation difference

The code were compared by using the same heat flux implementation, but during my derivation of the boundary layer equations with the Levy-Lees transformation (see Appendix B) I found that the term containg u_e^2 was to be set to 0, since that it represent the tangential velocity of the flow at the edge of the boundary layer, and in the stagnation point, this term goes to 0.

Before doing further investigation, it is worth to mention that on the GitHub repository of the code the user can find the heat flux implementation with $u_e = 0$ in the **master** branch, and the heat flux implementation with $u_e \neq 0$ in the **old_bottin_version** branch. The two branches are identical, except for the heat flux implementation.

In my opinion, and after studying the theory further, I strongly believe that the heat flux implementation with $u_e = 0$ is the correct one, since that close to the stagnation point:

$$u_e = \left(\frac{du_e}{d\xi}\right)_{\xi=0} \xi + O(\xi) \tag{5.1}$$

and for $\xi \to 0$, the term $u_e \to 0$, which is the case of interest.

The heat flux will still depend on the freestream velocity, but only for the velocity gradient in front of the heat flux probe.

As professor Barbante made my notice, the term in the differential equation with u_e is dependent on $\frac{u_e^2}{C_p T_e}$, which for a high enthalpy subsonic flow is very small, since that $u_e \ll C_p T_e$. This could be the reason why B. Bottin did not spot the error in the heat flux implementation. In his thesis, the heat flux computation was tested with a case with $u_e = 150m/s$, very small, which could explain why the error was not spotted.

Furthermore, in the code NEBOULA, in the test cases for the stagnation point, I notice that the heat flux was computed with $u_e = 0$.

Chapter 6

Code validation

A validation with experimental data has not been performed yet, since that the code is still in development.

Chapter 7

Future improvements

Some of the future improvements might be:

- Implement new Barker effect corrections.
- Implement new heat flux laws.
- Implement new velocity gradient laws.
- Improve the derivative order for the Jacobian matrix computation.

References

[1] J.D.Jr. Anderson. *Hypersonic and high temperature gas dynamics*. McGraw-Hill Book Company, New York, 1995.

[2] J. Applequist. *Introduction to Plasmatron X*. URL: `https://chess.grainger.illinois.edu/facilities/plasmatron-x` (visited on 03/18/2024).

[3] P.F. Barbante. "Accurate and Efficient Modelling of High Temperature Nonequilibrium Air Flows". PhD thesis. Von Karman Institute for Fluid Dynamics, Aeronautics/Aerospace department.

[4] B. Bottin. "Aerothermodynamic Model of an Inductively Coupled Plasma Wind Tunnel, Numerical and Expermental Determination of the Facility Performance". PhD thesis. Université de Liège, Facilté des Sciences Appliquées, Von Karman Institute for Fluid Dynamics, Aeronautics/Aerospace department, Oct. 1999.

[5] G. Bottin B. & Degrez. "Aerothermodynamic model of an inductively coupled plasma wind tunnel". In: *Journal of Thermophysics and Heat Transfer* 14.1 (Jan. 2000), pp. 126–135.

[6] *Differential operators in three dimesions*. URL: `https://en.m.wikipedia.org/wiki/Orthogonal_coordinates#Differential_operators_in_three_dimensions` (visited on 04/04/2024).

[7] G. Hermann S. & Klaus. *Boundary Layer Theory*. Ninth. Springer, 2017.

[8] A. F. Kolesnikow. "CONDITIONS OF SIMULATION OF STAGNATION POINT HEAT TRANSFER FROM A HIGH-ENTHALPY FLOW". In: *UDC 533.6.011.8* ().

[9] F. Panerai. *AE 480: Hypersonic Aerothermodynamics - Course Notes*.

[10] N. Peters. "Solution of the Boundary Layer Equations for Chemically Reacting Gases Using a Multi-Point Method". In: *German Aerospace Center, Research Report 72-58* (1972).

[11] Ira M. Cohen & David R. Dowling Pijush K. Kundu. *Fluid Mechanics*. Sixth. 2016.

[12] G.S.R. Sarma. "General introduction in Aerothermochemistry for hypersonic technology, VKI Lecture Series 1995-04". In: (Apr. 1995).

[13] M. James B. Scossings. "Development of numerical methods and study of coupled flow, radiation, and ablation phenomena for atmospheric entry". PhD thesis. Université Paris-Saclay, prepared at the Von Karman Institute for Fluid Dynamics, Aeronautics and Aerospace department, Sept. 2017.

[14] G. Vanden Abeele D. & Degrez. "An efficient computational model for inductive plasma flows". In: (June 1998).

Appendices

Appendix A

Derivation of the boundary layer equations for an axisymmetric flow in natural coordinates

In this appendix I am going to derive the boundary layer equations showed in Eq. **??** for an axisymmetric flow in natural coordinates. I decided to include this derivation, even if it is a little bit raw, because I could not find any explicit derivation online for this particular set of coordinates, since that all the authors were just using the equations without showing the derivation.

A.1 Natural coordinates for a body in axisymmetric flow

The natural coordinates for an axisymmetric body are defined as:

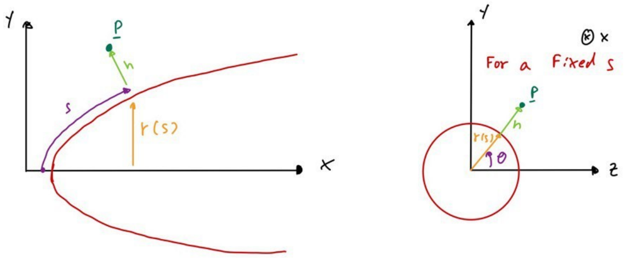

Figure A.1: Natural coordinate system in axisymmetric flows.

Taken (x, y, z) as the cartesian coordinates, with x along the revolution axis of the axisymmetric body, we define the natural coordinates as:

- s: the arc length along the body surface, with $s = 0$ at the stagnation point.
- n: the normal distance from the body surface for a certain s, with $n = 0$ on the body surface.
- θ: the angle between the z axis and the normal vector projected in the $z - y$ plane.

$r = r(s)$ is the radius of the axisymmetric body for a certain s, and **P** is the position vector of a point on the body surface:

$$\mathbf{P} = x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z \tag{A.1}$$

We can define the basis vector of these coordinates as the tangent vectors of the curves obtained by varying one coordinate, keeping the others fixed[6]:

$$\mathbf{e}_i = \frac{\partial \mathbf{P}}{\partial q^i} \tag{A.2}$$

where q^i are the natural coordinates, $q^i = [s, n, \theta]$.

Note that the vectors are not necessarily of equal length. The useful functions known as scale factors of the coordinates, h_i are simply the lengths of the basis vectors. The scale factors are sometimes called *Lamé coefficients*. The normalized basis vectors are notated with a hat and obtained by dividing by the length:

$$\hat{\mathbf{e}}_i = \frac{\mathbf{e}_i}{h_i} \tag{A.3}$$

In particular, I will call the normalized basis vectors $\hat{\mathbf{e}}_r$, $\hat{\mathbf{e}}_n$ and $\hat{\mathbf{e}}_\theta$.

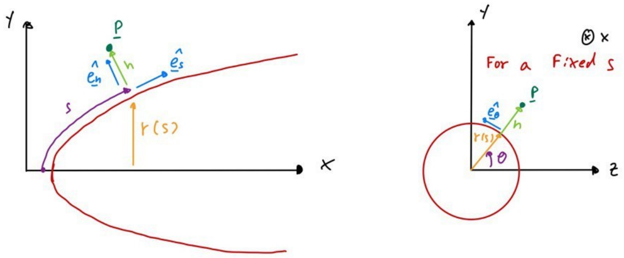

Figure A.2: Natural coordinate system in axisymmetric flows.

A.2 Del operator in natural coordinates

In order to write the boundary layer equations in natural coordinates, we need to express the del operator in these coordinates. The del operator in cartesian coordinates is defined as:

$$\nabla = \mathbf{e}_x \frac{\partial}{\partial x} + \mathbf{e}_y \frac{\partial}{\partial y} + \mathbf{e}_z \frac{\partial}{\partial z} \tag{A.4}$$

We can express the infinitesimal displacement vector in terms of the natural coordinates as:

$$d\mathbf{P} = \frac{\partial \mathbf{P}}{\partial s} ds + \frac{\partial \mathbf{P}}{\partial n} dn + \frac{\partial \mathbf{P}}{\partial \theta} d\theta = \mathbf{e}_s ds + \mathbf{e}_n dn + \mathbf{e}_\theta d\theta \tag{A.5}$$

By definition of the gradient of a function:

$$df = \nabla f \cdot d\mathbf{P} = \nabla f \cdot \mathbf{e}_s ds + \nabla f \cdot \mathbf{e}_n dn + \nabla f \cdot \mathbf{e}_\theta d\theta \tag{A.6}$$

where this hold true even if f is a tensor. From this, it follows that ∇ has to be[6]:

$$\nabla = \frac{\mathbf{e}_s}{h_s^2} \frac{\partial}{\partial s} + \frac{\mathbf{e}_n}{h_n^2} \frac{\partial}{\partial n} + \frac{\mathbf{e}_\theta}{h_\theta^2} \frac{\partial}{\partial \theta} = \frac{\hat{\mathbf{e}}_s}{h_s} \frac{\partial}{\partial s} + \frac{\hat{\mathbf{e}}_n}{h_n} \frac{\partial}{\partial n} + \frac{\hat{\mathbf{e}}_\theta}{h_\theta} \frac{\partial}{\partial \theta} \tag{A.7}$$

So, the hard part is to find the scale factors h_i. It is clear that from their definition:

$$h_i = |\frac{d\mathbf{P}}{dq^i}| \tag{A.8}$$

After finding the del operator in natural coordinates, it is easy (and very long) to derive the expression for the gradient, divergence and curl in natural coordinates. I will not show these derivations, but I will limit myself to show the final expression for the gradient and laplacian of a scalar and the divergence of a vector a vector in natural coordinates, since that I will need them in the following sections.

The gradient of a scalar function ϕ in natural coordinates is:

$$\nabla \phi = \frac{\hat{\mathbf{e}}_s}{h_s} \frac{\partial \phi}{\partial s} + \frac{\hat{\mathbf{e}}_n}{h_n} \frac{\partial \phi}{\partial n} + \frac{\hat{\mathbf{e}}_\theta}{h_\theta} \frac{\partial \phi}{\partial \theta} \tag{A.9}$$

The divergence of a vector $\mathbf{A} = A_s \mathbf{e}_s + A_n \mathbf{e}_n + A_\theta \mathbf{e}_\theta$ in natural coordinates is:

$$\nabla \cdot \mathbf{A} = \frac{1}{h_s h_n h_\theta} \left[\frac{\partial}{\partial s} (A_s h_n h_\theta) + \frac{\partial}{\partial n} (A_n h_s h_\theta) + \frac{\partial}{\partial \theta} (A_\theta h_s h_n) \right] \tag{A.10}$$

The laplacian of a scalar function ϕ in natural coordinates is:

$$\nabla^2 \phi = \frac{1}{h_s h_n h_\theta} \left[\frac{\partial}{\partial s} \left(\frac{h_n h_\theta}{h_s} \frac{\partial \phi}{\partial s} \right) + \frac{\partial}{\partial n} \left(\frac{h_s h_\theta}{h_n} \frac{\partial \phi}{\partial n} \right) + \frac{\partial}{\partial \theta} \left(\frac{h_s h_n}{h_\theta} \frac{\partial \phi}{\partial \theta} \right) \right] \tag{A.11}$$

A.3 Lamé coefficients for natural coordinates in axisymmetric flows

The expression that I showed until now are valid for any curvilinear coordinates. I want now to find the scale factors for the natural coordinates in axisymmetric flows.

I will do it by using geometrical consideration at infinitesimal level, but the reader should keep in mind that may exist a more rigorous way to find them.

A.3.1 The scale factor h_s

The scale factor h_s is defined as:

$$h_s = |\frac{\partial \mathbf{P}}{\partial s}| \tag{A.12}$$

so, in order to find it, we need to vary the arc length s by an infinitesimal amount ds and see how the position vector $\mathbf{P}$ changes. In particular, we are interested in finding:

$$\frac{\partial \mathbf{P}}{\partial s} = \frac{\partial x}{\partial s}\mathbf{e}_x + \frac{\partial y}{\partial s}\mathbf{e}_y + \frac{\partial z}{\partial s}\mathbf{e}_z \tag{A.13}$$

Let's consider the following drawing:

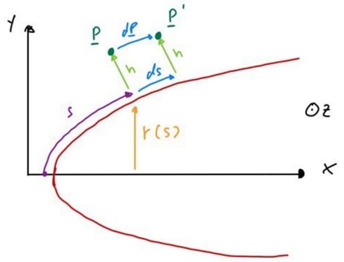

Figure A.3: Variation of the position vector $\mathbf{P}$ for an infinitesimal variation of the arc length s.

where the variation of the position vector $\mathbf{P}$ for an infinitesimal variation of the arc length s is shown. In order to properly understand what is happening, let's zoom in the drawing:

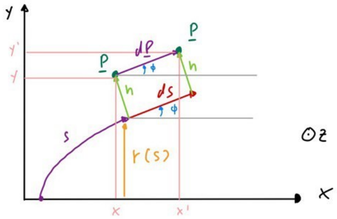

Figure A.4: Zoom in of the variation of the position vector $\mathbf{P}$ for an infinitesimal variation of the arc length s.

We see that, for an infinitesimal variation of ds, we can approximate the variation as linear, and from this, also the variation of the position vector $\mathbf{P}$ as linear. In particular we see that $d\mathbf{s} = d\mathbf{P}$. We defined ϕ as the angle between the x axis and $\mathbf{ds}$, that is equal to the angle between the x axis and $d\mathbf{P}$.

It is easy to see that:

$$dx = x' - x = dP\cos\phi = ds\cos\phi \tag{A.14}$$

And

$$dy = y' - y = dP\sin\phi = ds\sin\phi \tag{A.15}$$

So:

$$\frac{\partial x}{\partial s} = \cos\phi \tag{A.16}$$

And

$$\frac{\partial y}{\partial s} = \sin\phi \tag{A.17}$$

Clearly:

$$\frac{\partial z}{\partial s} = 0 \tag{A.18}$$

since that the z coordinate does not change for an infinitesimal variation of the arc length s.

So, the scale factor h_s is:

$$h_s = |\frac{\partial \mathbf{P}}{\partial s}| = \sqrt{\left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2 + \left(\frac{\partial z}{\partial s}\right)^2} = \sqrt{\cos^2\phi + \sin^2\phi + 0} = 1 \tag{A.19}$$

So, the scale factor h_s is equal to 1.

Furthermore, we found that the normalized basis vector $\hat{\mathbf{e}}_s$ is:

$$\hat{\mathbf{e}}_s = \cos\phi\mathbf{e}_x + \sin\phi\mathbf{e}_y \tag{A.20}$$

where the angle ϕ is the angle between the x axis and the tangent vector $\mathbf{e}_s$.

A.3.2 The scale factor h_n

The scale factor h_n is defined as:

$$h_n = |\frac{\partial \mathbf{P}}{\partial n}| \tag{A.21}$$

so, in order to find it, we need to vary the normal component n by an infinitesimal amount dn and see how the position vector $\mathbf{P}$ changes. In particular, we are interested in finding:

$$\frac{\partial \mathbf{P}}{\partial n} = \frac{\partial x}{\partial n}\mathbf{e}_x + \frac{\partial y}{\partial n}\mathbf{e}_y + \frac{\partial z}{\partial n}\mathbf{e}_z \tag{A.22}$$

Let's consider the following drawing:

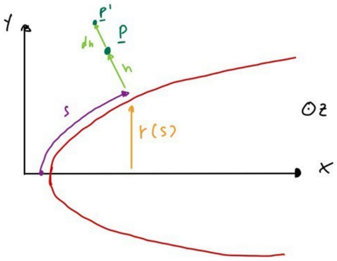

Figure A.5: Variation of the position vector **P** for an infinitesimal variation of the normal distante n.

where the variation of the position vector **P** for an infinitesimal variation of the normal distante n is shown. In order to properly understand what is happening, let's zoom in the drawing:

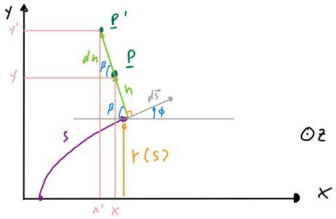

Figure A.6: Zoom in of the variation of the position vector **P** for an infinitesimal variation of the normal distance n.

We see that, for an infinitesimal variation of dn, we can approximate the variation as linear, and from this, also the variation of the position vector **P** as linear. In particular we see that $d\mathbf{n} = d\mathbf{P}$. We defined β as the angle between the negative x axis and $\mathbf{dn}$, and we see that it is the complement of ϕ, so that $\beta = 90° - \phi$.

It is easy to see that:

$$dx = x' - x = -dP\cos\beta = -dn\sin\phi \tag{A.23}$$

And

$$dy = y' - y = dP\sin\beta = dn\cos\phi \tag{A.24}$$

So:

$$\frac{\partial x}{\partial n} = -\sin\phi \tag{A.25}$$

And

$$\frac{\partial y}{\partial n} = \cos\phi \tag{A.26}$$

Clearly:

$$\frac{\partial z}{\partial n} = 0 \tag{A.27}$$

since that the z coordinate does not change for an infinitesimal variation of the normal distance n.

So, the scale factor h_n is:

$$h_n = |\frac{\partial \mathbf{P}}{\partial n}| = \sqrt{\left(\frac{\partial x}{\partial n}\right)^2 + \left(\frac{\partial y}{\partial n}\right)^2 + \left(\frac{\partial z}{\partial n}\right)^2} = \sqrt{\sin^2\phi + \cos^2\phi + 0} = 1 \tag{A.28}$$

So, the scale factor h_n is equal to 1.

Furthermore, we found that the normalized basis vector $\hat{\mathbf{e}}_n$ is:

$$\hat{\mathbf{e}}_n = -\sin\phi\mathbf{e}_x + \cos\phi\mathbf{e}_y \tag{A.29}$$

where the angle ϕ is the angle between the x axis and the tangent vector $\mathbf{e}_s$.

A.3.3 The scale factor h_θ

The scale factor h_θ is defined as:

$$h_\theta = |\frac{\partial \mathbf{P}}{\partial \theta}| \tag{A.30}$$

so, in order to find it, we need to vary the angle θ by an infinitesimal amount $d\theta$ and see how the position vector $\mathbf{P}$ changes. In particular, we are interested in finding:

$$\frac{\partial \mathbf{P}}{\partial \theta} = \frac{\partial x}{\partial \theta}\mathbf{e}_x + \frac{\partial y}{\partial \theta}\mathbf{e}_y + \frac{\partial z}{\partial \theta}\mathbf{e}_z \tag{A.31}$$

Let's consider the following drawing:

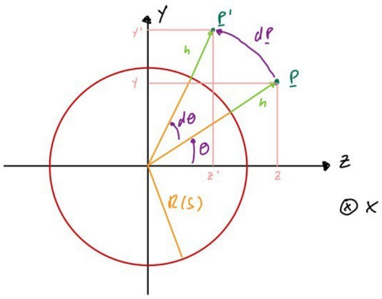

Figure A.7: Variation of the position vector **P** for an infinitesimal variation of the angle θ.

where the variation of the position vector $\mathbf{P}$ for an infinitesimal variation of angle θ is shown. We see that, for an infinitesimal variation of $d\theta$, we have a arc variation of the position vector $\mathbf{P}$.

We recall the addition formulas for the sine and cosine, that are:

$$\sin(a + b) = \sin a\cos b + \cos a\sin b \tag{A.32}$$

$$\cos(a+b) = \cos a \cos b - \sin a \sin b \tag{A.33}$$

It is easy to see that:

$$dz = z' - z = (R+n)\cos(\theta + d\theta) - (R+n)\cos\theta = (R+n)\left(\cos(\theta)\cos(d\theta) - \sin(\theta)\sin(d\theta) - \cos(\theta)\right) \tag{A.34}$$

And

$$dy = y' - y = (R+n)\sin(\theta + d\theta) - (R+n)\sin\theta = (R+n)\left(\sin(\theta)\cos(d\theta) + \cos(\theta)\sin(d\theta) - \sin(\theta)\right) \tag{A.35}$$

For $d\theta \to 0$, we can approximate the trigonometric functions as:

$$\cos(d\theta) \approx 1 \tag{A.36}$$

$$\sin(d\theta) \approx d\theta \tag{A.37}$$

So:

$$\frac{\partial z}{\partial \theta} = -(R+n)\sin(\theta) d\theta \tag{A.38}$$

And

$$\frac{\partial y}{\partial \theta} = (R+n)\cos(\theta) d\theta \tag{A.39}$$

Clearly:

$$\frac{\partial x}{\partial \theta} = 0 \tag{A.40}$$

since that the x coordinate does not change for an infinitesimal variation of angle θ.

So, the scale factor h_θ is:

$$h_\theta = |\frac{\partial \mathbf{P}}{\partial \theta}| = \sqrt{\left(\frac{\partial x}{\partial \theta}\right)^2 + \left(\frac{\partial y}{\partial \theta}\right)^2 + \left(\frac{\partial z}{\partial \theta}\right)^2} = \sqrt{0 + (R+n)^2 \sin^2\theta + (R+n)^2\cos^2\theta} = R(s) + n \tag{A.41}$$

So, the scale factor h_s is equal to $R(s) + n$.

Furthermore, we found that the normalized basis vector $\hat{\mathbf{e}}_\theta$ is:

$$\hat{\mathbf{e}}_\theta = \cos\theta \mathbf{e}_y - \sin\theta \mathbf{e}_z \tag{A.42}$$

Now I am going to make the first approximation for this derivation, that is the following: The radius of the body $R(s)$ is much larger than the normal distance n, so that $R(s) \gg n$. This is a reasonable approximation as we will see in the boundary layer equations derivation, since that we will assume that the boundary layer is very thin compared to the radius of the body.

So, the scale factor h_θ is approximately equal to $R(s)$.

A.3.4 Summary of the scale factors

The scale factors for the natural coordinates in axisymmetric flows are:

$$h_s = 1 \tag{A.43}$$

$$h_n = 1 \tag{A.44}$$

$$h_\theta = R(s) \tag{A.45}$$

So, the gradient, divergence and laplacian in natural coordinates for axisymmetric flows are:

$$\nabla\phi = \hat{\mathbf{e}}_s \frac{\partial \phi}{\partial s} + \hat{\mathbf{e}}_n \frac{\partial \phi}{\partial n} + \frac{\hat{\mathbf{e}}_\theta}{R(s)} \frac{\partial \phi}{\partial \theta} \tag{A.46}$$

$$\nabla \cdot \mathbf{A} = \frac{1}{R(s)}\left[\frac{\partial}{\partial s}\left(R(s) A_s\right) + \frac{\partial}{\partial n}\left(R(s) A_n\right) + \frac{\partial}{\partial \theta}\left(A_\theta\right)\right] \tag{A.47}$$

$$\nabla^2\phi = \frac{1}{R(s)}\left[\frac{\partial}{\partial s}\left(R(s)\frac{\partial \phi}{\partial s}\right) + \frac{\partial}{\partial n}\left(R(s)\frac{\partial \phi}{\partial n}\right) + \frac{\partial}{\partial \theta}\left(\frac{1}{R(s)}\frac{\partial \phi}{\partial \theta}\right)\right] \tag{A.48}$$

A.4 Boundary layer theory

After deriving the del operator in natural coordinates for axisymmetric flows, we can now derive the boundary layer equations.

First of all, let's remember that the problem is axisymmetric, so the flow does not depend on the angle θ. This means that all the derivatives with respect to θ are zero, $\frac{\partial}{\partial\theta} = 0$.

We will consider the axisymmetric natural coordinates (s, n), and the velocity vector will be $\mathbf{u} = u\hat{\mathbf{e}}_s + v\hat{\mathbf{e}}_n$. In order to get a more familiar notation, we will use (x, y) as the coordinates, with x along the body surface and y normal to the body surface.

In order to derive the boundary layer equations, we need to start from the Navier-Stokes equations Eq. 2.179 and we need to cast them in a nondimensional form, in order to understand which terms can be dropped.

A.4.1 What is the boundary layer

The boundary layer is the region adjacent to a surface, where the effects of friction and thermal conduction are dominant. The flow external to the boundary layer, However, is not greatly affected by such effects, and can be considered inviscid and adiabatic.

This division of fluid flow into two distanct regions, the viscous boundary layer adjacent to the surface and the inviscid flow external to the boundary layer, is arguably the most important concept in fluid dynamics since the beginning of the 20th century[1].

In particular, we will consider **Laminar boundary layers**, where phenomena like separation and transition to turbulence are not considered, with a big Reynold number, in order to simplify the equations.

The boundary layer thickness δ is an important concept, and goes to zero as the viscosity of the fluid goes to zero. The foundamental assumption of boundary layer theory is that the boundary layer is very thin compared to other length scales, such the radius of curvature of the surface on which the boundary layer develops. This justifies our approximation of the scale factor h_θ.

Across this thin layer, which can exist only in high Reynolds number flow, the velocity varies rapudly enough for viscous effects to be important.

Outside the boundary layer, the flow is Eulerian, so that the Navier-Stokes equations are simplified to the Euler equations and are easy to solve.

A.4.2 Boundary layer boundary conditions

In order to solve the boundary layer equations, we need to specify the boundary conditions for the velocity, pressure and enthalpy.

The boundary conditions on the body surface are:

$$\begin{cases} u(x,0) = 0 \\ v(x,0) = 0 \\ p(x,0) = p_w \\ h(x,0) = h_w \end{cases} \tag{A.49}$$

where the first condition is given by the no-slip condition, the second condition is given by the no-penetration condition, the third condition is given by the pressure on the body surface and the fourth condition is given by the enthalpy on the body surface.

The boundary conditions at the edge of the boundary layer are:

$$\begin{cases} u(x,\delta) = U_\infty \\ p(x,\delta) = p_\infty \\ h(x,\delta) = h_\infty \end{cases} \tag{A.50}$$

This conditions are given by the flow outside the boundary layer.

A.4.3 Nondimensionalization of the Navier-Stokes equations

In order to nondimensionalize the Navier-Stokes equations, we need to define the following reference quantities:

- ρ_∞: a reference density of the fluid, the one outside the boundary layer.
- V_∞: a reference velocity of the fluid, the one outside the boundary layer (tangential component).

- p_∞: a reference pressure of the fluid, the one outside the boundary layer.
- L: a reference length of the body (on which u changes in a relevant way).
- μ_∞: a reference dynamic viscosity of the fluid, the one outside the boundary layer.
- T_∞: a reference temperature of the fluid, the one outside the boundary layer.
- $h_\infty = C_p T_\infty$: a reference enthalpy of the fluid.
- k_∞: a reference thermal conductivity of the fluid, the one outside the boundary layer.
- $q_\infty = k_\infty \frac{T_\infty}{L_\infty}$: a reference heat flux of the fluid.

The nondimensional quantities are defined as:

$$\tilde{\rho} = \frac{\rho}{\rho_\infty} \tag{A.51}$$

$$\tilde{u} = \frac{u}{V_\infty} \tag{A.52}$$

$$\tilde{v} = \frac{v}{V_\infty} \tag{A.53}$$

$$\tilde{p} = \frac{p}{p_\infty} \tag{A.54}$$

$$\tilde{x} = \frac{x}{L} \tag{A.55}$$

$$\tilde{y} = \frac{y}{L} \tag{A.56}$$

$$\tilde{R} = \frac{R}{L} \tag{A.57}$$

$$\tilde{\mu} = \frac{\mu}{\mu_\infty} \tag{A.58}$$

$$\tilde{h} = \frac{h}{h_\infty} \tag{A.59}$$

$$\tilde{\mathbf{q}} = \frac{q}{q_\infty} \tag{A.60}$$

We can now nondimensionalize the Navier-Stokes equations Eq. 2.179.

A.4.4 Continuity equation

The expanded continuity equation is:

$$\frac{1}{R}\frac{\partial}{\partial x}(\rho u R) + \frac{1}{R}\frac{\partial}{\partial y}(\rho v R) = 0 \tag{A.61}$$

The nondimensionalized continuity equation is:

$$\frac{\partial}{\partial \tilde{x}}(\tilde{\rho}\tilde{u}\tilde{R}) + \frac{\partial}{\partial \tilde{y}}(\tilde{\rho}\tilde{v}\tilde{R}) = 0 \tag{A.62}$$

after dividing by $\rho_\infty V_\infty$.

This equation cannot be simplified further, so we will keep it in this form, but let's do some consideration about the order of magnitude of the terms.

The basic assumption of boundary-layer theory is that a boundary layer is **very thin** in comparison with the scale of the body L, so $\delta \ll L$, where δ is the boundary layer thickness.

Now, since that $\tilde{u}$ varies from 0 at the wall to 1 at the edge of the boundary layer, we can say that $\tilde{u}$ is in the order of magnitude of 1, synbolized by $\tilde{u} \sim O(1)$. Similarly, $\tilde{\rho} \sim O(1)$. Also, because x varies from 0 to c, we can say that $\tilde{x} \sim O(1)$, and similarly for $\tilde{R} \sim O(1)$. However, because y varies from 0 to δ, where $\delta \ll L$, than $\tilde{y} \sim O(\delta/L)$. Without loss of generality, we can say that $L = 1$, so that $\tilde{y} \sim O(\delta)$. Putting everything inside the continuity equation, we get:

$$\frac{[O(1)][O(1)][O(1)]}{[O(1)]} + \frac{[O(1)][O(1)][\tilde{v}]}{[O(\delta)]} = 0 \tag{A.63}$$

so clearly $\tilde{v} \sim O(\delta)$.

This means that the normal velocity v is much smaller than the tangential velocity u, and this is a key assumption of boundary layer theory.

So, in the **Boundary layer theory**, the continuity equation is:

$$\frac{\partial}{\partial x}(\rho u R) + \frac{\partial}{\partial y}(\rho v R) = 0 \tag{A.64}$$

A.4.5 X-Momentum equation

Let's start by writing the x-momentum equation:

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \left(\nabla \cdot \overline{\overline{\tau}}\right)_x \tag{A.65}$$

We need to expand the viscous stress tensor $\overline{\overline{\tau}} = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) - \frac{2}{3}\mu \nabla \cdot \mathbf{u}$. Let us recall that:

$$\overline{\overline{\tau}}_x = \left[2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \nabla \cdot \mathbf{u}\ ,\ \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] = \left[\frac{4}{3}\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \frac{u}{R}\frac{\partial R}{\partial x}\ ,\ \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] \tag{A.66}$$

So the expanded x-momentum equation is:

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(\frac{4}{3}\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \frac{u}{R}\frac{\partial R}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right) \tag{A.67}$$

Let's observe the first advective term $\rho u \frac{\partial u}{\partial x}$. A measure of this term is: $\rho u \frac{\partial u}{\partial x} \sim \rho_\infty V_\infty \frac{V_\infty}{L} \sim \rho_\infty V_\infty^2/L$. If we take a look to the viscous term $\frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right)$, this has the order of magnitude $\mu_\infty \frac{V_\infty}{\delta^2}$. The magnitude of δ can now be estimated by noting that the advective and viscous terms should be of the same order within the boundary layer, so, by equating the magnitudes of the two terms, we get:

$$\delta \sim \sqrt{\nu L/V_\infty} \sim \frac{1}{Re} \tag{A.68}$$

where Re is the Reynolds number of the flow. We clearly see that for $Re \to \infty$, the boundary layer thickness $\delta \to 0$, and the viscous effects are confined to a thin layer near the surface.

A formal simplification of the equations of motion within the boundary layer can now be performed. The basic idea is that variations across the boundary layer occur over a much shorter length scale than variations along the layer:

$$\frac{\partial}{\partial x} \sim \frac{1}{L}\ ,\ \frac{\partial}{\partial y} \sim \frac{1}{\delta} \tag{A.69}$$

where $\delta \ll L$ when $Re \gg 1$.

Let's now nondimensionalize the x-momentum equation:

$$\tilde{\rho}\tilde{u}\frac{\partial \tilde{u}}{\partial \tilde{x}} + \tilde{\rho}\tilde{v}\frac{\partial \tilde{u}}{\partial \tilde{y}} = -\frac{1}{\gamma_\infty M_\infty^2}\frac{\partial \tilde{p}}{\partial \tilde{x}} + \frac{1}{Re_\infty}\frac{\partial}{\partial \tilde{x}}\left(\frac{4}{3}\tilde{\mu}\frac{\partial \tilde{u}}{\partial \tilde{x}} - \frac{2}{3}\tilde{\mu}\frac{\partial \tilde{v}}{\partial \tilde{y}} - \frac{2}{3}\tilde{\mu}\frac{\tilde{u}}{\tilde{R}}\frac{\partial \tilde{R}}{\partial \tilde{x}}\right) + \frac{1}{Re_\infty}\frac{\partial}{\partial \tilde{y}}\left(\tilde{\mu}\left(\frac{\partial \tilde{v}}{\partial \tilde{x}} + \frac{\partial \tilde{u}}{\partial \tilde{y}}\right)\right) \tag{A.70}$$

where $Re_\infty = \frac{\rho_\infty V_\infty L}{\mu_\infty}$ is the Reynolds number of the flow outside the boundary layer, $\gamma_\infty = fracC_{p,\infty}C_{v,\infty}$ is the ratio of specific heats of the fluid outside the boundary layer, and M_∞ is the Mach number of the flow outside the boundary layer. The term in front of the pressure gradient is obtained in the following way:

$$\frac{p_\infty}{\rho_\infty V_\infty^2} = \frac{R_\infty T_\infty}{V_\infty^2}\frac{\gamma_\infty}{\gamma_\infty} = \frac{c_\infty^2}{\gamma_\infty V_\infty^2} = \frac{1}{\gamma_\infty M_\infty^2} \tag{A.71}$$

where we used the ideal gas equation of state $p = \rho R T$,the definition of the Mach number $M_\infty = \frac{V_\infty}{c_\infty}$ And the definition of the speed of sound $c_\infty = \sqrt{\gamma_\infty R_\infty T_\infty}$.

Now we are ready to study the order of the terms:

$$\tilde{\rho}\tilde{u}\frac{\partial \tilde{u}}{\partial \tilde{x}} = O(1)\ ,\ \tilde{\rho}\tilde{v}\frac{\partial \tilde{u}}{\partial \tilde{y}} = O(1)\ ,\ -\frac{\partial \tilde{p}}{\partial \tilde{x}} = O(1) \tag{A.72}$$

$$\frac{\partial}{\partial \tilde{x}}\left(\frac{4}{3}\tilde{\mu}\frac{\partial \tilde{u}}{\partial \tilde{x}}\right) = O(1) \tag{A.73}$$

$$\frac{\partial}{\partial \tilde{x}}\left(-\frac{2}{3}\tilde{\mu}\frac{\partial \tilde{v}}{\partial \tilde{y}}\right) = O(1) \tag{A.74}$$

$$\frac{\partial}{\partial \tilde{x}}\left(-\frac{2}{3}\tilde{\mu}\frac{\tilde{u}}{\tilde{R}}\frac{\partial \tilde{R}}{\partial \tilde{x}}\right) = O(1) \tag{A.75}$$

$$\frac{\partial}{\partial \tilde{y}}\left(\tilde{\mu}\frac{\partial \tilde{v}}{\partial \tilde{x}}\right) = O(1) \tag{A.76}$$

$$\frac{\partial}{\partial \tilde{y}}\left(\tilde{\mu}\frac{\partial \tilde{u}}{\partial \tilde{y}}\right) = O(\frac{1}{\delta^2}) \tag{A.77}$$

Since that we assumed that the Reynolds number is large, and we found that $\delta^2 \sim \frac{1}{Re}$, we can say that:

$$\frac{1}{Re_\infty} = O(\delta^2) \tag{A.78}$$

This is true only if the Reynolds number is large, so that the boundary layer is thin, that is the hypotesis we made at the beginning.

We also assume that:

$$\frac{1}{\gamma_\infty M_\infty^2} = O(1) \tag{A.79}$$

So, the x-momentum equation in the order form is:

$$O(1)+O(1) = O(1)+O(\delta^2)(O(1)+O(1)+O(1))+O(\delta^2)(O(1)+O(\frac{1}{\delta^2})) = O(1)+O(\delta^2)+O(\delta^2)+O(\delta^2)+O(\delta^2)+O(1) \tag{A.80}$$

We can see that there are 4 terms with an order $O(\delta^2)$, and the others have order $O(1)$. So, in the **Boundary layer theory approximation**, we can drop the terms with order $O(\delta^2)$, and we get:

$$\tilde{\rho}\tilde{u}\frac{\partial \tilde{u}}{\partial \tilde{x}} + \tilde{\rho}\tilde{v}\frac{\partial \tilde{u}}{\partial \tilde{y}} = -\frac{1}{\gamma_\infty M_\infty^2}\frac{\partial \tilde{p}}{\partial \tilde{x}} + \frac{1}{Re_\infty}\frac{\partial}{\partial \tilde{y}}\left(\tilde{\mu}\frac{\partial \tilde{u}}{\partial \tilde{y}}\right) \tag{A.81}$$

So, in the **Boundary layer theory**, the x-momentum equation is:

$$\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right) \tag{A.82}$$

A.4.6 Y-Momentum equation

Let's start by writing the y-momentum equation:

$$\rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \left(\nabla\cdot\overline{\overline{\tau}}\right)_y \tag{A.83}$$

We need to expand the viscous stress tensor $\overline{\overline{\tau}} = \mu(\nabla\mathbf{u}+\nabla\mathbf{u}^T) - \frac{2}{3}\mu\nabla\cdot\mathbf{u}$. Let us recall that:

$$\overline{\overline{\tau}}_y = \left[\mu\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\ ,\ 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\nabla\cdot\mathbf{u}\right] = \left[\mu\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\ ,\ \frac{4}{3}\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\frac{u}{R}\frac{\partial R}{\partial x}\right] \tag{A.84}$$

So the expanded y-momentum equation is:

$$\rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left(\mu\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\right) + \frac{\partial}{\partial y}\left(\frac{4}{3}\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\frac{u}{R}\frac{\partial R}{\partial x}\right) \tag{A.85}$$

Let's now nondimensionalize the y-momentum equation:

$$\tilde{\rho}\tilde{u}\frac{\partial \tilde{v}}{\partial \tilde{x}} + \tilde{\rho}\tilde{v}\frac{\partial \tilde{v}}{\partial \tilde{y}} = -\frac{1}{\gamma_\infty M_\infty^2}\frac{\partial \tilde{p}}{\partial \tilde{v}} + \frac{1}{Re_\infty}\frac{\partial}{\partial \tilde{x}}\left(\tilde{\mu}\left(\frac{\partial \tilde{v}}{\partial \tilde{x}}+\frac{\partial \tilde{u}}{\partial \tilde{y}}\right)\right) + \frac{1}{Re_\infty}\frac{\partial}{\partial \tilde{y}}\left(\frac{4}{3}\tilde{\mu}\frac{\partial \tilde{v}}{\partial \tilde{y}} - \frac{2}{3}\tilde{\mu}\frac{\partial \tilde{u}}{\partial \tilde{x}} - \frac{2}{3}\tilde{\mu}\frac{\tilde{u}}{\tilde{R}}\frac{\partial \tilde{R}}{\partial \tilde{x}}\right) \tag{A.86}$$

where $Re_\infty = \frac{\rho_\infty V_\infty L}{\mu_\infty}$ is the Reynolds number of the flow outside the boundary layer.

Now we are ready to study the order of the terms:

$$\tilde{\rho}\tilde{u}\frac{\partial \tilde{v}}{\partial \tilde{x}} = O(\delta)\ ,\ \tilde{\rho}\tilde{v}\frac{\partial \tilde{v}}{\partial \tilde{y}} = O(\delta) \tag{A.87}$$

$$\frac{\partial}{\partial \tilde{x}}\left(\tilde{\mu}\frac{\partial \tilde{v}}{\partial \tilde{x}}\right) = O(\delta) \tag{A.88}$$

$$\frac{\partial}{\partial \tilde{x}}\left(\tilde{\mu}\frac{\partial \tilde{u}}{\partial \tilde{y}}\right) = O(\frac{1}{\delta}) \tag{A.89}$$

$$\frac{\partial}{\partial \tilde{y}}\left(\frac{4}{3}\tilde{\mu}\frac{\partial \tilde{v}}{\partial \tilde{y}}\right) = O(\frac{1}{\delta}) \tag{A.90}$$

$$\frac{\partial}{\partial \tilde{y}}\left(-\frac{2}{3}\tilde{\mu}\frac{\partial \tilde{u}}{\partial \tilde{x}}\right) = O(\frac{1}{\delta}) \tag{A.91}$$

$$\frac{\partial}{\partial \tilde{y}}\left(-\frac{2}{3}\tilde{\mu}\frac{\tilde{u}}{\tilde{R}}\frac{\partial \tilde{R}}{\partial \tilde{x}}\right) = O(\frac{1}{\delta}) \tag{A.92}$$

Let's remember that $\frac{1}{Re_\infty} = O(\delta^2)$.

So, the y-momentum equation in the order form is:

$$O(\delta)+O(\delta) = -\frac{1}{\gamma_\infty M_\infty^2}\frac{\partial \tilde{p}}{\partial \tilde{y}}+O(\delta^2)(O(\delta)+O(\frac{1}{\delta}))+O(\delta^2)(O(\frac{1}{\delta})+O(\frac{1}{\delta})+O(\frac{1}{\delta})) = -\frac{1}{\gamma_\infty M_\infty^2}\frac{\partial \tilde{p}}{\partial \tilde{y}}+O(\delta^3)+O(\delta)+O(\delta)+O(\delta)+O(\delta) \tag{A.93}$$

From the above equation, we can see that $-\frac{1}{\gamma_\infty M_\infty^2}\frac{\partial \tilde{p}}{\partial \tilde{y}} = O(\delta)$ or smaller. Since that δ is very small, this implies that $-\frac{1}{\gamma_\infty M_\infty^2}\frac{\partial \tilde{p}}{\partial \tilde{y}}$ is very small, so:

$$\frac{\partial \tilde{p}}{\partial \tilde{y}} = 0 \tag{A.94}$$

So, in the **Boundary layer theory**, the y-momentum equation is:

$$\frac{\partial p}{\partial y} = 0 \tag{A.95}$$

This equation is very important, because it tells us that at a given x location, the pressure is constant throughout the boundary layer in a direction normal to the surface. This implies that the pressure distribution at the outer edge of the boundary layer is impressed directly to the surface without change, so $p = p(x) = p_e(x)$, where $p_e(x)$ is the pressure distribution outside the boundary layer.

As we said, the pressure outside the boundary layer is given by the Euler equations. Let's consider the x-momentum equation of the Euler equations:

$$\rho u \frac{\partial u}{\partial x} = -\frac{\partial p}{\partial x} \tag{A.96}$$

So, the pressure gradient inside the boundary layer is given by:

$$\frac{\partial p}{\partial x} = -\rho_e u_e \frac{\partial u_e}{\partial x} \tag{A.97}$$

where the subscript e denotes the values of the variables outside the boundary layer. The v component of the velocity has been neglected because the edge is supposed to be very close to the wall, since that the boundary layer is very thin, so $v_e \approx 0$.

The reader should note that this does not hold for high Mach numbers, as explained by Anderson[1], since that we could get $\frac{1}{\gamma_\infty M_\infty^2} = O(\delta)$ or more, and so $\frac{\partial \tilde{p}}{\partial \tilde{y}} = O(1)$. In those cases, the Viscous shock equations are better suited, but more computational expensive.

A.4.7 Energy equation

Let's start by writing the energy equation:

$$\rho u \frac{\partial h}{\partial x} + \rho v \frac{\partial h}{\partial y} = \mathbf{u}\cdot\nabla p - \nabla\cdot\mathbf{q} + \overline{\overline{\tau}} : \nabla\mathbf{u} \tag{A.98}$$

I will not go through the entire derivation, since that it is very similar to the previous ones. I will just define the Prandtl number:

$$\mathrm{Pr} = \frac{\mu_\infty C_{p,\infty}}{k_\infty} \tag{A.99}$$

and state that $\mathrm{Pr} = O(1)$.

The energy equation in the **Boundary layer theory** is:

$$\rho u \frac{\partial h}{\partial x} + \rho v \frac{\partial h}{\partial y} = u\frac{\partial p}{\partial x} + \mu\left(\frac{\partial u}{\partial y}\right)^2 - \frac{\partial q^y}{\partial y} \tag{A.100}$$

where q^y is the heat flux in the y direction.

A.5 Summary of the equations and the hypotheses

The equations of the boundary layer theory are:

$$\frac{\partial}{\partial x}\left(\rho u R\right) + \frac{\partial}{\partial y}\left(\rho v R\right) = 0 \tag{A.101}$$

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) \tag{A.102}$$

$$\frac{\partial p}{\partial y} = 0 \tag{A.103}$$

$$\rho u \frac{\partial h}{\partial x} + \rho v \frac{\partial h}{\partial y} = u \frac{\partial p}{\partial x} + \mu \left(\frac{\partial u}{\partial y}\right)^2 - \frac{\partial q^y}{\partial y} \tag{A.104}$$

The hypotheses of the boundary layer theory are:

- The Reynolds number Re inside the boundary layer is very large, so the boundary layer thickness, $\delta \sim \frac{1}{Re}$ is very small. This means that the boundary layer thickness is small compared to the radius of curvature of the body.
- The flow is laminar inside the boundary layer, so no turbulence is present.
- The Mach number is not too high, so that $\frac{1}{M_\infty^2} = O(1)$.

Appendix B

Boundary layer equations in self-similar form

In this appendix I am going to derive the boundary layer equations in self-similar form. This is a very useful form of the equations, because it allows to solve the equations in a more simple way, since that the equations are reduced to a single ordinary differential equation.

B.1 Levy-Lees transformation

The Levy-Lees transformation is made of two transformations:

- Mangler transformation: in order to transform the axisymmetric boundary layer to equivalent 2D boundary layer
- Howarth, Illingworth, Stewartson transformation: in order to transform a compressible boundary layer to an equivalent incompressible one

The transformation is:

$$\xi(x) = \int_1^x \rho_e \mu_e u_e R^2 dx' \tag{B.1}$$

$$\eta(x,y) = \frac{Ru_e}{\sqrt{2\xi}} \int_0^y \rho dy' \tag{B.2}$$

The first quantity depends on x only, since that all the edge quantities are functions of x only.

For the *multivariable chain rule*, we have:

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} \tag{B.3}$$

$$\frac{\partial}{\partial y} = \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta} \tag{B.4}$$

since that $\frac{\partial \xi}{\partial y} = 0$.

Furthermore, we have:

$$\frac{\partial \xi}{\partial x} = \rho_e \mu_e u_e R^2 \tag{B.5}$$

and:

$$\frac{\partial \eta}{\partial y} = \frac{Ru_e}{\sqrt{2\xi}}\rho \tag{B.6}$$

we will see that there is no need to find $\frac{\partial \eta}{\partial x}$.

We can introduce a stream function Ψ such that:

$$\rho u R = \frac{\partial \Psi}{\partial y} \tag{B.7}$$

$$\rho v R = -\frac{\partial \Psi}{\partial x} \tag{B.8}$$

and the adimensional stream function $f(\xi,\eta)$ such that:

$$f(\xi,\eta) = \frac{\Psi}{\sqrt{2\xi}} \tag{B.9}$$

from which:

$$\Psi = \sqrt{2\xi} f \tag{B.10}$$

The reader can verify that $[\Psi] = \frac{\mathrm{Kg}}{s}$, $[\xi] = \frac{\mathrm{Kg}^2}{m^2}$, so that $[f] = [\varnothing]$.

By using Eq. B.7, Eq. B.10 and Eq. B.3 we get:

$$\rho u R = \frac{\partial \Psi}{\partial y} = \frac{\partial}{\partial y}\left(\sqrt{2\xi} f\right) = \sqrt{2\xi}\frac{\partial f}{\partial \eta}\frac{\partial \eta}{\partial y} = \sqrt{2\xi}\frac{\partial f}{\partial \eta}\frac{R u_e}{\sqrt{2\xi}}\rho \tag{B.11}$$

So:

$$\frac{\partial f}{\partial \eta} = f' = F = \frac{u}{u_e} \tag{B.12}$$

By using Eq. B.8, Eq. B.10 and Eq. B.4 we get:

$$\rho v R = -\frac{\partial \Psi}{\partial x} = -\frac{\partial f}{\partial x}\sqrt{2\xi} - f\frac{\partial}{\partial x}\sqrt{2\xi} = -\frac{\partial \xi}{\partial x}\frac{\partial f}{\partial \xi}\sqrt{2\xi} - \frac{\partial \eta}{\partial x}\frac{\partial f}{\partial \eta}\sqrt{2\xi} - f\frac{1}{\sqrt{2\xi}}\frac{\partial \xi}{\partial x} = -\sqrt{2\xi}\frac{\partial f}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{1}{\sqrt{2\xi}}\frac{\partial \xi}{\partial x}V \tag{B.13}$$

where:

$$V = -f - 2\xi\frac{\partial f}{\partial \xi} \tag{B.14}$$

is the adimensional normal velocity in the transformed coordinates [5].

From Eq. B.13 we can derive an expression for V as a function of the other variables:

$$V = \frac{2\xi}{\frac{\partial \xi}{\partial x}}\left(f'\frac{\partial \eta}{\partial x} + \frac{\rho v R}{\sqrt{2\xi}}\right) \tag{B.15}$$

We clearly see that f' and V are adimensional quantities.

For the energy equation, we define the adimensional temperature as:

$$g = \frac{T}{T_e} \tag{B.16}$$

Now we are ready to derive the boundary layer equations with the new coordinates.

B.2 Continuity equation

Let's start by understanding what $\frac{\partial}{\partial \eta}\left(\frac{\partial \eta}{\partial x}\right)$ is. It is going to be clearer later why we are interested in this quantity.

$$\frac{\partial}{\partial \eta}\left(\frac{\partial \eta}{\partial x}\right) = \frac{1}{\frac{\partial \eta}{\partial y}}\frac{\partial}{\partial y}\left(\frac{\partial \eta}{\partial x}\right) = \frac{1}{\frac{\partial \eta}{\partial y}}\frac{\partial^2 \eta}{\partial x \partial y} = \frac{1}{\frac{\partial \eta}{\partial y}}\frac{\partial}{\partial x}\left(\frac{\partial \eta}{\partial y}\right) = \frac{\sqrt{2\xi}}{R u_e \rho}\frac{\partial}{\partial x}\left(\frac{R u_e \rho}{\sqrt{2\xi}}\right) = \frac{\sqrt{2\xi}}{R u_e \rho}\frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi}\left(\frac{R u_e \rho}{\sqrt{2\xi}}\right) + \frac{\partial \eta}{\partial x}\frac{\partial \rho}{\partial \eta}\frac{1}{\rho} \tag{B.17}$$

by remebering that $\frac{R u_e}{\sqrt{2\xi}}$ does not depend on η.

Now, we are interested in finding $\frac{\partial V}{\partial \eta}$:

$$\frac{\partial V}{\partial \eta} = \frac{2\xi}{\frac{\partial \xi}{\partial x}}\left(\frac{\partial F}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial}{\partial \eta}\left(\frac{\rho v R}{\sqrt{2\xi}}\right) + F\frac{\partial}{\partial \eta}\left(\frac{\partial \eta}{\partial x}\right)\right) \tag{B.18}$$

The expression we want to find for the continuity equation is:

$$2\xi\frac{\partial F}{\partial \xi} + \frac{\partial V}{\partial \eta} + F = 0 \tag{B.19}$$

For simplicity, I will name the following terms of the continuity equation:

$$(1) = 2\xi\frac{\partial F}{\partial \xi} \tag{B.20}$$

$$(2) = \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{\partial\eta}{\partial x}\frac{\partial F}{\partial\eta} \tag{B.21}$$

$$(3) = \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{\partial}{\partial\eta}\left(\frac{\rho v R}{\sqrt{2\xi}}\right) \tag{B.22}$$

$$(4) = 2\xi F\frac{\sqrt{2\xi}}{Ru_e\rho}\frac{\partial}{\partial\xi}\left(\frac{Ru_e\rho}{\sqrt{2\xi}}\right) \tag{B.23}$$

$$(5) = \frac{2\xi}{\frac{\partial\xi}{\partial x}}F\frac{\partial\eta}{\partial x}\frac{\partial\rho}{\partial\eta}\frac{1}{\rho} \tag{B.24}$$

$$(6) = F \tag{B.25}$$

So that the continuity equation is $(1) + (2) + (3) + (4) + (5) + (6) = 0$.

Let's start the derivation:

$$\frac{\partial}{\partial x}\left(\rho u R\right) + \frac{\partial}{\partial y}\left(\rho v R\right) = 0 \tag{B.26}$$

$$\frac{\partial\xi}{\partial x}\frac{\partial}{\partial\xi}(\rho u R) + \frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(\rho u R) + \frac{\partial\eta}{\partial y}\frac{\partial}{\partial\eta}(\rho v R) = 0 \tag{B.27}$$

$$\frac{\partial\xi}{\partial x}\frac{\partial}{\partial\xi}(\rho u R) + \frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(\rho u R) + Ru_e\rho\frac{\partial}{\partial\eta}(\frac{\rho v R}{\sqrt{2\xi}}) = 0 \tag{B.28}$$

We multiply by $\frac{2\xi}{\frac{\partial\xi}{\partial x}}$:

$$2\xi\frac{\partial}{\partial\xi}(\rho u R) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(\rho u R) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}Ru_e\rho\frac{\partial}{\partial\eta}\left(\frac{\rho v R}{\sqrt{2\xi}}\right) = 0 \tag{B.29}$$

We divide by $Ru_e\rho$:

$$\frac{2\xi}{Ru_e\rho}\frac{\partial}{\partial\xi}(\rho u R\frac{u_e}{u_e}) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{1}{Ru_e\rho}\frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(\rho u R) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{\partial}{\partial\eta}\left(\frac{\rho v R}{\sqrt{2\xi}}\right) = 0 \tag{B.30}$$

We recognize the term (3):

$$\frac{2\xi}{Ru_e\rho}\frac{\partial}{\partial\xi}(\rho F R u_e) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{1}{Ru_e\rho}\frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(\rho u R) + (3) = 0 \tag{B.31}$$

We expand the first term:

$$2\xi\frac{\partial F}{\partial\xi} + \frac{2\xi}{Ru_e\rho}F\frac{\partial}{\partial\xi}(\rho R u_e) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{1}{Ru_e\rho}\frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(\rho u R) + (3) = 0 \tag{B.32}$$

We recognize the term (1):

$$\frac{2\xi}{Ru_e\rho}F\frac{\partial}{\partial\xi}(\rho R u_e) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{1}{Ru_e\rho}\frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(\rho u R) + (1,3) = 0 \tag{B.33}$$

We expand the last term:

$$\frac{2\xi}{Ru_e\rho}F\frac{\partial}{\partial\xi}(\rho R u_e) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{1}{\rho}\frac{\partial\eta}{\partial x}\frac{\partial\rho}{\partial\eta}F + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{1}{Ru_e\rho}\frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}(uR)\rho + (1,3) = 0 \tag{B.34}$$

We recognize the term (5):

$$\frac{2\xi}{Ru_e\rho}F\frac{\partial}{\partial\xi}(\rho R u_e) + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{\partial\eta}{\partial x}\frac{\partial F}{\partial\eta} + (1,3,5) = 0 \tag{B.35}$$

We recognize the term (2):

$$\frac{2\xi}{Ru_e\rho}F\frac{\partial}{\partial\xi}(\rho R u_e\frac{2\xi}{2\xi}) + (1,2,3,5) = 0 \tag{B.36}$$

We expand the first term:

$$\frac{2\xi}{Ru_e\rho}\frac{\rho R u_e}{2\xi}F\frac{\partial}{\partial\xi}(2\xi) + \frac{2\xi}{Ru_e\rho}F\frac{\partial}{\partial\xi}\left(\frac{\rho R u_e}{2\xi}\right)2\xi + (1,2,3,5) = 0 \tag{B.37}$$

$$F + F + \frac{4\xi^2 F}{Ru_e\rho}\frac{\partial}{\partial\xi}\left(\frac{\rho R u_e}{\sqrt{2\xi}\sqrt{2\xi}}\right) + (1,2,3,5) = 0 \tag{B.38}$$

We recognize (6) and we expand the second term:

$$F + \frac{4\xi^2 F}{Ru_e\rho}\frac{\partial}{\partial\xi}\left(\frac{\rho R u_e}{\sqrt{2\xi}}\right)\frac{1}{\sqrt{2\xi}} + \frac{4\xi^2 F}{Ru_e\rho}\frac{\partial}{\partial\xi}\left(\frac{1}{\sqrt{2\xi}}\right)\frac{\rho R u_e}{\sqrt{2\xi}} + (1,2,3,5,6) = 0 \tag{B.39}$$

$$F + \frac{2\xi\sqrt{2\xi}F}{Ru_e\rho}\frac{\partial}{\partial\xi}\left(\frac{\rho R u_e}{\sqrt{2\xi}}\right) + \frac{4\xi^2 F}{\sqrt{2\xi}}\frac{\partial}{\partial\xi}\left(\frac{1}{\sqrt{2\xi}}\right) + (1,2,3,5,6) = 0 \tag{B.40}$$

We recognize (4):

$$F + \frac{4\xi^2 F}{\sqrt{2\xi}}\frac{\partial}{\partial\xi}\left(\frac{1}{\sqrt{2\xi}}\right) + (1,2,3,4,5,6) = 0 \tag{B.41}$$

We expand the last term:

$$F + \frac{4\xi^2 F}{\sqrt{2\xi}}\left(-\frac{1}{2}\right)(2\xi)^{-\frac{3}{2}}2 + (1,2,3,4,5,6) = 0 \tag{B.42}$$

$$F - \frac{4\xi\xi F}{\sqrt{2\xi}}\frac{1}{2\xi\sqrt{2\xi}} + (1,2,3,4,5,6) = 0 \tag{B.43}$$

$$F - F + (1,2,3,4,5,6) = 0 \tag{B.44}$$

We get get the continuity equation: (1) + (2) + (3) + (4) + (5) + (6) = 0.

To summarize, the continuity equation is:

$$2\xi\frac{\partial F}{\partial\xi} + \frac{\partial V}{\partial\eta} + F = 0 \tag{B.45}$$

B.3 Momentum equation

Let's now derive the momentum equation. We start from:

$$\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} = -\frac{dp_e}{dx} + \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right) \tag{B.46}$$

Let's remember that:

$$-\frac{dp_e}{dx} = \rho_e u_e\frac{du_e}{dx} \tag{B.47}$$

So:

$$\rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} = \rho_e\, u_e\frac{du_e}{dx} + \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right) \tag{B.48}$$

Let's change variable:

$$\rho u\frac{\partial\xi}{\partial x}\frac{\partial u}{\partial\xi} + \rho u\frac{\partial\eta}{\partial x}\frac{\partial u}{\partial\eta} + \rho v\frac{\partial\eta}{\partial y}\frac{\partial u}{\partial\eta} = \rho_e u_e\frac{\partial\xi}{\partial x}\frac{du_e}{d\xi} + \frac{\partial\eta}{\partial y}\frac{\partial}{\partial\eta}\left(\mu\frac{\partial\eta}{\partial y}\frac{\partial u}{\partial\eta}\right) \tag{B.49}$$

We now expand the derivatives:

$$\rho u\rho_e\mu_e u_e R^2\frac{\partial u}{\partial\xi} + \rho u\frac{\partial\eta}{\partial x}\frac{\partial u}{\partial\eta} + \rho v\frac{Ru_e}{\sqrt{2\xi}}\rho\frac{\partial u}{\partial\eta} = \rho_e u_e\rho_e\mu_e u_e R^2\frac{du_e}{d\xi} + \frac{Ru_e}{\sqrt{2\xi}}\rho\frac{\partial}{\partial\eta}\left(\mu\frac{Ru_e}{\sqrt{2\xi}}\rho\frac{\partial u}{\partial\eta}\right) \tag{B.50}$$

We now divide everything by ρu_e^2:

$$\frac{u}{u_e}\rho_e\mu_e R^2\frac{\partial u}{\partial\xi} + \frac{u}{u_e^2}\frac{\partial\eta}{\partial x}\frac{\partial u}{\partial\eta} + v\frac{R}{\sqrt{2\xi}u_e}\rho\frac{\partial u}{\partial\eta} = \frac{\rho_e}{\rho}\rho_e\mu_e R^2\frac{du_e}{d\xi} + \frac{R}{\sqrt{2\xi}u_e}\frac{\partial}{\partial\eta}\left(\mu\frac{Ru_e}{\sqrt{2\xi}}\rho\frac{\partial u}{\partial\eta}\right) \tag{B.51}$$

Let's now rearrange some terms and let's remember that the edge properties and ξ do not depend on η. Furthermore, $F = \frac{u}{u_e}$:

$$\rho_e\mu_e F R^2\frac{\partial u}{\partial\xi} + F\frac{\partial\eta}{\partial x}\frac{\partial F}{\partial\eta} + \frac{\rho R v}{\sqrt{2\xi}}\frac{\partial F}{\partial\eta} = \frac{\rho_e}{\rho}\rho_e\mu_e R^2\frac{du_e}{d\xi} + \frac{R^2 u_e}{2\xi}\frac{\partial}{\partial\eta}\left(\mu\rho\frac{\partial F}{\partial\eta}\right) \tag{B.52}$$

Let's now multiply by $\frac{2\xi}{\rho_e\mu_e u_e R^2} = \frac{2\xi}{\frac{\partial\xi}{\partial x}}$:

$$2\xi\frac{F}{u_e}\frac{\partial u}{\partial\xi} + \frac{2\xi}{\frac{\partial\xi}{\partial x}}F\frac{\partial\eta}{\partial x}\frac{\partial F}{\partial\eta} + \frac{2\xi}{\frac{\partial\xi}{\partial x}}\frac{\rho R v}{\sqrt{2\xi}}\frac{\partial F}{\partial\eta} = \frac{\rho_e}{\rho}\frac{2\xi}{u_e}\frac{du_e}{d\xi} + \frac{1}{\rho_e\mu_e}\frac{\partial}{\partial\eta}\left(\mu\rho\frac{\partial F}{\partial\eta}\right) \tag{B.53}$$

We now define the dimensionless parameter α which takes into account the pressure gradient in the streamwise direction:

$$\alpha = \frac{2\xi}{u_e}\frac{du_e}{d\xi} \tag{B.54}$$

By rearranging:

$$2\xi\frac{F}{u_e}\frac{\partial u}{\partial \xi} + \frac{\partial F}{\partial \eta}\left(\frac{2\xi}{\frac{\partial \xi}{\partial x}}F\frac{\partial \eta}{\partial x} + \frac{2\xi}{\frac{\partial \xi}{\partial x}}\frac{\rho R v}{\sqrt{2\xi}}\right) = \alpha\frac{\rho_e}{\rho} + \frac{\partial}{\partial \eta}\left(\frac{\mu\rho}{\rho_e\mu_e}\frac{\partial F}{\partial \eta}\right) \tag{B.55}$$

Or:

$$2\xi\frac{F}{u_e}\frac{\partial u}{\partial \xi} + \frac{\partial F}{\partial \eta}\left(\frac{2\xi}{\frac{\partial \xi}{\partial x}}\left(F\frac{\partial \eta}{\partial x} + \frac{\rho v R}{\sqrt{2\xi}}\right)\right) = \alpha\frac{\rho_e}{\rho} + \frac{\partial}{\partial \eta}\left(\frac{\mu\rho}{\rho_e\mu_e}\frac{\partial F}{\partial \eta}\right) \tag{B.56}$$

We recognize the definition of V:

$$2\xi\frac{F}{u_e}\frac{\partial u}{\partial \xi} + V\frac{\partial F}{\partial \eta} = \alpha\frac{\rho_e}{\rho} + \frac{\partial}{\partial \eta}\left(\frac{\mu\rho}{\rho_e\mu_e}\frac{\partial F}{\partial \eta}\right) \tag{B.57}$$

We define:

$$l_0 = \frac{\mu\rho}{\rho_e\mu_e} \tag{B.58}$$

So:

$$2\xi\frac{F}{u_e}\frac{\partial u}{\partial \xi} + V\frac{\partial F}{\partial \eta} = \alpha\frac{\rho_e}{\rho} + \frac{\partial}{\partial \eta}\left(l_0\frac{\partial F}{\partial \eta}\right) \tag{B.59}$$

Let's analyse the first term:

$$2\xi\frac{F}{u_e}\frac{\partial u}{\partial \xi} = 2\xi\frac{F}{u_e}\frac{\partial(u_eF)}{\partial \xi} = 2\xi\frac{F}{u_e}u_e\frac{\partial F}{\partial \xi} + 2\xi\frac{F}{u_e}F\frac{\partial u_e}{\partial \xi} = 2\xi F\frac{\partial F}{\partial \xi} + F^2\frac{2\xi}{u_e}\frac{\partial u_e}{\partial \xi} = 2\xi F\frac{\partial F}{\partial \xi} + \alpha F^2 \tag{B.60}$$

By inserting in Eq. B.59:

$$2\xi F\frac{\partial F}{\partial \xi} + \alpha F^2 + V\frac{\partial F}{\partial \eta} = \alpha\frac{\rho_e}{\rho} + \frac{\partial}{\partial \eta}\left(l_0\frac{\partial F}{\partial \eta}\right) \tag{B.61}$$

Finally, by moving αF^2 to the right hand side and by grouping the terms, we get the momentum equation:

$$2\xi F\frac{\partial F}{\partial \xi} + V\frac{\partial F}{\partial \eta} = \alpha\left(\frac{\rho_e}{\rho} - F^2\right) + \frac{\partial}{\partial \eta}\left(l_0\frac{\partial F}{\partial \eta}\right) \tag{B.62}$$

We will prove later that for an axisymmetric boundary layer, $\alpha = 1/2$, so:

$$2\xi F\frac{\partial F}{\partial \xi} + V\frac{\partial F}{\partial \eta} = \frac{1}{2}\left(\frac{\rho_e}{\rho} - F^2\right) + \frac{\partial}{\partial \eta}\left(l_0\frac{\partial F}{\partial \eta}\right) \tag{B.63}$$

is obtained.

Let's remember that all the thermodynamics and transport properties are considered at the equilibrium, and that the edge quantities are indicated with the subscript e.

B.4 Energy equation

Let's now go through the energy equation, the hardest one. We start from the energy equation:

$$\rho u\frac{\partial h}{\partial x} + \rho v\frac{\partial h}{\partial y} = u\frac{dp_e}{dx} + \mu\left(\frac{\partial u}{\partial y}\right)^2 + \frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) \tag{B.64}$$

We now want to go from the enthalpy to the temperature. We can write:

$$\frac{\partial h(p,T)}{\partial x} = \frac{\partial h}{\partial T}\frac{\partial T}{\partial x} + \frac{\partial h}{\partial p}\frac{\partial p}{\partial x} \tag{B.65}$$

$$\frac{\partial h(p,T)}{\partial y} = \frac{\partial h}{\partial T}\frac{\partial T}{\partial y} + \frac{\partial h}{\partial p}\frac{\partial p}{\partial y} = \frac{\partial h}{\partial T}\frac{\partial T}{\partial y} \tag{B.66}$$

since that $\frac{\partial p}{\partial y} = 0$. Furthermore, $p = p_e(x)$. Let's recall that at the equilibrium:

$$\frac{\partial h}{\partial T} = c_p \tag{B.67}$$

So:

$$\frac{\partial h(p,T)}{\partial x} = C_p \frac{\partial T}{\partial x} + \frac{\partial h}{\partial p}\frac{dp_e}{dx} \tag{B.68}$$

$$\frac{\partial h(p,T)}{\partial y} = C_p \frac{\partial T}{\partial y} \tag{B.69}$$

We get:

$$\rho u C_p \frac{\partial T}{\partial x} + \rho v C_p \frac{\partial T}{\partial y} = u\frac{dp_e}{dx} - \rho u \frac{\partial h}{\partial p}\frac{dp_e}{dx} + \mu\left(\frac{\partial u}{\partial y}\right)^2 + \frac{\partial}{\partial y}\left(\lambda \frac{\partial T}{\partial y}\right) \tag{B.70}$$

We are now ready to change variables. Let's remember that $\frac{dp_e}{dx} = -\rho_e u_e \frac{du_e}{dx}$:

$$\rho u C_p \frac{\partial \xi}{\partial x}\frac{\partial T}{\partial \xi} + \rho u C_p \frac{\partial \eta}{\partial x}\frac{\partial T}{\partial \eta} + \rho v C_p \frac{\partial \eta}{\partial y}\frac{\partial T}{\partial \eta} = -\rho_e u_e u \frac{du_e}{dx} + \rho_e u_e \rho u \frac{\partial h}{\partial p}\frac{du_e}{dx} + \mu\left(\frac{\partial \eta}{\partial y}\frac{\partial u}{\partial \eta}\right)^2 + \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta}\left(\lambda \frac{\partial \eta}{\partial y}\frac{\partial T}{\partial \eta}\right) \tag{B.71}$$

and:

$$\rho u C_p \frac{\partial \xi}{\partial x}\frac{\partial T}{\partial \xi} + \rho u C_p \frac{\partial \eta}{\partial x}\frac{\partial T}{\partial \eta} + \rho v C_p \frac{\partial \eta}{\partial y}\frac{\partial T}{\partial \eta} = -\rho_e u_e u \frac{\partial \xi}{\partial x}\frac{du_e}{d\xi} + \rho_e u_e \rho u \frac{\partial \xi}{\partial x}\frac{\partial h}{\partial p}\frac{du_e}{d\xi} + \mu\left(\frac{\partial \eta}{\partial y}\frac{\partial u}{\partial \eta}\right)^2 + \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta}\left(\lambda \frac{\partial \eta}{\partial y}\frac{\partial T}{\partial \eta}\right) \tag{B.72}$$

Let's expand the derivatives:

$$\begin{aligned} \rho u C_p \rho_e \mu_e u_e R^2 \frac{\partial T}{\partial \xi} + \rho u C_p \frac{\partial \eta}{\partial x}\frac{\partial T}{\partial \eta} + \rho v C_p \frac{R u_e}{\sqrt{2\xi}}\rho \frac{\partial T}{\partial \eta} = -\rho_e u_e u \rho_e \mu_e u_e R^2 \frac{du_e}{d\xi} + \\ + \rho_e u_e \rho u \rho_e \mu_e u_e R^2 \frac{\partial h}{\partial p}\frac{du_e}{d\xi} + \mu\left(\frac{R u_e}{\sqrt{2\xi}}\rho \frac{\partial u}{\partial \eta}\right)^2 + \frac{R u_e}{\sqrt{2\xi}}\rho \frac{\partial}{\partial \eta}\left(\lambda \frac{R u_e}{\sqrt{2\xi}}\rho \frac{\partial T}{\partial \eta}\right) \end{aligned} \tag{B.73}$$

Let's divide by u_e and rearrange the terms. Let's also remember that $F = \frac{u}{u_e}$ and $g = \frac{T}{T_e}$:

$$\begin{aligned} \rho \rho_e \mu_e u_e R^2 F C_p \frac{\partial T}{\partial \xi} + \rho T_e F C_p \frac{\partial \eta}{\partial x}\frac{\partial g}{\partial \eta} + \rho T_e C_p \frac{\rho v R}{\sqrt{2\xi}}\frac{\partial g}{\partial \eta} = -\rho_e^2 u_e^2 \mu_e R^2 F \frac{du_e}{d\xi} + \\ + \rho_e^2 u_e^2 \mu_e \rho F R^2 \frac{\partial h}{\partial p}\frac{du_e}{d\xi} + \mu \frac{R^2 u_e^3}{2\xi}\rho^2 \left(\frac{\partial F}{\partial \eta}\right)^2 + \frac{R^2 u_e}{2\xi}\rho T_e \frac{\partial}{\partial \eta}\left(\lambda \rho \frac{\partial g}{\partial \eta}\right) \end{aligned} \tag{B.74}$$

Let's divide by $\rho C_p T_e$:

$$\begin{aligned} \rho_e \mu_e u_e R^2 \frac{F}{T_e}\frac{\partial T}{\partial \xi} + \frac{\partial g}{\partial \eta}\left(F\frac{\partial \eta}{\partial x} + \frac{\rho v R}{\sqrt{2\xi}}\right) = -\frac{\rho_e^2}{\rho} u_e^2 \frac{\mu_e}{C_p T_e} R^2 F \frac{du_e}{d\xi} + \\ + \rho_e^2 u_e^2 \frac{\mu_e}{C_p T_e} F R^2 \frac{\partial h}{\partial p}\frac{du_e}{d\xi} + \mu \frac{R^2 u_e^3}{2\xi C_p T_e}\rho \left(\frac{\partial F}{\partial \eta}\right)^2 + \frac{R^2 u_e}{2\xi C_p}\frac{\partial}{\partial \eta}\left(\lambda \rho \frac{\partial g}{\partial \eta}\right) \end{aligned} \tag{B.75}$$

Let's multiply by 2ξ:

$$\begin{aligned} 2\xi \rho_e \mu_e u_e R^2 \frac{F}{T_e}\frac{\partial T}{\partial \xi} + \frac{\partial g}{\partial \eta} 2\xi \left(F\frac{\partial \eta}{\partial x} + \frac{\rho v R}{\sqrt{2\xi}}\right) = -2\xi \frac{\rho_e^2}{\rho} u_e^2 \frac{\mu_e}{C_p T_e} R^2 F \frac{du_e}{d\xi} + \\ + 2\xi \rho_e^2 u_e^2 \frac{\mu_e}{C_p T_e} F R^2 \frac{\partial h}{\partial p}\frac{du_e}{d\xi} + \mu \frac{R^2 u_e^3}{C_p T_e}\rho \left(\frac{\partial F}{\partial \eta}\right)^2 + \frac{R^2 u_e}{C_p}\frac{\partial}{\partial \eta}\left(\lambda \rho \frac{\partial g}{\partial \eta}\right) \end{aligned} \tag{B.76}$$

Let's recall that:

$$\alpha = \frac{2\xi}{u_e}\frac{du_e}{d\xi} \tag{B.77}$$

We can rewrite:

$$\begin{aligned} 2\xi \rho_e \mu_e u_e R^2 \frac{F}{T_e}\frac{\partial T}{\partial \xi} + \frac{\partial g}{\partial \eta} 2\xi \left(F\frac{\partial \eta}{\partial x} + \frac{\rho v R}{\sqrt{2\xi}}\right) = -\frac{2\xi}{u_e}\frac{du_e}{d\xi}\frac{\rho_e^2}{\rho} u_e^3 \frac{\mu_e}{C_p T_e} R^2 F + \\ + \frac{2\xi}{u_e}\frac{du_e}{d\xi}\rho_e^2 u_e^3 \frac{\mu_e}{C_p T_e} F R^2 \frac{\partial h}{\partial p} + \mu \frac{R^2 u_e^3}{C_p T_e}\rho \left(\frac{\partial F}{\partial \eta}\right)^2 + \frac{R^2 u_e}{C_p}\frac{\partial}{\partial \eta}\left(\lambda \rho \frac{\partial g}{\partial \eta}\right) \end{aligned} \tag{B.78}$$

So:

$$\begin{aligned}2\xi\rho_e\mu_e u_e R^2\frac{F}{T_e}\frac{\partial T}{\partial\xi}+\frac{\partial g}{\partial\eta}2\xi\left(F\frac{\partial\eta}{\partial x}+\frac{\rho vR}{\sqrt{2\xi}}\right)&=-\alpha\frac{\rho_e^2}{\rho}u_e^3\frac{\mu_e}{C_pT_e}R^2F+\\&+\alpha\rho_e^2u_e^3\frac{\mu_e}{C_pT_e}FR^2\frac{\partial h}{\partial p}+\mu\frac{R^2u_e^3}{C_pT_e}\rho\left(\frac{\partial F}{\partial\eta}\right)^2+\frac{R^2u_e}{C_p}\frac{\partial}{\partial\eta}\left(\lambda\rho\frac{\partial g}{\partial\eta}\right)\end{aligned}\tag{B.79}$$

Let's now divide by $\rho_e\mu_e u_e R^2=\frac{\partial\xi}{\partial x}$:

$$\begin{aligned}2\xi\frac{F}{T_e}\frac{\partial T}{\partial\xi}+\frac{\partial g}{\partial\eta}\frac{2\xi}{\frac{\partial\xi}{\partial x}}\left(F\frac{\partial\eta}{\partial x}+\frac{\rho vR}{\sqrt{2\xi}}\right)&=-\alpha\frac{\rho_e}{\rho}u_e^2\frac{1}{C_pT_e}F+\\&+\alpha\rho_e u_e^2\frac{1}{C_pT_e}F\frac{\partial h}{\partial p}+\frac{\mu}{\mu_e}\frac{u_e^2}{C_pT_e}\frac{\rho}{\rho_e}\left(\frac{\partial F}{\partial\eta}\right)^2+\frac{1}{C_p\rho_e\mu_e}\frac{\partial}{\partial\eta}\left(\lambda\rho\frac{\partial g}{\partial\eta}\right)\end{aligned}\tag{B.80}$$

Let's rearrange and recognize V:

$$2\xi\frac{F}{T_e}\frac{\partial T}{\partial\xi}+V\frac{\partial g}{\partial\eta}=-\alpha F\frac{\rho_e}{\rho}\frac{u_e^2}{C_pT_e}+\alpha F\rho_e\frac{u_e^2}{C_pT_e}\frac{\partial h}{\partial p}+\frac{\rho\mu}{\rho_e\mu_e}\frac{u_e^2}{C_pT_e}\left(\frac{\partial F}{\partial\eta}\right)^2+\frac{1}{C_p}\frac{\partial}{\partial\eta}\left(\frac{\lambda\rho}{\rho_e\mu_e}\frac{\partial g}{\partial\eta}\right)\tag{B.81}$$

Let's recognize l_0:

$$2\xi\frac{F}{T_e}\frac{\partial T}{\partial\xi}+V\frac{\partial g}{\partial\eta}=-\alpha F\frac{\rho_e}{\rho}\frac{u_e^2}{C_pT_e}+\alpha F\rho_e\frac{u_e^2}{C_pT_e}\frac{\partial h}{\partial p}+l_0\frac{u_e^2}{C_pT_e}\left(\frac{\partial F}{\partial\eta}\right)^2+\frac{1}{C_p}\frac{\partial}{\partial\eta}\left(\frac{\lambda\rho}{\rho_e\mu_e}\frac{\partial g}{\partial\eta}\right)\tag{B.82}$$

Let's define:

$$E=\frac{u_e^2}{C_pT_e}\tag{B.83}$$

and:

$$\chi=\frac{\lambda\rho}{\rho_e\mu_e}\tag{B.84}$$

We get:

$$2\xi\frac{F}{T_e}\frac{\partial T}{\partial\xi}+V\frac{\partial g}{\partial\eta}=-\alpha\,E\,F\frac{\rho_e}{\rho}+\alpha\,E\,F\rho_e\frac{\partial h}{\partial p}+l_0E\left(\frac{\partial F}{\partial\eta}\right)^2+\frac{1}{C_p}\frac{\partial}{\partial\eta}\left(\chi\frac{\partial g}{\partial\eta}\right)\tag{B.85}$$

Let's now expand the first term:

$$2\xi\frac{F}{T_e}\frac{\partial T}{\partial\xi}=2\xi\frac{F}{T_e}\frac{\partial T_e\,g}{\partial\xi}=2\xi\frac{F}{T_e}T_e\frac{\partial g}{\partial\xi}+2\xi\frac{F}{T_e}g\frac{\partial T_e}{\partial\xi}=2\xi\,F\,\frac{\partial g}{\partial\xi}+2\xi\frac{F}{T_e}g\frac{\partial T_e}{\partial\xi}\tag{B.86}$$

Let's substitute:

$$2\xi\,F\,\frac{\partial g}{\partial\xi}+2\xi\frac{F}{T_e}g\frac{\partial T_e}{\partial\xi}+V\frac{\partial g}{\partial\eta}=-\alpha\,E\,F\frac{\rho_e}{\rho}+\alpha\,E\,F\rho_e\frac{\partial h}{\partial p}+l_0E\left(\frac{\partial F}{\partial\eta}\right)^2+\frac{1}{C_p}\frac{\partial}{\partial\eta}\left(\chi\frac{\partial g}{\partial\eta}\right)\tag{B.87}$$

Let's rearrange:

$$2\xi\,F\,\frac{\partial g}{\partial\xi}+V\frac{\partial g}{\partial\eta}=\frac{1}{C_p}\frac{\partial}{\partial\eta}\left(\chi\frac{\partial g}{\partial\eta}\right)+E\left(l_0\left(\frac{\partial F}{\partial\eta}\right)^2+\alpha F\left(\rho_e\frac{\partial h}{\partial p}-\frac{\rho_e}{\rho}\right)\right)-2\xi\frac{F}{T_e}g\frac{\partial T_e}{\partial\xi}\tag{B.88}$$

We finally get the energy equation by remember that $\alpha=1/2$ and that $T_e=T_e(x)$:

$$2\xi\,F\,\frac{\partial g}{\partial\xi}+V\frac{\partial g}{\partial\eta}=\frac{1}{C_p}\frac{\partial}{\partial\eta}\left(\chi\frac{\partial g}{\partial\eta}\right)+E\left(l_0\left(\frac{\partial F}{\partial\eta}\right)^2+\frac{1}{2}F\left(\rho_e\frac{\partial h}{\partial p}-\frac{\rho_e}{\rho}\right)\right)-2\xi\frac{F}{T_e}g\frac{dT_e}{d\xi}\tag{B.89}$$

Let's recall that all the thermodynamics and transport properties are considered at the equilibrium, and that the edge quantities are indicated with the subscript e.

B.5 Boundary conditions

Let's now derive the boundary conditions for the transformed equations.

Let's start from:

$$u(x, y = 0) = u_w = 0 \tag{B.90}$$

$$v(x, y = 0) = v_w = 0 \tag{B.91}$$

$$T(x, y = 0) = T_w \tag{B.92}$$

$$u(x, y = \infty) = u_e(x) \tag{B.93}$$

$$T(x, y = \infty) = T_e(x) \tag{B.94}$$

Now, in term of $F = u/u_e$: for every ξ, we have that:

- If $\eta \to 0$, this corresponds to $y \to 0$, so we want $u(x, y = 0) = 0$ and $T(x, y = 0) = T_w$. So we have that $F(\xi, \eta = 0) = 0$ and $g(\xi, \eta = 0) = T_w/T_e$.
- If $\eta \to \infty$, this corresponds to $y \to \infty$, so we want $u(x, y = \infty) = u_e(x)$ and $T(x, y = \infty) = T_e(x)$. So we have that $F(\xi, \eta = \infty) = 1$ and $g(\xi, \eta = \infty) = 1$.

Let's also note that for $\eta \to 0$ we have $\frac{\partial \eta}{\partial x} \to 0$, so $V(\xi, \eta = 0) = 0$.

B.6 Alpha parameter and velocity gradient

Let's now prove that for an axisymmetric boundary layer, $\alpha = 1/2$. The following results are mainly extracted from [1].

We assume that the inviscid velocity distribtion at the outer edge of the boundary layer in the stagnation region is given by the classical incompressible result:

$$u_e = \beta\, x \tag{B.95}$$

where:

$$\beta = \left(\frac{du_e}{dx}\right)_{x=0} \tag{B.96}$$

is the velocity gradient at the stagnation point.

Let's try to understand how ξ behaves close to the stagnation point:

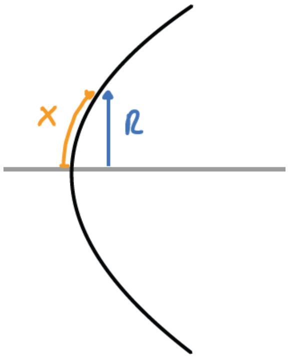

Figure B.1: Behaviour of ξ close to the stagnation point

Let's zoom in the stagnation point:

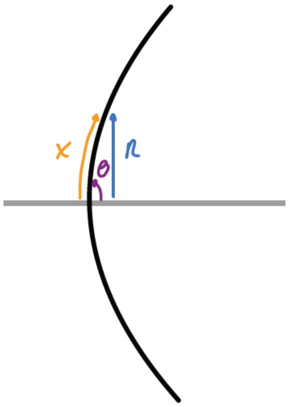

Figure B.2: Behaviour of ξ close to the stagnation point

We notice that:

$$R = x\,\sin(\theta) \tag{B.97}$$

$$\theta \rightarrow \frac{\pi}{2} \tag{B.98}$$

So:

$$R \rightarrow x \tag{B.99}$$

Let's take the limit:

$$\lim_{x\to 0} \xi = \lim_{x\to 0} \int_0^x \rho_e \mu_e u_e R^2 ds = \left(\rho_e \mu_e\right)_s \int_0^x a\, x' \left(x'\right)^2 dx' = \left(\rho_e \mu_e\right)_s a \int_0^x \left(x'\right)^3 dx' = \left(\rho_e \mu_e\right)_s a \frac{x^4}{4} \tag{B.100}$$

where the subscript s indicates that the properties are evaluated at the stagnation point. Clearly for $x = 0$, we get $\xi = 0$.

Now we can compute the derivative:

$$\left(\frac{d\xi}{dx}\right)_s = \left(\rho_e \mu_e\right)_s a x^3 \tag{B.101}$$

where the subscript s indicates that we are close to the stagnation point.

We can now compute:

$$\left(\frac{du_e}{d\xi}\right)_s = \left(\frac{du_e}{dx}\right)_s \left(\frac{dx}{d\xi}\right)_s = \frac{\left(du_e/dx\right)_s}{\left(d\xi/dx\right)_s} = \frac{a}{\left(\rho_e \mu_e\right)_s a x^3} = \frac{1}{\left(\rho_e \mu_e\right)_s x^3} \tag{B.102}$$

We can finally compute α:

$$\alpha = \frac{2\xi}{u_e} \frac{du_e}{d\xi} = \frac{2\left(\rho_e \mu_e\right)_s a \frac{x^4}{4}}{ax} \frac{1}{\left(\rho_e \mu_e\right)_s x^3} = \frac{1}{2} \tag{B.103}$$

So we have proved that for an axisymmetric boundary layer, $\alpha = 1/2$.

B.7 Transformed boundary layer equations at the stagnation point

In this section we will derive the transformed boundary layer equations at the stagnation point. We will start from the transformed equations:

$$2\xi \frac{\partial F}{\partial \xi} + \frac{\partial V}{\partial \eta} + F = 0 \tag{B.104}$$

$$2\xi F \frac{\partial F}{\partial \xi} + V \frac{\partial F}{\partial \eta} = \frac{1}{2}\left(\frac{\rho_e}{\rho} - F^2\right) + \frac{\partial}{\partial \eta}\left(l_0 \frac{\partial F}{\partial \eta}\right) \tag{B.105}$$

$$2\xi F \frac{\partial g}{\partial \xi} + V \frac{\partial g}{\partial \eta} = \frac{1}{C_p} \frac{\partial}{\partial \eta}\left(\chi \frac{\partial g}{\partial \eta}\right) + E\left(l_0 \left(\frac{\partial F}{\partial \eta}\right)^2 + \frac{1}{2} F \left(\rho_e \frac{\partial h}{\partial p} - \frac{\rho_e}{\rho}\right)\right) - 2\xi \frac{F}{T_e} g \frac{dT_e}{d\xi} \tag{B.106}$$

We are going to assume that, for physical reasons, the derivatives $\frac{\partial F}{\partial \xi}$ and $\frac{\partial g}{\partial \xi}$ are finite when $\xi \to 0$. This means that the velocity and temperature gradients are finite at the stagnation point. Now, we want to take the limit as $\xi \to 0$ of the transformed equations.

Let's start from the continuity equation: when $\xi \to 0$, we have:

$$2\xi \frac{\partial F}{\partial \xi} \to 0 \tag{B.107}$$

So the continuity equation becomes:

$$\frac{\partial V}{\partial \eta} + F = 0 \tag{B.108}$$

Let's now consider the momentum equation: when $\xi \to 0$, we have:

$$2\xi F \frac{\partial F}{\partial \xi} \to 0 \tag{B.109}$$

So the momentum equation becomes:

$$V \frac{\partial F}{\partial \eta} = \frac{1}{2}\left(\frac{\rho_e}{\rho} - F^2\right) + \frac{\partial}{\partial \eta}\left(l_0 \frac{\partial F}{\partial \eta}\right) \tag{B.110}$$

Let's now consider the energy equation: when $\xi \to 0$, we have:

$$2\xi F \frac{\partial g}{\partial \xi} \to 0 \tag{B.111}$$

$$2\xi \frac{F}{T_e} g \frac{dT_e}{d\xi} \to 0 \tag{B.112}$$

$$E = \frac{u_e^2}{C_p T_e} \to 0 \tag{B.113}$$

By assuming that $\frac{\partial h}{\partial p}$ is finite, the energy equation becomes:

$$V \frac{\partial g}{\partial \eta} = \frac{1}{C_p} \frac{\partial}{\partial \eta}\left(\chi \frac{\partial g}{\partial \eta}\right) \tag{B.114}$$

Regarding the boundary conditions, nothing changes:

$$F(0) = 0 \tag{B.115}$$

$$g(0) = T_w/T_e \tag{B.116}$$

$$V(0) = 0 \tag{B.117}$$

$$F(\infty) = 1 \tag{B.118}$$

$$g(\infty) = 1 \tag{B.119}$$

B.8 Heat flux formula

Let's now derive the formula for the heat flux at the wall. In general:

$$q = \lambda \frac{\partial T}{\partial y} = \lambda \frac{\partial \eta}{\partial y} \frac{\partial T}{\partial \eta} = \lambda \frac{R u_e}{\sqrt{2\xi}} \rho T_e \frac{\partial g}{\partial \eta} \tag{B.120}$$

Now, let's focus on the wall, and in the stagnation point:

$$q_w = \lambda_w \rho_w T_e \left(\frac{\partial g}{\partial \eta}\right)_w \frac{x\, a\, x}{\sqrt{2\, \rho_e \mu_e a x^4/4}} = \lambda_w \rho_w T_e \left(\frac{\partial g}{\partial \eta}\right)_w \sqrt{\frac{2\, a}{\rho_e \mu_e}} \tag{B.121}$$

where all the proprerties have to be evaluated at the stagnation point. Let's recall that a is the velocity gradient at the stagnation point, which will be also indicated as β from now on. So we get the final version:

$$q_w = \lambda_w \rho_w T_e \left(\frac{\partial g}{\partial \eta}\right)_w \sqrt{\frac{2\, \beta}{\rho_e \mu_e}} \tag{B.122}$$

Appendix C

Velocity gradient considerations

In this appendix we will discuss the velocity gradient laws that we used in the code.

The value of β is important to correctly assesment of the heat flux rate. In most hypersonic applications, it generally suffices to apply the Newtonian theory. In subsonic plasma facilities, however, this is no longer possible and accurate expressions of the velocity gradient have to be provided.

Clearly the gradient of the velocity depends on the velocity U and on a characteristic distance of the model R_m. This is why, to get a general expression, we look for the paramenter:

$$\beta \frac{R_m}{U} \tag{C.1}$$

which is called **Stagnation variable** and it's a dimensionless number.

Let's consider the following cases:

- **Subsonic hemisphere:** the velocity gradient over the nose of a spherical body of radius R_m is a flow of velocity U can be found from classical potential theory[4]:

$$\beta \frac{R_m}{U} = \frac{3}{2} \tag{C.2}$$

- **Subsonic cylinder:** the velocity gradient a cylinder with radius radius R_m is a flow of velocity U can be found from classical potential theory[4]:

$$\beta \frac{R_m}{U} = 2 \tag{C.3}$$

- **Subsonic ESA sample holder:** This is for a typical geometry of the ESA sample holder, idealised by a cylinder of radius R_m with a flat face placed perpendicularly to the flow. This is our main case of interest, since this geometry is similar to the one of the Plasmatron X wind tunnel. From Kolesnikov[8], the velocity gradient is expressed as:

$$\beta = \frac{U}{R_m^*} \tag{C.4}$$

  where R_m^* is the effective nose radius of the sample holder and it has to be determined.

  From calculated and experimental data on the velocity gradients at the outer edge of the boundary layer, Kolesnikov found:

$$\begin{cases} R_m^* = R_m\left(2 - \sigma - 1.68(\sigma-1)^2 - 1.28(\sigma-1)^3\right) \quad \text{for} \quad \sigma \leq 1 \\ R_m^* = R_c \quad \text{for} \quad \sigma > 1 \end{cases} \tag{C.5}$$

  where:

$$\sigma = \frac{R_m}{R_c} \tag{C.6}$$

  where R_c is the jet radius of the flow investing the sample holder. The velocity used in the calculation of the velocity gradient is the freestream velocity of the jet U, on the axis of the jet.

  We can now compute $\beta = \frac{U}{R_m^*}$:

$$\begin{cases} \beta = \frac{U}{R_m(2-\sigma-1.68(\sigma-1)^2-1.28(\sigma-1)^3)} \quad \text{for} \quad \sigma \leq 1 \\ \beta = \frac{U}{R_c} \quad \text{for} \quad \sigma > 1 \end{cases} \tag{C.7}$$

And the stagnation variable $\beta \frac{R_m}{U}$ is:

$$\begin{cases} \beta \frac{R_m}{U} = \frac{1}{2-\sigma-1.68(\sigma-1)^2-1.28(\sigma-1)^3} & \text{for} \quad \sigma \leq 1 \\ \beta \frac{R_m}{U} = \sigma \quad \text{for} \quad \sigma > 1 \end{cases} \tag{C.8}$$

C.1 Numerical solution for the heat flux

After choosing a discretization step, the system of equation can be solved iteratively[4, 3, 10]

- Step 1: an initial guess is made for the velocity profile F and for the temperature profile g. The *Hartree profile* can be chosen as initial guess.
- Step 2: V is obtained by integration of the continuity equation:

$$V = -\int_0^\eta F(\eta')d\eta' \tag{C.9}$$

  with $V(0) = 0$. Computations with Simpson's composite formulas provide sufficient accuracy
- Step 3: The value of F is updated by solving the momentum equation.
- Step 4: The value of g is updated by solving the energy equation.
- Step 5: these steps are repeated until convergence is reached. In particular, we stop when a maximum number of iterations is reached or when the absolute error between two consecutive iterations is under a certain tolerance.

Since we have to discretize our domain, we cannot reach $\eta \to \infty$, and a certain value η_δ as too be chosen. Usually $\eta_\delta = 6$ can be safely chosen[4].

Step 3 and 4 involve the solution of a nonlinear ordinary differential equation in the η variable. Let $w(\eta)$ be either F or g:

$$a(w)\frac{d^2w}{d\eta^2} + b(w)\frac{dw}{d\eta} + c(w)w = d(w) \tag{C.10}$$

with boundary conditions:

$$\begin{cases} w(0) = w_0 \\ w(\eta_\delta) = w_\delta \end{cases} \tag{C.11}$$

It is possible to remove the nonlinearity by evaluating the coefficient a,b,c,d from the previous iterations (i.e. the values from F and g from the previous iterations are used to compute the coefficients). This is justified by the iterative procedure.

By doing so, the equation becomes linear:

$$a\frac{d^2w}{d\eta^2} + b\frac{dw}{d\eta} + cw = d \tag{C.12}$$

The coefficients are derived now:

C.1.1 Momentum equation

The momentum equation is:

$$V\frac{dF}{d\eta} = \frac{1}{2}\left(\frac{\rho_e}{\rho} - F^2\right) + \frac{1}{\partial\eta}\left(l_0\frac{dF}{d\eta}\right) \tag{C.13}$$

or:

$$\frac{1}{d\eta}\left(l_0\frac{dF}{d\eta}\right) - V\frac{dF}{d\eta} + \frac{1}{2}\left(\frac{\rho_e}{\rho} - F^2\right) = 0 \tag{C.14}$$

By expanding the derivative, we get:

$$l_0\frac{d^2F}{d\eta^2} + \frac{d\,l_0}{d\eta}\frac{dF}{d\eta} - V\frac{dF}{d\eta} + \frac{1}{2}\left(\frac{\rho_e}{\rho} - F^2\right) = 0 \tag{C.15}$$

or:

$$l_0\frac{d^2F}{d\eta^2} + \left(\frac{d\,l_0}{d\eta} - V\right)\frac{dF}{d\eta} = \frac{1}{2}\left(\frac{F^2 - \rho_e}{\rho}\right) \tag{C.16}$$

So:

$$a = l_0 \tag{C.17}$$

$$b = \frac{d\,l_0}{d\eta} - V \tag{C.18}$$

$$c = 0 \tag{C.19}$$

$$d = \frac{1}{2}\left(F^2 - \frac{\rho_e}{\rho}\right) \tag{C.20}$$

C.1.2 Energy equation

The energy equation is:

$$V\frac{dg}{d\eta} = \frac{1}{C_p}\frac{1}{d\eta}\left(\chi\frac{dg}{d\eta}\right) \tag{C.21}$$

or:

$$\frac{1}{C_p}\frac{1}{d\eta}\left(\chi\frac{dg}{d\eta}\right) - V\frac{dg}{d\eta} = 0 \tag{C.22}$$

By expanding the derivative, we get:

$$\frac{\chi}{C_p}\frac{d^2g}{d\eta^2} + \frac{1}{C_p}\frac{d\chi}{d\eta}\frac{dg}{d\eta} - V\frac{dg}{d\eta} = 0 \tag{C.23}$$

or:

$$\frac{\chi}{C_p}\frac{d^2g}{d\eta^2} + \left(\frac{1}{C_p}\frac{d\chi}{d\eta} - V\right)\frac{dg}{d\eta} = 0 \tag{C.24}$$

So:

$$a = \frac{\chi}{C_p} \tag{C.25}$$

$$b = \frac{1}{C_p}\frac{d\chi}{d\eta} - V \tag{C.26}$$

$$c = 0 \tag{C.27}$$

$$d = 0 \tag{C.28}$$

C.1.3 andiamo avanti

Since Eq. **??** must be solved numerically, the derivatives must be replaced properly. Instead of the conventional difference methods, *Multi-point methods* allow "higher accuracy with the same number of grid points, by incorporating the form of the differential equation into a higher-order Taylor expansion"[10]. The approach followed by is the following (the same used in [3, 4, 10]): a Lagrangian polynomial of degree is used:

$$w(\eta) = \frac{1}{2}w_{n+1}(t^2+t) + w_n(1-t^2) + \frac{1}{2}w_{n-1}(t^2-t) + \alpha t(1-t^2) + \beta t^2(1-t^2) \tag{C.29}$$

where the running variable is:

$$t = \frac{\eta - \eta_n}{\Delta\eta} \tag{C.30}$$

and w_k represent the solution at the support point k of the discretization. This approach uses three adjacent points, at $t = -1, 0, 1$, and maps them to the $w(\eta)$ function values w_{n-1}, w_n, w_{n+1}. It is important to notice that two free parameters are present, α and β. These parameters are the ones which provide an higher order accuracy.

Now, the following steps are taken[10, 3]:

- Step 1: the first and second derivative of $w(\eta)$ are computed. They will depend on w_{n-1}, w_n, w_{n+1}, α and β.
- Step 2: The derivative are inserted in the linear differential equation. Three equations are written by requiring that the differential equation holds for $t = -1, 0, 1$.
- Step 3: The free parameters α and β are eliminated by multiplying the three equations for proper determinants, and by adding them together.
- Step 4: A simple equation in the form:

  $$A_n w_{n+1} + B_n w_n + C_n w_{n-1} = D_n \tag{C.31}$$

  is obtained, where A_n, B_n, C_n, D_n are depend on a_n,b_n,c_n,d_n,$\Delta\eta$.
- Step 5: Repeating this procedure for every support point create a tridiagonal system, which can be efficiently solved by Thomas's algorithm. Particular attention has to be paid for the first and last row, where the boundary conditions have to be considered.
- Step 6: The system is solved and w_n is determined for every support point.

This procedure is applied for F and g, with proper coefficient.

C.1.4 Coefficients

The derivatives of Eq. (C.29) are computed as follows [**German˙paper**, **barbante˙article**, 3]:

$$\frac{\mathrm{d}\, w}{\mathrm{d}\, \eta} = \frac{1}{2}\frac{w_{n+1}}{\Delta\eta}(2t+1) - 2\frac{w_n}{\Delta\eta}t + \frac{1}{2}\frac{w_{n-1}}{\Delta\eta}(2t-1) + \frac{\Gamma}{\Delta\eta}(1-3t^2) + \frac{\Lambda}{\Delta\eta}2t(1-2t^2) \tag{C.32}$$

$$\frac{\mathrm{d}^2\, w}{\mathrm{d}\, \eta^2} = \frac{w_{n+1}}{\Delta\eta^2} - 2\frac{w_n}{\Delta\eta^2} + \frac{w_{n-1}}{\Delta\eta^2} - 6\frac{\Gamma}{\Delta\eta^2}t + 2\frac{\Lambda}{\Delta\eta^2}(1-6t^2) \tag{C.33}$$

using the relationship:

$$\frac{\mathrm{d}}{\mathrm{d}\, \eta} = \frac{1}{\Delta\eta}\frac{\mathrm{d}}{\mathrm{d}\, t} \tag{C.34}$$

To eliminate the two free parameters, Γ and Λ, this approach requires satisfying the differential equation, Eq. (**??**), at three collocation points. For calculating approximate values of w at these points, it is logical to choose $n-1$, n, and $n+1$, corresponding to $t=-1,0,1$, respectively [**German˙paper**, **barbante˙article**, 3]. From this point forward, $c=0$ is considered, as this is the case for $F(\eta)$ and $g(\eta)$.
After defining:

$$\bar{a} = \frac{a}{\Delta\eta^2}\,, \qquad \bar{b} = \frac{b}{\Delta\eta}\,, \qquad \bar{d} = d \tag{C.35}$$

the following matrix:

$$K_{ij} = \begin{bmatrix} \bar{a}_{n-1} - 1.5\bar{b}_{n-1} & -2\bar{a}_{n-1} + 2\bar{b}_{n-1} & \bar{a}_{n-1} - 0.5\bar{b}_{n-1} & 6\bar{a}_{n-1} - 2\bar{b}_{n-1} & -10\bar{a}_{n-1} + 2\bar{b}_{n-1} \\ \bar{a}_{n} - 0.5\bar{b}_{n} & -2\bar{a}_{n} & \bar{a}_{n} + 0.5\bar{b}_{n} & \bar{b}_{n} & -2\bar{a}_{n} \\ \bar{a}_{n+1} + 0.5\bar{b}_{n+1} & -2\bar{a}_{n+1} - 2\bar{b}_{n+1} & \bar{a}_{n+1} + 0.5\bar{b}_{n+1} & -6\bar{a}_{n+1} - 2\bar{b}_{n+1} & -10\bar{a}_{n+1} - 2\bar{b}_{n+1} \end{bmatrix} \tag{C.36}$$

and column vectors:

$$J_i = \begin{bmatrix} \bar{d}_{n-1} & \bar{d}_n & \bar{d}_{n+1} \end{bmatrix}^T \tag{C.37}$$

$$X_j = \begin{bmatrix} w_{n-1} & w_n & w_{n+1} & \Gamma & \Lambda \end{bmatrix}^T \tag{C.38}$$

can be obtained, so that:

$$K_{ij}\, X_j = J_i \tag{C.39}$$

represent the three imposed conditions (a derivation with $c \neq 0$ can be found in [**German˙paper**]). The system is under-determined, but a single equation can be obtained using Gaussian elimination. The following determinants are defined [**German˙paper**]:

$$\Delta_{n-1} = K_{34}K_{25} - K_{24}K_{35}\,, \qquad \Delta_n = K_{14}K_{35} - K_{34}K_{15}\,, \qquad \Delta_{n+1} = K_{24}K_{15} - K_{14}K_{25} \tag{C.40}$$

By multiplying each row of K_{ij} by its corresponding determinant and summing the three rows, a single row is obtained, representing Eq. (C.31), where:

$$\begin{aligned} A_n &= K_{13} \cdot \Delta_{n-1} + K_{23} \cdot \Delta_n + K_{33} \cdot \Delta_{n+1} \\ B_n &= K_{12} \cdot \Delta_{n-1} + K_{22} \cdot \Delta_n + K_{32} \cdot \Delta_{n+1} \\ C_n &= K_{11} \cdot \Delta_{n-1} + K_{21} \cdot \Delta_n + K_{31} \cdot \Delta_{n+1} \\ D_n &= \bar{d}_{n-1} \cdot \Delta_{n-1} + \bar{d}_n \cdot \Delta_n + \bar{d}_{n+1} \cdot \Delta_{n+1} \end{aligned} \tag{C.41}$$

Assuming $N+1$ points on the grid, where $n = 0, 1, \ldots, N$, the boundary conditions imply that $N-1$ internal nodes must be determined, as $n=0$ and $n=N$ are specified. Equation (C.31) can be applied for each internal node, with slight modifications for $n=1$ and $n=N-1$, where the boundary conditions must be incorporated. Integrating these conditions yields the following equations, respectively:

$$\begin{aligned} A_1\, w_2 + B_1\, w_1 &= D_1 - C_1\, w_0 \\ B_{N-1}\, w_{N-1} + C_{N-1}\, w_{N-2} &= D_{N-1} - A_{N-1}\, w_N \end{aligned} \tag{C.42}$$

Finally, a tridiagonal system for the $N-1$ internal nodes is obtained, with dimension $(N-1) \times (N-1)$, which can be efficiently solved using Thomas' algorithm:

$$\begin{bmatrix} B_1 & A_1 & & & & \\ C_2 & B_2 & A_2 & & & \\ & C_3 & B_3 & A_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & C_{N-2} & B_{N-2} & A_{N-2} \\ & & & & C_{N-1} & A_{N-1} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{N-2} \\ w_{N-1} \end{bmatrix} = \begin{bmatrix} D_1 - C_1\, w_0 \\ D_2 \\ D_3 \\ \vdots \\ D_{N-2} \\ D_{N-1} - A_{N-1}\, w_N \end{bmatrix} \tag{C.43}$$

C.2 Mutation++ links for C++, python and Fortran

After cloning Mutation++ from the repository, navigate to the directory where the repo has been cloned.

Procedure described for Ubuntu 20.04.6, tried on WSL on Windows 11. Procedure taken from `https://github.com/mutationpp/Mutationpp/blob/master/docs/installation.md` and from `https://github.com/mutationpp/Mutationpp/issues/189` for the Fortran wrapper.

C.2.1 C++

This is the first step to take.

A c++ compiler compatible with C++11 revision of the language is needed. CMake is needed (¿=3.5).

Be sure to be in the Mutation++ directory, and run:

```
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX:PATH=$(realpath ../install) ..
make -j N install
```

where N is the number of cores to use for the compilation. Mutation++ will be installed in the *install* directory, which you can find in the Mutation++ directory.

Some Linux distributions install libraries in *$CMAKE_INSTALL_PREFIX/lib64* instead of *$CMAKE_INSTALL_PREFIX/lib*. For this reason, after you compiled and installed the code, check if you have a lib or lib64 LIBDIR. Next, add the following lines to your *.bashrc* file in your home directory.

```
export MPP_DIRECTORY=path_to_mutation++_directory
export MPP_DATA_DIRECTORY=$MPP_DIRECTORY/data
export PATH=$MPP_DIRECTORY/install/bin:$PATH
export LD_LIBRARY_PATH=$MPP_DIRECTORY/install/<LIBDIR>:$LD_LIBRARY_PATH
# Change <LIBDIR> with ‘lib‘ or ‘lib64‘
```

Importation with CMake

In order to use Mutation++ in your CMake project, make sure your CMakeLists.txt file is in this form:

```
cmake_policy(SET CMP0048 NEW) #policy for version warnings, if needed
cmake_minimum_required(VERSION 3.0) #minimum version of cmake
cmake_policy(SET CMP0076 NEW) # other policy for version warnings, if needed
project(project_name VERSION 1.0) # declare the project name and version, if needed
set(CMAKE_CXX_STANDARD 11) # set the C++ standard
set(CMAKE_CXX_STANDARD_REQUIRED True) # set the C++ standard as required
add_subdirectory(classes) # add the classes directory #add subdirectories if needed
find_package(mutation++ REQUIRED) # THIS IS THE IMPORTANT LINE
add_executable(project_name main.cpp) # add the executable
target_sources(project_name PUBLIC presentation.h presentation.cpp) #add other sources
# Link the libraries (subdirectories)
target_link_libraries(project_name PUBLIC classes)
target_link_libraries(Neboula PRIVATE mutation++::mutation++)  # THIS IS THE IMPORTANT LINE
# Include the directories
target_include_directories(Neboula PUBLIC
                        "${PROJECT_BINARY_DIR}"  # for Configs.h and other files in the binary directory
                        "${PROJECT_SOURCE_DIR}/classes" # for classes directory
                        )
```

In subdirectories, you can have a CMakeLists.txt file like this:

```
add_library(classes parameters_class.h parameters_class.cpp)
target_sources(classes PUBLIC flow_conditions_class.h flow_conditions_class.cpp)
find_package(mutation++ REQUIRED)
target_link_libraries(classes PRIVATE mutation++::mutation++)
```

Now you can simply import Mutation++ in your C++ code with:

```
#include "mutation++.h"
```

C.2.2 Python

P.S. python installation is needed, with pip.

It is now possible to install the python wrapper for Mutation++. Before doing so, you can edit the wrapper files to add more functions if needed. The wrapper files are located in *interface/python/src*. In particular, you will be interested in the *pyMixture.cpp* file. Follow the file structure to add more functions, referring to the C++ code in the *src* directory for the functions you want to add. The syntax is:

```
.def("py_fun_name", C++_fun_name,
       "Fun description")
```

where C++_fun_name could be something like *&Mutation::Mixture::speciesName*

When satisfied, simply run:

```
pip install .
```

in Mutation++ directory.

More specifically, e.g., you have a certain python version, you can run:

```
python3 -m pip install .
```

in Mutation++ directory.

If you get an error related to PEP517 update essential packages:

```
pip install --upgrade pip setuptools wheel
```

If other errors pop up, see the installation guide for PlasFlowSolver provided in the previous sections of this manual to install the necessary dependencies.

Now you can simply import Mutation++ in your python code with:

```
import mutationpp
```

or something like:

```
import mutationpp as mpp
```

C.2.3 Fortran

P.S. Fortran compiler is needed, with CMake.

Now, before installing the Fortran wrapper, you can edit the wrapper files to add more functions if needed. The wrapper files are located in *interface/fortran/*. Here the wrapper is made of 3 files, *cwrapper.h*, *cwrapper.cpp* and *cwrapper_interface.f90*. Follow the files structure to add more functions, referring to the C++ code in the *src* directory for the functions you want to add. For the *cwrapper.h* file the syntax is:

```
double NAME_MANGLE(soret_thermal_conductivity)();
```

where we have the type (double) and the function name (soret_thermal_conductivity). The *cwrapper.cpp* file will have the syntax:

```
double NAME_MANGLE(soret_thermal_conductivity)()
{
    return p_mix->soretThermalConductivity();
}
```

where the first line is the same as the one in the *cwrapper.h* file, and the rest is the C++ function call. The *cwrapper_interface.f90* file will have the syntax:

```
real(kind=8) function mpp_soret_thermal_conductivity()
end function
```

where the type to return is specified (real(kind=8)) and the function name is specified (mpp_soret_thermal_conductivity).

When satisfied, go in the Mutation++ directory and run:

```
mkdir build
cd build
ccmake ..
```

In the CMake interface, set *BUILD_FORTRAN_WRAPPER* to ON, by moving with the arrows and pressing enter when you are on the option. Then, press c to configure. Press it again if the *g* option is not available. Then, press g to generate the makefile. Finally, run:

```
cmake --build .
```

or the previous ninja command.

The library will be installed in the *install* directory.

Importation with CMake

The important lines here are:

```
project("myapp" LANGUAGES "Fortran") #create the project "myapp" in Fortran
file(GLOB project_SRC CONFIGURE_DEPENDS "${CMAKE_SOURCE_DIR}/src/*.f90") #get the source files
if(NOT TARGET mutation++::fortran_interface)
find_package(mutation++ REQUIRED)
endif()
```

before adding the executables. Then:

```
add_executable("myapp"  ${project_SRC}) #add the executables
target_link_libraries("myapp"
PRIVATE
    mutation++::fortran_interface
) #link the libraries
target_include_directories("myapp"
PUBLIC
    $<TARGET_PROPERTY:mutation++::fortran_interface,Fortran_MODULE_DIRECTORY>
)
```

Now you can simply import Mutation++ in your Fortran code with:

```
use mutationpp
```