Politecnico di Torino

LIÈGE université

Master in Aerospace Engineering

Stabilization strategies for high-order simulations of plasma flows

Supervisors:
Prof. Domenic D'Ambrosio
Prof. Koen Hillewaert

Candidate:
Sebastiano Lassandari

Politecnico di Torino - Anno Accademico 2025/2026

Abstract

High resolution scale-resolving simulations have recently received more interest due to the increase in available computational power. For such simulations, high-order accurate schemes have significant benefits since they are much less subjected to numerical error and therefore less likely to suppress flow instabilities, but they also present new challenges and issues to be aware of compared to more traditional ones. In particular, the numerical interpolation around sharp discontinuities may be a source of instabilities and numerical oscillations, which can lead to non-physical results, such as negative density, pressure or temperature, and cause the simulation to crash. In this work different strategies and methods are presented to stabilise a high-resolution simulation of non-neutral plasmas subjected to a magnetic field - such as also encountered in HET -, the canonical Penning discharge case study characterized by expansion into a vacuum at the boundary. These range from shock-capturing methods, such as artificial viscosity, to a smoother approach to the transition between an outflow and an inflow in case of tangential flow, allowing to mitigate the issues related to the over-imposition of boundary conditions. The results show the improvement on the stability of the simulation due to the applied methods, reducing the appearance of sporadic oscillations and of non-physical results, until an enduring stabilization is established.

Contents

Figures

Tables

Symbols

a	Sound speed
A	Jacobian Matrix
$\boldsymbol{B}$	Magnetic field
B	Magnetic field magnitude
c	Light velocity, characteristic variables
C	Collision term
$\mathbf{d}$	Diffusive flux
e	Energy density, generic mesh element
$\boldsymbol{E}$	Electric field
f	Generic face, boundary smoothen function
$\mathbf{f}$	Convective flux
$\boldsymbol{f}$	Volume force vector
$\boldsymbol{F}$	Generic force vector
$\boldsymbol{F}_L$	Lorentz force
$\mathbf{g}$	Flux components
h	enthalpy density
$\mathcal{H}$	Flux function
I	Identity matrix
$\boldsymbol{I}$	Current density
m	Mass
m_u	Mass of a single particle
M	Mach number
n	Number particle density
$\boldsymbol{n}$	Normal vector
p	Pressure, Interpolation order
$\boldsymbol{q}$	Thermal flux vector
q	Conserved quantities
Q	Charge density

Q_0	Elementary charge
s	Source term
S	Surface, artificial viscosity smoothen indicator
t	Time
T	Temperature
u	Velocity component in the x direction, generic velocity component
$\boldsymbol{u}$	Solution state vector
$\boldsymbol{u}_h$	Approximated solution
$\mathcal{U}$	Space vector
$\mathcal{U}_h$	Trial space
v	Velocity component in the y direction, generic test function, generic velocity
$\boldsymbol{v}$	Particle velocity vector
$\boldsymbol{V}$	Mean velocity vector
V	Electric potential, Volume
$\mathcal{V}$	Test space
w	Velocity component in the z direction
α	Artificial viscosity sensor
β	Shock Strength
γ	Interface flux
δ	Shock Thickness
$\mathcal{E}$	Mesh
ε	Artificial viscosity coefficient
ε_0	Vacuum permittivity, maximum artificial viscosity sensor
λ	eigenvalues, characteristic velocity
Λ	Diagonal matrix
μ	Viscosity coefficient
ρ	Density, diffusive spectral radius
σ	Penalty term
ϕ	Shape functions
Ω	Domain

Superscripts and subscripts

$_e$	Electrons, generic element
$_f$	Generic face
$_i$	Ions
$_m$	Generic variable index
$_{ref}$	Reference value
$_t$	Time derivative
$_{th}$	Thermal
$_{\boldsymbol{v}}$	Velocity derivate
$_{vac}$	Vacuum
$_{\boldsymbol{x}}$	Spatial derivative
$_\alpha$	Generic species
$_\perp$	Perpendicular component
$_\parallel$	Parallel component

Introduction

The study of plasma has seen an increase of interest in recent years, especially in the aerospace field. From the formation of a plasma sheathing around a re-entry body to the performances of propulsion systems that use plasma flows. Precise and reliable numerical simulations become of fundamental importance to increase our knowledge of plasma behaviour while also reducing the research and development costs.

For a long period of time, due to the limits imposed onto us by the computational resources at our disposal, the simulations that were carried out in this field were often simplified or too inaccurate to properly characterize some of the aspects of plasma flows. However, with the improvement that computer technology has seen in the last years, new methods have been explored, in particular the Discontinuous Galerkin (DG) has gained a lot of attention. The Discontinuous Galerkin Method is a variant of the Finite-Element Method (FEM), that allows to more complex simulations, because it combines high order accuracy and geometrical flexibility, it also incorporates elements of Finite Volume Methods making it more robust due to numerical conservation.

The simulations presented in this work have been obtained performed ForDGe, a software developed in the university of Liège that make use of Discontinuous Galerkin Method to preform high order simulations of gasses and plasmas. This last case is the one we are interested in. Different models can be used describe the behaviour of plasmas, such as Magnetohydrodynamics (MHD), Particle-In-Cell (PIC) or Multi-Fluid. This latter model is the one implemented in ForDGe, and is characterized by considering the plasma as a mixture of different fluids (electrons, ions and neutral atoms), each governed by its own conservation laws and coupled by electro-magnetic fields. This model allows for a reduction in the computational cost usually required by PIC or other kinetic models, without losing some interesting phenomena, such as ion-acoustic waves, that the MHD model fails to capture.

Chapter 1

Plasma-Dynamics

1.1 Plasma characteristics and Conservation Laws

This section discusses the peculiar properties of plasmas and the equations that can be used to describe their behaviour in non-rarefied conditions. The assumptions made and the equations shown here, together with the electric potential equation in sec. 1.2, are made according to [2].

Plasma can be considered a highly excited state of matter, composed of individual atoms, ions and electrons and whose dynamic is dominated by the electro-magnetic forces. In particular it can said that a plasma is composed by the union and interaction between its charged particles and the fields that they generate[1].

An example that can be found in nature of plasma is represented by ionized gases, a gas mixture composed of atoms, both neutral and charged ones, and free electrons. Different challenges arise from this composition of the gas, in particular the difference in mass between heavy particles, atoms and ions, and electrons, with the mass of the latter possibly being 4 or more orders of magnitude smaller than the others. This result, for a plasma in thermal equilibrium, in large differences in thermal speed, defined as:

$$\begin{aligned} v_{th,e} &= \sqrt{\frac{3k_B T_e}{m_e}} \\ v_{th,i} &= \sqrt{\frac{3k_B T_i}{m_i}} \end{aligned} \tag{1.1}$$

where v_{th} is the thermal velocity, k_B is the Boltzmann constant, T is the temperature of a single species and m its mass (the subscript e indicates the electrons and the

subscript i the ions). It can be noticed that, since $m_i \gg m_e$, it can be expected that $v_{th,e} \gg v_{th,i}$. This difference creates issues to the progression in time of the simulation, especially for high-order simulations, because electrons would need, in order to avoid creating instabilities in the solution, a time step much smaller compared to characteristic time of ions related phenomena. Moreover temperature itself is affected by this difference, as it is a macroscopic quantity to express mean random kinetic energy of each particle. Thermal equilibrium on a microscopic scale is reached through the collision between particles, however the difference in mass result in a very inefficient energy transfer in electron-ion collisions, resulting in plasmas - in particular rarefied ones since the probability of collision is even scarcer - where electrons and ions exist at different temperatures[1].

A thermodynamic system, such as a plasma, can be described with different approaches, one being the Boltzmann equations, which allows us to study, through a statistical approach, the evolution of the system through time. These equations consider a distribution function $f_\alpha(t, \boldsymbol{x}, \boldsymbol{v})$, that describes the probability of finding a particle of species α with a velocity $\boldsymbol{v}$, in a position $\boldsymbol{x}$ at a time t. A generalized version of the Boltzmann equations is presented in Eq. (1.2), where $\boldsymbol{F}_\alpha$ are the forces applied to the particle α, m_α its mass, and C_α the collision term.

$$\partial_t f_\alpha + \boldsymbol{v} \cdot \partial_{\boldsymbol{x}} f_\alpha + \left(\frac{\boldsymbol{F}_\alpha}{m_\alpha} \right) \cdot \partial_{\boldsymbol{v}} f_\alpha = C_\alpha \tag{1.2}$$

Conservation laws can be derived from the Boltzmann equation by taking the velocity moments of the distribution function $f_\alpha(t, \boldsymbol{x}, \boldsymbol{v})$ and integrating it over the entire velocity space. In particular, considering the Euler equations, that are the equations used in this work, they are obtained by considering the moments up to the second order, $(1,\ \boldsymbol{v},\ \frac{1}{2}v^2)$, related respectively to the macroscopic quantities density, momentum and energy. The Euler equations are shown in Eq.(1.4). Some assumption must be made to use this specific set of equations:

- the fluid taken in consideration has to respect the assumption of continuum, which means that the *mean free path* l_0 - the average distance between two collisions - of the particles composing the plasma is small compared to the characteristic length L of the domain that we want to study. This condition allows us to switch from having to consider each individual particle velocity and collision to use macroscopic variables, such as density, pressure or temperature[14]. The continuum assumption can be verified using the Knudsen

number, defined as the ratio:

$$Kn = \frac{l_0}{L} \tag{1.3}$$

A fluid can be considered as a continuum when $Kn < 10^{-2}$. This condition is valid if the characteristic length is sufficiently large or the number of particles per unit volume n is very high, as the mean free path is $l_0 \propto \frac{1}{n}$;

- the viscosity[1] is very low and, therefore, can be assumed to be zero;
- the heat flux is assumed to not be present ($\boldsymbol{q} = 0$).

$$\begin{cases} \dfrac{\partial \rho_\alpha}{\partial t} + \nabla \cdot (\rho_\alpha \boldsymbol{V}_\alpha) = 0 \\ \dfrac{\partial (\rho_\alpha \boldsymbol{V}_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \boldsymbol{V}_\alpha \boldsymbol{V}_\alpha) = \nabla p_\alpha + \boldsymbol{f}_\alpha \\ \dfrac{\partial (\rho_\alpha e_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \boldsymbol{V}_\alpha h_\alpha) = \boldsymbol{f}_\alpha \cdot \boldsymbol{V}_\alpha \end{cases} \tag{1.4}$$

In this equations ρ_α is the density of the α-th species, $\boldsymbol{V}_\alpha$ its velocity, p_α its pressure, $\boldsymbol{f}_\alpha$ the forces applied to the particles, e_α its energy density and h_α its enthalpy. Those equations describe the evolution through time
It is important to note that the mass density, momentum density and energy density in these equations are their average distribution over all velocities.
In the specific case of plasma dynamics, it is preferred to use the number density (number of particles per unit volume), n_α, in place of the density ρ_α, according to the relation:

$$n_\alpha = \frac{\rho_\alpha}{m_{u,\alpha}} \tag{1.5}$$

where $m_{u,\alpha}$ is the mass of a single particle α. By applying this normalization to all equations in (1.4), the variables pass from being density, momentum density and energy density to being the particle number density n_α, the particle momentum density $n_\alpha \boldsymbol{V_\alpha}$ and the particle energy density $n_\alpha e_\alpha$. So for the α-th species the conservation equations become as in Eq. (1.6).

[1] The real, physical one

$$\begin{cases} \dfrac{\partial n_\alpha}{\partial t} + \nabla \cdot (n_\alpha \boldsymbol{V}_\alpha) = 0 \\ \dfrac{\partial (n_\alpha \boldsymbol{V}_\alpha)}{\partial t} + \nabla \cdot (n_\alpha \boldsymbol{V}_\alpha \boldsymbol{V}_\alpha) = \nabla \left(\dfrac{p_\alpha}{m_{u,\alpha}} \right) + \dfrac{\boldsymbol{f}_\alpha}{m_{u,\alpha}} \\ \dfrac{\partial (n_\alpha e_\alpha)}{\partial t} + \nabla \cdot (n_\alpha \boldsymbol{V}_\alpha h_\alpha) = \dfrac{\boldsymbol{f}_\alpha \cdot \boldsymbol{V}_\alpha}{m_{u,\alpha}} \end{cases} \tag{1.6}$$

This set of equations is written once for each of the species considered in the problem.

1.2 Electric potential

Differently from classic fluids, plasmas are subjected to electro-magnetic forces.
As previously stated, plasmas are characterized by the presence of electrically charged particles, such as electrons and ions. Those particles generate and interact with electromagnetic fields, based on the Maxwell's equations, Eq. (1.7), and on the Lorentz Force, Eq. (1.8). Maxwell's equations are presented in the following form:

$$\begin{cases} \nabla \cdot \boldsymbol{E} = \dfrac{Q}{\varepsilon_0} \\ \nabla \cdot \boldsymbol{B} = 0 \\ \nabla \times \boldsymbol{E} = -\dfrac{1}{c} \dfrac{\partial \boldsymbol{B}}{\partial t} \\ \nabla \times \boldsymbol{B} = \dfrac{1}{c} \left(4\pi \boldsymbol{I} + \dfrac{\partial \boldsymbol{E}}{\partial t} \right) \end{cases} \tag{1.7}$$

where $\boldsymbol{E}$ is the electric field, Q the charge density, ε_0 the vacuum permittivity, $\boldsymbol{B}$ the magnetic field, c the vacuum light speed, $\boldsymbol{I}$ the current density.
The Lorentz force applied on a single charged particle instead is:

$$\boldsymbol{F}_L = \pm Q_0 (\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \tag{1.8}$$

where Q_0 is the elementary charge and $\boldsymbol{v}$ is the velocity of particle. The sign of the force is assigned based on the particle charge, + for positively charged (ions), − for negatively charged ones (electrons).

One can notice that in Eq. (1.7), if the magnetic field is constant, a condition that can be assumed correct for some cases like electrostatic space propulsion or the case studies in this document, then the electric field becomes **solenoidal**, since:

$$\nabla \times \boldsymbol{E} = 0$$

Therefore, it is possible to define the electric field as a function of an electric potential V:

$$\boldsymbol{E} = -\nabla V \tag{1.9}$$

Rewriting the divergence of the electric field, one can find:

$$\nabla \cdot \boldsymbol{E} = \nabla \cdot (-\nabla V) = -\nabla^2 V \tag{1.10}$$

$$\nabla^2 V = -\frac{Q}{\varepsilon_0} \tag{1.11}$$

Eq. (1.11) is the Poisson equation that governs the electric potential and it is an elliptic term. The Poisson equation and the electric potential are then needed to evaluate the electric force $\boldsymbol{f}_e$ that acts upon each species:

$$\boldsymbol{f}_e = n_\alpha Q_\alpha \boldsymbol{E} = n_\alpha Q_\alpha (-\nabla V) \tag{1.12}$$

where Q_α is the elementary charge of the species.

Eq. 1.11 must, therefore, be included in the system of equations that governs our plasma, along with the conservation laws for each α species, with the electric potential V becoming a new variable. The system of equations is presented in eq. 1.13.

$$\left\{\begin{array}{l} \dfrac{\partial n_\alpha}{\partial t} + \nabla \cdot (n_\alpha \boldsymbol{V}_\alpha) = 0 \\ \dfrac{\partial (n_\alpha \boldsymbol{V}_\alpha)}{\partial t} + \nabla \cdot (n_\alpha \boldsymbol{V}_\alpha \boldsymbol{V}_\alpha) = \nabla \left(\dfrac{p_\alpha}{m_{u,\alpha}} \right) + \dfrac{\boldsymbol{f}_\alpha}{m_{u,\alpha}} \\ \dfrac{\partial (n_\alpha e_\alpha)}{\partial t} + \nabla \cdot (n_\alpha \boldsymbol{V}_\alpha h_\alpha) = \dfrac{\boldsymbol{f}_\alpha \cdot \boldsymbol{V}_\alpha}{m_{u,\alpha}} \\ \nabla^2 V = -\dfrac{Q}{\varepsilon_0} \end{array}\right\} \forall \alpha, \alpha = \{e, i, n\} \tag{1.13}$$

It is important to notice that the introduction of this new variable to the system has to be met with appropriate specific initial condition and boundary condition in order to have a well-posed problem. In particular, given that the Poisson equation is a diffusive equation, it requires that the boundary condition are applied everywhere.

1.3 Hyperbolic Equations and Boundary Conditions

Unsteady Euler equations are a hyperbolic set of equations with respect to time. This type of equations are characterized by wave-like phenomena [7] and physical systems based on them present properties associated typically to wave propagation, such as having a finite propagation speed of information or a directionality in property transfer.

Considering at the moment for simplicity a 1-D case, then the generic system of N conservation equations can be written as:

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial x} f(q) = 0 \tag{1.14}$$

with q the conserved quantity vector and f their respective fluxes. It is possible to linearize these equations using the flux Jacobian matrix $A \in \mathbb{R}^N \times \mathbb{R}^N$:

$$A = \frac{\partial f}{\partial q} \Rightarrow \frac{\partial q}{\partial t} + \frac{\partial f}{\partial q}\frac{\partial q}{\partial x} = \frac{\partial q}{\partial t} + A\frac{\partial q}{\partial x} = 0 \tag{1.15}$$

Such a system is hyperbolic if A is full rank and it has only real eigenvalues. Now, we can decompose the matrix as:

$$A = R\Lambda L^T \tag{1.16}$$

with Λ the diagonal matrix whose elements are the eigenvalues of the problem λ_i, R the matrix of the right eigenvectors r_i and L the matrix of the left eigenvectors l_i, with $i = 1, ..., N$.

Returning to equation (1.15), it can be rewritten as:

$$L^T \left(\frac{\partial q}{\partial t} + A \frac{\partial q}{\partial x} \right) = 0 \tag{1.17}$$

And remembering the matrix A decomposition:

$$L^T \frac{\partial q}{\partial t} + L^T R \Lambda L^T \frac{\partial q}{\partial x} = 0 \tag{1.18}$$

It is important to know that the left eigenvectors and their respective right counterparts are mutually orthogonal, that means:

$$l_i^T \cdot r_j = \delta_{ij} \quad \forall i, j = 1, ..., N \Rightarrow L^T = R^{-1} \tag{1.19}$$

And since, by definition, $R^{-1}R = I$, where I is the unit matrix, the equation (1.18) becomes:

$$L^T \frac{\partial q}{\partial t} + \Lambda L^T \frac{\partial q}{\partial x} = 0 \tag{1.20}$$

It is possible to define a new vector of characteristic variables as:

$$c = L^T q \tag{1.21}$$

So we can finally obtain the linear system of equations:

$$\frac{\partial c}{\partial t} + \Lambda \frac{\partial c}{\partial x} = 0 \tag{1.22}$$

The characteristic variables, while not being the most interesting variables to have as a direct output, are an important tool in the study of the behaviour of conservation laws, as they give us important information regarding how perturbations and waves propagate inside our domain, allowing for a better formulation of the boundary conditions. As a matter of fact, the equations in (1.22) are written in a way that the makes the evolution of each characteristic variable c_i is independent from the others, and shows that each signals its transported at its own wave speed along the corresponding characteristic line $\frac{dx}{dt} = \lambda_i$ in the x-t plane. To find the eigenvalues λ_i, one needs to solve the characteristic equation related to the matrix A:

$$\det |A - \lambda I| = 0 \tag{1.23}$$

From this equation, whose resolution is omitted here, three eigenvalues for each species can be found for the 1-D conservation laws:

$$\lambda_1 = u - a \qquad , \qquad \lambda_2 = u \qquad , \qquad \lambda_3 = u + a \tag{1.24}$$

with a the sound speed. It is important to notice that these eigenvalues represent the velocity of propagation of signals of the problem.

To understand how to properly set boundary conditions for this system of equations, let's consider an inlet/outlet type of boundary, with the normal parallel to x and positive pointing inward. For this type of boundary we can imagine that there will be a number N_+ of incoming characteristics ($\lambda_+ > 0$) and a number N_- of outgoing characteristics ($\lambda_- < 0$). We can consider the mismatch between the solution derived from the boundary conditions, q_{BC}, and the solution from the state inside the domain, q_{dom}, as a perturbation δq. It can be found that this perturbation is the sum of the information contained in the incoming and outgoing waves:

$$\delta q = q_{BC} - q_{dom} = \delta q_+ + \delta q_- \tag{1.25}$$

Recalling the definition of the characteristic variables, we can also define the perturbation of the characteristic variables related to the incoming characteristics and the perturbation of the characteristic variables related to the outgoing characteristics by multiplying the perturbation δq by the corresponding left eigenvectors, L_+ and L_-, that means:

$$\delta c_+ = L_+ \delta q \qquad\qquad \delta c_- = L_- \delta q \tag{1.26}$$

We now focus on the characteristics incoming into the domain. Instead of evaluating the desired value of the incoming characteristic variables, in actual applications we want to impose a number N_{BC} of boundary conditions for a certain physical variable b_i ($i = 1, ..., N_{BC}$). To do that we write:

$$\delta c_+ = L_+ \delta q = L_+ \frac{\partial q}{\partial b}(b_i^* - b_{dom,i}) = B_+(b_i^* - b_{dom,i}) \tag{1.27}$$

where b_i^* and $b_{dom,i}$ are, respectively, the imposed value and the value in the domain of the physical variable b_i, $\dfrac{\partial q}{\partial b}$ is the Jacobian matrix of the variables q over the variables b and $B_+ = L_+ \dfrac{\partial q}{\partial b}$ is a matrix in $\mathbb{R}^{N_+ \times N_{BC}}$. In order to have a well-defined algebraic equation, it is necessary for the rank of B_+ to be $\text{rank}(B_+) \geq N_+$, which implies that the number of imposed conditions should be $N_{BC} \geq N_+$. However the general rule used in practice is to enforce $\text{rank}(B_+) = N_{BC} = N_+$, as there are several reasons for avoiding to end up in the situation where you have $N_{BC} > N_+$:

1) In the worst case scenario, we will have an *over-constrained* simulation. Since we are imposing more conditions than the number of characteristics, and therefore information, that is coming from outside the domain, the imposed values will enter into conflict with the information contained in the characteristics δc_-. In an over-constrained condition, the solution may present different issues: from a loss of accuracy (in the rare case the imposed values are very close to the values in the domain) to generation of numerical instabilities that invalidates the solution or even lead to a crash;

2) A slightly better scenario occurs if the simulation makes use of a Riemann solver at the boundary. In such a case it will solve a Riemann problem between the imposed boundary conditions and the values in the domain, enforcing values that are a compromise between the ones we wanted to impose and the ones in the domain. While this method would avoid an immediate crash, the actually imposed values would deviate from the ones we desired, therefore it is better to avoid such situation.

Knowing about the characteristic decomposition of the Conservation laws is the base of the theory behind the *upwind methods* for the computation of convective fluxes. These methods, as the name suggest, have being developed to follow the direction of the characteristics, that is to follow the way signals and information propagates in the domain. These methods contribute to the stability to the flux,

in particular in proximity of shocks, where other methods, such as central methods, may create numerical wiggles. Examples of these methods are: Godunov, Osher and Roe (also called Flux Difference Splitting methods), Steger-Warming, Van Leer and AUSM (also called Flux Vector Splitting methods).

1.4 Gibbs oscillations

When we numerically solve fluid-dynamics problems, we should always be aware that the solution found using these methods will not be an exact solution, but an approximated one. The approximated solution ability to adequately represent the exact one depends on multiple factors, such as the number of cells in which the domain has been divided or the numerical scheme used to compute the fluxes between cells, but also the reconstruction of the solution inside each cell, which is usually done by the means of a polynomial function. Most numerical solvers work using constant or linear functions, due to their simple implementation and low computational cost, but in recent years high-order methods have gained more interest, thanks to the growth in computational capabilities available. Increasing the degree of the polynomial reconstruction allows for a better representation of the solution and a substantial reduction of the error in the approximation, however new problems arise from going from constant or linear reconstruction to higher order polynomial expansion, one being the appearance of the Gibbs phenomenon.
The Gibbs phenomenon, or Gibbs oscillation, is a numerical issue related to the reconstruction of irregular functions, such as a step function, that appears when you truncate an infinite series (i.e. Fourier series, Taylor series) to a finite number of terms to approximate a function. For regular functions, a finite number of terms can satisfyingly approximate such functions, as the coefficients of different terms have different weights in representing the function, therefore it is possible to discard the terms of less weight while maintaining a reasonable degree of error compared to the original function. However, in the case of irregular function, such as a step function or other functions that present a discontinuity, the different terms of the infinite series have all similar weights, making the truncated series not able to properly represent the desired function and, in particular, in proximity of discontinuities the approximated function tends to show a non-physical oscillatory behaviour, the Gibbs phenomenon.
Considering CFD analysis, Gibbs oscillation may appear, more or less severely, in the presence of discontinuities, such as shocks. For these type of simulations, the Gibbs phenomenon can be a source of instabilities, as some of those oscillations may

result in non-physical solutions, such as the possibility of having negative values for density, pressure or temperature, which are results numerically possible, but that are physically incorrect. It can be noted that Gibbs oscillation in CFD appears only for high-order simulations, while constant and linear reconstructions avoid the issues related to this phenomenon because they are bounded in the admissible values by the solution in the neighbour cell in proximity of the discontinuity, avoiding the formation of new local extremes.

Chapter 2

Discontinuous Galerkin Method

The Discontinuous Galerkin (DG) Method is a Finite Element Method (FEM). Differently from traditional FEMs, the DG Method, as its name suggests, allows its elements to have discontinuous solutions between one another, a characteristic much more common in Finite Volume Method (FVM). The DG Method, therefore, can present itself as a combination of FEM and FVM, resulting in properties such as flexibility in handling complex geometry, insuring numerical conservation, adaptability and high order accuracy [4, 13, 5].

2.1 General principles

Considering a system of N conservation equations, which can be written in the generic form:

$$\frac{\partial q_m}{\partial t} + \nabla \cdot \mathbf{f}_m + \nabla \cdot \mathbf{d}_m + s_m = 0 \tag{2.1}$$

with $m = 1, ..., N$ the variable index, q the conserved quantities, $\mathbf{f}$ the convective fluxes, $\mathbf{d}$ the diffusive fluxes and s the source terms. It is possible to define a space vector $\mathcal{U}$ and a solution state vector $\boldsymbol{u} \in \mathcal{U}^N$:

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix}, \qquad u_m \in \mathcal{U} \quad \forall m = 1, ..., N \tag{2.2}$$

All the components of eq. (2.1) are functions of $\boldsymbol{u}$, the position $\boldsymbol{x}$ and the time t.

As previously stated, the domain Ω of the problem is approximated as a collection of geometrically simple elements e, so that:

$$\Omega \approx \mathcal{E} = \cup e \tag{2.3}$$

It can be defined a *trial space* $\mathcal{U}_h$, composed of vectors of polynomial functions of interpolations order p. These functions in the case of DG Method are regular inside the domain of each element e, but not necessarily continuos across different elements.

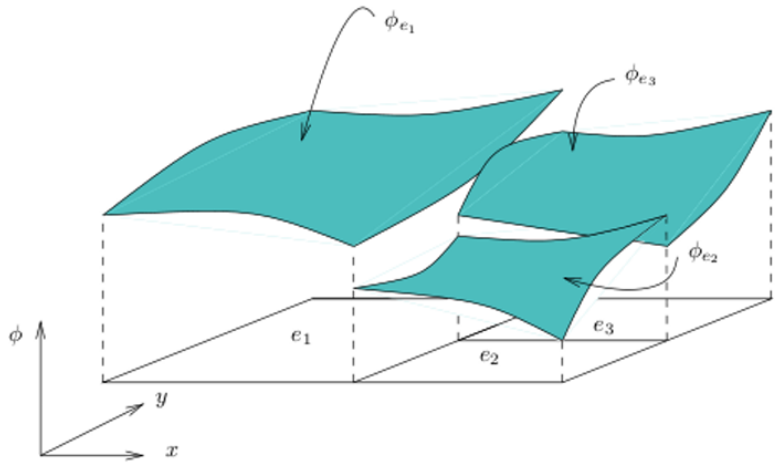


Figure 2.1: Non-conformal discontinuous finite element interpolation. Illustration taken from [6]

The exact solution can be approximated as a linear combination of a finite set shape functions ϕ_i:

$$\boldsymbol{u}_m(\boldsymbol{x}, t) \approx \boldsymbol{u}_{h,m}(\boldsymbol{x}, t) = \sum_i \phi_i(\boldsymbol{x}) u_{mi}(t) \quad , \qquad m = 1, ..., N \tag{2.4}$$

where $u_{mi}(t)$ are the time-dependent expansion weights, that it denotes how much each ϕ_i contributes to the approximated solution for the m-th variable. Substituting this approximated solution in eq. (2.1), the equations them-self are not exactly satisfied everywhere in the domain. The DG method, as any other FEM method, requires

instead that the residuals of those equations are orthogonal to all the elements v of a chosen test space $\mathcal{V}$, that is:

$$\int_\Omega v\left(\frac{\partial q}{\partial t} + \nabla\cdot\mathbf{f} + \nabla\cdot\mathbf{d} + s\right)dV = 0 \quad , \quad \forall v = (v_1, ..., v_N)^T \in \mathcal{V}^N \tag{2.5}$$

In the case that $\mathcal{V} = \mathcal{U}_h$, the approach is called Galerkin Variational formulation. The eq. (2.5) can be rewritten as a sum over each element e:

$$\sum_e \int_e \left(v\frac{\partial q}{\partial t} + v\nabla\cdot\mathbf{f} + v\nabla\cdot\mathbf{d} + vs\right)dV = 0 \quad , \quad \forall v \in \mathcal{U}_h \tag{2.6}$$

Moreover the flux components, that can be rewritten for simplicity as $\mathbf{g} = \mathbf{f} + \mathbf{d}$, can be further expanded using an integration by part:

$$\sum_e \int_e v\nabla\cdot\mathbf{g}dV = -\sum_e\left(\int_e \nabla v\cdot\mathbf{g}dV - \oint_{\partial e} v\mathbf{g}\cdot\boldsymbol{n}dS\right) \tag{2.7}$$

with ∂e the boundary of the element and $\boldsymbol{n}$ the normal vector to the face[1]. The boundary is given by the union of the faces f of the element, $\partial e = \cup f$, and therefore one can rewrite eq. (2.7) as:

$$-\sum_e\left(\int_e \nabla v\cdot\mathbf{g}dV - \oint_{\partial e} v\mathbf{g}\cdot\boldsymbol{n}dS\right) = -\sum_e\left(\int_e \nabla v\cdot\mathbf{g}dV - \sum_{f\in\partial e}\int_f v\mathbf{g}\cdot\boldsymbol{n}dS\right) \tag{2.8}$$

As it is currently presented all interfaces are counted twice (each interface belongs to two elements). It is possible to fix a common normal vector $\boldsymbol{n}$ for each face and design the element whose normal points outward aligns with $\boldsymbol{n}$ as e^+ ($\Rightarrow \boldsymbol{n}^+ = \boldsymbol{n}$), whereas the element with an opposing outward normal is designed as e^- ($\Rightarrow \boldsymbol{n}^- = -\boldsymbol{n}$). So eq. (2.7) can once again be rewritten as:

[1]The normal vector is conventionally positive when directed outward.

$$\sum_e \int_e v\nabla\cdot\mathbf{g} = -\sum_e \left(\int_e \nabla v\cdot\mathbf{g}dV - \sum_{f\in\partial e}\int_f v\mathbf{g}\cdot\boldsymbol{n}dS\right) =$$
$$= -\sum_e \int_e \nabla v\cdot\mathbf{g}dV + \sum_f \int_f (v^+\mathbf{g}(\boldsymbol{u}^+) - v^-\mathbf{g}(\boldsymbol{u}^-))\cdot\boldsymbol{n}dS = \tag{2.9}$$
$$= -\sum_e \int_e \nabla v\cdot\mathbf{g}dV + \sum_f \int_f (v^+\mathbf{g}(\boldsymbol{u}^+)\cdot\boldsymbol{n}^+ + v^-\mathbf{g}(\boldsymbol{u}^-)\cdot\boldsymbol{n}^-)dS$$

To simplify the notation, we introduce the jump operator $[[\cdot]]$ as:

$$[[a]] = a^+\boldsymbol{n}^+ + a^-\boldsymbol{n}^- \qquad [[\mathbf{a}]] = \mathbf{a}^+\cdot\boldsymbol{n}^+ + \mathbf{a}^-\cdot\boldsymbol{n}^- \tag{2.10}$$

Therefore :

$$\sum_e \int_e v\nabla\cdot\mathbf{g} = -\sum_e \int_e \nabla v\cdot\mathbf{g}dV + \sum_f \int_f [[v\mathbf{g}(\boldsymbol{u})]] \tag{2.11}$$

Introducing also the the average operator $\{\{\cdot\}\}$:

$$\{\{a\}\} = \frac{a^+ - a^-}{2} \qquad \{\{\mathbf{a}\}\} = \frac{\mathbf{a}^+ + \mathbf{a}^-}{2} \tag{2.12}$$

We can identify the following identity:

$$[[v\mathbf{g}]] = [[v]]\{\{\mathbf{g}\}\} + \{\{v\}\}[[\mathbf{g}]] \tag{2.13}$$

We then substitute the jump $[[v\mathbf{g}]]$ by a consistent, but not equivalent, interface flux γ:

$$\gamma(\boldsymbol{u}^+,\boldsymbol{u}^-;v^+,v^-;\boldsymbol{n}) \approx [[v\mathbf{g}]] = v^+\boldsymbol{g}(\boldsymbol{u}^+)\cdot\boldsymbol{n}^+ + v^-\boldsymbol{g}(\boldsymbol{u}^-)\cdot\boldsymbol{n}^- \tag{2.14}$$

So, in the end, eq. (2.6) take the form:

$$\sum_e \int_e v\left(\frac{\partial q}{\partial t} + s\right)dV - \sum_e \int_e \nabla v\cdot\mathbf{g}dV + \sum_f \int_f \gamma(\boldsymbol{u}^+,\boldsymbol{u}^-;v^+,v^-;\boldsymbol{n})dS = 0 \quad , \quad \forall v\in\mathcal{U}_h \tag{2.15}$$

Up at this point, no assumptions have been made about the interface flux γ other than that it has to be consistent, in the sense that the exact solution should satisfy eq. (2.15). In particular it has to be true that, when $\boldsymbol{u}^+ = \boldsymbol{u}^- = \boldsymbol{u}$, the interface flux revert to:

$$\gamma(\boldsymbol{u}, \boldsymbol{u}; v^+, v^-; \boldsymbol{n}) \approx [[v\mathbf{g}]] = [[v]]\{\{\mathbf{g}(\boldsymbol{u})\}\} + \{\{v\}\}[[\mathbf{g}(\boldsymbol{u})]] = [[v]] \cdot \mathbf{g}(\boldsymbol{u}) \tag{2.16}$$

The problem now is how to choose an adequate function γ to guarantee stability. Different choices can be made based on the nature of the problem: the case of hyperbolic problems is discussed in section (2.2), instead the case of elliptic problems is found in section (2.3).

2.2 Hyperbolic DG

Hyperbolic equation systems can be represented in the following typical form:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \mathbf{f} = 0 \tag{2.17}$$

with, as said earlier, $\mathbf{f}$ being the convective flux. Confronting this equation with eq. (2.15) one can notice that:

$$\gamma(\boldsymbol{u}^+, \boldsymbol{u}^-; v^+, v^-; \boldsymbol{n}) \approx v^+ \mathbf{f}(\boldsymbol{u}^+) \cdot \boldsymbol{n}^+ + v^- \mathbf{f}(\boldsymbol{u}^-) \cdot \boldsymbol{n}^- \tag{2.18}$$

In FVM two-point interface, *flux function* $\mathcal{H}(\boldsymbol{u}^+, \boldsymbol{u}^-; \boldsymbol{n})$ are used to convey the flux through a face between two cell. This function has to satisfy the following properties:

- *Consistency*: If the solution on either elements is the same, $\boldsymbol{u}^+ = \boldsymbol{u}^- = \boldsymbol{u}$, then the function should recover the value of the actual flux, that means:

$$\mathcal{H}(\boldsymbol{u}, \boldsymbol{u}; \boldsymbol{n}) = \mathbf{f}(\boldsymbol{u}) \cdot \boldsymbol{n}$$

- *Conservativity*: The numeric flux has to be unique and balanced at the interface, which means considering two adjacent elements, their numerical fluxes are such that:

$$\mathcal{H}(\boldsymbol{u}^+, \boldsymbol{u}^-; \boldsymbol{n}) = -\mathcal{H}(\boldsymbol{u}^-, \boldsymbol{u}^+; -\boldsymbol{n})$$

- *Stability*: The numerical flux should be chosen in a way that prevents the solution from diverging, while ensuring it remains in well defined mathematical bounds.

The interface flux in Discontinuous Galerkin is typically chosen as:

$$\gamma(\boldsymbol{u}^+, \boldsymbol{u}^-; v^+, v^-; \boldsymbol{n}) = [[v]]\mathcal{H}(\boldsymbol{u}^+, \boldsymbol{u}^-; \boldsymbol{n}) \tag{2.19}$$

This choice allows to extend the intrinsic stability of FVM to high order DGM.

2.3 Elliptic DG

A purely Elliptic system of equations can take the following form:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \mathbf{d}(\boldsymbol{q}, \nabla \boldsymbol{q}) = 0 \tag{2.20}$$

where $\mathbf{d}$ is the diffusive flux. Considering the case of a linear elliptic system, the diffusive flux is:

$$\begin{aligned} \mathbf{d} &= -\mathcal{D} \cdot \nabla \boldsymbol{q} \\ d_m^k &= -D_{mn}^{kl} \frac{\partial q_n}{\partial u^l} \end{aligned} \tag{2.21}$$

with $\mathcal{D}$ a positive definite 4^{th} order tensor. Compared to hyperbolic problems, which more easily adapt to the properties of DG methods due to their directional propagation of information along characteristic lines, elliptic problems, that are global and have not a preferred direction, require a different formulation to adapt into the DG method. Different methods have been developed for elliptic problems, which can be categorized in two group: *Interior Penalty Methods*(IPM), that can be seen as a generalization of the classical continuous finite element method, and *Lifting methods*, which use an additional variational problem to find an approximation of the solution gradient to obtain a discretization of elliptic equations. in the work presented here, only the former ones considered.

The implementation of Interior Penalty Methods can be done by adding to the Galerkin variational formulation (eq. (2.15)) a Galerkin penalty term and a boundary flux term, to retain consistency, so that the equation becomes:

$$\sum_e \int_e v\frac{\partial \boldsymbol{q}}{\partial t}\, dV - \sum_e \int_e \nabla v \cdot \mathbf{d}\, dV + \sum_f \sigma_f \int_f [[v]] \cdot [[\boldsymbol{q}]]\, dS + \sum_f \int_f [[v]] \cdot \{\{\mathcal{D} \cdot \nabla \boldsymbol{q}\}\}\, dS = 0 \tag{2.22}$$

where σ_f is the penalty coefficient. It is possible to add another term to equation (2.22) in order to either provide simmetry ($\theta = 1$) or relax the conditions on σ_f ($\theta = -1$):

$$\begin{aligned} \sum_e \int_e v\frac{\partial \boldsymbol{q}}{\partial t}\, dV - \sum_e \int_e \nabla v \cdot \mathbf{d}\, dV + \sum_f \sigma_f \int_f [[v]] \cdot [[\boldsymbol{q}]]\, dS \\ + \sum_f \int_f [[v]] \cdot \{\{\mathcal{D} \cdot \nabla \boldsymbol{q}\}\}\, dS \\ + \theta \sum_f \int_f [[\boldsymbol{q}]] \cdot \{\{\mathcal{D} \cdot \nabla v\}\}\, dS = 0 \end{aligned} \tag{2.23}$$

Particular consideration must be given to the penalty coefficient σ_f, that depends on various factors, including geometry and diffusive property of the gas. This term has to be[9]:

$$\sigma_f > \frac{(1+\theta)^2}{8} \max_{e \ni f} (n_f^e C_{e,f} \rho(\mathcal{D})) \tag{2.24}$$

where n_f^e is the number of faces for the element e, $\rho(\mathcal{D})$ is the spectral radius of the diffusive tensor and $C_{e,f}$ is a global geometric factor equal to:

$$C_{e,f} = (1+p)\frac{A(f)}{V(e)} \tag{2.25}$$

where p is the interpolation order, $A(f)$ is the area of the face f and $V(e)$ is the volume of the element e.

The constant θ in eq. (2.23) can in theory assume any value, but the only value of interest are the following:

- $\theta = 0$, which return to the form of eq. (2.22), and it is called the *Incomplete Interior Penalty Method* (IIPM). In this case the penalty coefficient has to be higher than a critical value, that is $\sigma_f > \sigma_{f,IIP}$;

- $\theta = 1$, it is called the *Symmetric Interior Penalty Method* (SIPM). In this case the critical value is $\sigma_{f,SIP} = 2\sigma_{f,IIP}$;
- $\theta = -1$, it is called the *Non-symmetric Interior Penalty Method* (NIPM). In this case the penalty coefficient just need to be $\sigma_f > 0$.

Confronting the equation (2.15) with equation (2.23), the interface flux for the Interior Penalty Method is written as:

$$\gamma_{IPM}(u^+, u^-, v^+, v^-, \boldsymbol{n}) = \sigma_f[[v]] \cdot [[\boldsymbol{q}]] + [[v]] \cdot \{\{\mathcal{D} \cdot \nabla \boldsymbol{q}\}\} + \theta[[\boldsymbol{q}]] \cdot \{\{\mathcal{D} \cdot \nabla v\}\} \quad (2.26)$$

2.4 Shock Capturing

High-order methods, such as Discontinuous Galerkin, are an essential tool for improving the accuracy of fluid-dynamics simulations, but they present new challenges compared to more traditional methods. One such challenge is the ability to handle strong shocks in the domain. The presence of strong shocks, or more generally of steep discontinuities, leads to the generation of non-physical oscillations in the solution, which in turn can be a source of numerical instability and may produce physically impossible results, such as negative density or pressure.

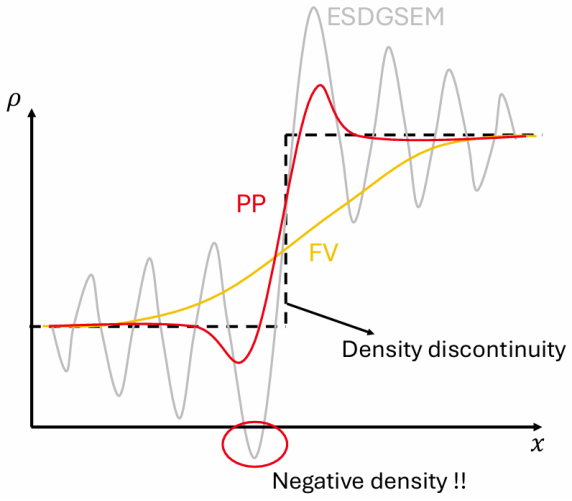


Figure 2.2: Numerical oscillations of high-order simulations near strong shocks/discontinuities. Example: density discontinuity

Different strategies have been developed to properly deal with this issue, ranging from reducing the order of the method in proximity of shocks, to limiters, to the addition of artificial viscosity. This latter approach will be the sole one which will be explored in this work.

2.4.1 General premise

Considering a generic conservation law, defined on a domain Ω, written in the form:

$$\frac{\partial q}{\partial t} + \nabla \cdot \boldsymbol{F} = 0 \tag{2.27}$$

where q are the conserved quantities and $\boldsymbol{F}$ is their flux vector. The artificial viscosity stabilization method is characterized by adding a dissipative [12] term to the equation:

$$\frac{\partial q}{\partial t} + \nabla \cdot \boldsymbol{F} = \nabla \cdot (\varepsilon \nabla q) \tag{2.28}$$

The term ε in this equation serves to regulate the intensity of added viscosity. The value of this parameter should vary in the domain, so that if an element is distant from shocks the conservation law return to the form in eq. (2.27), i.e. $\varepsilon = 0$. Whereas, since eq. (2.28) causes shocks to spread over a thickness of order $\mathcal{O}(\varepsilon)$, in proximity of shocks the value of ε should be chosen appropriately to ensure the shock is correctly approximated given the available spatial resolution, fig. 2.3. Therefore near shocks $\varepsilon = \mathcal{O}(h/p)$, where h is the element size and p the polynomial order.

2.4.2 Sensor

Another important component of shock-capturing strategies is the ability to identify the elements containing a shock, therefore it is fundamental to choose an appropriate sensor for the problem currently under investigation. In this work, two sensors, both based upon irregularities in the solution, will be used: the sensor presented by Persson and Peraire [12] and the one by Hennemann [3].

The Persson and Peraire sensor, in order to determine the elements affected by shocks, makes use of a characteristic of the polynomial coefficients used to approximate the solution. In particular, in the case of a smooth solution the contribution of higher order are expected to decay quickly, as most of the energy is contained in the

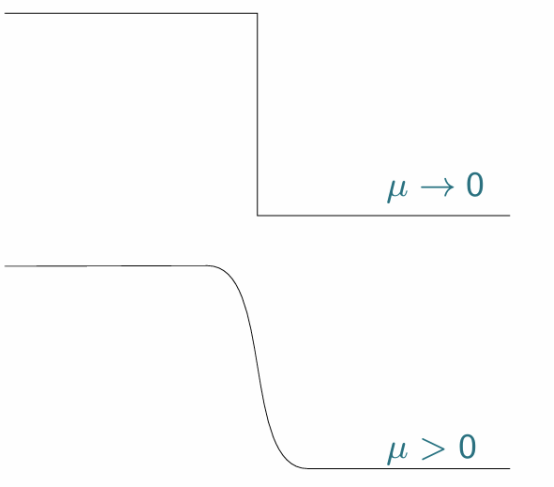


Figure 2.3: Effect of viscosity on shocks. In this case $\mu = \varepsilon$

low-order coefficients. Instead in case of a steep discontinuity the decay rate is slower and is related to the strength of the discontinuity. Following this logic, it is possible to define a smoothness indicator within each element that confronts the energy of only the highest order polynomial mode with the energy in the entire solution:

$$S_e = \frac{(u - \hat{u}, u - \hat{u})_e}{(u, u)_e} \tag{2.29}$$

where u is the solution of polynomial order p, $\hat{u}$ the solution truncated to order $p-1$ and $(\cdot, \cdot)_e$ the standard L_2 inner product over the element e. From the value obtained by this indicator it is possible to calculate a blending value within an element, α_e, according to the following function:

$$\alpha_e = \begin{cases} 0\,, & \text{if} \quad s_e < s_0 - \kappa \\ \dfrac{1}{2}\left(1 + \sin \dfrac{\pi(s_e - s_0)}{2\kappa}\right), & \text{if} \quad s_0 - \kappa \leq s_e \leq s_0 + \kappa \\ 1\,, & \text{if} \quad s_e > s_0 + \kappa \end{cases} \tag{2.30}$$

with $s_e = \log_{10} S_e$, whereas s_0 and κ, are chosen empirically in order to achieve a sharp shock profile while maintaining numerical stability.

The Hennemann sensor is also based on the same principle presented by Persson

and Peraire, and therefore makes use of the smoothness indicator S_e presented in (2.29). To determine whether a shock is detected, Hennemann defines a threshold value, function of the polynomial order, $\mathcal{T}(o)$ as:

$$\mathcal{T}(o) = a\ 10^{-c(o+1)^{\frac{1}{4}}} \tag{2.31}$$

where a and c are constants with values $a = 0.5,\ \ c = 1.8$, following the considerations of Hennemann in [3]. It is now possible to define a blending value for each element for the Hennemann sensor, based on the function defined as follows:

$$\alpha_e = \frac{1}{1 + \exp\left(\frac{-s}{\mathcal{T}}(S_e - \mathcal{T})\right)} \tag{2.32}$$

The constant $s = 9.21024$ here is chosen such that for $S_e = 0$ one can find $\alpha_e = \alpha_{min} = 0.0001$. Lastly, the blending value can be reassigned as:

$$\alpha_e = \begin{cases} 0\,, & \text{if } \ \alpha_e < \alpha_{min} \\ \alpha_e\,, & \text{if } \ \alpha_{min} \leq \alpha_e \leq 1 - \alpha_{min} \\ 1\,, & \text{if } \ \alpha_e > 1 - \alpha_{min} \end{cases} \tag{2.33}$$

Eventually it is possible to impose an upper-limit to the blending value, α_{max}, so that $\alpha_e = \min[\alpha_e, \alpha_{max}]$.

Now that the sensors have been defined, the necessary next step is to present the way artificial viscosity is actually implemented. To do so, different approaches have been developed, but in this work only two will be discussed: the *Laplacian Artificial Viscosity* and the *Physical Artificial Viscosity*.

2.4.3 Laplacian Artificial Viscosity

Considering the model equation previously presented:

$$\frac{\partial q}{\partial t} + \nabla \cdot \boldsymbol{F} = \nabla \cdot (\varepsilon \nabla q) \tag{2.34}$$

The Laplacian artificial viscosity approach uses a constant element-wise viscosity coefficient ε_e to substitute the ε coefficient in the formulation. This new coefficient is computed based on the blending value obtained by the sensor:

$$\varepsilon_e = \varepsilon_0 \alpha_e \tag{2.35}$$

where the constant $\varepsilon_0 \propto h/p$ is a reference viscosity, from which it depends the thickness of the shock represented in the domain, and it's the same for all the cell. As one can notice, when a cell is far from the shock the value of $\alpha_e = 0 \Rightarrow \varepsilon_e = 0$ and the artificial viscosity disappear, whereas near the shock the value increase till it becomes equal to ε_0 itself.

2.4.4 Physical Artificial Viscosity

An alternative approach to artificial viscosity take inspiration to the physical viscosity of perfect gases. For this case the model equation can be written as:

$$\frac{\partial q}{\partial t} + \nabla \cdot \boldsymbol{F} = \nabla \cdot \boldsymbol{F}^v \tag{2.36}$$

with $\boldsymbol{F}^v$ the viscous fluxes, where each component is composed as:

$$F_i^v = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \boldsymbol{u} \cdot \boldsymbol{\tau}_i - Q_i \end{pmatrix}, \quad i = 1, 2, 3 \tag{2.37}$$

With the same logic applied to the Laplacian artificial viscosity, the physical one too should only be used near shocks, while it disappear away from them. Therefore a sensor has to be implemented to localize the cells which present a shock, and add the artificial viscosity in those.

For the artificial viscous fluxes, the stresses are given as:

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \nabla \cdot \boldsymbol{u} \right], \quad i, j = 1, 2, 3 \tag{2.38}$$

while the heat flux are:

$$Q_i = -\kappa \frac{\partial T}{\partial x_i}, \quad i = 1, 2, 3 \tag{2.39}$$

with μ the viscosity coefficient, $\boldsymbol{u}$ the velocity vector, δ_{ij} the Kronecker delta, T the temperature, and κ the thermal conductivity. When physical artificial viscosity is applied, the terms μ and κ are composed by the contribution of a physical component and an artificial one, so that:

$$\begin{aligned} \mu &= \mu_{phy} + \mu_{AV} \\ \kappa &= \kappa_{phy} + \kappa_{AV} \end{aligned} \tag{2.40}$$

In order to choose the right amount of artificial viscosity according to [12], we must first remember that our objective is to be able to numerically capture the shock by spreading it over a thickness $\mathcal{O}(h/p)$. To achieve that is important to know[12] that the shock thickness δ_s is proportional to μ, in particular one have that:

$$\frac{\delta_s \Delta u \rho^*}{\mu^*} \approx 1 \tag{2.41}$$

where Δu is the change in normal velocity across the shock, ρ^* and μ^* are respectively the density and the viscosity evaluated at sonic conditions. Since both Δu and ρ^* are functions of the up-stream conditions, eq. (2.41) allows us to determine the amount of μ^* necessary to obtain the desired δ_s, from here it is possible to find the appropriate value of μ, and therefore how much μ_{AV} has to be added, by assuming that μ is a function of the temperature, so that:

$$\frac{\mu}{\mu^*} = \frac{f(T)}{f(T^*)} \tag{2.42}$$

where T^* is the sonic temperature. For what it regards κ, it is possible to determine its value based on μ considering the Prandtl number that the model use, however it is important to note that in cases, such us the one presented in this work, where the heat transfer is ignored, then the thermal conductivity is $\kappa = 0$.

2.4.5 Artificial Viscosity & Penalty coefficient in Plasmas

The addition of the artificial viscosity, that is a diffusive term, turn the equations presented in section 1.1 from being purely hyperbolic to being parabolic equations, at least in the elements where artificial viscosity is present. This new term have to be treated in a similar fashion as the one that was shown in section 2.3. In particular, when artificial viscosity is applied then the diffusive spectral radius ρ has to be rewritten to add the contribute of artificial viscosity. This in turn, as can be

seen in eq. (2.24), has a considerable effect in the penalty coefficient σ_f. In case we are considering the Laplacian type of artificial viscosity, then the diffusive spectral radius becomes:

$$\rho = \rho_{phy} + 2\rho_{AV} \tag{2.43}$$

where ρ_{phy} is the term related to the physical property of the plasma, like dynamic viscosity, bulk viscosity and thermal conductivity, and ρ_{AV} the artificial viscosity coefficient.

While in gas-dynamics the value of the diffusive spectral radius can be given the same for each equation with a diffusive term within a reasonable degree of error, when you consider plasmas some consideration must be done. First of all, we are no longer considering just one species, but a multi-species fluid, therefore it would be more correct to use species-specific diffusive spectral radius, ρ_α. It is important to notice that, considering the equations and what was stated in section 1.1, the physical diffusion of plasmas is very low, and therefore ignored, and the constant typically associated to it are considered to be zero, so the term $\rho_{phy,\alpha} = 0 \quad \forall\alpha$. The other very important difference compared to gas-dynamics is the presence of another diffusive equation, the electric potential equation from section 1.2, that has a value of diffusive spectral radius much higher[2] than the ones from the conservation equations of the different species, and therefore must be used alongside the other ρ, where, for each species α, ρ_α is used with the relative conservation equations, and $\rho_{ele-pot}$ is used on the electric potential equation.

[2]different order of magnitude

Chapter 3

Penning discharge test case

The reference test case analysed in this manuscript is a simulation of a low-temperature partially magnetized plasma, in which we want to observe the formation of rotating spokes, a type of self-organized structure in the plasma that travels at the acoustic speed of ions, under the influence of a $\boldsymbol{E} \times \boldsymbol{B}$ field (cross-field). This kind of phenomena have been found in different systems, from Hall-Effect Thruster [11, 8], to magnetron discharges to penning discharge [10], both experimentally and numerically.
In this chapter the characteristics of the test case will be presented (sec 3.1) and the approaches taken to improve the stability of this simulation, that is the definition of the boundary condition applied to this problem (sec 3.2) and the use of artificial viscosity (sec 3.3).

3.1 Test case Characteristics

The test case considered is modelled after the penning discharge presented in [10]. Some assumption are made to the geometry and the physics to simplify the simulation, such as:

- 2D Square domain;
- Only 2 species considered: electrons and ions;
- Electric Field, generated by charge disparity, parallel to the plane;
- Magnetic Field constant and perpendicular to the domain.

The domain is delimited by **grounded absorbing walls**, that means that the boundaries have a zero electric potential (grounded walls) and when a particle exit the domain through the boundary it can no longer enter (absorbing walls). The boundary conditions will be discussed further in the next section.

A representation of the simulation set-up is shown in fig. 3.1.

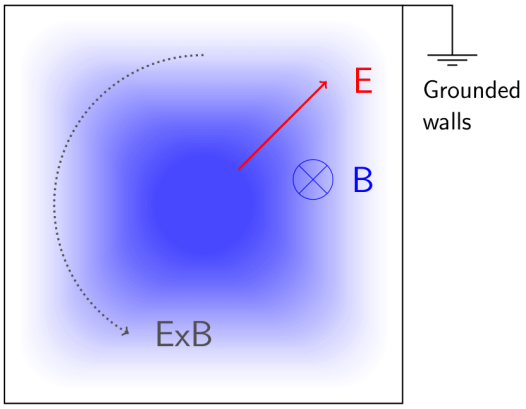


Figure 3.1: Penning Discharge simulation set-up.

The domain chosen for the simulation differs from [10], as the characteristic length of the domain is set to be 1 mm, an order of magnitude less. This choice ends up having an effect on non-dimensional characteristic numbers, for example the Knudsen number Kn, being $Kn \propto 1/L$, becomes 10 times bigger, it is however important to note that the assumption of continuum regime can still be considered valid with a reasonable degree of certainty. This choice has been made so that an inferior number of cells was enough to reach a satisfying degree, enough to resolve the debye length. In order to replicate the physical phenomena, the magnitude of the magnetic field has been changed to make up for the decrease in length, so that the gyromagnetic radius r_L of electrons becomes half the domain length, making them magnetically trapped inside the domain. The relationship:

$$r_L = \frac{mv_\perp}{qB} = \frac{L}{2} \Rightarrow B \propto \frac{1}{L} \tag{3.1}$$

where m and q are respectively the mass and the charge of an electron and $v_\perp$ the velocity perpendicular to the magnetic field. Regarding the ions, it was chosen to use Argon as the reference species for this test case. In table 3.1 some reference values

used in the definition of the simulation are presented.

Name	Symbol	Unit	Value
Characteristic Length	L	mm	1
Time Step	Δt	ps	10
Reference Number Density	n_{ref}	$1/m^3$	2.736e15
Reference Temperature	T_{ref}	K	11600
Ion atomic mass	m_a	u	40
Magnetic field	B	T	0.1

Table 3.1: Simulation Reference values

The *initial conditions* of the test case present a uniform condition, with electrons and ions having an equal number density everywhere, with both species considered to be still at the start of the computation. The initial temperature of the species differs, with the ions starting with a value of $T_i = 0.1$ eV, whereas the electrons start with a temperature of $T_e = 2$ eV. Regarding the electric potential variable, V, that is set to 0 at the start of the simulation. The initial conditions are presented in tab. 3.2.

Name	Symbol	Unit	Electrons(-)	Ions(+)
Number Density	n	$1/m^3$	2.736e15	2.736e15
Velocity Vector	$\boldsymbol{u}$	m/s	0	0
Temperature	T	eV	2	0.1
Electric Potential	V	V	0	

Table 3.2: Initial Conditions of the variables n, $\boldsymbol{u}$, T and V

3.2 Boundary Conditions

The development of appropriate boundary conditions for this test case has been quite the challenging problem.

As previously stated, the domain is delimited by grounded absorbing walls, which can be broken down into two parts:

- *Grounded walls*: Related to electric potential variable. It is resolved by imposing:

$$V = 0 \quad \forall\, f \in \partial\Omega \tag{3.2}$$

where $\partial\Omega$ is the border of the domain and f is the face of an element at the border;

- *Absorbing walls*: It is a type of boundary condition such that when a particle exits the domain it can no longer enter it. In particular in this test case we tried to develop a *vacuum-like boundary condition*, to simulate operating condition of space device. The vacuum conditions are described using the density and the temperature, which for each species would mean to impose as a condition at the boundary that $n_{vac} = 0$ and $T_{vac} = 0$, those conditions, however, can not be used in a numerical simulation, as they can very easily lead to numerical errors or to instabilities in the computation, such as:

  - Singularities can occur if the density becomes exactly 0, as the value of the density is used to extract the velocity from the momentum ($\boldsymbol{V} = \frac{n\boldsymbol{V}}{n}$) or the energy per unit volume ($e = \frac{ne}{n}$), which poses problems for $n = 0$;
  - The speed of sound is computed as $a = \sqrt{\gamma RT}$, which poses problems if $T \leq 0$, as it would be difficult to evaluate a, and therefore the Mach number M, that is important to know to impose the correct boundary;
  - The imposition of the conditions $n_{vac} = 0$ and $T_{vac} = 0$ may lead to numerical oscillations, as even a little fluctuation may produce non-physical results, such as a negative density or a negative temperature.

  To avoid these issues it has been decided to use vacuum-like conditions, which means that at the boundary very small positive values of density and temperature have been imposed, instead of directly 0. The vacuum values, normalized with respect to the reference values, for density, n_{vac}, and temperature, T_{vac}, are defined as presented in tab 3.3.

Symbol	-	+
n_{vac}	0.02	0.02
T_{vac}	0.5	0.05

Table 3.3: Normalized vacuum condition. (- for electrons, + for ions)

The boundaries are therefore not treated as a physical wall, but as a delimitation between the domain of our interest and an outside vacuum region, where

the particles of the plasma can only go from inside the domain to outside and not in inverse. Given this condition, it is important to properly discern the situation at the boundary so that it is possible to appoint the right type and quantity of boundary conditions. The fundamental factor to consider is the signed Mach number, M, and we consider that, when there is the condition of flow going from inside the domain to outside of it, we define the Mach number as positive, $M > 0$, whereas, when there is the condition of flow going from outside the domain to inside of it, the Mach number is defined as negative, $M < 0$.

At first, the boundary conditions were defined to impose to all the species the vacuum density in the case of the subsonic outflow at the boundary, while, in case of subsonic inflow, all the species were imposed the vacuum temperature and zero velocity, both normal and tangential to the boundary. As one can expect, for supersonic outflow no boundary condition is imposed and for supersonic inflow all the boundary conditions are imposed. In table 3.4 is presented a summary of the boundary conditions written above.

		Density	Normal velocity	Tangential velocity	Temperature
		n	$u_\perp$	$u_\parallel$	T
Supersonic outflow	$M_n > 1$	-	-	-	-
Subsonic outflow	$0 < M_n < 1$	n_{vac}	-	-	-
Subsonic inflow	$-1 < M_n < 0$	-	0	0	T_{vac}
Supersonic inflow	$M_n < -1$	n_{vac}	0	0	T_{vac}

Table 3.4: First attempt for boundary conditions.
Used for both electrons and ions

This approach, while being theoretically correct, was not possible to use for the simulations because, as it will be shown later, it would lead to high instabilities and crashes related to the behaviour of the electrons, that, in particular near the corners, reach negative values. Even with the help of shock capturing methods, such as artificial viscosity, it was not possible to eliminate those instabilities.

Different attempts have been made to change, even slightly, the boundary conditions of electrons in order to avoid the issues exposed before, from reducing locally the value of the imposed density to trying to impose the temperature instead for a subsonic outflow. Unfortunately, none of these ideas managed to solve the crashes that we were getting from the simulations.
The solution to these issues and an increase in the stability of the simulation has been reached only after taking a different approach: instead of having an imposed value for density or temperature as a boundary condition in a subsonic outflow, the electrons present the imposition of $v_\perp = 0$, which means the momentum normal to the boundary will be null too. In table 3.5 is presented a summary of the boundary conditions for the electrons.
This choice of boundary conditions for the electrons is made possible due to the physics of the test case itself. The electrons, being under the influence of both the electric and the magnetic fields, will be magnetically bounded inside the domain and will start to turn in their movement in a counter-clockwise way, in particular, the particles closer to the boundary, and also the firsts to start to move, will present after some time a direction of motion that is tangent to the boundary itself. Given this characteristic, applying a $v_\perp = 0$ as a boundary condition for a subsonic outflow is feasible as it will have the effect of reducing through time the normal velocity, leading to a tangential flow around the border.

		Density	Normal velocity	Tangential velocity	Temperature
		n	$u_\perp$	$u_\parallel$	T
Supersonic outflow	$M_n > 1$	-	-	-	-
Subsonic outflow	$0 < M_n < 1$	-	0	-	-
Subsonic inflow	$-1 < M_n < 0$	n_{vac}	-	0	T_{vac}
Supersonic inflow	$M_n < -1$	n_{vac}	0	0	T_{vac}

Table 3.5: Boundary conditions scheme for electrons

For the ions, instead, the boundary conditions have to remain as showed earlier, as, to simplify the computation, they are not considered to be affected by the

		Density	Normal velocity	Tangential velocity	Temperature
		n	$u_\perp$	$u_\parallel$	T
Supersonic outflow	$M_n > 1$	-	-	-	-
Subsonic outflow	$0 < M_n < 1$	n_{vac}	-	-	-
Subsonic inflow	$-1 < M_n < 0$	-	0	0	T_{vac}
Supersonic inflow	$M_n < -1$	n_{vac}	0	0	T_{vac}

Table 3.6: Boundary condition scheme for ions

magnetic field and are not expected to start to rotate within the computational domain. A summary of the boundary conditions for the ions is presented in tab. 3.6.

While it is possible to avoid some of the issues around the boundary - in particular in the corners - of the domain by applying the boundary conditions for the electrons as presented in tab. 3.5, by doing so new problems arise. The way the boundary conditions are implemented leads to very low absolute values of normal momentum and therefore the flow oscillate between being outflowing ($M_n > 0$) and inflowing ($M_n < 0$), with rapid and localized switches between boundary conditions. This situation is present, in general, in flows tangential to a boundary and could easily produce numerical errors, non-physical results and even arrive to crash the whole simulation, in particular for high-order methods, such as Discontinuous Galerkin, that are highly susceptible to over-imposition and under-imposition of boundary conditions.
To avoid the rise of such issues and to increase the overall stability of the simulation it has been opted to avoid a clean-cut switch system between the 4 conditions - supersonic outflow, subsonic outflow, subsonic inflow, supersonic inflow - but to make use of a more smooth transition between the boundary conditions applied in those different conditions. The smoothing of the transition is dealt with the application of an appropriate function of the signed mach number, $f(M)$, that operates as follows: taking into consideration, for example, the density, the imposition of the vacuum value is done as it follows:

$$n_{BC} = (n_{dom} - n_{vac})f(M) + n_{vac} \tag{3.3}$$

where n_{dom} is the value of the density in the domain in proximity to the border, n_{vac} the vacuum density and n_{BC} the actual value of the density that will be applied at the boundary. It is important to notice that eq. (3.3) is written in a way so that:

- if $f(M) = 0$ then:

$$n_{BC} = (n_{dom} - n_{vac}) * 0 + n_{vac} \Rightarrow n_{BC} = n_{vac}$$

  This means that when $f(M) = 0$, we are in a condition where we are imposing the density as a boundary condition.

- if $f(M) = 1$ then:

$$n_{BC} = (n_{dom} - n_{vac}) * 1 + n_{vac} = n_{dom} - n_{vac} + n_{vac} \Rightarrow n_{BC} = n_{dom}$$

  This means that when $f(M) = 1$, we are in a condition where we are not imposing the density as a boundary condition.

The function $f(M)$ has to be different based on the variable it has to be applied to. For variables that have to be imposed in the subsonic outflow - normal velocity for electrons and density for ions - we need that $f(M) = 0$ when $0 < M < 1$ or when $M < -1$ (supersonic inlet, all variable imposed), whereas in the other cases $f(M) = 1$. To allow a smooth transition between imposed boundary conditions while satisfying those conditions, the function $f(M)$ was developed as a combination of different functions containing the elemental function tanh. Those functions are:

$$\begin{aligned} \mathrm{a}(M) &= \frac{\tanh(SM) + 1}{2} \\ \mathrm{b}(M) &= \frac{\tanh(S(1 - M)) + 1}{2} \\ \mathrm{c}(M) &= \frac{\tanh(S(-M - 1)) + 1}{2} \end{aligned} \tag{3.4}$$

where the parameter S is a term used to increase or decrease the width of the transition region between boundary conditions, such that, when $S \to \infty$, the

function return to provide discontinuous shift between boundary conditions, whereas when $S \to 0$, the transition becomes more smooth. In this work S has been set to 100. The function $f(M)$, shown in figure 3.2, for the subsonic outlet variables is composed as:

$$f(M) = \mathrm{a}(M) + \mathrm{b}(M) + \mathrm{c}(M) + 1 \tag{3.5}$$

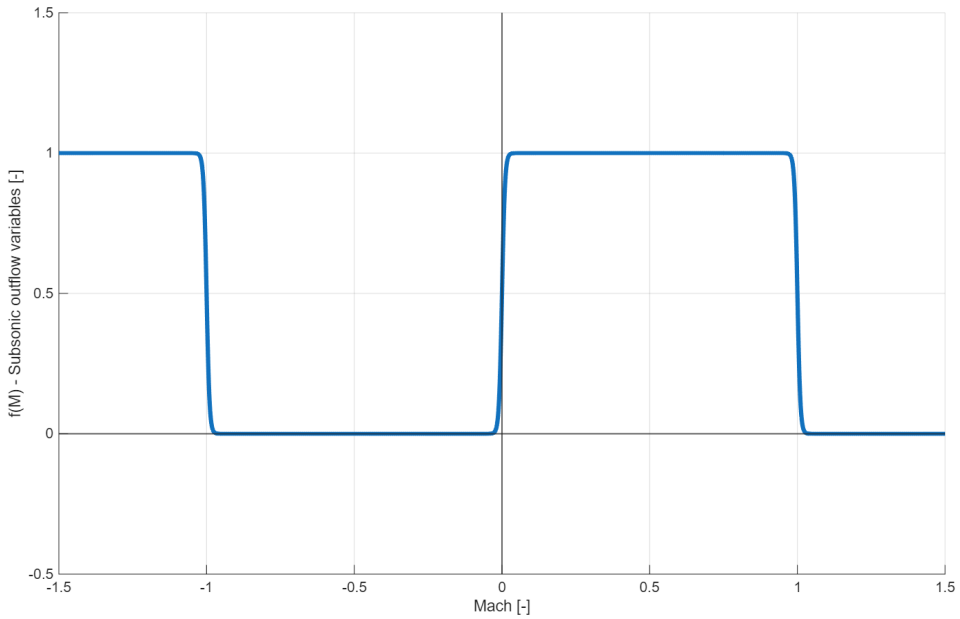


Figure 3.2: Function $f(M)$ for variables imposed in subsonic outflow

Instead for variables that we want to impose in the case of a subsonic inflow, then we need that the function $f(M)$ associated with those functions has to be 0 for $-1 < M < 0$ and for $M < -1$, while for $0 < M < 1$ and $M > 1$ it has to be 1. Therefore in this case the function $f(M)$ is defined as:

$$f(M) = \frac{\tanh(S(-M)) + 1}{2} \tag{3.6}$$

In fig. 3.3 it is represented the behaviour of the function $f(M)$ for values of Mach number between -1.5 and 1.5.

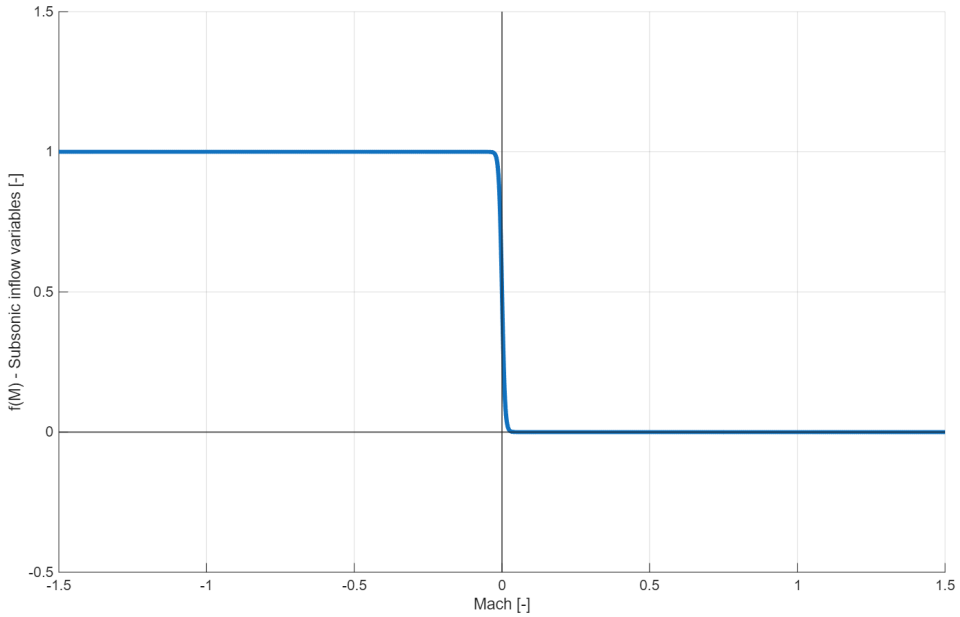


Figure 3.3: Function $f(M)$ for variables imposed in subsonic inflow

In this way have been defined the boundary conditions related to the equation as written in eq. (1.13). In the following section (3.3), given that the conservation equations will be altered by the introduction of a diffusive term, and therefore appropriate considerations regarding the boundary conditions have to be taken into account.

3.3 Artificial Viscosity

Another fundamental component in the objective of stabilizing this simulation has been the use of artificial viscosity in the computation.

This shock-capturing method has been applied to this problem in order to deal with the instabilities that generates at the border due to the sharp discontinuity between the value of variables inside the domain and the value imposed at boundary. Artificial viscosity is used to smoothen the values of the variables in the elements

more affected by those discontinuities.

Regarding this test case, it has been applied a *Laplacian artificial viscosity*, using either the Persson-Peraire sensor or the Hennemann one. From sec. 2.4, we can see that, for Laplacian artificial viscosity, the viscosity coefficient related to each element, ε_e, is given by:

$$\varepsilon_e = \varepsilon_0 \alpha_e$$

with α_e a value assigned by the sensor and ε_0 is a parameter constant inside each element, that in the simulation is computed as:

$$\varepsilon_0 = \mu_0 \, \lambda_{max,e} \, f(h,p) \tag{3.7}$$

where μ_0 is a constant computed the same for the entire domain, $\lambda_{max,e}$ is the highest absolute characteristic velocity of the element e and $f(h,p)$ a function of the cell size h and of the interpolation order p.
The constant μ_0 is defined[15] as:

$$\mu_0 = 0.27\delta\frac{\beta - 1}{\beta} \tag{3.8}$$

where δ and β are parameter, respectively called Shock Thickness and Shock Strength. The function $f(h,p)$ instead depend on the desired subcell resolution. For the simulations the subcell resolution chosen was "constant", which means that:

$$f(h,p) = \frac{h}{p} \tag{3.9}$$

In the following table, 3.7, are presented some information about the characteristic of artificial viscosity used in the simulation

With the utilization of artificial viscosity, we introduce in the conservation equations a diffusive term, eq. (2.28). The presence of a diffusive term in the conservation equations requires to supply a boundary condition for each equation. This means that other than the vacuum boundary conditions described in sec. 3.2, another set of boundary conditions has to be applied related to these equations that have no imposed boundary condition. These supplementary boundary conditions are Neumann boundary conditions, in contrast to the Dirichlet boundary conditions in sec. 3.2, and they are set to impose the gradients of the variables to be 0, as we consider that,

Name	Symbol	Value
Shock Thickness	δ	0.5, 0.8
Shock Strength	β	1.2
Interval (only for PerssonPeraire)	κ	2.0
Subcell Resolution	-	Constant
Sensor	-	Persson Peraire, Hennemann

Table 3.7: Characteristics of Artificial Viscosity

outside the domain, the diffusive fluxes related to artificial viscosity are not present, given that they are a numerical instrument to stabilize the simulation.
In order to correctly apply the Neumann boundary conditions in a way that does not go into conflict with the Dirichlet boundary conditions, they are implemented as follow:

$$g_{BC} = (g_{dom} - 0) * (1 - f(M)) + 0 \tag{3.10}$$

where g_{BC} is the value of the normal gradient of a generic variable we actually imposed, g_{dom} is the value inside domain and $f(M)$ is the function as described in 3.2, so that when $f(M) = 0$ then $g_{BC} = g_{dom}$, which means that the Neumann boundary condition is not applied, whereas when $f(M) = 1$ then $g_{BC} = 0$ and therefore the Neumann boundary condition is applied. In tab. 3.8 and tab. 3.9 are presented a summary to better clarify which Neumann boundary conditions should be applied based on the Mach number, respectively for electrons and ions.

		Density	Normal velocity	Tangential velocity	Temperature
		n	$u_\perp$	$u_\parallel$	T
Supersonic outflow	$M_n > 1$	0	0	0	0
Subsonic outflow	$0 < M_n < 1$	0	-	0	0
Subsonic inflow	$-1 < M_n < 0$	-	0	-	-
Supersonic inflow	$M_n < -1$	-	-	-	-

Table 3.8: Neumann boundary conditions scheme for electrons

		Density	Normal velocity	Tangential velocity	Temperature
		n	$u_\perp$	$u_\parallel$	T
Supersonic outflow	$M_n > 1$	0	0	0	0
Subsonic outflow	$0 < M_n < 1$	-	0	0	0
Subsonic inflow	$-1 < M_n < 0$	0	-	-	-
Supersonic inflow	$M_n < -1$	-	-	-	-

Table 3.9: Neumann boundary condition scheme for ions

3.4 Results

This section will explore the results that have been gathered through many simulations with different set-ups with the objective of stabilizing the test case. In particular a comparison between all these different set-ups based on the running time of the simulation will be first presented, while the following subsections will explore more thoroughly the results obtained and confront the issues concerning the stability.

With the objective of stabilizing the simulation to the point of reaching a non-crashing simulation, numerous attempts have been made changing through different set-ups and inputs, and the results are shown in the table 3.10. These simulations differ for various conditions, some related artificial viscosity, such as its use or not, the type of sensor used or the intensity, others instead related to the boundary conditions, particularly which variable between density, n, temperature, T, or normal velocity, $u_\perp$, is used as the electron subsonic outflow imposed condition.

Artificial Viscosity		Electrons subsonic outflow BC		
Sensor	δ	n	$u_\perp$	T
No AV	-	0.208	9.46	0.274
PP	0.5	0.180	67.26	0.357
PP	0.8	0.202	✓	0.319
H	0.5	0.219	11.14	0.241
H	0.8	0.223	8.98	0.372

Table 3.10: Crash time (ns) of simulations with different set-ups.(PP = Persson-Peraire, H = Hennemann)

3.4.1 n and T imposed as outflow

The results obtained by using the density as the imposed variable for the electrons subsonic outflow and the ones obtained by using temperature are quite similar. For these simulations, to avoid an immediate crash, and therefore to visualize what was happening, the time step is set to $\Delta t = 1$ ps. The data in tab. 3.10 related to n and T were also taken with this time step.

Starting with the one related to density, it would have been the most appropriate set-up, since it gave both the electrons and the ions the same set of boundary conditions. However, as it can be seen for tab. 3.10, this set-up is quickly lead to crashes, barely reaching around 0.2 ns of simulation time, incredibly smaller compared to the simulation time of $1 \div 10$ μs needed to observe the rotating spokes that we expect to appear in this test case.
This rapid evolution into a crash is caused by instabilities generated at the boundary, following the strong discontinuity between the value of the density inside the domain and the value we impose as a boundary condition. These instabilities are particularly important in the region around the four corners of the domain where, as shown in fig. 3.4, the value of the density becomes negative.

Even when artificial viscosity is applied, the situation doesn't improve much. In particular, when the Persson-Peraire sensor is used, the simulation resulted in lasting less compared to the case without artificial. From fig. 3.5 and fig. 3.6, it can be seen some differences compared to the case without artificial viscosity, mainly that the density does not appear, from the obtained results, to become negative before the crash, however values very close to 0 are still present in the regions near the corners, so it is possible that some instabilities may have generated from this regions, possibly also due to the effect of a too high artificial viscosity.

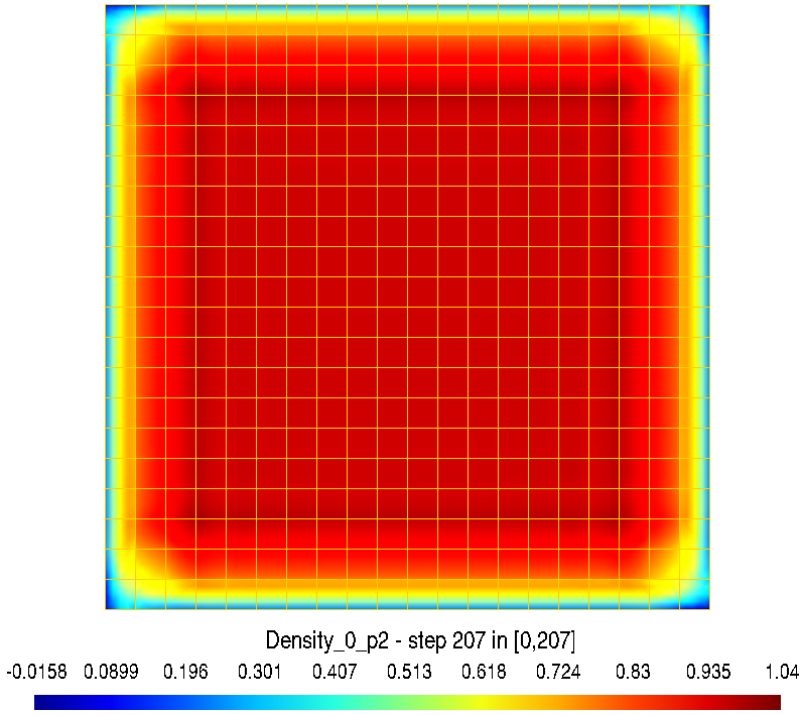
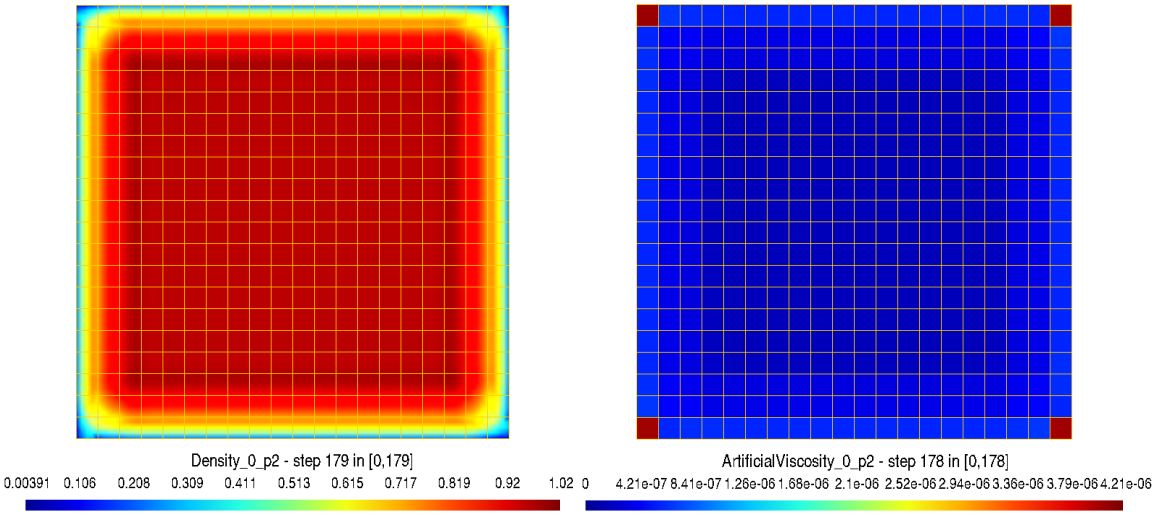


Figure 3.4: Electrons density - Results for case: density boundary condition, no artificial viscosity. Time: 0.207 ns

Instead, when an Hennemann sensor is used, that is typically less sensible to smaller shocks and therefore more focused on the strongest ones, the simulation manage to push a bit forward compared to the case without artificial viscosity. In fig. 3.7 and fig. 3.8 are presented the results for the density and the artificial viscosity with the Hennemann sensor at the same timestep as the last available for the case without artificial viscosity. It can be seen that the minimum of the density is higher compared to the one in fig. 3.4 (which is negative), sign that the artificial viscosity is indeed helping in smoothing the more problematic regions of the domain. However, even with this type of artificial viscosity has not been possible to obtain a stable running simulation given this kind of boundary conditions.

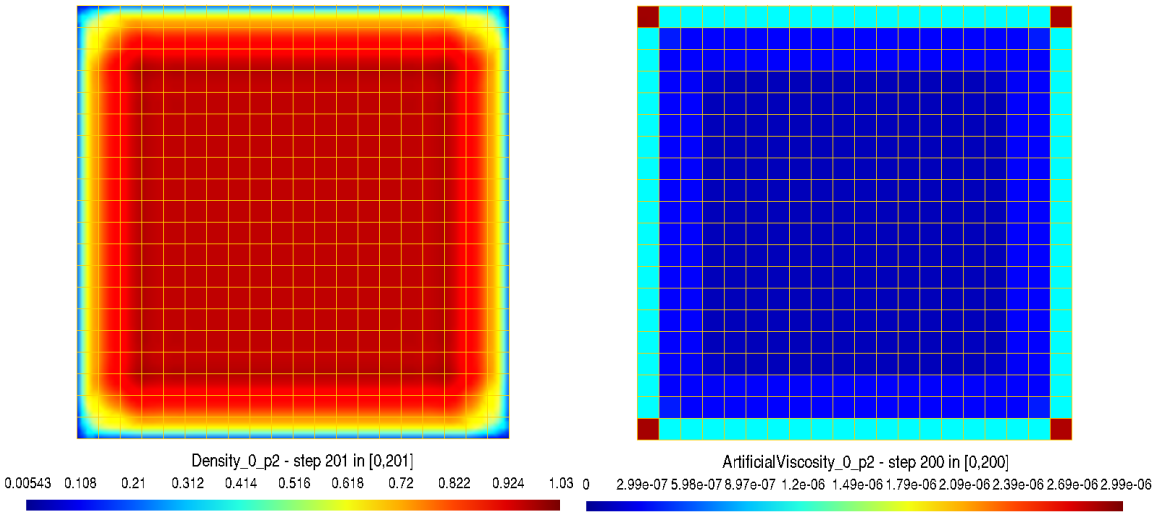
Regarding the set-up where the temperature is used as the imposed boundary condition for electron in case of a subsonic outflow, it produce similar results and problems as the set-up with density.

Fig. 3.9a shows the profile of the density of the second to last available step of the simulation without artificial viscosity, and it can be seen that there are regions where the density had become negative, leading to instabilities and numerical errors,



(a) Electrons density (b) Electrons artificial viscosity

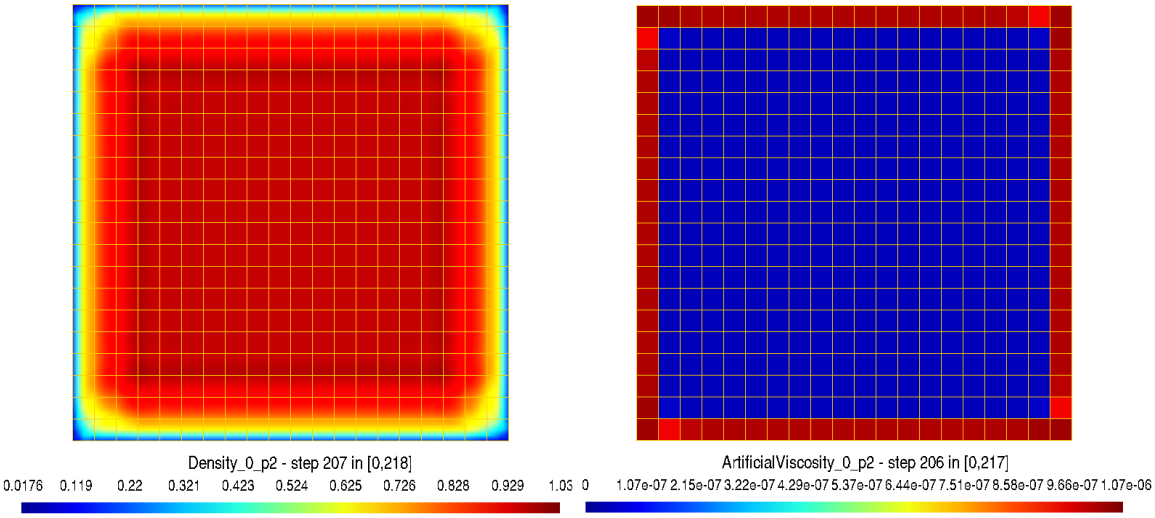
Figure 3.5: Results for case: density boundary condition, Persson-Peraire sensor, $\delta = 0.5$. Time: 0.179 ns



(a) Electrons density (b) Electrons artificial viscosity

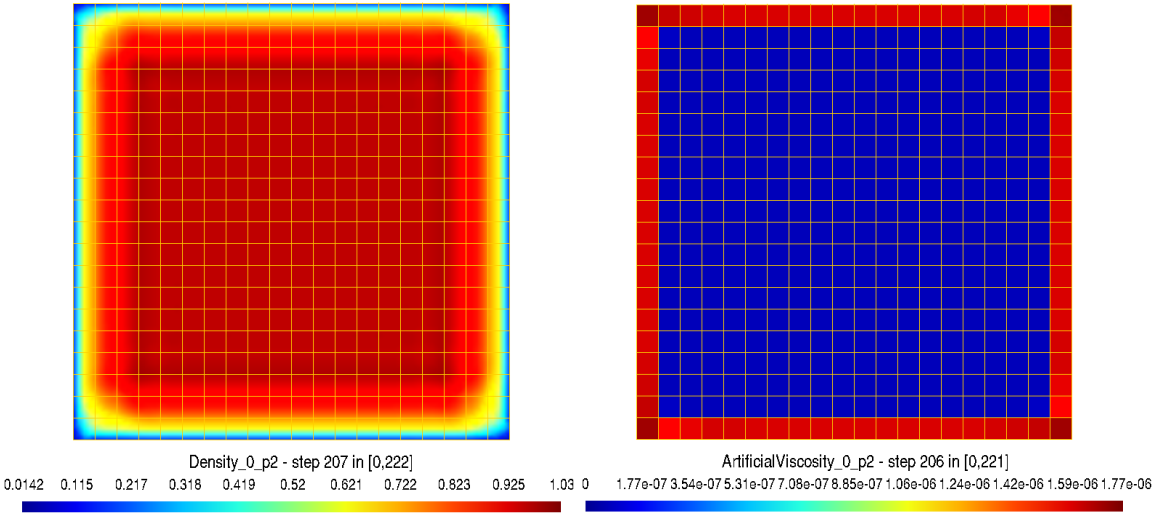
Figure 3.6: Results for case: density boundary condition, Persson-Peraire sensor, $\delta = 0.8$. Time: 0.201 ns

which can be visualized in the last time step, fig. 3.9b, causing a crash. Similarly



(a) Electrons density (b) Electrons artificial viscosity

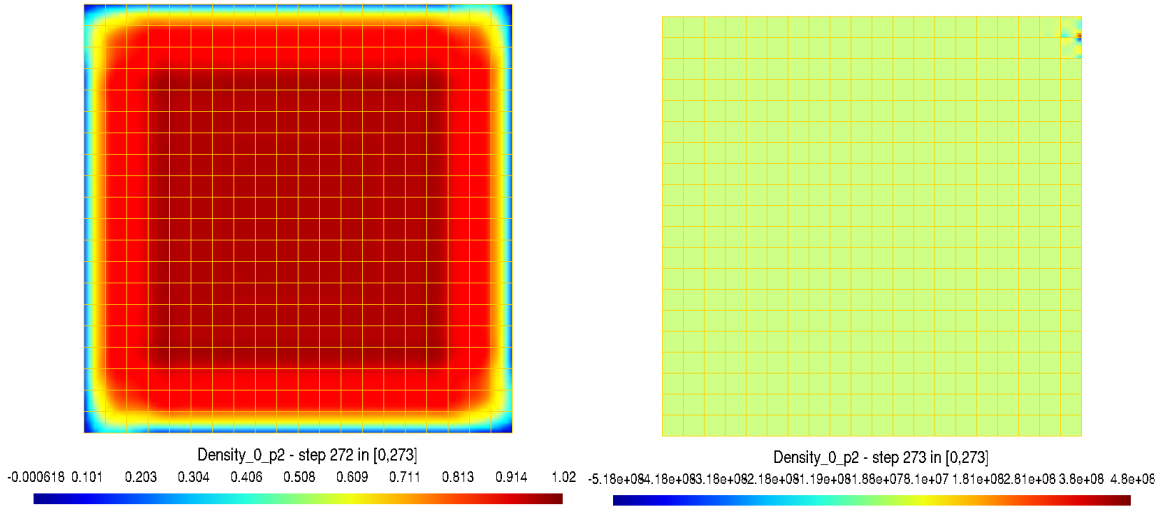
Figure 3.7: Results for case: density boundary condition, Hennemann sensor, $\delta = 0.5$, time of crash for case without artificial viscosity (0.207 ns)



(a) Electrons density (b) Electrons artificial viscosity

Figure 3.8: Results for case: density boundary condition, Hennemann sensor, $\delta = 0.8$, time of crash for case without artificial viscosity (207 ns)

can be said for energy, presented in figs. 3.9c and 3.9d.



(a) Electrons density - Time: 0.272 ns

(b) Electrons density - Time: 0.273 ns

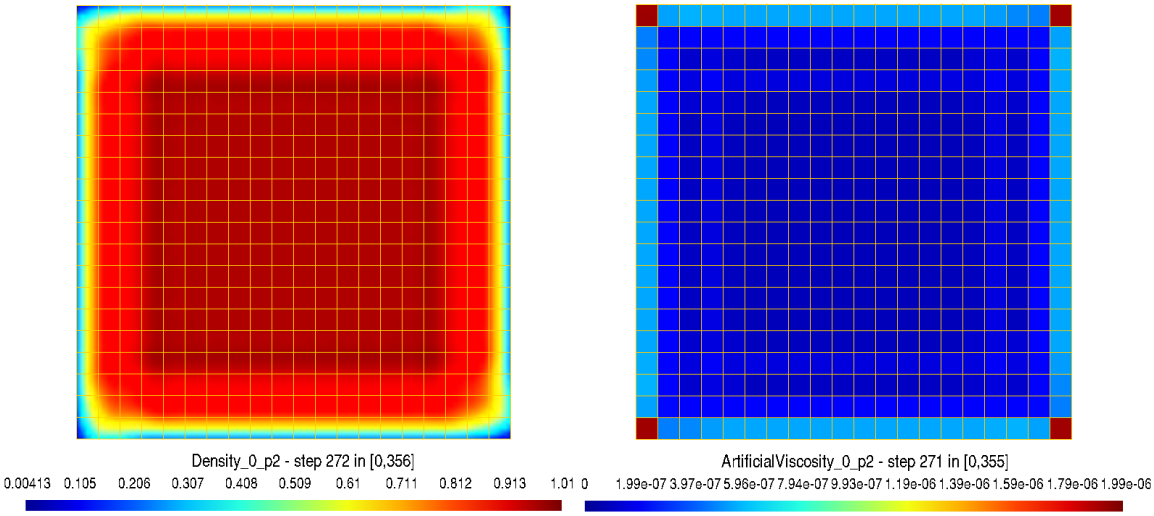
(c) Electrons energy - Time: 0.272 ns

(d) Electrons energy - Time: 0.273 ns

Figure 3.9: Results for case: temperature boundary condition, no artificial viscosity. Time: 0.272 ns – 0.273 ns

It is interesting to see the effects of artificial viscosity on the simulations with the temperature set-up, which greatly varies depending on the sensor and the value of the shock thickness δ. Considering the Persson-Peraire sensor, we can see that, contrary to the case of density as boundary condition, the simulations with this sensor have both surpassed the running time of the case without artificial viscosity. In particular

it can be noticed that when δ was equal 0.5, the simulation lasted longer compared to $\delta = 0.8$. As shown in figs. 3.10 and 3.11, artificial viscosity manages to correctly avoid the instabilities that appeared in fig. 3.9. However, it still does not manage to stabilize the simulation, as it crash after a few more steps.



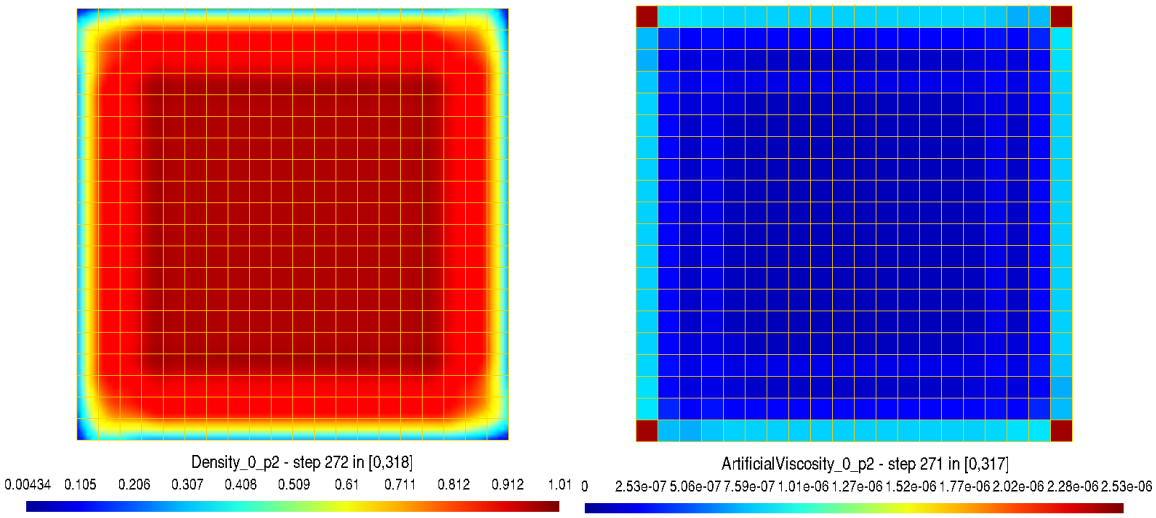
(a) Electrons density

(b) Electrons artificial viscosity

Figure 3.10: Results for case: temperature boundary condition, Persson Peraire sensor, $\delta = 0.5$, second to last time step before crash without artificial viscosity (0.272 ns)

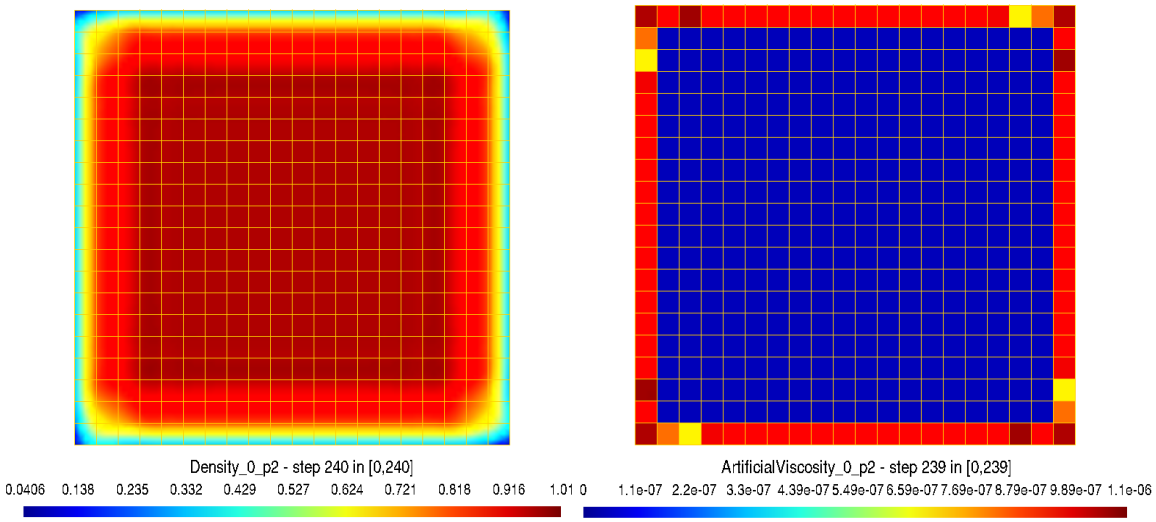
When instead the Hennemann sensor is applied, the results were very different, with the case $\delta = 0.5$ (fig. 3.12) that crashes before the simulation without artificial viscosity, whereas the case $\delta = 0.8$ (fig. 3.13)ran for the longest simulation time for this choice of boundary conditions.

The different results obtained in the application of artificial viscosity between the set-up with density and the one with temperature, but also between different sensors, present a good example of the sensibility of artificial viscosity as a shock capturing method, as its behaviour varies greatly based on the characteristics of the problem it is applied to.



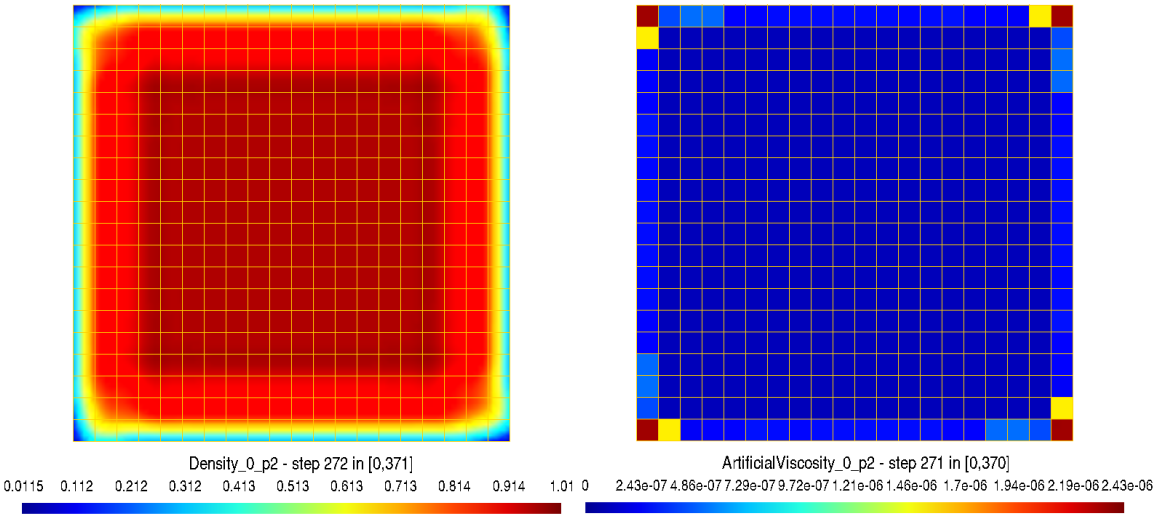
(a) Electrons density (b) Electrons artificial viscosity

Figure 3.11: Results for case: temperature boundary condition, Persson Peraire sensor, $\delta = 0.8$, second to last time step before crash without artificial viscosity (0.272 ns)



(a) Electrons density (b) Electrons artificial viscosity

Figure 3.12: Results for case: temperature boundary condition, Hennemann sensor, $\delta = 0.5$. Time: 0.240 ns



(a) Electrons density

(b) Electrons artificial viscosity

Figure 3.13: Results for case: temperature boundary condition, Hennemann sensor, $\delta = 0.8$, second to last time step before crash without artificial viscosity (0.272 ns)

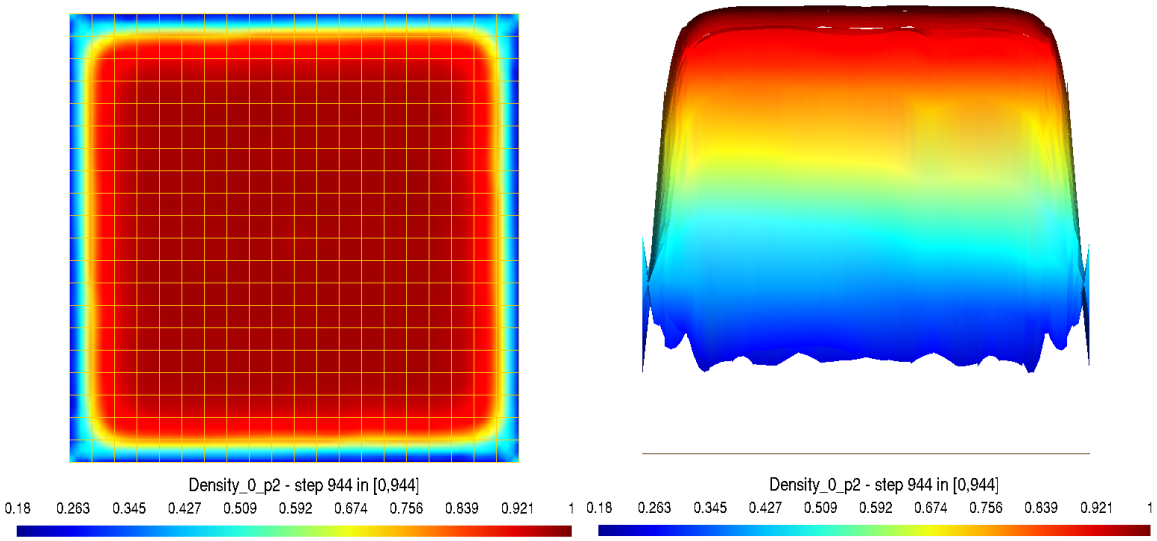
3.4.2 $u_\perp$ imposed as outflow

The most interesting results are the ones obtained in the simulations where the imposed boundary condition for a subsonic outflow for electrons is the perpendicular velocity, $u_\perp$, as, differently from when density or temperature are imposed, in this case the simulation time manage to reach the order of ns, and even, with the contribution of artificial viscosity, to become stable, as in that it showed no sign of a possible crash even after a long simulation time.

Starting from the case without the use of artificial viscosity, the evolution of the simulation at the boundary is much slower compared to the other two set-up presented, allowing for a more smooth gradient of values of quantities, such as density, in proximity to the boundary, avoiding the issues that were present in the other cases. In particular the region around the corners, that before was a great source of instabilities, it seems to be less critical with this set of boundary conditions. However even in this case the simulation does not result to not be stable, ending in a crash (fig. 3.14a). After some time it was possible to observe in the region of the boundary the formation of some oscillations, as shown in fig. 3.14b[1], that could be related to

[1]This image, as well as the others like this, is obtained by translating in the normal direction the

the crash.

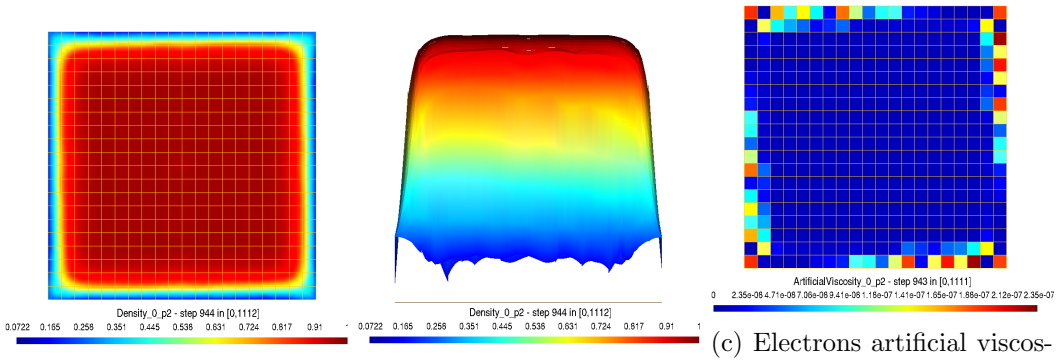


(a) Electrons density (b) Electrons density

Figure 3.14: Results for case: normal velocity boundary condition, no artificial viscosity. Time: 9.44 ns

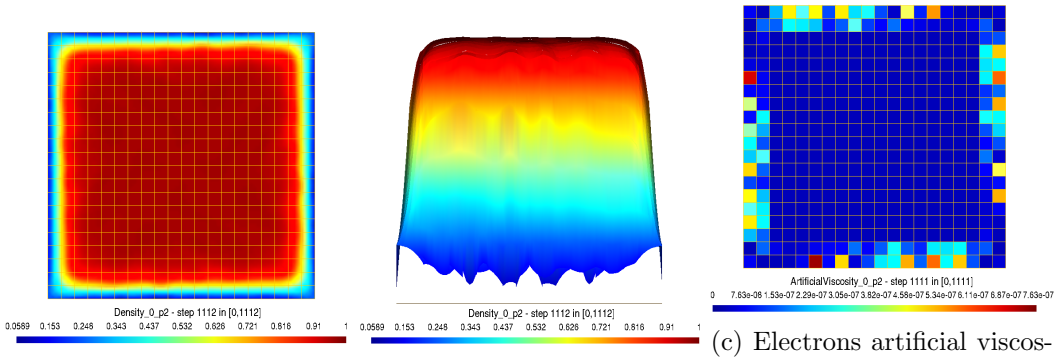
To deal with this new problems at the boundary, artificial viscosity has been used, with greatly different results between the chosen sensor. Starting with the Hennemann sensor, this one has provided little improvements to the simulation stability ($\delta = 0.5$) or even made the situation worse ($\delta = 0.8$). As can be seen in figs. 3.15, 3.16 and 3.17, the same non-uniformities at the boundary that were present for the case without artificial viscosities. These formations seems to not be natural but some form of numerical error, in that way it could also be possible to explain the reason why the artificial viscosity seems to lose the expected symmetry it should show.
To sum up, given the results obtained, artificial viscosity applied with an Hennemann sensor doesn't seem to be able to correctly solve the stability issues that this simulation has.

profile, for example, of the density of a factor equal to $n * 0.001$. This type of figures help visualize the value of the density in the domain and find eventual non-uniformity.



(a) Electrons density (b) Electrons density (c) Electrons artificial viscosity

Figure 3.15: Results for case: normal velocity boundary condition, Hennemann sensor, $\delta = 0.5$, time of crash without artificial viscosity (9.44 ns)

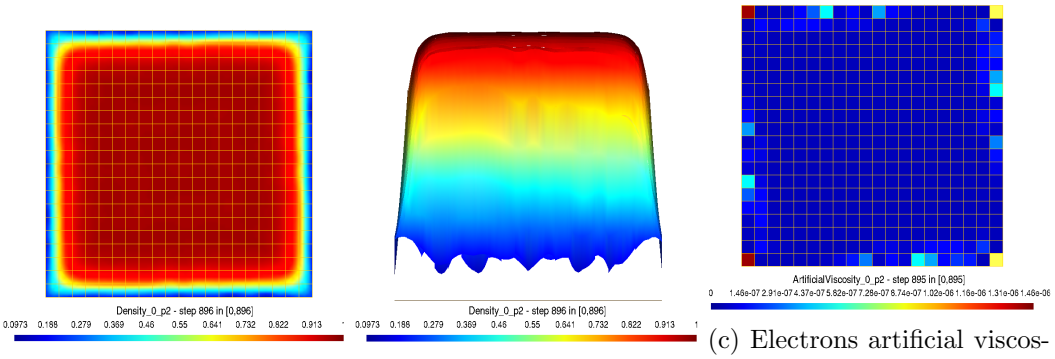


(a) Electrons density (b) Electrons density (c) Electrons artificial viscosity

Figure 3.16: Results for case: normal velocity boundary condition, Hennemann sensor, $\delta = 0.5$. Time: 11.12 ns

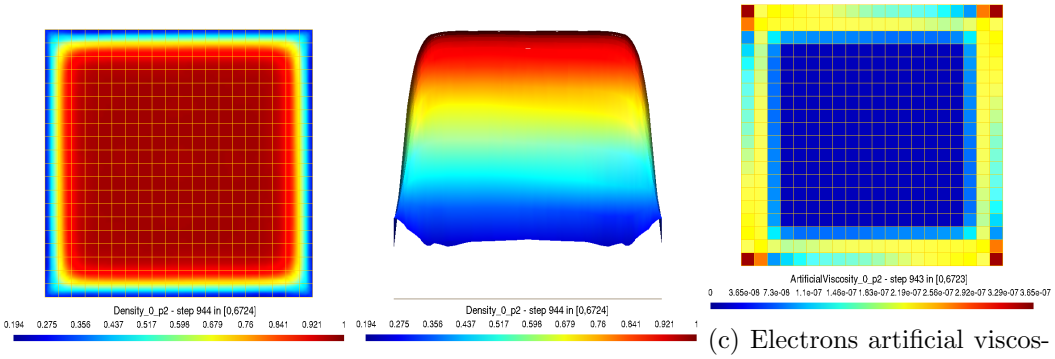
The situation instead change when artificial viscosity is applied to the simulation with the Persson-Peraire sensor. In the case with $\delta = 0.5$, the simulation time improves greatly, passing from 9.46 ns to 67.26 ns. It can be observed in fig. 3.18 that the non-uniformities at the boundary, seen in both the non artificial viscosity case and those with the Hennemann sensor, do not appear at the same time as the previous simulations, but a bit later, at around 11 ns, and they appear to be less pronounced than before.

Looking a bit later in the simulation, new spikes and irregularities start to appear in regions close to the boundary. These new formations manifest around 50 ns, and



(a) Electrons density (b) Electrons density (c) Electrons artificial viscosity

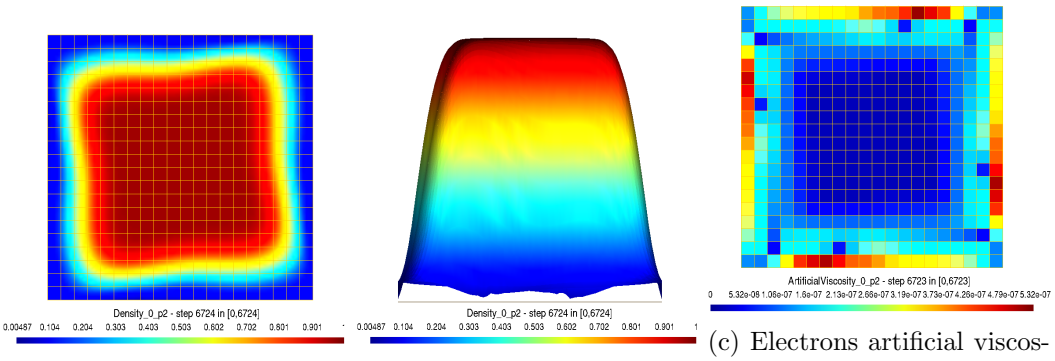
Figure 3.17: Results for case: normal velocity boundary condition, Hennemann sensor, $\delta = 0.8$. Time: 8.96 ns



(a) Electrons density (b) Electrons density (c) Electrons artificial viscosity

Figure 3.18: Results for case: normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.5$, time of crash without artificial viscosity (9.44 ns)

they persist until the end of the simulation (fig. 3.19), probably contributing to the crash of the same.

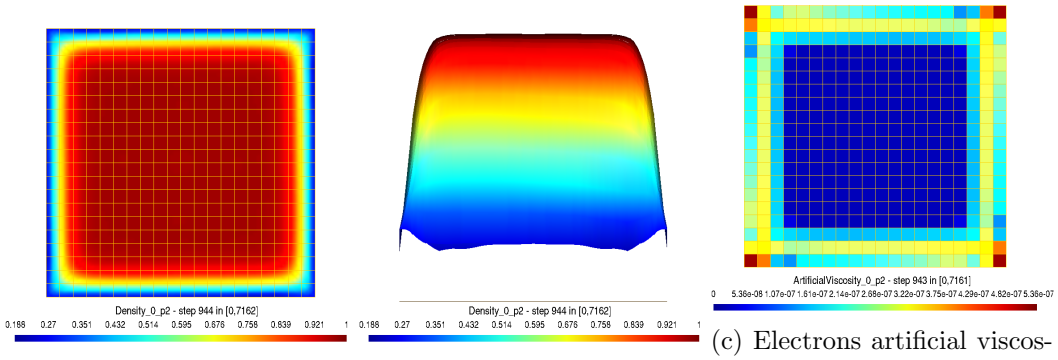


(a) Electrons density (b) Electrons density (c) Electrons artificial viscosity

Figure 3.19: Results for case: normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.5$. Time: 67.24 ns

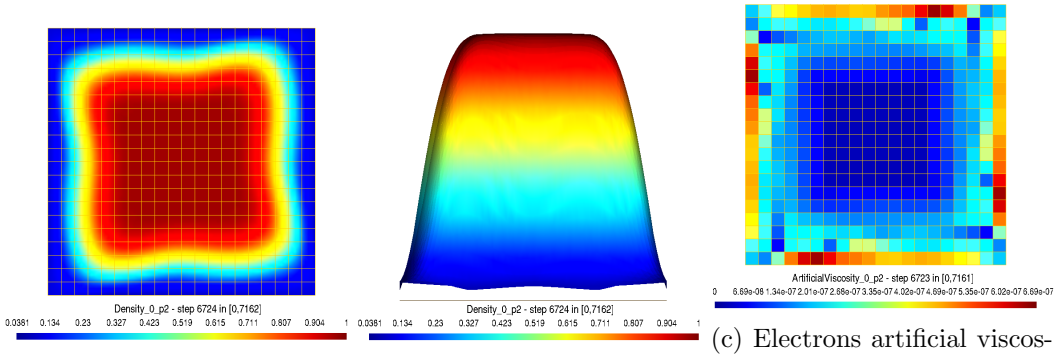
Out of all the different set-up tried and presented in this work, the only one that managed to obtain a stable simulation is characterized by: normal velocity as boundary condition for subsonic outflow of electrons and artificial viscosity with Persson-Peraire sensor and $\delta = 0.8$.
Similarly to the other simulations with $u_\perp$ as a boundary condition, this simulation manage to avoid the issues and instabilities that were observed when instead it was used n or T for the same scope.
This simulation with Persson Peraire and $\delta = 0.8$ results to be different from the others that use $u_\perp$ as a boundary condition as the spikes and non-uniformities that were present in the cases without artificial viscosity and when Hennemann sensor was applied, while still appearing around 9 ns, are much less evident and do not cause a crash in the simulation (fig. 3.20) - they also appear to behave more like a wave that some instability - similar to what observed in the other simulation with a Persson-Peraire($\delta = 0.5$).
Compared to this other simulation, however, this one that use an higher value of shock-thickness manage to avoid completely the formation of the new spikes later in the simulation, ensuring that the simulation do not crash at around $60 \div 70$ ns, but instead the region around its boundary evolves in a slow and uniform way.

The simulation characterized by normal velocity as a boundary condition for the subsonic outflow of electrons and the use of artificial viscosity with a Persson-Peraire sensor and shock-thickness $\delta = 0.8$ managed to be stable, allowing us for the first time to reach simulation time of order hundreds of ns to μs, and showed us the evolution of the plasma inside the domain, particularly the evolution of the ions,



(a) Electrons density (b) Electrons density (c) Electrons artificial viscosity

Figure 3.20: Results for case: normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.8$, time of crash without artificial viscosity (9.44 ns)



(a) Electrons density (b) Electrons density (c) Electrons artificial viscosity

Figure 3.21: Results for case: normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.8$, time of crash (67.24 ns) of the case normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.5$

which, for the first time, have enough time to evolve in a meaningful way.

Looking at how the density of the electrons changes through time, presented in fig. 3.22, it shows that, after an initial period, figs. 3.22a, 3.22b and 3.22c, where the simulation seems to stabilize and the density around the boundary becomes more and more close to the vacuum density, the bulk of the density left in the domain seems to start to rotate in a counter-clockwise way. This rotation can be visualized in figs. 3.22d, 3.22e and 3.22f. To determine if this rotation was one of the rotating spokes that we wanted to observe from this test case, the data of the electron density

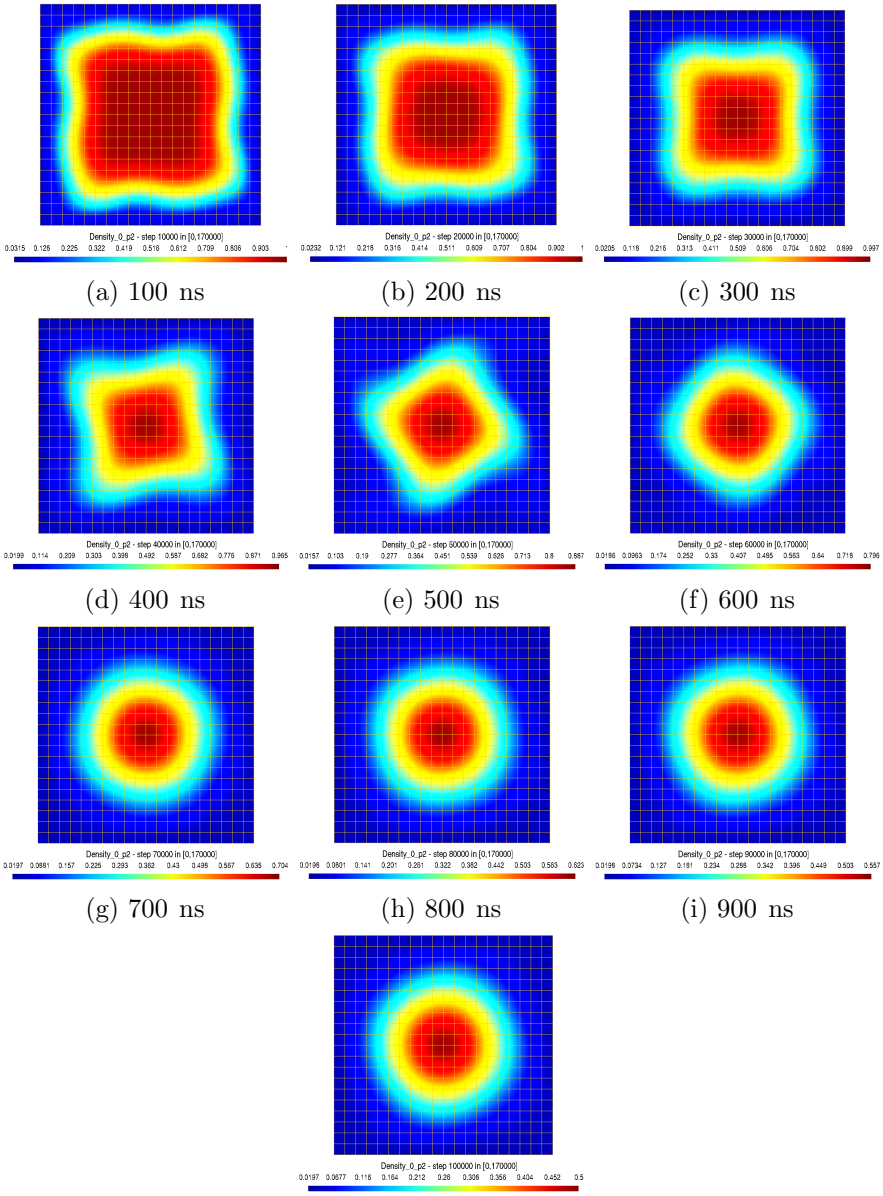
has been analysed by taking in consideration the values of this variable over time in a point interest for the rotation and carrying out a Fourier transformation to find the dominant wave frequency f of this physical instability, in order to find its phase wave velocity through the equation:

$$v_{phase} = \frac{2\pi r_p f}{m} \tag{3.11}$$

where r_p is the distance of the chosen point from the centre of the instability[2] and m is the number of wave lobes. The point of interest chosen was positioned at 0.17 mm to the right of the centre of the domain and are observable for this position 4 lobes, obtaining a wave velocity of $v_{phase} = 887.16$ m/s, comparable with the ion thermal velocity which, considering the problem at hand, result to be $v_{th,i} = 850.50$ m/s.
After some time it can be noticed that the bulk of the density appears to reduce to a circular form, and remains like this through time, as can be seen in figs. 3.22g to 3.22j. It is important to notice that, for the way the simulation is formulated, in particular the lack of a source term for the particles, the density inside the domain will slowly start to decrease, until in the whole domain the density is the same as the vacuum density.

Another interesting aspect to observe is the evolution of the density of ions inside the domain. As it shown in fig. 3.24, it can be observed that the evolution of the ions is quite different from that of the electrons, in particular at the start, as the ions need more time to start their motion, therefore even after $100 - 200$ ns, figs. 3.24a and 3.24b the region close to the boundary is still at the a phase where most of the particles have just started to feel the vacuum outside. A little after 400 ns we can observe, figs. 3.24d to 3.24i that, similarly to what happened for the density of the electrons, the central region of the domain - where the great part of the ions is - starts to rotate. This rotation continue all the way till around 1 μ s, where the central part becomes a circle, not showing a clear sign of rotation anymore. It has to be noted that even in this case, as the simulation time progresses, the total number of ions inside the domain decrease.

[2]In this case the centre of the domain.



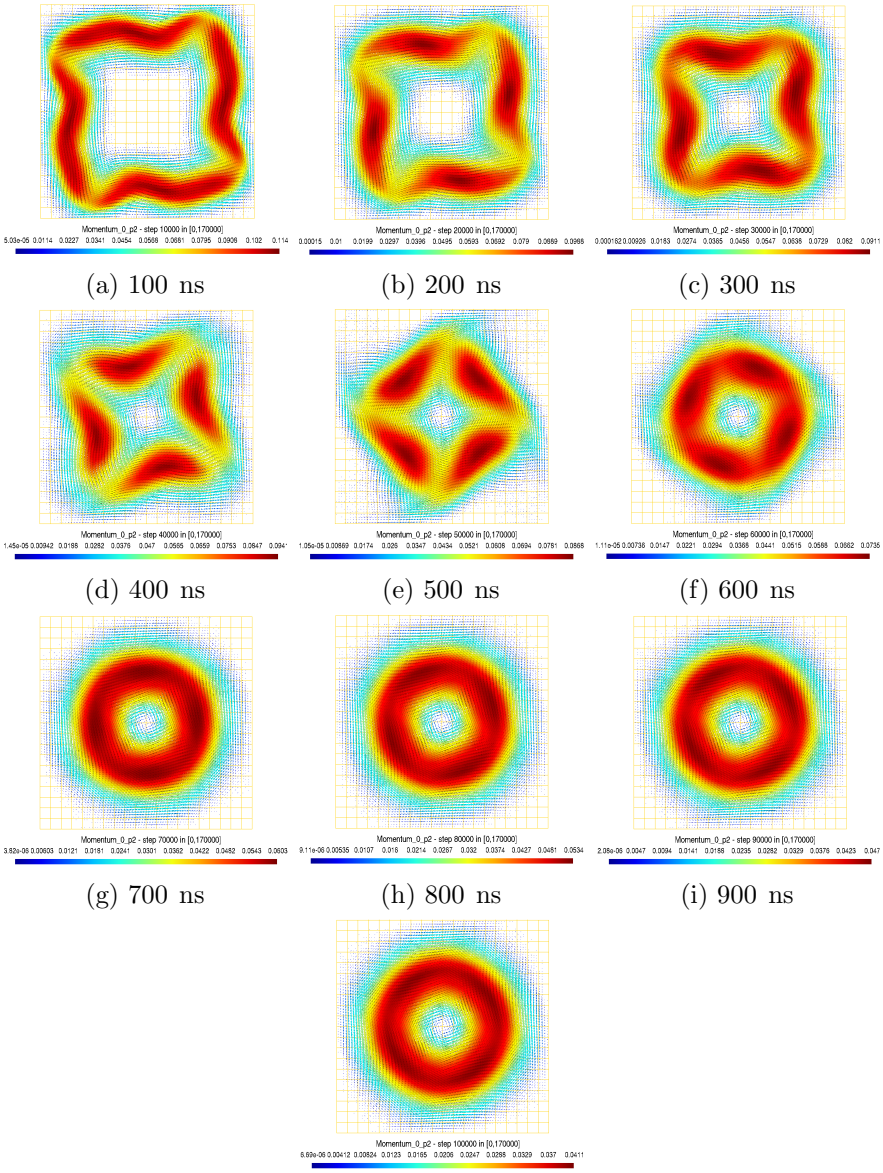
(a) 100 ns (b) 200 ns (c) 300 ns

(d) 400 ns (e) 500 ns (f) 600 ns

(g) 700 ns (h) 800 ns (i) 900 ns

(j) 1000 ns

Figure 3.22: Electron density - test case: normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.8$



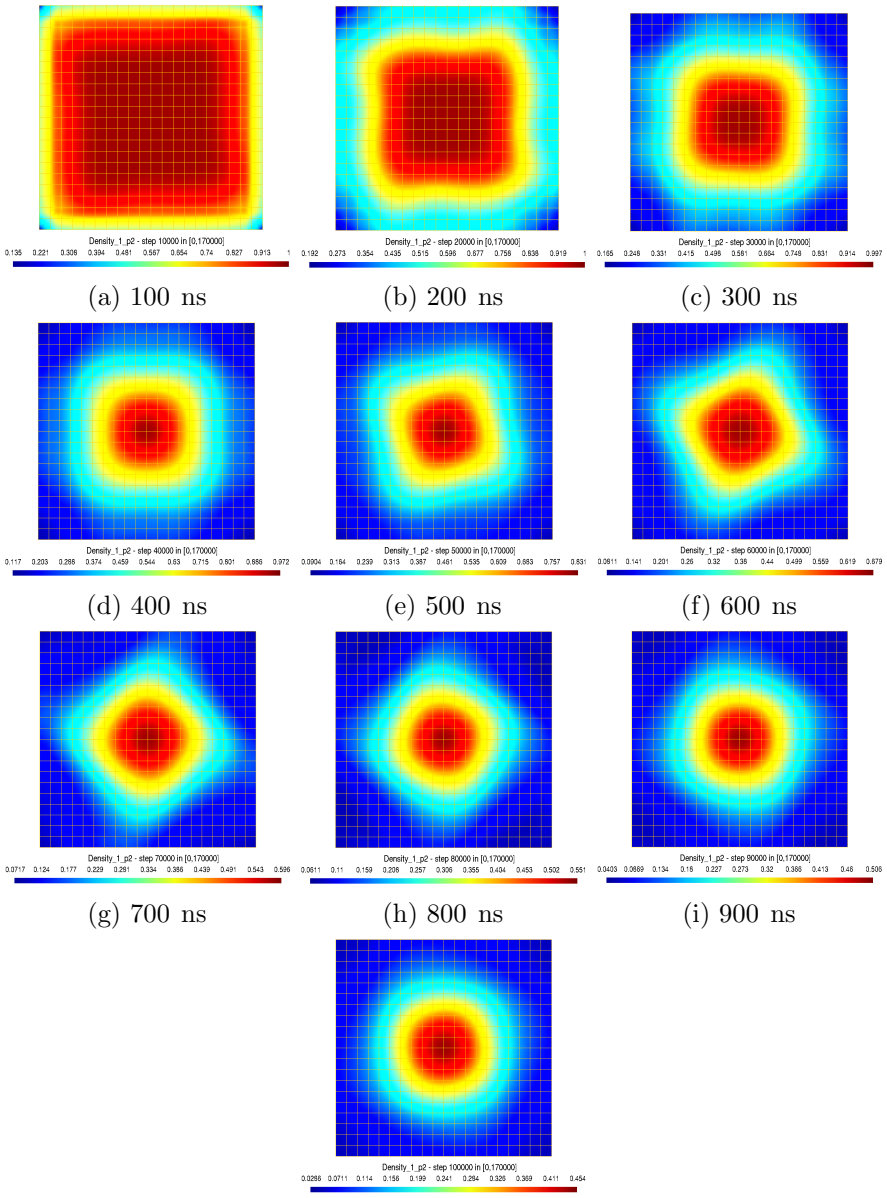
(a) 100 ns (b) 200 ns (c) 300 ns

(d) 400 ns (e) 500 ns (f) 600 ns

(g) 700 ns (h) 800 ns (i) 900 ns

(j) 1000 ns

Figure 3.23: Electron momentum vector - test case: normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.8$



(a) 100 ns (b) 200 ns (c) 300 ns

(d) 400 ns (e) 500 ns (f) 600 ns

(g) 700 ns (h) 800 ns (i) 900 ns

(j) 1000 ns

Figure 3.24: Ion density - test case: normal velocity boundary condition, Persson-Peraire sensor, $\delta = 0.8$

Chapter 4

Conclusions

The main focus of this work has been the stabilization of the Penning discharge test case in an high-order simulation with a multi-fluid model, an approach that differs from what is usually found in literature, where is more common to see PIC simulation to deal with this test case.
The stability of the test case was compromised by instabilities and numerical errors generated in proximity to the boundary, where strong shocks formed due to the imposed boundary conditions with the intent of simulating vacuum outside the domain. Moreover the academic standard to simulate vacuum conditions in plasma-dynamics usually relies on an over-imposition of boundary conditions, which, while it can be allowed for zero-th or first order simulations, for high-order simulations it must be avoided as it introduces severe numerical errors.
The solution proposed made use of the physics of the problem to define a set of boundary condition which managed to avoid an almost instant simulation failure. Their actual implementation involved a strategy to avoid over-imposition of boundary conditions in the event of rapid and frequent changes in the number of boundary conditions that should be imposed, such as a flux parallel to the boundary for the test case at hand, continuously switching between an outgoing and an ingoing flux, by smoothing the passage between a condition and another.
To contrast the formation of instabilities artificial viscosity, a shock-capturing method, has been applied to the computation. While artificial viscosity had already been implemented for mono-species fluids, it was adapted to also work in a coherent way for the multi-fluid model of plasmas. Artificial viscosity proved to be an important component in the stabilization of the simulation, managing to solve different instabilities that formed in the computation, however, as the results show, this shock-capturing method requires an trial and error process to discover the most efficient type of

sensor and intensity of the artificial viscosity for the problem at hand, making it a complicate tool to use.
Through the combined effects of the boundary conditions implemented and of artificial viscosity, a long-lasting and stable simulation of the Penning discharge test case has been established, allowing the visualization of new form of physical instabilities in the plasma, expected in this type of test case.
While these results are encouraging, they account for only a step toward the complete development of a Penning discharge test case. Future developments include: to introduce of a source term to maintain the number particle density, to verify that artificial viscosity does not alter the physical instabilities observed in the test case.
It is important, however, to note that, while a stability has been reached, it is quite fragile, as small changes in the parameters defining the boundary conditions or artificial viscosity may results in new simulation failure. This issue shows a root problem in the difficult implementation of the Penning discharge test case in the approach presented in this work, as this test case, which is presented in literature in a way that provoke strong shocks right at the boundary, is hardly compatible with a high-order simulation, which generates numerical instabilities in the presence of strong shocks and needs particular attention in the correct implementation of boundary conditions.

Bibliography

[1] Claudio Chiuderi and Marco Velli. *Fisica del plasmi: Fondamenti e applicazioni astrofisiche*. UNITEXT. Springer Milano, 2012. ISBN: 978-88-470-1847-1. DOI: 10.1007/978-88-470-1848-8.

[2] Julien Clotuche. "Improving the Conditioning of the Acoustic Subsystem and Source Terms in Multifluid Plasma Equations of State Integration Internship". Unpublished master's thesis. Université de Liège, Liège, Belgique, 2023. URL: https://matheo.uliege.be/handle/2268.2/16754.

[3] Sebastian Hennemann et al. "A provably entropy stable subcell shock capturing approach for high order split form DG for the compressible Euler equations". In: *Journal of Computational Physics* (Oct. 21, 2020). DOI: 10.1016/j.jcp.2020.109935.

[4] Koen Hillewaert. "Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries". PhD thesis. UCLouvain, 2013. URL: https://hdl.handle.net/2078.5/205460 (visited on 04/26/2026).

[5] Koen Hillewaert. *Notes - "The discontinuous Galerkin method: Introduction to the theory and practice"*. Université de Liège, 2025.

[6] Koen Hillewaert, Cecile GOFFAUX, and Corentin Carton de Wiart. "Development Of A High-order Discontinuous Galerkin Method For The DNS And LES Of Natural Convection Flows In Buildings." In: Aug. 28, 2013. DOI: 10.26868/25222708.2013.1501.

[7] Charles Hirsch. *Numerical Computation of Internal and External Flows*. Second. Vol. 1. Elsevier/Butterworth-Heinemann, 2007.

[8] R Kawashima, K Hara, and K Komurasaki. "Numerical analysis of azimuthal rotating spokes in a crossed-field discharge plasma". In: *Plasma Sources Science and Technology* 27.3 (Mar. 2018), p. 035010. ISSN: 0963-0252. DOI: 10.1088/

1361-6595/aab39c. URL: https://doi.org/10.1088/1361-6595/aab39c (visited on 04/30/2026).

[9] Nayan Levaux et al. *A monolithic Cartesian extended discontinuous Galerkin solver for conjugate heat transfer*. Rochester, NY, July 7, 2026. DOI: 10.2139/ssrn.7070226. URL: https://papers.ssrn.com/abstract=7070226 (visited on 07/08/2026).

[10] A R Mansour and K Hara. "Full fluid moment modeling of rotating spokes in Penning-type configuration". In: *Plasma Sources Science and Technology* 31.5 (May 2022), p. 055012. DOI: 10.1088/1361-6595/ac6a73. URL: https://doi.org/10.1088/1361-6595/ac6a73.

[11] S Mazouffre et al. "Rotating spoke instabilities in a wall-less Hall thruster: experiments". In: *Plasma Sources Science and Technology* 28.5 (May 2019), p. 054002. DOI: 10.1088/1361-6595/ab07fc. URL: https://doi.org/10.1088/1361-6595/ab07fc.

[12] Per-Olof Persson and J. Peraire. "Sub-Cell Shock Capturing for Discontinuous Galerkin Methods". In: *AIAA paper* 2 (Jan. 9, 2006). DOI: 10.2514/6.2006-112.

[13] Antoine Reuter. "Implementation of a monolithic convex limiting approach for the stabilization of high speed turbulent simulations using a Discontinuous Galerkin method Integration internship". Unpublished master's thesis. Université de Liège, Liège, Belgique, 2025. URL: https://matheo.uliege.be/handle/2268.2/24768.

[14] Magin Thierry. *Lecture notes - "Physical Gas Dynamics"*. von Karman Institute for Fluid Dynamics, 2010.

[15] Ray Vandenhoeck and Andrea Lani. "Implicit high-order flux reconstruction solver for high-speed compressible flows". In: *Computer Physics Communications* 242 (Sept. 1, 2019), pp. 1–24. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2019.04.015. URL: https://www.sciencedirect.com/science/article/pii/S0010465519301328 (visited on 07/09/2026).