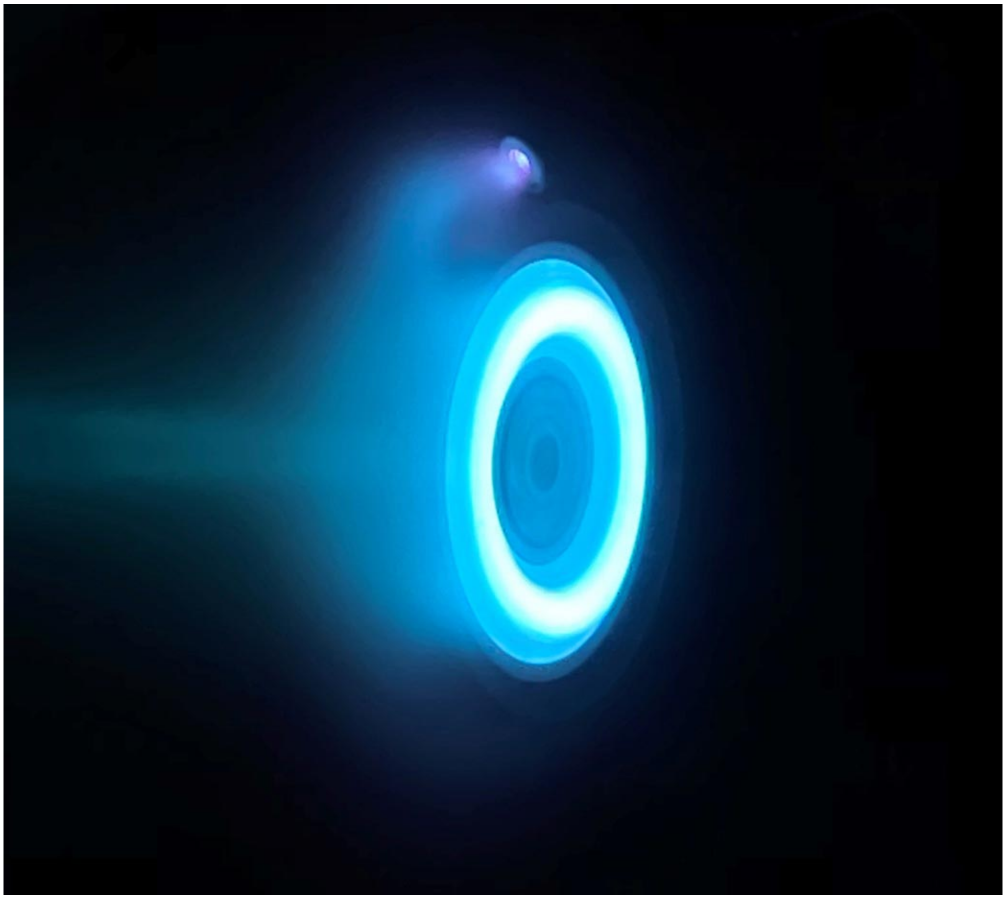
Travail de Fin d'Etudes : Discontinuous Galerkin Finite Element Method Applied to Plasma Flows

Auteur : Corthouts, Nicolas
Promoteur(s) : Hillewaert, Koen
Faculté : Faculté des Sciences appliquées

Diplôme : Master en ingénieur civil physicien, à finalité approfondie
Année académique : 2019-2020
URI/URL : http://hdl.handle.net/2268.2/10171

Discontinuous Galerkin Finite Element Method Applied to Plasma Sheaths



Author: Nicolas Corthouts

Advisors: Koen Hillewaert, Thierry Magin

August 21, 2020

Acknowledgements

J'aimerais remercier tout d'abord ma famille : ma mère et mon père pour leur soutien constant, pas seulement pour ce travail, mais aussi tout au long de mes études. Ma petite soeur Dana, dont les connaissances en anglais ont corrigé bien des rapports. Ma petite amie Sarah, pour son soutien quotidien et nos séances d'étude intensives.

J'aimerais aussi remercier Giuseppe Gangemi pour sa patience et sa disponibilité, même durant ses vacances.

Enfin, je remercie mes promoteurs, Koen Hillewaert et Thierry Magin, qui m'ont montré la voie lorsque j'étais perdu dans les méandres des méthodes numériques et des écoulements de plasmas.

First, I would like to thank my family : my mother and my father for the constant support, not only during this work, but also all along my cursus. My little sister Dana, whose knowledge in English has corrected so many reports. My girlfriend Sarah, for its daily support and the intense study sessions.

I also would like to thank Giuseppe Gangemi for his patience and availability, even during its holidays.

Finally, I thank my two advisors, Koen Hillewaert and Thierry Magin, who showed me the path when I was lost in the deep labyrinth of numerical methods and plasma flows.

Contents

Introduction

The numerical resolution of problems involving plasmas is of capital importance. In particular, low temperature plasmas are part of many technological applications : from the electrical propulsion (Hall thrusters, Fig. 1) to lighting. This relatively wide range of applications gave rise to many different numerical methods [1, 2], each adapted to specific requirements. Two main approaches are commonly used: the kinetic approaches, which consist of describing the behaviour of particles of plasma, and fluid/global models, which describe the average motion. The former can give very accurate results, but has a very high numerical cost, while the latter is less precise but keeps a reasonnable computation time. In practice, hybrid models (associating both a kinetic and global model) are used (see *e.g.* [3]).

The main difficulty in plasma numerical solvers is the large disparity between the electron and ion mass, which makes the problem very stiff. This mass ratio has an impact on the plasma physics for low pressure plasmas. Indeed, at low pressure, the collisions between electrons and ions are not as frequent as in the high pressure case. It results in a disparity in the electrons and ions motions. This disparity is at the origin of many physical phenomena: two-stream instabilities [6], electron drift instability in Hall thrusters [7, 8, 9] or even plasma sheaths [10].

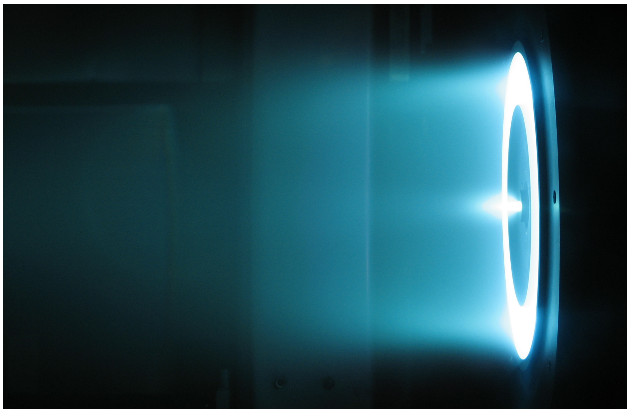


Figure 1: Xenon Hall thruster, one of the many possible plasma applications. [4] The cover image represents a thruster of the same kind. [5]

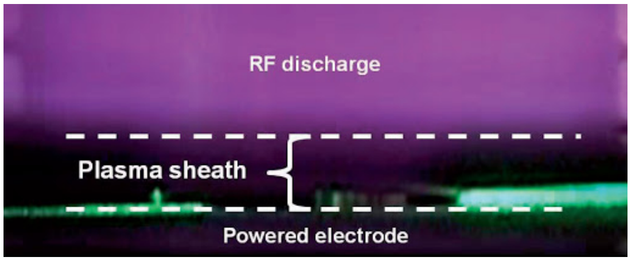


Figure 2: Plasma sheath forming near an electrode (darker region in the picture). [11]

Plasma sheaths are the near-electrode regions where the plasma is depleted in electrons. Electrodes collect the electrons and the plasma charges positively over a distance λ_D, called the Debye length. The presence of the sheath is generally observable through darker regions in the viscinity of an electrode (Fig 2).

In this work, we present an approximate resolution of the two-fluid model for plasma using the discontinuous Galerkin finite element method (DG-FEM). Usually, such model is coupled to a Poisson equation for the electric potential. However, the potential will be supposed to be known here, on the basis of analytical results from [12] and on statistical mechanics [13]. This apparently simple set-up contains the principal difficulties in plasma simulation: the small electron-to-ion mass ratio and the stability of the temporal schemes.

Recent breakthrough in the resolution of the equations in the plasma sheath have been made by Alvarez Laguna *et al.* [14] in the domain of asymptotic preserving schemes. Many numerical methods have also been developed for the resolution of the multi-fluid equations for plasma sheath [15, 16, 17, 18, 19, 20, 21, 22], but few analysis have been performed using DG-FEM. This work can be considered a first approach in the resolution of plasma sheath problems using this method.

The DG-FEM method is applied to a simple model problem. The problem consists of a unidimensional collisionless plasma obtained through argon RF discharge. The plasma is trapped in a cylinder whose wall is at constant potential. The problem can be considered unidimensional thanks to the angular and axial symmetries. The goal of this work is to give the basis of the method and point at difficulties encoutered. Therefore, this work can serve as a basis for future improvements of the method.

A. Laguna *et al.* have solved a similar problem in their work [14]. They managed to obtain very accurate results using an asymptotic-preserving (AP) scheme with a finite volume discretization. The great advantage of such method is that the Debye length does not need to be resolved, saving a lot of computation time when information in the sheath is not required. This work does not try to reproduce this feature, but can be used in the future in the potential combination of DG-FEM discretization with an AP scheme.

This work contains three parts: one for the modeling of the plasma, one for the numerical methods employed and one for the results obtained. In the first part, Boltzmann's equation is presented and the the two-fluid model is derived. The momentum and mass conservation equations for the ions and electrons are derived and coupled to Poisson's equation. Then, the analytical expression of the electric potential is derived. The mathematical structure of the equations is discussed, and the boundary conditions are presented.

In the second part, the unidimensional mesh is shown. The spatial discretization using the DG-FEM on this mesh is derived, followed by the presentation of the temporal schemes. We used two time discretizations: an explicit fourth order Runge-Kutta method and an implicit second order accurate backward Euler method. Both methods have an adaptative time step.

The last part corresponds to the presentation of the obtained results. Two basic cases to assess the discretization and the behaviour of the temporal schemes are discussed: the propagation of a step and the propagation of a plasma perturbation. Finally, the results corresponding to the plasma trapped in the cyclinder with equipotential walls are presented. They are compared to the results of a highly resolved TVD third-order reconstruction and third-order TVD Runge-Kutta scheme [23] with finite volume discretization.

Part I

Plasma Model

Chapter 1

Statistical theory and multi-fluid model

Before diving in the simulation of the plasma flow, the equations describing its motion have to be discussed. The structure of those equations will give information on how they should be treated numerically. First, the Boltzmann's equation, which describes the plasmas statistically, is presented. The first two moments of this equation give a multi-fluid model for the ions and electrons. They are coupled to a Poisson equation in order to close the system. The final system is a general formulation of the equations solved in the numerical simulations. Note that the Poisson equation will not be solved.

1.1 Boltzmann's equation

The Boltzmann's equation for a particle α (which can be an ion or an electron) reads (see *e.g.* [24] and Appendix B.1):

$$\partial_t f_\alpha + \boldsymbol{v}\partial_{\boldsymbol{x}} f_\alpha + \partial_{\boldsymbol{v}} \left(\frac{\boldsymbol{F}_\alpha}{m_\alpha} f_\alpha \right) = C_\alpha + I_\alpha \tag{1.1}$$

where $f_\alpha(\boldsymbol{x}, \boldsymbol{v}, t)$ represents the number of particles possessing velocity $\boldsymbol{v}$ and that have position $\boldsymbol{x}$, $\boldsymbol{F}_\alpha(\boldsymbol{x}, t)$ is the total force exerted on the particle α with velocity $\boldsymbol{v}$ at position $\boldsymbol{x}$ and m_α is the particle mass. $\partial_{\boldsymbol{x}}$ and $\partial_{\boldsymbol{v}}$ represent the divergence operators in the position and velocity spaces respectively. For a particle carrying a charge e_α in an electric field $\boldsymbol{E}$ and in abscence of magnetic field, one has

$$\boldsymbol{F}_\alpha = e_\alpha \boldsymbol{E} \tag{1.2}$$

As $\boldsymbol{F}_\alpha$ and f_α represent smooth averaged quantities over many particles, they do not take into account the contribution of thermal fluctuations. Such phenomena comprise the collisions between the molecules, represented by C_α and the ionization I_α. In the present work, it is assumed that the purely collisional effects are negligible compared to the ionization effects so that the Boltzmann's equation reads

$$\partial_t f_\alpha + \boldsymbol{v}\partial_{\boldsymbol{x}} f_\alpha + \partial_v \left(\frac{e_\alpha \boldsymbol{E}}{m_\alpha} f_\alpha \right) = I_\alpha \tag{1.3}$$

This equation serves as a basis for the multi-fluid model developed in the following section.

1.2 Multi-fluid model

The multi-fluid model is derived on the basis of the Boltzmann's equation Eq. 1.3. At first, the ionization is neglected, then is taken into account after all developements are performed. The multi-fluid considered here is assumed isothermal, meaning that both species have constant temperature. However, the temperature can differ from one species to the other.

Following the steps depicted in [25], the number density of the particle is defined

$$n_\alpha(\boldsymbol{x}) = \int_{\mathbb{R}^3} f_\alpha(\boldsymbol{x}, \boldsymbol{v}, t) d\boldsymbol{v}, \tag{1.4}$$

the mean velocity of the particle

$$\boldsymbol{V}_\alpha(\boldsymbol{x}) = \frac{1}{n_\alpha} \int_{\mathbb{R}^3} \boldsymbol{v} f_\alpha(\boldsymbol{x}, \boldsymbol{v}, t) d\boldsymbol{v}, \tag{1.5}$$

and integrating Eq. 1.3 multiplied by 1 and $m_\alpha \boldsymbol{V}_\alpha$ over $\mathbb{R}^3$ with respect to the velocities, one obtains the equations for density and momentum

$$\begin{aligned} &\partial_t n_\alpha + \partial_{\boldsymbol{x}}(n_\alpha \boldsymbol{V}_\alpha) && = 0 \\ &\partial_t(m_\alpha n_\alpha \boldsymbol{V}_\alpha) + \partial_{\boldsymbol{x}}(m_\alpha n_\alpha \langle \boldsymbol{v}\boldsymbol{v} \rangle) - e_\alpha n_\alpha \boldsymbol{E} && = 0 \end{aligned} \tag{1.6}$$

Knowing that $v_j = V_j + v'_j$ where v'_j is a perturbation velocity whose average is zero, one has

$$\langle v_i v_j \rangle = V_i V_j + \langle v'_i v'_j \rangle \tag{1.7}$$

If the plasma is isotropic,

$$\langle v'_i v'_j \rangle = 0, \quad i \neq j \tag{1.8}$$

and if the only non zero component of the average velocity is in the x direction, then

$$\begin{aligned} \langle v_i v_j \rangle &= 0 \quad i \neq j \\ \langle v_i v_j \rangle &= V_i^2 + \langle {v_i'}^2 \rangle \quad i = j \end{aligned} \tag{1.9}$$

As a consequence, Eq. 1.6 can be written in the form (if only the x component of the momentum is taken into account)

$$\begin{aligned} \partial_t n_\alpha + \partial_x (n_\alpha V_\alpha) &= 0 \\ \partial_t (m_\alpha n_\alpha V_\alpha) + \partial_x (m_\alpha n_\alpha \langle v_x^2 \rangle) - e_\alpha n_\alpha E &= 0 \end{aligned} \tag{1.10}$$

where V_α and E represent the average velocity and the electric field in the x direction respectively. The macroscopic temperature of the particle T_α can be derived from microscopic kinetic relation and the equipartition of energy

$$\frac{m_\alpha \langle {v_x'}^2 \rangle}{2} = \frac{k_B T_\alpha}{2} \tag{1.11}$$

with $v_x' = v_x - V_\alpha$ and k_B the Boltzmann constant. Also, because v_x' is a velocity fluctuation, it is on average zero, so that

$$\begin{aligned} \langle v_x^2 \rangle &= \langle (V_\alpha + v_x')^2 \rangle \\ &= V_\alpha^2 + \langle {v_x'}^2 \rangle \\ &= V_\alpha^2 + \frac{k_B T_\alpha}{m_\alpha} \end{aligned} \tag{1.12}$$

From the results of Eq. 1.12, assuming the mass of the particles remains constant and that the electric potential ϕ is such that $\partial_x \phi = E$, Eq. 1.6 becomes

$$\begin{aligned} \partial_t n_\alpha + \partial_{\boldsymbol{x}} (n_\alpha V_\alpha) &= 0 \\ m_\alpha \partial_t (n_\alpha V_\alpha) + \partial_x (m_\alpha n_\alpha V_\alpha^2 + n_\alpha k_B T_\alpha) &= e_\alpha n_\alpha \partial_x \phi \end{aligned} \tag{1.13}$$

Eq. 1.13 is not sufficient to completely determine the system as there is no equation for ϕ. The system is closed thanks to a Poisson equation for the electric potential

$$\partial_{xx} \phi = \frac{\rho}{\varepsilon_0} \tag{1.14}$$

where

$$\rho = \sum_{\alpha=1}^{N} n_\alpha e_\alpha \tag{1.15}$$

is the electric charge density and N is the number of different particles present in the plasma mixture.

Finally, the ionization rate has to be taken into account. Because it only influences the density number of a particle, the ionization rate is expected to appear in the right hand side of the density equation and have no effect whatsoever in the momentum balance. Eq. 1.13 with ionization taken into account and coupled to the Poisson equation can thus be written in a general way

$$\begin{aligned} &\partial_t n_\alpha + \partial_x (n_\alpha V_\alpha) &&= J_\alpha \\ &m_\alpha \partial_t (n_\alpha V_\alpha) + \partial_x (m_\alpha n_\alpha V_\alpha^2 + n_\alpha k_B T_\alpha) &&= e_\alpha n_\alpha \partial_x \phi \\ &\partial_{xx} \phi &&= \sum_{\alpha=1}^{N} n_\alpha e_\alpha \end{aligned} \tag{1.16}$$

where J_α represents the ionization phenomena.

Eq. 1.16 gives a very general way for dealing with plasmas. In the following Section, it is particularized to a practical case involving two species, leading to a two-fluid model. The value of J_α will be given in this particular case.

Chapter 2

Application to a simple case

Now the theory explained in Chapter 1 will be applied to a simplified case of plasma sheath. This leads to the reduction of Eq. 1.16 to a two-fluid model. The electric potential is assumed to be known and an analytical expression is derived. The boundary conditions in the plasma sheath are also discussed. Then the system is non dimensionalized and some of its important parameters such as the electron-to-ion mass ratio or the non dimensionalized electron Debye length are presented.

2.1 The two-fluid model

The two fluid isothermal model coupled with a Poisson equation is generally used to describe plasma sheaths [26, 27, 28]. Such a model is a particularisation of Eq. 1.16 to two particles: the electrons on one side and the ions on the other side. If the plasma considered is composed of argon, then the ions are charged with e and the electrons with $-e$, where $e = 1.6 \times 10^{-19}$C is the absolute value of the electron charge. The two-fluid model reads

$$\begin{aligned}
&\partial_t n_e + \partial_x(n_e u_e) &&= n_e \nu \\
&\partial_t n_i + \partial_x(n_i u_i) &&= n_e \nu \\
&m_e \partial_t(n_e u_e) + \partial_x(m_e n_e u_e^2 + p_e) &&= n_e e \partial_x \phi \\
&m_i \partial_t(n_i u_i) + \partial_x(m_i n_i u_i^2 + p_i) &&= -n_i e \partial_x \phi \\
&\partial_{xx}^2 \phi &&= \frac{n_e - n_i}{\epsilon_0} e
\end{aligned} \tag{2.1}$$

where the subscripts e and i refer to quantities attached to electrons and ions, n_e and n_i are number densities, u_e and u_i are velocities, m_e and m_i are the masses, $p_e = n_e k_B T_e$ and $p_i = n_i k_B T_i$ are the partial pressure assumed to follow perfect gas law, k_B is the Boltzmann

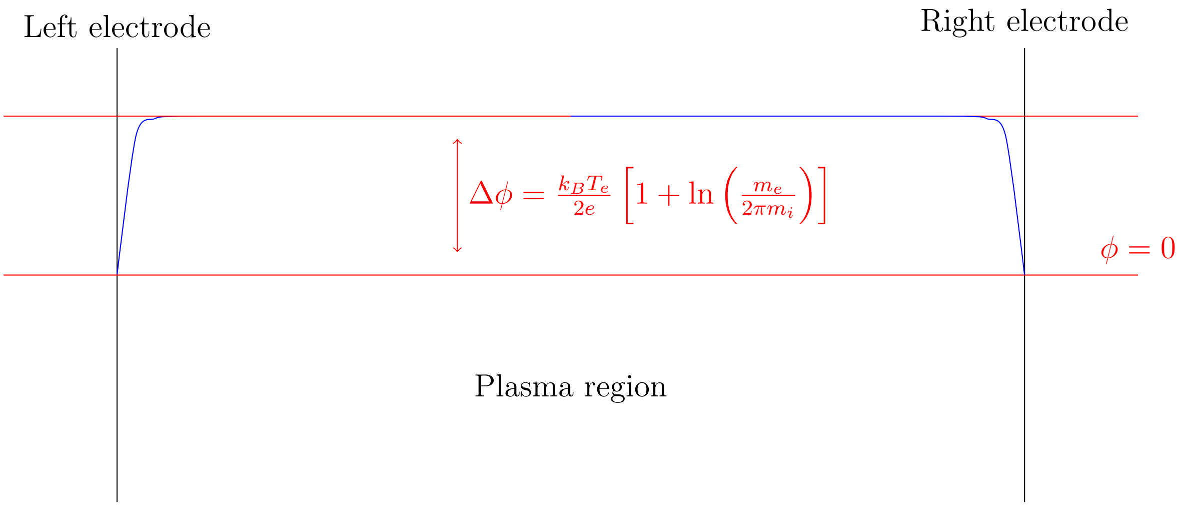

Left electrode
Right electrode
$\Delta\phi = \frac{k_B T_e}{2e}\left[1 + \ln\left(\frac{m_e}{2\pi m_i}\right)\right]$
$\phi = 0$
Plasma region

Figure 2.1: Visual representation of the problem. The plasma is trapped in a cylinder whose walls are an equipotential surface. The continuous line that join the two electrodes is an approximation of the electric potential.

constant, T_e and T_i are the temperatures, ϕ is the electric potential, e is the electric charge and ϵ_0 is the electric permittivity of the vacuum. Note that the ionization is the same for both particles and is $J_\alpha = \nu n_e$. Indeed, when an electron is created, an ion is also created so the two ionization must be the same. The parameter ν is the ionization rate.

In this work, we are mainly interested in the numerical resolution of the first four equations of Eq. 2.1. The last equation is left for future work. In that way, one has to find an analytical expression for the potential ϕ. The presentation of the considered problem along with the boundary conditions will help to determine the form to choose for ϕ.

2.2 Definition of the problem

The problem presented here is a plasma trapped inside a cylinder with equipotential walls of inifinite height. The cylinder radius is unity (in non dimensional units). The problem is unidimensional, with the position x representing the position of a fluid particle on the diameter of the cylinder. A schematical representation of the situation is given Fig. 2.1.

The plasma considered is assumed to have characteristics very close to a plasma obtained through Argon RF discharge. A brief reminder of the characteristics is presented Table 2.1. The purpose of this work is to find a steady state to this situation.

Now that the problem has been properly defined, the boundary conditions have to be discussed and the analytical expression of the electric potential has to be found.

2.3 Electric potential approximation

As pointed in the previous section, the goal of the work is to find a steady state to the situation. In that way, the analytical approximate expression for the potential used must be close to the real potential at steady state. In this section, several simplifying assumptions are made in order to find an analytical expression for the electric potential.

In the near wall region, there is a formation of a plasma sheath. The sheath extends from the electrodes towards the center of the domain over a distance λ, called the Debye length. In this region, the plasma cannot be considered quasi-neutral anymore (see Appendix B.2 for more details about plasma sheath and shielding). Sufficiently far from the electrodes, the plasma can be considered quasi-neutral such that

$$n_{e0} \simeq n_{i0} = 10^{16}\mathrm{m}^{-3} \tag{2.2}$$

where n_{e0} and n_{i0} are the electron and ion densities outside of the sheath region (considered constant). Inside the sheath region, this equality is not true anymore and there are spatial gradients of the electrons and ions number densities n_e and n_i.

The following derivations are performed in the part of space which extends from the left boundary ($x = 0$) to the center of the domain ($x = \frac{L}{2}$). They can be easily extended to the other part of the domain. We first begin by considering that in the sheath, the electron and ion inertia is neglected. This strong hypothesis lies on previous numerical results found by [14]. Fig. 2.2 shows that the electrons momentum in the near-wall region at steady state is almost constant in space. As a consequence, the total time derivative of both momentum are very close to zero and the system is at an approximate dynamic equilibrium at steady state in this region. This is a strong assumption, but it will help to find an approximation for the potential. If inertia is neglected, the momentum equations for both species becomes

ε	Electron to ion mass ratio	1.36×10^{-5}
κ	Ion to electron temperature ratio	0.025
λ	Normalised Debye length	3.5×10^{-3}
ν	Normalised ionization	0.0139
ω^{-1}	Normalised plasma period	1.29×10^{-5}
χ	Normalised square of the electron Debye length	10^{-4}

Table 2.1: Table containing non dimensional numbers for a plasma composed of Argon. [29]

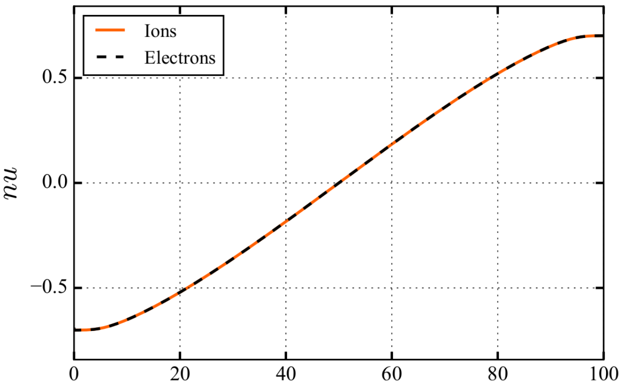


Figure 2.2: Momentum profile for ions and electrons found by Alvalrez Laguna *et al.* [14]. The momentum profile is nearly flat in the near wall region, hinting at an equilibrium condition at steady state in this area.

a balance between the pressure gradient and the effect of the electric field.

$$\begin{aligned} \partial_x p_e &= n_e e \partial_x \phi \\ \partial_x p_i &= -n_i e \partial_x \phi \end{aligned} \tag{2.3}$$

As $p_e = n_e k_B T_e$ and $p_i = n_i k_B T_i$, the integration of the previous equations (only the left wall is considered here) gives

$$n_e(x) = n_{e0} e^{\frac{e\phi(x)}{k_B T_e}} \quad n_i(x) = n_{i0} e^{-\frac{e\phi(x)}{k_B T_i}} \tag{2.4}$$

Because it is assumed that the potentential perturbation is much smaller than the electron thermal energy $e\phi \ll k_B T_e$, the relation $e\phi \ll k_B T_i$ is also true because $T_i/T_e < 1$ (*cfr.* Table 2.1). Eq. 2.4 can be expanded as (with $n_{e0} \simeq n_{i0}$)

$$n_e(x) \simeq n_{e0}\left(1 + \frac{e\phi(x)}{k_B T_e}\right) \quad n_i(x) \simeq n_{e0}\left(1 - \frac{e\phi(x)}{k_B T_i}\right) \tag{2.5}$$

Because the electric potential has to satisy the Poisson equation given in Eq. 2.1, one has from Eq. 2.5

$$\begin{aligned} \partial_{xx}\phi(x) &= \frac{n_{e0}e^2}{\epsilon_0 k_B}\left(\frac{1}{T_i} + \frac{1}{T_e}\right)\phi(x) \\ &= \frac{\phi(x)}{\lambda_D^2} \end{aligned} \tag{2.6}$$

where

$$\lambda_D^2 = \frac{\epsilon_0 k_B}{n_{e0}e^2}\left(\frac{T_i T_e}{T_i + T_e}\right) \tag{2.7}$$

is the square of the total Debye length. This quantity can be rewritten in the form

$$\frac{1}{\lambda_D^2} = \frac{n_{e,0}e^2}{\epsilon_0 k_B T_e} + \frac{n_{e,0}e^2}{\epsilon_0 k_B T_i} = \frac{1}{\lambda_{De}^2} + \frac{1}{\lambda_{Di}^2} \tag{2.8}$$

where λ_{De} and λ_{Di} represent respectively the electronic and ionic Debye lengths. The solution of such an equation is trivially given by

$$\phi(x) = Ae^{\frac{-x}{\lambda_D}} + Be^{\frac{x}{\lambda_D}} \tag{2.9}$$

where A and B are integration constants. The second term has to be dropped because the potential cannot grow indefinitely with x (note that the other term is dropped in the case of the right boundary). It has been shown [12] that the potential difference between one electrode and the potential at the plasma bulk is given by

$$\phi_0 = -\frac{k_B T_e}{2e}\left[1 + \ln\left(\frac{m_i}{2\pi m_e}\right)\right] \tag{2.10}$$

so that an approximation of the potential over near the electrode is

$$\phi(x) = -\frac{k_B T_e}{2e}\left[1 + \ln\left(\frac{m_i}{2\pi m_e}\right)\right]\left(e^{\frac{-x}{\lambda_D}} - 1\right) \tag{2.11}$$

Note that the potential at the wall is taken for reference. After performing the same analysis for the right part of the domain, an analytical expression for the electric potential over the whole domain is given by

$$\phi(x) = \begin{cases} \phi_0\left(e^{-\frac{x}{\lambda_D}} - 1\right) & x < L/2 \\ \phi_0\left(e^{\frac{x-L}{\lambda_D}} - 1\right) & x > L/2 \end{cases} \tag{2.12}$$

with L the domain length. A representation of this potential is given Fig. 2.1.

The potential found is normally valid only in the near-wall region and at steady state. Alvarez Laguna *et al.* [14] have also found the potential profile. It is compared to the approximate potential we derived (Fig. 2.3). By choosing $\lambda_D = 0.035$ in the approximate solution, there is an almost a perfect match between the numerical and the approximate potential inside the sheath. The two potential are very close in the center of the domain, while there is a region ($x \in [0.1; 0.4] \bigcup [0.6; 0.9]$) where there is a visible discrepancy.

Since the approximate potential found is valid only at steady state, the solution strategy will be to initialize the flow using the numerical solution found by Alvarez Lagunal *et al.* and

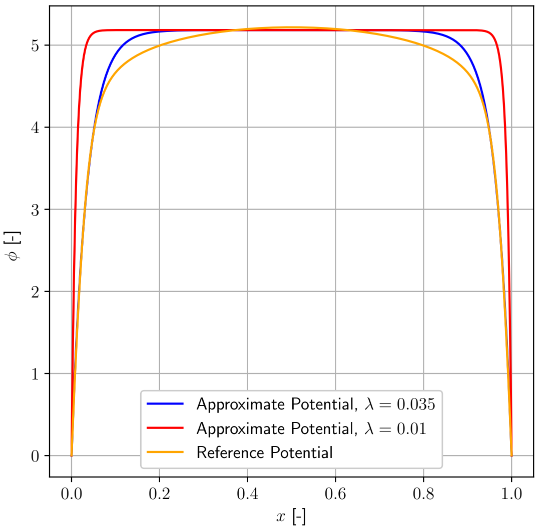


Figure 2.3: Approximate potential compared to the numerical electric potential found by Alvalrez Laguna *et al.* [14]. With a modification of the Debye length from $\lambda = 0.01$ to $\lambda = 0.035$, the approximate expression found is very close to the numerical one, except in the presheath.

to let the simulation evolve towards the steady state associated to the approximate potential.

2.4 Plasma boundaries

The potential at the boundaries is straightforward. As mentioned in the previous Section, the potential is referenced at the wall, so that

$$\phi(0) = \phi(L) = 0 \tag{2.13}$$

Concerning the ions and electrons, it requires more work to determine which boundary to apply. At steady state, a sheath is formed near the electrodes. This sheath can be described by the famous Child-Langmuir [30, 31] law. It is characterised by a depletion of electrons and represents a region where the plasma is no longer quasi-neutral. The different regions of the plasma near the right electrode are represented schematically Fig. 2.4 along with a very schematical representation of the densities.

Ions must satisfy what is called the Bohm criterion. It states that the ions have to be supersonic when entering from the presheath into the sheath. For a derivation of this criterion, see *e.g.* [28, 13]. The distribution function of electrons is Maxwellian and it is

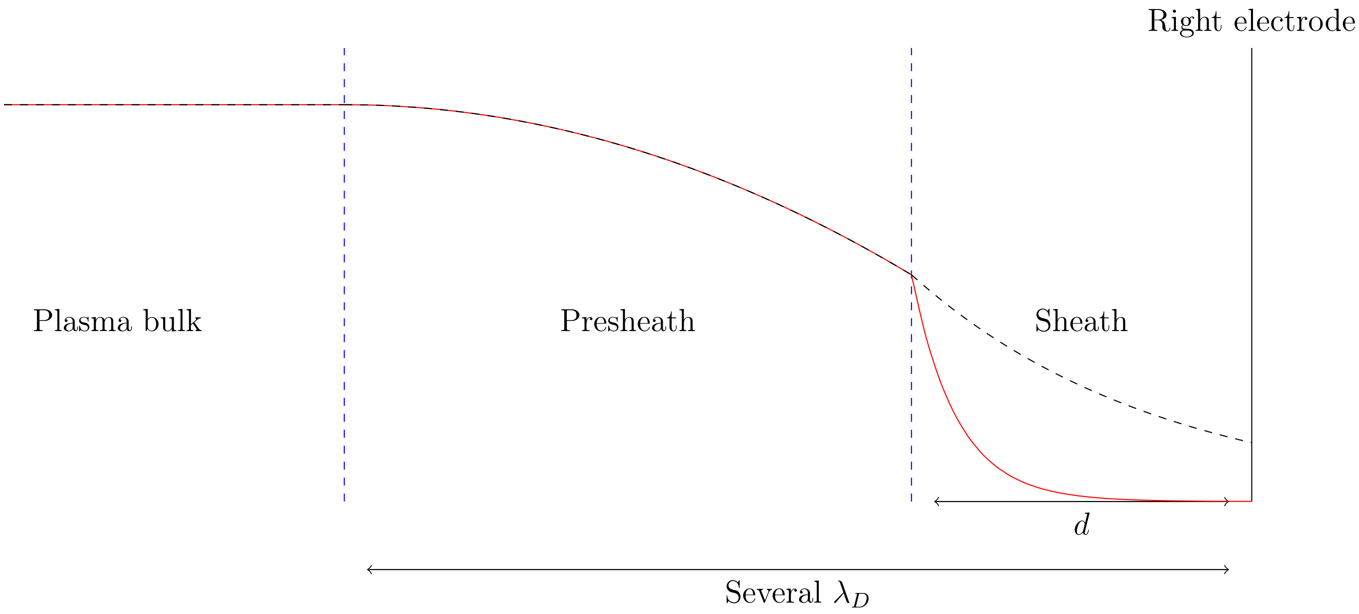


Figure 2.4: Schematic representation of the particle densities (plain line for electron and dashed line for ions) in the near wall region (the right one in this case). There are three distinct regions: the bulk, farthest from the wall, the presheath, middle region where the quasi-neutrality still applies, and the sheath where the quasi-neutrality breaks. The presheath and sheath extend over several Debye length λ_D.

assumed that all electrons that touch the electrode are absorbed. We also assume that there is no secondary electron emission. From there, it can be shown that the electron flux through each electrode is given by [29]

$$n_e u_e|_{left} = -n_e u_e|_{right} = -n_e \sqrt{\frac{k_B T_e}{2\pi m_e}} \tag{2.14}$$

Finally, the walls are supposedly at a uniform potential. It means that no current flows through the walls. As a consequence, the electron and ion flux must be the same at the walls to have a net current worth 0. As a consequence, Eq. 2.14 can be generalized

$$n_e u_e|_{left} = n_i u_i|_{left} = -n_e u_e|_{right} = -n_i u_i|_{right} = -n_e \sqrt{\frac{k_B T_e}{2\pi m_e}} \tag{2.15}$$

2.5 Non-dimensional form of the system

The non dimensional form of the equations helps accessing the relative importance of each term. Let us define the following reference quantites : n_0, a reference number density for both electrons and ions, T_e and T_i, the electron and ion temperatures, L_0, a reference length. Combination of these reference quantities produce new reference variables : the reference velocity is the Bohm velocity $u_0 = \sqrt{k_B T_e/m_i}$, $\phi_0 = k_B T_e/e$ is the reference potential and $t_0 = L_0/u_0$ is the reference time. Those quantities can be used to normalize Eq. 2.1:

$$\begin{aligned}
&\partial_{\tilde{t}}\tilde{n}_e + \partial_{\tilde{x}}(\tilde{n}_e\tilde{u}_e) && = \tilde{n}_e\tilde{\nu} \\
&\partial_{\tilde{t}}\tilde{n}_i + \partial_{\tilde{x}}(\tilde{n}_i\tilde{u}_i) && = \tilde{n}_e\tilde{\nu} \\
&\partial_{\tilde{t}}(\tilde{n}_e\tilde{u}_e) + \partial_{\tilde{x}}(\tilde{n}_e u_e^2 + \tilde{n}_e\varepsilon^{-1}) && = \varepsilon^{-1}\tilde{n}_e\partial_{\tilde{x}}\tilde{\phi} \\
&\partial_{\tilde{t}}(\tilde{n}_i\tilde{u}_i) + \partial_{\tilde{x}}(\tilde{n}_i\tilde{u}_i^2 + \tilde{n}_i\kappa) && = -n_i\partial_{\tilde{x}}\tilde{\phi} \\
&\partial_{\tilde{x}\tilde{x}}\tilde{\phi} && = \chi^{-1}(\tilde{n}_e - \tilde{n}_i)
\end{aligned} \tag{2.16}$$

The quantities are noted $\tilde{f}$ when they are non dimensional. To avoid clutter, the "tilde" is dropped in the following chapters, but kept here. Note the appearance of three parameters:

1. $\varepsilon = m_e/m_i \sim 10^{-5}$, the electron-to-ion mass ratio,

2. $\kappa = T_i/T_e \sim 2.5 \times 10^{-2}$ the ion-to-electron temperature ratio,

3. $\chi = \frac{\varepsilon_0 k_B T_e}{n_{e0} e^2 L_0^2} \sim 10^{-4}$ the square of the non dimensional electronic Debye length.

Those non dimensional numbers have an impact on numerics. ε and κ play a role in the propagation speed of the system, while $\chi = \lambda_{De}^2$, with λ_{De} the electronic Debye length, as pointed before, has an impact on the non dimensional potential expression, given by

$$\tilde{\phi}(x) = \begin{cases} \tilde{\phi}_0 \left(e^{-\frac{x}{\lambda_D}} - 1 \right) & x < L/2 \\ \tilde{\phi}_0 \left(e^{\frac{x-L}{\lambda_D}} - 1 \right) & x > L/2 \end{cases} \tag{2.17}$$

with

$$\tilde{\phi}_0 = -\frac{1}{2}\left[1 + \ln\left(\frac{m_i}{2\pi m_e} \right) \right] \tag{2.18}$$

Note also that the electron flux passing through the boundaries is given in normalized form by

$$n_e u_e\big|_{left} = n_i u_i\big|_{left} = -n_e u_e\big|_{right} = -n_i u_i\big|_{right} = -\frac{\tilde{n}_e}{\sqrt{2\pi\varepsilon}} \tag{2.19}$$

Now that the problem has been correctly defined and an analytical expression for the potential has been found, it is time to study the mathematical structure of Eq. 2.1. It gives information on how to discretize the system and solve it numerically. By combining the information from physics (this Chapter) and mathematics (the following Chapter), it will be possible to correctly discretize the equations numerically.

Chapter 3

Structure of the equations

Now that the physics have been discussed, the mathematical structure has to be developed. First, the eigenvalues and eigenvectors of the hyperbolic system composed of the four first equations of Eq. 2.16 are computed. Thanks to those, the characteristic velocities of the system are known and are used to compute the Courant-Friederichs-Lewy (CFL) condition. Finally, the characteristics entering and leaving the domain are discussed, since they have an impact on the boundary condition implementation.

3.1 Hyperbolic and parabolic systems

The normalized system describing the physics of the problem is given by

$$\begin{aligned}
&\partial_t n_e + \partial_x(n_e u_e) &&= n_e \nu \\
&\partial_t n_i + \partial_x(n_i u_i) &&= n_e \nu \\
&\partial_t(n_e u_e) + \partial_x(n_e u_e^2 + n_e \varepsilon^{-1}) &&= \varepsilon^{-1} n_e \partial_x \phi \\
&\partial_t(n_i u_i) + \partial_x(n_i u_i^2 + n_i \kappa) &&= -n_i \partial_x \phi \\
&\partial_{xx}\phi &&= \chi^{-1}(n_e - n_i)
\end{aligned} \tag{2.16}$$

Eq. 2.16 can be split into two subsystems: a hyperbolic one which comprises the four first equations of Eq. 2.16, and an elliptic one which consists of the Poisson equation. In this work, because it is assumed that ϕ is given, we are only interested in the hyperbolic system, and the Poisson equation will be ignored from now on.

Eq. 2.16 is also a conservative equation, and it can be written as

$$\partial_t \boldsymbol{u} + \partial_x \boldsymbol{F} = \boldsymbol{S} \tag{3.1}$$

where

$$\begin{aligned}
\boldsymbol{u} &= \begin{pmatrix} n_e & n_i & n_e u_e & n_i u_i \end{pmatrix} \\
\boldsymbol{F} &= \begin{pmatrix} n_e u_e & n_i u_i & n_e u_e^2 + n_e \varepsilon^{-1} & n_i u_i^2 + n_i \kappa \end{pmatrix} \\
\boldsymbol{S} &= \begin{pmatrix} n_e \nu & n_e \nu & \varepsilon^{-1} n_e \partial_x \phi & -n_i \partial_x \phi \end{pmatrix}
\end{aligned} \tag{3.2}$$

where $\boldsymbol{u}$ is the unknowns vector, $\boldsymbol{F}$ is the fluxes vector and $\boldsymbol{S}$ is the source vector.

3.2 Eigenvalues and eigenvectors

Eq. 3.1 can also be written as

$$\partial_t \boldsymbol{u} + \boldsymbol{A} \partial_x \boldsymbol{u} = \boldsymbol{S} \tag{3.3}$$

with

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \varepsilon^{-1} - u_e^2 & 0 & 2u_e & 0 \\ 0 & \kappa - u_i^2 & 0 & 2u_i \end{pmatrix} \tag{3.4}$$

The eigenvalues of $\boldsymbol{A}$, which represent the propagation speed of the information in the system, are

$$\begin{aligned}
\lambda_0 &= u_e - \sqrt{\varepsilon^{-1}} \\
\lambda_1 &= u_e + \sqrt{\varepsilon^{-1}} \\
\lambda_2 &= u_i - \sqrt{\kappa} \\
\lambda_3 &= u_i + \sqrt{\kappa}
\end{aligned} \tag{3.5}$$

while its eigen vectors are given by

$$\begin{aligned}
\boldsymbol{v}_0 &= \begin{pmatrix} 1 & 0 & \lambda_0 & 0 \end{pmatrix} \\
\boldsymbol{v}_1 &= \begin{pmatrix} 1 & 0 & \lambda_1 & 0 \end{pmatrix} \\
\boldsymbol{v}_2 &= \begin{pmatrix} 0 & 1 & 0 & \lambda_2 \end{pmatrix} \\
\boldsymbol{v}_3 &= \begin{pmatrix} 0 & 1 & 0 & \lambda_3 \end{pmatrix}
\end{aligned} \tag{3.6}$$

We can now understand the main numerical difficulty encountered while simulating the plasma sheath : there is a strong velocity disparity between the electrons and ions. For instance, if argon is considered, one has

$$\sqrt{\varepsilon^{-1}} \simeq 270 \ll \sqrt{\kappa} \simeq 0.15 \tag{3.7}$$

This means that the numerical scheme must be able to capture two phenomena happening at very different speed. This, coupled with the steep gradient in the region of length close to the Debye length near the electrodes, lead to a very stiff problem. The next section shows those difficulties in an example.

3.3 Physics and numerics

This section presents the impact of the physics on numerics. The plasma considered here is composed of argon. An order of magnitude of the non dimensional numbers that have an impact on numerics is given Table 2.1.

κ and ε have an impact on the propagation velocity (Eq. 3.5). Since ε is a very small parameter, the propagation of information associated to electrons is very high ($\sim 3 \times 10^2$). As a consequence, in order to keep a CFL number sufficiently low while having a good spatial discretization, the time step must be small. The CFL number is (if Δt and Δx are respectively the time and spatial steps respectively)

$$CFL = \frac{\Delta t |\lambda_{max}|}{\Delta x} < 1 \tag{3.8}$$

with $|\lambda_{max}| = \max(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$. Since $\varepsilon \ll \kappa$, one has in general $|\lambda_{max}| = \max(\lambda_0, \lambda_1)$. It has also been proven [32] that the stability of the system depends on the ability to resolve the electron plasma period. The plasma period is the characteristic time for each species to go back to its equilibrium position after being perturbed. Since the ions are usually much heavier than the electrons, their period is higher and the electrons give the more stringent condition on the time step. The caracteristic electron period is the time it requires to go back to its initial position, *i.e.* the time it requires to travel the distance λ_{De}, the electronic Debye length. Knowing that its dimensionless thermal caracteristic velocity is given by

$$v_{th,e} = \varepsilon^{-1/2} \tag{3.9}$$

and that the dimensionless electronic Debye length is given by (see Appendix B.2)

$$\lambda_{De}^2 = \frac{\epsilon_0 k_B T_e}{e^2 n_{e0} L_0^2}, \tag{3.10}$$

the plasma fraquency associated to electrons is given by

$$\omega_e = \frac{v_{th,e}}{\lambda_{De}} = \left(\frac{\epsilon_0 k_B T_e}{e^2 n_{e0} L_0^2 \varepsilon} \right)^{1/2} \tag{3.11}$$

The electron period is then correctly captured if

$$\Delta t \omega_e < 1 \tag{3.12}$$

Finally, the time step should also be able to capture the characteristic ionization time. In other words,

$$\Delta t \nu < 1 \tag{3.13}$$

All the previous conditions can be summarized as follow:

$$\max(CFL, \Delta t \omega_e, \Delta t \nu) < 1 \tag{3.14}$$

In the present case, the condition is automatically satisfied if Δx resolves the Debye length and the CFL condition is fullfilled. In that way, the electron-to-ion mass ratio is the most stringent parameter and leads to a very stiff system. The maximum time step to obtain a stable system is

$$\Delta t_{max} \simeq 10^{-5} \tag{3.15}$$

Because of the high restriction on the time step, an implicit time marching method is usually preferred.

3.4 Boundary conditions and characteristics at steady state

As explained in Section 2.4, ions are supersonic in the sheath. They are also expected to be supersonic when they collide with the wall. Consequently, the characteristics associated to λ_2 and λ_3 at the boundaries are all going outside of the domain. It means that the plasma flow takes all information from inside the domain and convect it outside.

In contrast, the electron velocity is of order $\frac{1}{\sqrt{2\pi\varepsilon}} \simeq 108$ which is below the electron sound velocity $\sqrt{\varepsilon^{-1}} \simeq 270$. In that way, at each boundary, there is always one characteristic which enters the domain while the other leaves the domain. In other words, the system takes information from outside the domain and convect it inside.

This analysis will allow to correctly set the boundary conditions of the numerical simulation. In this case, since only two characteristics enter the system, only two boundary condition, one per boundary, have to be set to define a steady state. During the simulation, the system might not be in the conditions explained before. Especially, the electrons can become supersonic and the ions subsonic. In that case, the boundary conditions have to be modified in consequence. The detail of the boundary condition can be found Section 5.3.

Knowing that the numerical flux used for the discretization will be a true upwind flux, the boundary condition does not matter outside the domain when a species is supersonic. However, when the species is subsonic, the boundary has to be imposed. The strategy is then to over-specify the boundary conditions : when the numerical flux requires the information, it selects it outside the domain and ignores it when not required.

Part II

Numerical Methods

Chapter 4

Mesh and unknowns

Before diving in the discretization of the equations, the mesh has to be defined. Because the problem is unidimensional, the mesh is going to be simple. However, as previously mentioned, steep gradients are expected in the near wall regions, so the presented mesh is refined at the boundaries of the domain. Finally, the mapping of the unknowns is presented.

4.1 Spatial discretization

Let us consider the mesh refinement symmetric about the center of the domain, such as the one given Fig. 4.1. In that fashion, the following developments are done only for the left half of the domain. It can be easily adapted to the right side of the domain. ℓ_1 is the length of the closest element near the wall, E is the total number of elements and L is the total length of the domain. Let us define $\gamma > 0$ the refinement coefficient, such that

$$\ell_{i+1} = \gamma \ell_i \tag{4.1}$$

with ℓ_i the length of the i^{th} element of the domain. One can easily prove that

$$\ell_i = \gamma^{i-1} \ell_1 \tag{4.2}$$

The mesh can then be constructed by knowledge of ℓ_1 and the refinement coefficient γ. Note that for $\gamma = 1$, the mesh is uniform, $\gamma < 1$ the mesh is more refined in the center of the domain and for $\gamma > 1$ it is more refined in the near wall region.

Two cases have to be considered, according to the parity of the number of elements over the domain:

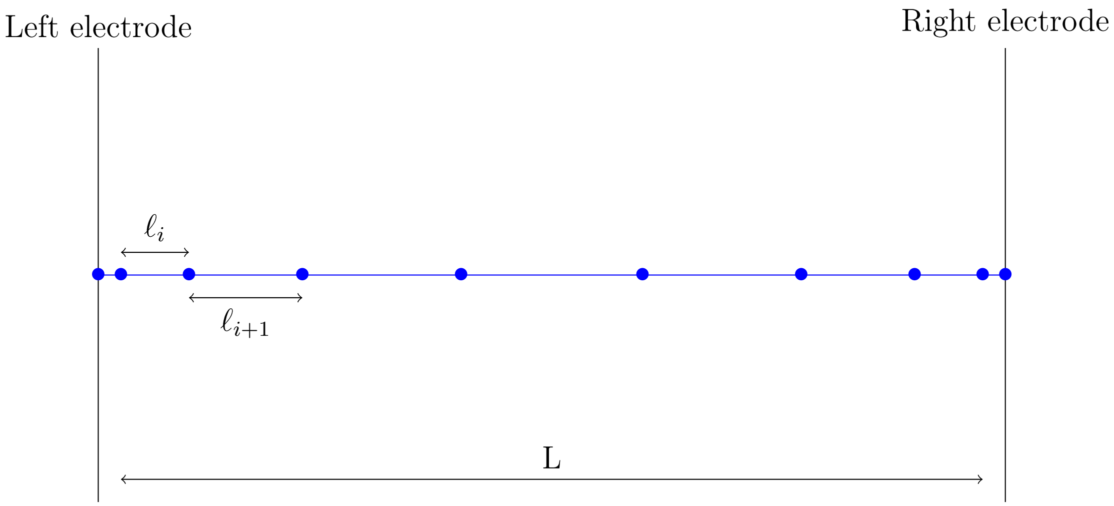


Figure 4.1: Representation of a coarse mesh that is refined near the boundaries.

Case 1: E is even In that case,

$$\sum_{i=1}^{E/2} \ell_i = \sum_{i=1}^{E/2} \gamma^{i-1} \ell_1 = L/2 \tag{4.3}$$

As a result, ℓ_1 can be retrieved by the simple formula

$$\ell_1 = \frac{L}{2\displaystyle\sum_{i=1}^{E/2} \gamma^{i-1}} \tag{4.4}$$

Case 2: N is odd This case is slightly different from the former because there is a central element.

$$\sum_{i=1}^{(E-1)/2} \ell_i + \frac{\ell_{(E+1)/2}}{2} = \sum_{i=1}^{(E-1)/2} \gamma^{i-1} \ell_1 + \frac{\gamma^{(E-1)/2}}{2} \ell_1 = L/2 \tag{4.5}$$

As a consequence,

$$\ell_1 = \frac{L}{2\left[\displaystyle\sum_{i=1}^{(E-1)/2} \gamma^{i-1} + \frac{\gamma^{(E-1)/2}}{2}\right]} \tag{4.6}$$

If $\gamma = 1$, then

$$\sum_{i=1}^{\beta} \gamma^{i-1} = \beta \tag{4.7}$$

while if $\gamma \neq 1$, then

$$\sum_{i=1}^{\beta} \gamma^{i-1} = \frac{1-\gamma^{\beta}}{1-\gamma} \tag{4.8}$$

Table 4.1 summarises all possible situations and their associated value of ℓ_1.

E even		E odd	
$\gamma \neq 1$	$\gamma = 1$	$\gamma \neq 1$	$\gamma = 1$
$\dfrac{L(1-\gamma)}{2(1-\gamma^{E/2})}$	$\dfrac{L}{E}$	$\dfrac{L}{2\left[\frac{1-\gamma^{(E-1)/2}}{1-\gamma}+\frac{\gamma^{(E-1)/2}}{2}\right]}$	$\dfrac{L}{E}$

Table 4.1: Values of ℓ_1 for all possible cases of refinement.

The geometrical length J_i of each elements is then given as a function of γ by

$$J_i = \begin{cases} \dfrac{\gamma^{i-1}L}{2\displaystyle\sum_{i=1}^{E/2}\gamma^{i-1}} & E \text{ even} \quad i = 1, 2, ...E/2 \\ \dfrac{\gamma^{i-1}L}{2\left[\displaystyle\sum_{i=1}^{(E-1)/2}\gamma^{i-1} + \dfrac{\gamma^{(E-1)/2}}{2}\right]} & E \text{ odd} \quad i = 1, 2, ...(E+1)/2 \end{cases} \tag{4.9}$$

Note also that the length can be more easily and generally expressed as a function of the coordinates of the extreme points of the elements

$$J_i = x_{i,right} - x_{i,left} \tag{4.10}$$

where $x_{i,right}$ and $x_{i,right}$ are the coordinates of the point at the right and left boundary of the element respectively. Note that this length is also the geometrical jacobian of the element.

The refinement parameter γ as a function of the number of desired elements $\mathcal{E}$ over the segment $[0;\lambda]$ and the total number of elements E, with $\lambda \leq L/2$ is computed here. In that case, one has

$$\sum_{i=1}^{\mathcal{E}} \ell_i = \sum_{i=1}^{\mathcal{E}} \gamma^{i-1}\ell_1 = \lambda \tag{4.11}$$

If $\gamma \neq 1$,

$$\frac{1-\gamma^{\mathcal{E}}}{1-\gamma} = \frac{\lambda}{\ell_1} \tag{4.12}$$

As a consequence, if *at least* $\mathcal{E}$ elements are expected in $[0;\lambda]$, then γ must be chosen such that

$$\frac{1-\gamma^{\mathcal{E}}}{1-\gamma} \leq \frac{\lambda}{\ell_1} \tag{4.13}$$

In the case of uniform mesh ($\gamma = 1$), the condition translates the fact that each element must be smaller than $1/(\lambda\mathcal{E})$, *i.e.*

$$\ell_1 \leq \frac{1}{\lambda\mathcal{E}} \tag{4.14}$$

4.2 Numbering of the unknowns

Because the system will be solved using the discontinuous Galerkin finite element method (DG-FEM, which is extensively detailed in the following Chapter), the numbering of the unknwons has been performed with this spatial discretization in mind. Such numbering takes into account the fact that each quantity is doubled at the frontier of each element.

There are 4 unknowns to the system: the two densities, and the two momentum. To be clear, let us first consider only the electron density n_e, with a mesh composed of a single element of arbitrary order. The values of n_e are labelled u_j^0, with j corresponding to the j^{th} node of the mesh. The nodes are numbered from the left electrode to the right electrode. A representation of the numbering is given Fig. 4.2.

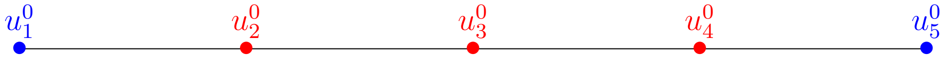


Figure 4.2: Illustration of an element. It consists of two points at the extremities with inner nodes equally spaced. The number of inner nodes depends on the order of the element and the basis function chosen for the discretization.

If the considered mesh is now composed of two elements, the numbering is continued in the same way. However, the interface nodes of each elements (except for the ones corresponding to the boundaries) are labelled twice: one time in each element. A graphical representation is given Fig. 4.3.

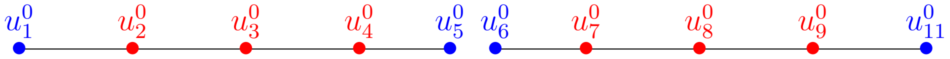


Figure 4.3: Illustration of a series of elements. The numerotation insinde an element is very similar to the single element case. However, the extremities nodes are labelled twice.

The same procedure can be repeated for any number of elements, and for each unknown. Until now, we have considered two indices: one for the node, the other for the unknown.

However, to simplify the numerical implementation, it is always easier to have only one index. In that fashion, if N is the total number of nodes on the mesh, then we decide to number from u_1 to u_N the electron density, u_{N+1} to u_{2N} the ion density, u_{2N+1} to u_{3N} the electron momentum and from u_{3N+1} to u_{4N} the ion momentum, with N the total number of nodes in the mesh.

With these geometrical considerations and the numerotation, we can now move to the numerical method itself.

Chapter 5

Discontinuous Galerkin Finite Element Method

In this Chapter, the discontinuous Galerkin finite element method (DG-FEM) is presented and applied to the plama system of equations. This method can be considered a finite element method allowing for discontinuities in the discrete space. Each element communicates information to the other through numerical fluxes. It can also be viewed as a finite volume method with its solution represented by functions that are not necessarily constant.

Such method has a vast domain of application [33] and has been in the scientific landscape since 1973 [34]. However, it started being used more frequently about 20 years ago [35]. Working with DG-FEM has several advantages : the discontinuity of the discretization space allows to have very compact stencils. It also gives a lot of flexibility and is then very attractive for multi-physics and/or multi-domain simulations. It also allows to incorporate conservation principles in the method by choosing appropriate numerical fluxes.

The Chapter is organized as follow. First, the method is presented in a general way for any type of conservative system. Then, the discretization is applied to the current problem. Then, the numerical flux is presented. Finally, the implementation of the boundary conditions is given.

5.1 Application to a conservative system

This Section presents the DG-FEM applied to a very general conservative system of equations. Because the final application is unidimensional, the following developments are perfomed in one dimension.

Let us consider the conservative general system

$$\partial_t \boldsymbol{u}(x,t) + \partial_x \boldsymbol{F}(\boldsymbol{u},x,t) = \boldsymbol{S}(\boldsymbol{u},x,t), \quad x \in \Omega, t \in \mathbb{R}^+ \tag{5.1}$$

with Ω the unidimensional domain, x the position on Ω and t the temporal variable. $\boldsymbol{u}$ is the vector of unknwons, $\boldsymbol{F}$ is the flux vector and $\boldsymbol{S}$ represents the source terms. The weak form of Eq. 5.1 can be obtained by mutiplying both members by $\Psi(x)$, $\Psi \in C_1(\Omega)$, and integrating over the whole domain Ω. After integration by parts, the weak form is (to avoid clutter, the arguments of the functions have not been recalled)

$$\int_\Omega (\partial_t \boldsymbol{u} - \boldsymbol{S})\Psi dV - \int_\Omega \boldsymbol{F}\partial_x \Psi dV + \int_{\partial\Omega} \boldsymbol{F}\cdot\boldsymbol{n}\Psi dS = 0 \tag{5.2}$$

where $\boldsymbol{n}$ is the outward pointing normal to $\partial\Omega$.

Let us now consider that Ω is partitioned in E non-overlapping elements. Mathematically, it means that

$$\Omega = \bigcup_{k=1}^{E} \Omega_k, \quad \Omega_i \cap \Omega_j = \emptyset, \forall i \neq j \tag{5.3}$$

With each Ω_k corresponding to an element. Note that this partition implies that

$$\partial\Omega = \bigcup_f I_f \tag{5.4}$$

where $\partial\Omega$ is the frontier of Ω and I_f reprensents one of the interfaces bewteen any two elements or between a fontier element and the boundary. In the following, $\partial\Omega_k$ will represent the frontier of Ω_k. Let us now consider the set $C^0_\infty(\Omega_k)$ of functions which are

1. Infinitely continuously differentiable on Ω_k.

2. Whose support is Ω_k.

$C^0_\infty(\Omega_k)$ is called the set of test functions over Ω_k, with k chosen arbitrarily. Let φ_i be a basis function of $C^0_\infty(\Omega_k)$. Then, $\varphi_i \in C_1(\Omega)$ and the following equality is valid for any φ_i

$$\int_\Omega (\partial_t \boldsymbol{u} - \boldsymbol{S})\varphi_i dV - \int_\Omega \boldsymbol{F}\partial_x \varphi_i dV + \int_{\partial\Omega} \boldsymbol{F}\cdot\boldsymbol{n}\varphi_i dS = 0 \tag{5.5}$$

The basis test functions chosen here are the Lagrange polynomials $\varphi_j(x)$ that interpolate nodes placed equidistantly on each element. Such subset of functions is called $\mathcal{L}^0_\infty(\Omega_k)$ and $\mathcal{L}^0_\infty(\Omega_k) \subset C^0_\infty(\Omega_k)$. The general expression of such function over a given element is is given

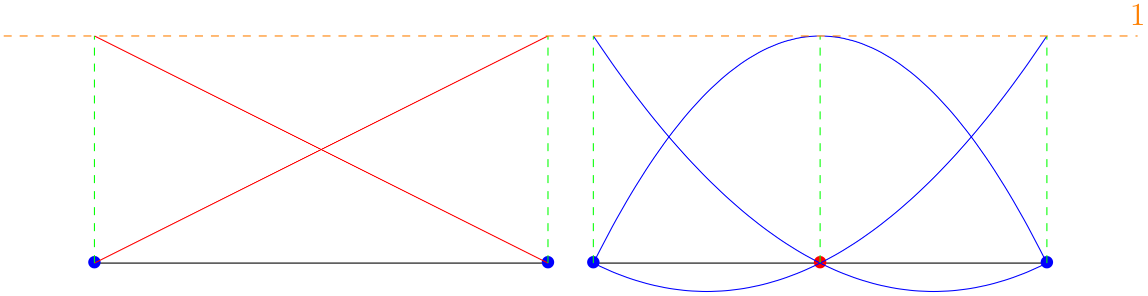


Figure 5.1: Illustration of the test functions of order 1 (left) and 2 (right). The properties of the shape function are immediate: they are 0 at all nodes except for one where they worth 1.

by [37]

$$\varphi_j(x) = \begin{cases} \displaystyle\prod_{i=a, i\neq j}^{a+\mathcal{N}} \frac{x - x_i}{x_j - x_i} & \text{if } x \text{ is inside the element} \\ 0 & \text{if } x \text{ is oustide the element} \end{cases} \tag{5.6}$$

where x_i is the coordinate of the i^{th} node of an element, a is the index of the leftmost node of the element, $\mathcal{N}$ is the number of nodes that the element of order $\mathcal{N} - 1$ possess. Note that Eq. 5.6 does not require the node to be equidistant. An illutration of the shape functions of order 1 and 2 is given Fig. 5.1. Those functions have the following properties:

1. $\varphi_j(x_i) = \delta_{ij}$, with δ_{ij} the Kroenecker delta.

2. φ_j is linearly independent of φ_i, $\forall j \neq i$.

The first property is important in the representation of constant fields. The second one states that the φ_i's form a basis of $\mathcal{L}^0_\infty(\Omega_k)$. The Galerkin approximation [36] takes advantage of this property : it projects $\boldsymbol{u}$, $\boldsymbol{F}$ and $\boldsymbol{S}$ on the basis formed by the φ_i's. This means that the trial function space, the space on which the quantities are projected, is the same as the test function space. Those projections will make a more or less good approximation of the solution, depending on the order of the problem and the order of the φ_i's chosen. After projection, the approximate expressions are

$$\begin{aligned} \boldsymbol{u}_h(x) &\equiv \sum_{j=1}^{N} \boldsymbol{u}(x_j)\varphi_j(x) \\ \boldsymbol{F}_h(x) &\equiv \sum_{j=1}^{N} \boldsymbol{F}(x_j)\varphi_j(x) \\ \boldsymbol{S}_h(x) &\equiv \sum_{j=1}^{N} \boldsymbol{S}(x_j)\varphi_j(x) \end{aligned} \tag{5.7}$$

The subscript h is chosen to emphasize the difference between the approximated and the true solutions. It will be omitted in the following to avoid clutter. Injecting Eq. 5.7 in Eq. 5.5 gives (Einstein's summation convention has been adopted)

$$\int_{\Omega} \varphi_j \varphi_i dV (\partial_t \boldsymbol{u}_j - \boldsymbol{S}_j) - \int_{\Omega} \varphi_j \partial_x \varphi_i dV \boldsymbol{F}_j + \int_{\partial\Omega} \boldsymbol{F} \cdot \boldsymbol{n} \varphi_i dS = 0 \tag{5.8}$$

$\boldsymbol{u}_j$, $\boldsymbol{F}_j$ and $\boldsymbol{S}_j$ are notations for $\boldsymbol{u}(x_j)$, $\boldsymbol{F}(x_j)$ and $\boldsymbol{S}(x_j)$ respectively. Let us define the mass $\mathbf{M}$ and stiffness $\mathbf{K}$ matrices as follow:

$$\begin{aligned} M_{ij} &\equiv \int_{\Omega} \varphi_i \varphi_j dV \\ K_{ij} &\equiv \int_{\Omega} \varphi_i \partial_x \varphi_j dV \end{aligned} \tag{5.9}$$

Those matrices depend only on the mesh used, and only have to be computed once. Because the $\varphi_i's$ are polynomial, the integrals of Eq. 5.9 can be computed using Gauss integration [38].

On the other hand, the last term of the right hand side of Eq. 5.8 has not been discretized yet. This term involves the inter-element flux which whose discretization is called the numerical flux, and has no general expression. However, some guidelines allow to construct a numerical flux $\boldsymbol{F}^*$ that is convergent [39]:

1. the numerical flux depends on the value of $\boldsymbol{u}$ at the two sides of the boundary. Indeed, $\boldsymbol{u}_h$ is not well defined at the interface of two elements, since it has a value on the left and on the right of the boundary, the numerical flux has to be function of $\boldsymbol{u}_L$ and $\boldsymbol{u}_L$. One can choose the numerical flux to be the same for two elements sharing the same frontier. Combined with the projection on the normal $\boldsymbol{n}$, it then respects the facts that all information leaving one element arrives in the other.

$$\boldsymbol{F}^*(\boldsymbol{u}_{left}, \boldsymbol{u}_{right}) \tag{5.10}$$

2. the numerical flux is consistent. The numerical flux is said to be consistent if it is Lipschitz continuous in both the left and right states, *i.e.* if there exist a constant C such that

$$\begin{aligned} |\boldsymbol{F}^*(\boldsymbol{u}_{left}, \boldsymbol{u}_{right}) - \boldsymbol{F}^*(\boldsymbol{u}_{left,bis}, \boldsymbol{u}_{right})| &\leq C|\boldsymbol{u}_{left} - \boldsymbol{u}_{left,bis}| \\ |\boldsymbol{F}^*(\boldsymbol{u}_{left}, \boldsymbol{u}_{right}) - \boldsymbol{F}^*(\boldsymbol{u}_{left}, \boldsymbol{u}_{right,bis})| &\leq C|\boldsymbol{u}_{right} - \boldsymbol{u}_{right,bis}| \end{aligned} \tag{5.11}$$

and if

$$\boldsymbol{F}^{*}(\boldsymbol{u},\boldsymbol{u}) = \boldsymbol{F}(\boldsymbol{u}) \tag{5.12}$$

3. the numerical flux is stable. One example of stable flux are the E-flux. A numerical flux is an E-flux if, $\forall \boldsymbol{v} \in \{\theta \boldsymbol{u}_{left} + (1-\theta)\boldsymbol{u}_{right}, \theta \in [0,1]\}$, one has

$$(\boldsymbol{F}^{*}(\boldsymbol{u}_{left}, \boldsymbol{u}_{right}) - \boldsymbol{F}(v))(\boldsymbol{u}_{left} - \boldsymbol{u}_{right}) \geq 0 \tag{5.13}$$

   This stability property guarantees that the entropy of the system does not grow (the "E" in "E-flux" is for "Entropy-consistent"). It may seems suprising that the entropy is expected to decrease, and not to increase. The mathematical definition of the entropy corresponds to an energy measure of the solution [39]. It must then decrease with time, or at least remain constant.

The discretized equation for a general unidimensional equation is given by

$$M_{ij}(\partial_t \boldsymbol{u}_j - \boldsymbol{S}_j) - K_{ij}\boldsymbol{F}_j + \int_{\partial\Omega} \boldsymbol{F}^{*} \cdot \boldsymbol{n}\varphi_i dS = 0 \tag{5.14}$$

For the simplicity of the notation, let us define

$$\mathcal{F}_i = \int_{\partial\Omega} \boldsymbol{F}^{*} \cdot \boldsymbol{n}\varphi_i dS \tag{5.15}$$

Because of the properties of the outward pointing normal,

$$\mathcal{F}_i = \begin{cases} -F_i^{*} & \text{if } i \text{ corresponds to the left boundary of the element.} \\ F_i^{*} & \text{if } i \text{ corresponds tothe right boundary of the element.} \\ 0 & \text{in every other case.} \end{cases} \tag{5.16}$$

where $F_i^{*} = F^{*}(x_i)$, and x_i the position of a node.

The geometry of the domain has now to be taken into account in Eq. 5.14. The impact of the geometry is mostly felt in the mass matrix. Let us consider a reference element in the reference space with coordinate ξ. The reference element is a line of length 1 which extends in the interval $\xi \in [-0.5; 0.5]$. As a consequence, the jacobian of each element is given by its length (cfr Eq. 4.9 and Eq. 4.10). Fig. 5.2 illustrates the duality between the two spaces. Let us consider the mass and stiffness matrices for the reference element, labeled by $\hat{\boldsymbol{\mathcal{M}}}$ and

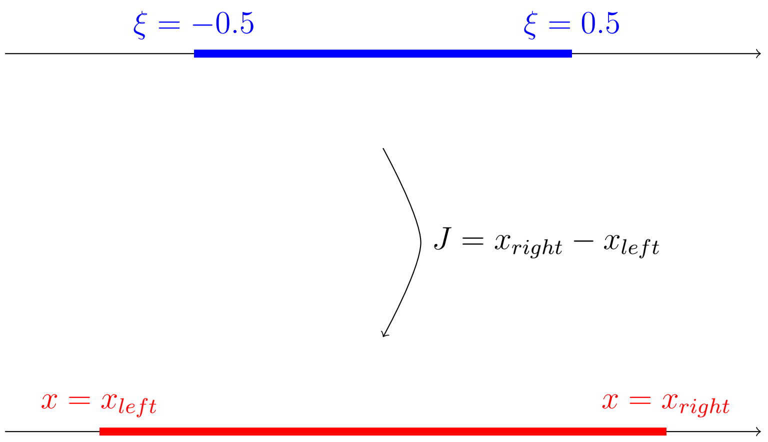


Figure 5.2: From the reference element in the ξ space to the real x space. The link between the two spaces is the Jacobian whose formula is given on the picture.

$\hat{\boldsymbol{\mathcal{K}}}$ respectively, which are such that

$$\left.\begin{aligned} \hat{\mathcal{M}}_{mn} &\equiv \int_{-0.5}^{0.5} \varphi_m(\xi)\varphi_n(\xi)d\xi \\ \hat{\mathcal{K}}_{mn} &\equiv \int_{-0.5}^{0.5} \frac{\partial\varphi_m(\xi)}{\partial\xi}\varphi_n(\xi)d\xi \end{aligned}\right. , \quad m, n = 1, 2, ..., \mathcal{N} \tag{5.17}$$

If J_k^G represents the geometrical Jacobian of the k^{th} element, one has $J_k^G d\xi = dx$ and

$$\begin{aligned} M_{ij} &= \int_{\Omega_k} \varphi_i(x)\varphi_j(x)dx = \int_{-0.5}^{0.5} \varphi_i(\xi)\varphi_j(\xi)J_k^G d\xi = \mathcal{M}_{ij}J_k^G \\ K_{ij} &= \int_{\Omega_k} \varphi_i(x)\partial_x\varphi_j(x)dx = \int_{-0.5}^{0.5} \varphi_i(\xi)(J_k^G)^{-1}\partial_\xi\varphi_j(\xi)J_k^G d\xi = \mathcal{K}_{ij} \end{aligned} \tag{5.18}$$

where the matrices $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{K}}$ are the bloc-diagonal matrices whose diagonal consist of the repetion of the $\hat{\boldsymbol{\mathcal{M}}}$ and $\hat{\boldsymbol{\mathcal{K}}}$ matrices respectively. Consequently, Eq. 5.14 can be written as

$$\mathcal{M}_{ij}J_k^G(\partial_t u_j - S_j) - \mathcal{K}_{ij}\boldsymbol{F}_j + \mathcal{F}_i = 0 \tag{5.19}$$

This expression will allow to store in the memory only the mass and stiffness matrices of the reference element and to multiply the mass matrix by the jacobian to obtain the mass matrix of an element. The stiffness matrix does not even need any further computation, as the stiffness is the same for all.

5.2 Application to the plasma equations

The mapping of variables and Eq. 5.19 is first presented. Then, the numerical flux used is given and finally the discretization of the ionization rate is given.

5.2.1 Mapping

The mapping with the variables is shown here. Recalling that the system to be discretized is

$$\begin{aligned} &\partial_t n_e + \partial_x(n_e u_e) &&= n_e \nu \\ &\partial_t n_i + \partial_x(n_i u_i) &&= n_e \nu \\ &\partial_t(n_e u_e) + \partial_x(n_e u_e^2 + n_e \varepsilon^{-1}) &&= \varepsilon^{-1} n_e \partial_x \phi \\ &\partial_t(n_i u_i) + \partial_x(n_i u_i^2 + n_i \kappa) &&= -n_i \partial_x \phi \end{aligned} \tag{2.16}$$

and following the numbering of the unknowns given in Section 4.2, one defines

$$u_j = \begin{cases} n_e(x_j) & 1 \le j \le N \\ n_i(x_j) & N+1 \le j \le 2N \\ n_e u_e(x_j) & 2N+1 \le j \le 3N \\ n_i u_i(x_j) & 3N+1 \le j \le 4N \end{cases}$$

$$F_j = \begin{cases} u_{2N+j} & 1 \le j \le N \\ u_{2N+j} & N+1 \le j \le 2N \\ u_j^2/u_{-2N+j} + \varepsilon^{-1} u_{-2N+j} & 2N+1 \le j \le 3N \\ u_j^2/u_{-2N+j} + \kappa u_{-2N+j} & 3N+1 \le i \le 4N \end{cases} \tag{5.20}$$

$$S_j = \begin{cases} u_j \nu & 1 \le j \le N \\ u_{-N+j} \nu & N+1 \le j \le 2N \\ \varepsilon^{-1} u_{-2N+j} \partial_x \phi(x_j) & 2N+1 \le j \le 3N \\ -u_{-2N+j} \partial_x \phi(x_j) & 3N+1 \le j \le 4N \end{cases}$$

where x_j represents the position of node j of the mesh and N is the total number of nodes on the mesh. The gradient of the potential is computed from the nodal values of the potential and the gradient of the shape functions as follow

$$\partial_x \phi(x) = \sum_{j=1}^{N} \phi(x_j) \partial_x \varphi_j(x) \tag{5.21}$$

The gradient of the electric potential can be computed only once as it is fixed and can be reused at will. From there, Eq. 5.19 can be written as the following set of equations:

1. if $1 \leq j \leq N$:

$$\mathcal{M}_{ij} J_k^G (\partial_t u_j - u_j \nu) - \mathcal{K}_{ij} u_{2N+j} + \mathcal{F}_i = 0 \tag{5.22}$$

2. if $N+1 \leq j \leq 2N$

$$\mathcal{M}_{ij} J_k^G (\partial_t u_j - u_{-N+j} \nu) - \mathcal{K}_{ij} u_{2N+j} + \mathcal{F}_i = 0 \tag{5.23}$$

3. if $2N+1 \leq j \leq 3N$

$$\mathcal{M}_{ij} J_k^G (\partial_t u_j - \varepsilon^{-1} u_{-2N+j} \partial_x \phi(x_j) \nu) - \mathcal{K}_{ij} \left(u_j^2 / u_{-2N+j} + \varepsilon^{-1} u_{-2N+j} \right) + \mathcal{F}_i = 0 \tag{5.24}$$

4. if $3N+1 \leq j \leq 4N$

$$\mathcal{M}_{ij} J_k^G (\partial_t u_j + u_{-2N+j} \partial_x \phi(x_j)) - \mathcal{K}_{ij} \left(u_j^2 / u_{-2N+j} + \kappa u_{-2N+j} \right) + \mathcal{F}_i = 0 \tag{5.25}$$

5.2.2 Numerical flux

The numerical flux used for this work is the Roe numerical flux [40]. This Section presents briefly how such a flux is found.

Let us recall that it was possible to rewrite the plasma system of equations as

$$\partial_t \boldsymbol{u} + \boldsymbol{A} \partial_x \boldsymbol{u} = \boldsymbol{S} \tag{3.3}$$

where $\boldsymbol{A} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}$. The main idea used by Roe to solve this equation is to approximate $\boldsymbol{A}(\boldsymbol{u})$ by $\boldsymbol{A}_h(\boldsymbol{u}_l, \boldsymbol{u}_r)$. $\boldsymbol{A}_h$ should follow the following constructions rules:

1. $\boldsymbol{A}_h(\boldsymbol{u}_l, \boldsymbol{u}_r)(\boldsymbol{u}_l - \boldsymbol{u}_r) = \boldsymbol{F}(\boldsymbol{u}_l) - \boldsymbol{F}(\boldsymbol{u}_r)$ for conservativity.

2. $\boldsymbol{A}_h$ is diagonalizable with real eigenvalues to keep the hyperbolicity of the system.

3. $\boldsymbol{A}_h(\boldsymbol{u}_l, \boldsymbol{u}_r) \rightarrow \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}(\boldsymbol{u})$ when $\boldsymbol{u}_l, \boldsymbol{u}_r \rightarrow \boldsymbol{u}$ for consistency.

For a more complete explanation of these constructions rules, the reader can refer to [41].

The numerical flux is obtained from $\boldsymbol{A}$ from the expression

$$\boldsymbol{F}^{*}(\boldsymbol{u}_l, \boldsymbol{u}_r) = \frac{\boldsymbol{F}(\boldsymbol{u}_l) + \boldsymbol{F}(\boldsymbol{u}_R)}{2} - |\boldsymbol{A}_h(\boldsymbol{u}_r, \boldsymbol{u}_l)|(\boldsymbol{u}_r - \boldsymbol{u}_l) \tag{5.26}$$

where $|\boldsymbol{A}_h(\boldsymbol{u}_r, \boldsymbol{u}_l)|$ is the matrix that has the same eigenvectors as $\boldsymbol{A}_h$, but whose eigenvalues are the absolute values of the eigenvalues of $\boldsymbol{A}_h$. If $\bar{\lambda}_i(u_r, u_l)$ and $\bar{\boldsymbol{v}}_i(u_r, u_l)$ are respectively an eigenvalue and an eigenvector of $\boldsymbol{A}_h$, one has

$$\boldsymbol{F}^{*} = \frac{\boldsymbol{F}(u_l) + \boldsymbol{F}(u_r)}{2} - \frac{1}{2}\sum_{\alpha=0}^{3} a_\alpha |\bar{\lambda}_\alpha| \bar{\boldsymbol{v}}_\alpha \tag{5.27}$$

Because the system of equations used here does not take into account any diffusion term, the left hand side of Eq. 2.16 is a system of Euler equations for fluid dynamics. Roe [40] has derived the form of the $\boldsymbol{A}_h$ matrix. The main idea is the following. We want to evaluate the numerical flux of Eq. 5.26 by evaluating it at the average of a variable p, named $\bar{p}$. p must be chosen such that both the conserved variables and the physical flux are quadratic functions of it. This property allows to write the physical flux as

$$F_R - F_R = \underbrace{\frac{\partial F}{\partial p}}_{=\boldsymbol{A}(\bar{p})}(\bar{p})(p_r - p_l) \tag{5.28}$$

By choosing $\boldsymbol{A}_h(\bar{p}) = \boldsymbol{A}(\bar{p})$, all requirements demanded by Roe are fullfilled (*cfr* Appendix C.1). This is where the simplicity of the method lies : the numerical flux can be determined using the original matrix $\boldsymbol{A}$ evaluated at a given average. The final form of $\boldsymbol{A}_h$ is

$$\boldsymbol{A}_h = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \varepsilon^{-1} - \bar{u}_e^2 & 0 & 2\bar{u}_e & 0 \\ 0 & \kappa - \bar{u}_i^2 & 0 & 2\bar{u}_i \end{pmatrix} \tag{5.29}$$

where

$$\bar{u}_e = \frac{\frac{n_e u_{e,L}}{\sqrt{n_{e,L}}} + \frac{n_e u_{e,R}}{\sqrt{n_{e,R}}}}{\sqrt{n_{e,L}} + \sqrt{n_{e,R}}} \tag{5.30}$$

and

$$\bar{u}_i = \frac{\frac{n_i u_{i,L}}{\sqrt{n_{i,L}}} + \frac{n_i u_{i,R}}{\sqrt{n_{i,R}}}}{\sqrt{n_{i,L}} + \sqrt{n_{i,R}}} \tag{5.31}$$

are respectively called the electron and ion Roe-averaged velocities. Eq. 5.29 is the same matrix as $\boldsymbol{A}$ (Eq. 3.4), except that all velocities have been replaced by the corresponding Roe velocities. Consequently, the eigenvalues and eigenvectors of $\boldsymbol{A}_h$ are the ones of $\boldsymbol{A}$ where the Roe velocities are used:

$$\begin{aligned}
\bar{\lambda}_0 &= \bar{u}_e - \sqrt{\varepsilon^{-1}} \\
\bar{\lambda}_1 &= \bar{u}_e + \sqrt{\varepsilon^{-1}} \\
\bar{\lambda}_2 &= \bar{u}_i - \sqrt{\kappa} \\
\bar{\lambda}_3 &= \bar{u}_i + \sqrt{\kappa} \\
\bar{\boldsymbol{v}}_0 &= \begin{pmatrix} 1 & 0 & \bar{\lambda}_0 & 0 \end{pmatrix} \\
\bar{\boldsymbol{v}}_1 &= \begin{pmatrix} 1 & 0 & \bar{\lambda}_1 & 0 \end{pmatrix} \\
\bar{\boldsymbol{v}}_2 &= \begin{pmatrix} 0 & 1 & 0 & \bar{\lambda}_2 \end{pmatrix} \\
\bar{\boldsymbol{v}}_3 &= \begin{pmatrix} 0 & 1 & 0 & \bar{\lambda}_3 \end{pmatrix}
\end{aligned} \tag{5.32}$$

Knowing from 5.26 and Eq. 5.27 that

$$\boldsymbol{u}_R - \boldsymbol{u}_L = \sum_{\alpha=0}^{3} a_\alpha \boldsymbol{v}_\alpha \tag{5.33}$$

it can be easily found from the eigen values and vectors of $\boldsymbol{A}_h$ that

$$\begin{aligned}
a_0 &= \frac{(n_{e,R} - n_{e,L})\bar{\lambda}_1 - (n_e u_{e,R} - n_e u_{e,L})}{2\sqrt{\varepsilon^{-1}}} \\
a_1 &= -\frac{(n_{e,R} - n_{e,L})\bar{\lambda}_0 - (n_e u_{e,R} - n_e u_{e,L})}{2\sqrt{\varepsilon^{-1}}} \\
a_2 &= \frac{(u_{i,R} - u_{i,L})\bar{\lambda}_3 - (n_i u_{i,R} - n_i u_{i,L})}{2\sqrt{\kappa}} \\
a_3 &= -\frac{(u_{i,R} - u_{i,L})\bar{\lambda}_2 - (n_i u_{i,R} - n_i u_{i,L})}{2\sqrt{\kappa}}
\end{aligned} \tag{5.34}$$

Let us consider the index i corresponding to any boundary node of an element. One has

$$\boldsymbol{F}_j^* = \begin{cases}
\dfrac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} - \dfrac{a_{0,j}|\bar{\lambda}_{0,j}| + a_{1,j}|\bar{\lambda}_{1,j}|}{2} & 1 \leq j \leq N \\
\dfrac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} - \dfrac{a_{2,j}|\bar{\lambda}_{2,j}| + a_{3,j}|\bar{\lambda}_{3,j}|}{2} & N+1 \leq j \leq 2N \\
\dfrac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} - \dfrac{a_{0,j}|\bar{\lambda}_{0,j}|\bar{\lambda}_{0,j} + a_{1,j}|\bar{\lambda}_{1,j}|\bar{\lambda}_{1,j}}{2} & 2N+1 \leq j \leq 3N \\
\dfrac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} - \dfrac{a_{2,j}|\bar{\lambda}_{2,j}|\bar{\lambda}_{2,j} + a_{3,j}|\bar{\lambda}_{3,j}|\bar{\lambda}_{3,j}}{2} & 3N+1 \leq j \leq 4N
\end{cases} \tag{5.35}$$

Eq. 5.35 seems complex. However, it can be shown that (see Appendix C.2 for demonstration)

1. if $\bar{\lambda}_0 > 0 \implies \bar{\lambda}_1 > 0$,

$$\boldsymbol{F}_j^* = \boldsymbol{F}_{j,l} \quad 1 \leq j \leq N, \quad 2N+1 \leq j \leq 3N \tag{5.36}$$

2. if $\bar{\lambda}_1 < 0 \implies \bar{\lambda}_0 < 0$,

$$\boldsymbol{F}_j^* = \boldsymbol{F}_{j,R} \quad 1 \leq j \leq N, \quad 2N+1 \leq j \leq 3N \tag{5.37}$$

3. if $\bar{\lambda}_2 > 0 \implies \bar{\lambda}_3 > 0$,

$$\boldsymbol{F}_j^* = \boldsymbol{F}_{j,l} \quad N+1 \leq j \leq 2N, \quad N+1 \leq j \leq 2N \tag{5.38}$$

4. if $\bar{\lambda}_3 < 0 \implies \bar{\lambda}_2 < 0$,

$$\boldsymbol{F}_j^* = \boldsymbol{F}_{j,R} \quad N+1 \leq j \leq 2N, \quad 3N+1 \leq j \leq 4N \tag{5.39}$$

Eq. 5.36, Eq. 5.37, Eq. 5.38 and Eq. 5.39 reveal that the Roe numerical flux is upwind. Indeed, all these cases correspond to suspersonic electrons or ions going to the right or the left. For those cases, the flux takes the information from its source : upwind with respect to the supersonic velocity considered. Let us take Eq. 5.36 as an example. When $\bar{\lambda}_0 > 0$, it means that the velocity of electrons is supersonic and propagates from left to right. As a consequence, the system takes the information from the left, and convects it to the right: the numerical flux equals the physical flux evaluated at the left. One of the main advantage of such flux is that it uses only the information it needs to be computed. This property will be useful in the application of the boundary conditions.

Two cases have yet to be discussed. Their demonstration can also be found in Appendix C.2.

1. if $\bar{\lambda}_0 < 0$ and $\bar{\lambda}_1 > 0$,

   (a) $1 \leq j \leq N$,

$$\boldsymbol{F}_j^* = \frac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} + \frac{1}{2\sqrt{\varepsilon^{-1}}} \left[\lfloor u_j \rceil (\bar{u}_{e,j}^2 - \varepsilon^{-1}) - \bar{u}_{e,j} \lfloor u_{2N+j} \rceil \right] \tag{5.40}$$

(b) $2N+1 \leq j \leq 3N$,

$$\boldsymbol{F}_j^* = \frac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} + \frac{1}{2\sqrt{\varepsilon^{-1}}}\left[\lfloor u_{-2N+j} \rceil (\bar{u}_{e,j}^2 - \varepsilon^{-1})\bar{u}_{e,j} - (\bar{u}_{e,j}^2 + \varepsilon^{-1})\lfloor u_j \rceil\right] \quad (5.41)$$

2. if $\bar{\lambda}_2 < 0$ and $\bar{\lambda}_3 > 0$,

(a) $N+1 \leq j \leq 2N$,

$$\boldsymbol{F}_j^* = \frac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} + \frac{1}{2\sqrt{\kappa}}\left[\lfloor u_j \rceil (\bar{u}_{i,j}^2 - \kappa) - \bar{u}_{i,j}^2 \lfloor u_{2N+j} \rceil\right] \quad (5.42)$$

(b) $3N+1 \leq j \leq 4N$,

$$\boldsymbol{F}_j^* = \frac{\boldsymbol{F}_{j,L} + \boldsymbol{F}_{j,R}}{2} + \frac{1}{2\sqrt{\kappa}}\left[\lfloor u_{-2N+j} \rceil (\bar{u}_{e,j}^2 - \kappa)\bar{u}_{e,j} - (\bar{u}_{e,j}^2 + \kappa)\lfloor u_j \rceil\right] \quad (5.43)$$

where the j index refers to the value at the j^{th} node and $\lceil u_j \rfloor$ is the difference between the two values of u_j at the boundary of an element. If the boundary is a right boundary,

$$\lfloor u_j \rceil = u_j - u_{j-1} \quad (5.44)$$

while, for the right boundary of an element,

$$\lfloor u_j \rceil = u_{j+1} - u_j \quad (5.45)$$

Note that at the left (right) boundary of the domain, u_{j-1} (u_{j+1}) are not defined. They must be determined through the boundary conditions.

5.2.3 Ionization rate

In order to discretize the ionization rate, the methodology developped by [14] is used also here. It consists in discretizing ν by

$$\nu(t) = \frac{|n_i u_i(L,t)| + |n_i u_i(0,t)|}{\int_0^L n_e(x,t)dx} \quad (5.46)$$

This choice is justified by the fact that, at steady state, the ionization rate is the one that exactly compensates the ion flux going out of the boundaries. This equilibrium state is simply transposed at all times of the simulation.

Another way to formulate the balance given by Eq. 5.46 is to state that the temperature is determined by the geometry of the system. Indeed, the ionization rate ν depends on temperature through the relation

$$\nu = Ae^{\frac{-\varepsilon_{ion}}{k_B T_e}} \tag{5.47}$$

with A a normalization constant and ε_{ion} the ionization energy. Eq. 5.47 and Eq. 5.46 in combination offer an evaluation at each time step of the electronic temperature, where geometry plays a important role through the evaluation of the mass of the electrons (denominator of the right hand side of Eq. 5.46).

5.3 Boundary conditions

The boundary conditions are an important part in the simulation set-up. If not correctly set, the solver will never be able to find the solution. This Section incorporates the boundary conditions determined in Section 2.4 to the numerical discretization.

As explained in the previous Section, the Roe numerical flux is an upwind flux, *i.e.* it selects the information on both sides of the boundary of the elements. In the supersonic case, it will select the physical flux on only one side, while for subsonic cases it will take information coming from both sides of the frontier. Because overconstraining the boundary conditions does not impact the Roe numerical flux thanks to its selection, we can set a boundary condition for both ions and electrons at the boundary.

The main idea behind the application of boundary condition is to impose a fixed value to the electron and ion flux at the boundaries and to impose an expression for the electron and ion densities which depends on the inner variables. This will change depending on the fact that the caracteristics are entering or leaving the system.

$$\begin{aligned} F_j &= -\bar{F} \\ F_{j+N-1} &= \bar{F} \end{aligned} \quad , \quad j = 2N+1, 3N+1 \tag{5.48}$$

where $\bar{F}$ obeys the following rules:

1. if the particle velocity is positive (negative) on the right (left) boundary, then $\bar{F}$ is chosen to correspond ro Eq. 2.19.

2. if the particle velocity is negative (positive) on the right (left) electrode, then $\bar{F}$ is chosen as the value of the inner flux.

Concerning the densities, one has

$$\begin{aligned} u_1 &= \sqrt{2\pi\varepsilon}|u_{2N+1}| \\ u_N &= \sqrt{2\pi\varepsilon}|u_{3N}| \\ u_{N+1} &= u_{N+2} \\ u_{2N} &= u_{2N-1} \end{aligned} \tag{5.49}$$

The two first equations of Eq. 5.49 account for the fact that the outer electron density is determined by the ratio between the absolute value of the inner electron flux and the physical electron velocity. This ratio is set to the expected value of leaving electrons The two last equations states simply that the ghost cell value of the ion density is always equal to the inner cell one.

Now that the spatial discretization has been properly performed, the question of time discretization remain. The temporal schemes are discussed in the following Chapter.

Chapter 6

Temporal schemes

This Chapter deals with the time discretization of the system. This discretization is performed along two angles of attack: an explicit method, and an implict one. First, the fourth order Runge-Kutta method [42, 43] is presented. Then, the backward Euler method second order accurate is developed, along with the jacobian of the system. Both methods have an adaptative time step that satisfies a given CFL number.

6.1 Optimal time step

At each simulation time, the optimal time step to satisfy a given CFL is computed. First, the velocities of the electrons and ions are computed at each node. Velocities are computed for each species such that (v_i for ions and v_e for electrons velocities respectively):

$$\begin{aligned} v_i &= \frac{n_i u_i}{n_i} \\ v_e &= \frac{n_e u_e}{n_e} \end{aligned} \tag{6.1}$$

Then, all the eigenvalues of the system are computed at each node of the mesh. The largest eigenvalue of all in magnitude λ_{max} is the largest propagation velocity. Then, considering that a CFL is given, that the elements are of order p and that the smallest element is of length l_{min}, one has [45]

$$\frac{|\lambda_{max}|\Delta t(2p+1)}{l_{min}} = CFL \tag{6.2}$$

so that the optimal time step for the given CFL is

$$\Delta t = \frac{l_{min} CFL}{(2p+1)|\lambda_{max}|} \tag{6.3}$$

6.2 Fourth order Runge-Kutta method

The fourth order Runge-Kutta method (RK4) is easy to implement and is explicit, meaning that it allows a relatively fast resolution of the problem compared to an implicit one.

Starting from Eq. 5.14 explicited in the form

$$(\partial_t u_j)^n = \underbrace{(J_k^G)^{-1}\mathcal{M}_{ij}^{-1}\left[\mathcal{K}_{jl}F_l - \mathcal{F}_j\right] + S_j}_{f(u_j^{n-1},t)} \tag{6.4}$$

the RK4 method for such an equation states that

$$\partial_t u_j \simeq \frac{(\partial_t u_j)^n - (\partial_t u_j)^{n-1}}{\Delta t^n} + \frac{1}{6}\left(k_{1,j} + 2k_{2,j} + 2k_{3,j} + k_{4,j}\right) \tag{6.5}$$

where n refers to the time step at which the $\partial_t u_j$ is evaluated and Δt^n is the simulation time step at timestep n, and [44]

$$\begin{aligned}
k_{1,j} &= f(u_j^n, t^n) \\
k_{2,j} &= f(u_j^n + \frac{\Delta t^n}{2}\, k_{1,j}, t^n + \frac{\Delta t^n}{2}) \\
k_{3,j} &= f(u_j^n + \frac{\Delta t^n}{2}\, k_{2,j}, t^n + \frac{\Delta t^n}{2}) \\
k_{4,j} &= f(u_j^n + \Delta t^n\, k_{3,j}, t^n + \Delta t^n)
\end{aligned} \tag{6.6}$$

with t^n the instant corresponding to the n^{th} time iteration. The implementation of the method is very straightforward. However, it is not the case for the backward Euler method.

6.3 Backward euler method second order accurate

While the Runge-Kutta method is explicit and easy to implement, the backward Euler method is implicit and needs much more care. However, the implicit aspect of the method makes it inherently more stable than the Runge Kutta one. One has

$$(\partial_t u_j)^n = \underbrace{(J_k^G)^{-1}\mathcal{M}_{ij}^{-1}\left[\mathcal{K}_{il}F_l - \mathcal{F}_j\right] + S_j}_{f(u_j^n,t)} \tag{6.7}$$

The main difference between Eq. 6.7 and Eq. 6.4 is that f is a function of u_j^n and not of u_j^{n-1}. The discretization of the temporal term second order accurate with an adaptative time step is (see Appendix D for proof)

$$\begin{aligned}\partial_t u_j^n \quad &\simeq \underbrace{\left[\frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_n \Delta t_{n-1}} - \frac{\Delta t_n}{\Delta t_{n-1}(\Delta t_n + \Delta t_{n-1})}\right]}_{=a_1} u_j^n \\ &\underbrace{-\frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_n \Delta t_{n-1}}}_{=a_2} u_j^{n-1} \\ &\underbrace{+\frac{\Delta t_n}{\Delta t_{n-1}(\Delta t_n + \Delta t_{n-1})}}_{=a_3} u_j^{n-2}\end{aligned} \tag{6.8}$$

where n is the current time step of the resolution. The system can be expressed as

$$\mathcal{M}_{ij} J_k^G \left(a_1 u_j^n + a_2 u_j^{n-1} + a_3 u_j^{n-2} - S_j^n\right) - \mathcal{K}_{ij} F_j^n + \mathcal{F}_i = 0 \tag{6.9}$$

In order to solve the system, the Newton-Raphson [46] method is applied. Such method consists in general to solve an equation involving a non linear system $\boldsymbol{B}$ depending on a varibable $\boldsymbol{u}$ (the unknown) of the form

$$\boldsymbol{B}(\boldsymbol{u}) = \boldsymbol{0} \tag{6.10}$$

The method simply consists in expanding $\boldsymbol{B}$ in Taylor series around a certain value $\boldsymbol{u}_0$, called the initial guess vector

$$\boldsymbol{B}(\boldsymbol{u}) \simeq \boldsymbol{B}(\boldsymbol{u}_0) + \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{u}}(\boldsymbol{u}_0)(\boldsymbol{u} - \boldsymbol{u}_0) \tag{6.11}$$

and then to inject this linearized system in the original equation so that

$$\boldsymbol{u} \simeq \boldsymbol{u}_0 - \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{u}}(\boldsymbol{u}_0)\boldsymbol{B}(\boldsymbol{u}_0) = \boldsymbol{u}_1 \tag{6.12}$$

where $\boldsymbol{u}_1$ is the approximation of $\boldsymbol{u}$. Then, $\boldsymbol{B}(\boldsymbol{u}_1)$ is evaluated. If it is close enough to zero (the threshold is user defined), the algorithm stops and u_1 is taken as the approximate solution of the system. If it is not, u_1 becomes u_0 in another iteration of the algorithm.

In the present work, the matrix $\boldsymbol{B}$ corresponds to

$$B_i = \mathcal{M}_{ij} J_k^G \left(a_1 u_j^n + a_2 u_j^{n-1} + a_3 u_j^{n-2} - S_j^n\right) - \mathcal{K}_{ij} F_j^n + \mathcal{F}_i \tag{6.13}$$

and its jacobian must be computed. Before computing the jacobian, we define the (i, l)

element of the jacobian of a vector $\boldsymbol{q}$:

$$J_{il} \equiv \frac{\partial q_i}{\partial u_l} \equiv q_i' \tag{6.14}$$

The jacobian of the temporal term of Eq. 6.9 is given immediately by (Einstein summation convention is used)

$$J_{il}^t = a_1 \delta_{jl} \tag{6.15}$$

The other jacobians are less straightforward to compute.

Source term jacobian In order to compute the jacobian of the source term, the jacobian of the ionization rate must be known.

$$\nu' = \frac{\partial \nu}{\partial u_l} = \frac{\partial}{\partial u_l}\left[\frac{|n_e u_e(L)| - |n_e u_e(0)|}{M_e}\right] \tag{6.16}$$

where M_e is the approximate total electronic mass. The total electronic mass is computed using Gauss formula

$$\int_0^L n_e dx \simeq M_e = \sum_{g=1}^{G} n_e(x_g) w(x_g) \tag{6.17}$$

where x_g is the position of the g^{th} gauss point whose weight is given by $w(x_g)$ and G is the total number of Gauss points over the domain. Note that the Gauss weight is linked to the corresponding gauss weight in the reference $\omega(\xi_{g'})$ element by

$$w(x_g) = \omega(\xi_{g'}) J_k^G \tag{6.18}$$

where J_k^G is the geometrical jacobian of the k^{th} element which contains the gauss point g and $\xi_{g'}$ is the position of the corresponding gauss point in the reference element to x_g. $n_e(x_g)$ can be approximated by reconstruction with the shape functions:

$$n_{e,h}(x_g) \equiv \sum_{j=1}^{N} n_e(x_j) \varphi_j(x_g) \tag{6.19}$$

so that

$$M_e = \sum_{g=1}^{G} \sum_{j=1}^{N} n_e(x_j) \varphi_j(x_g) \omega(\xi_g) J_k^G \tag{6.20}$$

As a consequence,

$$M'_e = \begin{cases} \displaystyle\sum_{g=1}^{G} \varphi_l(x_g)\omega(\xi_g)J_k^G & 1 \leq j \leq N \\ 0 & \text{else} \end{cases} \tag{6.21}$$

From those developments, ν' is given by (following the numerotation of Section 4.2)

$$\nu' = \begin{cases} \dfrac{\text{sign}(u_{3N})}{M_e}\delta_{3N,l} - \dfrac{|u_{3N}| - |u_{2N+1}|}{M_e^2}M'_e \\ \dfrac{\text{sign}(u_{2N+1})}{M_e}\delta_{2N+1,l} - \dfrac{|u_{3N}| - |u_{2N+1}|}{M_e^2}M'_e \end{cases} \tag{6.22}$$

which directly leads to the jacobian of the source term

$$J^S_{jl} = \begin{cases} \nu' u_j + \nu & \text{if} \quad j = l, \qquad 1 \leq j \leq N \\ \nu' u_{j-N} + \nu & \text{if} \quad j - N = l, \quad N < j \leq 2N \\ \varepsilon^{-1}\partial_x\phi & \text{if} \quad i - 2N = l, \quad 2N < i \leq 3N \\ -\partial_x\phi & \text{if} \quad i - 2N = l, \quad 2N < i \leq 3N \\ 0 & \text{else} \end{cases} \tag{6.23}$$

Physical flux jacobian This jacobian is straightforward and gives

$$J^F_{jl} = \begin{cases} 1 & \text{if} \quad j + 2N = l, \quad 1 \leq j \leq N \\ 1 & \text{if} \quad j + 2N = l, \quad N < j \leq 2N \\ \varepsilon^{-1} - \left(\dfrac{u_j}{u_{j-2N}}\right)^2 & \text{if} \quad j - 2N = l, \quad 2N < j \leq 3N \\ \dfrac{2u_j}{u_{j-2N}} & \text{if} \quad j = l, \qquad 2N < j \leq 3N \\ \kappa - \left(\dfrac{u_j}{u_{j-2N}}\right)^2 & \text{if} \quad j - 2N = l, \quad 3N < j \leq 4N \\ \dfrac{2u_j}{u_{j-2N}} & \text{if} \quad j = l, \qquad 3N < j \leq 4N \\ 0 & \text{else} \end{cases} \tag{6.24}$$

Jacobian of the numerical flux The jacobian of the numerical flux requires the knowledge of the jacobian of

1. the left and right states at u_j,

2. the Roe velocity at u_j,

For the left and right states jacobian, one has

$$\begin{aligned}
\frac{\partial u_{j,L}}{\partial u_l} &= \begin{cases} \delta_{j-1,l} \text{ if it's the left boundary of an element.} \\ \delta_{j,l} \text{ if it's the right boundary of an element.} \end{cases} \\
\frac{\partial u_{j,R}}{\partial u_l} &= \begin{cases} \delta_{j,l} \text{ if it's the left boundary of an element.} \\ \delta_{j+1,l} \text{ if it's the right boundary of an element.} \end{cases}
\end{aligned} \qquad 2 \leq j \leq 4N-1 \tag{6.25}$$

The cases where $j = 1$ and $j = 4N$ must take into account the boundary conditions. In that case, the jacobian follows the rules of Eq. 6.25, except that

$$\begin{aligned}
u'_{1,L} &= sign(u_{2N})\delta_{2N+1,l}\sqrt{2\pi\varepsilon} \\
u'_{N,R} &= sign(u_{2N})\delta_{3N,l}\sqrt{2\pi\varepsilon} \\
u'_{N+1,L} &= 0 \\
u'_{2N,R} &= 0 \\
u'_{2N+1,L} &= \delta_{2N+1,l} \\
u'_{3N,R} &= \delta_{3N-1,l} \\
u'_{3N+1,L} &= 0 \\
u'_{4N,L} &= 0
\end{aligned} \tag{6.26}$$

These results allow to compute the derivative of $\lceil u_j \rfloor$ (Eq. 5.44 and Eq. 5.45), which is simply in a compact way

$$\lceil u_j \rfloor' = u'_{j,R} - u'_{j,L} \tag{6.27}$$

The jacobians of the Roe-averaged velocities can be proven to be

$$\begin{aligned}
(\bar{u}_{e,j})' &= \frac{\frac{u_L^{2\,\prime}}{\sqrt{\bar{u}_L^0}} + \frac{u_R^{2\,\prime}}{\sqrt{u_R^0}} - \frac{u_L^2 u_L^{0\,\prime}}{2u_L^0\sqrt{u_L^0}} - \frac{u_R^2 u_R^{0\,\prime}}{2u_R^0\sqrt{u_R^0}} - \left(\frac{u_L^{0\,\prime}}{(\sqrt{2u_L^0}} + \frac{u_R^{0\,\prime}}{\sqrt{2u_R^0}}\right) u_{e,j}}{\sqrt{u_L^0} + \sqrt{u_R^0}} \\
(u_{i,j})' &= \frac{\frac{u_L^{3\,\prime}}{\sqrt{u_L^1}} + \frac{u_R^{3\,\prime}}{\sqrt{u_R^1}} - \frac{u_L^3 u_L^{1\,\prime}}{3u_L^1\sqrt{u_L^1}} - \frac{u_R^3 u_R^{1\,\prime}}{3u_R^1\sqrt{u_R^1}} - \left(\frac{u_L^{1\,\prime}}{(\sqrt{3u_L^1}} + \frac{u_R^{1\,\prime}}{\sqrt{3u_R^1}}\right) u_{e,j}}{\sqrt{u_L^1} + \sqrt{u_R^1}}
\end{aligned} \tag{6.28}$$

where $u_{e,j}$ (Eq. 5.30) and $u_{i,j}$ (Eq. 5.31) are the electronic and ionic Roe-averaged velocities. The upperscripts 1, 2, 3 and 4 refer to the electron and ion densities, and the electron and ion momentum respectively at the j^{th} node.

From the results of Section 5.2.2, one has

1. if $\bar{\lambda}_0 > 0 \implies \bar{\lambda}_1 > 0$,

$$F_j^{*\prime} = F'_{j,l} \quad 1 \leq j \leq N, \quad 2N+1 \leq j \leq 3N \tag{6.29}$$

2. if $\bar{\lambda}_1 < 0 \implies \bar{\lambda}_0 < 0$,

$$F_j^{*\prime} = F'_{j,R} \quad 1 \leq j \leq N, \quad 2N+1 \leq j \leq 3N \tag{6.30}$$

3. if $\bar{\lambda}_2 > 0 \implies \bar{\lambda}_3 > 0$,

$$F_j^{*\prime} = F'_{j,L} \quad N+1 \leq j \leq 2N, \quad N+1 \leq j \leq 2N \tag{6.31}$$

4. if $\bar{\lambda}_3 < 0 \implies \bar{\lambda}_2 < 0$,

$$F_j^{*\prime} = F'_{j,R} \quad N+1 \leq j \leq 2N, \quad 3N+1 \leq j \leq 4N \tag{6.32}$$

5. if $\bar{\lambda}_0 < 0$ and $\bar{\lambda}_1 > 0$,

   (a) $1 \leq j \leq N$,

$$\begin{aligned} F_j^{*\prime} &= \frac{F'_{j,L} + F'_{j,R}}{2} \\ &+ \frac{\lfloor u_j \rceil'(\bar{u}_{e,j}^2 - \varepsilon^{-1}) + \lfloor u_j \rceil 2u_{\bar{e},j}\bar{u}'_{e,j} - \bar{u}'_{e,j}\lfloor u_{2N+j} - \bar{u}_{e,j}\lfloor u_{2N+j} \rceil'}{2\sqrt{\varepsilon^{-1}}} \end{aligned} \tag{6.33}$$

   (b) $2N+1 \leq j \leq 3N$,

$$\begin{aligned} F_j^{*\prime} &= \frac{F'_{j,L} + F'_{j,R}}{2} \\ &+ \frac{\lfloor u_{-2N+j} \rceil'(\bar{u}_{e,j}^2 - \varepsilon^{-1})\bar{u}_{e,j} + \lfloor u_{-2N+j} \rceil 2\bar{u}_{e,j}^2\bar{u}'_{e,j}}{2\sqrt{\varepsilon^{-1}}} \\ &+ \frac{\lfloor u_{-2N+j} \rceil(\bar{u}_{e,j}^2 - \varepsilon^{-1})\bar{u}'_{e,j} - (\bar{u}_{e,j}^2 + \varepsilon^{-1})\lfloor u_j \rceil' - 2\bar{u}_{e,j}\bar{u}'_{e,j}\lfloor u_j \rceil}{2\sqrt{\varepsilon^{-1}}} \end{aligned} \tag{6.34}$$

6. if $\bar{\lambda}_2 < 0$ and $\bar{\lambda}_3 > 0$ and

(a) $N+1 \leq j \leq 2N$,

$$\begin{aligned} F_j^{*\prime} &= \frac{F'_{j,L} + F'_{j,R}}{2} \\ &+ \frac{\lfloor u_j \rceil'(\bar{u}_{i,j}^2 - \varepsilon^{-1}) + \lfloor u_j \rceil 2u\bar{}_{i,j}\bar{u}'_{i,j} - \bar{u}'_{i,j}\lfloor u_{2N+j} - \bar{u}_{i,j}\lfloor u_{2N+j}\rceil'}{2\sqrt{\kappa}} \end{aligned} \tag{6.35}$$

(b) $3N+1 \leq j \leq 4N$,

$$\begin{aligned} F_j^{*\prime} &= \frac{F'_{j,L} + F'_{j,R}}{2} \\ &+ \frac{\lfloor u_{-2N+j} \rceil'(\bar{u}_{i,j}^2 - \varepsilon^{-1})\bar{u}_{i,j} + \lfloor u_{-2N+j} \rceil 2\bar{u}_{i,j}^2\bar{u}'_{i,j}}{2\sqrt{\kappa}} \\ &+ \frac{\lfloor u_{-2N+j} \rceil (\bar{u}_{i,j}^2 - \varepsilon^{-1})\bar{u}'_{i,j} - (\bar{u}_{i,j}^2 + \varepsilon^{-1})\lfloor u_j \rceil' - 2\bar{u}_{i,j}\bar{u}'_{i,j}\lfloor u_j \rceil}{2\sqrt{\kappa}} \end{aligned} \tag{6.36}$$

The previous developments allow to express the jacobian of the numerical flux term. From Eq. 5.16, one can recover

$$J_{jl}^{F^*} = \begin{cases} 0 \text{ if } j \text{ is not a point at the frontier of an element.} \\ (F^*)'_j \text{ if } j \text{ is the right boundary point of an element.} \\ -(F^*)'_j \text{ if } j \text{ is the left boundary point of an element.} \end{cases} \tag{6.37}$$

Injecting the results from Eq. 6.15, Eq. 6.23, Eq. 6.24 and Eq. 6.37 in the jacobian of the left hand side of Eq. 6.9 gives

$$J_{il} = \mathcal{M}_{ij} J_k^G \left(J_{jl}^t - J_{jl}^S \right) - \mathcal{K}_{ij} J_{jl}^F + J_{il}^{F^*} \tag{6.38}$$

The algorithm to resolve the system from initial step $k-1$ to step k is

$$\begin{aligned} u_j^k &= u_j^{k-1} \\ &- J_{jl}^{-1} \left(M_{lm} \left(\frac{3u_m^n - 4u_m^{n-1} + u_m^{n-2}}{2} - S_m^n \right) - K_{lm} F_m^n + \mathcal{F}_l \right) \Bigg|_{u_j^{k-1}} \end{aligned} \tag{6.39}$$

Because the method is second order accurate in time, it normally requires two initial conditions. To avoid that, the first time step is computed with a first order backward euler method. As a consequence, it modifies the jacobian associated to the time step at the beginning of the simulation and the expression of the temporal term:

$$\partial_t u_j^n \simeq \frac{u_j^n - u_j^{n-1}}{\Delta t^n} \tag{6.40}$$

and

$$J_{jl} = \frac{\delta_{jl}}{\Delta t} \tag{6.41}$$

The space and time discretization are now properly established. The following Chapter discusses the obtained results.

Part III

Results

Chapter 7

The step and the wave

In order to prove that the discretization has been properly done, the propagation of a step and of a small sinusoidal perturbation governed by the plasma equations are performed. Those case are simpler than the final model problem, but their good implementation is crucial for the rest of the work.

7.1 The step

The goal of this simulation is to analyze the propagation of the waves given by a step of ions and electrons density. The source terms are neglected here, and so is the electric potential. The equation set is thus given by

$$\begin{aligned}
&\partial_t n_e + \partial_x(n_e u_e) &&= 0\\
&\partial_t n_i + \partial_x(n_i u_i) &&= 0\\
&\partial_t(n_e u_e) + \partial_x(n_e u_e^2 + n_e \varepsilon^{-1}) &&= 0\\
&\partial_t(n_i u_i) + \partial_x(n_i u_i^2 + n_i \kappa) &&= 0
\end{aligned} \tag{7.1}$$

which gives a system of two uncoupled fluids : one made of electrons, the other amde of ions. The electron-to-ion mass ratio is $\varepsilon = 1.36 \times 10^{-5}$ while the temperature ratio is $\kappa = 0.025$. We used cyclic boundary conditions. The step extends over a quarter of the domain, as shown Fig. 7.1. The velocity is set to zero for both ions and electrons everywhere in the domain.

Choosing a discretization of 500 equally spaced elements of order 2, a domain of length 1, the densities obtained for a $CFL = 0.1$ at time $t = 0.000265$ are given for the RK4 method Fig. 7.3 and Fig. 7.4. They are compared to their initial states. The simulation shows that shocks and relaxation waves are formed. After a time of 2.64×10^{-4}, a shock has propagated

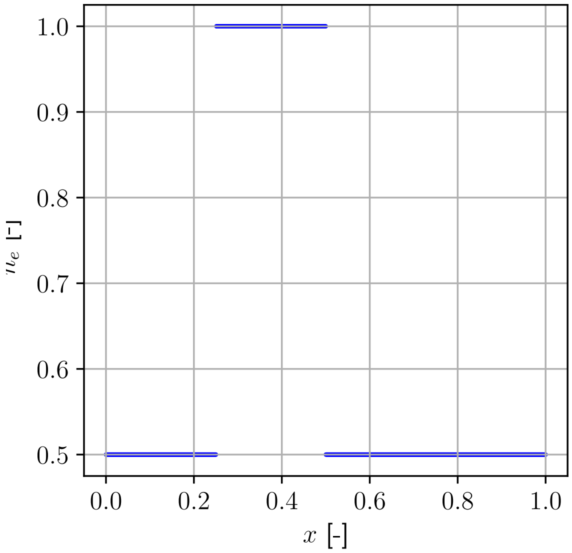


Figure 7.1: Density state of ions and electrons at the initial state $t = 0$. Both initial states are similar so only the electron state is represented.

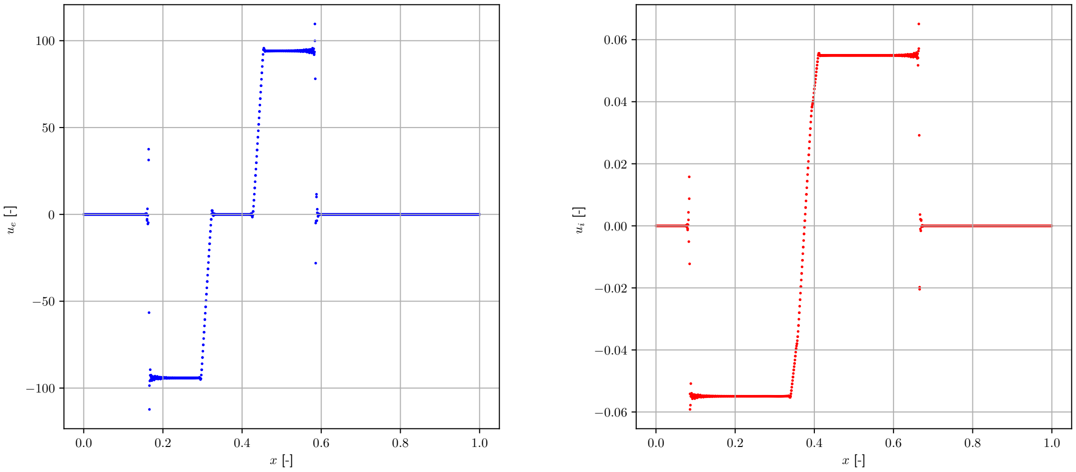
over a distance of approximately 0.1 for the electrons, while for the shock has propagated of about 0.15 in a time 8.78×10^{-1} for the ions. We can make the following observations:

1. The electrons and ions propagates both at subsonic velocity. Indeed, by analyzing the velocities of both species (Fig. 7.2), it is clear that their velocities are lower than their speed of sound:
$$\begin{aligned} 0.06 &\ll \sqrt{\kappa} &\simeq 0.158 \\ 90 &\ll \sqrt{\varepsilon^{-1}} &\simeq 271 \end{aligned} \tag{7.2}$$

2. the shock propagates at a velocity 340 for the electron and 0.170 for the ions, which is close to their respective soundspeed.

3. The system is conservative. Indeed, the mass of electrons and ions have been monitored all along the simulation (the numerical solution was obtained for up to $t = 3$). They remain approximately constant at a value of 6.1 and close to the theoretical mass (which is of about 0.625).

The results show that the system is able to reproduce the main characteristics of a hyperbolic conservative system.



(a) Electron velocity at $t = 2.65 \times 10^{-4}$

(b) Ion velocity at $t = 8.78 \times 10^{-1}$

Figure 7.2: Electronic and ionic velocities at $t = 2.65 \times 10^{-4}$ and $t = 8.78 \times 10^{-1}$ respectively for the step case. Both species are subsonic in this case.

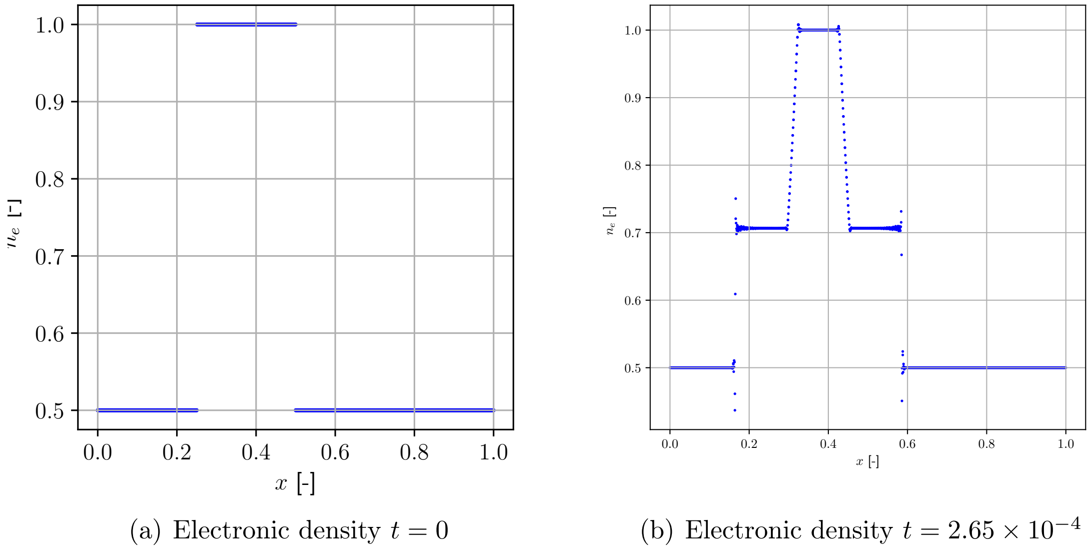
7.2 The wave perturbation

The wave perturbation is a more complex case than the previous one. It takes into account the source terms in the momentum equations. The system to be solved is now the following:

$$\begin{aligned}
&\partial_t n_e + \partial_x(n_e u_e) &&= 0 \\
&\partial_t n_i + \partial_x(n_i u_i) &&= 0 \\
&\partial_t(n_e u_e) + \partial_x(n_e u_e^2 + n_e \varepsilon^{-1}) &&= n_e \varepsilon \partial_x \phi \\
&\partial_t(n_i u_i) + \partial_x(n_i u_i^2 + n_i \kappa) &&= -n_i \partial_x \phi
\end{aligned} \tag{7.3}$$

The goal of this section is to simulate the propagation of a periodic perturbation in a stable plasma. The advantage of this model is that an analytical solution has already been found and can be compared to an analytical result [47, 48]. If the wavelength of the perturbation is set to $k = 2\pi$, one can choose the frequency $\omega = 8.8857268$, and initialize the unknown field using the analytical solution

$$\begin{aligned}
n_e &= 1 + 2.41425 \times 10^{-2} \sin(2\pi x) \\
n_i &= 1 + 2.41425 \times 10^{-2} \sin(2\pi x) \\
n_e u_e &= \left[1 + 10^{-2} \sin(2\pi x)\right] \left[1 + 2.41425 \times 10^{-2} \sin(2\pi x)\right] \\
n_i u_i &= 3.41425 \times 10^{-2} \sin(2\pi x) \left[1 + 2.41425 \times 10^{-2} \sin(2\pi x)\right]
\end{aligned} \tag{7.4}$$



(a) Electronic density $t = 0$

(b) Electronic density $t = 2.65 \times 10^{-4}$

Figure 7.3: Density state of electrons at $t = 0$ and $t = 2.65 \times 10^{-4}$.

The potential is always assumed to be known and to follow the sinusoidal law

$$\phi(x,t) = 2.41421 \times 10^{-2} \sin(2\pi x - 8.8857268t) \tag{7.5}$$

Consequently, the potential gradient expresses at every time step

$$\partial_x \phi(x,t) = 4.82842\pi \times 10^{-2} \cos(2\pi x - 8.8857268t) \tag{7.6}$$

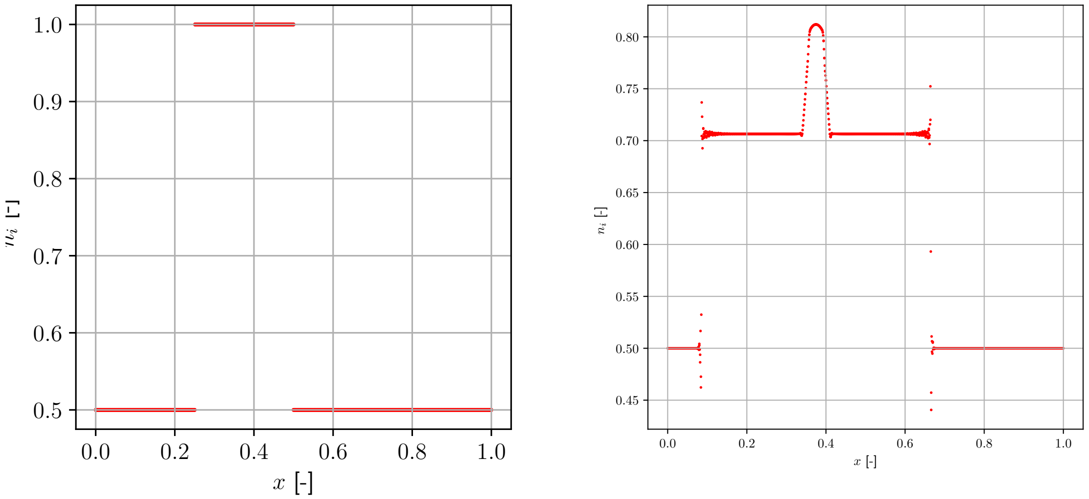
The domain chosen has a length of 1. We choose to discretize it using 500 equally spaced elements of order 2. The boundary conditions are cyclic. We choose $CFL = 0.1$. The plasma is considered thermal, so that $\kappa = 1$. The chosen electron-to-ion mass ratio is $\varepsilon = 10^{-4}$.

A comparison between the RK4 method and the analytical solution is given Fig. 7.5. The numerical results are in strong agreement with the analytical ones, except for the electron momentum. There are several possible causes:

1. The spatial discretization. To assess this, of mesh mesh of higher order or more refined is necessary.

2. The Roe numerical flux, which might give unphysical results in some situations for low-Mach number. Running a simulation with higher order elements should reduce the error in that case.

3. The time discretization. Trying another method, such as the Backward Euler, might



(a) Ionic density $t = 0$

(b) Ionic density $t = 8.78 \times 10^{-1}$

Figure 7.4: Density state of ions at $t = 0$ and $t = 8.78 \times 10^{-1}$.

be helpful.

4. The oscillations might be transient component of the flow that is damped with time. A simulation on a longer time can be run.

5. This could be an instability of the system.

The potential cannot be the cause here as its analytical solution is known and applied.

For the two first causes, a convergence study at $CFL = 0.1$ with order 5 elements has been performed with 50, 100, 200 and 500 elements evaluated at $t = 1$. The results are given Fig. 7.6. No matter how high the spatial resolution of the problem is, the momentum numerical solution presents fast oscillations that do not occur in the analytical solution (Fig. 7.7). These spurious oscillations are not due to the onset of the system, because they persist even for much longer simulation duration.

Similar results have been found using the backward Euler method (Fig. 7.8) with 500 elements of order 2. The only remaining possibility is an instability of the system that we cannot get rid of. However, Alvarez Laguna *et al.* faced a similar problem in their work and corected it by the application of a low-Mach number correction. This also could be a solution to the problem.

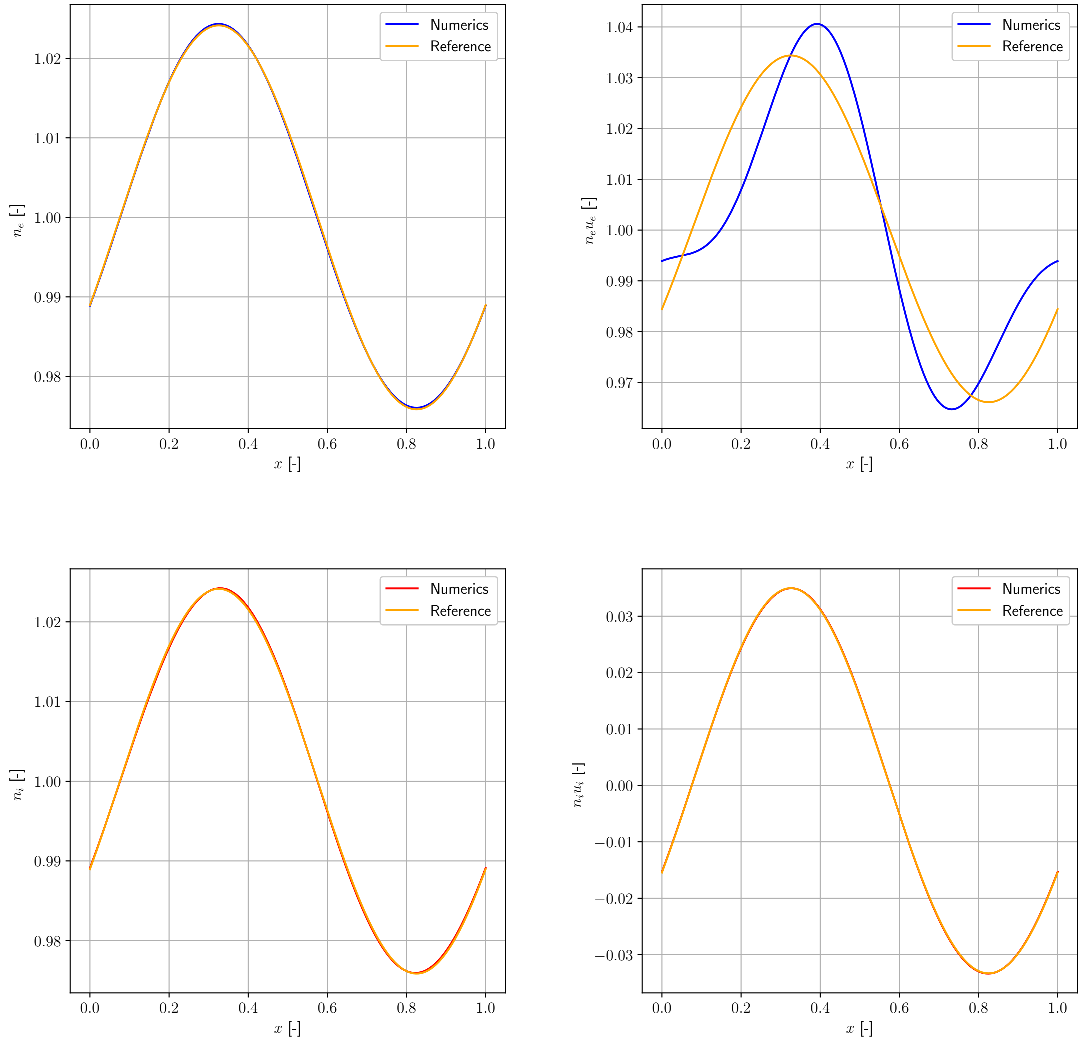


Figure 7.5: Comparison of the reference analytical solution for the perturbation propagation and the numerical solution using a RK4 method with $CFL = 0.1$ and 500 equally spaced elements of order 2. The simulation time is $t = 0.057$. Every quantity numerically computed is in very good agreement with the analytical solution except for the electron momentum.

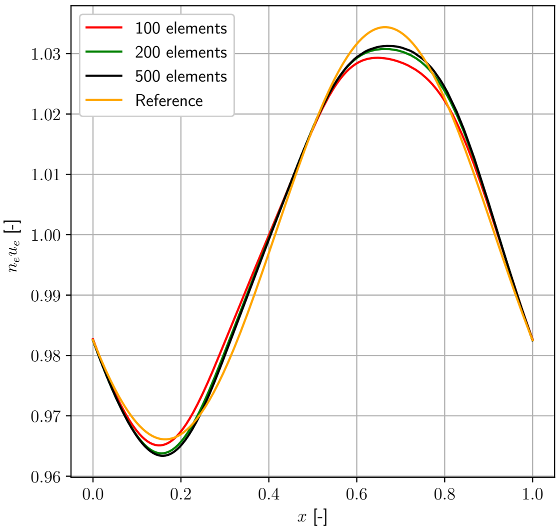


Figure 7.6: Convergence assessment of the electron momentum with the RK4 method and $CFL = 0.1$. The results are a bit more satisfactory than the less resolved case, but there are still discrepancies with the reference solution.

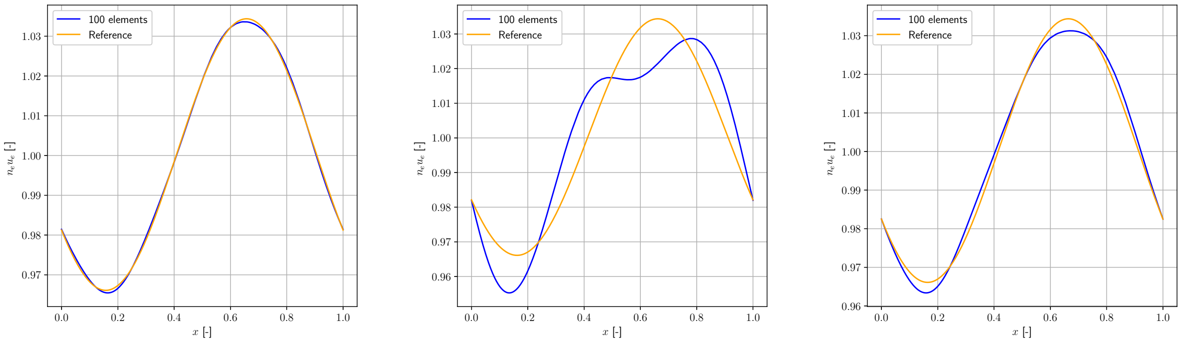


Figure 7.7: Temporal evolution of the momentum with a RK4 method, 500 elements of order 5. Spurious oscillations travel through the solution. Each graph represents the results obtained at $t = 0.999838$, $t = 0.99838$ and $t = 0.994978$.

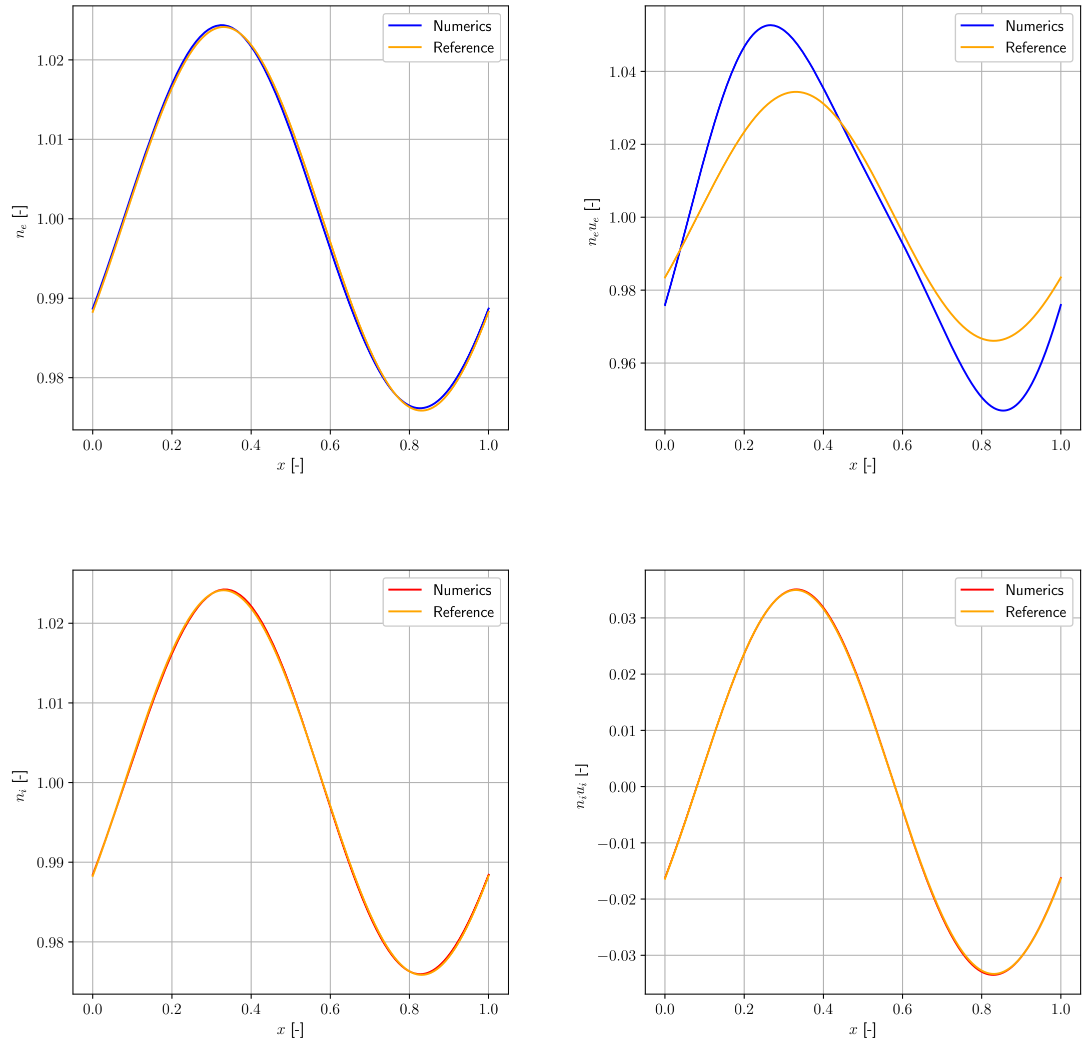


Figure 7.8: Comparison of the reference analytical solution for the perturbation propagation and the numerical solution using a backward Euler method with $CFL = 0.9$ and 500 equally spaced elements of order 2. The simulation time is $t = 0.057$. Every quantity numerically computed is in very good agreement with the analytical solution except for the electron momentum.

Chapter 8

Complete model problem

The goal of this chapter is to give an overview on the numerical problems encountered in the application of the DG-FEM method to a plasma sheath resolution. These problem have to be solved in future works in order to enhance the method.

8.1 Problem set up

The problem consists of a unidimensional plasma isothermal (but with the electronic and ionic temperature that differs) trapped between two electrodes at the same potential. This situation is a model of the plasma trapped inside a cylinder whose surface is an equipotential. The equations to be solved are the complete system given by

$$\begin{aligned}
&\partial_t n_e + \partial_x(n_e u_e) &&= n_e \nu \\
&\partial_t n_i + \partial_x(n_i u_i) &&= n_e \nu \\
&\partial_t(n_e u_e) + \partial_x(n_e u_e^2 + n_e \varepsilon^{-1}) &&= n_e \varepsilon \partial_x \phi \\
&\partial_t(n_i u_i) + \partial_x(n_i u_i^2 + n_i \kappa) &&= -n_i \partial_x \phi
\end{aligned} \tag{2.16}$$

with the boundary conditions described in Section 5.3. The ionization rate is chosen to balance at every time the ion flux that goes out of the system.

$$\nu(t) = \frac{|n_i u_i(L,t)| + |n_i u_i(0,t)|}{\int_0^L n_e(x,t)dx} \tag{5.46}$$

The plasma is assumed come from argon RF discharge, so its electron-to-ion mass ratio is $\varepsilon = 1.36 \times 10^{-5}$ and its ion-to-electron temperature ratio is given by $\kappa = 0.025$. The Debye length of the electron is 0.01. Note that the effective Debye length chosen is 0.035 in order to fit better the results obtained by Alvarez Laguna *et al.*[14]. The approximation of the electric potential is given by

$$\phi(x) = \begin{cases} \phi_0 e^{-\frac{x}{\lambda_D}} & x < L/2 \\ \phi_0 e^{\frac{x-L}{\lambda_D}} & x > L/2 \end{cases} \tag{8.1}$$

with

$$\phi_0 = -\frac{1}{2}\left[1 + \ln\left(\frac{m_i}{2\pi m_e}\right)\right] \tag{8.2}$$

Because the expression of the potential is valid for flows near the steady state condition, the flow is initialized near the steady state found by Alvarez Laguna *et al.*[14]. The expected electron flux at steady state leaving the boundaries is set to 0.4.

8.2 Resolution with RK4 method

At first, a simulation with initial condition

$$\begin{aligned} n_e &= n_i &= 1 \\ n_e u_e &= n_i u_i &= 0 \end{aligned} \tag{8.3}$$

is performed with $\lambda_D = 0.01$. This simulation assesses if the approximation of the potential can be used even far from the equilibrium with a value of λ close to the electronic Debye length. The resolution is done with the RK4 method using a mesh with a refinement coefficient $\gamma = 1.01$ and 500 elements of order 2. The CFL worth 0.46. The results obtained are far from the expected behaviour of the system (Fig. 8.1). They are compared to a reference solution which solves a problem with a third order TVD finite volume scheme [14]. The electronic density diminishes in time while it should converge towards a constant profile. The electron momentum oscillates with time and behaves like a damping wave. On the other hand, the ionic and momentum profile tend towards a constant state which is very far from the expected steady state.

In order to have a better overview of the situation, the TVD solution has been plugged in the solver and the residual has been assessed. The residual chosen was the following

$$\mathcal{M}_{ij} J_k^G \partial_t u_j^n - \mathcal{K}_{ij} F_j^n + \mathcal{F}_i = 0 \tag{8.4}$$

with $\partial_t u_j^n$ chosen as the backward Euler method first order accurate. We proceeded as follow:

1. because the reference solution is obtained through a finite volume method, only the value of the unknowns is given at the center of the cells,

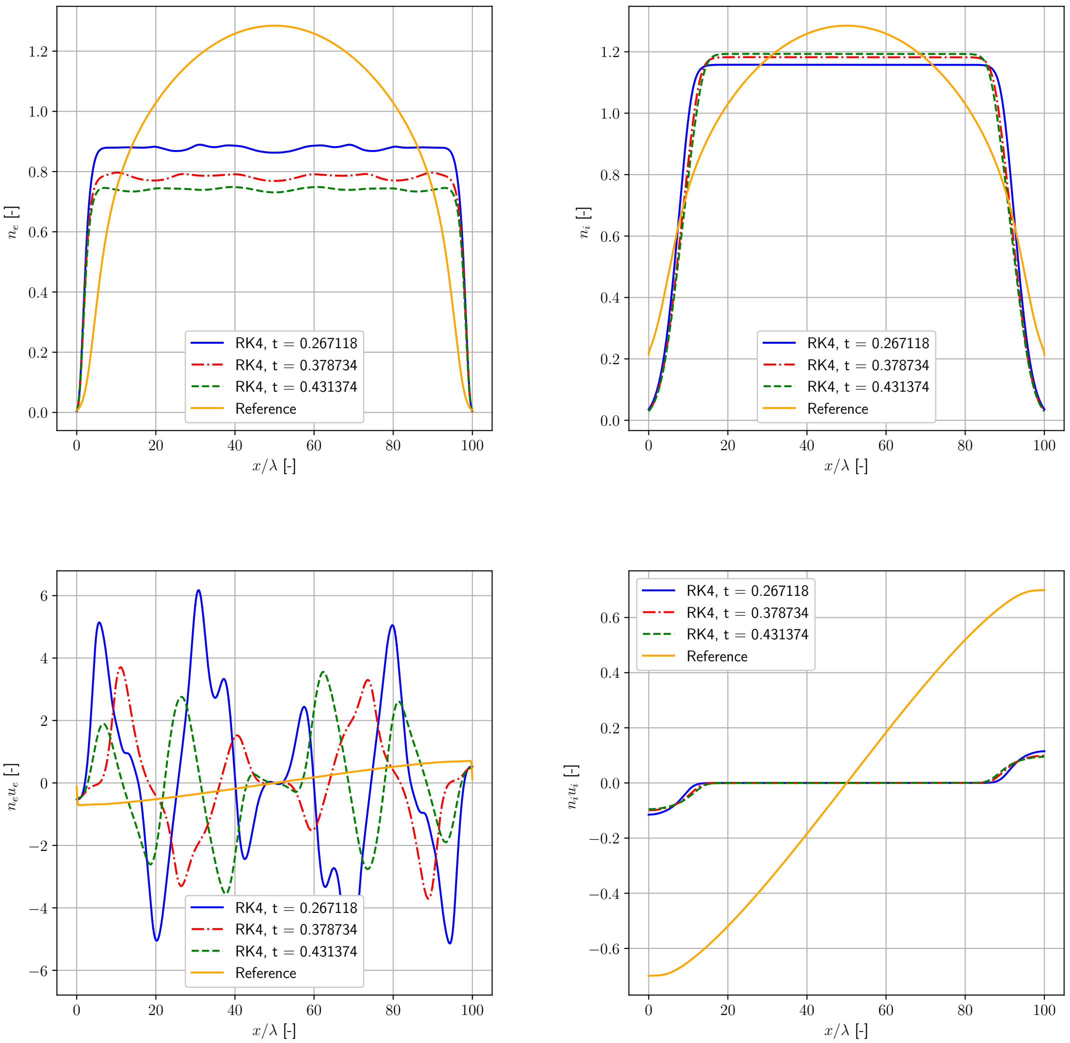


Figure 8.1: Comparison of the behaviour of the plasma numerical solution using DG-FEM and a RK4 method at several times of the simulation with a third order TVD scheme using 500 mesh points. The situation represented is a plasma trapped between two equipotential walls, with electron-to-ion mass ration $\varepsilon = 1.36 \times 10^{-5}$ and ion-to-electron temperature ratio $\kappa = 0.025$. The reference solution is at steady state. The spatial discretization used for the RK4 method consists of 500 elements of order 2 with refinement coefficient $\gamma = 1.01$ and $CFL = 0.46$. The systems evolves far from its expected steady state behaviour.

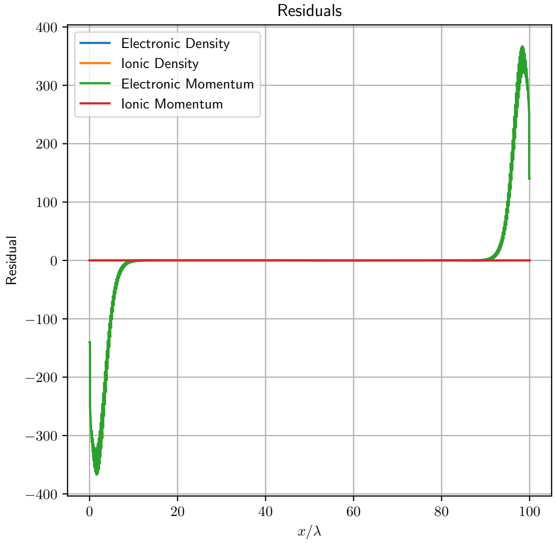


Figure 8.2: Residuals obtained when the reference solution is plugged in the DG-FEM solver and order 1 elements are used. It is clear that the electrons are likely to give some problems, since the residual of the electron momentum is very high near the boundaries.

2. we then set a mesh of order one whose nodes correpond to the center of the FV scheme cells,

3. the central values obtained using the FV scheme are assigned to the nodes of the mesh,

4. the value at the nodes corresponding to the boundaries (to the electrodes) have been extrapolated from the values of the closest nodes.

The residuals obtained for all quantities are given Fig. 8.2. The residuals are extremely good, except for the electron momentum residual, which is extremely high in the near wall regions. This might be due to the mesh that is not refined enough in the near wall region. Unfortunately, we had not the opportunity to test this hypothesis, since we did not have access to higher resolved data. The residual was computed for a cubic spline interpolation of the results of the TVD scheme, but gave spurious results.

A simulation with initial condition corresponding to the reference solution, with 1000 elements of order 3 and a refinement coefficient $\gamma = 1.01$ and a $CFL = 0.1$ is tested. All quantities are maintained by the DG-FEM solver, except for the electron momentum which is not stable at all (Fig. 8.3).

Finally, a test is performed using the approximate potential, with $\lambda = 0.035$ this time. There are 500 elements of order 2 and $CFL = 0.7$. The initial condition is taken as an

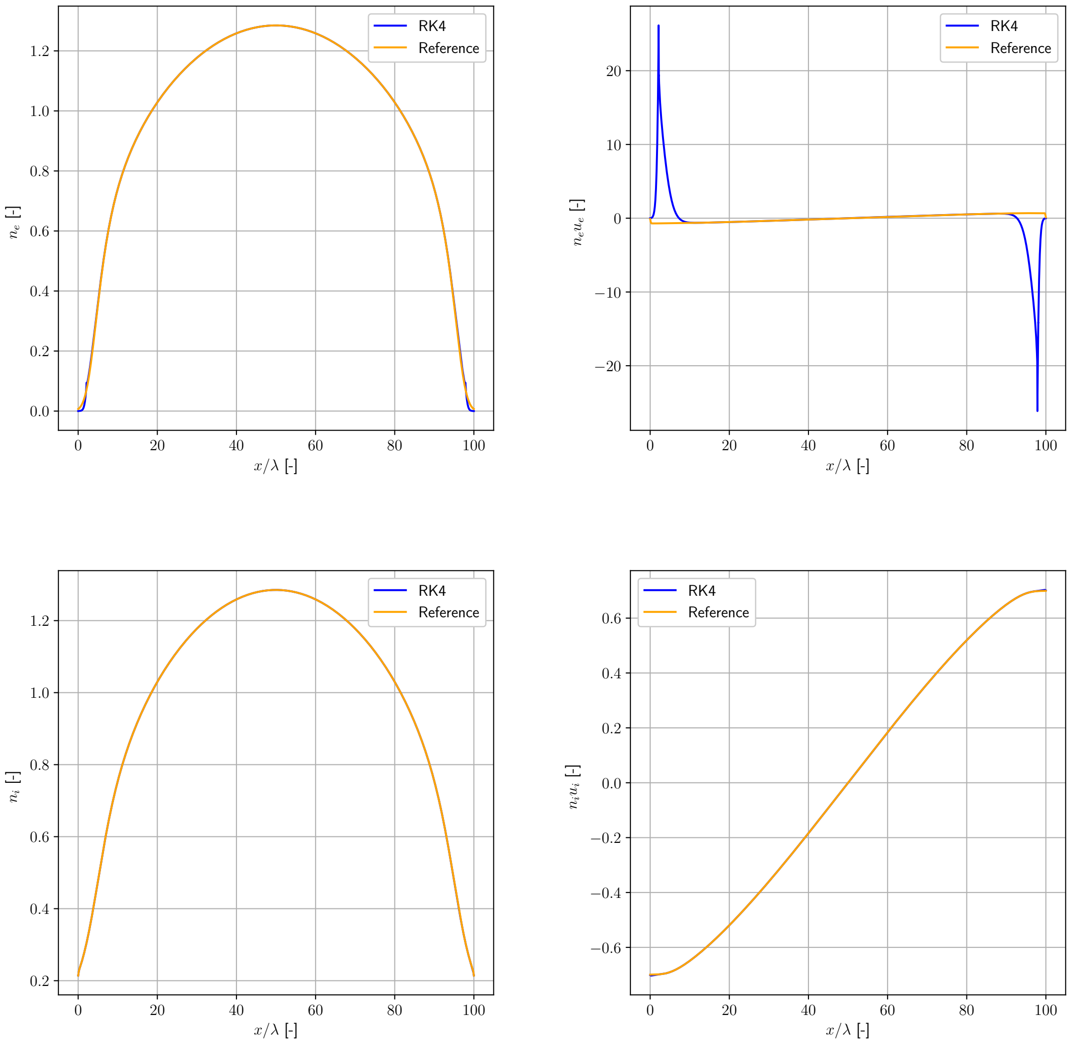


Figure 8.3: Comparison of the reference analytical solution for the perturbation propagation and the numerical solution using a RK4 method with $CFL = 0.4$ and 1000 elements of order 3 with refinement coefficient $\gamma = 1.01$. The simulation time is $t = 0.0004$. The initial condition was the reference solution. It is clear that the system does not manage to keep the electronic momentum, while all other quantities can be maintained.

interpolation of the steady state of Alvarez Laguna *et al.*. The results are compared to the reference steady state (Fig. 8.4). While still being far from the sought steady state, the numerical solution is in general in better agreement with the expected results than the previous simulations. Especially for the electron density which is almost a perfect fit in the near wall region. However, there are still strong oscillations of the electron momentum.

8.3 Resolution with the Backward Euler method

The backward Euler method is non avantageous compared to the RK4 method. Indeed, while cumulating the same disadvantage in the solutions obtained as the RK4 method, its computational cost is much higher. Even if it allows to use a CFL twice as large as the CFL used for the RK4 method, it still takes about 8 times the resolution time of the RK4 method.

8.4 Conclusions

The numerical scheme implemented as such is not able to reproduce correctly the behaviour of the plasma sheath. The main problem comes from the electron momentum, which is subjected to spurious oscillations. The cause is yet undetermined, but several hypothesis can be made.

1. The electron momentum is very sensitive to the electric potential, and the use of the approximate potential might lead to spurious solutions. Solving the Poisson equation might help to find the actual solution of the problem.

2. The oscillations might be instabilities of the system itself. However, Alvarez Laguna *et al.* have also experienced these oscillations with their A-P scheme [14]. They corrected the problem by bringing a correction factor for the low-Mach numbers to the electron equation.

3. The reference solution plugged in the solver is not completely in accordance with physical results in the near wall region for the electron momentum. Indeed, near the wall, the ion and electron momentum differ from each other, while they should be equal. The error is of the order of 50%.

4. The boundary conditions could also play an important role. They might require more work and be better defined.

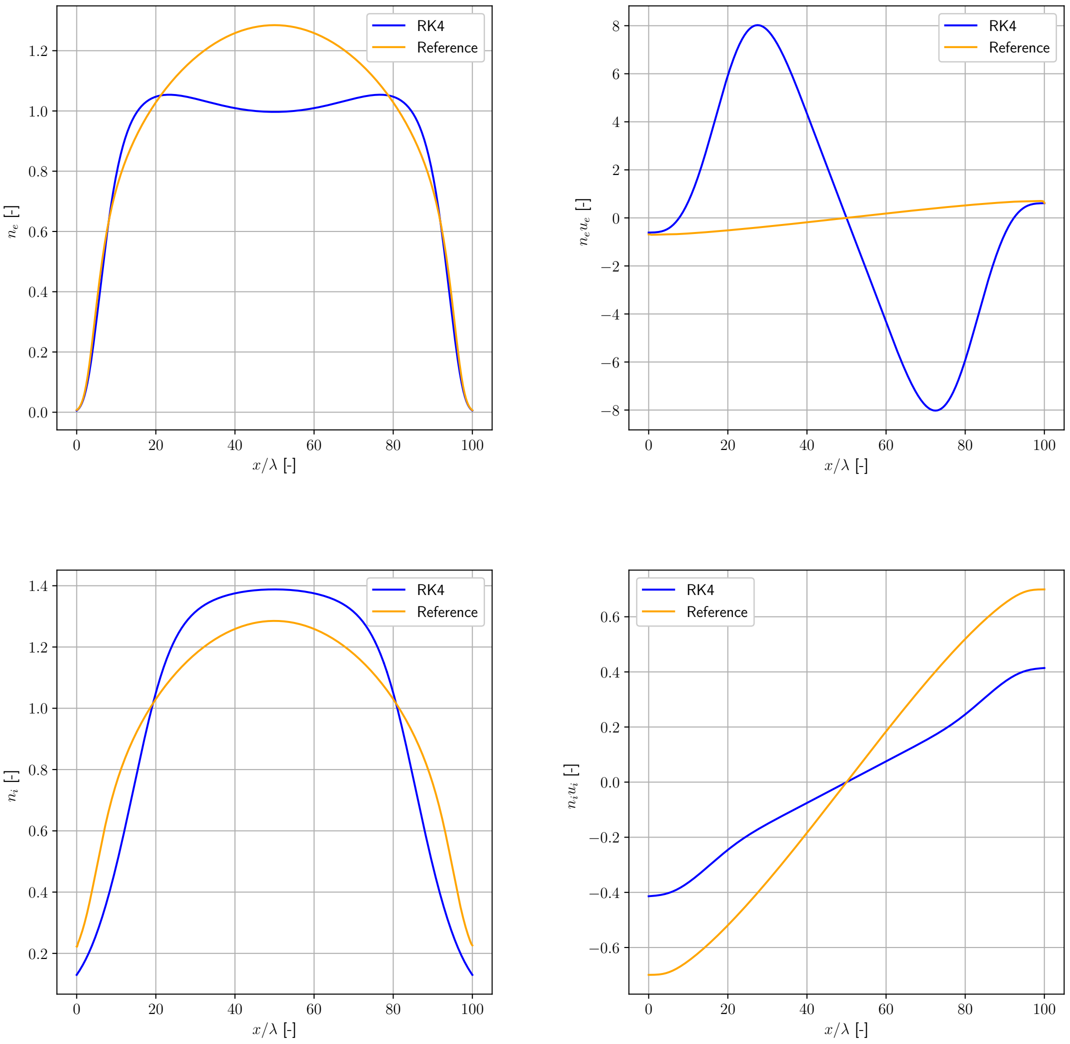


Figure 8.4: Comparison of the reference analytical solution for the perturbation propagation and the numerical solution using a RK4 method with $CFL = 0.7$ and 500 elements of order 2 with refinement coefficient $\gamma = 1.01$. The simulation time is $t = 0.548390$. The approximate potential has been used here. The initial condition was the reference solution.

The previous statement should be taken with care, these are possible causes only and food for thought for furture developments.

Conclusion and Future work

In this work, we have modeled a plasma trapped in a cylinder with equipotential walls. We have presented the physics that lies behind such a problem and the numerical difficulties arising from the formation of a sheath. Then, the discontinuous Galerkin finite element method (DG-FEM) has been applied to the set of equations. Two temporal schemes have been shown : a fourth order Runge Kutta method and a second order accurate backward Euler method. Both had adaptative time steps.

These methods have been tested on three practical cases: the step evolution, the perturbation wave propagation and the complete model problem. While the two first applications worked relatively well, the last one failed. The principal failure of the simulation is due to the electron momentum which presents spurious oscillations. Several possible causes :

1. The approximate expression of the potential used, which is not consistent with the rest of the system and should be solved properly, using FEM for example.

2. Boundary conditions, which might not be adapted.

3. The Roe numerical flux might bring instabilities at low Mach number as in the work of Alvarez Laguna *et al.*. A low-Mach number correction might bring some stability to the system.

Those statements are possible cause of the failure of the system. They should lead the future developments of the method.

No matter how hypothetical the previous causes are, the Poisson equation has to be solved at one point or anoter. It could be solved either by simple FEM, or by DG-FEM. The first method is more straightforward than the second one, but the DG-FEM on a Poisson can be implemented by using penalty terms [49]. The idea is then to

1. start from an initial field with constant fields,

2. solve the hyperbolic system,

3. plug the densities obtained through the hyperbolic system in the Poisson equation,

4. solve the Poisson equation for the electric potentential,

5. starting from the densities computed previously and the electric potential, create a new initial condition state and restart the algorithm until the electric potential and the quantities from the hyperbolic system are constant.

This resolution is crucial and might solve by itself all the previous mentionned problems, given the fact that the electron momentum is very sensitive to the electric potential.

Another improvement consists of bringing a correction to the Roe numerical flux for the electrons. Since the ions are supersonic, their Roe flux has no need to be corrected. Liou [50] proposed a factor depending on the mach number that multiplies the dissipative part of the numerical flux.

The system can also be solved by decoupling the acoustic and transport phenomenae of electrons as it has been done in [14]. The main idea behind the resolution is to solve two separate systems: one that accounts for the fast dynamics (acoustic one), the other for the slow dynamics (transport one). The solution procedes in two steps: one that first solve the fast dynamics to give a state at t^*, then the other that solves from the state at t^* to the following time step. This approach tries to reproduce the fast response of acoustic dynamics and the slow adaptation of the transport phenomenae, and has been proven to be asymptotic preserving [14]. The advantages of such method is that is is able to solve the system without resolving the Debye length.

Appendices

Appendix A

Implementation

The implementation of the program has been performed using C++ language. The Eigen [51] and openMP [52] have been used in order to ease the matrix inversion necessary - in particular in Eq. 6.39 - and to accelerate the resolution of the problem. The complete code can be found on *GitHub* at the address `https://github.com/nicocorth/1D-DG-FEM-Plasma-Solver/tree/Restructuration` after asking for access to the author.

A.1 Code organization

The code is organized around two classes : one related to the information of the mesh - called *Mesh* -, and another related to the information for the resolution of the problem, called *ConservationEquation*.

A.1.1 *Mesh* class

The *Mesh* structure contains all information related to the mesh, *i.e.*

1. The mass and stiffness matrices,
2. The position of each node,
3. The position and weight of the Gauss points,
4. The mass and stiffness matrices of the reference element.
5. The value of the shape functions at the gauss point.
6. Some member functions that allow to compute all information listed above.

The information of the *Mesh* is used in the *ConservationEquation* structure. In fact, the latter contains a *Mesh* class.

A.1.2 *ConservationEquation* class

In more detail, the *ConservationEquation* class contains the following information:

1. A *Mesh* structure containing the information about the mesh required for the resolution,
2. The electron mass,
3. The ionization rate,
4. The value at each node of each unknown for the current time step, the previous time step and even the time step before,
5. The mass and stiffness matrices of the reference element,
6. The CFL number at which the simulation is performed,
7. The current and precedent time steps used for the discretization, along with the final simulation time, the step at which the simulation is and the time corresponding to it, and the interval of steps at which the solver saves the results,
8. The boolean variable that is *true* if the steady state of the system is reached and *false* if not. The threshold of this variable is dictated by the maximum value taken by the temporal term,
9. The functions that allows the program to solve the plasma equations using the method described in this work: computation of the residual and each term of the residual along with their jacobian for both the RK4 and the Backward Euler adapatative methods.

A.2 Matrix inversion using Eigen

Eq. 6.39 requires the inversion of the jacobian. In order to facilitate the computation, the Eigen library [51] has been used. This library allows an easy and efficient way to inverse large, sparse matrices. The inversion used here is based on a LU decomposition [53].

The inverse jacobian of the system is non trivially found. This method allowed to greatly simplify the implementation.

A.3 Parallelization using OpenMP

Because the system involves large electron propagation velocities, the time step required is highly constrained, especially in the Runge-Kutta case. More over, since the gradients in the near wall regions are steep, the smallest element in the near-wall region is very small, which is even more restrictive. The non dimensional time step implied are of the order of 5×10^{-7} for simulations that last more than one non-dimensional time unit.

The solution found to accelerate the resolution is to parallel the code using OpenMp [52]. The parallelization intervene at the level of the residual and the residual jacobian.

Appendix B

Elements of Plasma theory

Some basis of plasma physics are presented here. This is far from being a complete presentation of plasmas phenomena, but those important for the comprehension of the present works are presented. In particular, the distributions functions and the plasma sheilding (leading to the plasma sheath) are discussed.

B.1 Boltzmann's equation derivation

The Boltzmann distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ represents the density number of particles that are at the macroscopic position $\boldsymbol{x}$ and that have a velocity $\boldsymbol{v}$ at instant t. The time evolution of such quantity is given by

$$\frac{df}{dt}(\boldsymbol{x}, \boldsymbol{v}, t) = S \tag{B.1}$$

where S represents a source of variation of f of any kind (*e.g.* collisions or ionization). This equation can be expanded in (by the chain rule)

$$\partial_t f + \partial_t \boldsymbol{x} \nabla_{\boldsymbol{x}} f + \partial_t \boldsymbol{v} \nabla_{\boldsymbol{v}} f = S \tag{B.2}$$

By definition, $\partial_t \boldsymbol{x} = \boldsymbol{v}$ and the temporal derivative of the velocity is an acceleration, which is nothing else than the total force $\boldsymbol{F}$ undergone by the particle divided by the mass m of a particle. Thus,

$$\partial_t f + \boldsymbol{v} \nabla_{\boldsymbol{x}} f + \frac{\boldsymbol{F}}{m} \nabla_{\boldsymbol{v}} f = S \tag{B.3}$$

If the particle has a charge q and is only submitted to an electric field $\boldsymbol{E}$, then $\boldsymbol{F} = q\boldsymbol{E}$ and

$$\partial_t f + \boldsymbol{v} \nabla_{\boldsymbol{x}} f + \frac{q\boldsymbol{E}}{m} \nabla_{\boldsymbol{v}} f = S \tag{B.4}$$

B.2 Plasma shielding

Plasma shielding is a unique feature. From the collective behaviour of their particles, plasmas are able to reduce the electric fields very efficiently. Let us consider a plasma with an ion of charge $+e$ and its electrons of charge $-e$, $e = 1.6 \times 10^{-19}$C. At equilibrium, the electrons and ions densities obey an exponential law, function of the electric potential $\phi(\boldsymbol{r})$ (in radial coordinates)

$$\begin{aligned} n_e(\boldsymbol{r}) &= n_{e,0} e^{\frac{e\phi(\boldsymbol{x})}{k_B T_e}} \\ n_i(\boldsymbol{r}) &= n_{i,0} e^{-\frac{e\phi(\boldsymbol{x})}{k_B T_i}} \end{aligned} \tag{B.5}$$

where T_e and T_i are the electronic and ionic temperatures, $n_{e,0}$ and $n_{i,0}$ the equilibrium values of the electronic and ionic densities. In case the perturbation potential is small compared to the plasma thermal energy, the densities can be expanded in

$$\begin{aligned} n_e(\boldsymbol{r}) &\simeq n_{e,0}\left(1 + \frac{e\phi(\boldsymbol{x})}{k_B T_e}\right) \\ n_i(\boldsymbol{r}) &\simeq n_{i,0}\left(1 - \frac{e\phi(\boldsymbol{x})}{k_B T_i}\right) \end{aligned} \tag{B.6}$$

If the electric potential is assumed to be created by a charge density Q at $\boldsymbol{r} = \boldsymbol{0}$, the Poisson equation reads

$$\Delta\phi(\boldsymbol{r}) = -\frac{Q\delta(\boldsymbol{r}) - e n_{e,0}\frac{e\phi}{k_B T_e} - e n_{i,0}\frac{e\phi}{k_B T_i}}{\epsilon_0} \tag{B.7}$$

The solution to this equation is found to be

$$\phi(\boldsymbol{r}) = \frac{Q}{4\pi\epsilon_0 \boldsymbol{r}^2} e^{-r/\lambda_D} \tag{B.8}$$

and is the famous Debye-Hückel potential [54]. The parameter λ_D is called the Debye length and reads

$$\frac{1}{\lambda_D^2} = \frac{e^2 n_{e,0}}{\epsilon_0 k_B T_e} + \frac{e^2 n_{i,0}}{\epsilon_0 k_B T_i} \tag{B.9}$$

The potential created by the charge density Q is felt only over a distance λ_D. This distance depends on the ions and electrons thermal energy. This shielding is at the basis of the plasma sheath that appears in the near-wall region : the electrode collects the electrons and creates a depleted region. This region extends about several Debye lengths from the wall. Farther away from the boundary, the gas become quasi-neutral again : the potential has been shielded. Note that the developments of this Section can be found in [13].

Appendix C

Roe numerical flux

We present the derivation of the Roe matrix for the Eulerian fluids, along with the Roe-averaged velocities. We also show the upwind nature of the Roe numerical flux. This is the ability for the flux to select the information that comes from upwind regions when the flow is supersonic.

C.1 Roe-averaged quantities

In this Section, we determine the p state variable that is used for assessing the Roe numerical flux. We will consider here a single Eulerian fluid since the developments extends very easily to multi-fluid eulerian systems. Roe was the first to present these results [40].

The system of mass and momentum conservation for a eulerian isothermal fluid reads (considering there are no sources)

$$\begin{aligned} &\partial_t \rho + \partial_x(\rho u) &&= 0 \\ &\partial_t(\rho u) + \partial_x \left[\frac{(\rho u)^2}{\rho} + a^2 \rho \right] &&= 0 \end{aligned} \tag{C.1}$$

where ρ, u and a represent respectively the density, velocity and sound speed of the fluid. ρu is the momentum. This equation can be rewritten in matrix form as

$$\partial_t \begin{pmatrix} \rho \\ \rho u \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ a^2 - \frac{(\rho u)^2}{\rho^2} & 2\frac{\rho u}{\rho} \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ \rho u \end{pmatrix} = \mathbf{0} \tag{C.2}$$

Let us consider now the variables

$$\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \sqrt{\rho} \\ \frac{\rho u}{\sqrt{\rho}} \end{pmatrix} \tag{C.3}$$

Since $\rho > 0$ (a density cannot be negative or zero), it can be assumed that z_1 and z_2 are always well defined. The link between this new set of variabels and the previous is

$$\begin{pmatrix} \rho \\ \rho u \end{pmatrix} = \begin{pmatrix} z_1^2 \\ z_1 z_2 \end{pmatrix} \tag{C.4}$$

The previous equation states that the initial state variable ρ and ρu are quadratic functions of z_1 and z_2. It is easy to verify that

$$\boldsymbol{u}_L - \boldsymbol{u}_R = \begin{pmatrix} 2\bar{z}_1 & 0 \\ \bar{z}_2 & \bar{z}_1 \end{pmatrix} (\boldsymbol{z}_L - \boldsymbol{z}_R) \tag{C.5}$$

$$\boldsymbol{F}(\boldsymbol{u}_L) - \boldsymbol{F}(\boldsymbol{u}_R) = \begin{pmatrix} \bar{z}_2 & \bar{z}_1 \\ 2a^2\bar{z}_1 & 2\bar{z}_2 \end{pmatrix} (\boldsymbol{z}_L - \boldsymbol{z}_R) \tag{C.6}$$

where $\bar{z}_i = \frac{1}{2}(z_{i,L} + z_{i,R})$, $\boldsymbol{u} = \begin{pmatrix} \rho & \rho u \end{pmatrix}$ and $\boldsymbol{F}$ represents the flux vector. The conservation condition required by Roe for the construction of the numerical flux states that

$$\boldsymbol{A}_h = \begin{pmatrix} \bar{z}_2 & \bar{z}_1 \\ 2a^2\bar{z}_1 & 2\bar{z}_2 \end{pmatrix} \begin{pmatrix} 2\bar{z}_1 & 0 \\ \bar{z}_2 & \bar{z}_1 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & 1 \\ a^2 - \frac{\bar{z}_2^2}{\bar{z}_1^2} & 2\frac{\bar{z}_2^2}{\bar{z}_1} \end{pmatrix} \tag{C.7}$$

One can also verify that

$$\frac{\bar{z}_2^2}{\bar{z}_1} = \frac{\frac{\rho_L u_L}{\sqrt{\rho_L}} + \frac{\rho_R u_R}{\sqrt{\rho_R}}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{C.8}$$

and

$$\frac{\bar{z}_2^2}{\bar{z}_1^2} = \left(\frac{\frac{\rho_L u_L}{\sqrt{\rho_L}} + \frac{\rho_R u_R}{\sqrt{\rho_R}}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \right)^2 \tag{C.9}$$

are respectively the definitions of the Roe-averaged velocity and velocity squared.

C.2 Upwind nature of the Roe numerical flux

The upwind nature for the Roe numerical flux is shown for the electron density and momentum in the case where $\bar{\lambda}_0 > 0$ is proven. The demonstration for all other cases are very similar and are not given here.

Let us recall that

$$\begin{aligned}\bar{\lambda}_0 &= \bar{u}_e - \sqrt{\varepsilon^{-1}} \\ \bar{\lambda}_1 &= \bar{u}_e + \sqrt{\varepsilon^{-1}}\end{aligned} \tag{C.10}$$

As a consequence, $\bar{\lambda}_0 > 0 \implies \bar{\lambda}_1 > 0$ and the diffusive part of the numerical flux for the electron density expresses

$$\begin{aligned}-\frac{a_0|\bar{\lambda}_0| + a_1|\bar{\lambda}_1|}{2} &= -\frac{\left[(n_{e,R} - n_{e,L})\bar{\lambda}_1 - (n_e u_{e,R} - n_e u_{e,L})\right]\bar{\lambda}_0}{4\sqrt{\varepsilon^{-1}}} \\ &+ \frac{\left[(n_{e,R} - n_{e,L})\bar{\lambda}_0 - (n_e u_{e,R} - n_e u_{e,L})\right]\bar{\lambda}_1}{4\sqrt{\varepsilon^{-1}}} \\ &= \frac{(n_e u_{e,R} - n_e u_{e,L})(\bar{\lambda}_0 - \bar{\lambda}_1)}{4\sqrt{\varepsilon^{-1}}} \\ &= -\frac{n_e u_{e,R} - n_e u_{e,L}}{2}\end{aligned} \tag{C.11}$$

Knowing that the average part of the numerical flux reads

$$\frac{F_{n_e,L} + F_{n_e,R}}{2} = \frac{n_e u_{e,L} + n_e u_{e,R}}{2} \tag{C.12}$$

It is obvious that the total numerical flux is in this case

$$F^*_{n_e} = n_e u_{e,L} \tag{C.13}$$

which is the value of the physical flux upwind with respect to the propagation of information.

Concerning the numerical flux associated to the electron momentum, the demonstration is less trivial, but is performed the same way. The diffusive part of this numerical flux is given by

$$\begin{aligned}-\frac{a_0|\bar{\lambda}_0|\bar{\lambda}_0 + a_1|\bar{\lambda}_1|\bar{\lambda}_1}{2} &= -\frac{\left[(n_{e,R} - n_{e,L})\bar{\lambda}_1 - (n_e u_{e,R} - n_e u_{e,L})\right]\bar{\lambda}_0^2}{4\sqrt{\varepsilon^{-1}}} \\ &+ \frac{\left[(n_{e,R} - n_{e,L})\bar{\lambda}_0 - (n_e u_{e,R} - n_e u_{e,L})\right]\bar{\lambda}_1^2}{4\sqrt{\varepsilon^{-1}}} \\ &= \frac{(n_{e,R} - n_{e,L})\bar{\lambda}_0\bar{\lambda}_1(\bar{\lambda}_1 - \bar{\lambda}_0) + (n_e u_{e,R} - n_e u_{e,L})(\bar{\lambda}_0^2 - \bar{\lambda}_1^2)}{4\sqrt{\varepsilon^{-1}}} \\ &= \frac{(n_{e,R} - n_{e,L})\bar{\lambda}_0\bar{\lambda}_1 + (n_e u_{e,R} - n_e u_{e,L})(\bar{\lambda}_0 + \bar{\lambda}_1)}{2} \\ &= \frac{(n_{e,R} - n_{e,L})(\bar{u}_e^2 - \varepsilon^{-1})}{2} + (n_e u_{e,R} - n_e u_{e,L})\bar{u}_e\end{aligned} \tag{C.14}$$

Recalling that the Roe averaged velocity expresses

$$\bar{u}_e = \frac{\frac{n_e u_{e,L}}{\sqrt{n_{e,L}}} + \frac{n_e u_{e,R}}{\sqrt{n_{e,R}}}}{\sqrt{n_{e,L}} + \sqrt{n_{e,R}}} \tag{C.15}$$

so that

$$\bar{u}_e^2 = \frac{\frac{(n_e u_{e,L})^2}{n_{e,L}} + \frac{(n_e u_{e,R})^2}{n_{e,R}} + 2\frac{(n_e u_{e,L})(n_e u_{e,R})}{\sqrt{n_{e,R}}\sqrt{n_{e,L}}}}{(\sqrt{n_{e,L}} + \sqrt{n_{e,R}})^2} \tag{C.16}$$

and the previous expression of the diffusive part of the numerical flux reads

$$\begin{aligned}
&\frac{-(n_{e,R} - n_{e,L})\varepsilon^{-1}}{2} \\
&+ \frac{\frac{(n_e u_{e,L})^2 n_{e,R}}{n_{e,L}} + (n_e u_{e,R})^2 + 2\frac{(n_e u_{e,L})(n_e u_{e,R})\sqrt{n_{e,R}}}{\sqrt{n_{e,L}}}}{2(\sqrt{n_{e,L}} + \sqrt{n_{e,R}})^2} \\
&+ \frac{\frac{(n_e u_{e,R})^2 n_{e,L}}{n_{e,R}} + (n_e u_{e,L})^2 + 2\frac{(n_e u_{e,L})(n_e u_{e,R})\sqrt{n_{e,L}}}{\sqrt{n_{e,R}}}}{2(\sqrt{n_{e,L}} + \sqrt{n_{e,R}})^2} \\
&- \frac{\frac{n_e u_{e,L} n_{e,R}}{\sqrt{n_{e,L}}} + n_e u_{e,R}\sqrt{n_{e,R}}}{\sqrt{n_{e,L}} + \sqrt{n_{e,R}}} - \frac{n_e u_{e,L}\sqrt{n_{e,L}} + \frac{n_e u_{e,R}\sqrt{n_{e,L}}}{\sqrt{n_{e,R}}}}{\sqrt{n_{e,L}} + \sqrt{n_{e,R}}}
\end{aligned} \tag{C.17}$$

The development of this expression directly leads to

$$-\frac{a_0|\bar{\lambda}_0|\bar{\lambda}_0 + a_1|\bar{\lambda}_1|\bar{\lambda}_1}{2} = -\frac{\frac{(n_e u_{e,R})^2}{n_{e,R}} + \varepsilon^{-1} n_{e,R} - \frac{(n_e u_{e,L})^2}{n_{e,L}} - \varepsilon^{-1} n_{e,L}}{2} \tag{C.18}$$

so that the numerical flux corresponds to the left physical flux, *i.e.* the upwind one.

Appendix D

Adaptative Backward Euler method second order accurate

The derivation of the adaptative backward Euler method second order accurate is developed here. Let us consider a variable u. The evaluation of u at time t_n is noted u_n. One has the following Taylor developments:

$$\begin{aligned}
u_{n-1} &= u_n && - \Delta t_n \partial_t u_n && + \frac{1}{2}\Delta t_n^2 \partial_{tt} u_n && + \mathcal{O}(\Delta t_n^3) \\
u_{n-2} &= u_n && - (\Delta t_n + \Delta t_{n-1})\partial_t u_n && + \frac{1}{2}(\Delta t_n + \Delta t_{n-1})^2 \partial_{tt} u_n && + \mathcal{O}(\Delta t_n^3, \Delta t_{n-1}^3)
\end{aligned} \tag{D.1}$$

where Δt_n is the current timestep and Δt_{n-1} is the previous time step. Multiplying the first and second equations of Eq. D.1 by a and b respectively and adding both equations, one obtains

$$\begin{aligned}
a u_{n-1} + b u_{n-2} &= (a+b)u_n \\
&+ [-a\Delta t_n - b(\Delta t_n + \Delta t_{n-1})]\partial_t u_n \\
&+ \left[\frac{a}{2}\Delta t_n^2 + \frac{b}{2}(\Delta t_n + \Delta t_{n-1})^2\right]\partial_{tt} u_n
\end{aligned} \tag{D.2}$$

This leaves the following system of equation to solve for a and b:

$$\begin{cases} a\Delta t_n^2 + b(\Delta t_n + \Delta t_{n-1})^2 &= 0 \\ -a\Delta t_n - b(\Delta t_n + \Delta t_{n-1}) &= 1 \end{cases} \tag{D.3}$$

which gives the following value for a and b:

$$a = -\frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_n \Delta t_{n-1}} \quad b = \frac{\Delta t_n}{\Delta t_{n-1}(\Delta t_n + \Delta t_{n-1})} \tag{D.4}$$

The final expression is given by

$$\begin{aligned}\partial_t u_n \quad &\simeq \left[\frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_n \Delta t_{n-1}} - \frac{\Delta t_n}{\Delta t_{n-1}(\Delta t_n + \Delta t_{n-1})}\right] u_n \\ &- \frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_n \Delta t_{n-1}} u_{n-1} \\ &+ \frac{\Delta t_n}{\Delta t_{n-1}(\Delta t_n + \Delta t_{n-1})} u_{n-2}\end{aligned} \tag{D.5}$$

Bibliography

[1] A. Bogaerts and L. L. Alves. Special issue on numerical modelling of low-temperature plasmas for various applications - part ii: Research papers on numerical modelling for various plasma applications. *Plasma Processes and Polymers*, 14(4-5):1790041, 2017.

[2] L. L. Alves and A. Bogaerts. Special issue on numerical modelling of low-temperature plasmas for various applications - part i: Review and tutorial papers on numerical modelling approaches. *Plasma Processes and Polymers*, 14(1-2):1690011, 2017.

[3] I. D. Boyd K. Hara and V. I. Kolobov. One-dimensional hybrid-directy kinetic simulation of the discharge plasma in hall thrusters. *Physics of Plasmas*, 19:113508, 2012.

[4] Jet Propulsion Laboratory. Xenon hall effect. `https://fr.wikipedia.org/wiki/Propulseur_%C3%A0_effet_Hall#/media/Fichier:Xenon_hall_thruster.jpg`, May 2007. Consulted August 2020.

[5] Jet Propulsion Laboratory. Xenon hall thruster. `https://www.haveeru.com.mv/electrical-corridor-thruster-for-nasas-psyche-spacecraft-propulsion/`, July 2020. Consulted August 2020.

[6] F. Chen. *Introduction to plasma physics and controlled fusion.* Plenum Press, 1984.

[7] J. C. Adam et al. Physics, simulation and diagnostics of hall effect thrusters. *Plasma Physics and Controlled Fusion*, 50:124041, 2008.

[8] A. Ducrocq et al. High-frequency electron drift instability in the cross-field configuration of hall thrusters. *Physics of Plasmas*, 13:102111, 2006.

[9] A. Héron and J.C. Adam. Anomalous conductivity in hall thrusters: Effects of the non-linear coupling of the electron-cyclotron drift instability with secondary electron emission of the walls. *Physics of Plasmas*, 20:082313, 2013.

[10] V. N. Duarte and R.A. Clemente. Electron inertia effect on the planar plasma sheath problem. *Physics of Plasmas*, 18(4):043504, 2011.

[11] G.M.W. Kroesen et al. Particle charge and electrical field studied in the pre-sheath of an rf discharge. Technical report, ESA, 2011.

[12] Von. M. A. Liebermann and A. J. Lichtenberg. *Principles of plasma discharges and material processing.* Wiley, 1995.

[13] Alexander Piel. *Plasma Physics.* Springer, 2010.

[14] A. Alvarez Laguan et al. An asymptotic preserving well balanced scheme for the isothermal fluid equations in low-temperararature plasma applications. *hal-021166100*, 2019.

[15] E. M. Sousa and U. Shumlack. A blended continuous-discontinuous finite element method for solving the multi-fluid plasma model. *J. Comput. Phys.*, 326:56-75, 2016.

[16] U. Shumlack et al. Advanced physics calculations using a multi-fluid plasma model. *Computer Physics Communications*, 182(9):1767-1770, 2010.

[17] U. Shumlack and J. Loverich. Approximate riemann solver for the two-fluid plasma model. *J. Comput. Phys.*, 187(2):620-638, 2003.

[18] J. Loverich A. Hakim and U. Shumlack. A discontinuous galerkin method for ideal two fluid plasma equation. *Communications in in Computationnal Physics 9*, 240, 2011.

[19] J. Loverich A. Hakim and U. Shumlack. A high resolution wave propagation scheme for ideal two-fluid plasma equations. *J. Comput. Phys.*, 219(1):418-442, 2006.

[20] A. Laguna et al. Fully-implicit finite volume method for the ideal two-fluid plasma model. *Computer Physics Communications*, 231:31-44, 2018.

[21] A. Laguna et al. A versatile numerical method for the multi-fluid plasma model in partially- and fully-ionized plasmas. *Journal of Physics: Conference Series*, 1031(1):012015, 2018.

[22] I. A. Asensio et al. A gpu-enabled implicite finite volume solver for the ideal two-fluid plasma model on unstructured grids. *Computer Physics Communications*, 239:16-32, 2019.

[23] S. Gottlieb and C.-W. Shu. Total variation diminishing runge-kutta schemes. *Math. Comput.*, 67(221):73-85, 1998.

[24] Chapman and Coiling. *Mathematical Theory of Nonuniform Gases.* Cambridge University Press, 1953.

[25] S. I. Braginskii. Transport processes in a plasma. *Reviews of Plasma Physics*, 1:205, 1965.

[26] Lewi Tonks and Irving Langmuir. A general theory of the plasma of an arc. *Phys. Rev.*, 34:876-922, 1929.

[27] K-U Riemann et al. The plasma-sheath matching problem. *Plasma Physics and Controlled Fusion*, 47(11):1949, 2005.

[28] D. Bohm. *The Characteristics of Electrical Discharges in Magnetic Fields.* McGraw-Hill, 1949.

[29] P. Chabert and N. Braithwhaite. *Physics of radio-frequency plasmas.* Cambridge University Press, 2011.

[30] I. Langmuir. The interaction of electron and positive ion space charges in cathode sheaths. *Phy. Rev. Let.*, 33(1):954-989, 1923.

[31] C. D. Child. Discharge from hot cao. *Phy. Rev. Let.*, 32(1):492-511, 1911.

[32] P. Degong et al. Analysis of an asymptotic preserving scheme for the euler-poisson system in the quasi neutral limit. *SIAM Journal on Numerical Analysis*, 46(3):1298-1322, 2008.

[33] G. E. Karniadakis B. Cockburn and C.-W. Shu. *Discontinuous Galerkin Methods - Theory, Computation and Applications, volume 11 of Lecture Notes in Computer Science and Engineering.* Springer, 2000.

[34] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. *Technical Report LA-UR-73-0479, Los Alamos Scientific Laboratory.*, 1973.

[35] American Mathematical Society. Mathscinet. https://mathscinet.ams.org/mathscinet/. Accessed on August 2020.

[36] I. G. Bubnov. Report on the works of professor timoshenko which were awarded the zhuran-skyi prize. *Symposium of the Institute of Communication Engineers, All Union Special Planning Office (SPB)*, 81, 1913.

[37] E. Waring. Problems concerning interpolations. *Philosophical Transactions of the Royal Society*, 69:59-67, 1779.

[38] K. F. Gauss. *Methodus nova integralium valores per approximationem inveniendi.* Göttingen, Heinrich Dietrich, 1815.

[39] A. Ern D. A. Di Pietro. *Mathematical Aspects of Discontinuous Galerkin Methods.* Springer, 2012.

[40] P. L. Roe and. Approximate riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357-372, 1981.

[41] R. J. Levêque. *Numerical Methods for Conservative Laws.* Birkhaüser, 1992.

[42] C. Runge. Über die numerische auflösung von differentialgleichungen. *Math. Ann.*, 46:167-192, 1895.

[43] W. Kutta. Beitrag zur näherungsweisen integration totaler differentialgleichungen. *Zeitschr. für Math. u. Phys.*, 46:435-453, 1901.

[44] J. C. Butcher. Coefficients for the study of runge-kutta integration processes. *Journal of the Australian Mathematical Society*, 3(2):185-201, 1963.

[45] B. Cockburn and C.-W. Shu. Runge-kutta discontinuous galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16:173-261, 2001.

[46] J. Raphson. *Analysis aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, et expedita, ex nove infinitarum serierum doctrina deducta ac demonstrata.* Microfilm Copy: University Microfilms, Ann Abor, MI, 1690.

[47] P. Crispel et al. Trois formulations d'un modèle de plasma quasi-neutre. *Comptes Rendus. Mathématique. Académie des Sciences, Paris*, 4, 02 2004.

[48] P. Crispel et al. Quasi-neutral fluid model for current carrying plasmas. *Journal of Computational Physics*, 205(2):408-438, 2005.

[49] I. Babuska. The finite element method with penalty. *Math. Comp.*, 27:221-228, 1973.

[50] M.-S. Liou. A sequel to ausm, part ii: Ausm$^+$-up for all speeds. *Journal of Computational Fluid Dynamics*, 214:137-170, 2006.

[51] Tux Family and inria. Eigen. `http://eigen.tuxfamily.org/index.php?title=Main_Page`. Accessed on August 2020.

[52] OpenMP. Openmp. `https://www.openmp.org/`. Accessed on August 2020.

[53] J. R. Bunch and J. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28 (125): 231–236, 1974.

[54] E. Huckel P. Debye. The theory of electrolytes. i. lowering of freezing point and related phenomena. *Phys. Z.*, 24:185-206, 1923.